AD-A132 639



AD

DRSMC/SA/MR-1

# METHODS FOR CHOOSING BUFFER SIZE IN TANDEM PRODUCTION OPERATIONS

GEORGE J. SCHLENKER

AUGUST 1983

Distribution Statement

Approved for public release; distribution unlimited.



US ARMY ARMAMENT, MUNITIONS AND CHEMICAL COMMAND
SYSTEMS AND OPERATIONS ANALYSIS DIRECTORATE
ROCK ISLAND, ILLINOIS 61299

## DISPOSITION

Destroy this report when no longer needed. Do not return it to the originator.

## DISCLAIMER

The findings in this report are not to be construed as an official Department of the Army position.

## WARNING

Information and data contained in this document are based on input available at the time of preparation. Because the results may be subject to change, this document should not be construed to represent the official position of the US Army Armament, Munitions and Chemical Command unless so stated.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

| REPORT DOCUMENTATION PAGE | | READ INSTRUCTIONS BEFORE COMPLETING FORM |
|---|---|---|
| 1. REPORT NUMBER<br>DRSMC/SA/MR-1 | 2. GOVT ACCESSION NO. | 3. RECIPIENT'S CATALOG NUMBER |
| 4. TITLE (and Subtitle)<br>Methods for Choosing Buffer Size in Tandem Production Operations | | 5. TYPE OF REPORT & PERIOD COVERED<br>Report-Final<br>6. PERFORMING ORG. REPORT NUMBER |
| 7. AUTHOR(s)<br>George J. Schlenker | | 8. CONTRACT OR GRANT NUMBER(s) |
| 9. PERFORMING ORGANIZATION NAME AND ADDRESS<br>US Army Armament, Munitions and Chemical Command<br>Systems and Operations Analysis Directorate<br>Rock Island, IL 61299 | | 10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS |
| 11. CONTROLLING OFFICE NAME AND ADDRESS<br>US Army Armament, Munitions and Chemical Command<br>Systems and Operations Analysis Directorate<br>Rock Island, IL 61299 | | 12. REPORT DATE<br>August 1983<br>13. NUMBER OF PAGES<br>78 |
| 14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office) | | 15. SECURITY CLASS. (of this report)<br>UNCLASSIFIED<br>15a. DECLASSIFICATION/DOWNGRADING SCHEDULE |

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

Additions to/from the DISTRIBUTION LIST are invited and should be forwarded to the following address: Commander, US Army Armament, Munitions and Chemical Command, ATTN: DRSMC-SAS (R), Rock Island, IL 61299 AUTOVON 793-5041/6370

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Manufacturing operations
Manufacturing productivity
Industrial operations
Buffer size/capacity
Markov processes
Operations research
Numerical analysis

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This memorandum is concerned with the problem of selecting a capacity value for the buffer between two asynchronous production operations. The methodology study is motivated by shortcomings in present methods. Objectives for this study are that methods should be at least as efficient as stochastic simulation and that they should provide a means of examining both the utilization of the buffer and the productivity of the manufacturing process.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

DRSMC/SA/MR-1

# METHODS FOR CHOOSING BUFFER SIZE IN TANDEM PRODUCTION OPERATIONS

GEORGE J. SCHLENKER

August 1983

ABSTRACT

This memorandum is concerned with the problem of selecting a capacity value for the buffer between two asynchronous production operations. The methodology study is motivated by shortcomings in present methods. Objectives for this study are that methods should be at least as efficient as stochastic simulation and that they should provide a means of examining both the utilization of the buffer and the productivity of the manufacturing process.

Two methods were developed and implemented in computer programs. Both methods use the theory of Markov processes. The first method (GS.BUF) calculates steady-state probabilities for all states of a simple production system. For certain types of processes the results of this model are exact. The results offer a good approximation for many processes. This model calculates the system productivity explicitly, providing an opportunity for economic tradeoffs between buffer capacity and other parameters. The second model (BUF.CAP) focuses on the dynamics of the filling and emptying of the buffer, under the assumption of statistical independence between states of the two operations. This model admits the possibility of several machines working in parallel at each operation.

DRSMC-SAS (R)

5 August 1983

MEMORANDUM FOR RECORD

SUBJECT: Methods for Choosing Buffer Size in Tandem Production Operations

1. Reference:

a. DD 1498, HQ, US ARRCOM, DRSAR-SA, March 1983, title: Manufacturing Productivity Study.

b. Tech Report No. 82014, Menke, W. and Tran, D., ARRADCOM, November 1982, title: Simulation of Ammunition Production Lines.

2. Outline of the MFR

The following is an outline of this memorandum:

a. Background

b. Motivation

c. Definitions

d. Objectives

e. Methodology Overview

f. General Conclusions

g. Derivation of Equations for GS.BUF

h. Analytical Results of GS.BUF

i. Conclusions Regarding GS.BUF

j. Derivation of Equations for BUF.CAP

k. Results of BUF.CAP

l. Not Used

m. Conclusions Regarding BUF.CAP

n. Annexes -- Computer Source Programs GS.BUF and BUF.CAP

3. Background

Over the past year I have been involved in a methodology study concerned with estimation of the capacity of production lines. This study [Ref a] has produced a general computer simulation (TANDEMT) capable of performing stochastic simulation on manufacturing systems having quite general structures. By contrast, the problem addressed by this memorandum is quite restricted in scope. One of the objectives of the manufacturing productivity study is to develop the means for understanding the significance of equipment and/or procedural changes on the productivity of a specific manufacturing line. The methods discussed here concern only the effect of buffer capacity in a simple, asynchronous system with two operations in tandem and an intervening buffer.

4. Motivation

Ref b mentions shortcomings of procedures for sizing the buffer between manufacturing operations. An ad hoc method is proposed there for generating a reasonably sized buffer, suitable for use as an input datum to a stochastic simulation of a developmental production process. The method is not claimed to be logically rigorous. The proposal yields a single number, but lacks a measure of sensitivity of the process output (or buffer performance) to buffer size. A sound approach to this problem is needed which permits the calculation of the advantage of increasing buffer capacity. A proper method should permit efficient tradeoffs to be made between buffer capacity and other process parameters. The methods presented in this memorandum possess these attributes.

5. Definitions

The simplest tandem system consists of two operations running asynchronously with an intervening buffer. In this MFR the term "simple production system" refers specifically to this kind of system. More complex systems may be viewed as arrangements of these simple systems. Additionally, the term productivity is used here somewhat restrictively. Productivity is a measure of the efficiency of a production system to produce, within imposed machine limits. As used here, productivity is defined as the ratio of the average quantity produced, in steady state, to the production of a perfect system having the same machine rates. The machines comprising the production system are viewed in a general way. They can consist of automatic hardware or of humans with simple tools or anything in between. A property of a machine is that it fails or requires adjustment by a repairman at random operating intervals. The mean time between "failures" of a machine type is a property of this machine, abbreviated MTBF. Similarly, the mean time to repair is denoted MTTR. To simplify the analysis without undue loss of generality, it is assumed that sufficient repairmen are available so that essentially no time*

* Alternatively, at least m repairmen are present, where, $P\{m \text{ or less repairmen busy}\} \gtrsim 0.9$.

is spent by machines waiting in a repair queue. This is what is meant by a well maintained system.

## 6. Objectives

The principal objective of this study is to develop a method for sizing the buffer in a simple production system. The method should be at least as numerically efficient as stochastic simulation. Further, the method should provide a means of examining both the utilization of the buffer and the productivity of the manufacturing process.

## 7. Methodology Overview

Before getting into details of the analysis, we consider the general approach to the methods of this memorandum. Both of the methods are based on the theory of Markov processes. The computer programs which implement these methods are found in the Annexes. In the first method, with program name GS.BUF, the three components of a simple production system are viewed as operating together to generate various system states. The system states are defined in terms of the admissible states of the 1st machine operation, of the buffer, and of the 2nd machine operation. As shown in Figure 1, the 1st operation is considered a single machine whose states are (a) under repair, (b) waiting for a space in the buffer to place a completed part, and (c) operating on a part. These substates are numerically coded as 0, 1, and 2 respectively. The state of the buffer is just the number of parts occupying it. The admissible buffer substates are integers from 0 to the buffer capacity. The 2nd operation is also considered a single machine whose states are coded 0, 1, and 2 for (a) under repair, (b) waiting for a part to remove from the buffer, and (c) operating on a part.

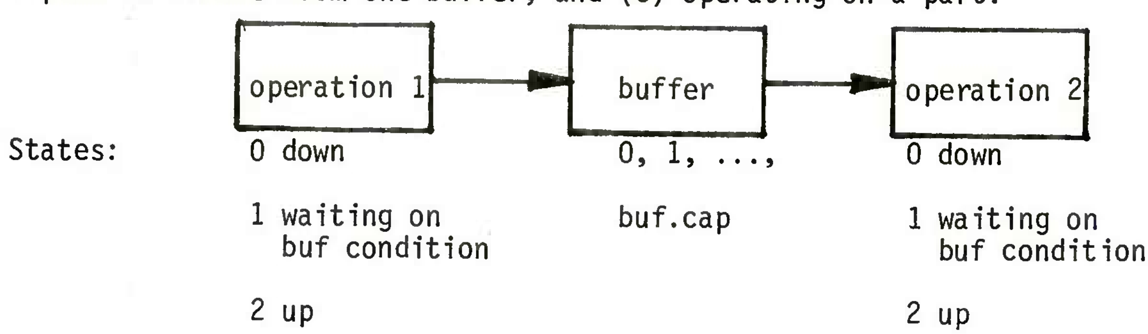



Figure 1. States of a Simple Production System

8. A state of the system is denoted by three integers each of which characterizes one of the consecutive components of this system. Thus, state 0, 0, 1 indicates that operation 1 is under repair, that the buffer is empty, and that operation 2 is waiting for a part to operate upon. For convenience the states are labeled with a single index i. The probability that the system occupies the i th state at time t is denoted $p_i(t)$. The linear equations which functionally relate $d(p_i(t))/dt$ to the various probabilities of state occupancy, for all states, are called the Kolmogorov equations. These differential - difference equations can be simply written under the assumptions of exponentially distributed -- time to fail, time to repair, and machine operation (or service) times. (As shown later these assumptions are not too restrictive.)

9. In stochastic steady state all the derivatives are set to zero. The resultant set of linear algebraic equations is solved for the state probabilities in steady state. Certain states are identified which collectively represent interesting conditions. For example, those states in which operation 1 is waiting or operation 2 is waiting for the buffer, etc. A condition of particular interest is: operation 2 is either waiting or down for repair. The probability that this condition obtains is the fraction of time that the simple system is nonproductive. The 1's complement of this probability is, thus, system productivity. The probabilities of these state conditions can be used in tradeoff analyses to size the buffer. Economic factors can be invoked to determine the value of increased productivity versus the cost of additional buffer capacity. This is the basis of the first method for sizing the buffer.

10. <u>A Second Approach</u>

Another approach to buffer sizing is considered. Altho lacking in the comprehensiveness of the first approach, it does provide an approximation of the probability that a specific number of buffer spaces would be required <u>if</u> the first and second machine operations were not constrained to wait for <u>a</u> buffer condition. Unlike the first approach, this method explicitly accounts for the possible existence of several identical machines working in parallel at each machine operation. The basis of the 2nd approach is to consider the states of the first operation to be statistically independent of the states of the second. Independence is a reasonable assumption if the buffer capacity is large and if there are no common causes of machine failure. Then, the Kolmogorov equations for the first operation and for the second are independent and have the same simple form. Generally, an operation has N machines operating in parallel and possesses N+1 states. In this case the state value of an operation is just the number of machines operating. In a two-operation system with $N_1$ machines in operation 1 and $N_2$ machines in operation 2, there are $(N_1+1)(N_2+1)$

system states. The method, implemented in BUF.CAP, starts with the system conditionally in each of its states in turn, and directly solves the differential - difference (Kolmogorov) equations to obtain the time-dependent average (expected) production from each of the operations, conditioned by the initial state. Because of the independence between operations, the actual numerical procedure considers each operation by itself and obtains the N+1 conditional trajectories -- time-dependent state probabilities and associated expected production -- storing the results. The difference in expected production from operations 1 and 2 at time t represents the expected value of the parts which would be added to the buffer, if product 1 > product 2, or would be removed from the buffer, if product 1 < product 2. Recall that this expected difference is conditional upon the initial state condition (IC). But because the IC's are random the expected production differences are actually random variables.

11. These differences are calculated for all system states at a time large enough to allow the state probabilities to approach their steady-state values (about 4*MTTR). Then the expected differences are rank ordered from smallest to largest. The probability that the system initial state would be occupied in steady state is calculated (via a product of binomial probabilities). These probabilities are associated with each of the ordered production - difference values. These probability densities are accumulated to yield the cumulative distribution function (c.d.f.) for the production difference expected under these conditions. The program BUF.CAP displays this c.d.f. In a balanced production system the expected value of the production difference from this distribution is zero. The mean and variance of the random variable from this distribution are calculated in BUF.CAP. The variance depends upon the number of machines at each operation, the machine rates, and the MTBF's and MTTR's. To reduce the risk of causing machines to wait for the buffer, the required buffer capacity is set to the range in expected production difference plus 4 standard deviations. This somewhat arbitrary assignment provides a reasonably small risk that the calculated buffer capacity would be inadequate. A measure of risk is provided explicitly in this approach, but no measure of productivity is given here. However, using stochastic simulation I have noted that the marginal change in productivity at the calculated buffer capacity is about 0.04 percent per percent change in capacity.

12. General Conclusions

The most general conclusions concerning the above methods are presented here. Details are presented in later parts of this memorandum. In production systems having operation times which are exponentially distributed and in which repair times are exponentially distributed, a steady-state Markov model of a simple production system provides a satisfactory approach to sizing the buffer. A model of this type is implemented in the computer program GS.BUF. Even when the machine operating rates are constants (or nearly so), GS.BUF provides a good approach, providing that several machines are working in parallel at each operation. When constant-rate single-machine operations obtain, stochastic simulation is the preferred approach to buffer sizing. The use of GS.BUF permits one to invoke economic factors in sizing the buffer. Machine operating rate (or operating time interval), repair time, and buffer capacity are all important variables which affect productivity. If one wishes to examine the (filling and emptying) dynamics of the buffer in detail, the method implemented in BUF.CAP is suggested. This method is also useful in obtaining a point estimate of buffer capacity when a more elaborate economic tradeoff is not appropriate.

13. Derivation of Equations for GS.BUF

The general outline of the theory for GS.BUF, provided in paragraph 7 ff., will be detailed here. To make the exposition simple, consider a simple production system with a buffer having only 3 spaces. The theory of Markov processes can be applied directly to the states defined in Figure 1 providing the operating times, time intervals between failure, and repair times are all exponential random variables. Other substates could be added to accommodate other distributions of these random variables. For the present, consider only the states given in Table 1.

14. Transitions occur between these states. The state numbers in Table 1 appearing at the left in sequence are used as indices to designate the probability of state occupancy. Thus, $p_1(t)$ refers to the probability that the system is in state 1 at time t. The states to which a particular state may transition are listed in the column labeled "transition-to states." Similarly, the states from which transfers may occur are listed in the column labeled "transition-from states." Ordinarily, Markov processes may be represented diagrammatically by a graphical network with nodes as states and arcs as transitions -- the state-transition diagram. The rate parameters for the transitions are affixed to the corresponding arcs. Because of the visual complexity of the graph for this process, only a partial state-transition diagram is shown in Figure 2. In this case the first seven states are isolated, and all states connecting each of these states are shown separately. The rate parameters shown in Figure 2 -- r, $\lambda$, $\mu$ -- are indexed with a 1 or 2 to indicate the operation to which it belongs. For example, for state 1, transitions to state 2 occur at the "birth" rate $\lambda_1$, which characterizes operation 1.

TABLE 1

DEFINITION OF STATES OF A SIMPLE PRODUCTION SYSTEM

Example with buffer capacity of 3 spaces.

| State Number | State Definition* | | | Transition-to States | Transition-from States |
|---|---|---|---|---|---|
| | Opn 1 | Buf | Opn 2 | | |
| 1 | 0 | 0 | 1 | 2 | 2, 4 |
| 2 | 2 | 0 | 1 | 1, 6 | 1, 6 |
| 3 | 0 | 0 | 0 | 4, 5 | 4, 5 |
| 4 | 0 | 0 | 2 | 1, 3, 6 | 3, 6, 8 |
| 5 | 2 | 0 | 0 | 3, 6, 9 | 3, 6 |
| 6 | 2 | 0 | 2 | 2, 4, 5, 10 | 4, 5, 10 |
| 7 | 0 | 1 | 0 | 8, 9 | 8, 9 |
| 8 | 0 | 1 | 2 | 4, 7, 10 | 7, 10, 12 |
| 9 | 2 | 1 | 0 | 7, 10, 13 | 5, 7, 10 |
| 10 | 2 | 1 | 2 | 6, 8, 9, 14 | 6, 8, 9, 14 |
| 11 | 0 | 2 | 0 | 12, 13 | 12, 13 |
| 12 | 0 | 2 | 2 | 8, 11, 14 | 11, 14, 16 |
| 13 | 2 | 0 | 0 | 11, 14, 17 | 9, 11, 14 |
| 14 | 2 | 2 | 2 | 10, 12, 13, 18 | 10, 12, 13, 18 |
| 15 | 0 | 3 | 0 | 16, 17 | 16, 17 |
| 16 | 0 | 3 | 2 | 12, 15, 18 | 15, 18 |
| 17 | 2 | 3 | 0 | 15, 18, 19 | 13, 15, 18 |
| 18 | 2 | 3 | 2 | 14, 16, 17, 20 | 14, 16, 17, 20 |
| 19 | 1 | 3 | 0 | 20 | 17, 20 |
| 20 | 1 | 3 | 2 | 18, 19 | 18, 19 |

* For operations 1 and 2, the integers in the state definition have the following meanings:

0 means "down" or under repair, with a part being held.

1 means waiting for a buffer condition--to place a part, for operation 1, and to remove a part, for operation 2.

2 means "up" or working on a part.

The integer characterizing the buffer state is the number of parts in the buffer.

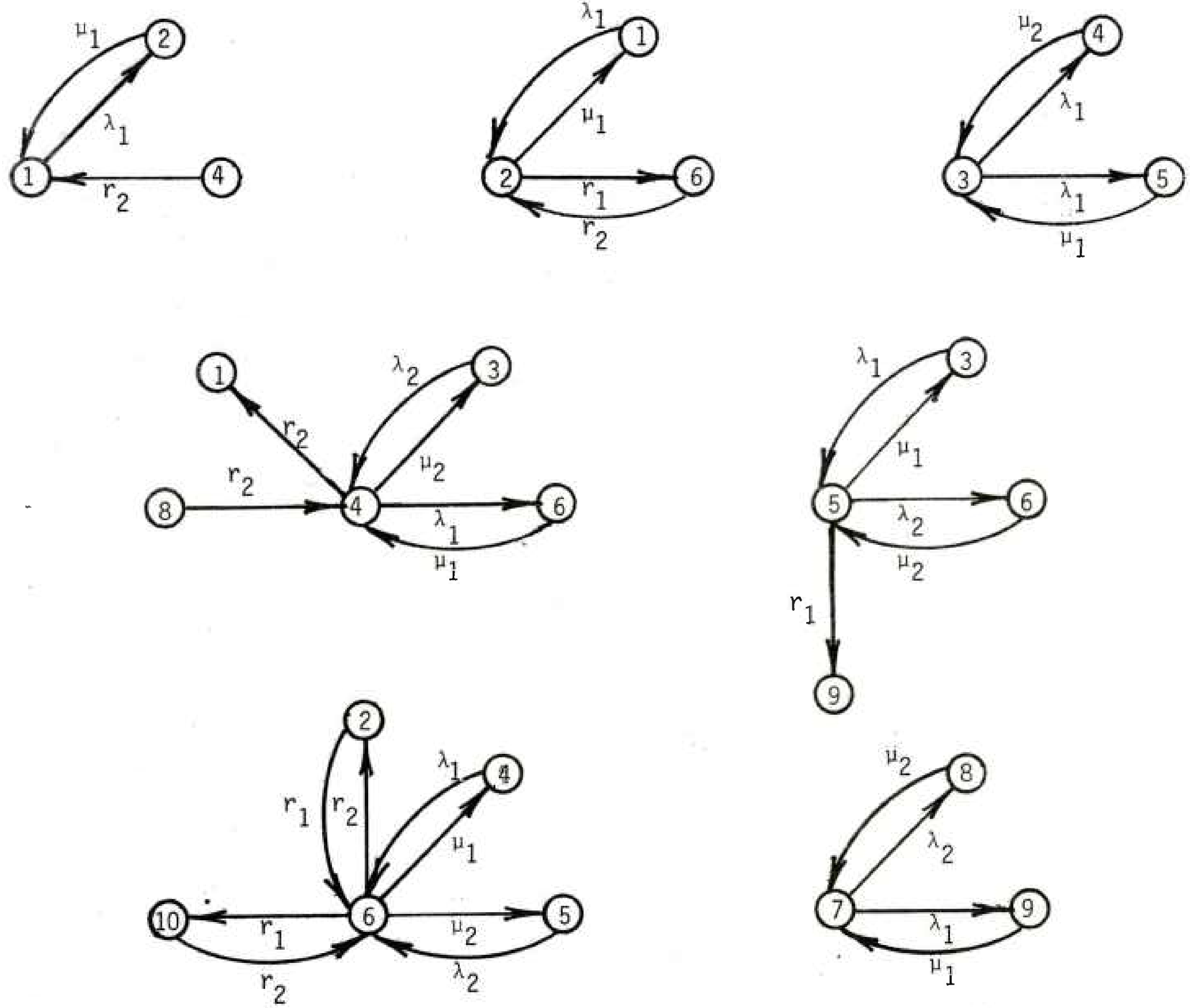



Figure 2. Partial State Transition Diagram for a Simple Production Process

Transitions to state 1 occur from states 2 and 4. The transition from state 2 occurs at the "death" rate $\mu_1$, associated with operation 1. The transition from state 4 occurs at the operating rate $r_2$ of operation 2. The birth rate for an operation is, of course, the rate at which that operation is restored to operational condition given that it is "down". Thus,

$$\lambda_i = 1/MTTR_i \qquad , \qquad i = 1,2 \ , \tag{1}$$

for operations 1 and 2. Similarly, the death rate for an operation is the rate at which a failure occurs, given that the operation is "up". Thus,

$$\mu_i = 1/MTBF_i \qquad , \qquad i = 1,2 \ . \tag{2}$$

The machine rates (reciprocals of mean service time) are denoted by $r_1$ and $r_2$, for operations 1 and 2 respectively.

15. With the aid of the state transition diagram, writing the Kolmogorov equations is a quite mechanical task. For example, as is customary in deriving transition equations, consider a small time increment h. Then, the probability that state 1 is occupied at time t + h is given as

P{state 1 at t+h} = P{no transition in h occurs from state 1, given occupation at t}

*P{state 1 at t} + P{transition from state 2, given occupation of state 2 at t}

*P{state 2 at t} + P{transition from state 4, given occupation of state 4 at t}

*P{state 4 at t}.

Using the abbreviated notation, this expression becomes

$$p_1(t+h) = (1-\lambda_1 h)p_1(t) + \mu_1 h p_2(t) + r_2 h p_4(t). \tag{3}$$

Then,

$$[p_1(t+h)-p_1(t)]/h = -\lambda_1 p_1(t) + \mu_1 p_2(t) + r_2 p_4(t). \tag{4}$$

Taking the limit as h approaches zero and omitting the functional dependence upon t,

$$\dot{p}_1 = -\lambda_1 \, p_1 + \mu_1 \, p_2 + r_2 \, p_4 \ . \tag{5}$$

So much for conventional derivations! This expression can be obtained directly from the state transition diagram by writing as terms the probabilities of all states which transition to a particular state--the 1st here--on the right with their transition rates as positive coefficients. The rates of all transitions <u>from</u> the particular state are collected and the negative of this sum is the coefficient of the particular state. The Kolmogorov state equations for the simple system with 3-space buffer are written compactly in Table 2. The first column in this table is the index (i) of the left hand side $\dot{p}_i$.

16. For a simple system with a buffer capacity of 3-spaces, there are 20 states. The display of the corresponding 20 equations is awkward, but still manageable. With increasing buffer size, writing the state equations explicitly is infeasible. Fortunately, this display is not necessary. For notational simplification, let $\underline{p}$ be a column vector of the state probabilities with ns (number of states) elements. Let B be a square matrix of coefficients with ns rows. Then, the Kolmogorov equations for a simple production system can be written, generally, as

$$\dot{\underline{p}} = B\underline{p} \quad . \tag{6}$$

Because the sum of the elements of $\underline{p}$ is unity, B is not of full rank. Thus, equation (6) is not solved directly. In obtaining a solution, however, it is useful to generate the elements ($b_{ij}$) of B. Because B is a stochastic matrix, the sum of each of its column vectors is zero (reflecting the fact that each arc in the state transition diagram both leaves and enters a node). This fact is exploited to check the validity of the equations actually solved in GS.BUF.

17. In Table 1, note that the first 2 states are waiting (i.e., 1) states for operation 2, whereas the last 2 states are waiting states for operation 1. In each of the states between the 2nd and 2nd to last, a regular pattern is observed. For a given buffer state two 0 states are assigned operation 1, with operation 2 taking the values 0 and 2. Next, two 2 states are assigned for operation 1, with operation 2 again taking the values 0 and 2. This pattern of four system states is followed for each value of the buffer state. Thus, with a buffer capacity m, the number of states is

$$ns = 4(m+1)+4$$

or

$$ns = 4(m+2) \quad . \tag{7}$$

Because of the regularities in the state definition noted above, there are regularities in the equations, which permit the elements of the B matrix to be written recursively. Starting with state 11 (11th row of B),

$$b_{ij} = b_{i-4,\, j-4} \;, \qquad 11 \le i \le ns-5 \;, \quad i-4 \le j \le ns \quad . \tag{8}$$

This fact greatly simplifies the process of generating the elements of the B matrix.

TABLE 2

STATE EQUATIONS FOR A SIMPLE SYSTEM
WITH 3-SPACE BUFFER

(See Table 1 for definition of states).

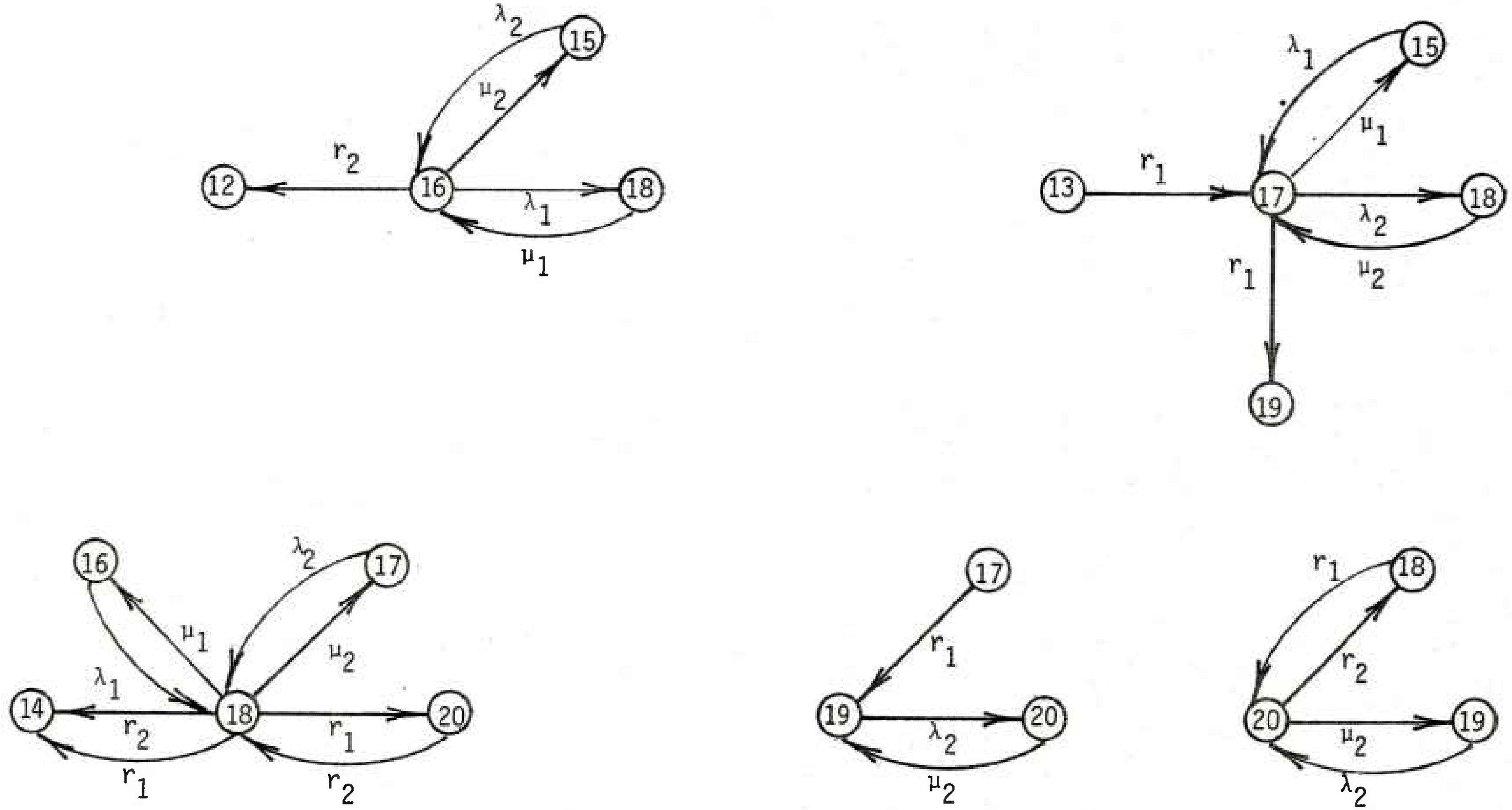
| Derivative of P{State No.} | Terms in Right-Hand-Side Sum |
|---|---|
| 1 | $-\lambda_1 p_1$ , $\mu_1 p_2$ , $r_2 p_4$ |
| 2 | $\lambda_1 p_1$ , $-(\mu_1+r_1)p_2$ , $r_2 p_6$ |
| 3 | $-(\lambda_1+\lambda_2)p_3$ , $\mu_2 p_4$ , $\mu_1 p_5$ |
| 4 | $\lambda_2 p_3$ , $-(\lambda_1+\mu_2+r_2)p_4$ , $\mu_1 p_6$ , $r_2 p_8$ |
| 5 | $\lambda_1 p_3$ , $-(\lambda_2+\mu_1+r_1)p_5$ , $\mu_2 p_6$ |
| 6 | $r_1 p_2$ , $\lambda_1 p_4$ , $\lambda_2 p_5$ , $-(\mu_1+\mu_2+r_1+r_2)p_6$, $r_2 p_{10}$ |
| 7 | $-(\lambda_1+\lambda_2)p_7$ , $\mu_2 p_8$ , $\mu_1 p_9$ |
| 8 | $\lambda_2 p_7$ , $-(\lambda_1+\mu_2+r_2)p_8$ , $\mu_1 p_{10}$ , $r_2 p_{12}$ |
| 9 | $r_1 p_5$ , $\lambda_1 p_7$ , $-(\lambda_2+\mu_1+r_1)p_9$ , $\mu_2 p_{10}$ |
| 10 | $r_1 p_6$ , $\lambda_1 p_8$ , $\lambda_2 p_9$ , $-(\mu_1+\mu_2+r_1+r_2)p_{10}$ , $r_2 p_{14}$ |
| 11 | $-(\lambda_1+\lambda_2)p_{11}$ , $\mu_2 p_{12}$ , $\mu_1 p_{13}$ |
| 12 | $\lambda_2 p_{11}$ , $-(\lambda_1+\mu_2+r_2)p_{12}$ , $\mu_1 p_{14}$ , $r_2 p_{16}$ |
| 13 | $r_1 p_9$ , $\lambda_1 p_{11}$ , $-(\lambda_2+\mu_1+r_1)p_{13}$ , $\mu_2 p_{14}$ |
| 14 | $r_1 p_{10}$ , $\lambda_1 p_{12}$ , $\lambda_2 p_{13}$ , $-(\mu_1+\mu_2+r_1+r_2)p_{14}$ , $r_2 p_{18}$ |
| 15 | $-(\lambda_1+\lambda_2)p_{15}$ , $\mu_2 p_{16}$ , $\mu_1 p_{17}$ |
| 16 | $\lambda_2 p_{15}$ , $-(\lambda_1+\mu_2+r_2)p_{16}$ , $\mu_1 p_{18}$ |

TABLE 2 (Cont)

STATE EQUATIONS FOR A SIMPLE SYSTEM
WITH 3-SPACE BUFFER

(See Table 1 for definition of states).

| Derivative of P{State No.} | Terms in Right-Hand-Side Sum |
|---|---|
| 17 | $r_1p_{13}$ , $\lambda_1p_{15}$ , $-(\lambda_2+\mu_1+r_1)p_{17}$ , $\mu_2p_{18}$ |
| 18 | $r_1p_{14}$ , $\lambda_1p_{16}$ , $\lambda_2p_{17}$ , $-(\mu_1+\mu_2+r_1+r_2)p_{18}$ , $r_2p_{20}$ |
| 19 | $r_1p_{17}$, $-\lambda_2p_{19}$ , $\mu_2p_{20}$ |
| 20 | $r_1p_{18}$ , $\lambda_2p_{19}$ , $-(\mu_2+r_2)p_{20}$ |




NOTE: For a general system with ns states, the above state numbers correspond as follow: ns = 20, ns -1 = 19, etc.

Figure 3. State Transition Diagram of the Last Five States for a Simple Production Process

18. The last 5 system states, which contain waiting states for operation 1, must be written explicitly. For the general case, these are:

$$\dot{p}_{ns-4} = \lambda_2 p_{ns-5} - (\lambda_1+\mu_2+r_2)p_{ns-4} + \mu_1 p_{ns-2} \tag{9}$$

$$\dot{p}_{ns-3} = r_1 p_{ns-7} + \lambda_1 p_{ns-5} - (\lambda_2+\mu_1+r_1)p_{ns-3} + \mu_2 p_{ns-2} \tag{10}$$

$$\dot{p}_{ns-2} = r_1 p_{ns-6} + \lambda_1 p_{ns-4} + \lambda_2 p_{ns-3} - (\mu_1+\mu_2+r_1+r_2)p_{ns-2} + r_2 p_{ns} \tag{11}$$

$$\dot{p}_{ns-1} = r_1 p_{ns-3} - \lambda_2 p_{ns-1} + \mu_2 p_{ns} \tag{12}$$

$$\dot{p}_{ns} = r_1 p_{ns-2} + \lambda_2 p_{ms-1} - (\mu_2+r_2)p_{ns} \quad . \tag{13}$$

The state transition diagram for the last five states is found in Figure 3.

19. To evaluate the system under stochastic steady state, the $\underline{\dot{p}}$ vector is set to zero.

Then,

$$B\underline{p} = \underline{0} \quad . \tag{14}$$

As noted above, the resultant set of equations contains one superfluous equation, since the state probabilities must sum to 1. To remedy this situation, (14) is converted to the linear matrix-vector equation

$$A\,\underline{x} = \underline{c}$$

with

$$A\ (nxn), \quad \underline{x}\ (nx1), \text{ and } \quad \underline{c}\ (nx1) \quad ,$$

where

$$n = ns-1 \quad .$$

This is done by assigning

$$p_1 = 1-\Sigma_{k=2}^{ns} \; p_k \quad . \tag{16}$$

The elements of the A matrix and $\underline{c}$ vector in (15) are obtained via the transformations

$$b'_{ij} = b_{ij} - b_{i1} \quad , \quad 2 \le i \le ns$$

and,

$$c_{i-1} = -b_{i1} \quad , \quad 2 \le i \le ns$$

$$a_{ij} = b'_{i+1,\, j+1} \; , \; 1 \le i \, , \, j \le n \; . \qquad (17)$$

Note that the first scalar equation in equation (14) is deleted in forming (15). The A matrix is of full rank, so $\underline{x}$ may be obtained by

$$\underline{x} = A^{-1}\underline{c} \quad . \qquad (18)$$

Then,

$$p_{i+1} = x_i \quad , \quad 1 \le i \le n \quad . \qquad (19)$$

Finally, $p_1$ is obtained from (16).

20. The probabilities of buffer occupancy are obtained from the state probability vector $\underline{p}$ (or, alternatively from $\underline{x}$) by

$$P\{\text{buffer is empty}\} = \Sigma_{k=1}^{6}\, p_k \quad . \qquad (20a)$$

$$P\{j \text{ parts in buffer}\} =$$

$$\Sigma_{k=1}^{4}\, p_{k+4j+2} \quad , \quad 1 \le j \le m-1 \quad . \qquad (20b)$$

$$P\{\text{buffer is full}\} = (\text{by definition})$$

$$P\{m \text{ parts in buffer}\} = \Sigma_{k=1}^{6}\, p_{k+4m+2} \quad . \qquad (20c)$$

The probability that operation 2 must wait for a buffer condition is just the sum of the first two state probabilities:

$$P\{\text{operation 2 must wait}\} = p_1 + p_2 \quad . \qquad (21)$$

The probability that operation 1 must wait for a buffer condition is the sum of the last two state probabilities:

$$P\{\text{operation 1 must wait}\} = p_n + p_{n+1} \quad . \tag{22}$$

The system productivity depends upon the probability that operation 2 is down. This last probability is given by

$$P\{\text{operation 2 is down}\} = \sum_{k=1}^{(ns-2)/2} p_{2k+1} \quad . \tag{23}$$

Finally, the system productivity is

$$1 - P\{\text{operation 2 is down}\} - P\{\text{operation 2 must wait}\} \ . \tag{24}$$

If one wants the steady-state production rate of the simple system, this value is obtained from the productivity by multiplying times $r_2$.

## 21. Analytic Results of GS.BUF

Calculations were made using GS.BUF to compare results with stochastic simulation and to do a sensitivity analysis of certain parameters. This computational experience provided timing estimates on our PR1ME 550 minicomputer. All calculations were made using double-precision arithmetic. The stochastic simulation used for comparison was a very simple implementation of TANDEMT. In all cases studied the time between failures and the time to repair were exponential random variables. Simulation runs were 40 24-hour days, starting with an empty system. Runs were made for instances in which the machine operating (or service) time is exponential and in which it is constant. The runs with constant service time are used to test the applicability of the Markov model in GS.BUF to a quite different model. Parenthetically, I note that other Markov models can be created to approximate the constant-rate case. By defining many substates to describe a machine operation, it is possible to describe a random service time whose coefficient of variation is quite small (if not zero). In fact, I constructed a specific case of such a model using 3 substates. The productivity calculated with that model is in better agreement with simulated results than is the case for GS.BUF. However, due to the large number of system states produced by this method, computational efficiency is poor, for a typical buffer size. Consequently, that approach was abandoned. It is recommended for single-machine operations having constant rates, that stochastic simulation be used to estimate system productivity. This approach also has the advantage of modeling other-than well maintained systems, which were considered here.

22. Results of GS.BUF in which the operations have a common set of parameters are shown on the first page of Table 3. Buffer capacity is treated as a parameter in the comparison between calculated and simulated results. This comparison indicates complete agreement within expected statistical variation. For simulations of this length the estimated standard error of the productivity estimate is about 0.11 to 0.14 for these examples. The typically somewhat larger estimated probability that the buffer is empty is a reflection that the simulation was not started in the steady state. However, due to the simulation length, this effect is small. On the second page of Table 3, a comparison is made in which the parameters of operation 1 are not the same as those of operation 2. Again, agreement is excellent. Whenever one is designing a buffer, a balanced pair of operations should be considered. When balanced, the operation thruput is a constant equal to the product of the machine rate, the machine availability, and the number of machines. Note that these operations are balanced.

23. Calculated and simulated productivities for an expanded set of buffer capacities are shown in Table 4. This sort of analysis can be used in making economic tradeoffs when choosing a buffer capacity. The results of TANDEMT are shown in Table 5. It is noted that the system productivity is always greater when the machine service (operating) times are constants than when they are exponential random variables. However, note that the difference (and relative difference) in productivities in these two instances diminishes as the buffer capacity increases. This fact suggests that regardless of the distribution of service times, GS.BUF may be a practical procedure to use for productivity estimation (and tradeoff) <u>when</u> the buffer capacity is large. Figure 4 shows the probability density functions of buffer state occupancy for cases in which the service times are constant and exponential random variables. The U-shaped densities are typical. Note that much larger probabilities of being at the buffer extremes is exhibited by a constant-rate system.

24. The sensitivity analysis using GS.BUF considers the effect of the following three parameters on system productivity: buffer capacity, MTTR, and machine rate. While not large, the ranges of these parameters are representative of many ammunition production operations. Several inferences can be drawn from the results shown in Table 6. Only one is mentioned at this point. Suppose the machine rate is given and the buffer size chosen to yield a particular productivity or, alternatively, chosen so that the marginal cost of additional buffer space just equals the value of additional production from a system with the incrementally larger buffer. If, later, a greater machine rate is available via, possibly, machine substitution, one can expect a <u>productivity decrease</u> if the buffer is not enlarged. Remember that productivity is a measure of production efficiency. Thus, doubling the machine rate -- with buffer fixed -- will increase the production rate, but will not double it.

TABLE 3

COMPARISON OF THE PROBABILITIES OF BUFFER OCCUPANCY: CALCULATED VERSUS SIMULATED

Parameters:
Two single-machine operations in tandem.
Operation (service) times are exponential.
Repair times are exponential.
MTBF = 100 minutes
MTTR = 25 minutes
Average operating rate = 1 part/min.

| Buffer Capacity | Buffer Status | State Probabilities Calculated | Simulated |
|---|---|---|---|
| 3 | 0 | 0.379 | 0.385 |
| | 1 | 0.121 | 0.125 |
| | 2 | 0.121 | 0.126 |
| | 3 | 0.379 | 0.364 |
| | | System Productivity* | |
| | | 0.595 | 0.599 |
| 5 | 0 | 0.320 | 0.312 |
| | 1 | 0.091 | 0.094 |
| | 2 | 0.089 | 0.095 |
| | 3 | 0.089 | 0.094 |
| | 4 | 0.091 | 0.094 |
| | 5 | 0.320 | 0.311 |
| | | System Productivity* | |
| | | 0.620 | 0.630 |
| 10 | 0 | 0.250 | 0.247 |
| | 1 | 0.059 | 0.061 |
| | 2 | 0.056 | 0.059 |
| | 3 | 0.055 | 0.056 |
| | 4 | 0.054 | 0.054 |
| | 5 | 0.053 | 0.056 |
| | 6 | 0.054 | 0.054 |
| | 7 | 0.055 | 0.056 |
| | 8 | 0.056 | 0.057 |
| | 9 | 0.059 | 0.060 |
| | 10 | 0.250 | 0.240 |
| | | System Productivity* | |
| | | 0.650 | 0.640 |

TABLE 3 (Cont)

COMPARISON OF THE PROBABILITIES OF BUFFER OCCUPANCY: CALCULATED VERSUS SIMULATED

Parameters:
Two single-machine operations in tandem.
Operation (service) times are exponential.
Repair times are exponential.
MTBF = 200 min (1st opn), = 100 min (2nd opn).
MTTR = 25 min (1st opn), = 12.5 min (2nd opn).
Average operating rate = 1/min.

| Buffer Capacity | Buffer States | State Probabilities Calculated | State Probabilities Simulated |
|---|---|---|---|
| 3 | 0 | 0.357 | 0.368 |
| | 1 | 0.140 | 0.143 |
| | 2 | 0.141 | 0.144 |
| | 3 | 0.362 | 0.345 |
| | | System Productivity* | |
| | | 0.695 | 0.696 |
| 5 | 0 | 0.287 | 0.291 |
| | 1 | 0.103 | 0.105 |
| | 2 | 0.104 | 0.106 |
| | 3 | 0.105 | 0.106 |
| | 4 | 0.108 | 0.109 |
| | 5 | 0.294 | 0.283 |
| | | System Productivity* | |
| | | 0.727 | 0.731 |

* System productivity is the ratio of the average production achieved to the maximum steady-state production from a system of perfect machines operating at the same rates.

TABLE 4

COMPARISON OF CALCULATED WITH
SIMULATED PRODUCTIVITY ESTIMATES FOR
A SIMPLE PRODUCTION SYSTEM*

| Buffer Capacity | Average Productivity Calculated | Simulated** |
|---|---|---|
| 3 | 0.595 | 0.599 |
| 5 | 0.620 | 0.630 |
| 10 | 0.650 | 0.640 |
| 20 | 0.680 | 0.684 |
| 40 | 0.710 | 0.713 |

* Parameters:
A single machine at each of two operations.
Exponential operation times.
Exponential repair times.
Average operating rate 1 part/minute.
Common MTBF = 100 minutes.
Common MTTR = 25 minutes.

** The standard error is about 0.011, based on a 40 day simulation.

TABLE 5

COMPARISON OF PRODUCTIVITY ESTIMATES FOR
SIMPLE PRODUCTION SYSTEMS* HAVING
CONSTANT VERSUS EXPONENTIAL OPERATING TIMES

Tabulated productivity is the ratio of expected production to the maximum production from a perfect system operating at the same machine rate.

| Buffer Capacity | Simulated Operating Times | |
| --- | --- | --- |
| | Constant | Exponential |
| 3 | 0.674 | 0.599 |
| 5 | 0.679 | 0.630 |
| 10 | 0.690 | 0.640 |
| 20 | 0.709 | 0.684 |
| 40 | 0.724 | 0.713 |

* Parameters:
Single machine operations.
Average operating rate 1/min.
Repair times are exponential.
MTBF = 100 minutes
MTTR = 25 minutes

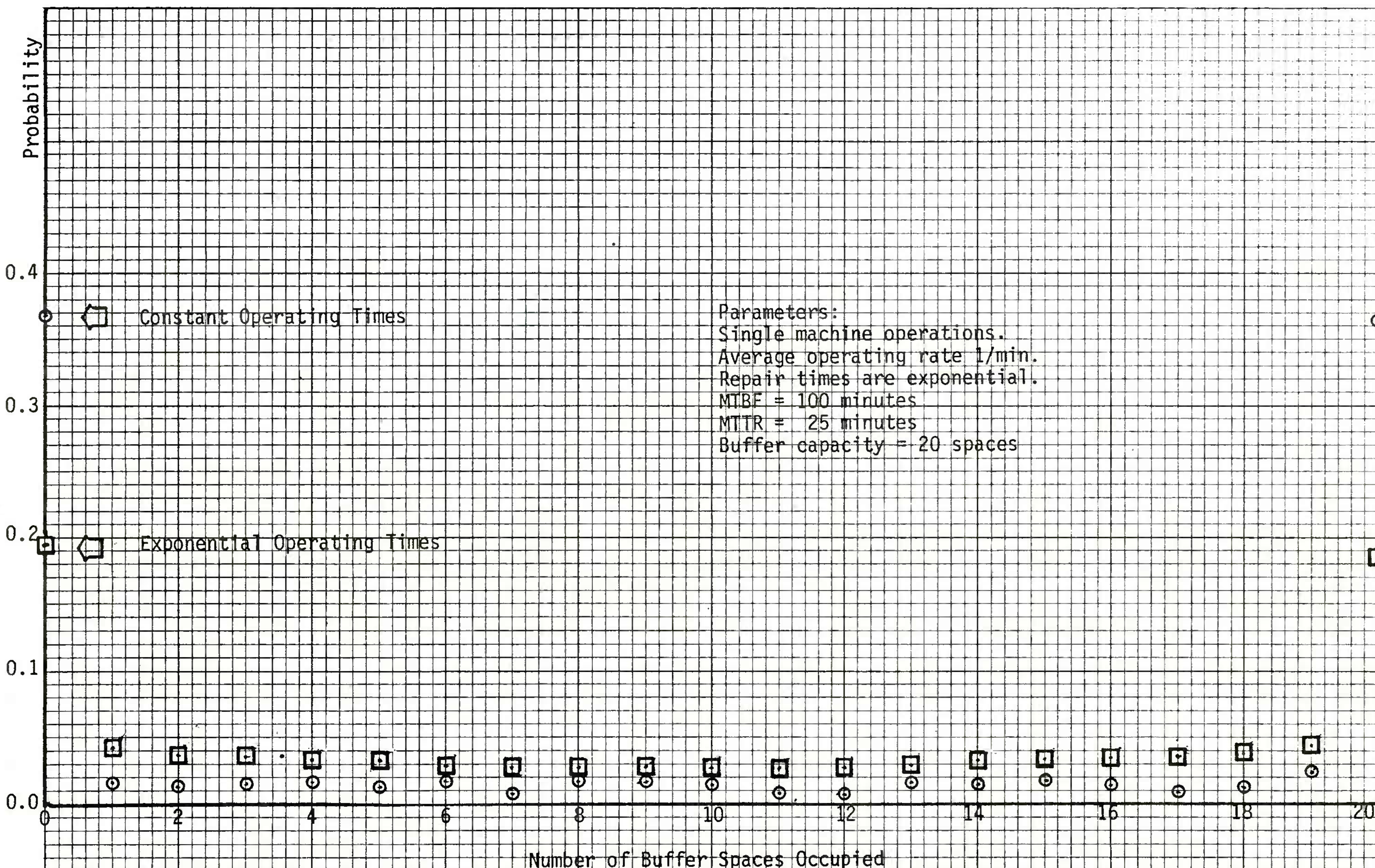



Figure 4 . Comparison of Probability Densities of Buffer Space for Production Systems Having Different Distributions of Operating Time

TABLE 6

CALCULATED PRODUCTIVITY OF A SIMPLE PRODUCTION SYSTEM* AS A FUNCTION OF SEVERAL VARIABLES

The tabulated productivity is the ratio of average production achieved to the maximum steady-state production from a perfect system operating at the same machine rate.

| Buffer Capacity | MTTR (minutes) | Machine Rate (parts/min) 1 | 2 |
|---|---|---|---|
| 10 | 12.5 | 0.770 | 0.761 |
| | 25.0 | 0.650 | 0.642 |
| 20 | 12.5 | 0.805 | 0.792 |
| | 25.0 | 0.680 | 0.666 |
| 40 | 12.5 | 0.835 | 0.819 |
| | 25.0 | 0.710 | 0.690 |

* System parameters:
Two single-machine operations in tandem.
Capacity of intermediate buffer is a parameter.
Common machine rate is a parameter.
Common MTTR is a parameter.
Exponential operation (service) times.
Exponential repair times.
Common MTBF = 100 minutes.

25. The system model of GS.BUF assumes single-machine operations. However, multi-machine operations can be approximated by this model by scaling the machine rate to Nr, where N is the number of machines in the operation, each having rate r. This approximation exploits a theorem of random processes on the pooling of many component processes. Cox* states that when many independent process events are pooled, the pooled process is approximately Poisson, i.e., the time between events is approximately exponential. This is true irrespective of the distributions of the component processes. In application to multi-machine operations, each machine's output gets pooled for that operation. Thus, the times between unit production events are approximately exponential random variables when the number of machines is large. Surprisingly, the number working in parallel at an operation does not need to be more than about 4 to yield a good approximation of the probability density function (p.d.f.) for buffer occupancy. This point is illustrated by the results in Table 7. Two multi-machine cases were simulated: 4 machines per operation working at machine rate 1/4 parts per minute and 5 machines working in parallel at machine rate 1/5 parts per minute. For comparison is shown the p.d.f. of buffer occupancy calculated with GS.BUF having a machine rate of 1 part per minute. One observes that the p.d.f.'s in these instances are nearly the same. Agreement between calculated and simulated productivities is not as good, however. The simulated productivities for the two examples are both 0.87 whereas the calculated productivity is 0.81.

26. Conclusions Regarding GS.BUF

Specific conclusions for the method of GS.BUF are summarized here. (a) When a simple production system satisfies the assumption of exponentially distributed service times, the results of GS.BUF are exact. This point has been verified with simulation. (b) With stochastic steady state, the probability density function (p.d.f.) of buffer occupancy is U-shaped. For operations having a common set of operating characteristics, the above p.d.f. is symmetric with respect to 0.5 · buffer capacity spaces. A prominent positive jump in the p.d.f. is observed at the first and last states. For a given buffer capacity the jump is more pronounced when the operating times are constant than when they are both exponential. This implies that relatively more time is spent at extreme states when the operating times are constant than when they are random. Nevertheless, the productivity of a system with constant operating rate is greater than that of a system with a random rate of the same mean value, other things being the same.

---

* Page 77 ff. Cox, D.R. Renewal Theory, London, distributed by Barnes and Noble, c. 1962.

TABLE 7

COMPARISON OF AN ANALYTIC APPROXIMATION* WITH SIMULATED PROBABILITY DISTRIBUTIONS OF BUFFER OCCUPANCY FOR SYSTEMS HAVING MULTIPLE MACHINES PER OPERATION

* Parameters of the Analytic Model:
Two single-machine operations in tandem.
Operation times are exponential.
Repair times are exponential.
MTBF for each machine = 100 minutes.
MTTR for each machine = 12.5 minutes.
Average operating rate = 1 part/min.
Buffer capacity = 20 spaces.

| Buffer State | Calc. pdf | Simulated+ pdf with 4 mach at rate 1/4 | 5 mach at rate 1/5 |
|---|---|---|---|
| 0 | 0.141 | 0.131 | 0.135 |
| 1 | 0.044 | 0.037 | 0.042 |
| 2 | 0.042 | 0.035 | 0.043 |
| 3 | 0.040 | 0.036 | 0.044 |
| 4 | 0.038 | 0.037 | 0.043 |
| 5 | 0.037 | 0.037 | 0.041 |
| 6 | 0.036 | 0.041 | 0.036 |
| 7 | 0.036 | 0.042 | 0.036 |
| 8 | 0.035 | 0.041 | 0.038 |
| 9 | 0.035 | 0.041 | 0.036 |
| 10 | 0.035 | 0.039 | 0.033 |
| 11 | 0.035 | 0.035 | 0.034 |
| 12 | 0.035 | 0.039 | 0.034 |
| 13 | 0.036 | 0.040 | 0.035 |
| 14 | 0.036 | 0.040 | 0.037 |
| 15 | 0.037 | 0.040 | 0.039 |
| 16 | 0.038 | 0.037 | 0.037 |
| 17 | 0.040 | 0.035 | 0.036 |
| 18 | 0.042 | 0.037 | 0.040 |
| 19 | 0.044 | 0.037 | 0.043 |
| 20 | 0.141 | 0.143 | 0.138 |

\+ Simulated machine operating rates are constants. The MTBF and MTTR parameters of the simulation are as given above.

(c) A sensitivity analysis of system parameters was performed using GS.BUF. All the following are shown to be important: Machine operating rate (or service time), repair time, and buffer capacity. (d) If the buffer capacity remains constant and the machine rate is doubled, a productivity decrease is experienced -- even tho the production rate of the new configuration is greater. This loss of productivity is more pronounced for a system with a large buffer than with a system whose buffer is inadequate. (e) Doubling the production rate and concurrently doubling buffer capacity increases the system productivity. Thus, the buffer size need not be doubled to preserve productivity if the operating rate is doubled. (f) At any reasonable buffer size, halving the MTTR produces a greater improvement in productivity than that achieved by doubling buffer capacity. (g) For systems having constant machine operating rates, GS.BUF can be used to approximate the system, only if many machines are working in parallel at each operation. The probability distribution of buffer state occupancy is approximately correct in this case. If only one fixed-rate machine exists per operation, it is recommended that stochastic simulation be used to estimate the system productivity.

27. <u>Derivation of Equations for BUF.CAP</u>

To motivate subsequent discussion, consider the state dynamics of a one-machine system. This system is regarded as operating independently of other buffers and production operations. In this case there are only two states -- down (0) and up (1). Markov transitions from down to up occur at the birth rate $\lambda$, and transitions from up to down occur at the death rate $\mu$. Thus, the equations for the probabilities of state occupancy are

$$\dot{p}_0(t) = -\lambda p_0+\mu p_1 \tag{25a}$$

$$\dot{p}_1(t) = \lambda p_0-\mu p_1 \quad . \tag{25b}$$

These equations can be solved directly or by using Laplace transforms and inverting.

$$p_0(t) = p_0(0)e^{-(\lambda+\mu)t} + \frac{\mu}{\lambda+\mu}(1-e^{-(\lambda+\mu)t}) \tag{26a}$$

$$p_1(t) = \frac{\lambda}{\lambda+\mu}(1-e^{-(\lambda+\mu)t}) + p_1(0)e^{-(\lambda+\mu)t} \quad . \tag{26b}$$

Note that the steady-state availability,

$p_1(\infty)$, or

$$A = \frac{\lambda}{\lambda+\mu} \quad . \tag{27}$$

Using the definitions of $\lambda$ and $\mu$, equations (1, 2),

$$A = \frac{MTBF}{MTBF+MTTR} \quad . \tag{28}$$

The (mathematically) expected production from this operation over the time interval (0, t) is given by

$$c(t) = r\int_0^t p_1(x)dx \quad , \tag{29}$$

where r is the machine operating rate.

From (26b) and (29),

$$\frac{c(t)}{r} = \frac{\lambda}{\lambda+\mu} t + \frac{\lambda}{(\lambda+\mu)^2} e^{-(\lambda+\mu)t}$$

$$- \frac{\lambda}{(\lambda+\mu)^2} + \frac{p_1(0)}{\lambda+\mu} \left(1-e^{-(\lambda+\mu)t}\right) \quad . \tag{30}$$

28. Consider two single-machine operations in series with a common set of operational parameters. When first observed, let the first operation be in state 1 and the second operation be in state 2. Call this initial condition IC1. Stated mathematically, let

$p_1(0) = 1$ for operation 1

and

$p_1(0) = 0$ for operation 2.

Then, the difference in expected production of these operations as t becomes large can be obtained from (30) as E[production difference, given IC1] = $r/(\lambda+\mu)$. (31)
(Notationally, E is the expected value operator.) If IC1 obtains, one expects to add the above production quantity to the contents of a buffer between the operations, given the assumed freedom from buffer constraints. Clearly, if the initial operation states had been the 1's complements of the above (IC2), a quantity of production would have been removed from the buffer equal to (31). This again assumes that at least that much was present initially. The IC's considered are the extreme conditions for this example. Therefore, at reasonably low risk of buffer inadequacy one might suggest using the range of E[production difference] over extreme IC's for sizing the buffer. In this case

E[production difference, IC1]-E[production difference, IC2]

$$= 2r/(\lambda+\mu) \quad . \tag{32}$$

However, this approach would ignore some obvious facts. First, we are dealing with conditional expected values. Thus, variation from these averages must occur. Secondly, we ignore the fact that the occupancy of the buffer at the commencement of delivery of additional product under IC1 may exceed $r/(\lambda+\mu)$. To reduce the risk associated with these contingencies, one can identify the variation in expected buffer state change. Specifically, one can calculate the standard deviation (SD) of this random variable. Then, some number of SD's can be added to the range in (32) to produce a buffer capacity requirement. I have somewhat arbitrarily chosen 4 SD's as a reasonably cautious number. The probability distribution of expected production difference is calculated by enumerating all possible IC's. In this example there are four -- (0, 0), (0, 1), (1, 0), and (1, 1). The probability that the system of two operations exists in one of these states is simply the product of the probability that operation 1 is in its given state and the probability that operation 2 is in its state. The probability that a one-machine operation is up (1) is A, and down (0) is (1-A). In general, in a N-machine operation

$$P\{\text{k machines are operating}\} = \binom{N}{k} A^k (1-A)^{N-k} . \qquad (33)$$

29. Clearly, the expected production difference between operations started in the same state is zero. Hence, in this example the (0, 0) and (1, 1) IC's yield E[production difference] = 0. Thus, the probability density of the E[production difference] for this example is the following: $-r/(\lambda+\mu)$ with probability $A(1-A)$, 0 with probability $A^2+(1-A)^2$, and $r/(\lambda+\mu)$ with probability $A(1-A)$. Since the mean value of this variable is zero, the variance is given as

$$[2r^2/(\lambda+\mu)^2]A(1-A) \qquad (34)$$

or, from (27), the variance of E[production difference] is

$$\frac{2r^2\lambda\mu}{(\lambda+\mu)^4} , \qquad (35)$$

and the SD is

$$\frac{(2\lambda\mu)^{1/2}r}{(\lambda+\mu)^2} . \qquad (36)$$

As a specific numerical example, let r=1 part per minute, MTBF = 100 minutes. MTTR = 25 minutes. Then, from (32), the expected range in production differences is 40 parts and, from (36), the standard deviation 11.3 parts.

This yields a required capacity of 85 spaces. To examine the marginal advantage in productivity for this buffer size, I ran GS.BUF at a buffer capacity of 85 and, again at 80. The productivities in these instances are 0.7426 and 0.7402, respectively. This implies a productivity change of about 0.04% per percent change in buffer capacity.

30. A Two-Machine Operation

As in the above example, it is not difficult to obtain an analytic solution to the time-dependent Markov model of a 2-machine production system. With two machines the system states are 0, 1, and 2. However, since

$$p_0(t) = 1 - p_1(t) - p_2(t) \quad , \tag{37}$$

only two equations are required to describe the system. Dropping the notation for explicit time dependence,

$$\dot{p}_1 = 2\lambda(1-p_1-p_2) - (\lambda+\mu)p_1 + 2\mu p_2 \tag{38a}$$

$$\dot{p}_2 = \lambda p_1 - 2\mu p_2 \quad . \tag{38b}$$

After some manipulation of these equations and using Laplace transforms, one obtains

$$p_1(t) = \frac{2\lambda\mu}{(\lambda+\mu)^2} + Ae^{-(\lambda+\mu)t} + Be^{-2(\lambda+\mu)t} \quad , \tag{39a}$$

with

$$A = \frac{a_0 - a_1(\lambda+\mu)}{\lambda+\mu} - \frac{4\lambda\mu}{(\lambda+\mu)^2} \tag{39b}$$

$$B = -\frac{a_0 - 2a_1(\lambda+\mu)}{\lambda+\mu} + \frac{2\lambda\mu}{(\lambda+\mu)^2} \tag{39c}$$

$$a_0 = \dot{p}_1(0) + 3(\lambda+\mu)p_1(0)$$

or

$$a_0 = 2\lambda p_0(0) + 2(\lambda+\mu)p_1(0) + 2\mu p_2(0) \tag{39d}$$

$$a_1 = p_1(0) \quad . \tag{39e}$$

$$p_2(t) = p_2(0)e^{-2\mu t} + \frac{\lambda^2}{(\lambda+\mu)^2} - \frac{\lambda^2}{(\lambda+\mu)^2} e^{-2\mu t} +$$

$$\frac{A}{\lambda-\mu} (e^{-2\mu t} - e^{-(\lambda+\mu)t}) + \frac{B}{2} (e^{-2\mu t} - e^{-2(\lambda+\mu)t}) \quad . \tag{40}$$

Steady-state results are obtained from (37), (39), and (40) by allowing t to approach $\infty$.

$$p_0(\infty) = \frac{\mu^2}{(\lambda+\mu)^2} \tag{41a}$$

$$p_1(\infty) = \frac{2\lambda\mu}{(\lambda+\mu)^2} \tag{41b}$$

$$p_2(\infty) = \frac{\lambda^2}{(\lambda+\mu)^2} \quad . \tag{41c}$$

The expected number of machines operating in steady-state in this case is

$$E[\text{number operating}] = p_1(\infty) + 2p_2(\infty) \quad , \tag{42a}$$

$$= 2\lambda/(\lambda+\mu) \quad . \tag{42b}$$

Since the number of machines, N, is 2 in this example, as anticipated,

$$E[\text{number operating}] = NA \quad . \tag{43}$$

The expected number of machines operating equals the max number times the intrinsic availability <u>only if</u> the system is well maintained. In writing the system equations (38a and b), it is assumed that if N machines are down for repair, N repairmen are working. Thus, the transition rate from the 0 state to the 1 state is given as $N\lambda$. If there were only M repairmen available for machine repairs, where $M<N$, the largest transition rate from a lower to a higher state would be $M\lambda$. In the latter case, the system state probabilities would depend upon <u>both</u> the number of machines per operation <u>and</u> the number of repairmen. To avoid this complication, it is assumed that the system is well maintained. This assumption is not as restrictive as it may seem. Approximately correct results are obtained for far weaker conditions, as will be shown later.

31. <u>An N-Machine Operation</u>

The state transition diagram for the general case is shown in Figure 5. Note that the transition from the j th to the j+1 st states occurs at the rate N-j (machines down) times $\lambda$. The downward transition from the k th to the k-1 st state occurs at the rate k (more to fail) times the unit death rate $\mu$. As was done for the previous Markov models, the Kolmogorov equations for this model can be written by inspection from the state transition diagram. Since this process is quite straightforward, it is not repeated here. The general result, with the deletion of the zero (or null) state is the familiar form

$$\dot{\underline{p}}(t) = A\underline{p}(t) + \underline{c} \quad , \tag{44}$$

with $\underline{c}' = [N\lambda, 0, 0, \ldots, 0]$ ,

where $\underline{p}$ and $\underline{c}$ are (Nx1) and where A is (NxN).

32. The solution procedure employs numerical integration using the rectangle rule with time step h:

$$\underline{p}(t+h) = \underline{p}(t) + h\dot{\underline{p}}(t) \quad , \tag{45}$$

with $\dot{\underline{p}}(t)$ given by (44). For a small time step (h), this procedure was found to be slightly faster (and easier) to implement than to use a double step (2h) with Euler's rule with a predictor and a corrector. A time step h of 0.1 minute is used in BUF.CAP. Notationally, let the conditional expected value of the output from an operation at time t (from the IC) be denoted $\overline{x}(t)$. By definition,

$$\overline{x}(t) = \int_0^t r\sum_{k=1}^{N} kp_k(t)dt \quad , \tag{45}$$

where r is the unit machine rate for this operation. The integrand in (45) is the average rate of production -- $d\overline{x}/dt$ -- from this machine operation. This derivative is saved at time intervals of $\Delta$ for optional printing. Call the numerical approximation $\dot{\overline{x}}(i\Delta)$, with integer i. In performing the numerical integration to calculate $\overline{x}(t)$, it is unnecessary to use a step as small as h. To yield about the same relative precision as obtained in calculating $\underline{p}(t)$, one can use a step $\Delta$ -- called DELTAT in BUF.CAP -- of 0.5 minute with Euler's rule:

$$\overline{x}(t+\Delta) \cong 0.5\Delta[\dot{\overline{x}}(t+\Delta) + \dot{\overline{x}}(t)] \; . \tag{46}$$

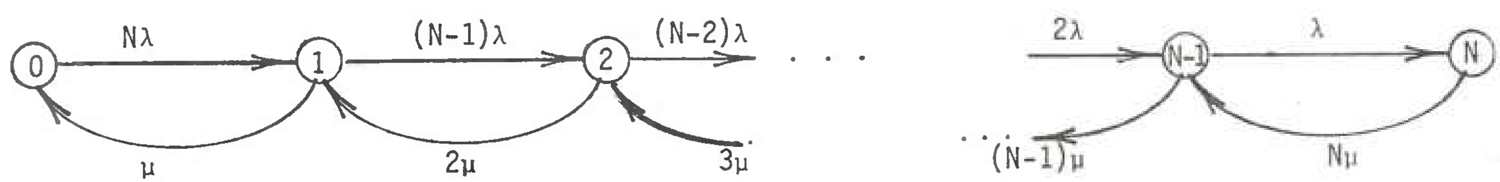



Figure 5 . State Transition Diagram for a Well Maintained N-Machine Operation

Denote by $\overline{x}_{ij}$ the conditional expected output of operation i with initial condition (IC) j. Then, the difference in expected production values is calculated at time $t_{max}$:

$$\overline{\Delta x_j} = \overline{x}_{1j}(t_{max}) - \overline{x}_{2j}(t_{max}) \quad . \tag{47}$$

This difference is essentially constant beyond $t_{max}$ = 2max ($MTTR_1$, $MTTR_2$). These expected production differences are treated in BUF.CAP in the manner described for the single-machine case (pgf. 28 ff). The probability distribution of $\overline{\Delta x_j}$ is displayed in order to facilitate the choice of buffer capacity on the basis of risk that a particular value is inadequate.

33. Sample Output from BUF.CAP

A sample run using the program BUF.CAP is shown in Table 8. The program input values are repeated at the top of the page. The output represents an abbreviated form of the available outputs. If the user chooses to display the trajectories of the conditional state probability vectors and cumulative production he may.

34. Stochastic simulation is used in the process of examining the advantage of increasing the buffer capacity beyond that required by BUF.CAP. Examples of the c.d.f.'s of buffer occupancy for several two-operation systems are shown in Figures 6 and 7. The max operation rate of both operations in each system is the same. In all cases shown here the machine service (or operation) time is constant. The probability distributions of buffer occupancy for two balanced systems are compared in Figure 6. One can observe that increasing the buffer capacity from 40 to 100 spaces, for the parameters shown, has the effect of significantly reducing the risk of encountering a full buffer. But, the probability of an empty buffer is nearly the same in both instances. Several comparisons are made in Figure 7. In these comparisons the buffer capacity is fixed at 40 spaces, and the operational rate is a constant 1 part/ minute. An effect on the c.d.f. of buffer occupancy occurs when the number of machines per operation increases. The effect of this increase is to reduce the probabilities associated with the extreme states of the buffer. This phenomenon is also shown in the output of BUF.CAP. Another observation of interest can be drawn from Figure 7. Thruout this study all analyses have been conducted assuming a well maintained system. As indicated, this implies at least as many repairmen as machines. For comparison, one simulation run was made with a system of 6 machines -- 4, in operation 1 and 2, in operation 2-- and with only one repairman. The c.d.f. of buffer occupancy for this case is shown in Figure 7. A rather small difference exists between the c.d.f. for this case and the c.d.f. for a comparable, well-maintained system. The reason

TABLE 8

SAMPLE OUTPUT FROM PROGRAM BUF.CAP WITH TERMINAL DIALOG DELETED

MACHINE INPUT DATA FOR BUFFER CAPACITY CALCULATION

| | OPERATION 1 | OPERATION 2 | |
|---|---|---|---|
| NO MACHINES | 4 | 4 | |
| MACH RATE | 0.25 | 0.25 | PARTS/MIN |
| MTBF | 100.00 | 100.00 | MINUTES |
| MTTR | 12.50 | 12.50 | MINUTES |
| AVAILABILITY | 88.89 | 88.89 | PERCENT |

REPAIR TIME (MINUTES) 50.0 ASSOCIATED STAT CONFIDENCE 0.982

EXPECTED BUFFER REQUIREMENTS ORDERED OVER ALL SYSTEM STATES
BASED ON A REPAIR TIME LAG OF 50.0 MINUTES

| ORDER INDEX | BUFFER SPACES | DIFFER SPACES | PROB DENS | CUML PROB | OPN 1 STATE | OPN 2 STATE |
|---|---|---|---|---|---|---|
| 1 | -10.9 | 0.0 | 0.0001 | 0.0001 | 0 | 4 |
| 2 | -8.2 | 2.7 | 0.0030 | 0.0031 | 1 | 4 |
| 3 | -8.2 | 2.7 | 0.0000 | 0.0032 | 0 | 3 |
| 4 | -5.5 | 5.5 | 0.0000 | 0.0032 | 0 | 2 |
| 5 | -5.5 | 5.5 | 0.0365 | 0.0397 | 2 | 4 |
| 6 | -5.5 | 5.5 | 0.0015 | 0.0413 | 1 | 3 |
| 7 | -2.7 | 8.2 | 0.1949 | 0.2361 | 3 | 4 |
| 8 | -2.7 | 8.2 | 0.0000 | 0.2361 | 0 | 1 |
| 9 | -2.7 | 8.2 | 0.0003 | 0.2364 | 1 | 2 |
| 10 | -2.7 | 8.2 | 0.0183 | 0.2547 | 2 | 3 |
| 11 | 0.0 | 10.9 | 0.0000 | 0.2547 | 1 | 1 |
| 12 | 0.0 | 10.9 | 0.0000 | 0.2547 | 0 | 0 |
| 13 | 0.0 | 10.9 | 0.0974 | 0.3521 | 3 | 3 |
| 14 | 0.0 | 10.9 | 0.0034 | 0.3556 | 2 | 2 |
| 15 | 0.0 | 10.9 | 0.3897 | 0.7453 | 4 | 4 |
| 16 | 2.7 | 13.7 | 0.0183 | 0.7636 | 3 | 2 |
| 17 | 2.7 | 13.7 | 0.0003 | 0.7639 | 2 | 1 |
| 18 | 2.7 | 13.7 | 0.0000 | 0.7639 | 1 | 0 |
| 19 | 2.7 | 13.7 | 0.1949 | 0.9587 | 4 | 3 |
| 20 | 5.5 | 16.4 | 0.0015 | 0.9603 | 3 | 1 |
| 21 | 5.5 | 16.4 | 0.0365 | 0.9968 | 4 | 2 |
| 22 | 5.5 | 16.4 | 0.0000 | 0.9968 | 2 | 0 |
| 23 | 8.2 | 19.1 | 0.0000 | 0.9969 | 3 | 0 |
| 24 | 8.2 | 19.1 | 0.0030 | 0.9999 | 4 | 1 |
| 25 | 10.9 | 21.9 | 0.0001 | 1.0000 | 4 | 0 |

AVERAGE BUFFER CHANGE _____ -0.00
STD DEV BUFFER CHANGE _____ 2.43

REQUIRED BUFFER CAPACITY = 32

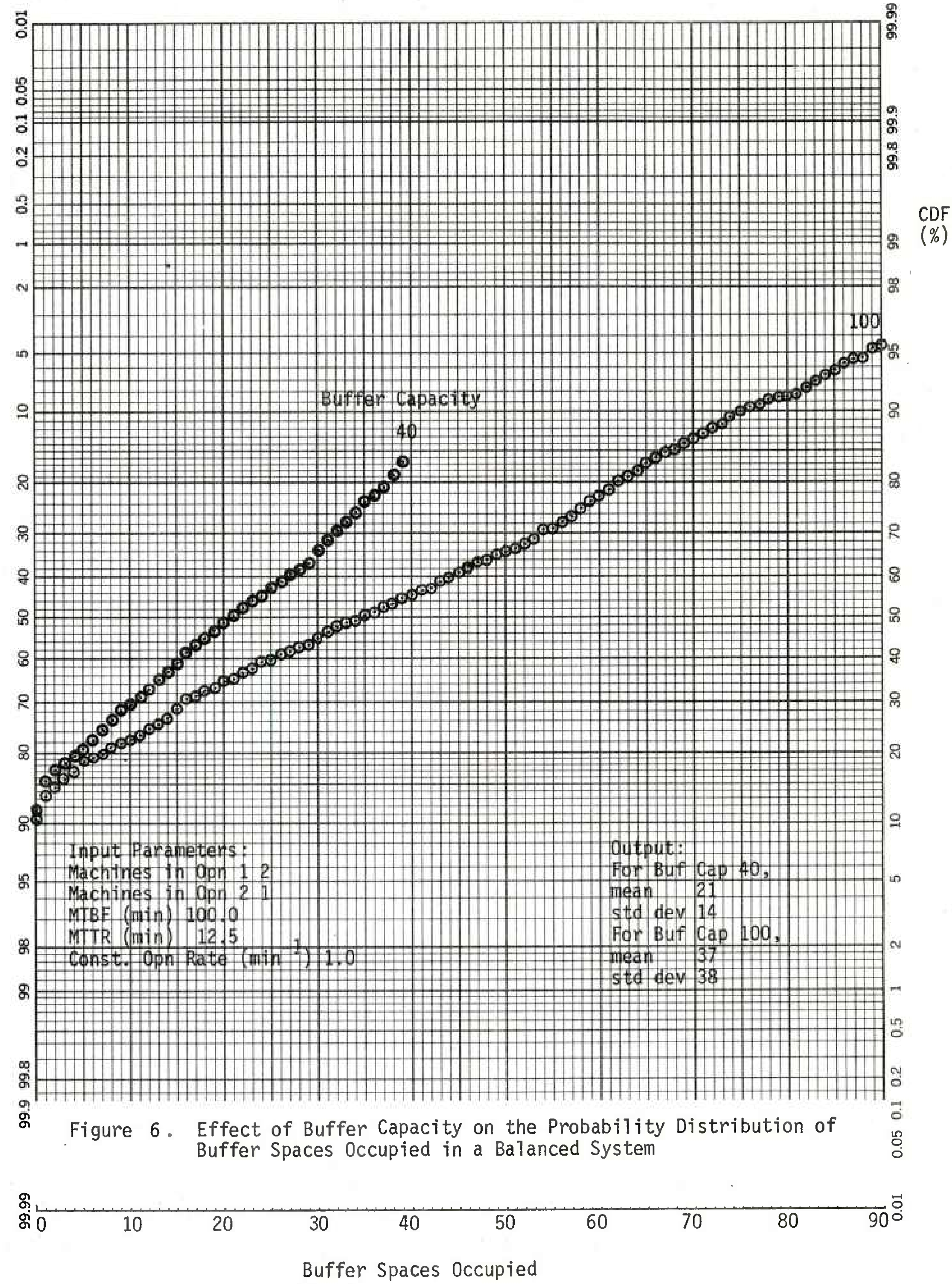



Figure 6. Effect of Buffer Capacity on the Probability Distribution of Buffer Spaces Occupied in a Balanced System

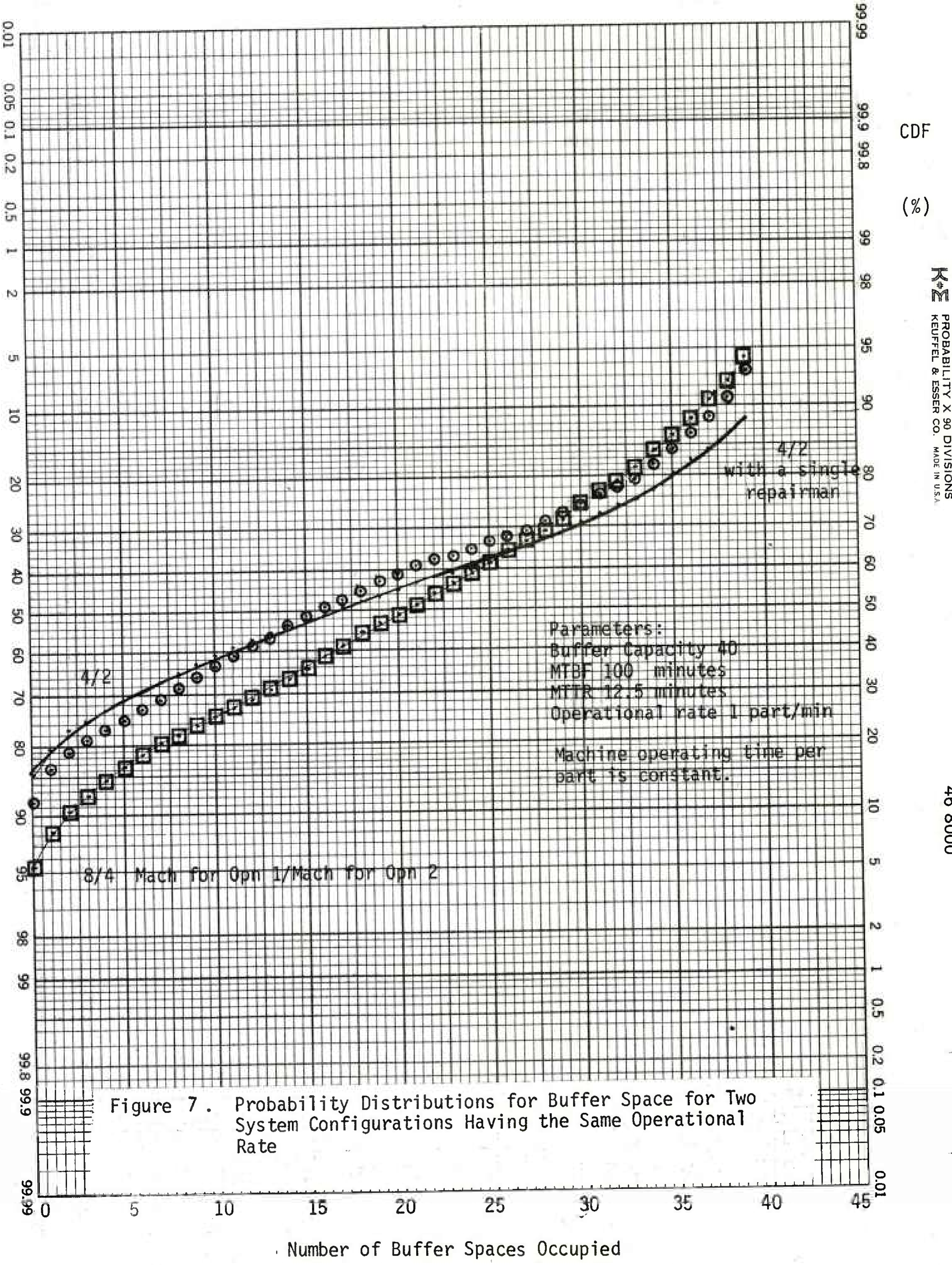



Figure 7. Probability Distributions for Buffer Space for Two System Configurations Having the Same Operational Rate

for the slight effect of additional repairmen here is that for this system the probability that other than one repairman is needed is only about 12%. Thus, the assumption of the analysis that the system is well maintained, may not be as restrictive as it first seems.

35. Conclusions Regarding BUF.CAP

(a) The risk of exceeding the buffer capacity requirement calculated by BUF.CAP is generally quite small -- typically less than 10%. Further, the marginal productivity change, evaluated at the required capacity, is nearly a constant 0.04% per % change in buffer capacity. (b) For a large -- say, >80 spaces -- buffer, BUF.CAP executes faster than GS.BUF. This may be a consideration for execution on small computers. Actually, the execution time of BUF.CAP depends on the number of machines at each operation, not on the required buffer capacity as such. (c) Unless the number of repairmen is at least equal to the total number of machines, there is a finite probability that machines must queue for repairs. If this happens, machine availability is not equal to intrinsic availability (A) and the expected number of machines is not equal to N·A. However, this situation is not as restrictive as it may seem. Both simulation and BUF.CAP show that the probability of exceeding a given buffer size decreases as the number of machines in each operation increases, at constant thruput. Under certain conditions the probability distribution of buffer occupancy is nearly unchanged by an increase in the number of repairmen. This occurs at a value of number of repairmen M such that prob (busy repairmen <=M)>0.9, approximately.

DISTRIBUTION LIST

| NO. OF COPIES | |
|---|---|
| 1 | HQDA (DAMO-ZA) |
| | WASH DC |
| | |
| 1 | Commander |
| | USADARCOM |
| | 5001 Eisenhower Ave. |
| | Alexandria, VA 22333 |
| 1 | ATTN: DRCRE-IP |
| 1 | DRCPA-S |
| 1 | DRCDP |
| | |
| | Director, US AMSAA |
| | Aberdeen Pg, MD 21005 |
| 1 | ATTN: Messrs Gilbert & Cohen DRXSY-DL |
| 1 | Ron Simmon DRXSY-R |
| | |
| | Commander USA |
| | Communications and Electronics Command |
| | Ft Monmouth, NJ 07703 |
| 1 | ATTN: DRSEL-PL-SA |
| | |
| | Commander CECOM (R&D) |
| | Ft Monmouth, NJ 07703 |
| 1 | ATTN: DRSEL-SAD |
| | |
| | Commander USAMICOM |
| | Redstone Arsenal |
| | AL 35809 |
| 1 | ATTN: DRSMI-DS |
| | |
| | Commander USATACOM |
| | Warren, MI 48090 |
| 1 | ATTN: DRSTA-V |
| | |
| | Office of Project Mgr |
| 1 | Cannon Arty Wpns, Dover DRCPM-CAWS |
| | NJ 07801 |
| | |
| | Commander, US Army Logistics Center |
| 1 | Fort Lee, VA 23801 ATTN: ATCL-S |
| | |
| | Director, US Army |
| 1 | Ammo Center SARAC-DO |
| | Savanna, IL 61074 |
| | |
| 1 | Commander USATECOM ATTN: DRSTE-SY |
| | Aberdeen PG, MD 21005 |
| | |
| | Director US AMETA |
| | Rock Is, IL 61299 |
| 1 | ATTN: DRXOM-QA |

DISTRIBUTION LIST (CONT)

| NO. OF COPIES | | |
|---|---|---|
| | Director, US Army | |
| 1 | Inventory Research Off | DRXMC-IRO |
| | Room 800, Custom House | |
| | 2nd & Chestnut Streets | |
| | Philadelphia, PA 19106 | |
| | | |
| 12 | Defence Technical Information Center | |
| | Cameron Station | |
| | Alexandria, VA 22314 | |
| | | |
| | Commander US AMCCOM (D) | |
| | Dover, NJ 07801 | |
| | ATTN: DRSMC | |
| 1 | John Mardo | -QAS |
| 1 | Bob Boner | -QAS |
| 1 | Mike Grum | -LCU-M |
| 1 | Carl Boane | -LCU-M |
| 1 | Kenneth Fuller | -QAR |
| 1 | Warren Aulh | -QAR |
| 1 | David Tran | -QA |
| 1 | William Menke | -QA |
| | | |
| | Commander | |
| | US Army Production Base | |
| | Modernization Agency | |
| | Dover, NJ 07801 | |
| 1 | ATTN: Sid Karlin | PBM-T |
| 1 | Jimmy Mok | PBM |
| | | |
| | Commander US AMCCOM (R) | |
| | Rock Is, IL 61299 | |
| | ATTN: DRSMC | |
| 1 | Craig Colledge | -IRP |
| 1 | George Cowan | -IRD |
| 1 | Mohan Devaraju | -IRI-M |
| 1 | Bob Gunnare | -IRI |
| 1 | Jim Kilar | -IRI |
| 1 | Matt Craighead | -PP |
| 1 | Don Eckman | -SAS |
| 1 | Fred Northey | -SAS |
| 1 | Jim Wasson | -SAS |
| 1 | Herman Michels | -SAL |
| 1 | Norm Trier | -SAL |
| 1 | Byron White | -SAA |
| 1 | Don Wahlgren | -ASM |
| 1 | Brian Walters | -ASA |
| 1 | Dennis Winegar | -LEM |
| 1 | Paul Woodhouse | -OPP |
| 1 | Dick Heider | -OPI |
| 3 | Sandi Paul | -LEP-L |

# ANNEXES

## COMPUTER SOURCE PROGRAMS

The program listings in these annexes are written in SIMSCRIPT 2.5. They do not employ any features unique to the PR1ME 550 minicomputer on which they were run. Cross-reference lists are included with the program statements to facilitate the identification of variable type and locations within the program. Since SIMSCRIPT is a language very English-like, programs can be followed easily without a flow chart. Therefore, no such diagrams are included. However, the major program blocks are announced via comments, which are distinguished from executable code by stating with double quote marks.

Potential users of these programs who do not have SIMSCRIPT compilers but do have FORTRAN compilers are assured that conversion to FORTRAN is straightforward. The code in FORTRAN is not much longer than that in SIMSCRIPT. If a FORTRAN version is implemented on a 32 bit (or less) machine, it is recommended that double-precision arithmetic be used. This is necessary to avoid truncation error when inverting large matrices.

Both main programs were designed for running interactively. Inputs are assigned following prompting messages sent to the terminal.

ANNEX 1

PROGRAM GS.BUF

A sketch of the method used to calculate the stochastic steady state of a simple production system is provided in the body of this memorandum under "Methodology Overview." Detailed system equations are derived in the section "Derivation of Equations for GS.BUF."

Input data is provided from the terminal in response to prompting messages such as "Input the operating rate for the 1st operation in parts per minute." Output is sent directly to the terminal for display. Since this output is often lengthy, it is recommended that a COMO file be established to display or print it.

```
MAIN ''FOR GS.BUF
DEFINE BUF.CAP, I, J, K, L, M, N, NS AS INTEGER VARIABLES
DEFINE ''STATE DESCRIPTION ARRAY'' SDA AS AN INTEGER, 2-DIMENSIONAL ARRAY
''
''FIRST INDEX OF SDA REFERS TO THE STATE NUMBER AND THE SECOND INDEX REFERS
''TO THE SYSTEM ELEMENT NUMBER--1, FOR OPERATION 1; 2, FOR THE BUFFER;
''AND 3, FOR OPERATION 2.
''
DEFINE BV, PV, AND BUF.STATE AS REAL, 1-DIMENSIONAL ARRAYS
DEFINE AM, BM, AND AMINV AS REAL, 2-DIMENSIONAL ARRAYS
''
''GET BUFFER CAPACITY FROM THE TERMINAL.
''
      PRINT 1 LINE THUS
INPUT THE INTEGER BUFFER CAPACITY.
      READ BUF.CAP
      IF BUF.CAP LE 2
         PRINT 1 LINE THUS
INPUT ERROR.  BUFFER CAPACITY IS TOO SMALL.
         STOP
      OTHERWISE
      LET M=BUF.CAP+1 ''BUFFER STATES
      LET NS=4*(BUF.CAP+2) ''NUMBER OF SYSTEM STATES
      LET N=NS-1 ''DIMENSION OF TRANSITION MATRIX
RESERVE SDA(*,*) AS NS BY 3
RESERVE AM(*,*) AS N BY N
RESERVE BM(*,*) AS NS BY NS
RESERVE BV(*) AS N
RESERVE BUF.STATE(*) AS M
''
''ASSIGN PARAMETER VALUES.
''
      PRINT 1 LINE THUS
INPUT THE OPERATING RATE FOR THE 1 ST OPERATION IN PARTS PER MINUTE.
      READ RATE1
      LET R1=RATE1
      PRINT 1 LINE THUS
INPUT THE OPERATING RATE FOR THE 2 ND OPERATION IN PARTS PER MINUTE.
      READ RATE2
      LET R2=RATE2
      PRINT 1 LINE THUS
INPUT THE MTBF FOR THE 1 ST OPERATION IN MINUTES.
      READ MTBF1
      PRINT 1 LINE THUS
INPUT THE MTBF FOR THE 2 ND OPERATION IN MINUTES.
      READ MTBF2
      PRINT 1 LINE THUS
INPUT THE MTTR FOR THE 1 ST OPERATION IN MINUTES.
      READ MTTR1
      PRINT 1 LINE THUS
INPUT THE MTTR FOR THE 2 ND OPERATION IN MINUTES.
      READ MTTR2
''
''CALCULATE MARKOV PARAMETERS.
''
      LET MU1=1.0/MTBF1
      LET MU2=1.0/MTBF2
```

```
      LET LA1=1.0/MTTR1
      LET LA2=1.0/MTTR2
      LET S1=1.0
      LET S2=1.0
      FOR I=1 TO 5 DO
           FOR J=1 TO 3 DO
                LET SDA(I,J)=0
           LOOP ''OVER J
      LOOP ''OVER I
      LET SDA(1,3)=1
      LET SDA(2,1)=2
      LET SDA(2.3)=1
      LET SDA(4,3)=2
      LET SDA(5,1)=2
      LET SDA(6,1)=2
      LET SDA(6.2)=0
      LET SDA(6,3)=2
      LET SDA(7,1)=0
      LET SDA(7,2)=1
      LET SDA(7,3)=0
      LET SDA(8.1)=0
      LET SDA(8,2)=1
      LET SDA(8,3)=2
      LET SDA(9,1)=2
      LET SDA(9,2)=1
      LET SDA(9.3)=0
      LET SDA(10,1)=2
      LET SDA(10.2)=1
      LET SDA(10,3)=2
 FOR K=2 TO BUF.CAP-1 DO
      LET SDA(7+4*(K-1),1)=0
      LET SDA(7+4*(K-1),2)=K
      LET SDA(7+4*(K-1),3)=0
      LET SDA(8+4*(K-1),1)=0
      LET SDA(8+4*(K-1),2)=K
      LET SDA(8+4*(K-1),3)=2
      LET SDA(9+4*(K-1),1)=2
      LET SDA(9+4*(K-1),2)=K
      LET SDA(9+4*(K-1),3)=0
      LET SDA(10+4*(K-1),1)=2
      LET SDA(10+4*(K-1),2)=K
      LET SDA(10+4*(K-1),3)=2
 LOOP ''OVER BUFFER STATES
      LET SDA(NS-5,1)=0
      LET SDA(NS-5,2)=BUF.CAP
      LET SDA(NS-5,3)=0
      LET SDA(NS-4,1)=0
      LET SDA(NS-4,2)=BUF.CAP
      LET SDA(NS-4,3)=2
      LET SDA(NS-3,1)=2
      LET SDA(NS-3,2)=BUF.CAP
      LET SDA(NS-3,3)=0
      LET SDA(NS-2,1)=2
      LET SDA(NS-2,2)=BUF.CAP
      LET SDA(NS-2,3)=2
      LET SDA(NS-1.1)=1
      LET SDA(NS-1,2)=BUF.CAP
```

```
        LET SDA(NS-1,3)=0
        LET SDA(NS,1)=1
        LET SDA(NS,2)=BUF.CAP
        LET SDA(NS,3)=2
        LET AVAIL1=MTBF1/(MTBF1+MTTR1)
        LET AVAIL2=MTBF2/(MTBF2+MTTR2)
    ''
    ''ECHO PARAMETER VALUES.
    ''
        SKIP 2 LINES
        PRINT 12 LINES WITH BUF.CAP,RATE1,MTBF1,MTTR1,AVAIL1,RATE2,MTBF2,MTTR2,
        AVAIL2 THUS
  PARAMETER VALUES FOR THE STEADY-STATE BUFFER MODEL

    BUFFER CAPACITY    * SPACES

OPERATION     MACH RATE     MTBF       MTTR       AVAIL-
NUMBER        (PARTS/MIN)   (MIN)      (MIN)      ABILITY
-----------------------------------------------------------
    1          ***.***      ***.**     ***.**     *.*****
    2          ***.***      ***.**     ***.**     *.*****
-----------------------------------------------------------

    ''
    ''FILL RIGHT HAND VECTOR.
    ''
        FOR I=1 TO N, LET BV(I)=0.0
    ''
    ''FILL ELEMENTS OF THE STATE TRANSITION MATRIX.
    ''
    FOR I=1 TO NS DO
        FOR J=1 TO NS DO
            LET BM(I,J)=0.0
        LOOP ''OVER J
    LOOP ''OVER I
        LET BM(1,1)=-LA1
        LET BM(1,2)=MU1
        LET BM(1,4)=R2
        LET BM(2,1)=LA1
        LET BM(2,2)=-(MU1+R1)
        LET BM(2,6)=S2*R2
        LET BM(3,3)=-(LA1+LA2)
        LET BM(3,4)=MU2
        LET BM(3,5)=MU1
        LET BM(4,3)=LA2
        LET BM(4,4)=-(LA1+MU2+R2)
        LET BM(4,6)=MU1
        LET BM(4,8)=R2
        LET BM(5,3)=LA1
        LET BM(5,5)=-(LA2+MU1+R1)
        LET BM(5,6)=MU2
        LET BM(6,2)=R1
        LET BM(6,4)=LA1
        LET BM(6,5)=LA2
        LET BM(6,6)=-(MU1+MU2+S1*R1+S2*R2)
        LET BM(6,10)=S2*R2
```

```
      LET BM(7,7)=-(LA1+LA2)
      LET BM(7,8)=MU2
      LET BM(7,9)=MU1
      LET BM(8,7)=LA2
      LET BM(8,8)=-(LA1+MU2+R2)
      LET BM(8,10)=MU1
      LET BM(8,12)=R2
      LET BM(9,5)=R1
      LET BM(9,7)=BM(5,3)
      LET BM(9,9)=BM(5,5)
      LET BM(9,10)=BM(5,6)
      LET BM(10,6)=S1*R1
      LET BM(10,8)=BM(6,4)
      LET BM(10,9)=BM(6,5)
      LET BM(10,10)=BM(6,6)
      LET BM(10,14)=BM(6,10)
  FOR I=11 TO NS-5 DO
      FOR J=I-4 TO NS DO
          LET BM(I,J)=BM(I-4,J-4)
      LOOP ''OVER COLUMNS (J)
  LOOP ''OVER ROWS (I)
      LET BM(NS-4,NS-5)=LA2
      LET BM(NS-4,NS-4)=-(LA1+MU2+R2)
      LET BM(NS-4,NS-2)=MU1
      LET BM(NS-3,NS-7)=R1
      LET BM(NS-3,NS-5)=LA1
      LET BM(NS-3,NS-3)=-(LA2+MU1+R1)
      LET BM(NS-3,NS-2)=MU2
      LET BM(NS-2,NS-6)=S1*R1
      LET BM(NS-2,NS-4)=LA1
      LET BM(NS-2,NS-3)=LA2
      LET BM(NS-2,NS-2)=-(MU1+MU2+S1*R1+S2*R2)
      LET BM(NS-2,NS)=R2
      LET BM(NS-1,NS-3)=R1
      LET BM(NS-1,NS-1)=-LA2
      LET BM(NS-1,NS)=MU2
      LET BM(NS,NS-2)=S1*R1
      LET BM(NS,NS-1)=LA2
      LET BM(NS,NS)=-(MU2+R2)
  ''
  ''CHECK THAT ALL COLUMN SUMS OF B(*,*) ARE ZERO.
  ''
  FOR J=1 TO N+1 DO
      LET SUM=0.0
      FOR I=1 TO N+1, ADD BM(I,J) TO SUM
      IF ABS.F(SUM) GE 1.0/10.0**8
          PRINT 1 LINE WITH J AND SUM THUS
ERROR :  SUM OF ** TH COLUMN OF STATE TRANSITION MATRIX = ...........
          FOR K=1 TO N+1 DO
              PRINT 1 LINE WITH K, BM(K,J) THUS
                  .***.****
 ***
          LOOP ''OVER K
          STOP
      OTHERWISE
  LOOP ''OVER COLUMNS OF BM(*,*)
  ''
  ''FOR ROWS BM(*,1) NE 0.0 PERFORM ROW SUM OPERATIONS.
```

```
''
FOR I=2 TO N+1 DO
    IF BM(I,1) NE 0.0
        LET CON=BM(I,1)
        FOR J=1 TO N+1, SUBTRACT CON FROM BM(I,J)
        SUBTRACT CON FROM BV(I-1)
    ALWAYS
LOOP ''OVER ROWS OF BM(*,*)
''
''COPY THE SUBMATRIX OF BM(*,*) INTO AM(*,*).
''
FOR I=1 TO N DO
    FOR J=1 TO N DO
        LET AM(I,J)=BM(I+1,J+1)
    LOOP ''OVER J
LOOP ''OVER I
    RELEASE BM(*,*)
''  RESERVE AMINV(*,*) AS N BY N
''
''CALCULATE THE MATRIX INVERSE OF AM(*,*).
''
    CALL MAT.INVERSE (AM(*,*),N) YIELDING AMINV(*,*)
''  RESERVE PV(*) AS N
''
''CALCULATE THE PROBABILITY STATE VECTOR PV(*).
''
    CALL MAT.VEC.MPY (AMINV(*,*), BV(*), N) YIELDING PV(*)
''
''CALCULATE THE PROBABILITY OF THE NULL STATE.
''
    LET P.NULL=1.0
    FOR I=1 TO N, SUBTRACT PV(I) FROM P.NULL
''
''CHECK FOR VALID PROBABILITY.
''
    IF P.NULL <0.0 OR P.NULL >1.0
        PRINT 1 LINE WITH P.NULL THUS
ERROR IN SS.BUF.  P.NULL = ..........
        STOP
    OTHERWISE
''
''GET BUFFER STATE PROBABILITIES.
''
    FOR K=1 TO M, LET BUF.STATE(K)=0.0
    ADD P.NULL TO BUF.STATE(1)
    FOR K=1 TO 5, ADD PV(K) TO BUF.STATE(1)
    FOR J=2 TO M-1 DO
        FOR K=1 TO 4 DO
            ADD PV(K+(J-1)*4+1) TO BUF.STATE(J)
        LOOP ''OVER (K) SUBSTATES
    LOOP ''OVER (J) STATES OF THE BUFFER
    FOR K=1 TO 6, ADD PV(K+4*(M-1)+1) TO BUF.STATE(M)
    LET P.OPN2.WAIT=PV(1)+P.NULL
    LET P.OPN1.WAIT=PV(NS-1)+PV(NS-2)
    LET P.SYS.WAIT=P.OPN1.WAIT+P.OPN2.WAIT
    LET P.OPN2.DN=0.0
    FOR K=1 TO (NS-2)/2 DO
```

```
          ADD PV(2*K) TO P.OPN2.DN
      LOOP ''TO GET PROB THAT 2 ND OPERATION IS BEING REPAIRED
      LET P.NO.OP=P.OPN2.DN+P.OPN2.WAIT
  ''
  ''PRINT HEADINGS.
  ''
      SKIP 4 LINES
      PRINT 6 LINES THUS
  PROBABILITIES OF OPN1-BUFFER-OPN2 SYSTEM STATES

------------------------------------------------------------------------
STATE    STATE      STATE DESCRIPTION
NUMBR    PROB       OPN1   BUF   OPN2
------------------------------------------------------------------------
      PRINT 1 LINE WITH P.NULL, SDA(I,1), SDA(I,2), SDA(I,3) THUS
 1        *.****      *      *      *
  FOR I=1 TO N DO
      PRINT 1 LINE WITH I+1, PV(I), SDA(I+1,1), SDA(I+1,2), SDA(I+1,3) THUS
**        *.****      *      *      *
  LOOP ''OVER I
      PRINT 9 LINES THUS
------------------------------------------------------------------------

    BUFFER STEADY-STATE OCCUPANCY PROBABILITIES

------------------------------------------------------------------------
BUFFER   STATE
STATE    PROB
------------------------------------------------------------------------
  FOR I=1 TO M DO
      PRINT 1 LINE WITH I-1, BUF.STATE(I)
      THUS
**       *.****
  LOOP ''OVER BUFFER STATES
      PRINT 5 LINES WITH P.OPN1.WAIT, P.OPN2.WAIT, P.SYS.WAIT THUS
------------------------------------------------------------------------

PROB THAT OPERATION 1 MUST WAIT WITH BUFFER FULL ____ *.****
PROB THAT OPERATION 2 MUST WAIT WITH BUFFER EMPTY ___ *.****
PROB THAT AN OPN IS NOT OPERATING DUE TO BUFFER _____ *.****
      LET PRODUCTIVITY=1.0-P.NO.OP
      LET E.PROD.RATE=RATE2*PRODUCTIVITY
      PRINT 4 LINES WITH P.OPN2.DN, P.NO.OP, PRODUCTIVITY, E.PROD.RATE THUS
PROB THAT OPERATION 2 IS DOWN FOR REPAIRS ___________ *.****
PROB THAT THE SYSTEM HAS NO OUTPUT RATE _____________ *.****
SYSTEM PRODUCTIVITY _________________________________ *.****
SYSTEM PRODUCTION CAPACITY (PARTS/MINUTE) ___________ **.****
      SKIP 4 LINES
      STOP
  END ''MAIN FOR SS.BUF
```

```
                    C R O S S - R E F E R E N C E

  NAME                    TYPE                                  MODE     LINE NUMBERS OF REFERENCES

ABS.F                     ROUTINE                               INTEGER   197
AM                        RECURSIVE VARIABLE   WORD  13  (2-D)  DOUBLE     10    24*   220   228
AMINV                     RECURSIVE VARIABLE   WORD  15  (2-D)  DOUBLE     10   228   233
AVAIL1                    RECURSIVE VARIABLE   WORD  55         DOUBLE    111   117
AVAIL2                    RECURSIVE VARIABLE   WORD  57         DOUBLE    112   117
BM                        RECURSIVE VARIABLE   WORD  14  (2-D)  DOUBLE     10    25*   128   131   132   133   134
                                                                          135   136   137   138   139   140   141
                                                                          142   143   144   145   146   147   148
                                                                          149   150   151   152   153   154   155
                                                                          156   157   158   159   160*  161*  162*
                                                                          163   164*  165*  166*  167*  170*  173
                                                                          174   175   176   177   178   179   180
                                                                          181   182   183   184   185   186   187
                                                                          188   189   190   196   200   209   210
                                                                          211*  220   223*
BUF.CAP                   RECURSIVE VARIABLE   WORD   1         INTEGER     2    15    16    20    21    79    94
                                                                           97   100   103   106   109   117
BUF.STATE                 RECURSIVE VARIABLE   WORD  12  (1-D)  DOUBLE      9    27*   249   250*  251*  254*  257*
                                                                          277
BV                        RECURSIVE VARIABLE   WORD  10  (1-D)  DOUBLE      9    26*   122   212*  233
CON                       RECURSIVE VARIABLE   WORD  62         DOUBLE    210   211   212
E.PROD.RATE               RECURSIVE VARIABLE   WORD  79         DOUBLE    282   283
I                         RECURSIVE VARIABLE   WORD   2         INTEGER     2    54*    56   122*  126*  128   168*
                                                                          169   170*  196*  208*  209   210   211*
                                                                          212*  218*  220*  238*  272*  273*  276*
                                                                          277*
J                         RECURSIVE VARIABLE   WORD   3         INTEGER     2    55*    56   127*  128   169*  170*
                                                                          194*  196   198   200   211*  219*  220*
                                                                          252*  254*
K                         RECURSIVE VARIABLE   WORD   4         INTEGER     2    79*    80    81*   82    83    84*
                                                                           85    86    87*   88    89    90*   91
                                                                          199*  200*  249*  251*  253*  254   257*
                                                                          262*  263
L                         RECURSIVE VARIABLE   WORD   5         INTEGER     2
LA1                       RECURSIVE VARIABLE   WORD  47         DOUBLE     50   131   134   137   141   144   148
                                                                          152   156   174   177   181
LA2                       RECURSIVE VARIABLE   WORD  49         DOUBLE     51   137   140   145   149   152   155
                                                                          173   178   182   186   189
M                         RECURSIVE VARIABLE   WORD   6         INTEGER     2    20    27   249   252   257*  276
MAT.INVERSE               ROUTINE                               INTEGER   228
MAT.VEC.MPY               ROUTINE                               INTEGER   233
MTBF1                     RECURSIVE VARIABLE   WORD  35         DOUBLE     38    48   111*  117
MTBF2                     RECURSIVE VARIABLE   WORD  37         DOUBLE     40    49   112*  117
MTTR1                     RECURSIVE VARIABLE   WORD  39         DOUBLE     42    50   111   117
MTTR2                     RECURSIVE VARIABLE   WORD  41         DOUBLE     44    51   112   117
MU1                       RECURSIVE VARIABLE   WORD  43         DOUBLE     48   132   135   139   142   145   150
                                                                          154   157   175   178   183
MU2                       RECURSIVE VARIABLE   WORD  45         DOUBLE     49   138   141   146   150   153   156
                                                                          174   179   183   187   190
N                         RECURSIVE VARIABLE   WORD   7         INTEGER     2    22    24*    26   122   194   196
                                                                          199   208   211   218   219   228   233
                                                                          238   272
NS                        RECURSIVE VARIABLE   WORD   8         INTEGER     2    21    22    23    25*    93    94
```

```
                                                                  95    96    97    98    99   100   101
                                                                 102   103   104   105   106   107   108
                                                                 109   110   126   127   168   169   173*
                                                                 174*  175*  176*  177*  178*  179*  180*
                                                                 181*  182*  183*  184*  185*  186*  187*
                                                                 188*  189*  190*  259*  262
P.NO.OP       RECURSIVE VARIABLE    WORD  74         DOUBLE      265   281   283
P.NULL        RECURSIVE VARIABLE    WORD  64         DOUBLE      237   238*  242*  243   250   258   271
P.OPN1.WAIT   RECURSIVE VARIABLE    WORD  68         DOUBLE      259   260   280
P.OPN2.DN     RECURSIVE VARIABLE    WORD  72         DOUBLE      261   263*  265   283
P.OPN2.WAIT   RECURSIVE VARIABLE    WORD  66         DOUBLE      258   260   265   280
P.SYS.WAIT    RECURSIVE VARIABLE    WORD  70         DOUBLE      260   280
PRODUCTIVITY  RECURSIVE VARIABLE    WORD  77         DOUBLE      281   282   283
PV            RECURSIVE VARIABLE    WORD  11  (1-D)  DOUBLE        9   233   238   251   254   257   258
                                                                 259*  263   273
R1            RECURSIVE VARIABLE    WORD  29         DOUBLE       33   135   145   147   150   159   163
                                                                 176   178   180   183   185   188
R2            RECURSIVE VARIABLE    WORD  33         DOUBLE       36   133   136   141   143   150   151
                                                                 156   158   174   183   184   190
RATE1         RECURSIVE VARIABLE    WORD  27         DOUBLE       32    33   117
RATE2         RECURSIVE VARIABLE    WORD  31         DOUBLE       35    36   117   282
S1            RECURSIVE VARIABLE    WORD  51         DOUBLE       52   150   163   180   183   188
S2            RECURSIVE VARIABLE    WORD  53         DOUBLE       53   136   150   151   183
SDA           RECURSIVE VARIABLE    WORD   9  (2-D)  INTEGER       3    23*   56    59    60    61    62
                                                                  63    64    65    66    67    68    69
                                                                  70    71    72    73    74    75    76
                                                                  77    78    80    81    82    83    84
                                                                  85    86    87    88    89    90    91
                                                                  93    94    95    96    97    98    99
                                                                 100   101   102   103   104   105   106
                                                                 107   108   109   110   271*  273*
SUM           RECURSIVE VARIABLE    WORD  60         DOUBLE      195   196*  197   198
UIB.W         IMPLIED SUBSCRIPT     SYS    4         INTEGER     116   269   284
```

```
 1  ROUTINE FOR MAT.VEC.MPY (AM, BV, NELMTS) YIELDING CV
 2  ''
 3  ''ROUTINE TO MULTIPLY THE SQUARE MATRIX AM , OF NELMTS BY NELMTS,
 4  ''BY THE VECTOR BV (NELMTS BY 1), YIELDING THE VECTOR CV (NELMTS BY 1).
 5  ''
 6  DEFINE I, J, K, NELMTS AS INTEGER VARIABLES
 7  DEFINE BV AND CV AS REAL, 1-DIMENSIONAL ARRAYS
 8  DEFINE AM AS A REAL, 2-DIMENSIONAL ARRAY
 9  RESERVE AM(*,*) AS NELMTS BY NELMTS
10  RESERVE BV(*) AS NELMTS
11  RESERVE CV(*) AS NELMTS
12  FOR I=1 TO NELMTS DO
13      LET CV(I)=0.0
14      FOR K=1 TO NELMTS DO
15          ADD AM(I,K)*BV(K) TO CV(I)
16      LOOP ''OVER K
17  LOOP ''OVER I
18  RETURN
19  END ''OF ROUTINE MAT.VEC.MPY
```

C R O S S - R E F E R E N C E

| NAME | TYPE | | | | MODE | LINE NUMBERS OF REFERENCES | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| AM | ARGUMENT | NO. | 1 | (2-D) | DOUBLE | 1 | 8 | 9* | 15 | | | |
| BV | ARGUMENT | NO. | 2 | (1-D) | DOUBLE | 1 | 7 | 10* | 15 | | | |
| CV | ARGUMENT | NO. | 4 | (1-D) | DOUBLE | 1 | 7 | 11* | 13 | 15* | | |
| I | RECURSIVE VARIABLE | WORD | 1 | | INTEGER | 6 | 12* | 13 | 15* | | | |
| J | RECURSIVE VARIABLE | WORD | 2 | | INTEGER | 6 | | | | | | |
| K | RECURSIVE VARIABLE | WORD | 3 | | INTEGER | 6 | 14* | 15* | | | | |
| MAT.VEC.MPY | ROUTINE | | | | INTEGER | 1 | | | | | | |
| NELMTS | ARGUMENT | NO. | 3 | | INTEGER | 1 | 6 | 9* | 10 | 11 | 12 | 14 |

```
ROUTINE FOR MAT.INVERSE (AM, N) YIELDING BM
''
''ROUTINE TO OBTAIN THE INVERSE OF THE N BY N MATRIX AM VIA THE
''COMPACT FORM OF THE GAUSS-JORDAN METHOD.  INVERSE IS RETURNED
''AS BM.  AM IS LEFT UNCHANGED.
''
DEFINE I, J, K, N AS INTEGER VARIABLES
DEFINE AM AND BM AS REAL, 2-DIMENSIONAL ARRAYS
RESERVE AM(*,*) AS N BY N
RESERVE BM(*,*) AS N BY N
''
''COPY AM INTO BM.  BM IS USED FOR GAUSSIAN REDUCTION.
''
FOR I=1 TO N DO
     FOR J=1 TO N DO
          LET BM(I,J)=AM(I,J)
     LOOP ''OVER J
LOOP ''OVER I
FOR I=1 TO N DO
     LET P=BM(I,I)
     IF P=0.0
          PRINT 2 LINES WITH I THUS
ERROR IN ROUTINE MAT.INVERSE.  THE ** TH DIAGONAL ELEMENT IS ZERO.
THE MATRIX CANNOT BE INVERTED.
          STOP
     OTHERWISE
     LET BM(I,I)=1.0
     FOR J=1 TO N DO
          LET BM(I,J)=BM(I,J)/P
     LOOP ''OVER J
     FOR J=1 TO N DO ''THE SECOND J-LOOP
          IF J=I
               GO TO EOJ ''END OF J-LOOP
          OTHERWISE
          LET P=BM(J,I)
          LET BM(J,I)=0.0
          FOR K=1 TO N DO
               SUBTRACT P*BM(I,K) FROM BM(J,K)
          LOOP ''OVER K
 'EOJ'LOOP ''OVER J
 LOOP ''OVER I
 RETURN
 END ''OF ROUTINE MAT.INVERSE
```

```
ROUTINE MAT.INVERSE               CACI SIMSCRIPT II.5 for PRIME Systems, Release 2.1        Page   12
Options = SEQUENCE,ID,SUBCHK,XREF,NOEXPLIST,TRACE3                27 JUL 1983 15:46:14

                        C R O S S - R E F E R E N C E

  NAME                           TYPE                           MODE     LINE NUMBERS OF REFERENCES

AM                           ARGUMENT           NO.   1  (2-D) DOUBLE     1     8     9*   16
BM                           ARGUMENT           NO.   3  (2-D) DOUBLE     1     8    10*   16    20    25    27*
                                                                         33    34    36*
EOJ                          LABEL                                       31    38
I                            RECURSIVE VARIABLE WORD  1        INTEGER    7    14*   16*   19*   20*   22    25*
                                                                         27*   30    33    34    36
J                            RECURSIVE VARIABLE WORD  2        INTEGER    7    15*   16*   26*   27*   29*   30
                                                                         33    34    36*
K                            RECURSIVE VARIABLE WORD  3        INTEGER    7    35*   36*
MAT.INVERSE                  ROUTINE                           INTEGER    1
N                            ARGUMENT           NO.   2        INTEGER    1     7     9*   10*   14    15    19
                                                                         26    29    35
P                            RECURSIVE VARIABLE WORD  4        DOUBLE    20    21    27    33    36
```

# ANNEX 2

## PROGRAM BUF.CAP

BUF.CAP calculates a recommended capacity for a buffer separating two tandem production operations. The methods employed in BUF.CAP are outlined in the body of this memorandum under "A Second Approach." The system equations for each production operation are displayed in various places depending on the number of machines (N) in a given operation. For N=1, see equation (26); for N=2, see (39, 40); and for N>2, a general form is provided in equation (44). The elements of the matrix A in (44) are given in the source program in LET statements.

Input data is provided from the terminal in response to prompting messages such as "Input the number of machines in 1st operation." To assure a system balance, the machine rates <u>assumed</u> in BUF.CAP, i.e. calculated endogenously, may not equal the <u>actual</u> or desired rates. To account for this absence of input rates, the buffer requirement from BUF.CAP must be scaled in proportion to the ratio of actual thruput to assumed thruput. The program output is sent to the terminal for display. This includes an echo of all program inputs. No output files are created. If the output is to be saved, a COMO file must be established.

```
Options = SEQUENCE,ID,SUBCHK,XREF,NOEXPLIST,TRACE3

   1  **BUF.CAP
   2  PREAMBLE
   3  NORMALLY MODE IS REAL
   4  DEFINE BINOM.DENS AS A REAL FUNCTION GIVEN 3 ARGUMENTS
   5  DEFINE BINOM.DIST AS A REAL FUNCTION GIVEN 3 ARGUMENTS
   6  END
```

C R O S S - R E F E R E N C E

| NAME | TYPE | MODE | LINE NUMBERS OF REFERENCES |
|---|---|---|---|
| BINOM.DENS | ROUTINE | DOUBLE | 4 |
| BINOM.DIST | ROUTINE | DOUBLE | 5 |

```
  MAIN
  ''
  ''DRIVER FOR BUF.CAP
  ''
  DEFINE I, IPRINT, AND REQ.CAP AS INTEGER VARIABLES
  DEFINE NV AS AN INTEGER, 1-DIMENSIONAL ARRAY
  DEFINE RATEV, MTBFV, MTTRV, AVAILV AS REAL, 1-DIMENSIONAL ARRAYS
  RESERVE NV(*) AS 2
  RESERVE RATEV(*), MTBFV(*), MTTRV(*), AVAILV(*) AS 2
  ''
  ''GET INPUT PARAMETERS FROM THE TERMINAL.
  ''
        PRINT 1 LINE THUS
INPUT THE NUMBER OF MACHINES IN 1 ST OPERATION.
        READ NV(1)
        PRINT 1 LINE THUS
INPUT THE NUMBER OF MACHINES IN 2 ND OPERATION.
        READ NV(2)
        FOR I=1 TO 2 DO
              PRINT 1 LINE WITH I THUS
INPUT THE MEAN TIME BETWEEN FAILURES (MINUTES) FOR MACHINES OF OPERATION  *.
              READ MTBFV(I)
              PRINT 1 LINE WITH I THUS
INPUT THE MEAN TIME TO REPAIR MACHINES OF OPERATION  *.
              READ MTTRV(I)
              LET AVAILV(I)=MTBFV(I)/(MTBFV(I)+MTTRV(I))
  LOOP ''OVER OPERATIONS
        LET MAXA=MAX.F(AVAILV(1),AVAILV(2))
        FOR I=1 TO 2, LET RATEV(I)=1.0/AVAILV(I)/NV(I)*MAXA
        PRINT 1 LINE THUS
IF DETAILED SYSTEM DYNAMICS ARE DESIRED, INPUT INTEGER 1.  OTHERWISE, 0.
        READ IPRINT
        CALL BUF.CAP (NV(*),RATEV(*),MTBFV(*),MTTRV(*),IPRINT) YIELDING REQ.CAP
        PRINT 1 LINE WITH REQ.CAP THUS
 REQUIRED BUFFER CAPACITY = ***
        STOP
  END ''MAIN
```

```
                    C R O S S - R E F E R E N C E

 NAME                    TYPE                               MODE      LINE NUMBERS OF REFERENCES

AVAILV          RECURSIVE VARIABLE   WORD   8  (1-D) DOUBLE     7     9*    22    24*    25
BUF.CAP         ROUTINE                                INTEGER   28
I               RECURSIVE VARIABLE   WORD   1         INTEGER    5    17*    18    19     20    21    22*
                                                                 25*
IPRINT          RECURSIVE VARIABLE   WORD   2         INTEGER    5    27     28
MAX.F           ROUTINE                                INTEGER   24
MAXA            RECURSIVE VARIABLE   WORD  20         DOUBLE    24    25
MTBFV           RECURSIVE VARIABLE   WORD   6  (1-D) DOUBLE     7     9*    19    22*    28
MTTRV           RECURSIVE VARIABLE   WORD   7  (1-D) DOUBLE     7     9*    21    22     28
NV              RECURSIVE VARIABLE   WORD   4  (1-D) INTEGER    6     8*    14    16     25    28
RATEV           RECURSIVE VARIABLE   WORD   5  (1-D) DOUBLE     7     9*    25    28
REG.CAP         RECURSIVE VARIABLE   WORD   3         INTEGER    5    28     29
```

```
ROUTINE FOR BUF.CAP (NV, RATEV, MTBFV, MTTRV, IPRINT) YIELDING REQ.CAP
''
''ROUTINE TO ESTIMATE THE REQUIREMENT FOR BUFFER CAPACITY BETWEEN
''TWO MACHINE OPERATIONS, EACH OF WHICH MAY HAVE SEVERAL IDENTICAL
''MACHINES OPERATING IN PARALLEL.  MACHINE MAINTENANCE QUEUES ARE
''DISALLOWED IN THIS MODEL.
''
DEFINE I, I1, IPRINT, J, J1, K, L, M, M1, M2, MAXL, N, N1, N2, REQ.CAP
     AS INTEGER VARIABLES
DEFINE INDEX, NV, STATE1V, AND STATE2V AS INTEGER, 1-DIMENSIONAL ARRAYS
DEFINE PNV, PDNV, PIV, AND BUFV AS REAL, 1-DIMENSIONAL ARRAYS
DEFINE RATEV, MTBFV, MTTRV, AVAILV AS REAL, 1-DIMENSIONAL ARRAYS
RESERVE NV(*),RATEV(*),MTBFV(*),MTTRV(*), AVAILV(*) AS 2
DEFINE PDN.ARRAY AS A REAL, 2-DIMENSIONAL ARRAY
RESERVE PDN.ARRAY(*,*) AS 2 BY *
     LET N1=NV(1)
     LET N2=NV(2)
     RESERVE PNV(*) AS N1
     RESERVE PDN.ARRAY(1,*) AS N1+1
     RESERVE PDN.ARRAY(2,*) AS N2+1
 ''
 ''ASSIGN PARAMETERS.
 ''
     LET DELTAT=0.5 ''(MINUTE) FOR  TIME STEP
     LET FACTOR=4.0 ''TIMES THE LARGEST MTTR
     LET STAT.CONF=1.0-EXP.F(-FACTOR)
     LET M=TRUNC.F(FACTOR*MAX.F(MTTRV(1),MTTRV(2))/DELTAT)
     RESERVE PDNV(*) AS M
     LET M1=TRUNC.F(FACTOR*MTTRV(1)/DELTAT)
     LET M2=TRUNC.F(FACTOR*MTTRV(2)/DELTAT)
     LET MAXL=(NV(1)+1)*(NV(2)+1)
     RESERVE PIV(*) AND BUFV(*) AS MAXL
     RESERVE INDEX(*), STATE1V(*), AND STATE2V(*) AS MAXL
     FOR L=1 TO MAXL, LET INDEX(L)=L
     FOR K=1 TO 2, LET AVAILV(K)=MTBFV(K)/(MTBFV(K)+MTTRV(K))
     LET A1=AVAILV(1)
     LET A2=AVAILV(2)
     LET RATE=RATEV(1)
     LET MTBF=MTBFV(1)
     LET MTTR=MTTRV(1)
 ''
 ''ECHO INPUT DATA.
 ''
     SKIP 4 LINES
     PRINT 10 LINES WITH N1,N2,RATEV(1),RATEV(2),MTBFV(1),MTBFV(2),
     MTTRV(1),MTTRV(2),100.0*A1,100.0*A2,M*DELTAT, STAT.CONF
     THUS
MACHINE INPUT DATA FOR BUFFER CAPACITY CALCULATION
               OPERATION 1      OPERATION 2
--------------------------------------------------------------
NO MACHINES _      ***            ***
MACH RATE ___     **.**          **.**        PARTS/MIN
MTBF ________    ***.**         ***.**        MINUTES
MTTR ________    ***.**         ***.**        MINUTES
AVAILABILITY      **.**          **.**        PERCENT
--------------------------------------------------------------
REPAIR TIME (MINUTES) ***.*     ASSOCIATED STAT CONFIDENCE  *.***
```

```
      SKIP 4 LINES
 FOR I1=1 TO N1+1 DO
      LET I=I1-1
 ''
 ''INITIALIZE PROBABILITY STATE VECTOR FOR 1 ST OPERATION.
 ''
      FOR K=1 TO N1, LET PNV(K)=0.0
      IF I NE 0
           LET PNV(I)=1.0
      ALWAYS
      CALL OPN.DYN GIVEN PNV(*),MTBF,MTTR,RATE,DELTAT,IPRINT,M
           YIELDING PDNV(*)
      LET PDN.ARRAY(1,I1)=PDNV(M)
 LOOP ''OVER STATES OF 1 ST OPERATION
      RELEASE PNV(*)
      RESERVE PNV(*) AS N2
      LET RATE=RATEV(2)
      LET MTBF=MTBFV(2)
      LET MTTR=MTTRV(2)
 FOR J1=1 TO N2+1 DO
      LET J=J1-1
 ''
 ''INITIALIZE PROBABILITY STATE VECTOR FOR 2 ND OPERATION.
 ''
      FOR K=1 TO N2, LET PNV(K)=0.0
      IF J NE 0
           LET PNV(J)=1.0
      ALWAYS
      CALL OPN.DYN GIVEN PNV(*),MTBF,MTTR,RATE,DELTAT,IPRINT,M
           YIELDING PDNV(*)
      LET PDN.ARRAY(2,J1)=PDNV(M)
 LOOP ''OVER STATES OF THE 2 ND OPERATION
 ''
 ''
 ''COMBINE OPERATIONAL STATES TO PRODUCE SYSTEM STATES.
 ''
      LET AVGB=0.0
      LET VARIB=0.0
 FOR I1=1 TO N1+1 DO
      LET I=I1-1
      LET PROB.STATE1=BINOM.DENS(A1,N1,I)
      FOR J1=1 TO N2+1 DO
           LET J=J1-1
           LET PROB.STATE2=BINOM.DENS(A2,N2,J)
           LET L=J1+(N2+1)*I
           LET STATE1V(L)=I
           LET STATE2V(L)=J
           LET PIV(L)=PROB.STATE1*PROB.STATE2
           LET BUFV(L)=PDN.ARRAY(1,I1)-PDN.ARRAY(2,J1)
           ADD PIV(L)*BUFV(L) TO AVGB
           ADD PIV(L)*BUFV(L)**2 TO VARIB
      LOOP ''OVER (J) 2 ND OPERATION STATES
 LOOP ''OVER (I) 1 ST OPERATION STATES
 ''
 ''RANK ORDER THE VALUES IN BUFV(*).
 ''
      CALL RANK.ORDER GIVEN BUFV(*) AND INDEX(*)
```

```
          LET SDB=SQRT.F(VARIB-AVGB**2)
          LET REQ.CAP=TRUNC.F(BUFV(MAXL)-BUFV(1)+4.0*SDB+0.5)
      ''
      ''PRINT HEADINGS.
      ''
          SKIP 2 LINES
          PRINT 7 LINES WITH M*DELTAT THUS
   EXPECTED BUFFER REQUIREMENTS ORDERED OVER ALL SYSTEM STATES
   BASED ON A REPAIR TIME LAG OF   ***.* MINUTES

-------------------------------------------------------------------------
ORDER   BUFFER   DIFFER   PROB     CUML     OPN 1     OPN 2
INDEX   SPACES   SPACES   DENS     PROB     STATE     STATE
-------------------------------------------------------------------------
          LET CDF=0.0
      FOR L=1 TO MAXL DO
          LET PROB=PIV(INDEX(L))
          ADD PROB TO CDF
          LET I=STATE1V(INDEX(L))
          LET J=STATE2V(INDEX(L))
          LET BUF.DIFF=BUFV(L)-BUFV(1)
          PRINT 1 LINE WITH L, BUFV(L), BUF.DIFF, PROB, CDF, I, J THUS
 ***     ***.*    ***.*    *.****   *.****    ***       ***
      LOOP ''OVER (L) SYSTEM STATES
          PRINT 2 LINES THUS
-------------------------------------------------------------------------

          PRINT 3 LINES WITH AVGB AND SDB THUS
    AVERAGE BUFFER CHANGE _____ ***.**
    STD DEV BUFFER CHANGE _____ ***.**

          RELEASE STATE1V(*)
          RELEASE STATE2V(*)
          RELEASE PDNV(*)
          RELEASE AVAILV(*)
          RELEASE PNV(*)
          RELEASE BUFV(*)
          RELEASE PIV(*)
          RELEASE INDEX(*)
          RELEASE.PDN.ARRAY(*,*)
          RETURN
      END ''MAIN FOR BUF.CAP
```

C R O S S - R E F E R E N C E

```
NAME         TYPE                                  MODE     LINE NUMBERS OF REFERENCES

A1           RECURSIVE VARIABLE  WORD  29          DOUBLE    36   45   88
A2           RECURSIVE VARIABLE  WORD  31          DOUBLE    37   45   91
AVAILV       RECURSIVE VARIABLE  WORD  21   (1-D)  DOUBLE    12   13*  35   36   37   126*
AVGB         RECURSIVE VARIABLE  WORD  52          DOUBLE    84   97*  105  122
BINOM.DENS   ROUTINE                               DOUBLE    88   91
BUF.CAP      ROUTINE                               INTEGER    1
BUF.DIFF     RECURSIVE VARIABLE  WORD  67          DOUBLE   118  119
BUFV         RECURSIVE VARIABLE  WORD  20   (1-D)  DOUBLE    11   32*  96   97   98   104  106*
                                                            118* 119  128*
CDF          RECURSIVE VARIABLE  WORD  63          DOUBLE   112  115* 119
DELTAT       RECURSIVE VARIABLE  WORD  23          DOUBLE    24   27   29   30   45   58   76
                                                            111
EXP.F        ROUTINE                               DOUBLE    26
FACTOR       RECURSIVE VARIABLE  WORD  25          DOUBLE    25   26   27   29   30
I            RECURSIVE VARIABLE  WORD   1          INTEGER    8   50   55   56   87   88   92
                                                             93  116  119
I1           RECURSIVE VARIABLE  WORD   2          INTEGER    8   49*  50   60   86*  87   96
INDEX        RECURSIVE VARIABLE  WORD  14   (1-D)  INTEGER   10   33*  34  104  114  116  117
                                                            130*
IPRINT       ARGUMENT            NO.    5          INTEGER    1    8   58   76
J            RECURSIVE VARIABLE  WORD   3          INTEGER    8   68   73   74   90   91   94
                                                            117  119
J1           RECURSIVE VARIABLE  WORD   4          INTEGER    8   67*  68   78   89*  90   92
                                                             96
K            RECURSIVE VARIABLE  WORD   5          INTEGER    8   35*  54*  72*
L            RECURSIVE VARIABLE  WORD   6          INTEGER    8   34*  92   93   94   95   96
                                                             97*  98* 113* 114  116  117  118
                                                            119*
M            RECURSIVE VARIABLE  WORD   7          INTEGER    8   27   28   45   58   60   76
                                                             78  111
M1           RECURSIVE VARIABLE  WORD   8          INTEGER    8   29
M2           RECURSIVE VARIABLE  WORD   9          INTEGER    8   30
MAX.F        ROUTINE                               INTEGER   27
MAXL         RECURSIVE VARIABLE  WORD  10          INTEGER    8   31   32*  33*  34  106  113
MTBF         RECURSIVE VARIABLE  WORD  35          DOUBLE    39   58   65   76
MTBFV        ARGUMENT            NO.    3   (1-D)  DOUBLE     1   12   13*  35*  39   45*  65
MTTR         RECURSIVE VARIABLE  WORD  37          DOUBLE    40   58   66   76
MTTRV        ARGUMENT            NO.    4   (1-D)  DOUBLE     1   12   13*  27*  29   30   35
                                                             40   45*  66
N            RECURSIVE VARIABLE  WORD  11          INTEGER    8
N1           PECUPSIVE VARIABLE  WORD  12          INTEGER    8   16   18   19   45   49   54
                                                             86   88
N2           RECURSIVE VARIABLE  WORD  13          INTEGER    8   17   20   45   63   67   72
                                                             89   91   92
NV           ARGUMENT            NO.    1   (1-D)  INTEGER    1   10   13*  16   17   31*
OPN.DYN      ROUTINE                               INTEGER   58   76
PDN.ARRAY    RECURSIVE VARIABLE  WORD  22   (2-D)  DOUBLE    14   15*  19*  20*  60   78   96*
                                                            131*
PDNV         RECURSIVE VARIABLE  WORD  18   (1-D)  DOUBLE    11   28*  59   60   77   78  125*
PIV          RECURSIVE VARIABLE  WORD  19   (1-D)  DOUBLE    11   32*  95   97   98  114  129*
PNV          FECURSIVE VARIABLE  WORD  17   (1-D)  DOUBLE    11   18*  54   56   58   62*  63*
                                                             72   74   76  127*
PROB         RECURSIVE VARIABLE  WORD  65          DOUBLE   114  115  119
```

```
PROB.STATE1        RECURSIVE VARIABLE   WORD  56        DOUBLE     88    95
PROB.STATE2        RECURSIVE VARIABLE   WORD  58        DOUBLE     91    95
RANK.ORDER         ROUTINE                              INTEGER   104
RATE               RECURSIVE VARIABLE   WORD  33        DOUBLE     38    58    64    76
RATEV              ARGUMENT             NO.    2  (1-D) DOUBLE      1    12    13*   38    45*   64
REQ.CAP            ARGUMENT             NO.    6        INTEGER     1     8   106
SDB                RECURSIVE VARIABLE   WORD  60        DOUBLE    105   106   122
SQRT.F             ROUTINE                              DOUBLE    105
STAT.CONF          RECURSIVE VARIABLE   WORD  27        DOUBLE     26    45
STATE1V            RECURSIVE VARIABLE   WORD  15  (1-D) INTEGER    10    33*   93   116   123*
STATE2V            RECURSIVE VARIABLE   WORD  16  (1-D) INTEGER    10    33*   94   117   124*
TRUNC.F            ROUTINE                              INTEGER    27    29    30   106
UIB.W              IMPLIED SUBSCRIPT    SYS    4        INTEGER    44    48   110
VARIB              RECURSIVE VARIABLE   WORD  54        DOUBLE     85    98*  105
```

```
ROUTINE FOR OPN.DYN GIVEN PNV, MTBF, MTTR, RATE, DELTAT, IPRINT, M
YIELDING PONV
''
''PROGRAM SOLVES THE SET OF DIFFERENTIAL EQUATIONS WHICH CHARACTERIZE THE
''PRODUCTION OF A MANUFACTURING OPERATION CONSISTING OF N IDENTICAL MACHINES
''OPERATING IN PARALLEL.  THE INITIAL STATE OF THIS SYSTEM IS REPRESENTED
''BY THE VALUES OF THE ELEMENTS OF THE N-DIMENSIONAL PROBABILITY VECTOR
''PNV(*), WHERE THE K TH ELEMENT IS THE PROBABILITY THAT K MACHINES ARE
''OPERATING.  THE MEAN TIME BETWEEN FAILURES OF MACHINES OF THIS TYPE IS
''ENTERED AS MTBF.  THE MEAN TIME TO REPAIR A MACHINE IS ENTERED AS MTTR.
''THE OPERATING PRODUCTION RATE OF EACH MACHINE IS ENTERED AS RATE.
''GENERALLY, THE TIME UNITS FOUND CONVENIENT ARE MINUTES.  TO ACCOMODATE
''A VARIETY OF SYSTEMS, THE PROGRAM USER MUST SUPPLY THE INTEGRATION TIME
''STEP IN COMPATIBLE TIME UNITS.  IF A PRINTOUT OF THE TRAJECTORY OF THE
''STATE VECTOR IS OESIREO, THE INTEGER SWITCH IS SET TO 1.  THE PROGRAM
''RETURNS M VALUES OF THE EXPECTED CUMULATIVE PRODUCTION, IN PDNV(*), FROM
''THIS SYSTEM WHEN IT OPERATES IN AN UNCONSTRAINED MANNER.
''
''INPUT:
''PNV ______________ VECTOR OF DIMENSION N, REPRESENTING THE PROBABILITY
''                   OF FINOING THE SYSTEM WITH K MACHINES OPERATING,0<K<=N.
''                   ON INPUT, PNV(*) IS THE INITIAL STATE.  THIS VECTOR IS
''                   USED LOCALLY AS THE SYSTEM STATE VECTOR AT TIME I*H.
''MTBF ____________ THE MEAN TIME BETWEEN FAILURES (MINUTES) DURING OPERATION
''                   FOR MACHINES OF THIS TYPE.
''MTTR ____________ THE MEAN TIME (MINUTES) TO REPAIR A MACHINE FOR ALL CLASSES
''                   OF FAILURES.
''RATE ____________ THE OPERATING PROOUCTION RATE PER MACHINE (PARTS/MIN).
''DELTAT __________ TIME STEP USEO IN FILLING THE PRODUCTION ARRAY.
''IPRINT __________ AN INTEGER SWITCH TO CHOOSE PRINTING OF THE TRAJECTORY
''                   OF THE STATE VECTOR (= 1 TO PRINT).
''M ________________ THE NO. OF DISCRETE TIME STEPS TAKEN ALONG THE TRAJECTORY.
''
''INTERNALLY:
''H ________________ THE INTEGRATION TIME STEP (MINUTES) = 0.2*DELTAT.
''OUTPUT:
''PDNV(*) _________ THE VECTOR OF AVG CUMULATIVE PRODUCTION OF THIS OPERATION
''                   WHEN STARTEO IN THE SPECIFIED STATE.
''
DEFINE I, IPRINT, J, K, L, M, N AS INTEGER VARIABLES
DEFINE PNV AND PONV AS REAL, 1-DIMENSIONAL ARRAYS
DEFINE PNDOT AS A REAL, 1-DIMENSIONAL ARRAY ''LOCALLY
OEFINE AM AS A REAL, 2-OIMENSIONAL ARRAY ''LOCALLY
RESERVE PDNV(*) AS M ''TIME STEPS
LET N=DIM.F(PNV(*))
RESERVE PNDOT(*) AS N ''LOCALLY
RESERVE AM(*,*) AS N BY N ''LOCALLY
''
''CALCULATE THE BIRTH- (LAMBDA) ANO OEATH- (MU) RATE PARAMETERS.
''
    LET LAMBDA=1.0/MTTR
    LET MU=1.0/MTBF
''
''CHECK IF TIME STEP IS REASONABLE.
''
    IF DELTAT GE 0.1*MIN.F(MTTR,MTBF)
        PRINT 3 LINES WITH OELTAT, MTBF, MTTR THUS
```

```
TROUBLE IN OPN.DYN.   TIME STEP = **.**** TOO LARGE.
MTBF = *****.***
MTTR = *****.***
             STOP
         OTHERWISE
         LET H=0.2*DELTAT ''INTEGRATION STEP FOR DIFFERENTIAL EQUATIONS
         IF N > 1
             GO TO L1
         OTHERWISE ''GET ANALYTIC SOLUTION FOR N=1
         LET FFREQ=LAMBDA+MU
         LET P10=PNV(1)
         LET P00=1.0-P10
         LET POINF=MU/FFREQ
         IF IPRINT=1
             SKIP 2 LINES
             PRINT 6 LINES THUS
  TRAJECTORY OF PRODUCTION OPERATION STATE PROBABILITIES

--------------------------------------------------------------------------------
TIME       CUM           MACHINES OPERATING          AVG        S.D.
(MIN)      PDN           0            1              RATE       RATE
--------------------------------------------------------------------------------
         ALWAYS
     FOR I=1 TO M DO
         LET TIME=I*DELTAT
         LET E1=EXP.F(-FFREQ*TIME)
         LET C=LAMBDA*TIME/FFREQ+(LAMBDA/FFREQ**2-P10/FFREQ)*(E1-1.0)
         LET PDNV(I)=RATE*C
         IF IPRINT=1
             IF MOD.F(I,5)=0
                 LET P0=P00*E1+POINF*(1.0-E1)
                 LET P1=1.0-P0
                 LET AVG.RATE=RATE*P1
                 LET SD.RATE=RATE*SQRT.F(P1-P1**2)
                 PRINT 1 LINE WITH TIME, PDNV(I), P0, P1, AVG.RATE, SD.RATE
                 THUS
***.**     ***.**     *.*****     *.*****          **.***     **.***
             ALWAYS
         ALWAYS
     LOOP ''OVER I
         GO TO L3
     'L1'IF N > 2
             GO TO L2
         OTHERWISE ''GET ANALYTIC SOLUTION FOR N=2
         LET FFREQ=LAMBDA+MU
         LET P10=PNV(1)
         LET P20=PNV(2)
         LET P00=1.0-P10-P20
         LET A0=2.0*LAMBDA*P00+2.0*FFREQ*P10+2.0*MU*P20
         LET A1=P10
         LET P1INF=2.0*LAMBDA*MU/FFREQ**2
         LET P2INF=LAMBDA**2/FFREQ**2
         LET A=(A0-A1*FFREQ)/FFREQ-4.0*LAMBDA*MU/FFREQ**2
         LET B=-(A0-2.0*A1*FFREQ)/FFREQ+2.0*LAMBDA*MU/FFREQ**2
         IF IPRINT=1
             SKIP 2 LINES
             PRINT 6 LINES THUS
```

```
    TRAJECTORY OF PRODUCTION OPERATION STATE PROBABILITIES

TIME      CUM        MACHINES OPERATING              AVG      S.D.
(MIN)     PDN        0        1        2             RATE     RATE
---------------------------------------------------------------------
          ALWAYS
          LET CUM=0.0
          LET AVGP=P10+2.0*P20
     FOR I=1 TO M DO
          LET TIME=I*DELTAT
          LET E1=EXP.F(-FFREQ*TIME)
          LET E2=EXP.F(-2.0*FFREQ*TIME)
          LET E3=EXP.F(-2.0*MU*TIME)
          LET P1=P1INF+A*E1+B*E2
          LET P2=P20*E3+P2INF*(1.0-E3)+A*LAMBDA/(LAMBDA-MU)*(E3-E1)+0.5*B*(E3-E2)
          LET P0=1.0-P1-P2
          LET AVG=P1+2.0*P2
          LET ER2=P1+4.0*P2
          LET SD.RATE=RATE*SQRT.F(ER2-AVG**2)
          LET AVG.RATE=RATE*AVG
          ADD 0.5*DELTAT*(AVG+AVGP) TO CUM
          LET AVGP=AVG
          LET PDNV(I)=RATE*CUM
          IF IPRINT=1
               IF MOD.F(I,5)=0
                    PRINT 1 LINE WITH TIME, PDNV(I), P0, P1, P2, AVG.RATE, SD.RATE
                    THUS
***.**    ***.**     *.****   *.****   *.****        **.***   **.***
               ALWAYS
          ALWAYS
     LOOP ''OVER I
          GO TO L3
     ''
     ''CALCULATE THE ELEMENTS OF THE STATE TRANSITION MATRIX (AM(*,*)).
     ''
     'L2'FOR I=1 TO N DO
          FOR J=1 TO N DO
               LET AM(I,J)=0.0
          LOOP ''OVER J
     LOOP ''OVER I
          LET CON=N*LAMBDA
          LET AM(1,1)=-(CON+(N-1)*LAMBDA+MU)
          LET AM(1,2)=2.0*MU-CON
          IF N > 2
               FOR K=3 TO N DO
                    LET AM(1,K)=-CON
               LOOP ''OVER K
          ALWAYS
          IF N > 2
               FOR K=2 TO N-1 DO
                    LET AM(K,K-1)=(N-K+1)*LAMBDA
                    LET AM(K,K)=-((N-K)*LAMBDA+K*MU)
                    LET AM(K,K+1)=(K+1)*MU
               LOOP ''OVER K
          ALWAYS ''CONTINUE TO STATE N
          LET AM(N,N-1)=LAMBDA
```

```
    LET AM(N,N)=-N*MU
''
''END OF STATE TRANSITION MATRIX.  GET INITIAL RATE VECTOR.
''
    CALL MAT.VEC.MPY(AM(*,*),PNV(*),N) YIELDING PNDOT(*)
    ADD CON TO PNDOT(1)
''
''PRINT HEADINGS FOR OUTPUT OF STATE VECTOR.
''
    IF IPRINT=1
        LET P1=0.0
        LET P2=0.0
        LET P3=0.0
        LET P4=0.0
        LET P5=0.0
        LET P6=0.0
        LET P7=0.0
        LET P8=0.0
        LET P9=0.0 ''FOR PRINTING IN ROWS
        SKIP 2 LINES
        PRINT 7 LINES WITH N, PNV(N) THUS
 TRAJECTORY OF PRODUCTION OPERATION STATE PROBABILITIES

INITIAL MAX STATE PROB(**) = *.****
-------------------------------------------------------------------------
TIME     CUM             NUMBER OF MACHINES IN OPERATION
(MIN)    PDN    1      2      3      4      5      6      7      8      9
-------------------------------------------------------------------------
    ALWAYS
''
''CREATE THE EXPECTED CUMULATIVE PRODUCTION VECTOR.
''
    LET CUM=0.0 ''CUM NORMALIZED PDN FROM PREVIOUS STEP
    LET AVGP=0.0
    FOR K=1 TO N DO
        ADD K*PNV(K) TO AVGP
    LOOP ''TO GET INITIAL AVERAGE NORMALIZED PRODUCTION RATE
FOR I=1 TO M ''TIME STEPS'' DO
    LET TM=I*DELTAT
    FOR L=1 TO 5 DO
        CALL MAT.VEC.MPY (AM(*,*), PNV(*), N) YIELDING PNDOT(*)
        ADD CON TO PNDOT(1)
        FOR K=1 TO N DO
            ADD H*PNDOT(K) TO PNV(K)
        LOOP ''OVER (K) COMPONENTS
    LOOP ''OVER (L) INTEGRATION SUBSTEPS
        LET AVG=0.0
    FOR K=1 TO N DO
        ADD K*PNV(K) TO AVG
    LOOP ''OVER COMPONENTS
    ADD 0.5*DELTAT*(AVG+AVGP) TO CUM
    LET PDNV(I)=RATE*CUM
    LET AVGP=AVG
    IF IPRINT=1
        IF MOD.F(I,5)=0
            LET P1=PNV(1)
            LET P2=PNV(2)
```

```
               IF N=2
                   GO TO L0
               OTHERWISE
               LET P3=PNV(3)
               IF N=3
                   GO TO L0
               OTHERWISE
               LET P4=PNV(4)
               IF N=4
                   GO TO L0
               OTHERWISE
               LET P5=PNV(5)
               IF N=5
                   GO TO L0
               OTHERWISE
               LET P6=PNV(6)
               IF N=6
                   GO TO L0
               OTHERWISE
               LET P7=PNV(7)
               IF N=7
                   GO TO L0
               OTHERWISE
               LET P8=PNV(8)
               IF N=8
                   GO TO L0
               OTHERWISE
               LET P9=PNV(9)
    'L0'       PRINT 1 LINE WITH TM,PDNV(I),P1,P2,P3,P4,P5,P6,P7,P8,P9
               THUS
***.**  ***.**  *.**** *.**** *.**** *.**** *.**** *.**** *.**** *.**** *.****
           ALWAYS
       ALWAYS
    ''
    ''END OF I TH TIME STEP OF SIZE DELTAT.
    ''
    LOOP ''OVER I
    'L3'IF IPRINT=1
           PRINT 2 LINES THUS
--------------------------------------------------------------------------------
       ALWAYS
    ''
    ''RELEASE ARRAYS PRIOR TO RETURNING.
    ''
       RELEASE PNDOT(*)
       RELEASE AM(*,*)
       RETURN
    END ''OPN.DYN
```

```
                          C R O S S - R E F E R E N C E

 NAME                       TYPE                                MODE      LINE NUMBERS OF REFERENCES

A               RECURSIVE VARIABLE   WORD  5B          DOUBLE    100   113   114
A0              RECURSIVE VARIABLE   WORD  50          DOUBLE     96   100   101
A1              RECURSIVE VARIABLE   WORD  52          DOUBLE     97   100   101
AM              RECURSIVE VARIABLE   WORD   7   (2-D)  DOUBLE     43    47*  136   140   141   144   I49
                                                                 150   151   154   155   159   1B8   248*
AVG             RECURSIVE VARIABLE   WORD  73          DOUBLE    116   118   119   120   121   194   196*
                                                                 198   200
AVG.RATE        RECURSIVE VARIABLE   WORD  44          DOUBLE     81    B3   119   125
AVGP            RECURSIVE VARIABLE   WORD  65          DOUBLE    107   120   I21   1B1   1B3*  198   200
B               RECURSIVE VARIABLE   WORD  60          DOUBLE    101   113   114
C               RECURSIVE VARIABLE   WORD  3B          DOUBLE     75    76
CON             RECURSIVE VARIABLE   WORD  77          DOUBLE    139   140   141   I44   160   189
CUM             RECURSIVE VARIABLE   WORD  63          DOUBLE    106   120*  I22   1B0   19B*  199
DELTAT          ARGUMENT             NO.    5          DOUBLE      1    56    57    60    73   109   120
                                                                 1B6   19B
DIM.F           ROUTINE                                INTEGER    45
E1              RECURSIVE VARIABLE   WORD  36          DOUBLE     74    75    79*  I10   113   114
E2              RECURSIVE VARIABLE   WORD  67          DOUBLE    111   113   114
E3              RECURSIVE VARIABLE   WORD  69          DOUBLE    112   114*
ER2             RECURSIVE VARIABLE   WORD  75          DOUBLE    117   11B
EXP.F           ROUTINE                                DOUBLE     74   110   111   112
FFREQ           RECURSIVE VARIABLE   WORD  25          DOUBLE     64    67    74    75*   92    96    98
                                                                  99   100*  101*  110   111
H               RECURSIVE VARIABLE   WORD  23          DOUBLE     60   191
I               RECURSIVE VARIABLE   WORD   I          INTEGER    40    72*   73    76    7B    B3   I0B*
                                                                 109   122   124   125   134*  136   185*
                                                                 1B6   199   202   233
IPRINT          ARGUMENT             NO.    6          INTEGER     1    40    6B    77   I02   123   164
                                                                 201   241
J               RECURSIVE VARIABLE   WORD   2          INTEGER    40   135*  136
K               RECURSIVE VARIABLE   WORD   3          INTEGER    40   143*  144   14B*  149*  150*  151*
                                                                 1B2*  183*  190*  191*  195*  196*
L               RECURSIVE VARIABLE   WORD   4          INTEGER    40   187*
                                                                 206   210   214   218   222   226   230
L0              LABEL                                            233
LI              LABEL                                             62    B9
L2              LABEL                                             90   134
L3              LABEL                                             B8   130   241
LAMBDA          RECURSIVE VARIABLE   WORD   8          DOUBLE     51    64    75*   92    96    98    99
                                                                 100   101   114*  139   140   149   150
                                                                 154
M               ARGUMENT             NO.    7          INTEGER     1    40    44    72   10B   1B5
MAT.VEC.MPY     ROUTINE                                INTEGER   159   188
MIN.F           ROUTINE                                INTEGER    56
MOD.F           ROUTINE                                INTEGER    7B   124   202
MTBF            ARGUMENT             NO.    2          DOUBLE      1    52    56    57
MTTR            ARGUMENT             NO.    3          DOUBLE      1    5I    56    57
MU              RECURSIVE VARIABLE   WORD  10          DOUBLE     52    64    67    92    96    9B   100
                                                                 101   112   114   I40   141   150   151
                                                                 155
N               RECURSIVE VARIABLE   WORD   5          INTEGER    40    45    46    47*   61    89   134
                                                                 135   139   140   142   143   147   14B
                                                                 149   150   154*  155*  159   175*  1B2
```

```
ROUTINE OPN.DYN                         CACI SIMSCRIPT II.5 for PRIME Systems, Release 2.1          Page   15
Options = SEQUENCE,ID,SUBCHK,XREF,NOEXPLIST,TRACE3                       27 JUN 1983 12:05:18
                                                                       188    190    195    205    209    213    217
                                                                       221    225    229
OPN.DYN                    ROUTINE                             INTEGER   1
P0                         RECURSIVE VARIABLE   WORD  40       DOUBLE   79     80     83    115    125
P00                        RECURSIVE VARIABLE   WORD  29       DOUBLE   66     79     95     96
P0INF                      RECURSIVE VARIABLE   WORD  31       DOUBLE   67     79
P1                         RECURSIVE VARIABLE   WORD  42       DOUBLE   80     81     82*    83    113    115    116
                                                                       117    125    165    203    233
P10                        RECURSIVE VARIABLE   WORD  27       DOUBLE   65     66     75     93     95     96     97
                                                                       107
P1INF                      RECURSIVE VARIABLE   WORD  54       DOUBLE   98    113
P2                         RECURSIVE VARIABLE   WORD  71       DOUBLE  114    115    116    117    125    166    204
                                                                       233
P20                        RECURSIVE VARIABLE   WORD  48       DOUBLE   94     95     96    107    114
P2INF                      RECURSIVE VARIABLE   WORD  56       DOUBLE   99    114
P3                         RECURSIVE VARIABLE   WORD  79       DOUBLE  167    208    233
P4                         RECURSIVE VARIABLE   WORD  81       DOUBLE  168    212    233
P5                         RECURSIVE VARIABLE   WORD  83       DOUBLE  169    216    233
P6                         RECURSIVE VARIABLE   WORD  85       DOUBLE  170    220    233
P7                         RECURSIVE VARIABLE   WORD  87       DOUBLE  171    224    233
P8                         RECURSIVE VARIABLE   WORD  89       DOUBLE  172    228    233
P9                         RECURSIVE VARIABLE   WORD  91       DOUBLE  173    232    233
PDNV                       ARGUMENT             NO.    8 (1-D) DOUBLE    2     41     44*    76     83    122    125
                                                                       199    233
PNDOT                      RECURSIVE VARIABLE   WORD   6 (1-D) DOUBLE   42     46*   159    160*   188    189*   191
                                                                       247*
PNV                        ARGUMENT             NO.    1 (1-D) DOUBLE    1     41     45     65     93     94    159
                                                                       175    183    188    191*   196    203    204
                                                                       208    212    216    220    224    228    232
RATE                       ARGUMENT             NO.    4       DOUBLE    1     76     81     82    118    119    122
                                                                       199
SD.RATE                    RECURSIVE VARIABLE   WORD  46       DOUBLE   82     83    118    125
SORT.F                     ROUTINE                             DOUBLE   82    118
TIME                       RECURSIVE VARIABLE   WORD  34       DOUBLE   73     74     75     83    109    110    111
                                                                       112    125
TM                         RECURSIVE VARIABLE   WORD  94       DOUBLE  186    233
UIB.W                      IMPLIED SUBSCRIPT    SYS    4       INTEGER  69    103    174
```

Options = SEQUENCE,ID,SUBCHK,XREF,NOEXPLIST,TRACE3

```
 1  ROUTINE FOR MAT.VEC.MPY (AM, BV, NELMTS) YIELDING CV
 2  ''
 3  ''ROUTINE TO MULTIPLY THE SQUARE MATRIX AM , OF NELMTS BY NELMTS,
 4  ''BY THE VECTOR BV (NELMTS BY 1), YIELDING THE VECTOR CV (NELMTS BY 1).
 5  ''
 6  DEFINE I, J, K, NELMTS AS INTEGER VARIABLES
 7  DEFINE BV AND CV AS REAL, 1-DIMENSIONAL ARRAYS
 8  DEFINE AM AS A REAL, 2-DIMENSIONAL ARRAY
 9  RESERVE AM(*,*) AS NELMTS BY NELMTS
10  RESERVE BV(*) AS NELMTS
11  RESERVE CV(*) AS NELMTS
12  FOR I=1 TO NELMTS DO
13      LET CV(I)=0.0
14      FOR K=1 TO NELMTS DO
15          ADD AM(I,K)*BV(K) TO CV(I)
16      LOOP ''OVER K
17  LOOP ''OVER I
18  RETURN
19  END ''OF ROUTINE MAT.VEC.MPY
```

C R O S S - R E F E R E N C E

| NAME | TYPE | | | | MODE | LINE NUMBERS OF REFERENCES | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| AM | ARGUMENT | NO. | 1 | (2-D) | DOUBLE | 1 | 8 | 9* | 15 | | | |
| BV | ARGUMENT | NO. | 2 | (1-D) | DOUBLE | 1 | 7 | 10* | 15 | | | |
| CV | ARGUMENT | NO. | 4 | (1-D) | DOUBLE | 1 | 7 | 11* | 13 | 15* | | |
| I | RECURSIVE VARIABLE | WORD | 1 | | INTEGER | 6 | 12* | 13 | 15* | | | |
| J | RECURSIVE VARIABLE | WORD | 2 | | INTEGER | 6 | | | | | | |
| K | RECURSIVE VARIABLE | WORD | 3 | | INTEGER | 6 | 14* | 15* | | | | |
| MAT.VEC.MPY | ROUTINE | | | | INTEGER | 1 | | | | | | |
| NELMTS | ARGUMENT | NO. | 3 | | INTEGER | 1 | 6 | 9* | 10 | 11 | 12 | 14 |

```
 1  FUNCTION FOR BINOM.DENS (P, N, K)
 2  ''
 3  ''FUNCTION CALCULATES THE PROBABILITY DENSITY OF THE BINOMIAL DISTRIBUTION
 4  ''WITH (AVG) PROBABILITY PARAMETER P, WITH SAMPLE SIZE PARAMETER N, AND
 5  ''WITH INTEGER ARGUMENT K.
 6  ''
 7      DEFINE I, K, N AS INTEGER VARIABLES
 8      LET Q=1.0-P
 9      LET BPF=Q**N
10      IF K LE 0
11          RETURN WITH BPF
12      OTHERWISE
13  FOR I=1 TO K DO
14      LET BPF=P*(N-I+1)/(I*Q)*BPF
15  LOOP ''OVER I
16      RETURN WITH BPF
17  END ''BINOM.DENS
```

C R O S S - R E F E R E N C E

| NAME | TYPE | | | MODE | LINE NUMBERS OF REFERENCES | | | |
|---|---|---|---|---|---|---|---|---|
| BINOM.DENS | ROUTINE | | | DOUBLE | 1 | | | |
| BPF | RECURSIVE VARIABLE | WORD | 4 | DOUBLE | 9 | 11 | 14* | 16 |
| I | RECURSIVE VARIABLE | WORD | 1 | INTEGER | 7 | 13* | 14* | |
| K | ARGUMENT | NO. | 3 | INTEGER | 1 | 7 | 10 | 13 |
| N | ARGUMENT | NO. | 2 | INTEGER | 1 | 7 | 9 | 14 |
| P | ARGUMENT | NO. | 1 | DOUBLE | 1 | 8 | 14 | |
| Q | RECURSIVE VARIABLE | WORD | 2 | DOUBLE | 8 | 9 | 14 | |

```
 1  FUNCTION FOR BINOM.DIST (P, N, K)
 2  ''
 3  ''FUNCTION PRODUCES THE CUMULATIVE BINOMIAL DISTRIBUTION WITH PROBABILITY
 4  ''PARAMETER P, SAMPLE-SIZE PARAMETER N, AND INTEGER ARGUMENT K.
 5  ''
 6  DEFINE I, K, N AS INTEGER VARIABLES
 7       LET Q=1.0-P
 8       LET FX=Q**N
 9       LET CDF=FX
10       IF K LE 0
11            RETURN WITH FX
12       OTHERWISE
13  FOR I=1 TO K DO
14       LET FX=P*(N-I+1)/(I*Q)*FX
15       ADD FX TO CDF
16  LOOP ''OVER I
17       RETURN WITH CDF
18  END ''BINOM.DIST
```

C R O S S - R E F E R E N C E

| NAME | TYPE | | | MODE | LINE NUMBERS OF REFERENCES | | | | |
|---|---|---|---|---|---|---|---|---|---|
| BINOM.DIST | ROUTINE | | | DOUBLE | 1 | | | | |
| CDF | RECURSIVE VARIABLE | WORD | 6 | DOUBLE | 9 | 15* | 17 | | |
| FX | RECURSIVE VARIABLE | WORD | 4 | DOUBLE | 8 | 9 | 11 | 14* | 15 |
| I | RECURSIVE VARIABLE | WORD | 1 | INTEGER | 6 | 13* | 14* | | |
| K | ARGUMENT | NO. | 3 | INTEGER | 1 | 6 | 10 | 13 | |
| N | ARGUMENT | NO. | 2 | INTEGER | 1 | 6 | 8 | 14 | |
| P | ARGUMENT | NO. | 1 | DOUBLE | 1 | 7 | 14 | | |
| Q | RECURSIVE VARIABLE | WORD | 2 | DOUBLE | 7 | 8 | 14 | | |

```
 1  ROUTINE TO RANK.ORDER GIVEN XV AND INDEX
 2  ''
 3  ''ROUTINE ACCEPTS THE VALUES IN THE VECTOR XV, OF DIMENSION N,
 4  ''AND RETURNS THE VALUES ORDERED IN ASCENDING ORDER.
 5  ''
 6  DEFINE I, II, J, AND N AS INTEGER VARIABLES
 7  DEFINE INDEX AS AN INTEGER, 1-DIMENSIONAL ARRAY
 8  DEFINE XV AS A 1-DIMENSIONAL ARRAY
 9  LET N=DIM.F(XV(*))
10  FOR I=1 TO N-1 DO
11      FOR II=I+1 TO N DO
12          IF XV(I) LE XV(II)
13              GO TO S
14          OTHERWISE ''SWAP VALUES
15          LET HOLD=XV(I)
16          LET XV(I)=XV(II)
17          LET XV(II)=HOLD
18          LET J=INDEX(I)
19          LET INDEX(I)=INDEX(II)
20          LET INDEX(II)=J
21  'S'LOOP ''OVER II
22  LOOP ''OVER I
23  END ''OF ROUTINE RANK.ORDER
```

C R O S S - R E F E R E N C E

| NAME | TYPE | | | | MODE | LINE NUMBERS OF REFERENCES | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DIM.F | ROUTINE | | | | INTEGER | 9 | | | | | | |
| HOLD | RECURSIVE VARIABLE | WORD | 5 | | DOUBLE | 15 | 17 | | | | | |
| I | RECURSIVE VARIABLE | WORD | 1 | | INTEGER | 6 | 10* | 11 | 12 | 15 | 16 | 18 |
| | | | | | | 19 | | | | | | |
| II | RECURSIVE VARIABLE | WORD | 2 | | INTEGER | 6 | 11* | 12 | 16 | 17 | 19 | 20 |
| INDEX | ARGUMENT | NO. | 2 | (1-D) | INTEGER | 1 | 7 | 18 | 19* | 20 | | |
| J | RECURSIVE VARIABLE | WORD | 3 | | INTEGER | 6 | 18 | 20 | | | | |
| N | RECURSIVE VARIABLE | WORD | 4 | | INTEGER | 6 | 9 | 10 | 11 | | | |
| RANK.ORDER | ROUTINE | | | | INTEGER | 1 | | | | | | |
| S | LABEL | | | | | 13 | 21 | | | | | |
| XV | ARGUMENT | NO. | 1 | (1-D) | DOUBLE | 1 | 8 | 9 | 12* | 15 | 16* | 17 |