Markov Processes for Reliability Analyses of Large Systems

Ioannis A. Papazoglou, Student member IEEE
Elias P. Gyftopoulos

Key Words – Time-dependent reliability and availability, Markov Systems, Mergeable Markov chains.

Reader Aids –
Purpose: Tutorial
Special math needed for explanations & results: Matrix algebra
Results useful to: Reliability theoreticians and engineers

Abstract – This paper presents a methodology for calculating the time-dependent reliability of a large system consisting of *s*-dependent components. A Markov-chain model is used and the numerical difficulties associated with large transition-probability matrices are reduced by a systematic ordering of the system states. A technique is also presented for the systematic merging of processes corresponding to systems exhibiting symmetries.

I. INTRODUCTION

The purpose of this paper is to present two techniques for reducing numerical difficulties in calculating time-dependent availability and reliability of a large system consisting of *s*-dependent components. In reliability analyses of engineering systems, *s*-dependencies among either failures or repairs or both must be considered. Such *s*-dependencies are introduced both by common-cause failures, and by maintenance procedures that are contingent on the state of the components, on the state of the system, and on the test method [1-3]. Many aspects of *s*-dependence can be analyzed more easily if the probabilistic behavior of the system can be simulated by a Markov process [4-10]. For large systems with many states, however, existing methods are not practical because of numerical difficulties. Of course methods for analyzing large systems consisting of *s*-independent components are available, e.g. [11].

After describing Markov-process simulation, we present two techniques each of which reduces the numerical complexity of the method. The techniques are:

1) Systematic ordering of operating and failed states. Zelentsov [12] has used a similar technique to calculate the mean time-to-failure of a large nonrepairable system consisting of 2-state components. Here, we use the ordering technique to calculate the time-dependent availability and reliability of a large repairable system consisting of components each of which can be in more than two states.

2) Mergeability of a Markov process. Several authors have discussed the criterion for a Markov process to be mergeable [7, 13 - 15]. Here, we present a systematic procedure for merging the process of systems exhibiting certain symmetries.

The body of the paper is organized as follows:

Section 3. Review of simulation of a system by a Markov process.
Section 4. Systematic ordering of states.
Section 5. Analytic formulation of specific repair policies.
Section 6. Mergeability of Markov processes.
Section 7. Calculations of time-dependent unreliability and unavailability of a simple system.
Appendix. Description of computer code used in the calculations.

2. NOTATION

N	number of components of the system
M	maximum number of failed components with which the system can still operate
k_v	number of states of component v
Z	set of all system-states
z	number of system-states, $z = \Pi_{v=1}^{N} k_v$
Δt	constant time interval between successive time-points
n	implies time-point n; $n = 0, 1, 2, \ldots$
$s(n)$	state of system at time-point n, $s = 1, 2, \ldots, z$
i, j	indices for system-state; both go from 1 to z
$h_{v;rg}(n\|s(n) = i)$	transition rate of component v from component-state r to component-state g at time-point n, given $'s(n) = i'$
$p_{ij}(n)$	transition probability from system-state i to system-state j
	$0 \leqslant p_{ij}(n) \leqslant 1, \ \Sigma_{j=1}^{z} p_{ij}(n) = 1$
X	subset of operating system-states
Y	subset of failed system-states
B, B'	general subsets; either can be X or Y
$B(K)$	subset of B which contains system-states with K failed components
$\pi_i(n, B)$	system-state probability at time-point n for $i \in B$; see note 1 below.
	$0 \leqslant \pi_i(n) \leqslant 1, \ \Sigma_{i=1}^{z} \pi_i(n) = 1$
$\underset{\sim}{\pi}(n, B)$	row vector with elements $\pi_i(n, B)$; see note 1 below
$\pi_i^K(n, B)$	system-state probability at time-point n for $i \in B(K)$
$\underset{\sim}{\pi}^K(n, B)$	row vector with elements $\pi_i^K(n, B)$

$\underset{\sim}{P}(n, B, B')$ matrices with elements $p_{ij}(n)$ from system-states of set B to system-states of set B'; see note 1 below

$\underset{\sim}{P}^{KL}$ submatrix with elements $p_{ij}(n)$ from system-states of subset $B(K)$ to system-states of subset $B'(L)$. The B, B' dependence will be evident from the context.

$p_{S;IJ}(n)$ transition probability from superstate I to superstate J at time-point n.

$a_{r,m}$ number of components of class r in component-state m

$A(n)$ availability: probability that the system will be operating at time-point n.

$R(n)$ reliability: probability that the system will be operating continuously from time-point 0 to n inclusive

Note 1. When B or B' is not specified, the entire set Z is implied.

3. MARKOV CHAIN MODEL

Assumptions

1) The state of the system is determined by the states of the N components.

2) The system changes its state according to a discrete-state, discrete-time Markov process; therefore, $\underset{\sim}{\pi}(n)$ obeys the relation [4-7]

$$\underset{\sim}{\pi}(n+1) = \underset{\sim}{\pi}(n) \cdot \underset{\sim}{P}(n) \tag{1}$$

3) The size of the time-step Δt is such that transition probabilities among system-states differing in the states of more than one component can be neglected. Thus,

$$p_{ij}(n) = \begin{cases} h_{v,rg}(n|s(n)=i) \cdot \Delta t & \text{if } i \neq j \text{ and if system-states } i, j \text{ differ only in the component-state of component v.} \\ 0 & \text{if } i \neq j \text{ and if system-states } i, j \text{ differ in the component-states of more than one component.} \\ 1 - \sum_{\substack{m=1 \\ m \neq i}}^{z} p_{im}(n) & \text{if } i = j. \end{cases} \tag{2}$$

The transition rate of each component at time n depends on the state of the system, namely on the states of other components, so common-cause failures are allowed. For example, the common-cause failure of two components can be modeled by assuming a certain failure rate when both components are operating and a properly higher failure rate when only one component is operating (see also Sec. 7).

Reliability and Availability. $\underset{\sim}{\pi}(n)$ contains the information for calculating reliability and availability. Eq. (1) can be rewritten as

$$[\underset{\sim}{\pi}(n+1, X), \underset{\sim}{\pi}(n+1, Y)] = [\underset{\sim}{\pi}(n, X), \underset{\sim}{\pi}(n, Y)] \cdot \begin{bmatrix} \underset{\sim}{P}(n, X, X) & \underset{\sim}{P}(n, X, Y) \\ \underset{\sim}{P}(n, Y, X) & \underset{\sim}{P}(n, Y, Y) \end{bmatrix}. \tag{3}$$

The probability that the system will be operating at n is the probability that the system will occupy any of the operating states at n. Thus the availability is

$$A(n) = \sum_{i \in X} \pi_i(n, X). \tag{4}$$

The probability that 'the system will not leave the subset of operating states X during the time period from 0 to n inclusive' is the probability that 'the system will be in X at n, given that transitions from Y back to X are not possible'. Thus the reliability is

$$R(n) = \sum_{i \in X} \pi_i^*(n, X) \tag{5}$$

where now $\underset{\sim}{\pi}^*(n, X)$ is the solution of (3) with $\underset{\sim}{P}(n, Y, X) = \underset{\sim}{0}$.

4. ORDERING OF STATES

Solving (3) requires the aid of a computer. When the number of possible system-states is large, however, the necessary computer storage and computer time are prohibitive because of the large size of the transition probability matrix. The computational effort can be reduced by ordering the system-states and using Assumption 3. System-states are ordered by partitioning X and Y into subsets $X(K)$, for $K = 0, 1, 2 \ldots, M$, and $Y(K')$, for $K' = 1, 2, \ldots, N$, respectively. X and Y are represented by the unions

$$X = X(0) \cup X(1) \ldots \cup X(M), \tag{6a}$$

$$Y = Y(1) \cup Y(2) \ldots \cup Y(M) \ldots \cup Y(N) \tag{6b}$$

Similarly, we can order $\underset{\sim}{\pi}(n, X)$ and $\underset{\sim}{\pi}(n, Y)$ into subvectors, and $\underset{\sim}{P}(n, X, X)$, $\underset{\sim}{P}(n, X, Y)$ $\underset{\sim}{P}(n, Y, X)$, and $\underset{\sim}{P}(n, Y, Y)$ into submatrices corresponding to the various $X(K)$ and $Y(K)$. Thus (3) becomes

$$[\underset{\sim}{\pi}^0(n+1, X), \ldots, \underset{\sim}{\pi}^N(n+1, Y)] = [\underset{\sim}{\pi}^0(n, X), \ldots, \underset{\sim}{\pi}^N(n, Y)] \begin{bmatrix} [\underset{\sim}{P}^{IJ}]_{XX} & [\underset{\sim}{P}^{IL}]_{XY} \\ [\underset{\sim}{P}^{KJ}]_{YX} & [\underset{\sim}{P}^{KL}]_{YY} \end{bmatrix} \tag{7}$$

where $I, J = 0, 1, 2, \ldots, M$, and $K, L = 1, 2, \ldots, N$. Moreover, by virtue of (2) it follows that $\underset{\sim}{P}(n)$ in (7) has the form [10]

$$\underset{\sim}{P}(n) = \left[\begin{array}{cc|cc} \underset{\sim}{P}^{00} \underset{\sim}{P}^{01} \underset{\sim}{0} \ldots \underset{\sim}{0} & \underset{\sim}{P}^{01} \underset{\sim}{0} \underset{\sim}{0} \ldots \underset{\sim}{0} \\ \underset{\sim}{P}^{10} \underset{\sim}{P}^{11} \underset{\sim}{P}^{12} \ldots \underset{\sim}{0} & \underset{\sim}{0} \underset{\sim}{P}^{12} \underset{\sim}{0} \ldots \underset{\sim}{0} \\ \vdots \quad \ddots \quad \ddots & \vdots \quad \ddots \quad \ddots \\ \underset{\sim}{0} \ldots \underset{\sim}{P}^{MM} & \underset{\sim}{0} \ldots \underset{\sim}{P}^{M,M+1} \ldots \underset{\sim}{0} \\ \hline \underset{\sim}{P}^{10} \underset{\sim}{0} \underset{\sim}{0} \ldots \underset{\sim}{0} & \underset{\sim}{P}^{11} \underset{\sim}{P}^{12} \underset{\sim}{0} \ldots \underset{\sim}{0} \\ \underset{\sim}{0} \underset{\sim}{P}^{21} \underset{\sim}{0} \ldots \underset{\sim}{0} & \underset{\sim}{P}^{21} \underset{\sim}{P}^{22} \underset{\sim}{P}^{23} \ldots \underset{\sim}{0} \\ \vdots \quad \ddots \quad \ddots & \vdots \quad \ddots \quad \ddots \\ \underset{\sim}{0} \ldots \underset{\sim}{P}^{M+1,M} & \underset{\sim}{0} \ldots \underset{\sim}{P}^{M+1,M+1} \ldots \underset{\sim}{0} \\ \vdots \quad \ddots & \vdots \quad \ddots \quad \ddots \\ \underset{\sim}{0} \ldots \underset{\sim}{0} & \underset{\sim}{0} \ldots \underset{\sim}{P}^{NN} \end{array}\right] \tag{8}$$

where $I, J = 0, 1, \ldots, M$, and $K, L = 1, 2, \ldots, N$, and, for convenience the time and subset dependence of the submatrices of matrices $\underset{\sim}{P}(n, X, X)$ etc. have been omitted from both (7) and (8).

We see from (7) and (8) that the ordering of states reduces the numerical complexity of the problem in a systematic manner. For example, the structure of submatrix $\underset{\sim}{P}(n, X, Y)$ in (8) indicates that transitions from an operating state with I failed components to a failed state with L failed components is not possible if:

1) $|I - L| > 1$, namely if more than one component-state transitions must occur;
2) $I = L + 1$, namely if a failed component is reparied, since such a repair in an operating state cannot bring the system into a failed state.

Again, (8) indicates that only $5M + 3N + 1$ submatrices of the ordered $\underset{\sim}{P}(n)$ need be stored instead of the $(M + N + 1)^2$ submatrices of the unordered $\underset{\sim}{P}(n)$. Moreover, the ordering results in computing-time savings because solving (3) is much faster when $\underset{\sim}{P}(n)$ is ordered than when it is not.

5. REPAIR POLICIES AND SPECIAL SYSTEMS

For certain repair policies and certain special systems some of the submatrices $\underset{\sim}{P}^{KL}$ in (8) are zero. Four examples are given below.

a) *No-online repair*: If online repair is not possible, then submatrices $\underset{\sim}{P}^{IJ}$ of the lower diagonal stripe of $\underset{\sim}{P}(n, X, X)$ in (8), are zero:

$$\text{in } \underset{\sim}{P}(n, X, X),\ \underset{\sim}{P}^{I+1,I} = \underset{\sim}{0} \text{ for } 0 \leqslant I \leqslant M - 1. \tag{9}$$

b) *"Cold" standby operation*: If standby operation of a system is assumed cold (no components can fail while the system is not operating), then submatrices $\underset{\sim}{P}^{KL}$ of the upper diagonal stripe of $\underset{\sim}{P}(n, Y, Y)$ in (8) are zero:

$$\text{in } \underset{\sim}{P}(n, Y, Y),\ \underset{\sim}{P}^{K-1,K} = \underset{\sim}{0} \text{ for } 2 \leqslant K \leqslant N. \tag{10}$$

c) *Selective repair*: In general, if a system is failed, the first component to be repaired can be any of the failed components. Under a selective repair policy, however, it is possible to repair that particular component which brings the system back into operation. When a selective repair policy is possible, then we have in (8):

$$\text{in } \underset{\sim}{P}(n, Y, Y),\ \underset{\sim}{P}^{K+1,K} = \underset{\sim}{0} \text{ for } K < M + 1. \tag{11}$$

d) *Components with one operating state*: If a system consists of components that cannot transit between failed states and have only one operating state, then submatrices $\underset{\sim}{P}^{II}$ of $P(n, X, X)$ and $\underset{\sim}{P}^{KK}$ of $\underset{\sim}{P}(n, Y, Y)$ are diagonal because a system transition from a given state to another with the same number of failed components requires at least the simultaneous repair and failure of two different components:

$$\begin{aligned} &\text{in } \underset{\sim}{P}(n, X, X),\ \underset{\sim}{P}^{II} = [\delta_{ij} p_{ij}(n)]^{II} \text{ for } I = 0, 1, 2, \ldots, M \\ &\text{in } \underset{\sim}{P}(n, Y, Y),\ \underset{\sim}{P}^{KK} = [\delta_{ij} p_{ij}(n)]^{KK} \text{ for } K = 1, 2, \ldots, N \end{aligned} \tag{12}$$

where δ_{ij} is the Kronecker delta ($\delta_{ij} = 1$ if $i = j$, $\delta_{ij} = 0$ otherwise).

6. MERGEABLE MARKOV PROCESS

The computational effort involved in evaluating reliability and availability can be substantially reduced if values of partial sums of $\pi_i(n, X)$'s rather than individual values of $\pi_i(n, X)$'s can be obtained. Such partial sums can be calculated for systems for which the Markov process is mergeable [8, 10, 13, 15]. A Markov process is mergeable if:

1) its states can be divided into groups each of which forms a superstate;
2) its transition probabilities are sufficient to express the transition probabilities among superstates.

Because the number of superstates of a mergeable Markov process is much smaller than the number of states, the dimensions of the probability vector and the transition probability matrix of the superstates are much smaller than those of the states, and therefore the computational effort is reduced.

A Markov process is mergeable if, for all superstates I and J consisting of r states indexed $i = 1, 2, \ldots, r$ and m states indexed $j = 1, 2, \ldots, m$, respectively, we have

$$p_{s;IJ}(n) \equiv \sum_{j \in J} p_{ij}(n) \text{ for all values of } i. \tag{13}$$

In general, the grouping of states into superstates, and the verification of (13) is a prohibitively time consuming procedure [7, 15]. For systems exhibiting symmetries, however, the procedure is expedited as described immediately below.

We consider systems consisting of components that can be grouped into classes so that the functions, number of states, conditional failure rates, and conditional repair rates of each

component in a class are the same as those of any other component in the class.

To simplify the notation and without loss of generality, we will discuss a system consisting of components that can be grouped into two classes, 1 and 2, and that can be found in either of two states, operating state 1 and failed state 2. For this system, all states that have $a_{r,m}$ components of class r in state m, and K components in failed states ($K = a_{1,2} + a_{2,2}$) are labeled by $T_\nu(K)$ given by the one-dimensional array

$$T_\nu(K) = \{a_{1,1}, a_{1,2}, a_{2,1}, a_{2,2}\} \text{ for } \nu = 1, 2, \ldots, \nu(K) \quad (14)$$

where $\nu(K)$ is the number of possible labels for each value of K. Thus, we can assign a $T_\nu(K)$ to each state of the system for $K = 0, 1, 2, \ldots, N$ and $\nu = 1, 2, \ldots, \nu(K)$. Moreover, for given values of K and ν and, therefore, for a given $T_\nu(K)$, we can group states labeled $T_\nu(K)$ into a superstate. Now, we will show that these superstates, for $K = 0, 1, 2, \ldots, N$ and $\nu = 1, 2, \ldots, \nu(K)$, satisfy (13).

1) Because only one component-transition per time step is possible, the only superstates that can be reached from a superstate $I_\nu(K)$, labeled $T_\nu(K)$, via failure of a component are superstates $I_\rho(K + 1)$ and $J_\mu(K + 1)$, labeled $T_\rho(K + 1)$ and $T_\mu(K + 1)$, where,

$$T_\rho(K+1) = \{a_{1,1} - 1, a_{1,2} + 1, a_{2,1}, a_{2,2}\} \quad (15a)$$

$$T_\mu(K+1) = \{a_{1,1}, a_{1,2}, a_{2,1} - 1, a_{2,2} + 1\} \quad (15b)$$

2) Because all components in class 1 perform the same function, states labeled $T_\rho(K + 1)$ will be either all operating or all failed, and the same will be true for states labeled $T_\mu(K + 1)$. Assuming that states labeled $T_\rho(K + 1)$ and $T_\mu(K + 1)$ are operating and failed, respectively, we conclude that from any state of operating superstate $I_\nu(K)$ the system can transit to $a_{1,1}$ states of operating superstate $I_\rho(K + 1)$, and to $a_{2,1}$ states of failed superstate $J_\mu(K + 1)$.

3) Because all components in a class have the same transition rates, the transition probabilities from all states i and i' of superstates $I_\nu(K)$ satisfy the relations

$$\sum_{j \in I_\rho(K+1)} p_{ij}(n) = \sum_{j \in I_\rho(K+1)} p_{i'j}(n) =$$

$$a_{1,1} h_{1;12}[n|S(n) = I_\nu(K)]\,\Delta t \quad (16a)$$

$$\sum_{j \in J_\mu(K+1)} p_{ij}(n) = \sum_{j \in J_\mu(K+1)} p_{i'j}(n) =$$

$$a_{2,1} h_{2;12}[n|S(n) = I_\nu(K)]\,\Delta t \quad (16b)$$

4) Reasoning as in steps 2 and 3 immediately above, we also conclude that from any state of superstate $I_\rho(K + 1)$ the system can transit to $a_{1,2} + 1$ states of superstate $I_\nu(K)$, and that

$$\sum_{i \in I_\nu(K)} p_{ki}(n) = \sum_{i \in I_\nu(K)} p_{k'i}(n) =$$

$$(a_{1,2} + 1) h_{1;21}[n|S(n) = I_\rho(K+1)]\,\Delta t \quad (17a)$$

$$\sum_{i \in I_\nu(K)} p_{\mu i}(n) = \sum_{i \in I_\nu(K)} p_{\mu' i}(n) =$$

$$(a_{2,2} + 1) h_{2;21}[n|S(n) = J_\mu(K+1)]\,\Delta t \quad (17b)$$

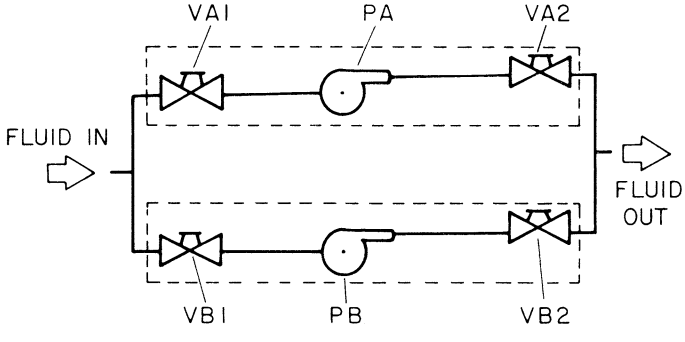
Eqs. (16) and (17) are formal statements that (13) is satisfied, and therefore that the process is mergeable.

An analogous systematic labeling and grouping of states can be used for the merging of the Markov process of a system consisting of components that can be grouped in more than two classes and that can be found in more than two states. The formal proof, however, will not be given here [16].

If the functions of all the components in a class are not the same, then states $T_\rho(K + 1)$ that can be reached from states $T_\nu(K)$ are not all operating or failed. Here, a number of superstates formed by means of the systematic procedure just cited do not satisfy (13). To proceed with the merging, we must consider a number of superstates larger than that implied by the labeling $T_\nu(K)$. This is accomplished on the computer by means of a code that generates superstates labeled $T_\nu(K)$ and that proceeds with alternate groupings until (13) is satisfied. In general, with a reasonable definition of classes of components, a few trials suffice to merge the process.

7. AN APPLICATION

To illustrate the methodology developed in Sections 3 to 6, we have calculated the time-dependent unavailability and unreliability of the system shown in Figure 1. The following assumptions were made about the system:



SAMPLE SYSTEM

Fig. 1.

1) The system consists of two pumps and four valves. Each pump can supply the required flow rate, but when both are operating each is operating at half capacity. The pumps can be in two states: operating and failed. Two valves are associated with each pump. The function of the valves is to

isolate the corresponding pump when it fails. Each valve can be in three states: operating, failed in the open position, and failed in the closed position.

2) The mission of the system is to supply point B with water at a certain flow rate, for a certain period of time T, and under a known environment.

3) The failure and repair rates are listed in Table I. They do not correspond to real data but have been selected solely for illustration. The following s-dependencies due to different operating conditions, repair capabilities, and possible common cause failures, have been assumed:

a) The failure rate of each pump is $k_{pi}\lambda_p (i = 0, 1)$, where i is the number of failed pumps and $k_{p0} = 1$. Similarly, the failure rate of each valve is $k_{vi}\lambda_{vj}$ ($i = 0, 1, 2, 3$ and $j = 1, 2$), where i denotes the number of failed valves, j the failure mode (open or closed position), and $k_{v0} = 1$;

b) Every repair is perfect, viz. restores the component to like-new and affects nothing else. The repair rate of each pump is $d_{pi}r_p (i = 1, 2$, and $d_{p1} = 1)$, where i denotes the number of failed pumps. The repair rate of each valve is $d_{vi}r_{vj}$ ($i = 1, 2, 3, 4; j = 1, 2$; and $d_{v1} = 1$).

A computer code, described briefly in the Appendix and given in full in the Supplement [17], calculates according to the methodology developed in Sections 4 and 5. The results are summarized in Figures 2 and 3. In Figure 2, curves 1 and 2 represent the unavailability of the system as a function of time with or without s-dependencies among the failures and repairs, respectively, and curve 3 represents the unavailability of the system if on-line repair is not possible. Figure 3 presents analogous results for the unreliability of the system.

Because the system consists of components that can be divided into two classes, one containing the two pumps and the other the four valves, and because the components of each class have similar functions, can be in the same number of states, and have the same conditional failure and repair rates, the Markov process that describes the system is mergeable (Sec. 6). The savings in computer storage and computer time resulting from the merging are listed in Table II.

TABLE I
Conditional failure and repair rates of the components of system in Figure 1

	Conditional failure rates (per 10^6 hours)		Conditional repair rates (per 10^6 hours)	
Pumps				
Two Up	30			
One Up	3000		10000	
None Up			5000	
Valves	to the 'open position'	to the 'closed position'	from the 'open position'	from the 'closed position'
Four Up	1	1	1000	1000
Three Up	5	5	500	500
Two Up	10	10	300	300
One Up	100	100	300	300
None Up			100	100

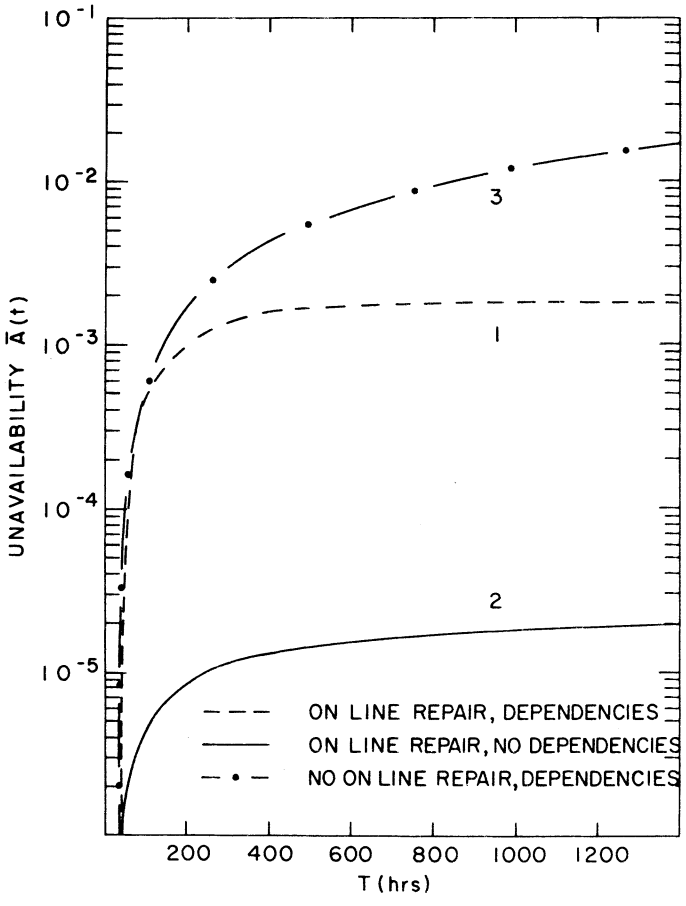


Fig. 2.

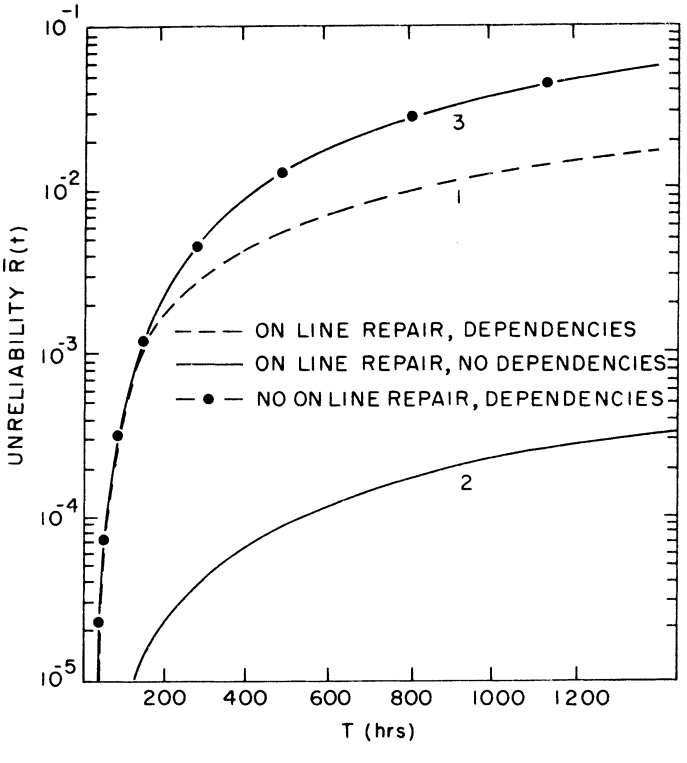


Fig. 3.

TABLE II
Comparison of the computer storage and computer time required for solving the original and merged Markov processes for system in Figure 1

	Number of States	Number of Elements of $\underset{\sim}{P}$ (a)	Number of Elements of $\underset{\sim}{P}$ that need be stored (a)	Time Required for generation of $\underset{\sim}{P}$ (sec) (b)	Time Required for Solution of (7) for 300 time steps (sec) (b)	Total time (sec) (b)
Original Process	287	82 369	17 676	1.0	13.0	14.0
Merged Process	67	4 489	967	1.7	0.8	2.5

a) With zero standby failure rates (Sections 5b-5d).
b) CDC, CYBER 70/ MODEL 76 Computer System (7600).

APPENDIX

A brief description of the code is presented here. For further details, please refer to the Supplement [17].

1. The input consists of the number of components, the number of states of each component, the transition rates, and a test subroutine (see step 3).
2. All possible states of the system are generated as a 2-dimensional array, each row represents a system-state and each element in the row a component-state.
3. With the aid of a 'test subroutine', the set of possible system-states is partitioned into subsets X and Y. The subroutine can be a fault tree or an event tree for the system under analysis and is part of the input.
4. X and Y are partitioned into subsets $X(K)$ and $Y(K)$ for all K.
5. $\underset{\sim}{P}$ is generated taking into account any repair policy or any special feature of the system (Sec. 5). The p_{ij}'s are calculated by comparing states i and j by determining the component that has changed state, and by examining the s-dependence of the conditional transition rate on the states of other components.
6. Eqs. (5) and (6) are solved over the required time period.

A variation of this code has also been written for systems that can be described by mergeable Markov processes. The number of classes and the number of components in each class are required as additional inputs. After the formation of subsets $X(K)$ and $Y(K)$, the code assigns a label $T_\nu(K)$ to each state and groups states into superstates as described in Section 6. The transition probabilities among superstates are calculated by means of (13).

The code is written in Fortran-IV and consists of about 1000 statements. The version for the mergeable Markov processes consists of 1500 Fortran statements. Both run on CDC 6000 or 7600 computer systems but the necessary adjustments for usage on IBM or other machines are minor. More information about the code as well as a source listing can be found in [17].

REFERENCES

[1] C.P. Neuman, N.M. Bonhomme, "Evaluation of maintenance policies using Markov chains and fault tree analysis", *IEEE Trans. Reliability*, vol R-24, 1975 Apr, pp 37-44.
[2] W.G. Gangloff, "Common mode failure analysis", *IEEE Trans. PAS.*, vol PAS-94, 1975 Jan/Feb, pp 27-30.
[3] B.J. Flehinger, "A Markovian model for the analysis of the effects of marginal testing on system reliability", *An. Math. Stat.*, vol 33, 1962 June, pp 754-766.
[4] R. Billinton, R.J. Ringlee, A.J. Wood, *Power-system Reliability Calculation*, MIT Press (1973).
[5] R. Barlow, F. Proschan, *Mathematical Theory of Reliability*, John Wiley & Sons, 1965.
[6] M.L. Shooman, *Probabilistic reliability; an engineering approach*, McGraw-Hill, Inc. 1968.
[7] R.A. Howard, *Dynamic Probabilistic Systems*, vol 1, John Wiley & Sons 1971.
[8] J.A. Buzacott, "Markov approach to finding failure times of repairable systems", *IEEE Trans. Reliability*, vol 19, 1970 Nov, pp 128-134.
[9] G.T.H. Lee, "Mean-times of interest in Markovian systems", *IEEE Trans. Reliability*, vol R-20, 1971 Feb, pp 16-21.
[10] I.A. Papazoglou, "Markovian analysis of reliability of nuclear reactor systems", SM thesis, M.I.T., Nucl. Eng. Dept., 1974 May.
[11] P.M. Lin, et al., "A new algorithm for symbolic system reliability analysis", *IEEE Trans. Reliability*, vol R-25, 1976 Apr, pp 2-14.
[12] B.P. Zelentsov, "Reliability analysis of large nonrepairable systems", *IEEE Trans. Reliability*, vol R-19, 1970 Nov, pp 132-136.
[13] G.C. Bacon, "The decomposition of stochastic automats", *Inform. & Contr.*, vol 7, 1964 Sept, pp 320-339.
[14] G. Singh, R. Billinton, "Frequency and duration concepts in system reliability evaluation", *IEEE Trans. Reliability*, vol R-24, 1975 Apr, pp 31-36.
[15] J.G. Kemeny, J.L. Snell, *Finite Markov Chains*, D. Van Nostrand Co. 1960.
[16] I.A. Papazoglou, "Markovian reliability analysis under uncertainty", ScD thesis, M.I.T., Nucl. Eng. Dept., 1977 Sep.
[17] Supplement: NAPS document Nos. 02972 & 02973; 136 pages total in these Supplements. For current ordering information see inside rear cover of a current issue. ASIS-NAPS; Microfiche Publications; POBox 3513, Grand Central Station; New York, NY 10017 USA.

BIOGRAPHICAL SKETCH

Ioannis A. Papazoglou; MIT Room 24-109; Cambridge, MA 02139 USA.

Ioannis A. Papazoglou (S'77) was born in Athens, Greece, on 1949 Dec 30. He received his Diploma in Mechanical and Electrical Engineering from the National Technical University of Athens in 1972, and his SM degree in Nuclear Engineering from the Massachusetts Institute of Technology. He is a candidate for the ScD degree in Nuclear Engineering at M.I.T. and is doing his doctoral research at Brookhaven National Laboratory where he is associated with the Fast Reactor Safety Division.

Elias P. Gyftopoulos; MIT Room 24-109; Cambridge, MA 02139 USA.

Elias P. Gyftopoulos was born in Athens, Greece, on 1927 Jul 4. He received his Diploma in Mechanical and Electrical Engineering from the National Technical University of Athens in 1953, and his ScD degree in Electrical Engineering from the Massachusetts Institute of Technology. He is Ford Professor of Engineering of the Nuclear Engineering Department at M.I.T. □□□

Manuscript received 1975 December 6; revised 1977 May 26.