SIMULATION AND RELIABILITY ANALYSIS OF A FLEXIBLE MANUFACTURING SYSTEM WITH AGV BASED MATERIAL HANDLING

by

FUHONG DAI

BASc , Northeast University of Technology of P.R. China , 1982

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF APPLIED SCIENCE

in

THE FACULTY OF GRADUATE STUDIES
(Department of mechanical Engineering)

We accept this thesis as conforming
to the required standard

THE UNIVERSITY OF BRITISH COLUMBIA

ABSTRACT

This thesis investigates the reliability of a flexible manufacturing system with AGV based material handling. An analytical model was built using the state space approach (Markov process). Although the method is shown to be tedious when the number of system components becomes large , however , it does provide a powerful approach for reliability analysis of complex systems such as flexible manufacturing facilities. To simplify the analysis, state merging and state truncating techniques were used in carrying out the calculations.

To compare with the analytical results , a simulation model was built using SLAM II discrete event modelling and simulation software. The results were very close to the analytical ones when same failure and repair rates were assumed for the basic components of the system. Overall, it was found that the simulation method was much simpler to develop and experiment with. A SLAM II simulation model of the system performance was also built to examine the operation of the FMS as a whole with failure and repair events.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

NOMENCLATURE

A + = the availability of the system

D (i , j) = transportation time from node i to node j

$f(x)$ = probability density function of random variable x

F (x) = cumulative distribution function of random variable x

F (i , j) = required flow matrix from node i to node j

IDX = index number of a machine to be visited

JT = job type

MTV (JT, IDX) = machine # to be visited for a given job type and index number

MTTF = mean time to failure

MTTR = mean time to repair

P (t) = state probability matrix

$P(t)$ = a row vector of state probability

Pc (i) = probability that the ith pickup node calls an idle AGV

P_{DN} = probability of the system being down

$P_i(t)$ = probability of being in state i at time t

$P_{ij}(x)$ = transition probability from state i to state j during the time interval x

P_T = probability of the subsets that is truncated

PST(JT, IDX) = processing time for a given job type and index number

P_{UP} = probability of the system being up

$P_w(j)$ = probability that the AGV waiting at j node

R = transition rate matrix

R_{ij} = transition rate from state i to state j

r = eigenvalue of matrix R

S = matrix formed by the right eigenvectors of matrix R

TIMST = SLAM II statement used to request the automatic collection of time-persistent statistics on SLAM II global variable XX (I)

T_l = pick-up and delivery time for total part flow

T_p = average transportation time for loaded vehicle

TRT(i , j) = travel time from node i to node j

T_v = average transportation time for the empty vehicle

T_w = specified working time for moving parts

X + = set contains states of success

X - = set contains states of failure

XX (I) = SLAM II global variable

x_{ij} = duration of state i under the condition of transiting to state j

Z (t) = random variable of system state at time t

λ = failure rate of a component

λ_{lm} = equivalent transition rate from subset l to subset m

μ = repair rate of a component

ACKNOWLEDGMENT

The author wishes to express his sincere thanks to Dr. F. Sassani for his guidance and encouragement throughout the preparation of this thesis

1.0 INTRODUCTION

1.1 Background and Motivation

Material handling systems are vital components of an automated manufacturing system. They are used to integrate various components of a modern manufacturing system to facilitate the flow of workpieces from one location to another. Material handling systems may consist of different components such as conveyors, an automated guided vehicles (AGV) fork lifts , and robots , etc. Material handling systems play an important role in the overall performance of the integrated manufacturing systems. We define an integrated manufacturing system as a manufacturing facility consisting of a set of work stations, loading and unloading stations , and an inventory system linked by a material handling system. The performance of the integrated manufacturing system is affected by various operational and technological factors. The operational factors include the loading, the material handling, the storage, the processing operations, and also the layout of the integrated manufacturing system. The technological factors include the characteristics of the workstation, the machines, the material handling system and the inventory systems [1] [2].

We call an integrated manufacturing system under a central computer control a flexible manufacturing system (FMS). Flexible manufacturing systems are commonly used to upgrade the performance of low to medium volume manufacturing systems, and are rapidly replacing the existing classical manufacturing systems.

Hand in hand with increased automation and the complexity of manufacturing systems of the flexible sort , or FMS , reliability has become one of the vital ingredients in FMS's planning, design and operational phases [3]. Reliability is important as it reflects the ability of an FMS to keep operating schedules. Reliability (or availability) modelling of an AGV based

FMS is of significant importance in the modern context. Statistical reliability techniques are advanced and have been applied to numerous electrical, transit, and mechanical systems [4].

1.2 Markov Processes, the State Space Approach

The State Space Approach (or Markov processes) is a very useful approach for system reliability analysis. A component may assume various states depending upon its failure and restorative modes. The system states describe the states of the components and the environment in which the system is operating. The set of all the possible states of the system is called the state space or the event space. The state space approach involves the following steps [5] :

i. Enumeration of all possible system states

ii. Determination of the interstate transition rates. If a diagram is drawn showing the various states and the interstate transition rates between the states, this is called the state transition diagram.

iii. If the components are independent, the system state probabilities may be found from the products of the component state probabilities.

iv. The states are then grouped into subsets depending upon the requirements of the analysis. In most cases, measures are required only for success or failure.

After the grouping has been done, the subset probabilities etc. can be calculated. If we define a random variable $Z(t)$ (this random variable is associated with each value of time t) as the state of a system at time "t", the family of the random variables $(Z(t), t \geq 0)$ is called a stochastic process. And the values assumed by the process are called the states of the system and the set of all possible states is called the state space.

Once the state occupied at a time point is known, if the previous history of the process is not involved in determining the subsequent probability distributions, the stochastic process is said to be Markovian or called the Markov process. The Markov process is sometime called memoryless, because the probability distribution of $Z(t)$ only depends on the latest of the time points and none prior to that (Equation 1-1).

Many of the problems encountered in system reliability analysis can be modelled using continuous parameter Markov chains [6]. For $t > v > u$, the Markov property for a continuous parameter Markov chain would be

$$P(Z(t) = k \mid Z(v) = j, Z(u) = i) = P(Z(t) = k \mid Z(v) = j) \qquad (1\text{-}1)$$

This property is basically of the form

$$P(Z(t+x) = j \mid Z(t) = i) \qquad (1\text{-}2)$$

and is termed as the probability of transition from state i to state j during the time interval t to $t + x$. The transition probability will be denoted by

$$P_{ij}(x) = P(Z(t+x) = j \mid Z(t) = i) \qquad (1\text{-}3)$$

for any x.

The conditional probability density function (1-2) for the process will be given by the **Chapman-Kolmogorov** equation :

$$P_{ij}(t + x) = \sum_{k} P_{ik}(t) P_{kj}(x) \qquad (1\text{-}4)$$

This equation will be used to further deduce the transition probability of the Markov Process. The verification of equation (1-4) is in reference [7].

1.3 The SLAM II Discrete Event Modelling and Simulation Method

The SLAM II (Simulation Language for Alternative Modelling) discrete event modelling and simulation method will be used in this research project [8] [9]. To simulate a discrete event model of a system using SLAM II, the user codes each discrete event as a FORTRAN subroutine. To assist the user in this task, SLAM II provides a set of FORTRAN subprograms for performing all commonly encountered functions such as event scheduling, statistics collection, and random sample generation. The advancing of simulated time (TNOW) and the order in which the event routines are processed are controlled by the SLAM II executive program. Thus, SLAM II relieves the simulation modeler of the task of sequencing events in their proper chronological order. Each event subroutine is assigned a positive integer numeric code called the *event code,* in the same fashion as the event code defined at an EVENT node. The event code is mapped onto a call to the appropriate event subroutine by subroutine EVENT (I) where the argument I is the event code. This subroutine is written by the user and consists of a computed GO TO statement indexed on I causing a transfer to the appropriate event subroutine call followed by a return. The executive control for a discrete event simulation is provided by subroutine SLAM which is called from a user-written *main* program. The SLAM II next event logic for simulating discrete event models is depicted in Figure 1-1. The SLAM II method will be employed in modelling and simulating a representative FMS in two aspects:

(1) Evaluation of the performance of the system.

(2) Collection of the statistics on reliability measures and the state probabilities (or the state availabilities) of the system.

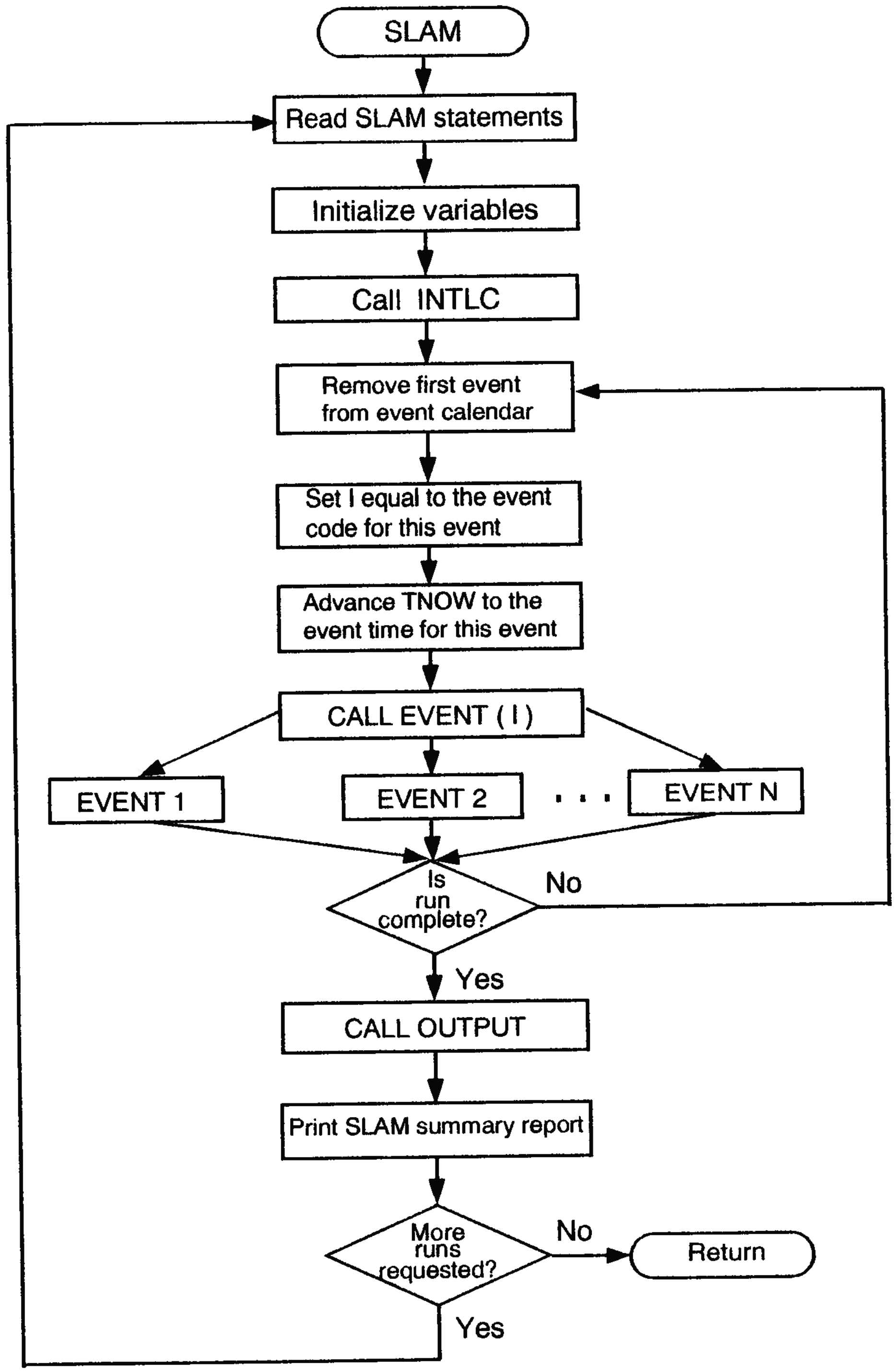

Figure 1-1 SLAM II Next Event Logic for Simulating Discrete Event Models

1.4 The Objectives

The objective of this study is to investigate the reliability (or availability) of a representative FMS by means of the methods outlined in section 1.2 and 1.3 , namely :

(1) Building up a mathematical model using the state space method (the Markov process);

(2) Using SLAM II discrete event modelling and simulation concepts.

The state space method and SLAM II discrete event modelling and simulation approach are the major tools in this study. The reliability measures thus obtained are contrasted against each other and the relative merits of each method are discussed. In chapter 2, a representative FMS with AGV based material handling system is presented. The system description of machines, AGVs, the plant layout and routings will also be given in this chapter.

The definition of reliability for AGV based FMS, the assumptions and system reliability models will be given in chapter 3. Chapter 4 describes the reliability analysis, the state truncation technique and state merging technique , and the reliability evaluations. Chapter 5 outlines the SLAM II simulation of FMS's reliability modelling. Chapter 6 deals with the FMS system performance simulation. Finally, chapter 7 presents the conclusions of this study.

2.0 THE FLEXIBLE MANUFACTURING SYSTEM (FMS) WITH AGV BASED MATERIAL HANDLING

2.1 The Chosen FMS

In a typical flexible manufacturing system, the number of machining centers is usually between two and six [10]. The chosen FMS as shown in Figure 2 - 1, consists of 2 AGVs (AGV1 and AGV2), 5 machines (MC1 to MC5) and 1 load /unload station (L/U). These facilities are linked to a network of computers that control their operations. There is no direct human element involved in the transportation of materials between various locations. Here , we assume that the environment is highly automated and once the tools are loaded and the parts assigned, the FMS can operate under complete computer controls. Each machine has an input buffer to accept the parts that will wait to be machined and an output buffer which accommodates the finished parts that will wait to be taken away either to next workstation (machine) or to the L/U station. Each input buffer and output buffer has a capacity to accept a maximum of 10 parts. The loading and unloading operation of parts for each machine will be completed by a robot. This means that each machine has a robot arm to carry out operations such as unloading the parts from AGVs to input buffer and loading the finished parts onto AGVs from the output buffer. Whenever an AGV is requested, always the nearest available one is dispatched. The flexible manufacturing systems are acclaimed for their flexibility to manufacture a large variety of parts with high efficiency and for their ability to respond quickly to parts changes. Two major reasons for such a flexibility are :

(a) Identical parts can have alternative routes within the system ;

(b) Each machine within the system is equipped with efficient tool changers

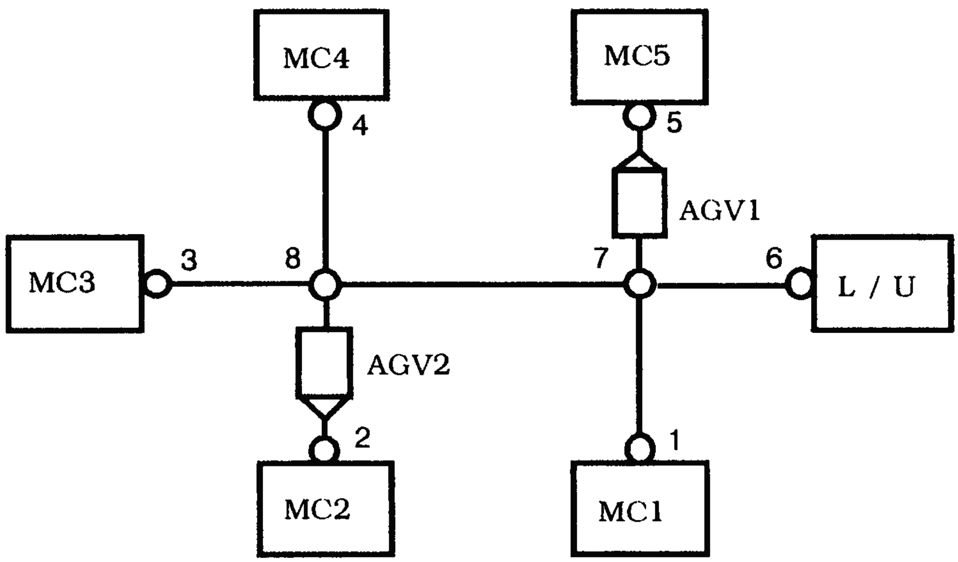

Figure 2-1. Layout of the flexible manufacturing system.

which significantly increase system capability during the manufacturing cycles.

2.2 AGV Based Material Handling System

During the past several years, automated guided vehicle system as a major material handling means has received much attention by designers and engineers of automated manufacturing systems [11]. The AGV systems have been widely used in flexible manufacturing systems because the AGV systems provide higher flexibility than the conventional systems. The AGV control system dispatches idle vehicles to move pallets, parts and tools between work centers within an FMS. Five types of AGVs are available :

(1) Unit load

(2) Towing

(3) Pallet truck

(4) Fork truck, and

(5) Assembly line vehicles

In this study, the unit load vehicles will be employed. This means that the vehicles only take one part (or one pallet) each time. Generally an AGV system contains four major components :

(1) transportation network

(2) vehicles

(3) interface between the production system and AGVs

(4) control system

Basically, there are three types of transport networks :

(1) single line

(2) simple loop and

(3) network type

In the case of multi-vehicle systems, the network-type system requires more complicated control logic. There are several design strategies for resolving the traffic problems. This is because at the junction of AGV tracks, vehicle interference and collision can take place when more than one vehicle try to use the same track. In such situations a control zone, which allows only one vehicle to pass through, can prevent the occurrence of collisions at the junctions. In addition, buffers may be provided for the vehicles waiting to use the control zones. Several designs of buffers have been suggested [12]. They include provision of loops, sidings and spurs on either sides of the AGV tracks. Figure 2-2 shows a layout of machines served by multiple vehicles. Buffers and control zones are provided at the junction of AGV tracks. Location 6 is the load/unload station and location 7 the central buffer work-in-process.

An alternate strategy is to divide the entire network of AGV tracks into a few small closed loops, each of which allows only one vehicle to circulate as shown in Figure 2-3. This is a modification of the layout shown in Figure 2-2. The layout in Figure 2-3 shows the two single-vehicle loops. Location 7 facilitates inter-loop transfer of materials. This modified design removes the problems of vehicle collision and interference and simplifies the traffic management. A central buffer, suitably placed, facilitates inter-loop transfer of jobs. The drawback of this arrangement is its inability to tackle vehicle breakdowns which will paralyze the loops. From the reliability consideration, this is not a good arrangement. The problems such as creation of bottleneck loops due to unbalanced loop loads and requirement of additional space, guide path and storage points may also arise.

In this study, a different design strategy will be tested for the layout shown in Figure 2-1.

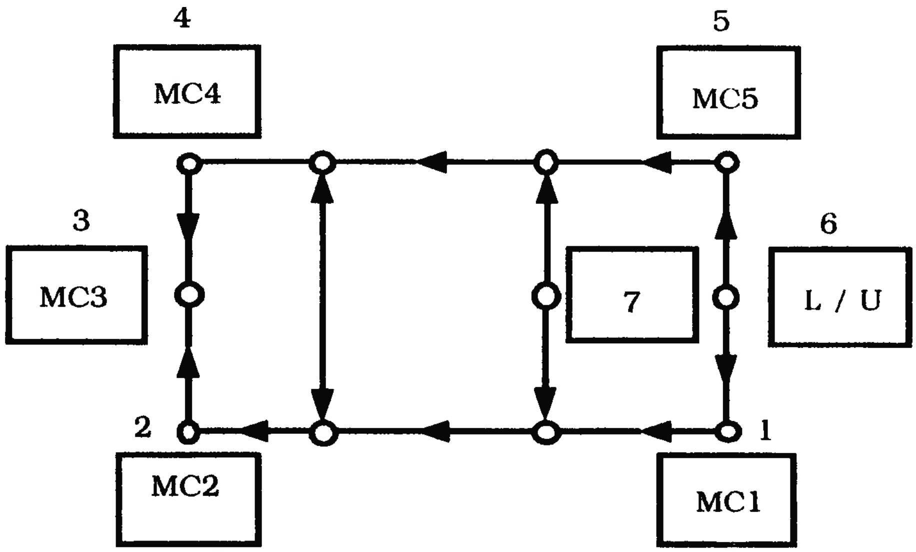

Figure 2-2 Layout of Machines Served by Multiple Vehicles

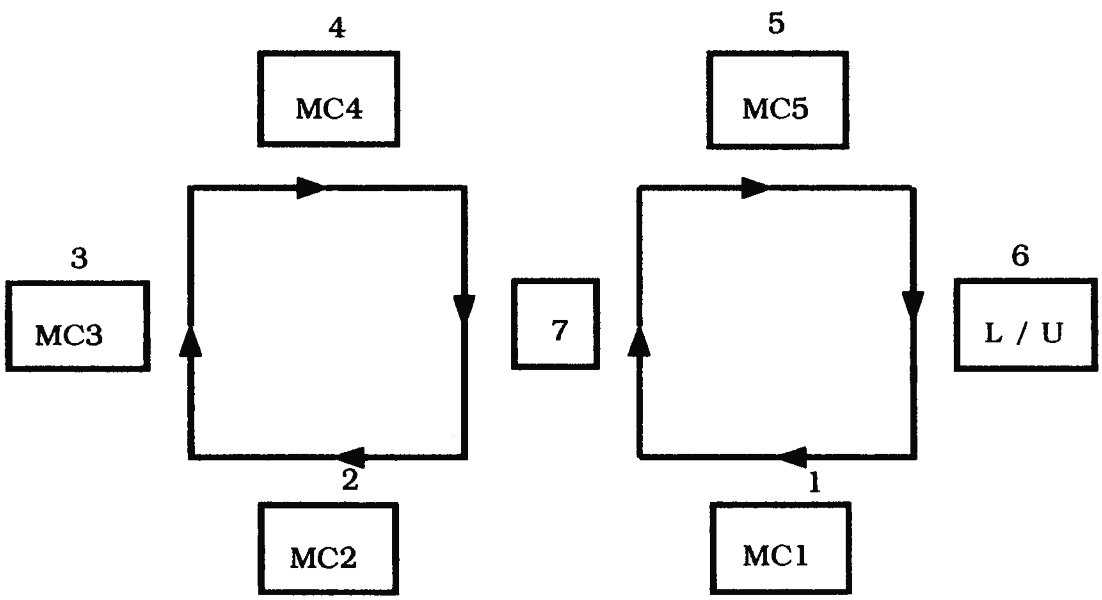

Figure 2-3 Layout of the Single Vehicle Loop Configuration

This is basically a simple network-type layout. The advantage of this type of layout is that for multi-vehicle systems, some of the AGV tracks can be used as the buffer, and more than one AGV can use the same track such as the track between the turning points 7 and 8. For example, when one AGV is requested to take a part from L/U station to MC2, MC3 or MC4, and at the same time another AGV is requested to take a part to MC1, MC5 or L/U station from other machining center, all the tracks between MC2, MC3 or MC4 and turning-point 8, and between MC1, MC5 or L/U station and turning-point 7 could be used as the buffer to accept the AGV which is waiting to use the track between turning-points 7 and 8. The central buffer for work-in-progress is dismissed because each machine has its own output buffer to accept the work-in-progress. According to the dispatching rule (this will be discussed in later chapters), AGVs are sent to the machines requesting either to take a part to the machine or to pick up a part from the machine. Whenever an AGV takes a part to a machine and after the part is unloaded, the output buffer of the machine will always be checked to see if there is any finished part waiting to be sent to its next destination, if so, the AGV will take the part which has the longest waiting time.

The estimation of the minimum number of vehicles required is another important factor. In order to determine the number of vehicles required, information regarding the jobs undergoing processing must be obtained. The jobs that are manufactured simultaneously, the processing times, and the arrival rate influence the traffic intensity in the system. Due to the versatility of the machines, often the jobs can be processed in more than one sequence. This permits alternate routing of the jobs due to machine failure or work-load balance considerations.

The desirable number of AGVs to accomplish the given load movements, assuming that the target production plan and job routings are known, must be determined when the AGV dispatching rules are investigated. Too many AGVs may create a higher possibility of collision and blocking, hence prohibiting the efficient control of the AGVs. The minimum number of AGVs needed to perform the assigned tasks must be considered in order to minimized the effect that too many AGVs may have on the dispatching rule performance. Dong-Soon Yim and R . J. Linn [13] developed an extended procedure to determine the minimum number of AGVs needed considering the random effects under steady-state if the idle time of the AGVs is ignored. Once the minimum number of AGVs is determined analytically, the minimum number of AGVs considering the time-dependent effects can be determined from the experimental simulation.

Let $F(i, j)$ be the required flow matrix from node i to node j (i.e. pickup or delivery point) for the movements of parts during a specified working time Tw. The required flow matrix is obtained from the target production rate and job routings. Let $D(i, j)$ be the transportation time from node i to node j obtained by the shortest route. As a conservative measure, it is assumed that there are always parts waiting for an AGV. The probability that the AGV is waiting at j node in steady state $Pw(j)$ is

$$Pw(j) = \sum_{i=1}^{n} F(i,j) \Big/ \sum_{i=1}^{n} \sum_{j=1}^{n} F(i,j) \qquad (2\text{-}1)$$

Also , the probability that the ith pickup node calls an idle AGV in steady-state, $Pc(i)$, is

$$Pc(i) = \sum_{j=1}^{n} F(i,j) \Big/ \sum_{i=1}^{n} \sum_{j=1}^{n} F(i,j) \qquad (2\text{-}2)$$

complete movement of a load includes

(1) an empty vehicle moves to a pick-up point

(2) picks up a part

(3) moves to a drop off point with the loaded part and

(4) delivers the part

The average transportation time for the empty vehicle, Tv, is then

$$Tv = \sum_{i=1}^{n} \sum_{j=1}^{n} Pw(j)\, Pc(i)\, D(j,i) \sum_{i=1}^{n} \sum_{j=1}^{n} F(i,j) \qquad (2\text{-}3)$$

and , the average transportation time for loaded vehicle, Tp, is

$$Tp = \sum_{i=1}^{n} \sum_{j=1}^{n} F(i,j)\, D(i,j) \qquad (2\text{-}4)$$

Letting l and u be the fixed pick-up and delivery time of a part at each workstation , the pick-up and delivery time for total part flow Tl is

$$Tl = \sum_{i=1}^{n} \sum_{j=1}^{n} F(i,j)(l+u) \qquad (2\text{-}5)$$

So, the minimum number of AGVs required to accomplish the load movements during Tw can be obtained as

$$\text{minimum number of AGVs} = (Tp + Tv + Tl) / Tw \qquad (2\text{-}6)$$

the simulation experiments performed by Dong-Soon Yim and R. J. Linn [13] showed that the minimum number of AGVs for the system they considered was two under the assumption of a target production rate 60 units per 8-h shift. Increasing the number of AGVs did not improve the system performance. For the study in this thesis, two AGVs will be used and the AGV dispatching rules and assumptions etc. will be discussed later in chapter 6 for performance simulation.

3.0 RELIABILITY DEFINITIONS AND MODEL CONSTRUCTION

3.1 Definitions

There are several definitions of reliability in the literature of system reliability analysis and the most often quoted one is " the probability that the system will perform its intended function for a given period of time under stated environmental conditions . " This definition is , however , inadequate for many occasions and is restrictive in its scope of application. It is more appropriate to talk of quantitative measures which when compared with reference indices , would indicate the expected consistency without deviation from the required performance.

It is usual in the literature to define reliability indices in terms of system success or failure. However , many complex systems usually have several levels of failure. For example , for a large piece of complex equipment we may not be able to simply say it is working or not working , as it may have many possible output states. So, it is therefore appropriate to define the calculated reliability measures in terms of a subset X which may contain any number of system states. If the success and failure states are denoted by X+ and X- , then **reliability** is the probability of being in X+ at time t without entered X-. The term reliability is used in many ways and most often in a qualitative sense to indicate the **ability of the system to perform its intended function.** As an intrinsic system parameter, the reliability can be measured by various indices. The definition given above is more specific and reliability is considered as a mathematical quantity which is itself a measure. This measurement may be time specific, i.e., function of time , or steady state when we refer to the equilibrium conditions. The former is required when concerning with the transient behavior of the system and the latter while considering the average behavior over long time.

The following indices are commonly used for repairable systems

(a) In the transient domain, the time specific availability of subset X+.

This is also called pointwise availability or instant availability and is the probability of the system being in any state contained in X+ at a particular instant of time *t*.

(b) Steady state availability of X+.

Commonly called availability , this is the limiting value of both pointwise availability and fractional duration . This can therefore, be interpreted in two ways. The first is the probability of being in a state contained in X+ at some point in time remote from the origin. The second is the time spent in X+ as fraction of the total time (0 , T) tends to be very large

3.2 <u>The Reliability Model Build-up of the FMS</u>

The FMS under study consists of five machines with two AGVs as material handling system as shown in Figure 2-1. The system operating sequence is as shown in Figure 3-1. According to the dispatching rules and prescribed job types, the nearest idle AGV (either AGV1 or AGV2) will be sent to the pick-up point to take the job and transport it to its destination (either of machines 1 to 5 or the load / unload station). Whenever a job arrives to the system, it is first taken by an AGV, the first machine is visited, then taken by an AGV again and the second machine is visited and so on. It always follows the sequence of AGV→MC→AGV→MC, until the job is completed, sent to the load/unload station where it is considered to have left the system .

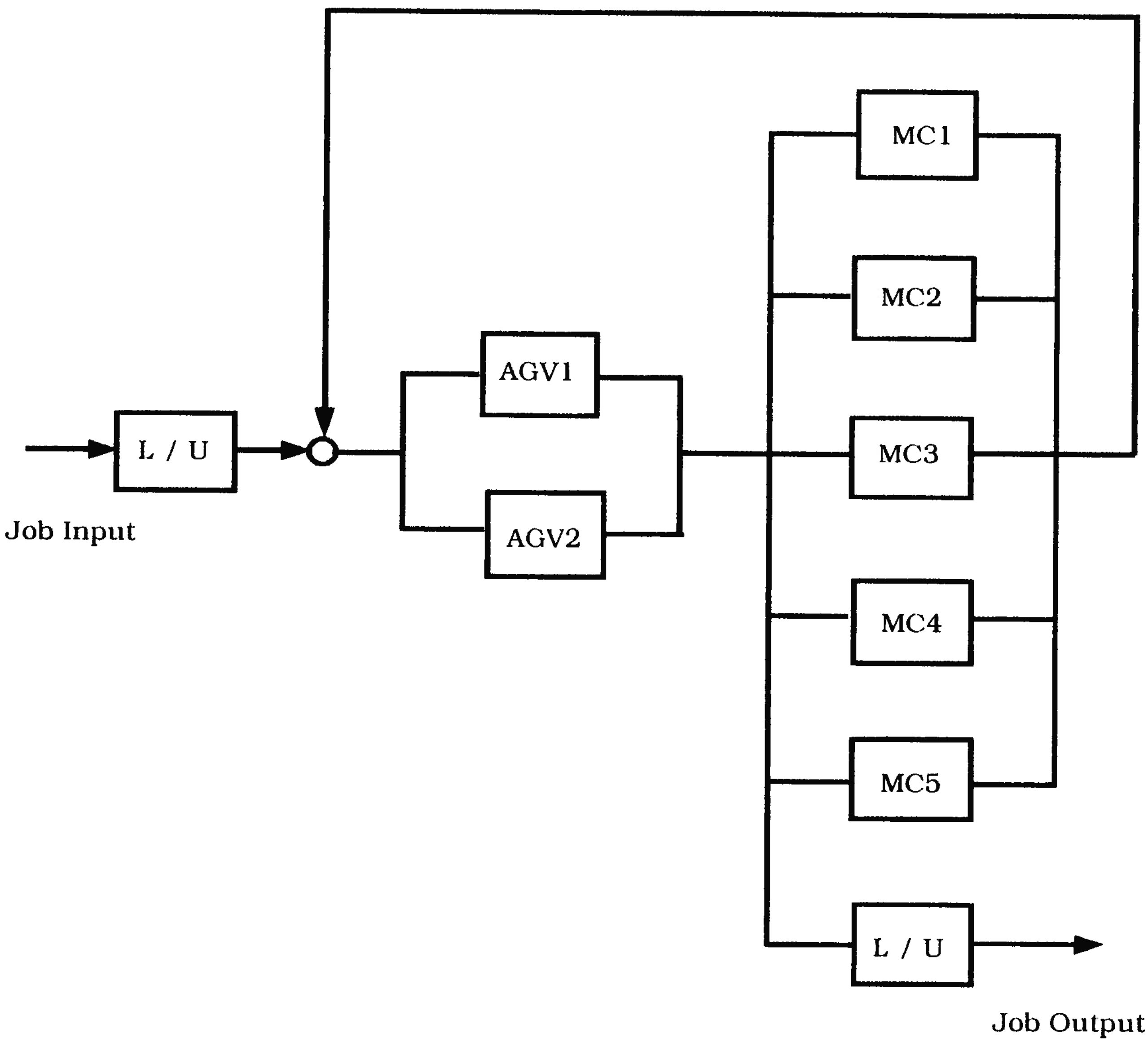

Figure 3-1 Functional diagram of the AGV based FMS.

In order to build up the reliability model of the FMS, the following assumptions are made :

(1) The failure and repair rates of AGVs and machines are constant. This means that the failure and repair rates(the transition rates) are not functions of state residence time.

(2) The failures of AGVs and machines are statistically independent.

(3) The repair operation will start immediately after the failures of the machines or AGVs. The repaired AGV or machine is as good as new.

(4) Both AGV1 and AGV2 are identical with same failure and repair rates. But for the five machines, the transition rates are different.

These assumptions are typical and common in the reliability analysis of systems. One further assumption that is made here is that there is always at least one of each job type in the system. According to the functional diagram of the FMS (Figure 3-1), the system can be divided into two major subsystems, subsystem 1 the material handling subsystem and subsystem 2 the processing subsystem. Each subsystem could be further broken down into small units such as subsystem 1 where its two components are AGV1 and AGV2, and subsystem 2 where its components are the five machines. Each machine or AGV consists of different machine parts and these machine parts are either repairable or nonrepairable. The replacement of a nonrepairable part is interpreted as a "repair" process. The reliability of an AGV or a machine is a function of the reliability of these elementary machine components .

The techniques for mathematically deriving the reliability measures, that are relevant to this study can be broadly classified as:

(i) state space approach

(ii) network method

(iii) decomposition using conditional probability approach

The state space approach is conceptually general and flexible and can be used for various systems with independent failures and makes it possible to take into account the dependent failures. But in very large systems which contains too many states, it may be difficult to apply this technique. For the FMS system under study that consists of seven components (2 AGVs and 5 machines), if we consider all the states of the system, there will be $2^7 = 128$ possible states and the transition rate matrix will contain $128 \times 128 = 16384$ elements. It becomes very cumbersome to conduct calculations.

The network approach, when applicable, usually provides a shorter route to the solution. But this approach is usually not suitable when dependent failures or repairs are involved. To apply this approach, it is necessary to recognize the difference between two types of function diagrams, the physical (or Block Schematic) Diagram and Logical (or Reliability Block) Diagram. The first diagram describes the actual connections between the components. Each block is a component and the diagram shows the manner in which they are actually connected. For the FMS under study, Figure 3-1 shows the physical connection between the components (AGVs and machines). But this is not the Logic or Reliability Diagram. For many systems, it is sometime very difficult or even impossible to prepare the Logic or Reliability Block Diagram.

The Decomposition Using the Conditional Probability Approach consist of breaking down a complex system into simple subsystems by the successive application of the conditional probability theorem. The idea is to first calculate the reliability measures of the simpler subsystems and then combine these results to obtain the values for the system. The selection of the component or subsystem which is the key component or subsystem is therefore important. This method can be used to simplify both the state space as well as the network approach.

For the system shown in Figure 3-1 , the state space approach becomes the only method available. Some techniques will be used to simplify the calculation. These will be depicted in the next chapter. To apply the state space approach , we must enumerate all possible system states as shown in Table 3 -1. The total system states are 128. The transition rate matrix will have 128×128 = 16384 elements as shown in Figure 3-2.

The criteria of success and failure of the FMS shown in Figure 3-1 are as follows :

(1) The system is in " up " state

(a) at least three of the five machines are in " up " states ;

(b) at least one of the two AGVs is in " up " state

(2) The system will fall into the " down " states if

(a) both AGV1 and AGV2 are " down " at same time ;

(b) more than two machines are " down " state coincidentally

The reliability definition of the FMS under study can now be stated as : " The probability of the successful functioning of both material handling subsystem with at least one AGV in up state and the processing subsystem with at least three machines in up state." Although, these criteria are to some extent arbitrary, however, the logic has been to strike a balance between the definitions of system "up" and system "down " states.

Cmpnt / State	AGV1	AGV2	MC1	MC2	MC3	MC4	MC5
1	U	U	U	U	U	U	U
2	D	U	U	U	U	U	U
3	U	D	U	U	U	U	U
4	U	U	D	U	U	U	U
•				•			
•				•			
•				•			
128	D	D	D	D	D	D	D

Table 3-1. System States.

$$R = \begin{bmatrix} R_{1\,1} & R_{1\,2} & R_{1\,3} & & R_{1\,128} \\ R_{2\,1} & R_{2\,2} & R_{2\,3} & \bullet\ \bullet\ \bullet & R_{2\,128} \\ R_{3\,1} & R_{3\,2} & R_{3\,3} & & R_{3\,128} \\ & & \bullet & & \\ & & \bullet & & \\ & & \bullet & & \\ R_{128\,1} & R_{128\,2} & R_{128\,3} & & R_{128\,128} \end{bmatrix}$$

Figure 3-2. State transition matrix (128 × 128 = 16384 elements).

4.0 RELIABILITY ANALYSIS OF THE FMS

4.1 Reliability Evaluation of The System

The FMS system shown in Figure 3-1 will be divided into two subsystems (subsystem 1 and subsystem 2) based on the system functions as shown in Figure 4-1. This functional subdivision leads to a more convenient reliability evaluations of subsystems 1 and 2 individually. The advantage in doing so is that the probabilities of the system can then be found by simple multiplication of the probabilities of the subsystem states. Another advantage is that the combination of the independent subsystems is simpler and the equivalent transition rate concept can be more conveniently employed.

The state space of each subsystem may be reduced either by merging states or by truncating very low probability states. The subsystems will then be combined into a complete system and the required reliability measure evaluated.

Two important concepts for reliability evaluation in large systems , namely , the merging of the states and truncating of the states will be employed in this study. The state merging technique will be used for subsystem 1 and the state truncation technique will be used for subsystem 2. These will be discussed in the following sections respectively.

4.2. The State Merging Technique

The basic idea is to find a state space which is equivalent to the original state space but is more convenient to use [2]. The method starts from the concept of equivalent transition rate. The state space X of the stochastic process Z(t) is assumed to be partitioned into two disjoint subsets X+ and X-. If any state of the subset is entered, that subset is said to have been encountered. We define

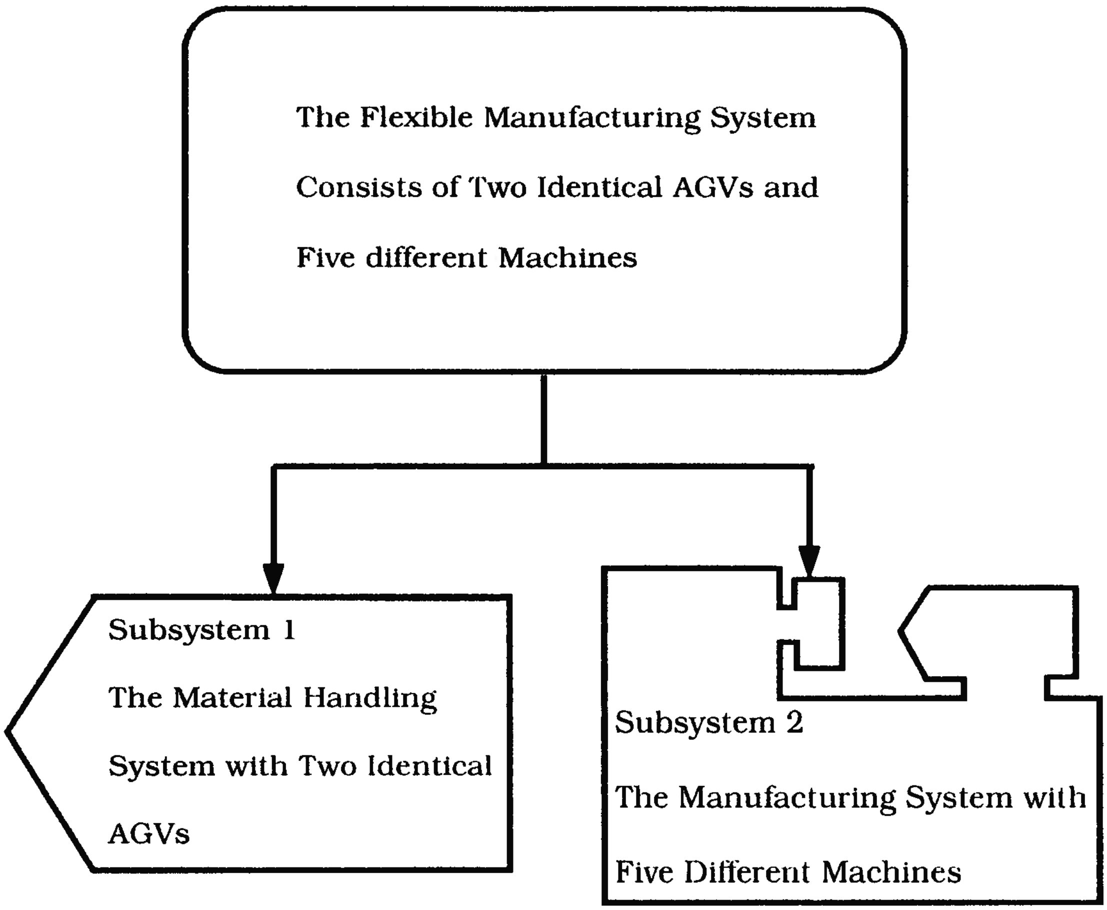

Figure 4-1. Subdivision of the flexible manufacturing system.

$\lambda_{X-X+}(t)$ = The equivalent transition rate from subset X- to X+ , then

$$\lambda_{X-X+}(t) = \sum_{i \in X-} \sum_{j \in X+} P_i(t)\, R_{ij} \Big/ \sum_{i \in X-} P_i(t) \qquad (4\text{-}1)$$

where

R_{ij} = The transition rate from state i to state j

$P_i(t)$ = The probability of being in state i at time t, for the given initial condition.

The important application of this concept is in reducing the system state space. The states (either in subset X- or in subset X+) can be merged and the equivalent transition rate from the merged states found by the application of Equation (4-1).

For subsystem 1, the material handling system of the FMS contains two identical AGVs. The total states of this subsystem is $2^2 = 4$ as shown in Figure 4-2. If this four state subsystem combines with subsystem 2 which contains five machines with $2^5 = 32$ states, the calculations are still complicated. From the reliability definition of the FMS, if at least one of the two AGVs in up state , the material handling system is in up state. The set of all the up states is { 1 , 2 , 3 }, denoted by l, which represent the up state of the material handling system. The only down state is state { 4 } and is represented by m. The state space of the material handling system could be merged into two states as shown in Figure 4-3.

Assume that the entire state space is partitioned into m subsets, X_i, i = 1, 2, 3, ..., m. The equivalent transition rate from subset X_l to subset X_m is obtained by using Equation (4-1) as

$$\lambda_{lm}(t) = \sum_{i \in X_l} \sum_{j \in X_m} P_i(t)\, R_{ij} \Big/ \sum_{i \in X_l} P_i(t) \qquad (4\text{-}2)$$

From the reduced state transition diagram of material handling system , from state l to state m , the equivalent transition rate is

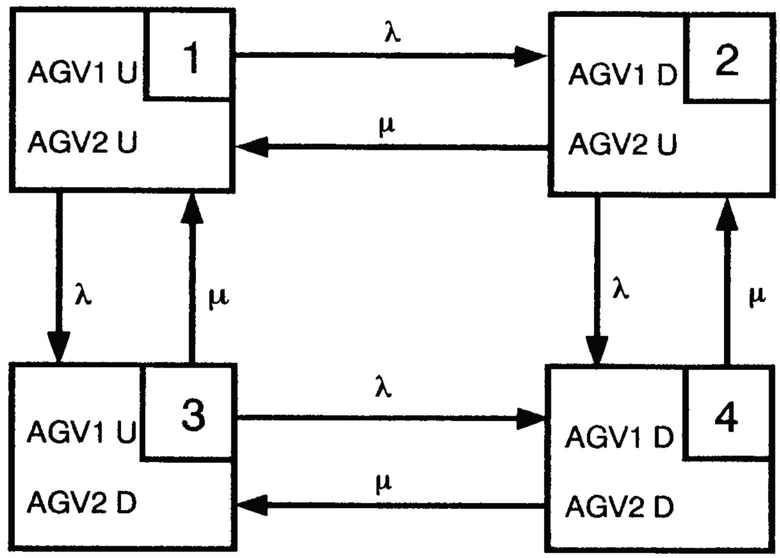

Figure 4-2. State transition diagram of material handling subystem.

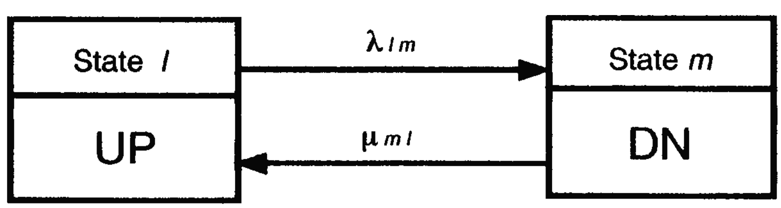

Figure 4-3. Reduced state transition diagram of material handling subystem.

$$\lambda_{lm} = \lambda(P_2 + P_3) / (P_1 + P_2 + P_3) \qquad (4\text{-}3)$$

and from state *m* to state *l*

$$\lambda_{ml} = 2\mu \qquad (4\text{-}4)$$

The four state subsystem as shown in Figure 4-2 is reduced to a two state system and the equivalent transition rates obtained as λ_{lm} and λ_{ml} could be calculated readily by using Equation (4-3) and (4-4). If this equivalent subsystem now combines with subsystem 2 (the processing system which contains $2^5 = 32$ states), the calculations could be much simplified.

4.3 The State Truncation Technique

The state space can be reduced by merging certain groups of states. Another technique is by truncating the state space , i.e. by neglecting the states whose contribution to the measures of system reliability is insignificant. In a system consisting of independent components, the probability of each state can be calculated individually by the product of individual component probabilities. The states required for determining the reliability measure are selected, their probabilities calculated and the reliability measure obtained. The system states which make a negligible contribution to the final results can be neglected and thus the system state space is reduced.

The philosophy behind truncation may be understood by examining the following equation for calculating the probability of the *i*th state

$$P_i = \sum_{k \in X-} P_k R_{ki} / \sum_{k \in X-} R_{ik} \qquad (4\text{-}5)$$

The contribution to P_i by a state $k \neq i$ is

$$P_k R_{ki} / \sum_{k \in X-} R_{ik}. \qquad (4\text{-}6)$$

i. e. the frequency of encountering state i from state k divided by the total transition rate out of i. Therefore if the states having low probability are deleted, the probability of state i will not be significantly affected. Of course the states have to be deleted prior to solving the set of linear equations. The procedure amounts to assuming that the deleted states have a probability equal to zero. Denoting the set of deleted states by X_T, the probability of this subset if there were no truncation is $P_T = \sum_{i \in X_T} P_i$, because the probability of the rest of the state space is now one, i. e.

$$\sum_{i \in (X - X_T)} P_i = 1 \qquad (4\text{-}7)$$

The probability P_T will be distributed over the states $i \in (X - X_T)$ where X is the system state space. If P_T is small, then the probability distribution of the rest of the states will not be significantly affected. The success of the truncation method depends upon selecting the low probability states for truncation. The following consideration should be kept in mind while employing the truncation technique :

(1) The probability P_i ($i \in X_T$) is less than P_j ($j \in (X - X_T)$). i .e. the biggest probability in the truncated subset should be less than the smallest probability of the remaining state space. If systems contains two state components , this is very easy to achieve. The state space may be divided into subsets , each subset having states of a certain level of coincident failures. For a system of n identical components, there will be $(n + 1)$ subsets. These subsets will have the following states :

subset number	state description
1	all components are up
2	one component is down
3	two components are down
•	•
•	•
•	•
$n + 1$	n components are down

An arbitrary level of truncation should be first selected; for example, the states having three or more than three coincident failures can be truncated. The computation can then be repeated by including the next subset, i. e. the states having three coincident failures. If the new values are not significantly different from the previous ones, the computation can be stopped, otherwise one more subset should be included and the computation repeated. In the state space truncation technique, the probabilities of the states adjacent to the truncation boundary are affected the most and the effect decreases when moving away from the boundary.

(2) After the states have been truncated, the state truncation diagram should be examined to see if the process of truncation has generated any absorbing states. The absorbing states can be located by examining the transition rate matrix. An absorbing state will have transitions into it but not out of it. The ijth element of the transition rate matrix gives the transition rate from state i to state j. Therefore if the ith row is empty (all elements are zero), this means that the ith state is absorbing. Either the absorbing state should be deleted or the state where truncation has generated this absorbing state should be retained.

One important method of state truncation technique is the sequential truncation. This can be described as the process of building the reliability model by adding components or adding subsystems one by one and deleting the low probability states at each step. In sequential truncation , the state probabilities are calculated at each step and the states with probabilities less than a reference value are deleted. The assumption, which is generally valid, is that the probability of a given state will be decreased after another component has been added to the system . Another method is the direct state space truncation. In direct truncation , the decision to delete states has to be made prior to the solution of the state probabilities.

For the FMS system , the subsystem of processing facilities contains five machines. The total states of this subsystem is $2^5 = 32$ as shown in Table 4-1. The transition matrix will contain $32 \times 32 = 1024$ elements. It is still tedious to be handled by mathematical means. The reliability definition of the FMS stated that if there are at least three machines in up state, the subsystem 2 is in up state, and if three machines coincidently catch the down state , the processing subsystem is considered being in down state . Using the direct state space truncation method , from Table 4-1, the subsets 4 ,5 and 6 will be truncated, i.e. we consider that the probability of three and more than three machines coincident failure is zero. So these 16 states with very low probability will be deleted. The state space diagram after truncation is shown in Figure 4-4 . If we define λi as the failure rate and μi as the repair rate of machine MC i ($i = 1,2,3,4,5$), the transition rate matrix can be determined as described in the following sections.

4.4 The Reliability Analysis of The FMS by Mathematical Approach

Based on the assumptions given in chapter 3, the reliability of the FMS's could be modelled as the continuous parameter Markov process mathematically as given in Equations (1-1) to (1-4). Equation (1-4) is the conditional probability density function for the continuous

Subset #	# of machines failed	# of identical state in the subset
1	0	$\binom{5}{0} = 1$
2	1	$\binom{5}{1} = 5$
3	2	$\binom{5}{2} = 10$
4	3	$\binom{5}{3} = 10$
5	4	$\binom{5}{4} = 5$
6	5	$\binom{5}{5} = 1$
Total # of states		32

Table 4-1 Subdividing the State Space According to Identical States

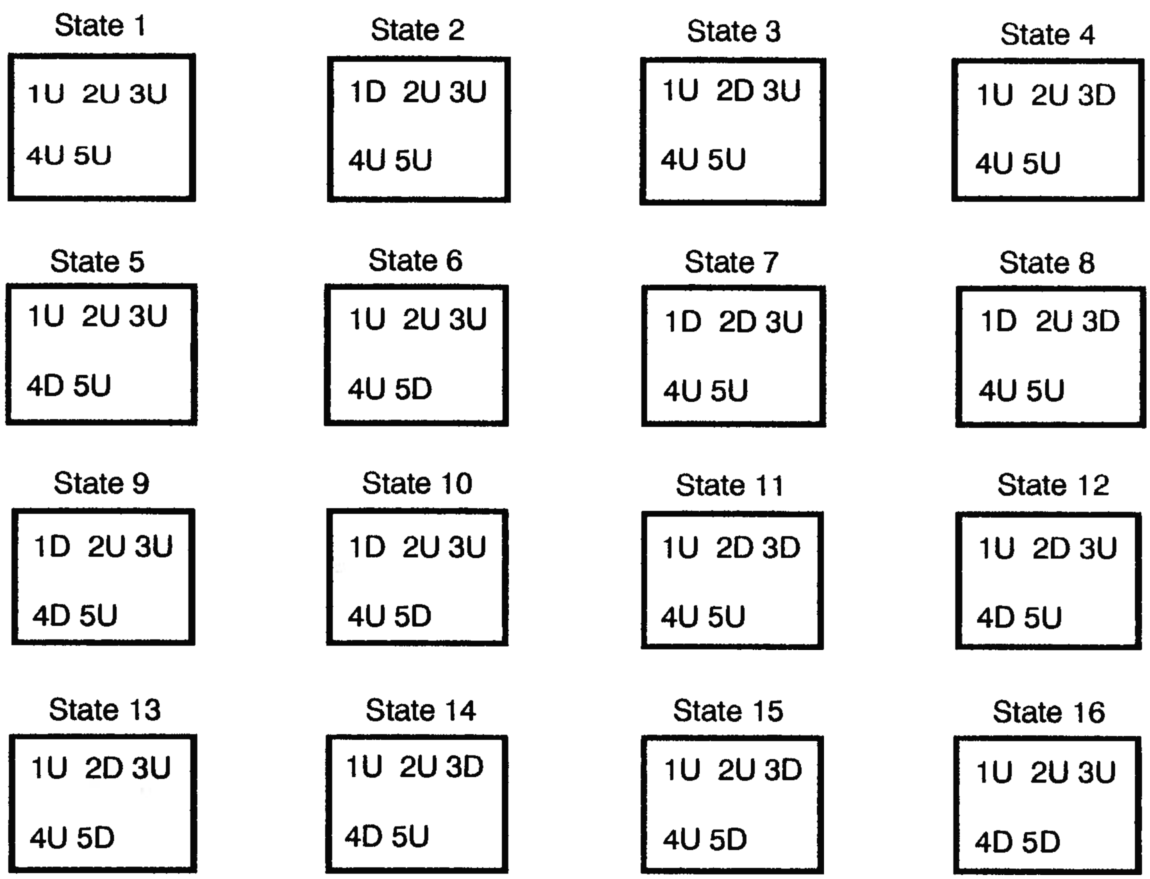

Figure 4-4. State space diagram of subsystem 2 after truncation, where *i*U : the *i*th machine is up ; *i*D : the *i*th machine is down *i* = 1,2,3,4,5.

parameter Markov process given by the Chapman-Kolmogorov equation , that is

$$P_{ij}(t+x) = \sum_{k} P_{ik}(t) P_{kj}(x)$$

The transition probabilities must satisfy the following conditions :

$$0 \le P_{ij}(x) \le 1 \qquad (4\text{-}8)$$

and

$$\sum_{j} P_{ij}(x) \le 1 \qquad (4\text{-}9)$$

If $\sum_{j} P_{ij}(x) = 1$ for all i and x , then the process is called the honest process.

In a continuous parameter case, the equivalent elements are the limiting values, i.e. as x $\to 0$, we define the transition intensity or the rate as

$$\underset{i \ne j}{R_{ij}} = \frac{dP_{ij}(x)}{dx} \Big|_{x=0} = \lim_{\Delta x \to 0} \frac{P_{ij}(\Delta x) - 0}{\Delta x}$$

that is

$$P_{ij}(\Delta x) = R_{ij}\,\Delta x + 0(\Delta x) \qquad (4\text{-}10)$$

for $i = j$

$$R_{ij} = \frac{dP_{ii}(x)}{dx} \Big|_{x=0} = \lim_{\Delta x \to 0} \frac{P_{ii}(\Delta x) - 1}{\Delta x}$$

thus

$$P_{ii}(\Delta x) = R_{ii}\Delta x + 1 + 0(\Delta x) \qquad (4\text{-}11)$$

Differentiating both sides of Equation (4-9) for equality and setting $x = 0$,

$$R_{ii} + \sum_{j \ne i} R_{ij} = 0$$

i.e.

$$R_{ii} = -\sum_{j \neq i} R_{ij} \qquad (4\text{-}12)$$

Substitute (4-12) into (4-11) therefore

$$P_{ii}(\Delta x) = 1 - \sum_{j \neq i} R_{ij}\, \Delta x + 0(\Delta x) \qquad (4\text{-}13)$$

In Equation (4-10), $P_{ij}(\Delta x)$ represents the probability of transition from state i to state j during the interval of length Δx and this is equal to $R_{ij}\Delta x$ plus a term which divided by Δx tends to zero as $\Delta x \to 0$. Equation (4-13) can be interpreted in a similar manner. Equation (1-4) can now be written for a small increment of time Δt as

$$\begin{aligned} P_{ij}(t + \Delta t) &= \sum_{k} P_{ik}(t)\, P_{kj}(\Delta t) \\ &= P_{ij}(t)\, P_{jj}(\Delta t) + \sum_{k \neq j} P_{ik}(t)\, P_{kj}(\Delta t) \end{aligned} \qquad (4\text{-}14)$$

where

$$P_{ij}(t) = P\left(Z(t) = j \mid Z(0) = i \right)$$

Substituting from (4-10) and (4-11)

$$P_{ij}(t + \Delta t) = P_{ij}(t)\,(1 + R_{jj}\Delta t) + \sum_{k \neq j} P_{ik}(t)\, R_{kj}\, \Delta t + 0(\Delta t)$$

i.e.

$$\frac{P_{ij}(t + \Delta t) - P_{ij}(t)}{\Delta t} = P_{ij}(t)\, R_{jj} + \sum_{k \neq j} P_{ik}(t)\, R_{kj} + \frac{0(\Delta t)}{\Delta t} \qquad (4\text{-}15)$$

and as $\Delta t \to 0$, Equation (4-15) becomes

$$P'_{ij}(t) = \sum_{k} P_{ik}(t) R_{kj} \qquad (4\text{-}16)$$

If $P_i(t)$ denotes the row vector whose jth element is $P_{ij}(t)$, i. e. the probability of being in the jth state at time t given that the process was initially in state i, then Equation (4-16) can be written as

$$P'_i(t) = P_i(t) R \qquad (4\text{-}17)$$

where R is the transition rate matrix whose ijth element is R_{ij}. In a more general form Equation (4-17) becomes

$$P'(t) = P(t) R \qquad (4\text{-}18)$$

where $P(t)$ has $P_{ij}(t)$ as its (ij)th element. The initial condition for (4-17) is

$$P(0) = I$$

where I is the identity matrix.

If, however, the initial state of the system is defined by a probability distribution in the form of a row vector $P(0)$, then the state probability distribution at time t is given by $P(0)P(t)$. The system of Equations (4-18) is termed as the system of forward equations. The time specific state probabilities can be found by solving the differential equation (4-18) in the matrix form $P'(t) = P(t) R$ with the initial condition $P(0) = I$. Where,

$P(t)$ = the matrix whose (ij)th term $P_{ij}(t)$ denotes the probability of being in state j given that the process was in state i at time $t = 0$

R = the transition rate matrix

Equation (4-18) is a system of linear differential equations with constant coefficients. If the eigenvalues of the transition rate matrix R are distinct, the solution of Equation (4-18) can be obtained in the form

$$P(t) = S\,D(t)\,S^{-1} \tag{4-19}$$

where

D (t) = the diagonal matrix whose ($i\,i$)th element is $\exp\{ r_i t \}$, r_i being the ith eigenvalue of matrix R

S = the matrix formed by the right eigenvectors of matrix R

S^{-1} = the matrix formed either by inverting S or from the left eigenvalue of matrix R,

If the distribution at $t = 0$ is given by the row vector $P(0)$, the distribution at t is given by

$$P(t) = P(0)\,\mathrm{P}(t) \tag{4-20}$$

The ith element of the row vector $P(t)$ is denoted by $P_i(t)$, represents the probability of being in state i for the given condition at time $t = 0$.

After finding $P_i(t)$, the availability of the system , $A_+(t)$, can be calculated using the following equation

$$A_+(t) = \sum_{i \in X+} P_i(t) \tag{4-21}$$

4.4.1 Material Handling Subsystem

In order to calculate the availabilities of the four state material handling subsystem as shown in Figure 4-2 , the failure rate and the repair rate of the identical AGVs as an example case are selected as follows

$$\lambda = \frac{1}{100} \quad (1 / \text{hr.})$$

$$\mu = 1.0 \quad (1 / \text{hr.})$$

Mathcad [14] is used to calculate the state probabilities. Note that to use Mathcad subscripts must be given in parentheses. First we input the transition rates $R(i,j)$ ($i,j = 1,2,3,4$) of the subsystem ,here $R(i,j)$ is the transition rate from state i to state j. Where

$R(1,2)=\lambda$	$R(1,3)=\lambda$	$R(1,4)=0.0$
$R(2,1)=\mu$	$R(2,3)=0.0$	$R(2,4)=0.0$
$R(3,1)=\mu$	$R(3,2)=0.0$	$R(3,4)=\lambda$
$R(4,1)=0.0$	$R(4,2)=\mu$	$R(4,3)=\mu$

The diagnal elements can be calculated from Equation (4-12) , that is ,

$$R(1,1)=-2\lambda \qquad R(2,2)=R(3,3)=-(\lambda+\mu) \qquad R(4,4)=-2\mu$$

The transition rate matrix can be shown as

$$R=[\,R(i,j)\,] \qquad (4\text{-}22)$$

Next step is to calculate the eigenvalues of matrix R through Mathcad

$$r=\text{eigenvals}(R) \qquad (4\text{-}23)$$

The diagonal matrix $D(t)$ whose (i i)th element is $\exp\{r_i t\}$ can be formed as ,

$$D(i,i)=\exp\{r_i t\} \qquad \text{and}$$

$$D(t)=[\,D(i,i)\,] \qquad (4\text{-}24)$$

The final step is to construct the matrices S and S^{-1}. The matrix S will be constructed first then inverting S to get matrix S^{-1}. The eigenvectors corresponding to each eigenvalue of matrix R can be found through Mathcad by defining,

$$S_i=\text{eigenvec}(R,\ r_i)$$

and each vector corresponds to a column vector of the matrix S, i.e.

$S(i, j) = (S_i)_j$ and the matrix

$$S = [S(i, j)] \qquad (4\text{-}25)$$

Now the state transition probability matrix could be found from Equation (4-19) as

$P(t) = S\,D(t)\,S^{-1}$ or

$P(t) = S\,D\,S^{-1}$

The output of the matrix is :

$$P = \begin{bmatrix} 0.98 & 0.01 & 0.01 & 0.00009803 \\ 0.98 & 0.01 & 0.01 & 0.00009803 \\ 0.98 & 0.01 & 0.01 & 0.00009803 \\ 0.98 & 0.01 & 0.01 & 0.00009803 \end{bmatrix}$$

If the original state at time $t = 0.0$ is given by a row vector as $P(0) = \{1, 0, 0, 0\}$, then the distribution of the state probability at time t can be calculated using Equation (4-20),

$$P(t) = P(0)\,\mathrm{P} = \{0.98 \quad 0.01 \quad 0.01 \quad 0.00009803\}$$

This corresponds to state probabilities of the two AGVs at time t, where

$P_1(t) = 0.98$

$P_2(t) = P_3(t) = 0.01$ and

$P_4(t) = 0.00009803$

According to the reliability definition of the FMS, the subset of the up states of the material handling subsystem should include $P_1(t)$, $P_2(t)$ and $P_3(t)$ i.e.

$$X+ = \{P_1(t), P_2(t), P_3(t)\}$$

and the subset of down state is given by

$$X- = \{ P_4(t) \}$$

The equivalent transition rate between these two subsets can be calculated by using Equation (4-2) , that is

$$\lambda_{m} = \frac{P2 + P3}{P1 + P2 + P3} \cdot \lambda$$

$$= \frac{0.01 + 0.01}{0.98 + 0.01 + 0.01} \cdot \frac{1}{100}$$

$$= 0.0002$$ and

$$\mu_{m1} = 2\,\mu$$

$$= 2.0$$

So the four state reliability diagram of Figure (4-2) , was merged into two state system as shown in Figure (4-3).

4.4.2 The Processing Subsystem

After truncation , this subsystem contains 16 states as shown in Figure 4-4, and all these 16 states are up states according to the reliability definition of the FMS system . The failure and repair rates of the machines, as a numerical example are given bellow :

machine	failure rate (1 / hr.)	repair rate (1 / hr.)
1	$\lambda 1 = 1/80$	$\mu 1 = 1/0.8$
2	$\lambda 2 = 1/90$	$\mu 2 = 1/0.9$
3	$\lambda 3 = 1/100$	$\mu 3 = 1/1.0$
4	$\lambda 4 = 1/110$	$\mu 4 = 1/1.1$
5	$\lambda 5 = 1/120$	$\mu 5 = 1/1.2$

Following the same sequence as for material handling subsystem, we input the transition rates and calculate the transition rate matrix through Mathcad , that is

$R = [\, R(i, j)\,]$ (where $i, j = 1, 2, 3, \ldots\ldots 16$)

then calculate the eigenvalues and the eigenvectors of matrix R to construct the diagonal matrix D (t) and the matrices S and S^{-1}. The state probability matrix can be calculated through Mathcad as

$$P = S\, D\, S^{-1}$$

Given the initial state of the subsystem in form of a row vector,

$$P(0) = \{1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\}$$

The distribution of the state probability at time t can be calculated using Equation (4-20)

$$P(t) = P(0)\,\mathrm{P}$$

$$= \{0.914\ \ 0.009\ \ 0.009\ \ 0.009\ \ 0.009\ \ 0.009\ \ 0.00009145\ \ 0.00009132\ \ 0.00009142\ \ 0.00009147\ \ 0.00009135\ \ 0.00009144\ \ 0.00009703\ \ 0.00009134\ \ 0.00009138\ \ 0.00009146\}$$

4.4.3 Combining Subsystems 1 and 2

Because each component in both subsystem 1 and subsystem 2 can be repaired independently, the two subsystems are also independent. After finding the state probabilities of subsystem 1 and subsystem 2 , the state probability of the FMS can be found by combining the probabilities of both subsystems

As demonstrated in 4.4.1, the subsystem 1 is merged into a two states subsystem (Figure 4-3) with the equivalent transition rates $\lambda_{lm} = 0.0002$ and $\lambda_{ml} = 2.0$. In 4.4.2 , after truncation , the 32 state subsystem was simplified into a 16 state subsystem by dismissing the states where the coincident failures of more than two machines are considered as zero. According to the reliability definition of the FMS , these 16 states are all up states. So the whole FMS system can be described by three subsets as shown in Figure 4-5 (a) ,

1. at least one of the two AGVs is in up state
2. both AGVs are down
3. at least three of the five machines are in up states

After combining the three subsets , the reduced state transition diagram is obtained as shown in Figure 4 -5 (b) . It can also be observed (Figure 4-5) that when the two subsystems are independent (fail and repair independently), the interstate transition modes remain unchanged after interaction. By applying the state merging and state truncating techniques, the FMS system is simplified as a two state system with the equivalent transition rate of λ_{lm} and λ_{ml} as shown in Figure 4 - 5 (b). The state probability of the FMS system can be found by the products of the state probabilities of subsystem 1 and subsystem 2 as

$P_{UP} = P1_{UP}\, P2_{UP}$ and

$P_{DN} = P1_{DN}\, P2_{UP}$

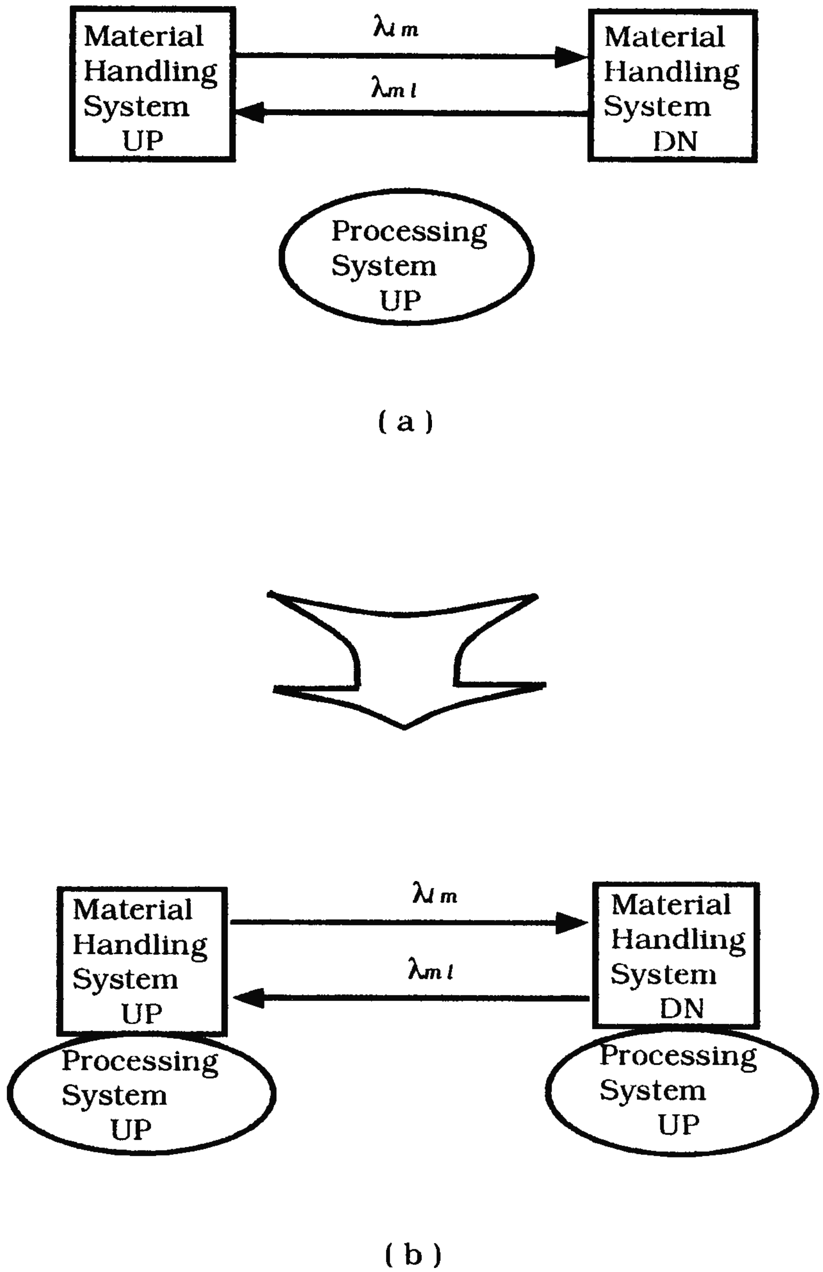

Figure 4-5. FMS state transition diagram after combining the three subsets.

The reliability of the system can be obtained through solving the following differential equation by setting $\lambda_{ml} = 0.0$ and defining RL the reliability of the FMS system,

$$\frac{d\,P_{UP}}{d\,t} + \lambda_{lm}\,P_{UP} = P_{DN}\,\lambda_{ml}$$

after integration, we obtain

$$RL(t) = P_{UP}(t) = \exp(-\lambda_{lm}\,t) \qquad (4\text{-}26)$$

5.0 SLAM II SIMULATION OF FMS RELIABILITY

5.1 The Processing Subsystem

To simulate the state probabilities of this subsystem using SLAM II , the simulation model should first be constructed. The purpose of this discrete event simulation is to collect statistics on each state of the processing subsystem . In the preceding chapters the FMS system was subdivided into the material handling subsystem and the processing subsystem . Because the processing subsystem contains more components , i. e. five machines , this subsystem is more complicated to handle and will be discussed first . As already has been outlined , the processing subsystem consists of 32 possible states. As the time advances , the time based random variable Z(t) should randomly encounter each of these 32 states. In simulation, the associated logic for processing the changes in state is an event. So a discrete event model of the subsystem is constructed by defining the event types that can occur and then modelling the logic associated with each event type. The state of the subsystem in a discrete event model is represented by variables which have attributes. The state of the model is initialized by specifying the initial values for the variables employed in the simulation and by the initial scheduling of the events. Here two types of events are defined for the simulation :

a) failure event and ;

b) repair event .

As discussed in the previous chapters , the transition rate is the hazard rate of the random variable defining the duration of state i under the condition of transiting to state j. The negative exponential is the only distribution having a constant hazard rate and therefore the random variables underlying the time homogenous Markov process must be exponentially distributed. i.e.

$$R_{ij} = \frac{1}{\text{mean value of } x_{ij}} \qquad (5\text{-}1)$$

where

R_{ij} = the transition rate or hazard rate and

x_{ij} = the duration of state i under the condition of transiting to state j

In the case where up state i transits to down state j, the mean value of x_{ij} is called the mean time to failure (MTTF) , and if the state is transiting from down state to up state then x_{ij} is called the mean time to repair (MTTR). Here x_{ij} is a continuous random variable. i.e. if i is the up state under the condition of transiting to the down state j :

$$\lambda = R_{ij} = \frac{1}{\text{MTTF}} \qquad (5\text{-}2)$$

if i is down state under the condition of transiting to the up state j then

$$\mu = R_{ij} = \frac{1}{\text{MTTR}} \qquad (5\text{-}3)$$

The distribution of the random variable x , is negative exponential and has a probability density function , for transiting to the failure state, defined as

$$f(x) = \lambda \cdot e^{-\lambda \cdot x} \qquad (5\text{-}4)$$

The corresponding cumulative distribution function is given by

$$F(x) = \int_0^x f(u)\,du$$

$$= 1 - e^{-\lambda \cdot x} \qquad (5\text{-}5)$$

Because each machine fails and is repaired independently, the failure and repair event will also be scheduled independently for each machine. That is, each machine is modelled as the two state Markov process. The SLAM II global variable XX (I) is used to represent the Up and Down states :

XX (I) = 1 , machine I is in Up state

XX (I) = 0 , machine I is in Down state

The combinations of states of all the machines give all the possible states of t he system. The time persistent statistics are collected when XX(I) is specified on the TIMST input statement. The output gives the state probabilities of the system concerned. As discussed in chapter 4, the 32 state processing subsystem is reduced to a 16 state subsystem by applying the state truncation technique. Statistics on all these 16 state probabilities is collected. The global variables for representing every status in the simulation are shown in Table 5-1. To initialize the simulation, all the initial values of the global variables are defined in SUBROUTINE INTLC of SLAM II. The initial condition of the system is all the machines in UP state. This is represented by defining the global variable XX (I) = 1(here I = 1, 2,· · ·, the number of machines) meaning that at the current instant of time, this state of the system is encountered. At any instant of time, the system can only stay in one state (we represent this state by setting the global variable XX (I) = 1) and all the rest of the states which are not encountered are represented by setting the global variable XX (I) = 0. To initially schedule the failure event in the SUBROUTINE INTLC, the SLAM II's SCHDL (KEVENT, DTIME, A) subroutine is called. Where KEVENT denotes the event code of the event being scheduled and DTIME denotes the number of time units from the current time, TNOW, that the event is to occur. Attributes associated with an event are specified by passing the buffer array A as the third argument of subroutine SCHDL. For this

Table 5-1. Global variables used for representing states in SLAM II simulation program

global variable XX (I)	variable description XX (I) = { 1.0 , the UP state or a state encountered; 0.0 , the DN state or a state not encountered
XX (1)	state of machine 1 (MC1)
XX (2)	state of machine 2 (MC2)
XX (3)	state of machine 3 (MC3)
XX (4)	state of machine 4 (MC4)
XX (5)	state of machine 5 (MC5)
XX (21)	system state 1
XX (22)	system state 2
XX (23)	system state 3
XX (24)	system state 4
XX (25)	system state 5
XX (26)	system state 6
XX (27)	system state 7
XX (28)	system state 8
XX (29)	system state 9
XX (30)	system state 10
XX (31)	system state 11
XX (32)	system state 12
XX (33)	system state 13
XX (34)	system state 14
XX (35)	system state 15
XX (36)	system state 16

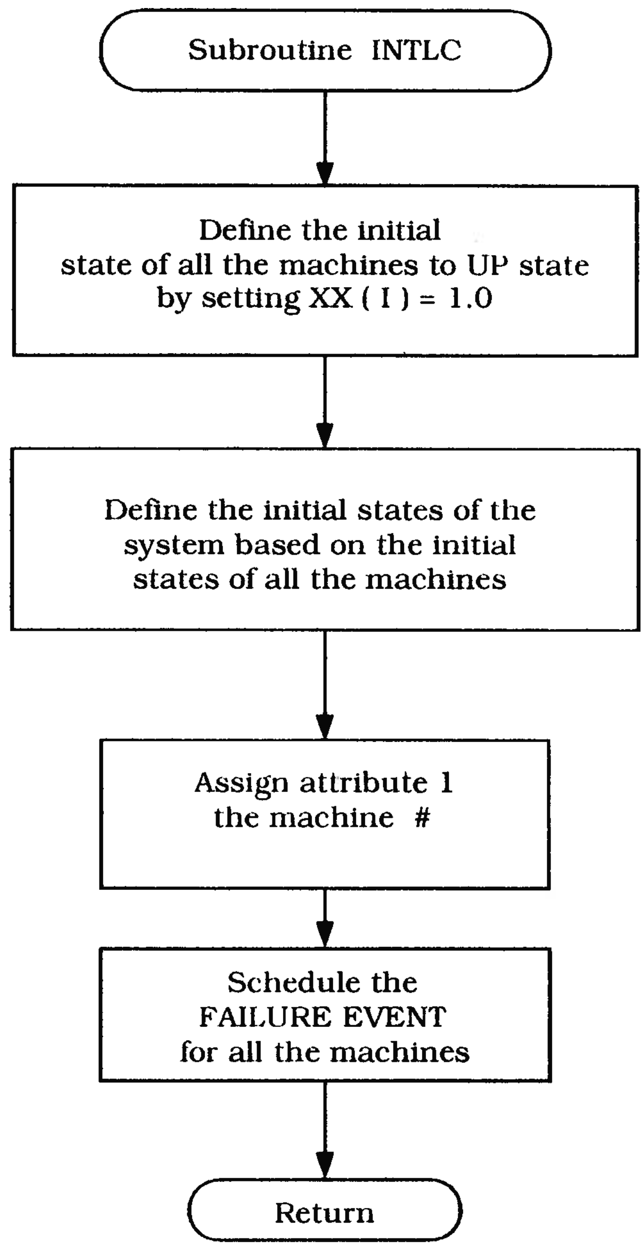

Figure 5-1 Flowchart of SUBROUTINE INTLC

simulation study, code 1 is used for failure event and code 2 for repair event. For the failure event , the number of time units is defined by an exponential distribution with the mean value of MTTF , i. e.

DTIME = EXPON (MTTF)

There is only one attribute (ATRIB (1)) being defined to identify the five machines, for example , ATRIB (1) =1.0 means that the failure event of machine 1 is scheduled. The flow chart of Subroutine INTLC is shown in Figure 5-1 .

Once the SUBROUTINE FAILURE is called , the failure event will happen. The serial number of a machine that has failed is identified by attribute 1 and the global variable XX (I) corresponding to the failed machine is set equal to zero indicating that the machine is catching the down state. The system state transits from the original one to current one by changing the value of global variable XX (I) in which the previous state is indicated by 0.0 and the current state by 1.0 meaning that the current state is encountered. The repair event will be scheduled in this subroutine for the failed machines. This is done by calling SLAM II subroutine of SCHDL (KEVENT, DTIME , A) . Where code KEVENT = 2, indicates a repair event and DTIME = EXPON (MTTR) is the repair time. The flow chart of SUBROUTINE FAILURE is shown in Figure 5-2.

When SUBROUTINE REPAIR is called , the failed machine has been repaired. The state of this machine is then changed from DN (down) to UP and the state of the system transits from the previous one, i.e. XX(I) =1.0, to current one, i.e. XX(I) =0.0. This is also done by resetting the value of the global variable XX (I) equal to 1.0 for the machine repaired and the system state encountered. The subsequent failure events are scheduled in SUB-ROUTINE REPAIR. The flow chart of subroutine REPAIR is shown in Figure 5-3. The SLAM II SUMMRY REPORT of all the statistics for time-persistent variables that represent every state of all the machines and the system is given in Figure 5-4 , where ,

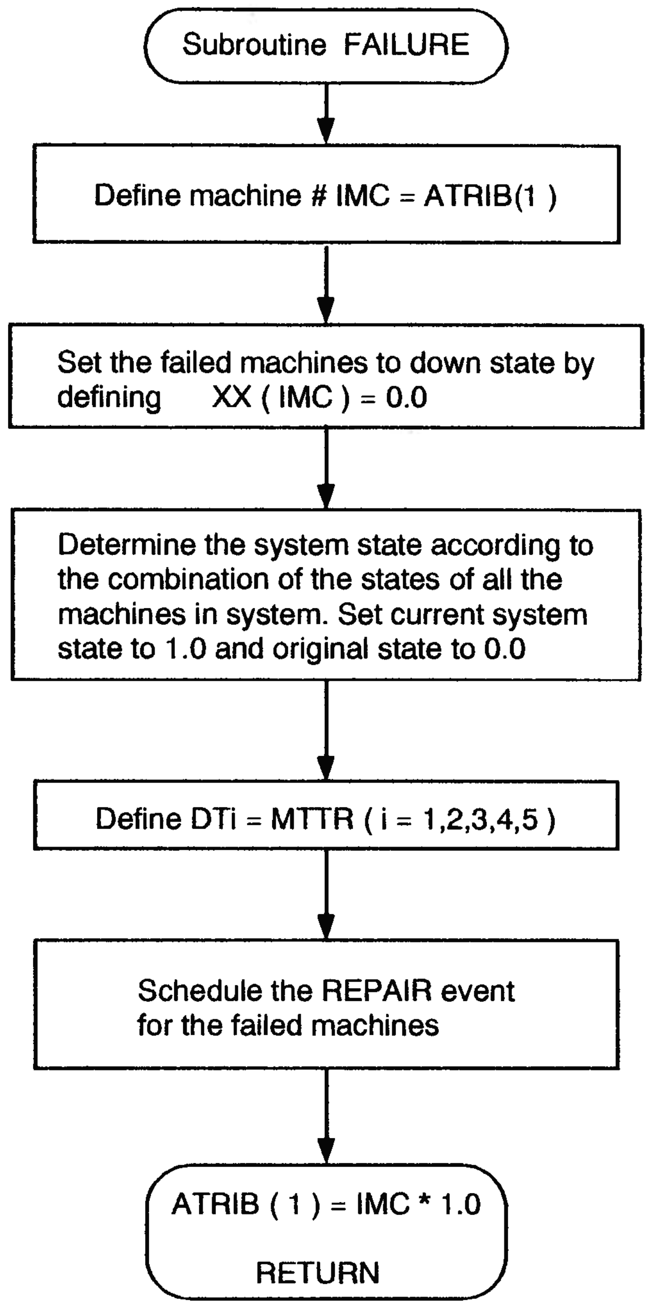

Figure 5-2 Flowchart of SUBROUTINE FAILURE

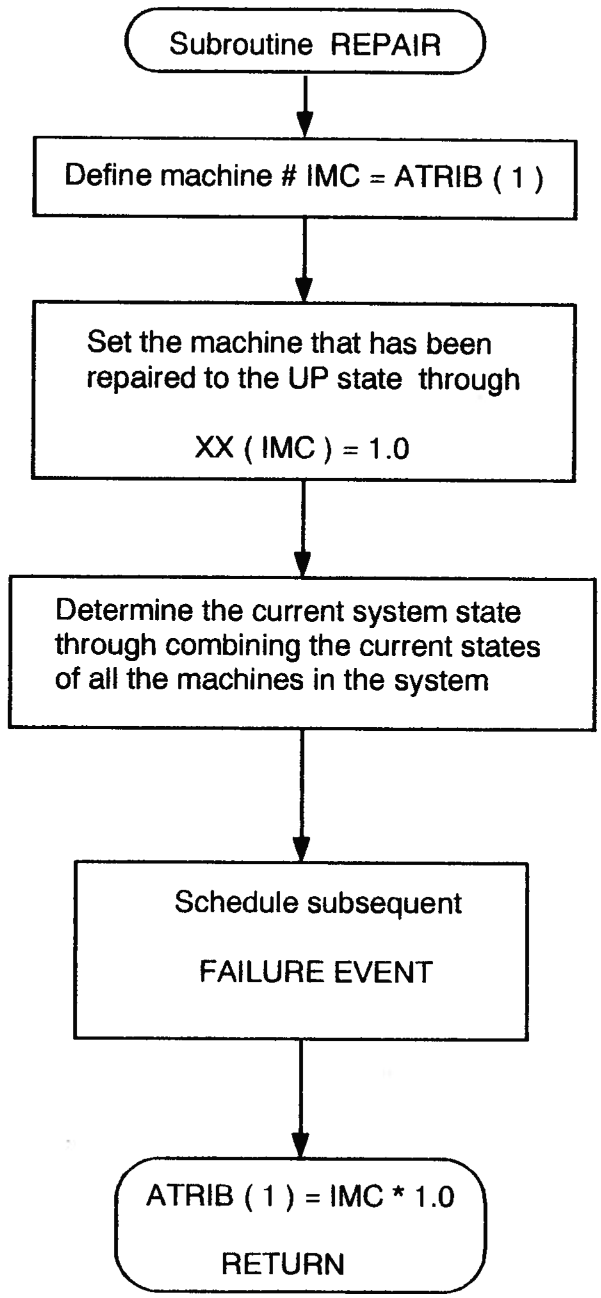

Figure 5-3 Flowchart of SUBROUTINE REPAIR

MC1STPRB , machine 1 state probability

MC2STPRB , machine 2 state probability

MC3STPRB , machine 3 state probability

MC4STPRB , machine 4 state probability

MC5STPRB , machine 5 state probability

MCSTPRB1 , system state probability of state 1

MCSTPRB2 , system state probability of state 2

•

•

MCSTPRB*i* , system state probability of state *i*

•

MCSTPRB15 , system state probability of state 15

MCSTPRB16 , system state probability of state 16

5.2 The Material Handling Subsystem

The SLAM II simulation for the material handling system (subsystem 1) is similar to the one for machining system (subsystem 2). The simulation for subsystem 1 is simpler because the AGV system has only two components AGV1 and AGV2 and the total system states are four. The SLAM II SUMMRY REPORT that shows the state probabilities of AGVs and subsystem 1 is given in Figure 5-5. Where,

AGV1STPRB , AGV1 state probability

AGV2STPRB , AGV2 state probability

AGVSTPRB1 , AGV system state probability of state 1

AGVSTPRB2 , AGV system state probability of state 2

S L A M   I I   S U M M A R Y   R E P O R T

SIMULATION PROJECT: PROCESSING SUBSYSTEM STATE PROBABILITY

DATE 5/25/1994   BY FUHONG DAI   RUN NUMBER   1 OF   1

CURRENT TIME   .5000E+05
STATISTICAL ARRAYS CLEARED AT TIME   .0000E+00

STATISTICS FOR TIME-PERSISTENT VARIABLES

	MEAN VALUE	STANDARD DEVIATION	MINIMUM VALUE	MAXIMUM VALUE	TIME INTERVAL	CURRENT VALUE
MC1STPRB	.990	.101	.00	1.00	50000.000	1.00
MC2STPRB	.991	.096	.00	1.00	50000.000	1.00
MC3STPRB	.991	.095	.00	1.00	50000.000	1.00
MC4STPRB	.991	.096	.00	1.00	50000.000	1.00
MC5STPRB	.990	.099	.00	1.00	50000.000	1.00
MCSTPRB1	.953	.211	.00	1.00	50000.000	1.00
MCSTPRB2	.010	.100	.00	1.00	50000.000	.00
MCSTPRB3	.009	.095	.00	1.00	50000.000	.00
MCSTPRB4	.009	.094	.00	1.00	50000.000	.00
MCSTPRB5	.009	.095	.00	1.00	50000.000	.00
MCSTPRB6	.010	.098	.00	1.00	50000.000	.00
MCSTPRB7	.000	.009	.00	1.00	50000.000	.00
MCSTPRB8	.000	.008	.00	1.00	50000.000	.00
MCSTPRB9	.000	.009	.00	1.00	50000.000	.00
MCSTPRB10	.000	.010	.00	1.00	50000.000	.00
MCSTPRB11	.000	.012	.00	1.00	50000.000	.00
MCSTPRB12	.000	.012	.00	1.00	50000.000	.00
MCSTPRB13	.000	.010	.00	1.00	50000.000	.00
MCSTPRB14	.000	.010	.00	1.00	50000.000	.00
MCSTPRB15	.000	.006	.00	1.00	50000.000	.00
MCSTPRB16	.000	.014	.00	1.00	50000.000	.00

Figure 5-4 SLAM II sumary report of processing subsystem

S L A M   I I   S U M M A R Y   R E P O R T

SIMULATION PROJECT: MATERIAL HANDLING SUBSYSTEM STATE PROBABILITY

DATE 5/25/1994   BY FUHONG DAI   RUN NUMBER   1 OF 1

CURRENT TIME   .5000E+05
STATISTICAL ARRAYS CLEARED AT TIME   .0000E+00

STATISTICS FOR TIME-PERSISTENT VARIABLES

	MEAN VALUE	STANDARD DEVIATION	MINIMUM VALUE	MAXIMUM VALUE	TIME INTERVAL	CURRENT VALUE
AGV1STPRB	.990	.100	.00	1.00	50000.000	1.00
AGV2STPRB	.990	.097	.00	1.00	50000.000	1.00
AGVSTPRB1	.981	.138	.00	1.00	50000.000	1.00
AGVSTPRB2	.010	.099	.00	1.00	50000.000	.00
AGVSTPRB3	.009	.097	.00	1.00	50000.000	.00
AGVSTPRB4	.000	.008	.00	1.00	50000.000	.00

Figure 5-5. SLAM II summary report of material handling subsystem.

AGVSTPRB3 , AGV system state probability of state 3

AGVSTPRB4 , AGV system state probability of state 4

The comparisons and conclusions are presented in chapter 7.0.

6.0 PERFORMANCE SIMULATION OF THE FMS

In this chapter, we present the performance simulation of the FMS to see the operations of the the system. Based on the system as shown in Figure 2-1, we can construct a computer model of the system. The jobs first arrive at the load/unload (L /U) station. A job type (JT) will be defined for each job. The job type determines :

(1) The number of machines to be visited

(2) The index number (IDX) of the machine to be visited, i.e. which machine will be visited first, second and third etc.

(3) The processing time of the job on each machine

In this study, for the purpose of demonstration, we arbitrarily define five job types. Table 6-1 shows the routings of the job types defined and Table 6-2 gives the processing time of each job type on the corresponding machines defined. In the simulation program , we use two dimensional arrays to represent :

(1) The machines to be visited, MTV (JT , IDX)

(2) The processing time, PST (JT , IDX)

The travel time (TRT) of AGV is given in Table 6-3 in the matrix form. A two dimensional array TRT (i , j) is used to represent the travel time of AGV, where i , j = 1,2,3,4,5,6,7,8. Constant processing times(minutes) are assumed for each job.

In this study, two AGVs are employed in the material handling system. Whenever an AGV is requested, always the nearest available one is called. The simulation starts from the SUBROUTINE INTLC and the simulation model is built up in this subroutine by defining two AGVs, five machines and one load / unload station. Each machine has an input buffer to store the parts waiting for processing and an output buffer to store the parts waiting to be

Job type \ Index No.	0	1	2	3	4	5	6
1	L	MC1	MC5	MC3	MC2	U	0
2	L	MC2	MC3	MC5	MC4	U	0
3	L	MC3	MC4	MC1	U	0	0
4	L	MC4	MC5	MC1	U	0	0
5	L	MC5	MC1	MC4	MC2	MC3	U

Table 6-1. Routings of the job types defined.

Job type \ Index No.	0	1	2	3	4	5	6
1	0.	20.	20.	30.	25.	0.	0.
2	0.	30.	20.	25.	20.	0.	0.
3	0.	25.	30.	25	0.	0.	0.
4	0.	25.	20.	30.	0.	0.	0.
5	0.	20.	20.	25.	20.	25.	0.

Table 6-2. Processing time of each job type
at the coresponding machines defined.

Location No.		To (j)							
		1	2	3	4	5	6	7	8
From (i)	1	0.0	4.0	4.0	4.0	2.0	2.0	1.0	3.0
	2	4.0	0.0	2.0	2.0	4.0	4.0	3.0	1.0
	3	4.0	2.0	0.0	2.0	4.0	4.0	3.0	1.0
	4	4.0	2.0	2.0	0.0	4.0	4.0	3.0	1.0
	5	2.0	4.0	4.0	4.0	0.0	2.0	1.0	3.0
	6	2.0	4.0	4.0	4.0	2.0	0.0	1.0	3.0
	7	1.0	3.0	3.0	3.0	1.0	1.0	0.0	2.0
	8	3.0	1.0	1.0	1.0	3.0	3.0	2.0	0.0

Table 6-3. AGV travel time between any two locations.

taken away. All the initial conditions of the AGVs and machines are also defined in this subroutine. After defining the travel time (TRT), machine to be visited (MTV) and the the processing time (PST) , the initial creation of the job will be scheduled by calling the SLAM II SUBROUTINE SCHDL (1 , 0.0 , ATRIB). Attribute 3 will be used to define the index number of the first machine that will be visited at the beginning. The flowchart of SUBROUTINE INTLC is shown in Figure 6-1. The subsequent events of the simulation are

(1) Generating job type (GNRJT)

(2) Despatching AGV (DSPAGV)

(3) Processing event (PROCS)

(4) Breaking down event (BRKDN) of the machines or AGVs and

(5) Repairing event (REPAIR) of the machines and AGVs

The GNRJT subroutine generates the subsequent jobs with five different job types defined uniformly by calling SLAM II's SUBROUTINE SCHDL (EVENT , DTIME , ATRIB) , where

EVENT = 1 , the event code and

DTIME = UNFRM (20.0 , 30.0) , the time increment

Whenever SUBROUTINE GNRJT is called, the input buffer of the machine to be visited by the job generated will be checked first. If the number of the parts waiting at the buffer is less than ten , the job generated will be sent to this buffer by calling the nearest available AGV. Otherwise the job generated will wait at the L/U station by storing the job in a storage area identified in SLAM II as file 12. To send the job to the machine assigned, event 2, dispatching AGV will be scheduled by calling SUBROUTINE SCHDL (2 , DTIME , ATRIB). The logic of SUBROUTINE GNRJT is shown in Figure 6-2 .

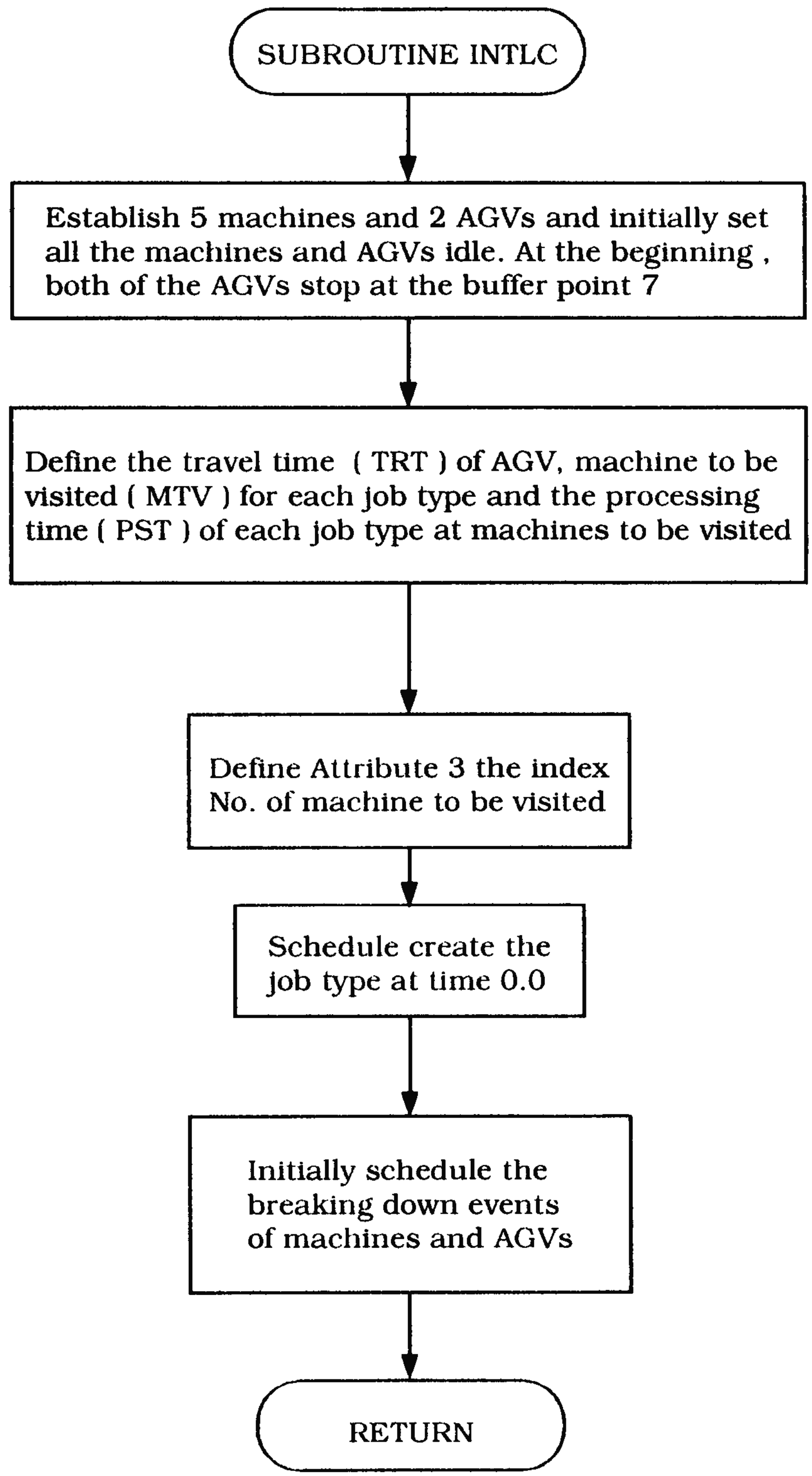

Figure 6-1 Flowchart of SUBROUTINE INTLC

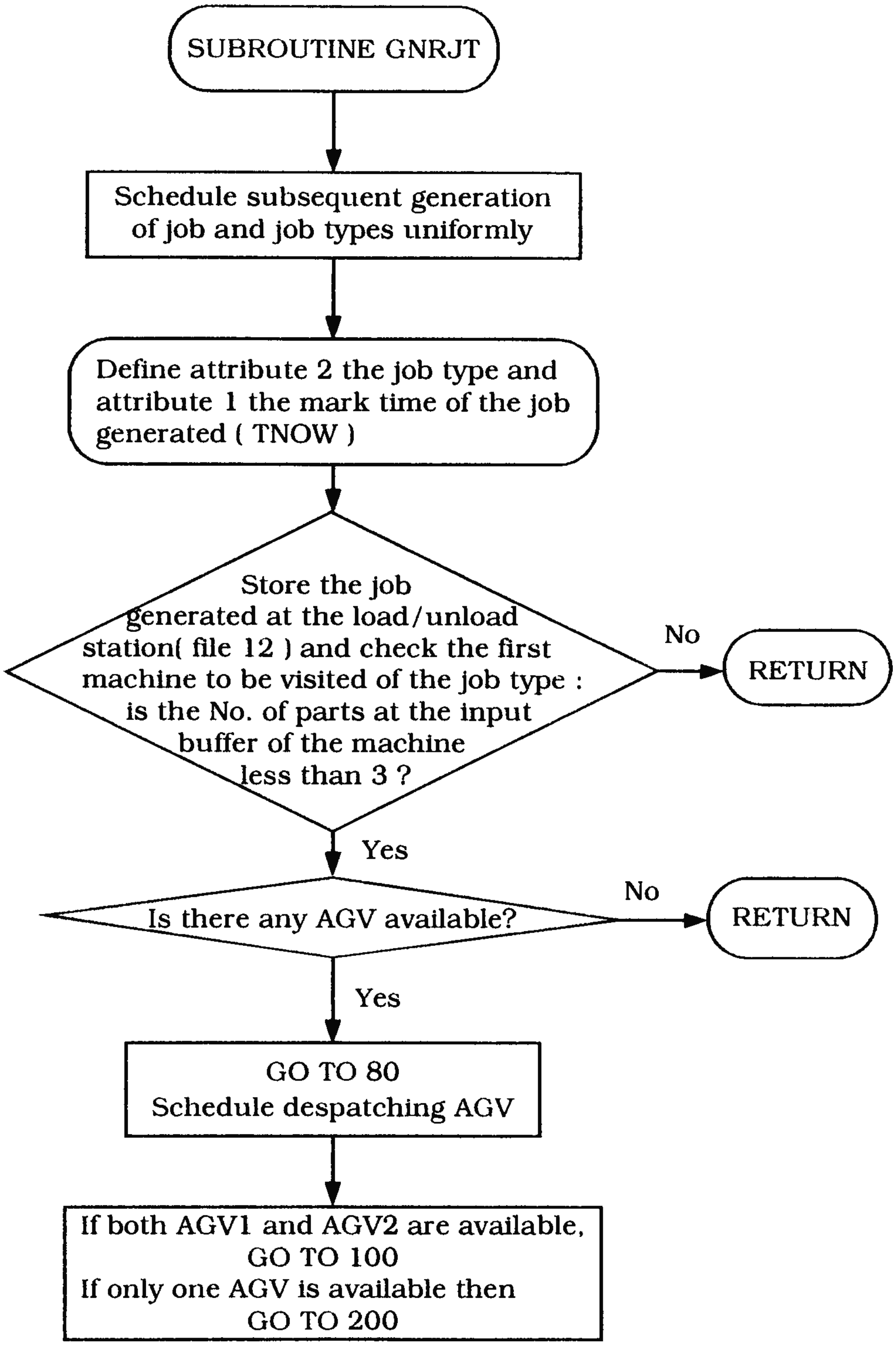

Figure 6-2 Flowchart of SUBROUTINE GNRJB (continued on next page)

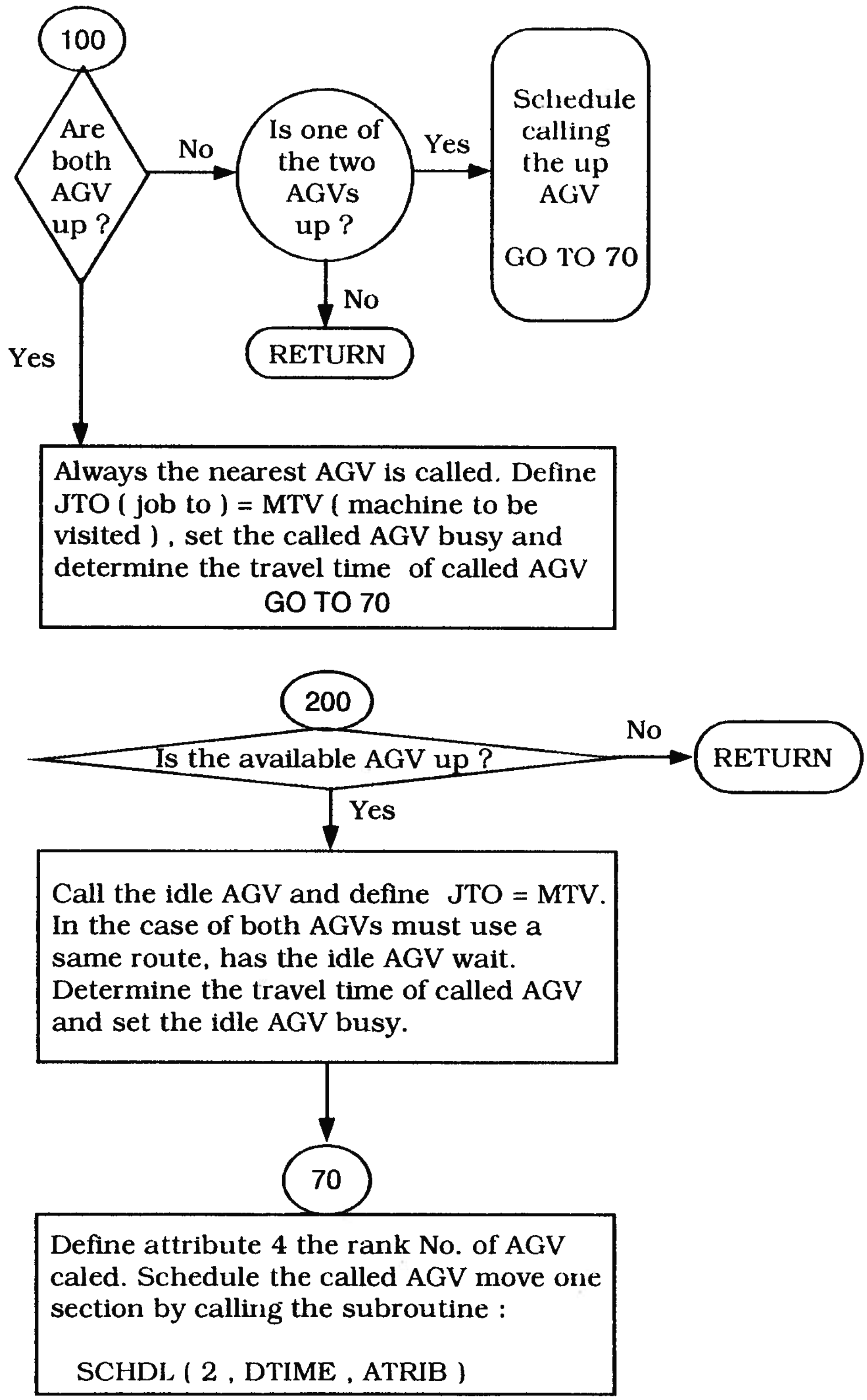

Figure 6-2 Continued

The event 2 is employed to schedule dispatching the busy AGVs, i. e. the AGV is either on the way to the calling point (the AGV is empty) or to the destination of the job of which the AGV is requested (the AGV is loaded with the job). The busy AGV will be scheduled to move section by section and the impending route of the scheduled AGV will always be checked whether the route is occupied by another busy AGV. If the route is occupied, a waiting time (which is equal to the travelling time of the AGV through the route) will be added to the travel time of the scheduled AGV. When the scheduled AGV arrives to its destination and the destination is a machine , the job will be unloaded from the AGV and the processing event will be scheduled if the machine is idle. In case the machine is busy , the job will be stored at the input buffer of the machine. If the destination is L / U station, it means that the job has completed all its processings . After unloading the job at the L / U station, the job leaves the system. The flow chart of SUBROUTINE DSPAGV is shown in Figure 6-3.

Whenever SUBROUTINE PROCS is called, a job processing has been completed at a machine. The input buffer of that machine will be checked whether there is any job waiting. If a job is waiting for processing, the subsequent processing event will be scheduled. In case there is no job waiting at the input buffer of that machine, the machine will be set to idle status. Figure 6-4 shows the flow chart of SUBROUTINE PROCS.

Event 4 is the break down event (BRKDN) . Once SUBROUTINE BRKDN is called, a machine or an AGV is down. The operation of a broken down machine or an AGV is halted by removing the processing event or the AGV dispatching event from the file of event calendar. The remaining operation time is stored in an array RMT(I) , where I = 1, 2, 3, 4, 5, 6, 7 corresponding to each busy AGV or machine whenever the breaking down event happens. The corresponding repair event will then be scheduled in this subroutine. The flowchart of SUBROUTINE BRKDN is shown in Figure 6-5.

The final event is the repair event (REPAIR). The broken down machine or an AGV is repaired whenever SUBROUTINE REPAIR is called. After finishing the repair work , the broken down machine or AGV will resume work and complete the remaining operation. The subsequent break down event will be scheduled for a repaired machine or AGV. Figure 6-6 shows the flowchart of SUBROUTINE REPAIR.

About a ten week (or 99000.0 minute) simulation was run and the following data were collected,

(1) Statistics for time-persistent variables and

(2) File statistics

The time-persistent variables include the utilization and the proportion of time in " up " state of each machine (machine 1 to machine 5) and each AGV (AGV1 and AGV2). The file statistics shows the queue size of each input buffer and output buffer, where file No. 1,2,3,4 and 5 are input buffers and 7,8,9,10, and 11 the output buffers corresponding to machine 1,2, 3, 4 and 5. The SLAM II periodic summary reports are shown in Figures 6 - 7 and the reports were produced every 33000.0 minutes.

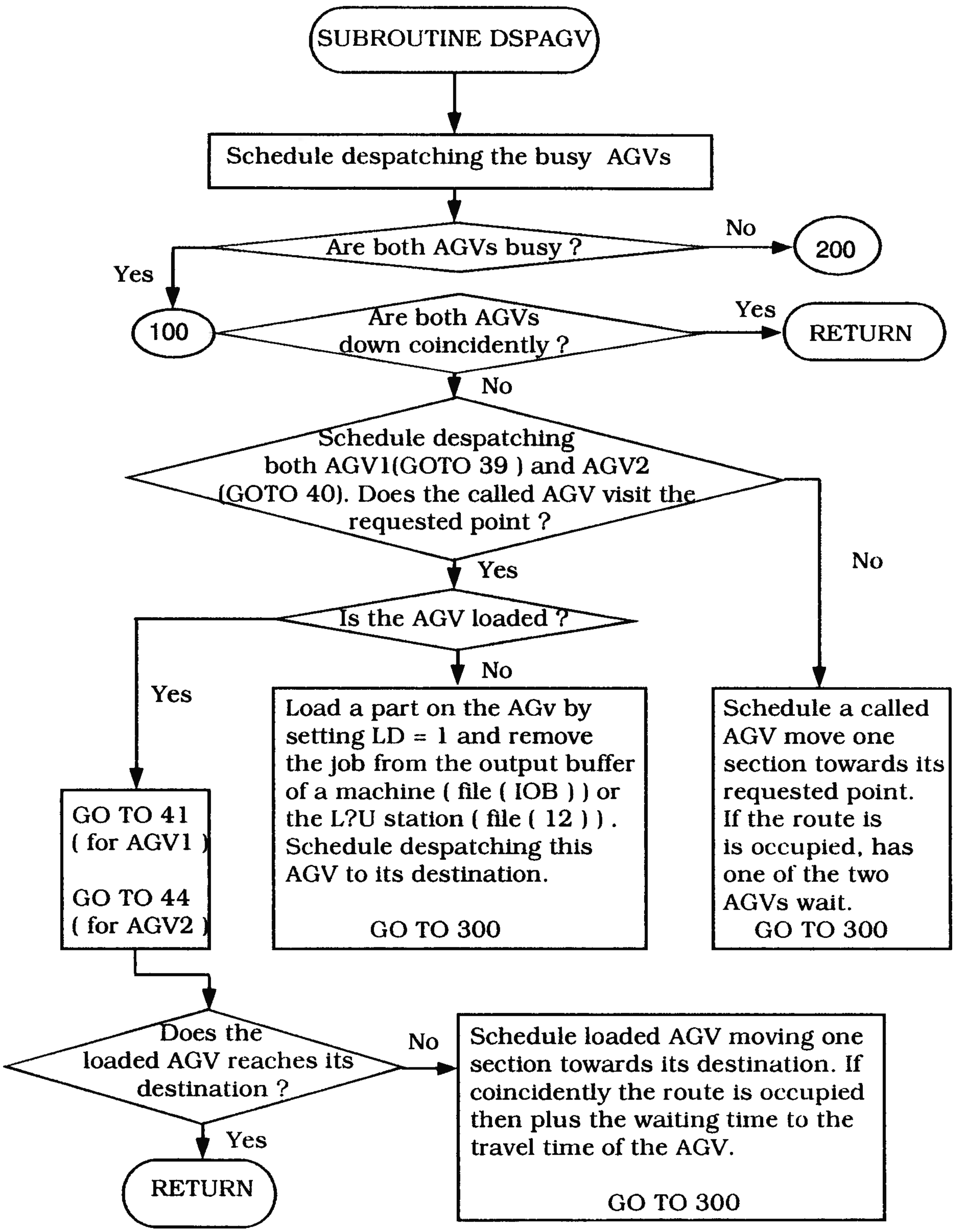

Figure 6-3 Flowchart of SUBROUTINE DSPAGV (continued on next page)

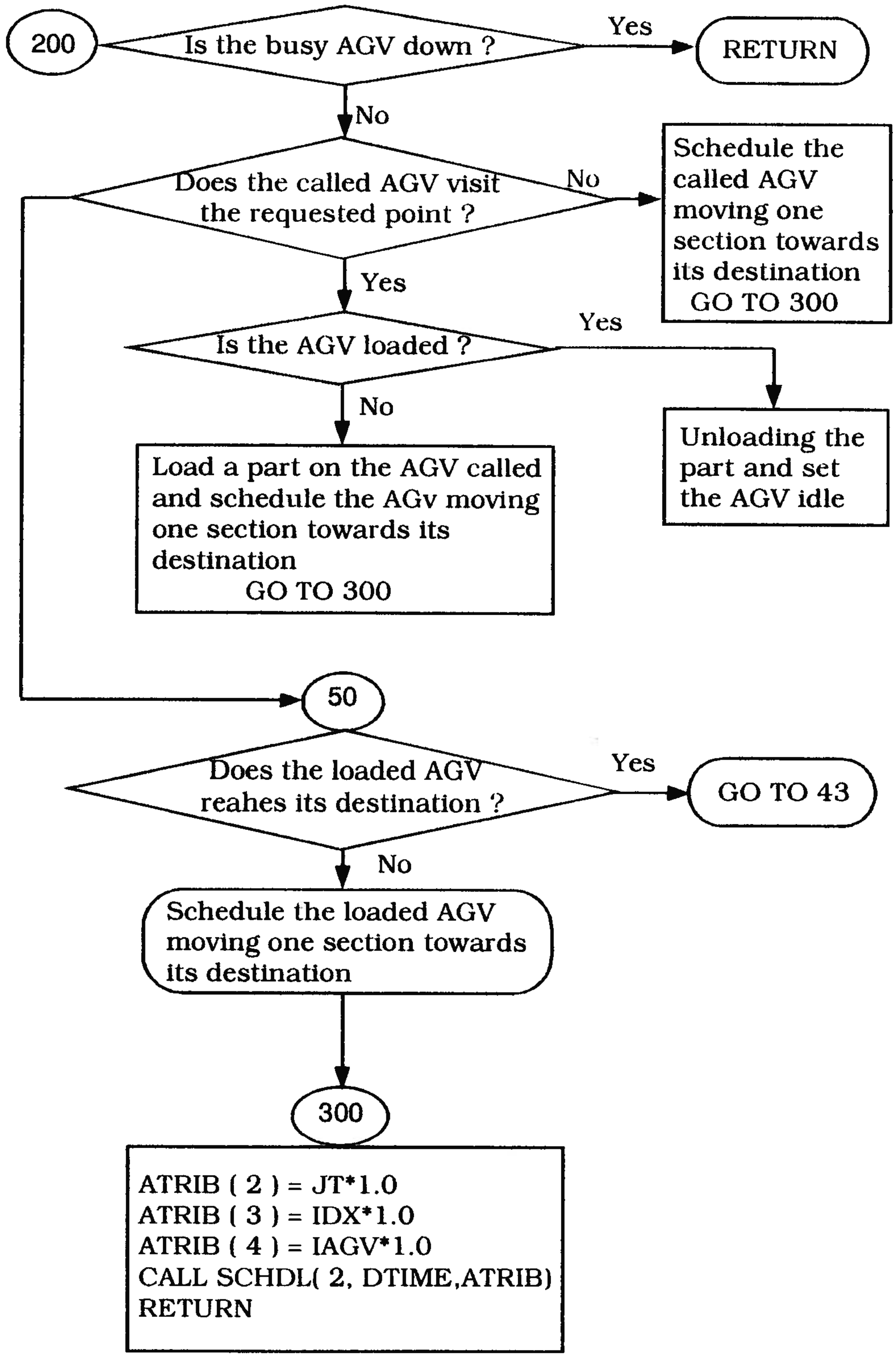

Figure 6-3 Continued

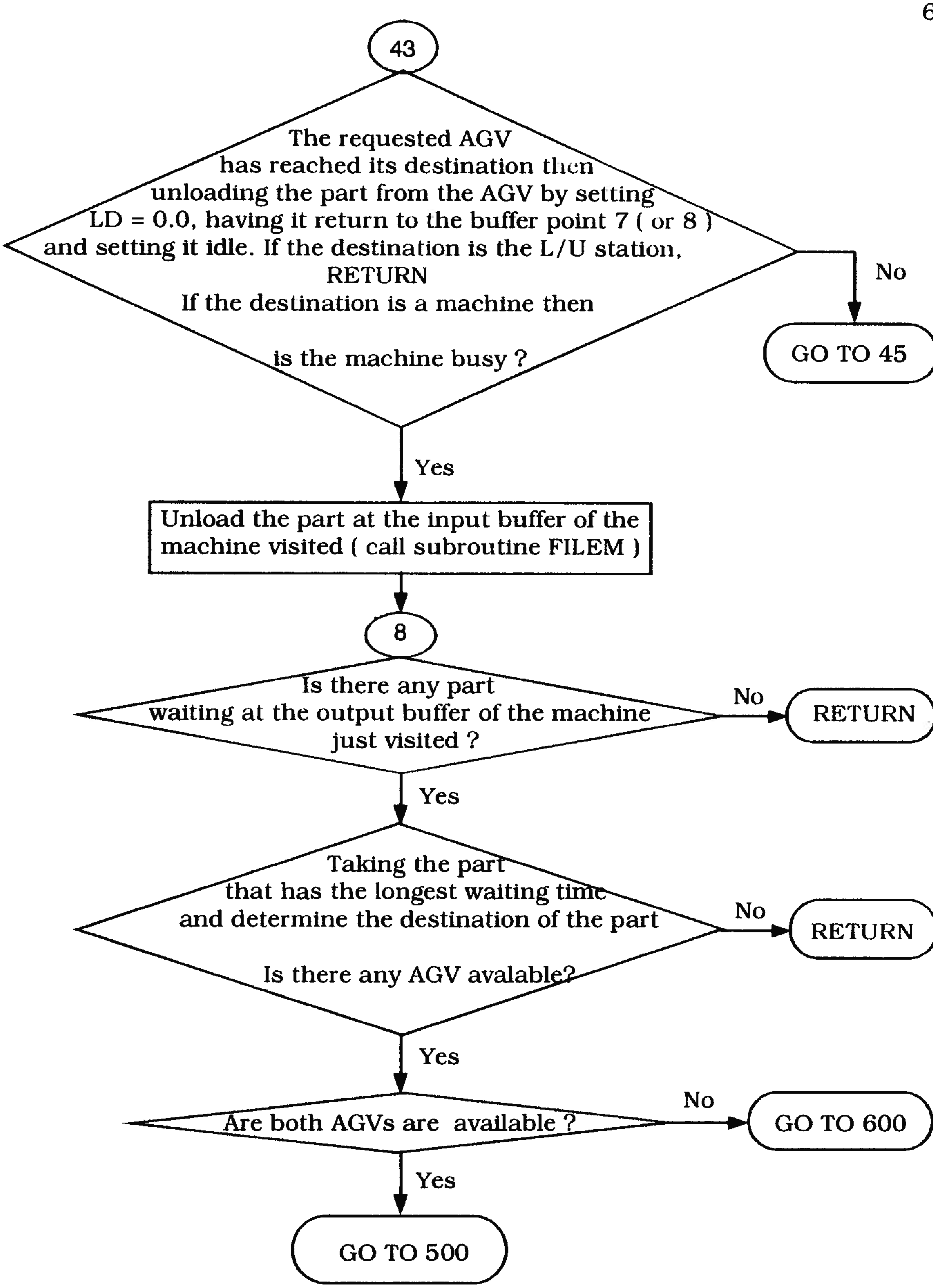

Figure 6-3 Continued

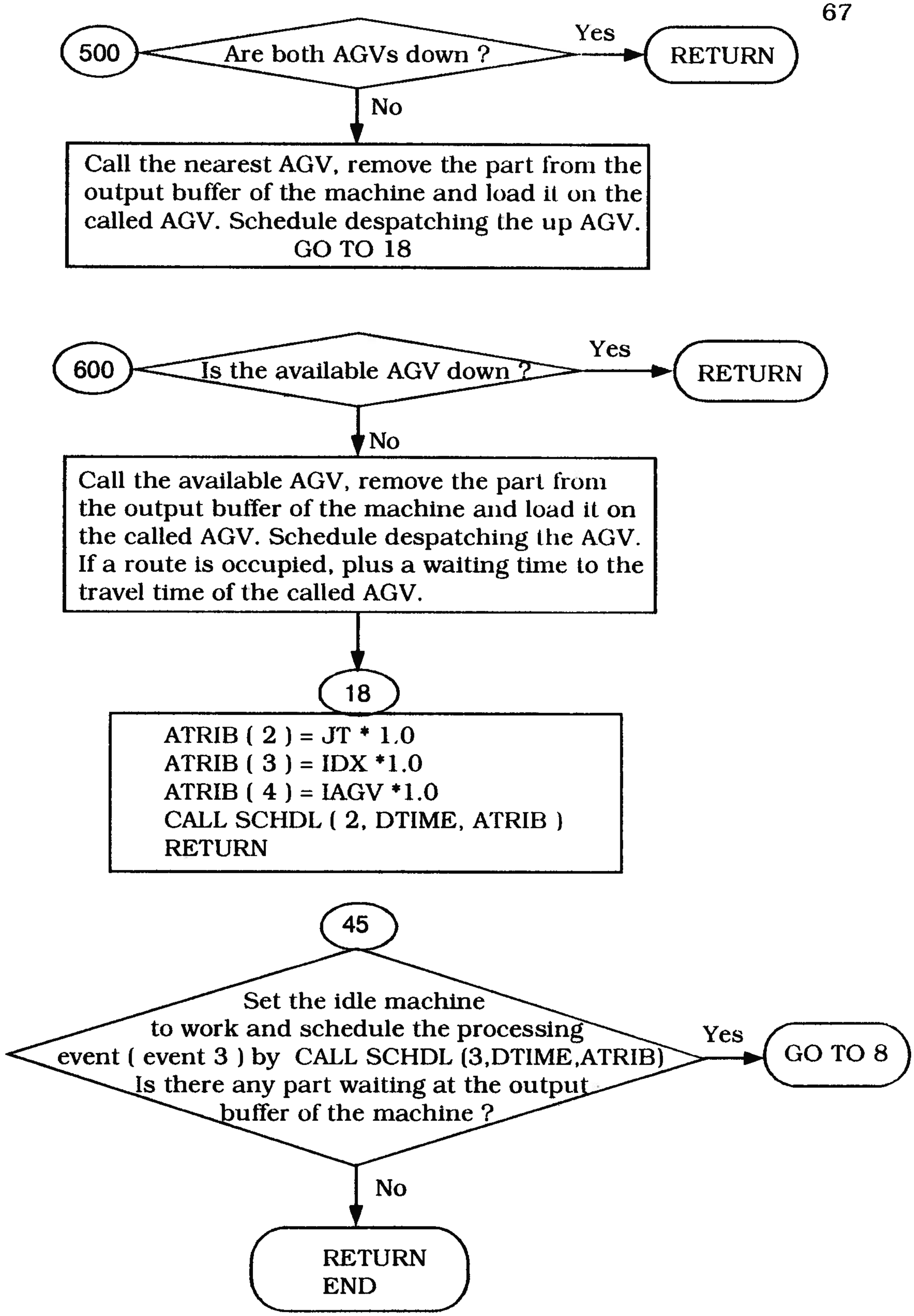

Figure 6-3 Continued

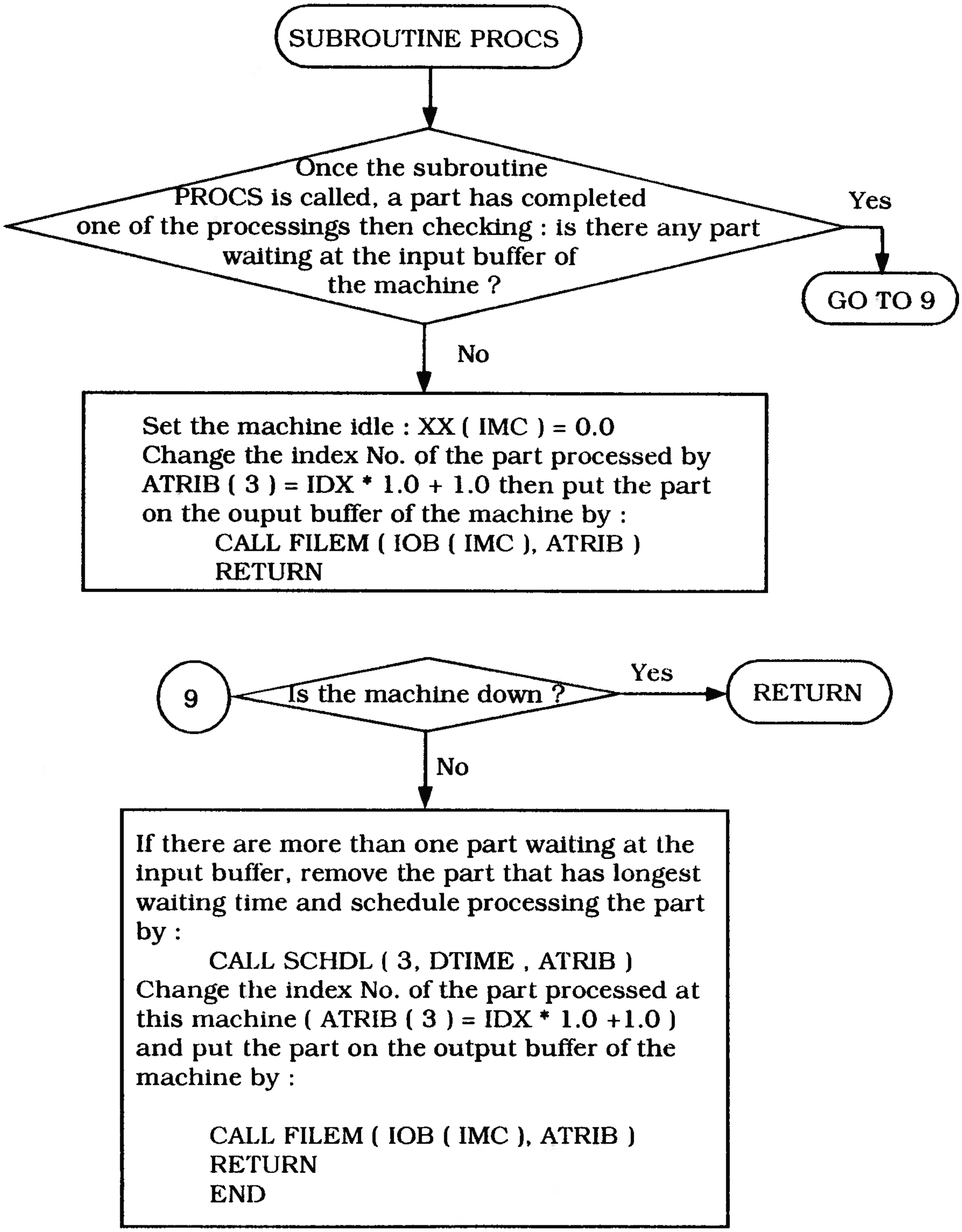

Figure 6-4 Flowchart of SUBROUTINE PROCS

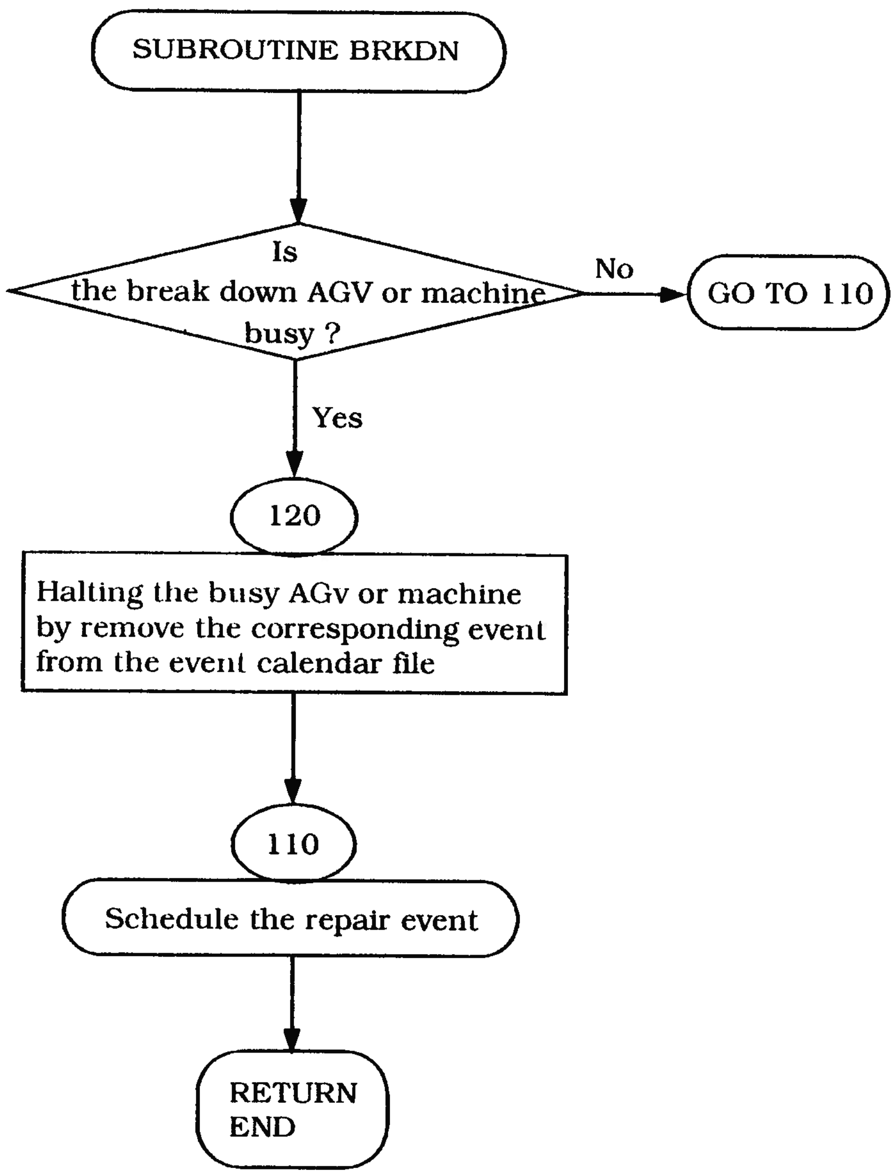

Figure 6-5 Flowchart of SUBROUTINE BRKDN

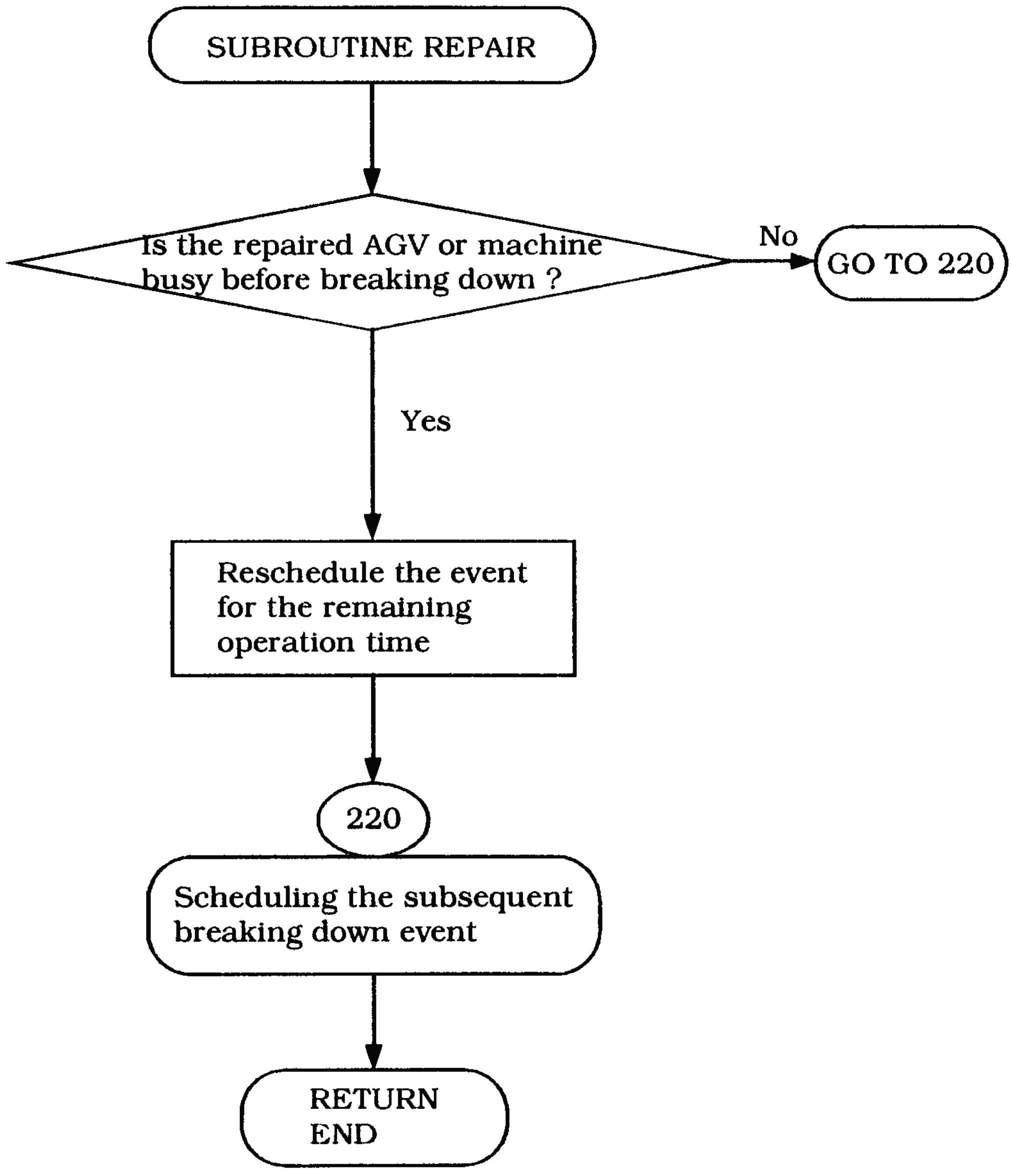

Figure 6-6 Flowchart of SUBROUTINE REPAIR

S L A M   I I   S U M M A R Y   R E P O R T

SIMULATION PROJECT : FMS PERFORMANCE   BY FUHONG DAI

DATE 5/10/1994   RUN NUMBER 1 OF 1

CURRENT TIME .3300E+05
STATISTICAL ARRAYS CLEARED AT TIME .0000E+00

STATISTICS FOR TIME-PERSISTENT VARIABLES

	MEAN VALUE	STANDARD DEVIATION	MINIMUM VALUE	MAXIMUM VALUE	TIME INTERVAL	CURRENT VALUE
UTILIZATION MC1	.485	.500	.00	1.00	33000.000	.00
UTILIZATION MC2	.576	.494	.00	1.00	33000.000	1.00
UTILIZATION MC3	.848	.359	.00	1.00	33000.000	1.00
UTILIZATION MC4	.629	.483	.00	1.00	33000.000	.00
UTILIZATION MC5	.306	.461	.00	1.00	33000.000	.00
UTILIZATION AGV1	.752	.432	.00	1.00	33000.000	1.00
UTILIZATION AGV2	.375	.484	.00	1.00	33000.000	.00
AGV1UPSTATE	.998	.043	.00	1.00	33000.000	1.00
AGV2UPSTATE	.995	.074	.00	1.00	33000.000	1.00
MC1UPSTATE	.997	.052	.00	1.00	33000.000	1.00
MC2UPSTATE	1.000	.000	1.00	1.00	33000.000	1.00
MC3UPSTATE	.994	.078	.00	1.00	33000.000	1.00
MC4UPSTATE	.997	.057	.00	1.00	33000.000	1.00
MC5UPSTATE	1.000	.000	1.00	1.00	33000.000	1.00

FILE STATISTICS

FILE NUMBER	LABEL/TYPE	AVERAGE LENGTH	STANDARD DEVIATION	MAXIMUM LENGTH	CURRENT LENGTH	AVERAGE WAIT TIME
1	IB1	.187	.511	4	0	13.934
2	IB2	.190	.503	5	0	11.677
3	IB3	1.065	1.251	6	1	29.565
4	IB4	.431	.864	6	0	20.241
5	IB5	.053	.240	2	0	9.465
6	UNLOAD	.000	.011	1	0	.433
7	OB1	3.514	1.005	5	5	117.853
8	OB2	3.960	1.428	6	5	109.914
9	OB3	4.554	1.618	7	5	98.095
10	OB4	4.166	1.612	7	5	120.812
11	OB5	2.518	.694	3	3	127.270
12	LOAD	70.500	44.417	150	149	1766.518
13	CALENDAR	12.997	1.584	17	12	13.217

Figure 6-7 SLAM II Summary Report of FMS System Performance
(continued on next page)

S L A M I I S U M M A R Y R E P O R T

SIMULATION PROJECT : FMS PERFORMANCE   BY FUHONG DAI

DATE 5/10/1994   RUN NUMBER 1 OF 1

CURRENT TIME .6600E+05
STATISTICAL ARRAYS CLEARED AT TIME .0000E+00

STATISTICS FOR TIME-PERSISTENT VARIABLES

	MEAN VALUE	STANDARD DEVIATION	MINIMUM VALUE	MAXIMUM VALUE	TIME INTERVAL	CURRENT VALUE
UTILIZATION MC1	.506	.500	.00	1.00	66000.000	1.00
UTILIZATION MC2	.575	.494	.00	1.00	66000.000	.00
UTILIZATION MC3	.856	.351	.00	1.00	66000.000	1.00
UTILIZATION MC4	.635	.481	.00	1.00	66000.000	1.00
UTILIZATION MC5	.304	.460	.00	1.00	66000.000	1.00
UTILIZATION AGV1	.757	.429	.00	1.00	66000.000	1.00
UTILIZATION AGV2	.384	.486	.00	1.00	66000.000	.00
AGV1UPSTATE	.995	.067	.00	1.00	66000.000	1.00
AGV2UPSTATE	.995	.074	.00	1.00	66000.000	1.00
MC1UPSTATE	.996	.064	.00	1.00	66000.000	1.00
MC2UPSTATE	.999	.038	.00	1.00	66000.000	1.00
MC3UPSTATE	.994	.078	.00	1.00	66000.000	1.00
MC4UPSTATE	.998	.040	.00	1.00	66000.000	1.00
MC5UPSTATE	.995	.072	.00	1.00	66000.000	1.00

FILE STATISTICS

FILE NUMBER	LABEL/TYPE	AVERAGE LENGTH	STANDARD DEVIATION	MAXIMUM LENGTH	CURRENT LENGTH	AVERAGE WAIT TIME
1	IB1	.243	.627	6	0	16.027
2	IB2	.185	.481	5	0	11.333
3	IB3	1.257	1.570	9	0	33.572
4	IB4	.451	.946	8	4	21.277
5	IB5	.075	.309	4	0	12.636
6	ULOAD	.000	.012	1	0	.363
7	OB1	4.218	1.333	7	6	135.459
8	OB2	4.603	1.317	6	6	127.538
9	OB3	6.056	2.377	10	9	128.902
10	OB4	4.477	1.946	9	1	128.806
11	OB5	3.184	1.045	5	4	163.040
12	LOAD	146.426	88.025	304	303	3668.993
13	CALENDAR	13.187	1.596	18	14	13.256

Figure 6-7 Continued

S L A M I I S U M M A R Y R E P O R T

SIMULATION PROJECT : FMS PERFORMANCE BY FUHONG DAI

DATE 5/10/1994 RUN NUMBER 1 OF 1

CURRENT TIME .9900E+05
STATISTICAL ARRAYS CLEARED AT TIME .0000E+00

STATISTICS FOR TIME-PERSISTENT VARIABLES

	MEAN VALUE	STANDARD DEVIATION	MINIMUM VALUE	MAXIMUM VALUE	TIME INTERVAL	CURRENT VALUE
UTILIZATION MC1	.509	.500	.00	1.00	99000.000	1.00
UTILIZATION MC2	.570	.495	.00	1.00	99000.000	1.00
UTILIZATION MC3	.848	.359	.00	1.00	99000.000	1.00
UTILIZATION MC4	.637	.481	.00	1.00	99000.000	.00
UTILIZATION MC5	.303	.460	.00	1.00	99000.000	1.00
UTILIZATION AGV1	.758	.428	.00	1.00	99000.000	.00
UTILIZATION AGV2	.381	.486	.00	1.00	99000.000	.00
AGV1UPSTATE	.995	.069	.00	1.00	99000.000	1.00
AGV2UPSTATE	.996	.065	.00	1.00	99000.000	1.00
MC1UPSTATE	.995	.067	.00	1.00	99000.000	1.00
MC2UPSTATE	.999	.031	.00	1.00	99000.000	1.00
MC3UPSTATE	.996	.064	.00	1.00	99000.000	1.00
MC4UPSTATE	.993	.081	.00	1.00	99000.000	1.00
MC5UPSTATE	.997	.059	.00	1.00	99000.000	1.00

FILE STATISTICS

FILE NUMBER	LABEL/TYPE	AVERAGE LENGTH	STANDARD DEVIATION	MAXIMUM LENGTH	CURRENT LENGTH	AVERAGE WAIT TIME
1	IB1	.234	.590	6	1	15.030
2	IB2	.185	.482	5	0	11.458
3	IB3	1.181	1.549	9	2	32.597
4	IB4	.466	.935	8	0	21.852
5	IB5	.069	.291	4	0	11.264
6	UNLOAD	.000	.011	1	0	.352
7	OB1	4.902	1.535	7	5	157.201
8	OB2	4.820	1.211	6	5	134.732
9	OB3	6.750	2.374	10	7	145.248
10	OB4	4.235	1.823	9	6	121.490
11	OB5	3.670	1.147	5	4	187.764
12	LOAD	223.703	133.291	455	454	5602.474
13	CALENDAR	13.280	1.620	18	13	13.384

Figure 6-7 Gontinued

7.0 CONCLUSIONS

Presented in this study was the examination of two different approaches for investigating the reliability (or availability) of a flexible manufacturing system with an AGV based material handling subsystem. The multiplicity of the AGVs, machines, processing routes and sequences, and consequently the states, make the problems of reliability analysis mathematically involved.

To build up an analytical model of the system, the state space approach(Markov processes) was employed. State truncation and state merging techniques enabled adequate simplification of the calculations. A 16×16 system of differential equations for the processing subsystem and a 4×4 system of differential equations for the material handling subsystem were solved for state probabilities i.e. the availabilities and unavailabilities, using Mathcad. The SLAM II simulation model built was based on the basic repairable components of the system and that two status transition modes were assumed. Different failure and repair rates were examined for both simulation and mathematical analysis. The simulation result of the availabilities and unavailabilities were very close to the analytical ones. Compared to the analytical model, the simulation procedure was easier to be built up by carefully defining the variables of every state of the system. For the analytical case, if based on the status of basic components of the system to carry out the calculations, several matrices must be composed and solved. This becomes much tedious if more system components are involved and larger matrices must be handled.

A performance simulation model was also developed using SLAM II discrete event simulation method. The model was used to examine the system operation. The result is useful for the engineers designing similar flexible manufacturing systems to that shown in Figure 2-1.

REFERENCES

[1] Pourbabai, B. " Analysis and Design of an Integrated Material Handling System " INT. J.PROD. RES. , 1988, VOL. 26 , NO. 7, PP1125-1236.

[2] Noble , J. S. , Tangchoco , J. M. A. " A Framework for Material Handling System Design Justification " INT. J. PROD. RES. , 1993, VOL. 31 NO. 1, PP81-106.

[3] Shang, J. S. , Tadikamalla, P. R. " Output Maximization of a CIM System : Simulation and Statistical Approach " INT. J. PROD. RES. , 1993, VOL. 31, No. 1, PP19-41.

[4] Dhillon, B. S. , Singh , C. " *Engineering Reliability, New Techniques and Applications*" 1981; Wiley.

[5] Singh, C. , Billinton, R. "*System Reliability Modelling and Evaluation*" 1977; Hutchinson of London.

[6] Dhillon, B. S. , Rayapati, S. N. " Models to Evaluate Reliability and Availability of Transportation systems ", Proceedings of the IASTED International Symposium on Modelling and Simulation, Lugano, Switzerland, 1991, PP306-309.

[7] Srinivasan , S. K. , Mehat , K. M. " Stochastic Processes " 1978 , McGRAW-HILL Book Company.

[8] Pritsker, A. A. B. " Introduction to simulation and SLAM II " 1984 , Halsted Press

[9] Banks, J. " The Simulation of Material Handling Systems " SIMULATION, NOV. 1990, PP261-270 .

[10] Leung, L. C. , Maheshwari, S. K. , Miller, W. A. " Concurrent Part Assignment and Tool Allocation in FMS with Material Handling Considerations" *INT. J. PROD. RES.*, 1993 vol. 31, No.1, PP117-138 .

[11] Mahadevan , B. , Narendran, T. T. " Design of an Automated Guided Vehicle-based Material Handling System for a Flexible Manufacturing System" *INT. J. PROD. RES.* , 1990, VOL 28, NO. 9, PP1611-1622.

[12] Egbelu, P. J. , and Tanchoco , J. M. A. , 1986 , " Potentials for bi-directional guide Path for Automated Guided Vehicle Based Systems" INT. J. PROD. RES. , 1986, VOL 24 PP 1075-1098.

[13] Yim, D. S. , Linn, R. J. " Push and Pull Rules for Dispatching Automated Guided Viehicles in a Flexible Manufacturing System " INT. J. PROD. RES. , 1993, VOL. 31, No. 1, PP43-57.

[14] " *Mathcad 3.1 , User's Guide* " 1992, MathSoft Inc.

Appendix I

MATHCAD Output of State Probability Calculations

of Material Handling Subsystem

Input the failure rate and repair rate

ORIGIN := 1 TOL := 0.0000001 $\lambda := \frac{1}{100}$ $\mu := 1.0$ $i := 1..4$ $j := 1..4$

Input the transition rates inmatrix form

$$R := \begin{bmatrix} -2\cdot\lambda & \lambda & \lambda & 0 \\ \mu & -(\lambda+\mu) & 0 & \lambda \\ \mu & 0 & -(\lambda+\mu) & \lambda \\ 0 & \mu & \mu & -2\cdot\mu \end{bmatrix}$$

Calculate the eigenvalues of matrix R and generate the diagnal matrix D at time t=5000.0

$r :=$ eigenvals (R) $t := 5000.0$ and

$D_{(i,i)} := \exp\left[(r_i)\cdot t\right]$

Calculate the eigenvectors and generate the matrix S :

$S1 := \text{eigenvec}(R, r_1)$ $S_{(i,1)} := S1_i$

$S2 := \text{eigenvec}(R, r_2)$ $S_{(i,2)} := S2_i$

$S3 := \text{eigenvec}(R, r_3)$ $S_{(i,3)} := S3_i$

$S4 := \text{eigenvec}(R, r_4)$ $S_{(i,4)} := S4_i$

Calculate the state probability matrix :

$$P := S\cdot D\cdot S^{-1}$$

Given the initial state by a row vector as

$$P0 := (1 \quad 0 \quad 0 \quad 0)$$

The state probabilities can be calculated by :

$$SP := P0\cdot P$$

The state probabilities of the material handling subsystem at time " t " with the initial

condition of P0 is as follow :

$$SP = (0.98 \quad 0.01 \quad 0.01 \quad 9.803\cdot 10^{-5})$$

Appendix II

MATHCAD Output of State Probability Calculations

of the Processing Subsystem

Input the failure rate and repair rate

$\lambda 1 := \frac{1}{80}$ $\lambda 2 := \frac{1}{90}$ $\lambda 3 := \frac{1}{100}$ $\lambda 4 := \frac{1}{110}$ $\lambda 5 := \frac{1}{120}$

$\mu 1 := \frac{1}{0.8}$ $\mu 2 := \frac{1}{0.9}$ $\mu 3 := \frac{1}{1.0}$ $\mu 4 := \frac{1}{1.1}$ $\mu 5 := \frac{1}{1.2}$

ORIGIN := 1 TOL := 0.0000001 i := 1.. 16 j := 1.. 16

Construct the transition rate matrix R and first define the diagnal elements :

$R01 := -(\lambda 1 + \lambda 2 + \lambda 3 + \lambda 4 + \lambda 5)$

$R22 := -(\mu 1 + \lambda 2 + \lambda 3 + \lambda 4 + \lambda 5)$

$R33 := -(\lambda 1 + \mu 2 + \lambda 3 + \lambda 4 + \lambda 5)$

$R44 := -(\lambda 1 + \lambda 2 + \mu 3 + \lambda 4 + \lambda 5)$

$R55 := -(\lambda 1 + \lambda 2 + \lambda 3 + \mu 4 + \lambda 5)$

$R66 := -(\lambda 1 + \lambda 2 + \lambda 3 + \lambda 4 + \mu 5)$

$R77 := -(\mu 1 + \mu 2)$

$R88 := -(\mu 1 + \mu 3)$

$R99 := -(\mu 1 + \mu 4)$

$R10 := -(\mu 1 + \mu 5)$

$R11 := -(\mu 2 + \mu 3)$

$R12 := -(\mu 2 + \mu 4)$

$R13 := -(\mu 3 + \mu 5)$

$R14 := -(\mu 3 + \mu 4)$

$R15 := -(\mu 3 + \mu 5)$

$R16 := -(\mu 4 + \mu 5)$

$$R1 := \begin{bmatrix} R01 & \lambda 1 & \lambda 2 & \lambda 3 & \lambda 4 & \lambda 5 \\ \mu 1 & R22 & 0. & 0. & 0. & 0. \\ \mu 2 & 0. & R33 & 0. & 0. & 0. \\ \mu 3 & 0. & 0. & R44 & 0. & 0. \\ \mu 4 & 0. & 0. & 0. & R55 & 0. \\ \mu 5 & 0. & 0. & 0. & 0. & R66 \\ 0. & \mu 2 & \mu 1 & 0. & 0. & 0. \\ 0. & \mu 3 & 0. & \mu 1 & 0. & 0. \\ 0. & \mu 4 & 0. & 0. & \mu 1 & 0. \\ 0. & \mu 5 & 0. & 0. & 0. & \mu 1 \\ 0. & 0. & \mu 3 & \mu 4 & 0. & 0. \\ 0. & 0. & \mu 4 & 0. & \mu 2 & 0. \\ 0. & 0. & \mu 5 & 0. & 0. & \mu 2 \\ 0. & 0. & 0. & \mu 4 & \mu 3 & 0. \\ 0. & 0. & 0. & \mu 5 & 0. & \mu 3 \\ 0. & 0. & 0. & 0. & \mu 5 & \mu 4 \end{bmatrix} \qquad R2 := \begin{bmatrix} 0. & 0. & 0. & 0. & 0. \\ \lambda 2 & \lambda 3 & \lambda 4 & \lambda 5 & 0. \\ \lambda 1 & 0. & 0. & 0. & \lambda 3 \\ 0. & \lambda 1 & 0. & 0. & \lambda 2 \\ 0. & 0. & \lambda 1 & 0. & 0. \\ 0. & 0. & 0. & \lambda 1 & 0. \\ R77 & 0. & 0. & 0. & 0. \\ 0. & R88 & 0. & 0. & 0. \\ 0. & 0. & R99 & 0. & 0. \\ 0. & 0. & 0. & R10 & 0. \\ 0. & 0. & 0. & 0. & R11 \\ 0. & 0. & 0. & 0. & 0. \\ 0. & 0. & 0. & 0. & 0. \\ 0. & 0. & 0. & 0. & 0. \\ 0. & 0. & 0. & 0. & 0. \\ 0. & 0. & 0. & 0. & 0. \end{bmatrix}$$

$$R3 := \begin{bmatrix} 0. & 0. & 0. & 0. & 0. \\ 0. & 0. & 0. & 0. & 0. \\ \lambda 4 & \lambda 5 & 0. & 0. & 0. \\ 0. & 0. & \lambda 4 & \lambda 5 & 0. \\ \lambda 2 & 0. & \lambda 3 & 0. & \lambda 5 \\ 0. & \lambda 2 & 0. & \lambda 3 & \lambda 4 \\ 0. & 0. & 0. & 0. & 0. \\ 0. & 0. & 0. & 0. & 0. \\ 0. & 0. & 0. & 0. & 0. \\ 0. & 0. & 0. & 0. & 0. \\ 0. & 0. & 0. & 0. & 0. \\ R12 & 0. & 0. & 0. & 0. \\ 0. & R13 & 0. & 0. & 0. \\ 0. & 0. & R14 & 0. & 0. \\ 0. & 0. & 0. & R15 & 0. \\ 0. & 0. & 0. & 0. & R16 \end{bmatrix}$$

Augmenting the three matrices R1 ,R2 and R3 :

R4 := augment (R1 ,R2)

R := augment(R4,R3)

$$R = \begin{bmatrix} -0.051 & 0.013 & 0.011 & 0.01 & 0.009 & 0.008 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1.25 & -1.289 & 0 & 0 & 0 & 0 & 0.011 & 0.01 & 0.009 & 0.008 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1.111 & 0 & -1.151 & 0 & 0 & 0 & 0.013 & 0 & 0 & 0 & 0.01 & 0.009 & 0.008 & 0 & 0 & 0 \\ 1 & 0 & 0 & -1.041 & 0 & 0 & 0 & 0.013 & 0 & 0 & 0.011 & 0 & 0 & 0.009 & 0.008 & 0 \\ 0.909 & 0 & 0 & 0 & -0.951 & 0 & 0 & 0 & 0.013 & 0 & 0 & 0.011 & 0 & 0.01 & 0 & 0.008 \\ 0.833 & 0 & 0 & 0 & 0 & -0.876 & 0 & 0 & 0 & 0.013 & 0 & 0 & 0.011 & 0 & 0.01 & 0.009 \\ 0 & 1.111 & 1.25 & 0 & 0 & 0 & -2.361 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1.25 & 0 & 0 & 0 & -2.25 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.909 & 0 & 0 & 1.25 & 0 & 0 & 0 & -2.159 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.833 & 0 & 0 & 0 & 1.25 & 0 & 0 & 0 & -2.083 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0.909 & 0 & 0 & 0 & 0 & 0 & 0 & -2.111 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.909 & 0 & 1.111 & 0 & 0 & 0 & 0 & 0 & 0 & -2.02 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.833 & 0 & 0 & 1.111 & 0 & 0 & 0 & 0 & 0 & 0 & -1.833 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.909 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1.909 & 0 & 0 \\ 0 & 0 & 0 & 0.833 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1.833 & 0 \\ 0 & 0 & 0 & 0 & 0.833 & 0.909 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1.742 \end{bmatrix}$$

Calculate the eigenvalues of matrix R and arbitrarily give the time point t=5000.0 then to generate the diagnal matrix D :

$r := \mathrm{eigenvals}(R)$ $\qquad$ $t := 5000.0$

$D_{(i,i)} := \exp\left[\left(r_i\right)\cdot t\right]$

Calculate the eigenvectors and generate the matrix S :

$S1 := \mathrm{eigenvec}(R, r_1)$	$S5 := \mathrm{eigenvec}(R, r_5)$	$S9 := \mathrm{eigenvec}(R, r_9)$	$S13 := \mathrm{eigenvec}(R, r_{13})$
$S2 := \mathrm{eigenvec}(R, r_2)$	$S6 := \mathrm{eigenvec}(R, r_6)$	$S10 := \mathrm{eigenvec}(R, r_{10})$	$S14 := \mathrm{eigenvec}(R, r_{14})$
$S3 := \mathrm{eigenvec}(R, r_3)$	$S7 := \mathrm{eigenvec}(R, r_7)$	$S11 := \mathrm{eigenvec}(R, r_{11})$	$S15 := \mathrm{eigenvec}(R, r_{15})$
$S4 := \mathrm{eigenvec}(R, r_4)$	$S8 := \mathrm{eigenvec}(R, r_8)$	$S12 := \mathrm{eigenvec}(R, r_{12})$	$S16 := \mathrm{eigenvec}(R, r_{16})$
$S_{(i,1)} := S1_i$	$S_{(i,5)} := S5_i$	$S_{(i,9)} := S9_i$	$S_{(i,13)} := S13_i$
$S_{(i,2)} := S2_i$	$S_{(i,6)} := S6_i$	$S_{(i,10)} := S10_i$	$S_{(i,14)} := S14_i$
$S_{(i,3)} := S3_i$	$S_{(i,7)} := S7_i$	$S_{(i,11)} := S11_i$	$S_{(i,15)} := S15_i$
$S_{(i,4)} := S4_i$	$S_{(i,8)} := S8_i$	$S_{(i,12)} := S12_i$	$S_{(i,16)} := S16_i$

Calculate the state probabiliy matrix using Equation,

$$P := S\cdot D\cdot S^{-1}$$

Given the initial condition by a row vector

$$P0 := (1\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0)$$

The state probabilities at time ' t ' with the initial condition P0 can be calculated through the Equation,

$$SP := P0\cdot P$$

The ouput of the state probabilities given by a colunm vector as follow :

$$SP^{T} = \begin{bmatrix} 0.914 \\ 0.009 \\ 0.009 \\ 0.009 \\ 0.009 \\ 0.009 \\ 9.145 \cdot 10^{-5} \\ 9.132 \cdot 10^{-5} \\ 9.142 \cdot 10^{-5} \\ 9.147 \cdot 10^{-5} \\ 9.135 \cdot 10^{-5} \\ 9.144 \cdot 10^{-5} \\ 9.703 \cdot 10^{-5} \\ 9.134 \cdot 10^{-5} \\ 9.138 \cdot 10^{-5} \\ 9.146 \cdot 10^{-5} \end{bmatrix}$$

The state probabilities of reduced state space of the processing subsystem

Appendix III

Computer Listing of SLAM II Discrete Event Simulation

of State Probabilities of Material Handling Subsystem

```
      SUBROUTINE EVENT(I)
      GO TO (1,2),I
C
C......DEFINE EVENT CODE 1 AS SUBROUTINE "FAILURE"
C
    1 CALL FAILURE
      RETURN
C
C......DEFINE EVENT CODE 2 AS SUBROUTINE " REPAIR "
C
    2 CALL REPAIR
      RETURN
      END
C----------------------------------------------------------------------
      SUBROUTINE INTLC
$INCLUDE:'PARAM.INC'
$INCLUDE:'SCOM1.COM'
C
C......INITIALIZE AGVS TO UP STATE
C
C......AGV1 IS UP
      XX(1)=1.0
C......AGV2 IS UP
      XX(2)=1.0
C......BOTH AGVS ARE UP
      XX(8)=1.0
C......ONLY AGV1 IS DOWN
      XX(9)=0.0
C......ONLY AGV2 IS DOWN
      XX(10)=0.0
C......BOTH AGV1 AND AGV2 ARE DOWN
      XX(11)=0.0
C
C......SCHEDULE EVENT 1 (FAILURE)
C
C......ASSIGN ATTRIBUTE 1  THE AGV #
C
      ATRIB(1)=1.0
      CALL SCHDL(1,EXPON(100.,1),ATRIB)
      ATRIB(1)=2.0
      CALL SCHDL(1,EXPON(100.,1),ATRIB)
C
      RETURN
      END
C-------------------------------------------------------------------------
      SUBROUTINE FAILURE
$INCLUDE:'PARAM.INC'
$INCLUDE:'SCOM1.COM'
C
      IAGV=ATRIB(1)
C
C......SCHEDULE REPAIR EVENT
C
      GO TO(201,202),IAGV
C
  201 XX(1)=0.0
      GO TO 300
  202 XX(2)=0.0
C
  300 XX(8)=0.0
      IF(XX(1).EQ.0.0.AND.XX(2).GT.0.0) THEN
        XX(9)=1.0
```

```
      ELSE
        IF(XX(1).GT.0.0.AND.XX(2).EQ.0.0) THEN
          XX(10)=1.0
        ELSE
          IF(XX(1).LT.1.0.AND.XX(2).LT.1.0) THEN
            XX(11)=1.0
          ENDIF
        ENDIF
      ENDIF
      GO TO(301,302),IAGV
  301 CALL SCHDL(2,EXPON(1.,1),ATRIB)
      GO TO 350
  302 CALL SCHDL(2,EXPON(1.,1),ATRIB)
C
  350 ATRIB(1)=IAGV*1.0
      RETURN
      END
C---------------------------------------------------------------------------
      SUBROUTINE REPAIR
$INCLUDE:'PARAM.INC'
$INCLUDE:'SCOM1.COM'
C
      IAGV=ATRIB(1)
C
      XX(8)=0.0
      XX(9)=0.0
      XX(10)=0.0
      XX(11)=0.0
C
      GO TO(401,402),IAGV
  401 XX(1)=1.0
      GO TO 450
  402 XX(2)=1.0
  450 IF(XX(1).GT.0.0.AND.XX(2).GT.0.0) THEN
        XX(8)=1.0
      ELSE
        IF(XX(1).EQ.0.0.AND.XX(2).GT.0.0) THEN
          XX(9)=1.0
        ELSE
          IF(XX(1).GT.0.0.AND.XX(2).EQ.0.0) THEN
            XX(10)=1.0
          ELSE
            IF(XX(1).EQ.0.0.AND.XX(2).EQ.0.0) THEN
              XX(11)=1.0
            ENDIF
          ENDIF
        ENDIF
      ENDIF
C
C......SCHEDULE SUBSEQUENT FAILURE EVENTS
C
      GO TO(101,102),IAGV
C
  101 CALL SCHDL(1,EXPON(100.,1),ATRIB)
      GO TO 550
  102 CALL SCHDL(1,EXPON(100.,1),ATRIB)
C
  550 ATRIB(1)=IAGV*1.0
      RETURN
      END
```

Appendix IV

Computer Listing of SLAM II Discrete Event Simulation of State Probabilities of the Processing Subsystem

```
      SUBROUTINE EVENT(I)
      GO TO (1,2),I
C
C......DEFINE EVENT CODE 1 AS SUBROUTINE "FAILURE"
C
    1 CALL FAILURE
      RETURN
C
C......DEFINE EVENT CODE 2 AS SUBROUTINE " REPAIR "
C
    2 CALL REPAIR
      RETURN
      END
C---------------------------------------------------------------------
      SUBROUTINE INTLC
$INCLUDE:'PARAM.INC'
$INCLUDE:'SCOM1.COM'
C
C......INITIALIZE MACHINES TO UP STATE
C
C......MC1 IS UP
      XX(3)=1.0
C......MC2 IS UP
      XX(4)=1.0
C......MC3 IS UP
      XX(5)=1.0
C......MC4 IS UP
      XX(6)=1.0
C......MC5 IS UP
      XX(7)=1.0
C......ALL THE MACHINES ARE UP
      XX(21)=1.0
C......ONLY MC1 IS DOWN
      XX(22)=0.0
C......ONLY MC2 IS DOWN
      XX(23)=0.0
C......ONLY MC3 IS DOWN
      XX(24)=0.0
C......ONLY MC4 IS DOWN
      XX(25)=0.0
C......ONLY MC5 IS DOWN
      XX(26)=0.0
C......ONLY MC1 AND MC2 ARE DOWN
      XX(27)=0.0
C......ONLY MC1 AND MC3 ARE DOWN
      XX(28)=0.0
C......ONLY MC1 AND MC4 ARE DOWN
      XX(29)=0.0
C......ONLY MC1 AND MC5 ARE DOWN
      XX(30)=0.0
C......ONLY MC2 AND MC3 ARE DOWN
      XX(31)=0.0
C......ONLY MC2 AND MC4 ARE DOWN
      XX(32)=0.0
C......ONLY MC2 AND MC5 ARE DOWN
      XX(33)=0.0
C......ONLY MC3 AND MC4 ARE DOWN
      XX(34)=0.0
C......ONLY MC3 AND MC5 ARE DOWN
      XX(35)=0.0
C......ONLY MC4 AND MC5 ARE DOWN
      XX(36)=0.0
C
```

```
C......SCHEDULE EVENT 1 (FAILURE)
C
C......ASSIGN ATTRIBUTE 1  THE MACHINE #
C
       ATRIB(1)=1.0
       CALL SCHDL(1,EXPON(80.,1),ATRIB)
       ATRIB(1)=2.0
       CALL SCHDL(1,EXPON(90.,1),ATRIB)
       ATRIB(1)=3.0
       CALL SCHDL(1,EXPON(100.,1),ATRIB)
       ATRIB(1)=4.0
       CALL SCHDL(1,EXPON(110.,1),ATRIB)
       ATRIB(1)=5.0
       CALL SCHDL(1,EXPON(120.,1),ATRIB)
C
       RETURN
       END
C-----------------------------------------------------------------------
       SUBROUTINE FAILURE
$INCLUDE:'PARAM.INC'
$INCLUDE:'SCOM1.COM'
C
       XMC=ATRIB(1)
       IMC=XMC
C
C......SCHEDULE REPAIR EVENT
C
       GO TO(201,202,203,204,205),IMC
C
  201  XX(3)=0.0
       GO TO 300
  202  XX(4)=0.0
       GO TO 300
  203  XX(5)=0.0
       GO TO 300
  204  XX(6)=0.0
       GO TO 300
  205  XX(7)=0.0
C
  300  SUM=XX(3)+XX(4)+XX(5)+XX(6)+XX(7)
       XX(21)=0.0
C......IF ENCOUNTERING STATE 2
       IF(XX(3).EQ.0.0.AND.SUM.EQ.4.0) THEN
         XX(22)=1.0
       ENDIF
C......IF ENCOUNTERING STATE 3
       IF(XX(4).EQ.0.0.AND.SUM.EQ.4.0) THEN
         XX(23)=1.0
       ENDIF
C......IF ENCOUNTERING STATE 4
       IF(XX(5).EQ.0.0.AND.SUM.EQ.4.0) THEN
         XX(24)=1.0
       ENDIF
C......IF ENCOUNTERING STATE 5
       IF(XX(6).EQ.0.0.AND.SUM.EQ.4.0) THEN
         XX(25)=1.0
       ENDIF
C......IF ENCOUNTERING STATE 6
       IF(XX(7).EQ.0.0.AND.SUM.EQ.4.0) THEN
         XX(26)=1.0
       ENDIF
C......IF ENCOUNTERING STATE 7
       IF(SUM.EQ.3.0) THEN
```

```
          IF(XX(3).EQ.0.0.AND.XX(4).EQ.0.0) THEN
            XX(27)=1.0
          ENDIF
C......IF ENCOUNTERING STATE 8
          IF(XX(3).EQ.0.0.AND.XX(5).EQ.0.0) THEN
            XX(28)=1.0
          ENDIF
C......IF ENCOUNTERING STATE 9
          IF(XX(3).EQ.0.0.AND.XX(6).EQ.0.0) THEN
            XX(29)=1.0
          ENDIF
C......IF ENCOUNTERING STATE 10
          IF(XX(3).EQ.0.0.AND.XX(7).EQ.0.0) THEN
            XX(30)=1.0
          ENDIF
C......IF ENCOUNTERING STATE 11
          IF(XX(4).EQ.0.0.AND.XX(5).EQ.0.0) THEN
            XX(31)=1.0
          ENDIF
C......IF ENCOUNTERING STATE 12
          IF(XX(4).EQ.0.0.AND.XX(6).EQ.0.0) THEN
            XX(32)=1.0
          ENDIF
C......IF ENCOUNTERING STATE 13
          IF(XX(4).EQ.0.0.AND.XX(7).EQ.0.0) THEN
            XX(33)=1.0
          ENDIF
C......IF ENCOUNTERING STATE 14
          IF(XX(5).EQ.0.0.AND.XX(6).EQ.0.0) THEN
            XX(34)=1.0
          ENDIF
C......IF ENCOUNTERING STATE 15
          IF(XX(5).EQ.0.0.AND.XX(7).EQ.0.0) THEN
            XX(35)=1.0
          ENDIF
C......IF ENCOUNTERING STATE 16
          IF(XX(6).EQ.0.0.AND.XX(7).EQ.0.0) THEN
            XX(36)=1.0
          ENDIF
        ENDIF
C
        DT1=.8
        DT2=.9
        DT3=1.
        DT4=1.1
        DT5=1.2
C
        GO TO(301,302,303,304,305),IMC
C
  301   CALL SCHDL(2,EXPON(DT1,1),ATRIB)
        GO TO 350
  302   CALL SCHDL(2,EXPON(DT2,1),ATRIB)
        GO TO 350
  303   CALL SCHDL(2,EXPON(DT3,1),ATRIB)
        GO TO 350
  304   CALL SCHDL(2,EXPON(DT4,1),ATRIB)
        GO TO 350
  305   CALL SCHDL(2,EXPON(DT5,1),ATRIB)
C
  350   ATRIB(1)=IMC*1.0
        RETURN
        END
C-------------------------------------------------------------------
```

```
      SUBROUTINE REPAIR
$INCLUDE:'PARAM.INC'
$INCLUDE:'SCOM1.COM'
C
      XMC=ATRIB(1)
      IMC=XMC
      XX(21)=0.0
      XX(22)=0.0
      XX(23)=0.0
      XX(24)=0.0
      XX(25)=0.0
      XX(26)=0.0
      XX(27)=0.0
      XX(28)=0.0
      XX(29)=0.0
      XX(30)=0.0
      XX(31)=0.0
      XX(32)=0.0
      XX(33)=0.0
      XX(34)=0.0
      XX(35)=0.0
      XX(36)=0.0
C
      GO TO(401,402,403,404,405),IMC
  401 XX(3)=1.0
      GO TO 450
  402 XX(4)=1.0
      GO TO 450
  403 XX(5)=1.0
      GO TO 450
  404 XX(6)=1.0
      GO TO 450
  405 XX(7)=1.0
      GO TO 450
  450 SUM=XX(3)+XX(4)+XX(5)+XX(6)+XX(7)
C
C......IF ENCOUNTERING STATE 1
      IF(SUM.EQ.5.0) THEN
        XX(21)=1.0
      ELSE
        IF(SUM.EQ.4.0) THEN
C......IF ENCOUNTERING STATE 2
          IF(XX(3).EQ.0.0) THEN
            XX(22)=1.0
          ENDIF
C......IF ENCOUNTERING STATE 3
          IF(XX(4).EQ.0.0) THEN
            XX(23)=1.0
          ENDIF
C......IF ENCOUNTERING STATE 4
          IF(XX(5).EQ.0.0) THEN
            XX(24)=1.0
          ENDIF
C......IF ENCOUNTERING STATE 5
          IF(XX(6).EQ.0.0) THEN
            XX(25)=1.0
          ENDIF
C......IF ENCOUNTERING STATE 6
          IF(XX(7).EQ.0.0) THEN
            XX(26)=1.0
          ENDIF
        ELSE
C
```

```
            IF(SUM.EQ.3.0) THEN
C......IF ENCOUNTERING STATE 7
            IF(XX(3).EQ.0.0.AND.XX(4).EQ.0.0) THEN
              XX(27)=1.0
            ENDIF
C......IF ENCOUNTERING STATE 8
            IF(XX(3).EQ.0.0.AND.XX(5).EQ.0.0) THEN
              XX(28)=1.0
            ENDIF
C......IF ENCOUNTERING STATE 9
            IF(XX(3).EQ.0.0.AND.XX(6).EQ.0.0) THEN
              XX(29)=1.0
            ENDIF
C......IF ENCOUNTERING STATE 10
            IF(XX(3).EQ.0.0.AND.XX(7).EQ.0.0) THEN
              XX(30)=1.0
            ENDIF
C......IF ENCOUNTERING STATE 11
            IF(XX(4).EQ.0.0.AND.XX(5).EQ.0.0) THEN
              XX(31)=1.0
            ENDIF
C......IF ENCOUNTERING STATE 12
            IF(XX(4).EQ.0.0.AND.XX(6).EQ.0.0) THEN
              XX(32)=1.0
            ENDIF
C......IF ENCOUNTERING STATE 13
            IF(XX(4).EQ.0.0.AND.XX(7).EQ.0.0) THEN
              XX(33)=1.0
            ENDIF
C......IF ENCOUNTERING STATE 14
            IF(XX(5).EQ.0.0.AND.XX(6).EQ.0.0) THEN
              XX(34)=1.0
            ENDIF
C......IF ENCOUNTERING STATE 15
            IF(XX(5).EQ.0.0.AND.XX(7).EQ.0.0) THEN
              XX(35)=1.0
            ENDIF
C......IF ENCOUNTERING STATE 16
            IF(XX(6).EQ.0.0.AND.XX(7).EQ.0.0) THEN
              XX(36)=1.0
            ENDIF
          ENDIF
        ENDIF
      ENDIF
C
C......SCHEDULE SUBSEQUENT FAILURE EVENTS
C
        GO TO(101,102,103,104,105),IMC
C
  101   CALL SCHDL(1,EXPON(80.,1),ATRIB)
        GO TO 550
  102   CALL SCHDL(1,EXPON(90.,1),ATRIB)
        GO TO 550
  103   CALL SCHDL(1,EXPON(100.,1),ATRIB)
        GO TO 550
  104   CALL SCHDL(1,EXPON(110.,1),ATRIB)
        GO TO 550
  105   CALL SCHDL(1,EXPON(120.,1),ATRIB)
C
  550   ATRIB(1)=IMC*1.0
        RETURN
        END
```

Appendix V

Computer Listing of SLAM II Discrete Event Simulation on Performance of the Flexible Manufacturing System

```
      SUBROUTINE EVENT(I)
      GO TO (1,2,3,4,5), I
    1 CALL GNRJT
      RETURN
    2 CALL DSPAGV
      RETURN
    3 CALL PROCS
      RETURN
    4 CALL BRKDN
      RETURN
    5 CALL REPAIR
      RETURN
      END
C---------------------------------------------------------
      SUBROUTINE INTLC
$INCLUDE:'PARAM.INC'
$INCLUDE:'SCOM1.COM'
      COMMON/DAI1/MTV(5,6)
      COMMON/DAI2/PST(5,6)
      COMMON/DAI3/TRT(8,8)
      COMMON/DAI4/JAGV(2)
      COMMON/DAI5/IOB(5)
      COMMON/DAI6/ISTP(2)
      COMMON/DAI7/ITO(2)
      COMMON/DAI8/LD(2)
      COMMON/DAI9/IUP(7)
      COMMON/DAI10/RMT(7)
      COMMON/DAI11/IJT(7)
      COMMON/DAI12/JIDX(7)
C......SET THE ORIGINAL STATUS OF THE MACHINES, AGVS AND OUTPUT BUFFERS
      NMC=5
      NLU=1
      NAGV=2
      NOB=5
C......ESTABLISH 5 MACHINES
      DO 10 I=1, (NMC+NLU)
      10  XX(I)=0.
C......ESTABLISH 2 AGVS
      DO 11 J=1,NAGV
      JAGV(J)=J+NMC+NLU+NOB
      XX(JAGV(J))=0.
C......AT THE BEGINNING, BOTH AGVS STOP AT LOCATION 7 AND
C......'ITO' IS THE DESTINATION OF AN AGV
      ISTP(J)=7
      ITO(J)=0
C......'LD'GIVES THE LOADING STATUS OF AGV AND '0' MEANS EMPTY
   11 LD(J)=0
C......ESTABLISH 5 OUTPUT BUFFERS (IOB)
      DO 12 K=1,NOB
   12 IOB(K)=K+NMC+NLU
C......SET THE ORIGINAL STATUS OF AGVS AND MACHINES UP
      DO 15 II=1,7
      IUP(II)=II+NMC+NOB+NAGV+NLU
   15 XX(IUP(II))=1.0
C......DEFINE THE TRAVEL TIME (TRT) OF AGV
      TRT(1,2)=8.
      TRT(1,3)=8.
      TRT(1,4)=8.
      TRT(1,5)=4.
      TRT(1,6)=4.
      TRT(1,7)=2.
      TRT(1,8)=6.
```

```
      TRT(2,1)=8.
      TRT(2,3)=4.
      TRT(2,4)=4.
      TRT(2,5)=8.
      TRT(2,6)=8.
      TRT(2,7)=6.
      TRT(2,8)=2.
      TRT(3,1)=8.
      TRT(3,2)=4.
      TRT(3,4)=4.
      TRT(3,5)=8.
      TRT(3,6)=8.
      TRT(3,7)=6.
      TRT(3,8)=2.
      TRT(4,1)=8.
      TRT(4,2)=4.
      TRT(4,3)=4.
      TRT(4,5)=8.
      TRT(4,6)=8.
      TRT(4,7)=6.
      TRT(4,8)=2.
      TRT(5,1)=4.
      TRT(5,2)=8.
      TRT(5,3)=8.
      TRT(5,4)=8.
      TRT(5,6)=4.
      TRT(5,7)=2.
      TRT(5,8)=6.
      TRT(6,1)=4.
      TRT(6,2)=8.
      TRT(6,3)=8.
      TRT(6,4)=8.
      TRT(6,5)=4.
      TRT(6,7)=2.
      TRT(6,8)=6.
      TRT(7,1)=2.
      TRT(7,2)=6.
      TRT(7,3)=6.
      TRT(7,4)=6.
      TRT(7,5)=2.
      TRT(7,6)=2.
      TRT(7,8)=4.
      TRT(8,1)=6.
      TRT(8,2)=2.
      TRT(8,3)=2.
      TRT(8,4)=2.
      TRT(8,5)=6.
      TRT(8,6)=6.
      TRT(8,7)=4.
C......DEFINE THE MACHINES TO BE VISITED(MTV)
      MTV(1,1)=1
      MTV(1,2)=5
      MTV(1,3)=3
      MTV(1,4)=2
      MTV(1,5)=6
      MTV(2,1)=2
      MTV(2,2)=3
      MTV(2,3)=5
      MTV(2,4)=4
      MTV(2,5)=6
      MTV(3,1)=3
      MTV(3,2)=4
```

```
      MTV(3,3)=1
      MTV(3,4)=6
      MTV(4,1)=4
      MTV(4,2)=5
      MTV(4,3)=1
      MTV(4,4)=6
      MTV(5,1)=5
      MTV(5,2)=1
      MTV(5,3)=4
      MTV(5,4)=2
      MTV(5,5)=3
      MTV(5,6)=6
C......DEFINE THE PROCESSING TIME 'PST'
      PST(1,1)=10.
      PST(1,2)=10.
      PST(1,3)=15.
      PST(1,4)=12.5
      PST(1,5)=0.5
      PST(1,6)=0.
      PST(2,1)=15.
      PST(2,2)=10.
      PST(2,3)=12.5
      PST(2,4)=10.
      PST(2,5)=0.5
      PST(2,6)=0.
      PST(3,1)=12.5
      PST(3,2)=15.
      PST(3,3)=12.5
      PST(3,4)=0.5
      PST(3,5)=0.
      PST(3,6)=0.
      PST(4,1)=12.5
      PST(4,2)=10.
      PST(4,3)=15.
      PST(4,4)=0.5
      PST(4,5)=0.
      PST(4,6)=0.
      PST(5,1)=10.
      PST(5,2)=10.
      PST(5,3)=12.5
      PST(5,4)=10.
      PST(5,5)=12.5
      PST(5,6)=0.5
C......DEFINE ATTRIBUTE 3 THE INDEX # OF A MACHINE TO BE VISITED
      ATRIB(3)=1.
C......SCHEDULE CREATING THE JOB AT TIME 0.
      CALL SCHDL(1,0.,ATRIB)
C......SCHEDULE THE INITIAL BREAK DOWN EVENT
      ATRIB(5)=1.0
      CALL SCHDL(4,EXPON(15000.,1),ATRIB)
      ATRIB(5)=2.0
      CALL SCHDL(4,EXPON(15000.,1),ATRIB)
      ATRIB(5)=3.0
      CALL SCHDL(4,EXPON(19200.,1),ATRIB)
      ATRIB(5)=4.0
      CALL SCHDL(4,EXPON(19800.,1),ATRIB)
      ATRIB(5)=5.0
      CALL SCHDL(4,EXPON(20400.,1),ATRIB)
      ATRIB(5)=6.0
      CALL SCHDL(4,EXPON(21000.,1),ATRIB)
      ATRIB(5)=7.0
      CALL SCHDL(4,EXPON(21600.,1),ATRIB)
```

```
      RETURN
      END
C-------------------------------------------------------------------
      SUBROUTINE GNRJT
$INCLUDE:'PARAM.INC'
$INCLUDE:'SCOM1.COM'
      COMMON/DAI1/MTV(5,6)
      COMMON/DAI2/PST(5,6)
      COMMON/DAI3/TRT(8,8)
      COMMON/DAI4/JAGV(2)
      COMMON/DAI5/IOB(5)
      COMMON/DAI6/ISTP(2)
      COMMON/DAI7/ITO(2)
      COMMON/DAI8/LD(2)
      COMMON/DAI9/IUP(7)
      COMMON/DAI10/RMT(7)
      COMMON/DAI11/IJT(7)
      COMMON/DAI12/JIDX(7)
C......SCHEDULE SUBSEQUENT GENERATION OF JOB TYPE
      CALL SCHDL(1,UNFRM(20.0,30.0,1),ATRIB)
C......GENERATE JOB TYPE UNIFORMLY
      Z=UNFRM(0.0,1.0,1)
      IF(Z.GE.0.0.AND.Z.LE.0.2) XJT=1.
      IF(Z.GT.0.2.AND.Z.LE.0.4) XJT=2.
      IF(Z.GT.0.4.AND.Z.LE.0.6) XJT=3.
      IF(Z.GT.0.6.AND.Z.LE.0.8) XJT=4.
      IF(Z.GT.0.8.AND.Z.LE.1.0) XJT=5.
C......DEFINE ATTRIBUTE 2 THE JOB TYPE #
      ATRIB(2)=XJT
      IDX=ATRIB(3)
C......ASSIGN ATTRIBUTE 1 THE MARK TIME
      ATRIB(1)=TNOW
C......DEFINE 'JT' JOB TYPE, 'JFM'JOB FROM AND 'JTO'JOB TO
C......STORE JOB GENERATD IN FILE 12
      CALL FILEM(12,ATRIB)
      JFM=6
      NN=NFIND(1,12,2,0,XJT,0.0)
      JT=XJT
C......CHECK IF THE SIZE OF THE INPUT BUFFER OF A MACHINE LESS THAN 10
      IF(NNQ(JT).LE.10.0) GO TO 80
      RETURN
C......SCHEDULE DESPATCHING AGV
   80 IF(XX(JAGV(1)).EQ.0..AND.XX(JAGV(2)).EQ.0.) GO TO 100
      IF(XX(JAGV(1)).EQ.0..AND.XX(JAGV(2)).GT.0.) GO TO 200
      IF(XX(JAGV(2)).EQ.0..AND.XX(JAGV(1)).GT.0.) GO TO 200
   90 RETURN
  100 IF(XX(IUP(1)).EQ.0.0.AND.XX(IUP(2)).EQ.0.0) GO TO 90
        CALL RMOVE(NN,12,ATRIB)
        JTO=MTV(JT,IDX)
        T1=TRT(ISTP(1),JFM)
        T2=TRT(ISTP(2),JFM)
        IF(T2.LT.T1) THEN
        IAGV=2
      ELSE
        IAGV=1
      ENDIF
      IF(XX(IUP(1)).EQ.0.0) THEN
        IAGV=2
      ELSE
        IF(XX(IUP(2)).EQ.0.0) THEN
          IAGV=1
        ENDIF
```

```
      ENDIF
C......SCHEDULE CALLING THE NEAREST AGV
      XX(JAGV(IAGV))=1.0
C......IF THE CALLED AGV STOPS AT 8
      IF(ISTP(IAGV).EQ.8) THEN
        DT=TRT(ISTP(IAGV),7)
        ITO(IAGV)=7
      ELSE
        DT=TRT(ISTP(IAGV),JFM)
        ITO(IAGV)=JFM
      ENDIF
      GO TO 70
C......IF ONLY ONE AGV IDLE
  200 IF(XX(JAGV(1)).EQ.0.0.AND.XX(IUP(1)).EQ.0.0) GO TO 90
      IF(XX(JAGV(2)).EQ.0.0.AND.XX(IUP(2)).EQ.0.0) GO TO 90
      CALL RMOVE(NN,12,ATRIB)
      JTO=MTV(JT,IDX)
      IF(XX(JAGV(1)).GT.0.) THEN
        XIDLE=2.
      ELSE
        XIDLE=1.
      ENDIF
      IDLE=XIDLE
      XX(JAGV(IDLE))=1.0
      IF(IDLE.EQ.1) THEN
        IBUSY=2
      ELSE
        IBUSY=1
      ENDIF
      IF(ISTP(IDLE).EQ.8) THEN
        ITO(IDLE)=7
        IF(ISTP(IBUSY).EQ.7.AND.ITO(IBUSY).EQ.8) THEN
          DT=TRT(ISTP(IDLE),7)*2
        ELSE
          DT=TRT(ISTP(IDLE),7)
        ENDIF
      ELSE
        DT=TRT(ISTP(IDLE),JFM)
        ITO(IDLE)=JFM
      ENDIF
      IAGV=IDLE
   70 ATRIB(2)=JT*1.0
      ATRIB(3)=IDX*1.
      ATRIB(4)=IAGV*1.
      IJT(IAGV)=JT
      JIDX(IAGV)=IDX
      CALL SCHDL(2,0.5*DT,ATRIB)
      RETURN
      END
C-----------------------------------------------------------------
      SUBROUTINE DSPAGV
$INCLUDE:'PARAM.INC'
$INCLUDE:'SCOM1.COM'
      COMMON/DAI1/MTV(5,6)
      COMMON/DAI2/PST(5,6)
      COMMON/DAI3/TRT(8,8)
      COMMON/DAI4/JAGV(2)
      COMMON/DAI5/IOB(5)
      COMMON/DAI6/ISTP(2)
      COMMON/DAI7/ITO(2)
      COMMON/DAI8/LD(2)
      COMMON/DAI9/IUP(7)
```

```
      COMMON/DAI10/RMT(7)
      COMMON/DAI11/IJT(7)
      COMMON/DAI12/JIDX(7)
C......SCHEDULE DESPATCHING THE BUSY AGVS
      IF(XX(JAGV(1)).GT.0..AND.XX(JAGV(2)).GT.0.) GO TO 100
      IF(XX(JAGV(1)).GT.0..AND.XX(JAGV(2)).EQ.0.) GO TO 200
      IF(XX(JAGV(2)).GT.0..AND.XX(JAGV(1)).EQ.0.) GO TO 200
C......IF BOTH AGVS ARE BUSY
  100 IF(XX(IUP(1)).EQ.0.0.AND.XX(IUP(2)).EQ.0.0) GO TO 320
      IAGV=ATRIB(4)
      GO TO(39,40), IAGV
C......FOR AGV(1)
   39 IF(XX(IUP(1)).EQ.0.0) GO TO 320
      ISTP(1)=ITO(1)
      JT=ATRIB(2)
      IDX=ATRIB(3)
      IF(IDX.EQ.1) THEN
      JFM=6
      ELSE
      JFM=MTV(JT,(IDX-1))
      ENDIF
      JTO=MTV(JT,IDX)
      IF(LD(1).EQ.1) GO TO 41
C......IF REQUESTED AGV ARRIVES
      IF(ISTP(1).EQ.JFM) THEN
        LD(1)=1
        IF(JFM.GT.1.AND.JFM.LT.5) THEN
          ITO(1)=8
          DT=TRT(JFM,8)
        ELSE
          ITO(1)=7
          DT=TRT(JFM,7)
        ENDIF
      ELSE
        ITO(1)=JFM
        DT=TRT(ISTP(1),JFM)
      ENDIF
C......SCHEDULE DESPATCHING AGV(1)
      GO TO 300
C......WHEN PART IS LOADED ON AGV(1)
   41 IF(ISTP(1).EQ.JTO) THEN
        IAGV=1
        GO TO 43
      ELSE
        IF(JTO.GT.1.AND.JTO.LT.5) THEN
          IF(ISTP(1).EQ.8) THEN
            ITO(1)=JTO
            IF(ISTP(2).EQ.JTO.AND.ITO(2).EQ.8) THEN
              DT=TRT(ISTP(1),ITO(1))*3
            ELSE
              DT=TRT(ISTP(1),ITO(1))*2
            ENDIF
          ELSE
            IF(ISTP(1).EQ.7) THEN
              ITO(1)=8
              DT=TRT(ISTP(1),ITO(1))
            ENDIF
          ENDIF
        ELSE
C......IF DESTINATION IS MC1,MC5 OR L/U STATION
          IF(ISTP(1).EQ.7) THEN
            ITO(1)=JTO
```

/00


```
            IF(ISTP(2).EQ.JTO.AND.ITO(2).EQ.7) THEN
              DT=TRT(ISTP(1),ITO(1))*3
            ELSE
              DT=TRT(ISTP(1),ITO(1))*2
            ENDIF
          ELSE
            IF(ISTP(1).EQ.8) THEN
              ITO(1)=7
              IF(ISTP(2).EQ.7.AND.ITO(2).EQ.8) THEN
                DT=TRT(ISTP(1),ITO(1))*2
              ELSE
                DT=TRT(ISTP(1),ITO(1))
              ENDIF
            ENDIF
          ENDIF
        ENDIF
      ENDIF
C......SCHEDULE DESPATCTCHING AGV(1)
      GO TO 300
C......FOR AGV(2)
   40 IF(XX(IUP(2)).EQ.0.0) GO TO 320
      ISTP(2)=ITO(2)
      JT=ATRIB(2)
      IDX=ATRIB(3)
      IF(IDX.EQ.1) THEN
       JFM=6
      ELSE
       JFM=MTV(JT,(IDX-1))
      ENDIF
       JTO=MTV(JT,IDX)
      IF(LD(2).EQ.1) GO TO 44
C......WHEN NO PART LAODED ON AGV(2)
      IF(ISTP(2).EQ.JFM) THEN
        LD(2)=1
        IF(JFM.GT.1.AND.JFM.LT.5) THEN
          ITO(2)=8
          DT=TRT(JFM,8)
        ELSE
          ITO(2)=7
          DT=TRT(JFM,7)
        ENDIF
      ELSE
        ITO(2)=JFM
        DT=TRT(ISTP(2),JFM)
      ENDIF
C......SCHEDULE DESPATCHING AGV(2)
      GO TO 300
C......WHEN PART IS LOADED ON AGV(2)
   44 IF(ISTP(2).EQ.JTO) THEN
        IAGV=2
        GO TO 43
      ELSE
C......TO MC2,MC3 OR MC4
        IF(JTO.GT.1.AND.JTO.LT.5) THEN
          IF(ISTP(2).EQ.8) THEN
            ITO(2)=JTO
            IF(ISTP(1).EQ.JTO.AND.ITO(1).EQ.8) THEN
              DT=TRT(ISTP(2),ITO(2))*3
            ELSE
              DT=TRT(ISTP(2),ITO(2))*2
            ENDIF
          ELSE
```

```
            IF(ISTP(2).EQ.7) THEN
              ITO(2)=8
              DT=TRT(ISTP(2),ITO(2))
            ENDIF
          ENDIF
        ELSE
C......IF AGV(2) GOES TO MC1,MC5 OR L/U STATON
          IF(ISTP(2).EQ.7) THEN
            ITO(2)=JTO
            IF(ISTP(1).EQ.JTO.AND.ITO(1).EQ.7) THEN
              DT=TRT(ISTP(2),ITO(2))*3
            ELSE
              DT=TRT(ISTP(2),ITO(2))*2
            ENDIF
          ELSE
            IF(ISTP(2).EQ.8) THEN
              ITO(2)=7
              IF(ISTP(1).EQ.7.AND.ITO(1).EQ.8) THEN
                DT=TRT(ISTP(2),ITO(2))*2
              ELSE
                DT=TRT(ISTP(2),ITO(2))
              ENDIF
            ENDIF
          ENDIF
        ENDIF
      ENDIF
      GO TO 300
C......IF ONLY ONE AGV BUSY
  200 IF(XX(JAGV(1)).GT.0.0) THEN
        IAGV=1
      ELSE
        IAGV=2
      ENDIF
      IF(XX(IUP(IAGV)).EQ.0.0) GO TO 320
      ISTP(IAGV)=ITO(IAGV)
      JT=ATRIB(2)
      IDX=ATRIB(3)
      IF(IDX.EQ.1) THEN
        JFM=6
      ELSE
        JFM=MTV(JT,(IDX-1))
      ENDIF
        JTO=MTV(JT,IDX)
C......IF THE BUSY AGV WAS LOADED
      IF(LD(IAGV).EQ.1) GO TO 50
C......IF THE BUSY AGV IS UNLOADED
      IF(ISTP(IAGV).EQ.JFM) THEN
        LD(IAGV)=1
        IF(JFM.GT.1.AND.JFM.LT.5) THEN
          ITO(IAGV)=8
          DT=TRT(JFM,8)
        ELSE
          ITO(IAGV)=7
          DT=TRT(JFM,7)
        ENDIF
      ELSE
C......IF MC2,MC3 OR MC4 CALL AGV
      IF(JFM.GT.1.AND.JFM.LT.5) THEN
        ITO(IAGV)=JFM
        DT=TRT(8,JFM)
      ELSE
        ITO(IAGV)=JFM
```

```
          DT=TRT(7,JFM)
        ENDIF
      ENDIF
      GO TO 300
C......IF THE PART IS LOADED ON AGV(BUSY)
   50 IF(ISTP(IAGV).EQ.JTO) GO TO 43
      IF(JTO.GT.1.AND.JTO.LT.5) THEN
        IF(ISTP(IAGV).EQ.8) THEN
          ITO(IAGV)=JTO
          DT=TRT(8,JTO)*2
        ELSE
          ITO(IAGV)=8
          DT=TRT(ISTP(IAGV),8)
        ENDIF
      ELSE
C......IF DESTINATION IS MC1, MC5 OR L/U STATION
        IF(ISTP(IAGV).EQ.7) THEN
          ITO(IAGV)=JTO
          DT=TRT(7,JTO)*2
        ELSE
          ITO(IAGV)=7
          DT=TRT(ISTP(IAGV),7)
        ENDIF
      ENDIF
C......SCHEDULE DESPATCHING AGV(1) OR AGV(2)
  300 ATRIB(2)=JT*1.
      ATRIB(3)=IDX*1.
      ATRIB(4)=IAGV*1.
      IJT(IAGV)=JT
      JIDX(IAGV)=IDX
      RMT(IAGV)=TNOW+DT
      CALL SCHDL(2,0.5*DT,ATRIB)
  320 RETURN
C......IF LOADED AGV ARRIVES AT ITS DESTINATION
   43 JTO=MTV(JT,IDX)
      ATRIB(2)=JT*1.
      ATRIB(3)=IDX*1.
      XX(JAGV(IAGV))=0.
      LD(IAGV)=0
      IF(JTO.GT.1.AND.JTO.LT.5) THEN
        ISTP(IAGV)=8
      ELSE
        ISTP(IAGV)=7
      ENDIF
      ATRIB(4)=0.0
      IF(JTO.EQ.0) GO TO 48
      IF(XX(IUP(JTO+2)).EQ.0.0) GO TO 46
      IF(XX(JTO).EQ.0.0) GO TO 45
   46 CALL FILEM(JTO,ATRIB)
      IF(JTO.EQ.6) GO TO 54
C......CHECK IF THERE ARE ANY PART WAITING
C......AT THE OUTPUT BUFFER OF MACHINE 'JTO'
      IF(NNQ(IOB(JTO)).EQ.0.0) GO TO 54
    8 DO 4 IJ=1,5
      XIJ=IJ
      N1=NFIND(1,IOB(JTO),2,0,XIJ,0.0)
      IF(N1.EQ.1) JJT=XIJ
    4 CONTINUE
      DO 5 K=1,6
      XK=K
      N2=NFIND(1,IOB(JTO),3,0,XK,0.0)
      IF(N2.EQ.1) IIDX=XK
```

```
    5   CONTINUE
        IF(XX(JAGV(1)).EQ.0..AND.XX(JAGV(2)).EQ.0.) GO TO 500
        IF(XX(JAGV(1)).EQ.0..AND.XX(JAGV(2)).GT.0.) GO TO 600
        IF(XX(JAGV(1)).GT.0..AND.XX(JAGV(2)).EQ.0.) GO TO 600
C......IF BOTH AGVS ARE BUSY
        RETURN
  500   IF(XX(IUP(1)).EQ.0.0.AND.XX(IUP(2)).EQ.0.0) GO TO 54
        T1=TRT(ISTP(1),JTO)
        T2=TRT(ISTP(2),JTO)
        IF(T2.LT.T1) THEN
          IIAGV=2.
        ELSE
          IIAGV=1.
        ENDIF
        IF(XX(IUP(1)).EQ.0.0) IIAGV=2
        IF(XX(IUP(2)).EQ.0.0) IIAGV=1
C......SCHEDULE CALLING THE NEAREST AGV
        XX(JAGV(IIAGV))=1.0
        IF(JTO.GT.0.AND.JTO.LT.6) THEN
          CALL RMOVE(1,IOB(JTO),ATRIB)
        ENDIF
C......IF MC2,MC3 OR MC4 REQUEST AGV
        IF(JTO.GT.1.AND.JTO.LT.5) THEN
C......IF THE CALLED AGV STOPS AT 8
          IF(ISTP(IIAGV).EQ.8) THEN
            DT=TRT(ISTP(IIAGV),JTO)
            ITO(IIAGV)=JTO
          ELSE
            DT=TRT(ISTP(IIAGV),8)
            ITO(IIAGV)=8
          ENDIF
C......IF MC1,MC5 OR L/U REQUESTS AGV
        ELSE
          IF(ISTP(IIAGV).EQ.7) THEN
            DT=TRT(ISTP(IIAGV),JTO)
            ITO(IIAGV)=JTO
          ELSE
            DT=TRT(ISTP(IIAGV),7)
            ITO(IIAGV)=7
          ENDIF
        ENDIF
        GO TO 18
C......IF ONE AGV IDLE AND ANOTHER ONE BUSY
  600   XIDLE=0.
        DO 601 I=1,2
        IF(XX(JAGV(I)).EQ.0.) XIDLE=I
  601   CONTINUE
        IDLE=XIDLE
        IF(XX(IUP(IDLE)).EQ.0.0) GO TO 54
        XX(JAGV(IDLE))=1.0
        IF(JTO.GT.0.AND.JTO.LT.6) THEN
          CALL RMOVE(1,IOB(JTO),ATRIB)
        ENDIF
        IF(IDLE.EQ.1) THEN
          IBUSY=2
        ELSE
          IBUSY=1
        ENDIF
C......MC1,MC5 OR L/U CALLS FOR AGV
        IF(JTO.LT.2.AND.JTO.GT.4) THEN
          IF(ISTP(IDLE).EQ.8) THEN
            ITO(IDLE)=7
```

```
          IF(ISTP(IBUSY).EQ.7.AND.ITO(IBUSY).EQ.8) THEN
            DT=TRT(ISTP(IDLE),7)*2
          ELSE
            DT=TRT(ISTP(IDLE),7)
          ENDIF
        ELSE
          DT=TRT(ISTP(IDLE),JTO)
          ITO(IDLE)=JTO
        ENDIF
      ELSE
C......MC2,MC3 OR MC4 CALLS FOR AGV
        IF(ISTP(IDLE).EQ.8) THEN
          DT=TRT(ISTP(IDLE),JTO)
          ITO(IDLE)=JTO
        ELSE
          ITO(IDLE)=8
          IF(ISTP(IBUSY).EQ.8.AND.ITO(IBUSY).EQ.7) THEN
            DT=TRT(ISTP(IDLE),8)*2
          ELSE
            DT=TRT(ISTP(IDLE),8)
          ENDIF
        ENDIF
      ENDIF
      IIAGV=XIDLE
C......SCHEDULE DESPATCHING AGV
   18 ATRIB(2)=JJT*1.
      ATRIB(3)=IIDX*1.
      ATRIB(4)=IIAGV*1.
      IJT(IIAGV)=JJT
      JIDX(IIAGV)=IIDX
      RMT(IIAGV)=TNOW+DT
      CALL SCHDL(2,0.5*DT,ATRIB)
   54 RETURN
   45 XX(JTO)=1.0
      DT1=1.5*PST(JT,IDX)
      IJT(JTO+2)=JT
      JIDX(JTO+2)=IDX
      RMT(JTO+2)=TNOW+DT1
   48 CALL SCHDL(3,DT1,ATRIB)
      IF(JTO.LT.6.AND.NNQ(IOB(JTO)).GT.0.0) GO TO 8
      RETURN
      END
C-----------------------------------------------------------------
      SUBROUTINE PROCS
$INCLUDE:'PARAM.INC'
$INCLUDE:'SCOM1.COM'
      COMMON/DAI1/MTV(5,6)
      COMMON/DAI2/PST(5,6)
      COMMON/DAI3/TRT(8,8)
      COMMON/DAI4/JAGV(2)
      COMMON/DAI5/IOB(5)
      COMMON/DAI6/ISTP(2)
      COMMON/DAI7/ITO(2)
      COMMON/DAI8/LD(2)
      COMMON/DAI9/IUP(7)
      COMMON/DAI10/RMT(7)
      COMMON/DAI11/IJT(7)
      COMMON/DAI12/JIDX(7)
C......WHEN PART ARRIVES TO THE MACHINE
      JT=ATRIB(2)
      IDX=ATRIB(3)
      IMC=MTV(JT,IDX)
```

```
      IF(NNQ(IMC).GT.0.) GO TO 9
      XX(IMC)=0.0
      IF(MTV(JT,(IDX+1)).EQ.0) GO TO 20
      IF(IMC.EQ.6) GO TO 20
      ATRIB(3)=IDX*1.+1.
      CALL FILEM(IOB(IMC),ATRIB)
      RETURN
    9 DO 10 IK=5,5
      YIK=IK
      M1=NFIND(1,IMC,2,0,YIK,0.0)
      IF(M1.EQ.1) JT=YIK
   10 CONTINUE
      DO 15 JK=1,6
      YJK=JK
      M2=NFIND(1,IMC,3,0,YJK,0.0)
      IF(M2.EQ.1) IDX=YJK
   15 CONTINUE
      IF(XX(IUP(IMC+2)).EQ.0.0) GO TO 20
      CALL RMOVE(1,IMC,ATRIB)
      DT1=1.5*PST(JT,IDX)
      IJT(IMC+2)=JT
      JIDX(IMC+2)=IDX
      RMT(IMC+2)=TNOW+DT1
      CALL SCHDL(3,DT1,ATRIB)
      IF(MTV(JT,(IDX+1)).EQ.0) GO TO 20
      IF(IMC.EQ.6) GO TO 20
      ATRIB(2)=JT*1.
      ATRIB(3)=IDX*1.+1.
      CALL FILEM(IOB(IMC),ATRIB)
   20 RETURN
      END
C-----------------------------------------------------------------
      SUBROUTINE BRKDN
$INCLUDE:'PARAM.INC'
$INCLUDE:'SCOM1.COM'
      COMMON/DAI1/MTV(5,6)
      COMMON/DAI2/PST(5,6)
      COMMON/DAI3/TRT(8,8)
      COMMON/DAI4/JAGV(2)
      COMMON/DAI5/IOB(5)
      COMMON/DAI6/ISTP(2)
      COMMON/DAI7/ITO(2)
      COMMON/DAI8/LD(2)
      COMMON/DAI9/IUP(7)
      COMMON/DAI10/RMT(7)
      COMMON/DAI11/IJT(7)
      COMMON/DAI12/JIDX(7)
C......SCHEDULE THE REPAIR EVENT
      DN=ATRIB(5)
      IDN=DN
      XX(IUP(IDN))=0.0
      GO TO(101,102,103,104,105,106,107),IDN
C......CHECK IF AN AGV OR A MACHINE IS BUSY
  101 RT=60.0
      IF(XX(JAGV(IDN)).EQ.0.0) GO TO 110
      ATRIB(4)=IDN*1.0
      GO TO 120
  102 RT=60.0
      IF(XX(JAGV(IDN)).EQ.0.0) GO TO 110
      ATRIB(4)=IDN*1.0
      GO TO 120
  103 RT=1.5*60.0
```

```
      IF(XX(1).EQ.0.0) GO TO 110
      GO TO 120
  104 RT=1.6*60.0
      IF(XX(2).EQ.0.0) GO TO 110
      GO TO 120
  105 RT=1.7*60.0
      IF(XX(3).EQ.0.0) GO TO 110
      GO TO 120
  106 RT=1.8*60.0
      IF(XX(4).EQ.0.0) GO TO 110
      GO TO 120
  107 RT=1.9*60.0
      IF(XX(5).EQ.0.0) GO TO 110
  120 K=NFIND(1,NCLNR,5,0,DN,0.0)
C......REMOVE THE BREAK DOWN EVENT FROM THE CALENDAR FILE
      IF(K.GT.0) THEN
        CALL RMOVE(K,NCLNR,ATRIB)
      ENDIF
  110 ATRIB(5)=DN
      ATRIB(2)=IJT(IDN)*1.0
      ATRIB(3)=JIDX(IDN)*1.0
      RMT(IDN)=RMT(IDN)+RT-TNOW
      CALL SCHDL(5,RT,ATRIB)
      RETURN
      END
C--------------------------------------------------------
      SUBROUTINE REPAIR
$INCLUDE:'PARAM.INC'
$INCLUDE:'SCOM1.COM'
      COMMON/DAI1/MTV(5,6)
      COMMON/DAI2/PST(5,6)
      COMMON/DAI3/TRT(8,8)
      COMMON/DAI4/JAGV(2)
      COMMON/DAI5/IOB(5)
      COMMON/DAI6/ISTP(2)
      COMMON/DAI7/ITO(2)
      COMMON/DAI8/LD(2)
      COMMON/DAI9/IUP(7)
      COMMON/DAI10/RMT(7)
      COMMON/DAI11/IJT(7)
      COMMON/DAI12/JIDX(7)
C......SCHEDULE THE SUBSEQUENT BREAKDOWN EVENT
      UP=ATRIB(5)
      JUP=UP
      XJT=ATRIB(2)
      XIDX=ATRIB(3)
      XX(IUP(JUP))=1.0
      GO TO(201,202,203,204,205,206,207),JUP
  201 TTF=15000.0
      GO TO 210
  202 TTF=15000.0
      GO TO 210
  203 TTF=19200.0
      GO TO 210
  204 TTF=19800.0
      GO TO 210
  205 TTF=20400.0
      GO TO 210
  206 TTF=21000.0
      GO TO 210
  207 TTF=21600.0
  210 ATRIB(5)=UP
```

```
C......SCHEDULE THE BREAK DOWN EVENT FOR THE REMAINING OPERATION TIME
       IF(JUP.LE.2.AND.XX(JAGV(JUP)).EQ.0.0) GO TO 220
       IF(JUP.GT.2.AND.XX(JUP-2).EQ.0.0) GO TO 220
       IF(JUP.LE.2) THEN
         IVNT=2
         ATRIB(4)=JUP*1.0
       ELSE
         IVNT=3
       ENDIF
       ATRIB(2)=XJT
       ATRIB(3)=XIDX
       CALL SCHDL(IVNT,RMT(JUP),ATRIB)
   220 CALL SCHDL(4,EXPON(TTF,1),ATRIB)
       RETURN
       END
```