ABSTRACT

Simulation of Manufacturing Cell Using HOOPS Graphics Software and SIMAN Simulation Language

by
Vipul Parikh

Control of jobs being processed in manufacturing shop floor is important to management. The performance of job shop is affected by shop congestion due to arrival rate, batch size, number of machines, buffer capacity, process time variation, and dispatching rules use to load the jobs on the machine.

The job shop is a very complex system and hence, it is difficult as well as time consuming to obtained result using analytical method. A SIMAN simulation language is used to create generic model of a manufacturing cell and simulation analysis of those parameters variation such as job shop efficiency, the average no. of parts in buffer and the product processing time in the system was carried out by taking various examples.

With slight variation of the above parameters in the experimental file of simulation program, the analysis of various job shop situation was done. Result indicated that the shop floor congestion due to the above parameters greatly affect the performance of the job shop.

SIMULATION OF MANUFACTURING CELL USING HOOPS GRAPHIC SOFTWARE AND SIMAN SIMULATION LANGUAGE

by
Vipul Parikh

A Thesis
Submitted to the Faculty of
New Jersey Institute of Technology
in Partial Fulfillment of the Requirements for the Degree of
Master of Science in Manufacturing Engineering

Department of Manufacturing Engineering

May 1991

Blank Page

APPROVAL PAGE

Simulation of Manufacturing cell
Using HOOPS Graphic Software
and
SIMAN Simulation Language

by
Vipul Parikh

Dr. E. Geskin, Thesis Adviser — Date
Professor of Mechanical Engineering, NJIT

Dr. E. Gordon, Committee Member — Date
Distinguished Research Professor of Electrical Engineering and Deputy Executive Director of the Center for Manufacturing Systems

Dr. N. Levy, Committee Member — Date
Associate professor of Mechanical Engineering, NJIT

BIOGRAPHICAL SKETCH

Author: Vipul Parikh

Degree: Master of Science in Manufacturing Engineering

Date: May, 1991

Undergraduate and Graduate Education:

- Master of Science in Manufacturing Engineering,
  New Jersey Institute of Technology, Newark, NJ 1991
- Post Diploma in Tool Engineering,
  Shree Bhagubhai Mafatlal Polytechnic, India, 1988
- Diploma in Mechanical Engineering
  Shree Bhagubhai Mafatlal Polytechnic, India, 1986

Major: Manufacturing Engineering

ACKN0WLEDGMENT

I would like to take this opportunity to express my sincere gratitude to Thesis Advisor , Dr. E. Geskin, for his support through out this research.

Special thanks to Dr. E. Gordon and Dr. N. Levy for serving as members of the committee and their guidance.

I would also like to thank Dr. Perel for his guidance . I wish to express my sincere gratitude to Mr. K. Reddy for his guidance, friendship, timely help, suggestions, and moral support throughout this research.

TABLE OF CONTENTS

Chapter **Pages**

LIST OF FIGURES

Figures **Page**

CHAPTER 1

SIMULATION

Simulation has been defined by Shannon as "the process of designing a computerized model of a system (or process) and conducting experiments with this model for the purpose either of understanding the behavior of the system or of evaluating various strategies for the operation of the system."

The simulation allows the assessment of the potential performance before a newly designed system is operable. It also permits the comparison of various operating strategies of a present system without affecting the actual performance of the system. It also allows time compression or expansion of the system's operation, i.e., it is possible to simulate months or years of activity by a system in a few minutes of computer time.

1.1 Concept of a system

Central to any simulation study is the idea of a system. The term system is defined in Funk and Wagnall's Standard Dictionary as "an orderly collection of logically related principles, facts or objects." When used in the context of a simulation study, the term system generally refers to a collection of objects with a well-defined set of interactions among them.

System can be defined more broadly than simply as a collection of objects and interactions. For example, all external factors are capable of causing a change in the system. These external factors form the system environment. The state of a system is the minimal collection of information with which its future behavior can be uniquely predicted in the absence of chance events. Since the inclusion of time in the consideration of a system implies that the state of a system changes, there must be some process or event that prompts this change.

Such a process or event is called an activity. The system state may change in response to activities internal or external to the system. Activities external to the system are referred to as exogenous, while activities internal to the system are referred to as endogenous. Though, in actual simulation, it is the change in the system state induced by an activity that is of primary interest. The system can be broadly classified as follows:

1.1.1 Continuous versus discrete systems

The terms continuous and discrete applied to a system refers to the nature or behavior of changes with respect to time in the system state. A system whose changes in state occur continuously over time are continuous systems, while systems whose changes occur in finite quanta, or jumps, are discrete. System that possesses the properties of both continuous and discrete systems, are called hybrid systems.

1.1.2 Stochastic versus deterministic systems

A deterministic system is a system in which a new state of the system is completely determined by the previous state and by the activity. A stochastic system contains a certain amount of randomness in its transitions from one state to another. In some cases it might not be possible to assign a probability to the state that the system will assume after a given state and activity. In other cases these probabilities are known or can be determined.

1.1.3 Open versus closed systems

A closed system is a system in which all state changes are prompted by endogenous activities. A open system are systems whose state changes in response to both exogenous and endogenous activities.

1.2 System Methodology

Simulation is based on a problem-solving method. When system simulation is used to solve a problem, the following time-tested steps or stages are applied:

1) Observation of the system.

2) Formulation of hypotheses or theories that account for the observe behavior.

3) Prediction of the future behavior of the system based on the assumption that the hypotheses are correct.

4) Comparison of the predicted behavior with the actual behavior.

1.2.1 Planning

The initial phase of problem-solving process is planning. Planning or pre - modeling phase includes the initial encounter with the system, the problem to be solved, and the factors pertaining to the system and its environment that are likely to affect the solution of the problem. This definition of the planning phase assumes that the problem has been adequately defined. Obviously, the more accurate and precise the problem statement is, the more smoothly the solution process can be continued.

Once the problem has been clearly defined, an estimate of the resources required to collect data and analyze the problem (system) can be made. Resources such as time, money, personnel, and special equipment should be considered. If crucial resources are not available, solution of the problem can be judged infeasible before a significant amount of time or money is spent. The alternative to discontinuing the project is to modify the definition of the problem. The problem can be restated so that it can successfully be solved with available resources. It is more desirable to modify objectives at an early stage than to fall short because crucial resources are unavailable.

The second major task in planning phase is to analyze the system. In this phase the analyst attempts to become familiar with all relevant aspects of the problem being studied. A thorough literature survey to discover previous approaches to similar problems may prove valuable in choosing possible courses of action. Consultation with other qualified person may provide insight into aspects of the problem that the primary researchers have overlooked or misunderstood.

1.2.2 Modeling

The second phase of the problem-solving process is the modeling phase. In this phase system model is constructed, which is a representation of the real system. The characteristics of this model should be representative of the characteristics of the real system. Many real systems are so complex that a model with characteristics identical to those of the real system would be equally complex and thus unmanageable and prohibitively expensive. One of the goals of this phase, then, is to select some minimal set of the system's characteristics so that the model approximates the real system yet remains cost-effective and manageable. The model includes these characteristics and a well-defined set of relations among these characteristics.

There are many types of models, including descriptive, physical, and mathematical, flowcharts, schematics, and computer programs. Descriptive models are just verbalizations of the system's composition and its response to a given stimulus. Physical models are scaled facsimiles of the system being analyzed. Mathematical models are abstract expressions of the relationship among system variables. The advantage of mathematical models is that it can be manipulated with pencil and paper. With these models it is not necessary to construct an expensive mock-up of the system to study its behavior. Their disadvantage is that an accurate description of a given system may require an expression so complex that it would make even the most facile mathematician shudder. Flowcharts and schematics display the basic logical interaction between system components and may be as detailed as desired. When used as a prelude to computer program models, flowcharts models are valuable aids to programming and program documentation.

If the system being studied is so complex that no representative model can be used, then the well-known problem reduction technique or subsystem modeling is an alternate approach. In this approach the system is divided into a collection of less complex subsystems. Each of these subsystem is modeled, and the overall system model is constructed by linking the subsystem models appropriately.

Three general approaches have been used in defining or identifying subsystems. The first method, called the flow approach, has been used to analyze systems characterized by the flow of physical or information items through the system. If the system is to be modeled is an automobile assembly plant, the flow approach would appear to be a feasible way of breaking this system into subsystems. In the flow approach, subsystems are identified by grouping aspects of the system that produce a particular physical or information change in the flow entity.

A second technique used in identifying subsystems is the functional approach. This technique is useful when there are no directly observable "flowing" entities in the system. Instead, a logical sequence of functions being performed must be identified. System characteristics that perform a given function are grouped to form a subsystem. This approach can be used in manufacturing processes that do not use assembly lines.

The third method for identifying or isolating subsystems is called the state-change approach. This procedure is useful in systems that are characterized by a large number of interdependent relationships which must be examined at regular intervals to detect state changes. System characteristics that respond to the same stimulus or set of stimuli are then grouped to form a subsystem.

Once the subsystems have been identified and isolated, they must be modeled. Of course, the subsystems could in turn be subdivided into sub-systems, and so on, as far as necessary. The goal is to subdivide the total system in such a way that each subsystem is easily comprehensible and readily identified. If a subsystem can not be identified and isolated, the analyst has no choice but to model the entire system with a single model and hope that this large model will be manageable. If the subsystem approach has been used, the submodels must be linked into an overall system model. The logical interactions of the subsystems must be identified and implemented in a form compatible with the overall model.

If the model is a computer program, as is the case in a simulation model, the modeling phase also involves the selection of the language in which the program is to be written. A number of considerations determine the choice of a programming language, including the difficulty of translating the model and interrelationships into the language, the presence or absence of facilities in the language to support such routine activities as queue management, generation of random numbers, output formatting, and the analyst's familiarity with the language. A crucial step in using a computer program to model a system is the construction of detailed system flowcharts. It provide a visual representation of the program's logic and serves as program documentation after the logic has been translated into code. As a general rule, the better the formulation of the model when the flowchart is made, the easier it is to create a complete computer program model of the system.

Another task in the modeling phase of a simulation study is the estimation of the system variables and parameters. At this point real-world data are summarized into a manageable statistical description of the system's characteristics. This is commonly done by collecting data over some period of time and then computing a frequency distribution for the desired variables. A number of pitfalls await the unsuspecting researchers in the collection and interpretation of data. The cost in terms of time, money, and personnel can become prohibitive; the data may be incomplete or inaccurate; the data may contain unsuspected interdependencies, periodicities, or complexities; the method of collection may introduce an inadvertent bias into the data. Other than the inherent cost of collecting data, most of these pitfalls can be avoided through the use of experimental design procedures.

1.2.3 Validation

A model is validated by proving that the model is a correct representation of the real system. Validation should not be confused with verification. When a computer program is verified, for example, the program is checked to ensure that the logic does what it was intended to do. A verified computer program can in fact represent an invalid model. The program may do exactly what the program intended, yet it may not represent the operation of the real system.

Just as a good experimental design can aid in the data collection in the modeling phase, so can validation aid in correctness of the simulation model. Most standard experimental designs require that observations be taken on the system variables that can be controlled. The simulation model must operate under identical conditions. Only then can valid inferences be drawn about the relationship between the resulting output (observations) of the real system and the outputs of the simulation model. In most designs, several factors can be varied simultaneously, and thus some information on interactive effects of those factors can be obtained. Several effects may be estimated from the same data. Careful design of the simulation experiments will prove beneficial in the long run.

1.3 Advantages and disadvantages of simulation

Simulation is not applicable in many cases, and even in some cases in which it does apply, there may be cheaper and easier ways of solving the problem. Phillips, Ravindran, and Solber (1.2) state that simulation is one of the easiest tools of management science to use but probably one of the hardest to apply properly and perhaps the most difficult one from which to draw accurate conclusions.

Advantages:

1) It permits controlled experimentation. A simulation experiment can be run a number of times with varying input parameters to test the behavior of the system under a variety of situations and conditions.
2) It permits time compression. Operation of the system over extended periods of time can be simulated in only minutes with ultra fast computers.
3) It permits sensitivity analysis by manipulation of input variables.
4) It does not disturb the real system. This is a great advantage, since most managers would be reluctant to try experimental strategies on an on-line system.
5) It is an effective training tool.

Disadvantages:

1) A simulation model may become expensive in terms of manpower and computer time. The cost of a simulation experiment can be minimized through

in-depth understanding of the system being simulated before the model is developed and through careful design of the simulation experiment.

2) Extensive development time may be encountered.
3) Hidden critical assumptions may cause the model to diverge from reality. Ideally this phenomenon should be detected in the validation phase of the simulation process, but it might go undetected, depending on the severity of the problem and the diligence with which the model is validated.
4) Model parameters may be difficult to initialize. These may require extensive time in collection, analysis, and interpretation.

1.4 Simulation terminology

1) About the model

a) A real world object is called an entity.
b) Characteristics or properties of entities are called attributes.
c) Any process that causes changes in a system is called an activity.
d) A description of all the entities, attributes, and activities, as they exist at some point in time is called the state of the system.

2) About the environment

a) The objects and processes (entities and activities) surrounding the system are called system environment.
b) Activities that occur within the system are called endogenous activities.
c) Activities in the environment that affect the system are called exogenous activities.
d) The classification of all activities as either endogenous or exogenous establishes the system boundary.
e) A system with no exogenous activities is called a closed system; otherwise the system is open.

3) About the system

a) Continuous systems include variables that can assume any real value in a prescribed set of intervals; these systems are characterized by smooth changes in the system state.

b) Discrete systems include variables that can assume only particular values from among a finite set of alternatives; these systems are characterized by discontinuous changes in the system state.

c) A system whose response is completely determined by its initial state and input is said to be deterministic.

d) A system whose response may take a system whose response may take range of values give the initial system state and input is said to be stochastic.

4) About the simulation

a) Validation refers to the proof that the model is a correct representation of the real system.

b) Verification refers to the proof that the simulation program is a faithful representation of the system model.

c) Experimental design refers to a sequence of simulation runs in which parameters are varied, with both economy and sound statistical methodology considered in achieving some specified goal.

CHAPTER 2

INTRODUCTION OF SIMULATION IN MANUFACTURING

There has been a dramatic increase in the use of simulation in manufacturing during the past few years. Increased foreign competition in many industries has resulted in a greater emphasis on using automation to improve productivity and quality and also to reduce cost. Since automated systems are often quite complex, they typically can be analyzed only by a powerful tool like simulation. Reduced computing costs and improvements in simulation languages (which have reduced model development time) have also led to increase the use of simulation. Finally, the availability of graphical animation has resulted in greater understanding and use of simulation for engineering managers.

2.1 Simulation: What & Why

2.1.1 Why are simulation models necessary?

It is often useful to study a dynamic real-world system to learn something about its behavior. However, it is generally necessary to use a model to study the performance of the system, since experimentation with the system itself would be disruptive, not cost-effective or simply impossible (e.g., the manufacturing facility has not yet been built). For example, a manufacturing firm that is contemplating building a large extension onto one of its plants may be unsure as to whether the potential gain in productivity would justify the construction cost. An appropriate model could shed some light on this question by allowing the operation of the plant to be studied as it currently existed and as it would be if the plant were expanded.

If the relationships which compose the model are simple enough, it may be possible to use mathematical methods (such as algebra, calculus or probability theory) to obtain the exact answers to the question of interest; this is called an analytic solution.

As an example of an analytic solution, consider the single machine tool. Jobs (or work pieces) arrive at the machine tool for processing. If the machine is idle when a job arrives, processing begins immediately. Otherwise, the job joins the end of a queue. When the machine finishes processing one job, it begins processing the first job in the queue (if any).

Let 'a' be the rate at which job arrive at the machine (i.e., jobs per unit time), and let 'p' be the rate at which the machine can process jobs. If we assume that 'a' is less than 'p', it can be shown under appropriate assumptions that the long-run average time a job spends in the system (in queue plus being processed),'w', is given by

$$w = 1/(p-a).$$

Thus, this formula can easily be used to determine the average time in system for various legitimate values of 'a' and 'p'.

Unfortunately, however, most manufacturing systems are too complex to allow realistic models to be evaluated mathematically, and these models must be studied by means of simulation. In a simulation we use a computer to evaluate a model numerically over a time period of interest, and data are gathered to estimate the desired true characteristics (e.g. throughput) of the model.

We will concentrate on a particular type of simulation called discrete event simulation. This type of simulation is generally stochastic in nature, which means that random samples from probability distributions are used to drive the model through time.

2.1.2 Step in a sound simulation study

There has been an unfortunate impression that simulation is simply an exercise in computer programming, albeit a complicated one. Consequently, many simulation "studies" have been composed of heuristic model building, coding, and a single run of the program to obtain "the answers." This approach neglects the important issues of how to develop a valid model and how to use a properly coded model to draw statistical inferences about the system of interest, and has led to erroneous conclusion being drawn from many simulation studies.

In light of the above discussion, the steps to be follow for sound simulation study and the relationships between them are discussed below. The explanation of those typical steps are as follows:

a) Formulate the problem and plan the study.

* State study objectives clearly.
* Delineate the system designs to be studied (if possible).
* Specify the criteria for comparing alternative system designs.
* Plan the study in terms of the number of people, the cost and the time required for each aspect of the study.

b) Collect data and define a model.

* Data hould be collected on the system of interest (if it exists) to specify input parameters and probability istributions (e.g., a machine repair time distribution).
* Data should be collected (if possible) on the performance of the system (e.g., throughput) to aid in validating the model.
* The level of model detail should be consistent with the study's objectives.

c) Valid?

* Involve people who are intimately familiar with the operations of the system (machine operators, industrial engineers, etc.) in the model building process.
* Analysts should interact with decision maker on a regular basis.

d) Construct a computer program and verify.

* Decision must be made to use a either general purpose language or special simulation languages.
* Traces, structured walk through and animation should be used to debug the model.

e) Make pilot runs.

* Use for validation purposes in Step 6.

f) Valid?

* Pilots runs can be used to test sensitivity of odel's output to small changes in an input parameter.
* Compare output data from an existing ystem (from Step 2) to output data from pilot runs of the same system.

g) Design experiments.

* Specify the system designs to be simulated.
* Specify the number of independent simulation runs for each alternative.
* Specify the length of each run.
* Specify the initial conditions for each simulation run (e.g., initial state of each machine, worker, etc.).

h) Make production runs.

* Simulation runs specified in Step 7.

j) Analyze output data.

* Estimate measures of performance for a particular system design.

* Determine best system design relative to some measure of performance.

k) Documents and implement results.

* Document model's assumptions as well as the computer program.
* Implement the results from the simulation study.

Some studies may not necessarily contain all of these steps in the order stated; some studies may contain steps which are not depicted in the diagram . Furthermore, a simulation study is not a strictly sequential process.

2.2 Objectives in simulation of manufacturing systems

2.2.1 Benefits of simulation in manufacturing

The general benefit of simulation in manufacturing is that it allows a manager to obtain a system wide view of the effect of changes on his or her manufacturing system (whether it exists or not). For example, the effect of adding an additional machine to a work station may be predictable by using simple queuing theory or back-of-the-envelope calculations; however, these techniques probably won't be adequate to determine what effect this change will have on the entire manufacturing system. Increasing the throughput at one work station might cause bottlenecks to develop at one or more other work stations.

Specific (potential) benefit of simulation in manufacturing are increased throughput, reduced in process inventories, increase utilization or machines and workers, increased on-time deliveries and reduced capital requirements.

2.2.2 Manufacturing environments in which simulation is applied

The following are three situations in which simulation is applied in manufacturing:

a) New equipment and buildings are required.

* New plants may required to produce a new product or to increase the production of products.

b) New equipment is required in an old building

* A new product may be produced in all or part of an existing building.

c) Upgrading of existing equipment or its operation.

* Concerned with producing the same product more efficiently.
* Changes may be in the equipment (e.g., ntroduction of a robot) or in operational procedures (e.g., scheduling rule employed).

2.2.3 Manufacturing issues that simulation is used to address

a) The need for and the quantity of equipment and personnel.

* Number and type of machines for a particular objective.
* Number, type and physical arrangement of carts, conveyors and other support equipments(e.g., pallets and fixtures).
* Location and size of inventory buffers.
* Evaluation of a change in product mix (impact of new products).
* Evaluation of the effect of a new piece of equipment on an existing manufacturing line.
* Evaluation of capital investments.
* Manpower requirements planning.

b) Performance evaluation.

* Throughput analysis.
* Makespan analysis.
* Bottleneck analysis.

c) Evaluation of operational procedures:

* Production scheduling (i.e., evaluating proposed policies for loading and sequencing machines).
* Evaluation of policies for component part or row material inventory

levels.

* Evaluation of control strategies (e.g., for an automated guided vehicle system or a flexible manufacturing system).

* Reliability analysis (e.g., effect of planned maintenance). Evaluation of quality control policies.

2.2.4 Measures of performance used in manufacturing simulation studies

a) *Throughput (number of jobs produced per unit of time).*

b) *Time in system for jobs (make-span).*

c) *Times jobs spend in queues.*

d) *Time that jobs spend in queues.* Time that jobs spend being transported.

e) *Sizes of in-process inventories* (WIP or queue sizes).

f) *Utilization of equipment and personnel* (i.e., proportion of time busy).

g) *Proportion of time that a machine is broken, blocked* (i.e., unable to operate until current job is removed), starved (waiting for a job or undergoing preventive maintenance).

h) *Proportion of jobs produced which must be reworked or scrapped.*

i) *Return on investment for a new or modified manufacturing system* (often given in terms of present value).

j) *Payback period* (time to earn back the money invested in a new or modified system).

2.3 Simulation languages for manufacturing system

One of the major tasks in building a simulation model of a manufacturing system is converting a flowchart of model of the system into an actual computer program. A simulation may use either a general-purpose language (e.g., FORTRAN or BASIC) or a simulation languages for thesis purpose.

The advantage of using a language like FORTRAN is that the language is probably already known by the analyst, that it is probably available on the analyst's computer, and that the required computer execution time may be less than for a model written in a simulation language since the program is tailored for the application. On the other hand, simulation models written in a general-purpose language tend to take a long time to develop, since these languages are not particularly oriented toward simulation modeling.

Simulation languages that are applicable to manufacturing problems may be further classified into two categories, general-purpose simulation languages and manufacturing-oriented languages/simulators. General-purpose simulation languages are useful for simulating a wide variety of systems (e.g., computer or military systems) in addition to manufacturing systems, but may contain certain features specifically for manufacturing.

Examples of languages in this category are GPSS, SEE WHY, SIMAN, SIMSCRIPT II.5, SLAM, and TESS. These languages allow an analyst to develop a simulation model of a manufacturing system in less time than would generally be required when using a language like FORTRAN.

Certain simulation software packages have been designed specifically for simulating manufacturing-type problems, including AUTOMOD, MAP/1 AND SIMFACTORY are actually simulators rather than languages, since a particular system within an available class of manufacturing system is modeled by entering data rather than doing actual programming.

The "hottest" feature which is currently available in many simulation languages is the capability for graphical animation of the simulation output. Important elements of a manufacturing systems, such as machines, workers and transporters, are represented by icons on a graphics terminal or CRT.

Every time when there is a change in the state of the simulation, there is a corresponding change in the graphical representation. Thus an analyst or manager can watch manufacturing system change graphically over time.

Some animation software packages operate as post-processors, while others operate as the simulation executes. Among those which operate in real time, several packages allow an analyst to stop a simulation during execution, change the parameters of the simulation and then continue execution for this "new" system design. The following are some potential uses of animation:

* Communicating the nature of a simulation model or its corresponding system to a manager.
* Debugging a simulation computer program
* Showing a simulation model is not valid.
* Training manufacturing personnel on the operation of a "new" system.

2.4 Developing valid and credible simulation models

Ideally, a simulation model would be accurate enough that any conclusions drawn from the model would be the same as those derived from physically experimenting with the system itself (if this were possible). Though it is not know for most simulation models whether this is the case or not. However, it is not necessary to have a one-to-one correspondence between each element of the actual system and each element of the simulation model. Indeed, a simulation model should be designed to meet a specified set of objectives, rather than to be university valid.

The following key ideas and techniques are useful for deciding the appropriate level of model detail, for validating a simulation model and for developing a model with high credibility.

a) *Define the issues to be investigated,* the alternative system designs of interest and the measures of performance for evaluation at the beginning of the study.

b) *Start with a simple model,* which can later be embellished. This allows the analyst to get results to the manager or sponsor in a reasonable amount of time.

c) *Use "experts" and sensitivity analyses* to help determine the level of model detail.

d) *Do not have more detail in the model* than is necessary to address the issues of interest. On the other hand, a model must have enough detail to be credible.

e) *If a similar existing manufacturing system exists,* talk to all important people associated with this system (machine operators, engineers, managers, vendors, etc.) and use this information to build the simulation model.

f) *Interact with the decision makers* (or managers) throughout the course of the simulation to help ensure that the model is both valid and understood.

g) *Perform a structured walk through of the model's flowchart* before an audience of all interested parties and before the actual coding of the program begins.

h) *First, build a simulation model of a similar existing system.* Compare the output data (e.g., throughput) from the model to output data from the actual existing system. If possible, perform this comparison by driving the model with actual shop floor input data (e.g., observed machine repair times).

2.5 Statistical issues in manufacturing simulation

Since the random samples from input probability distributions (e.g., a machine repair time distribution) are used to drive a simulation model through time, the output data form a simulation are also random samples from probability distributions. Therefore, it is important to model correctly the random inputs to a simulation model and also to design and analyze simulation experiments in a proper manner.

2.5.1 Simulation input modeling

Most manufacturing systems contain one or more (input) sources of randomness (random variables). Interarrival times of jobs at a machine, processing times of jobs at a machine, machine running times before breakdown, machine repair times and the outcomes of inspecting jobs (e.g., good, rework, or scrap) are possible examples of random variables in a manufacturing system.

Furthermore, in order to model the system correctly, each random variable must be represented by an appropriate probability distribution in the simulation model. A random number generator is then used to generate random samples from these distributions as the simulation advances through time. (A random number generator is typically a computer based mechanism for generating a "random" value between 0 and 1, with each possible value being equally likely).

It is very important to have an accurate assessment of various distribution and its related parameters when using simulation model as otherwise it can lead to false output data of the actual system.

2.5.2 Design and analysis of simulation experiments

One of the most common and potentially dangerous practices in simulating manufacturing system is that of making only one run (or replication) of a stochastic simulation. For example, suppose that a manufacturing system operates 16 hours a day, than we would like to estimate the mean (or expected) daily throughput or production. If we run the simulation only one time, the value of the throughput from the simulation output is only one observation from a probability distribution whose mean is the desired expected daily throughput. (This is absolutely not different from trying to estimate the mean of a population in classical statistics with only one data point). Furthermore, this single observed value of throughput may differ from the expected daily throughput by a large amount.

There is one additional issue related to the design and analysis of simulation experiments. When simulating manufacturing systems, we are often interested in the long-run behavior of the system; i.e., its behavior when operating in a "normal" manner. (In the above example, we were interested only in the behavior of the system over a 16 hours day.) On the other hand, simulations of manufacturing systems often begin with the system in an empty and idle state. This results in the output data (e.g., daily throughput) from the "beginning" of the simulation not being representative of the desired "normal" behavior of the system. Therefore, simulations are often run for a certain amount of time, the warm-up period, before the output data are actually used to estimate the desired measures of performance.

A graphical approach can be used for determining the length of the warm-up period.

2.6 Simulation analysis of a manufacturing system

Simulation can be iteratively used to design a manufacturing system, for e.g., the distance between stations for reducing material handling time, the number of machines needed in each work station, the buffer capacity for the work station, the maximum allowable queue, etc.

All of the above parameter can be tested for optimum value using simulation.

CHAPTER 3

SELECTION OF SIMULATION SOFTWARE

There has been a dramatic increase in the use of simulation for manufacturing analyses during the past few years. This has been caused by the greater complexity of automated system, reduced computing costs brought about by microcomputers and engineering workstations, improvements in simulation software which have reduced model development time, and the availability of graphical animation which has resulted in greater understanding and use of simulation by engineering managers.

Increased interest in simulation has, in turn, led to an explosion in the number of simulation packages with a strong orientation toward manufacturing problem, with more than 25 such products now being available. As a result, a person trying to select simulation software for his/her organization or for a particular application is now faced with a bewildering variety of choices in terms of technical capabilities, ease of use, and cost. The situation is exacerbated by frequent changes or additions to existing software and by regular introduction of entirely new simulation products. A person new to the field of simulation modeling could literally spend three or more months for carefully evaluating software for a particular simulation project.

The goals of this article are to provide the simulation analyst/engineer with a set of features which should be considered when evaluating simulation software, and also to present the result of an extensive vendor survey for 23 simulation products. In this survey, the vendors were asked to comment on more than 50 characteristics or features of each of their simulation software products.

There is an unfortunate impression that simulation is largely a complicated exercise in computer programming. Thus, in many simulation "studies" a significant amount of the effort is spent on "coding" the simulation model in a simulation package and, also, possibly, in selecting the software in the first place.

In, fact, It is to be believed that model coding will represent only 30 to 40 percent of the total required work in a typical sound simulation study. Other important activities include project formulation, data and information collection (e.g., control logic or a conveyor system), statistical modeling of system randomness such as machine breakdowns, validation of the model, and the statistical design and analysis of the simulation runs.

Furthermore, these tasks are, for the most part, not performed by existing simulation software, regardless of how easy these products are to use. Thus, it is incumbent on the simulation developer or user to have fair amount of expertise in simulation methodology per in addition to the use of one or more simulation products.

The remainder of this article is organized as follows. The next section discusses the major classes of simulation software and also the principal types of manufacturing analyses performed by simulation software and also the principal types of manufacturing analyses performed by simulation. Subsequently, some desirable features for simulation software are discussed. Finally, the results of the vendor survey are given.[11]

3.1 Types of simulation software

There are two major categories of software for simulating manufacturing or warehousing systems. A general-purpose simulation language is a simulation package which is general in nature (e.g., it could also be used for modeling computer or communication systems), but may have special features for manufacturing such as workstations or material handling modules. (There is one simulation language, AutoModII, which is specifically directed toward material handling and manufacturing problems.) A model is developed in a simulation language by writing a program using the language's modeling constructs.

The major strength of simulation languages is the ability of many of them to model almost any kind of manufacturing system, regardless of the complexity of the system's material handling equipment or control logic. Possible drawbacks of simulation languages are the need for programming expertise and the possibly long coding and debugging time associated with modeling complex manufacturing systems.

A manufacturing simulator is a computer package that allows one to simulate a system contained in a specific class of manufacturing system with little or no programming. For example, STARCELL is a simulator oriented toward manufacturing cells.

The particular system of interest (in the domain of the package) is typically selected for simulation by the use of menus or graphics, without the need for programming. The major advantage of a simulator is that "program" development time may be considerably less than that for a simulation language. This may be very important, due to the tight time constraints in many manufacturing environments.

Another advantage is that most simulators have modeling constructs specifically related to the components of a manufacturing system, which is particularly desirable for production personnel. Also, people without programming experience or who use simulation occasionally, (e.g., a manufacturing engineer) often prefer simulators because of their ease of use.

The major drawback of many simulators is that they are limited to modeling only those manufacturing configurations are allowed by their standard features. This difficulty can be largely overcome if the simulator has the ability to "drop down" into a lower-level language (e.g., FORTRAN) to program complicated decision logic.

Note that a complex model developed in a simulation language by an analyst can be made more accessible to manufacturing personnel by adding a flexible, menu-driven "front end" and also tailored output reports. The front end allows one to make a certain set of modifications to the model without programming.

There are two major types of manufacturing analyses for which simulation is used. In a high-level analysis, the system is modeled at an aggregate level and details of the operating or control logic are not included. (A high-level analysis is often performed in the initial phases of manufacturing system design, since detailed system information is not then available).

Typical objectives are determining the required numbers of machines and material handling equipment, evaluating the effect of a change in product volume or mix, and determining storage requirements for work-in-process. Manufacturing simulators are often used for high-level analyses, but a language could be used as well.

A detailed analysis is performed to fine-tune or "optimize" the performance of a system, and the corresponding simulation models typically represent operating or control logic in considerable detail. Most analyses of this type are done for existing systems, because of the need for a precise system description.

An example of detailed analysis would be determining the best operating strategy for a complicated conveyor system. Many detailed analyses are done using a simulation language because of the need to model complex decision logic, which may be unique to the system being studied. In some cases, simulators can also be used, particularly if they have the ability to drop down to a lower-level language.

In addition to the two types of manufacturing analyses discussed above, simulation is increasingly being used to support daily scheduling decisions on the shop floor. FACTOR and INTERFACE are scheduling oriented manufacturing simulators with utilities for accessing the necessary manufacturing databases.[12]

3.2 Desirable features

We now discuss six groups of specific features that , in the opinion of the authors, are important for simulation software to be used in the analysis of manufacturing systems. These groups roughly correspond to the tables of vendor responses presented in the next section; here, however, we will discuss certain features not included in the survey because of their non-quantifiable nature.

3.2.1 General Features

One of the most important features is modeling flexibility, because no two manufacturing system are exactly the same. If the simulation package does not have the necessary capabilities for a particular application, then the system must be approximated resulting in a model with unknown accuracy. For a simulator, it is desirable for parts (or entities) to have general attributes (e.g., part number, due date, etc.), which can be appropriately changed.

Ease of model development is another very important feature, due to the short time frame for many manufacturing analyses. Accuracy and speed of the modeling process will be increased if the package has good debugging aids, such as an interactive debugger and on-line help.

Fast model execution speed is particularly important when the simulation model is to be run on a microcomputer (PC). For a simulation model of a food manufacturing plant, it took seven hours to simulate two weeks of production on a (fast) 16 megahertz PC.

The maximum model size allowed by the simulation package may be an important factor when the model is to be executed on a PC. For some packages, the maximum model size is currently less than 100k bytes. This potential difficulty will become less important factor when the model is to be executed on a PC. For some packages, the maximum model size is currently less than 100k bytes. This potential difficulty will become less important since some vendors are beginning to offer extended model sizes based on the OS/2 operating system.

It is also desirable for software to be compatible across computer classes. Thus, for example, a model could be developed on a PC and executed on a minicomputer or mainframe.

3.2.2 Animation

Animation has become a widely accepted part of the simulation of manufacturing systems. It is particularly useful for communicating the essence of a simulation model (or of simulation itself) to managers or other manufacturing personnel, which greatly increases the model's credibility.

For systems with complex logic, animation may also be useful for "program" debugging, for model validation, and for suggesting new control strategies. Desirable animation features include ease of development user creation of high-resolution icons (using bit-mapped graphics), and smooth movement of icons across the computer screen.[13]

3.2.3 Statistical Capabilities

Since almost all manufacturing system exhibit random behavior, it is imperative for a simulation package to contain good statistical capabilities and for them to be actually used. In general, each source of randomness (e.g., processing times, machine operating times, machine repair times, etc.) needs to be modeled by a probability distribution, not just its mean .

A simulation package should contain a wide variety of standard distributions (e.g., exponential, gamma, and triangular), should be able to use distributions based on observed shop floor data, and should contain a multiple stream random-number generator to facilitate the comparison of alternative system designs.

Since random samples from the input probability distributions "drive" a simulation model through time, simulation output data (e.g., daily throughputs) are also random, and appropriate statistical techniques must be used to design and interpret the simulation runs. A simulation package should contain a command to make independent replication of the model automatically, with each replication using different random numbers, starting in the same initial state, and resetting the statistics to "zero".

It is also desirable to have the ability to specify a warm-up period (at the end of which output statistics are reset to zero) and to construct a confidence interval for a desired measure of performance (e.g., mean daily throughput) in order to determine the statistical accuracy of the simulation results.

3.2.4 Material Handling Modules

Material handling systems are an important part of most modern manufacturing systems and, furthermore, are often difficult to model. Therefore, the availability of flexible, easy-to-use modules for modeling transporters (e.g., forklift trucks), AGVS (including contention for guide paths), conveyors (both transport and accumulating), AS/RS, cranes, and robots can significantly reduce model development time. It should be noted, however, that the existing material handling modules in some simulation packages may not always be sufficient due to the great diversity of available material handling systems.[14]

3.2.5 Customer Support

Most users of simulation software require some level of on going support from the vendor. This can be in the form of general software training or may be in providing technical support for specific modeling problems encountered by the user. Good documentation, including numerous detailed examples, is important for software use as well as initial installation.

3.2.6 Output Reports

It is desirable for a simulation package to provide timesaving standard reports for commonly occurring performance statistics (e.g., utilizations, queue sizes, and throughput), but also allows tailored reports to be developed easily. (For example, standard reports are often not suitable for management presentations.)

Furthermore, it is often of interest to obtain high-quality graphical displays, (e.g., histograms or time plots of important variables) and to have access to the individual model output observations (rather than just the summary statistics) so that additional analyses can be performed.

The above discussion about the types of simulation software, have given a detailed list of features to consider when choosing software for a particular application. However, the choice of a simulation package may still be a difficult decision due to the proliferation of simulation products and their widely varying capabilities and prices. We recommend that potential modelers consider the following activities when making their decisions:

1) Carefully determine the types of manufacturing issues that you want to address by simulation, paying particular attention to the required level of

model detail.

2) Develop a short list of candidate simulation packages ased on your requirements in item (1) above, on features of the available software, and on cost considerations.

3) Talk to several users about each product on your list to get independent assessments of software strengths and weakness.

4) If possible, get a 30-day free trial for each product to see how it performs on applications of particular interest to you.

There is no simulation package which is completely convenient and appropriate for all manufacturing applications. Thus, organizations that perform a large amount of simulation may want to consider having several simulation packages, which are used for different types of analyses and by people with different backgrounds.[15]

CHAPTER 4

PITFALLS TO AVOID IN THE SIMULATION OF MANUFACTURING SYSTEM

The use of simulation to design and "optimize" manufacturing and warehousing systems continues to increase at a rapid pace. However, there is a common impression that simulation is largely a complicated exercise in computer programming. Thus, in many simulation "studies" the major emphasis is on simulation software selection and on model "coding".

In fact, simulation modeling is a sophisticated system analysis activity, and tee model coding represents only 30 to 40 percent of the total effort in a typical sound simulation study. Careful attention must also be paid to such activities as problem formulation, data and information collection (e.g., control logic for material handling equipment), probabilistic modeling of system randomness such as machine breakdowns, developing a model which is both valid and credible, and the statistical design and analysis of simulation experiments. If these important project activities are ignored, there is a significant likelihood that the model will produce erroneous results, or the study's conclusions will not be used in the decision-making process (even if they are correct).[12]

The pitfalls are broken into four categories: the model development process, the selection and use of simulation software, the modeling of system randomness, and the design and analysis of simulation experiments.

4.1 Modeling and Validation

Developing a simulation model of complex manufacturing system requires a certain amount of skills and expertise in order to manage the overall project effectively and also to decide what elements of the real system should be included in the model. The following pitfalls refer to the process of building and validating a simulation model.

Pitfall No.1: Failure to have a well defined set of objectives.

There has been a number of organizations embark on simulation studies without a clear statement of project goals. This is partly due to a lack of understanding of simulation and of the types of information that it can provide.

Every simulation project should begin with a definitive specification of overall objectives and also of the particular manufacturing issues to be addressed by the model. Simulation models are not universally valid, and the appropriate level of model detail can be addressed only when a precise statement of goals is available.

It is also important to identify significant performance measures (e.g., throughput or machine utilizations) since a model may be capable of providing an accurate estimate of one measure, but not another. Project goals should be set at an initial meeting which includes managers and all key project personnel.

Pitfall No.2: Inappropriate level of model detail.

In general, there should not be one-to-one correspondence between every element of the model and every element of the corresponding system. A model should have just enough detail to address correctly the manufacturing issues identified during project formulation, and for the model to be credible. If the model is not detailed enough, any conclusion drawn from the simulation study will be of doubtful validity.

Conversely, if the model has unnecessary detail or if the basic entity (or "part") moving through the model is too "small", the model execution time or memory requirements may be excessive, particularly on a micro computer. (In a simulation model for a bakery products manufacturing, we chose the basic part to be a box of cookies rather than an individual cookie to ensure reasonable computer execution times.)

Pitfall No.3: Failure to interact with management on a regular basis.

It is very important to interact with the manager or client and other key personnel on a regular basis throughout entire study. This helps ensure that the correct problem is being solved and that the manager's interest in the project is being maintained. More importantly, the model becomes more credible, since the manager understands and accepts its assumptions. In fact, it is desirable to have the manager and other "important" people "sign off" on key assumptions. This enhanced credibility will increase the likelihood of the model's results actually being used for decision making.[10]

Pitfall No.4: Insufficient simulation and statistics training.

Since model "coding" typically represent less than 50% of the work for a sound simulation study, it is necessary for the analyst to have a fair amount of expertise in areas other than the use of a simulation package. In particular, the analyst, needs to have formal training in simulation methodology, including validation techniques, selection of input probability distributions, and interpretation of simulation output data.

This subjects in turn requires a solid grounding in statistics and probabilistic modeling. This knowledge is required regardless of the simulation package used. Furthermore, the necessary training is usually not provided by seminars on a particular simulation product , but is available in university courses, and in public short courses specifically on simulation techniques.

4.2 Simulation / Animation software

The selection of an appropriate simulation package can have a big impact on the ultimate validity of the model and on the timeliness with which the project is completed.

Pitfall No.5: Inappropriate simulation software.

If the simulation package used for the study doesn't have sufficient modeling flexibility, the manufacturing system of interest will have to be approximated, resulting in a model of unknown accuracy. Thus, model results may not provide a reliable indication of actual system performance. It is also desirable for a simulation package to have a user-friendly environment which promotes rapid model development, and for the software to be usable by people without a high level of programming expertise.

Pitfall No.6: Misuse of animation.

Animation is certainly a powerful tool for communicating the essence of a simulation model to management, and in some cases it can aid in the debugging and validation process. However, the persuasive nature of a high quality animation can sometimes promote a false sense of security about the goodness of the model.

In particular, since only part of the model's logic can be portrayed in an animation, it is not possible to assess model correctness solely on the basis of the animation. Also, the efficacy of a particular manufacturing system design cannot be determined, in general, by watching the animation for a "short" amount of time. Rather, a careful statistical analysis of the simulation output data must be performed.

4.3 Modeling system randomness

Most manufacturing systems contain one or more (input) sources of randomness (random variables). Processing times of jobs at a machine, assembly times, machine running times before breakdown, machine repair times, set-up times, and the outcomes of inspecting jobs (e.g., good, rework, or scrap) are possible examples of random variables in a manufacturing system.

Furthermore, in order to model the system correctly, each random variable must be represented by an appropriate probability distribution in the simulation model, as the following pitfalls show.

Pitfall No.7: Replacing a distribution by its mean.

Simulation analyst sometimes represents a random variable in a simulation model by its postulated mean value, rather than using the corresponding probability distribution itself. This practice may be due to either a lack of definitive data on which to base an intelligent distribution selection or a misunderstanding of the impact of randomness on system performance measures.

To illustrate the potential danger of using only the mean, consider a manufacturing system consisting of a single machine tool at which jobs arrive to be processed. Suppose that the mean interarrival time of jobs is one minute and the mean processing time is 0.99 minute. Suppose further that the interarrival times and service times actually have an exponential distribution. Then it can be shown (Reference of Law and Kelton) using basic queuing theory that the long-run mean number of jobs waiting in the queue is approximately 98.

On the other hand, if we assume that each interarrival time is exactly one minute and each processing time is exactly 0.99 minutes (i.e., we eliminate all randomness), no job ever waits in the queue. The variability of the probability distributions, rather than just their means, has a significant impact on the congestion level in most queueing type system, e.g., manufacturing system.

Pitfall No.8: Using the wrong probability distribution.

It is generally necessary to model each source of system randomness by a probability distribution. However, it is also important to use the "correct" distribution. For example, consider the above single machine tool system with exponential interarrival times. Suppose that processing times actually have a gamma distribution which has the same general shape as histograms of processing times typically experienced in practice.

Suppose, however, that a normal distribution is used instead in the simulation model. The normal distribution is probably the most familiar distribution to most engineers since it is almost always learned in a first statistics course. Surprisingly, simulation results reported in Law and Vincent show that the performance measure mean number in queue will be 33 % in error if the normal rather than the gamma distribution is used. Errors of this magnitude are likely to result in incorrect simulation project decisions.

Pitfall No.9 : Incorrect modeling of machine down times.

The largest source of randomness in many manufacturing systems is which is associated with machine breakdowns. The following example shows that care must be taken when modeling breakdowns.

Suppose that a company is going to buy a new machine tool from a vendor who claims that the machine will be down 10% of the time. However, the vendor has no data on how long the machine will operate before breaking down or on how long it will take to repair the machine.

Some simulation analysts have accounted for random breakdowns by simply reducing the machine processing rate by 10%. This may produce quite inaccurate results.

4.4 Experimental design/analysis

One of the most important (and often neglected) aspect of a simulation study is the design and analysis of simulation experiments. Since random samples from the input probability distributions "drive" a simulation model through time, simulation output data (e.g., daily throughput) are also random. Thus, as the following discussion shows, output results must be interpreted carefully.

Pitfall No. 10: Misinterpretation of simulation results.

Assume that a manufacturing system operates 16 hours a day and that we would like to estimate the mean (or expected) daily throughput, than we run the simulation only one time, the value of the throughput from the simulation output is only one observation from a probability distribution whose mean is the desired expected daily throughput. (This is absolutely no different from trying to estimate the mean of a population in classical statistics with only one data point.) Furthermore, this single observed value of throughput may differ from the expected daily throughput by a large amount.

Pitfall No.11: Failure to account for the warm-up period.

When simulating manufacturing systems, we are often interested in the long-run behavior of the system; i.e., its behavior when operating in a "normal" manner. On the other hand, simulations of manufacturing systems often begin with the system in an empty and idle or some other unrepresentative state.

This results in the output data from the beginning of the simulation not being representative of the desired "normal" behavior of the system. Therefore, simulations are often run for a certain amount of time, the warm-up period, before the output data are actually used to estimate the desired measures of performance. Use of these warm-up period data would bias the output results.

CHAPTER 5

SIMULATION PROGRAM FOR MANUFACTURING CELL USING SIMAN

SIMAN simulation language is use to write a simulation program for manufacturing cell. The logical flow chart to design the SIMAN simulation program for the manufacturing cell is as shown in figure 1. The program was written using the HOOPS graphics software and SIMAN simulation language. With this developed software, the user does not have to be fully experienced about the SIMAN simulation software. The user will enter the necessary input data such as:

a) machining sequence.

b) machining time and its distribution.

c) material handling time and its distribution .

d) batch size.

e) interarrival time and distribution between subsequent batch.

f) run time for simulation program.

g) no. of runs, etc.

which will automatically create the experimental file necessary to run the SIMAN simulation program. Once the experimental file has been generated the simulation program can be run to obtain the simulation results. The necessary information about the software developed is attached in the appendix.

5.1 Introduction to SIMAN

SIMAN is a general purpose SIMulation ANalysis programming language for modeling combined discrete-continuous systems. The modeling framework of SIMAN allows component models based on three distinct modeling orientations to be combined in a single system model. For discrete change system either a

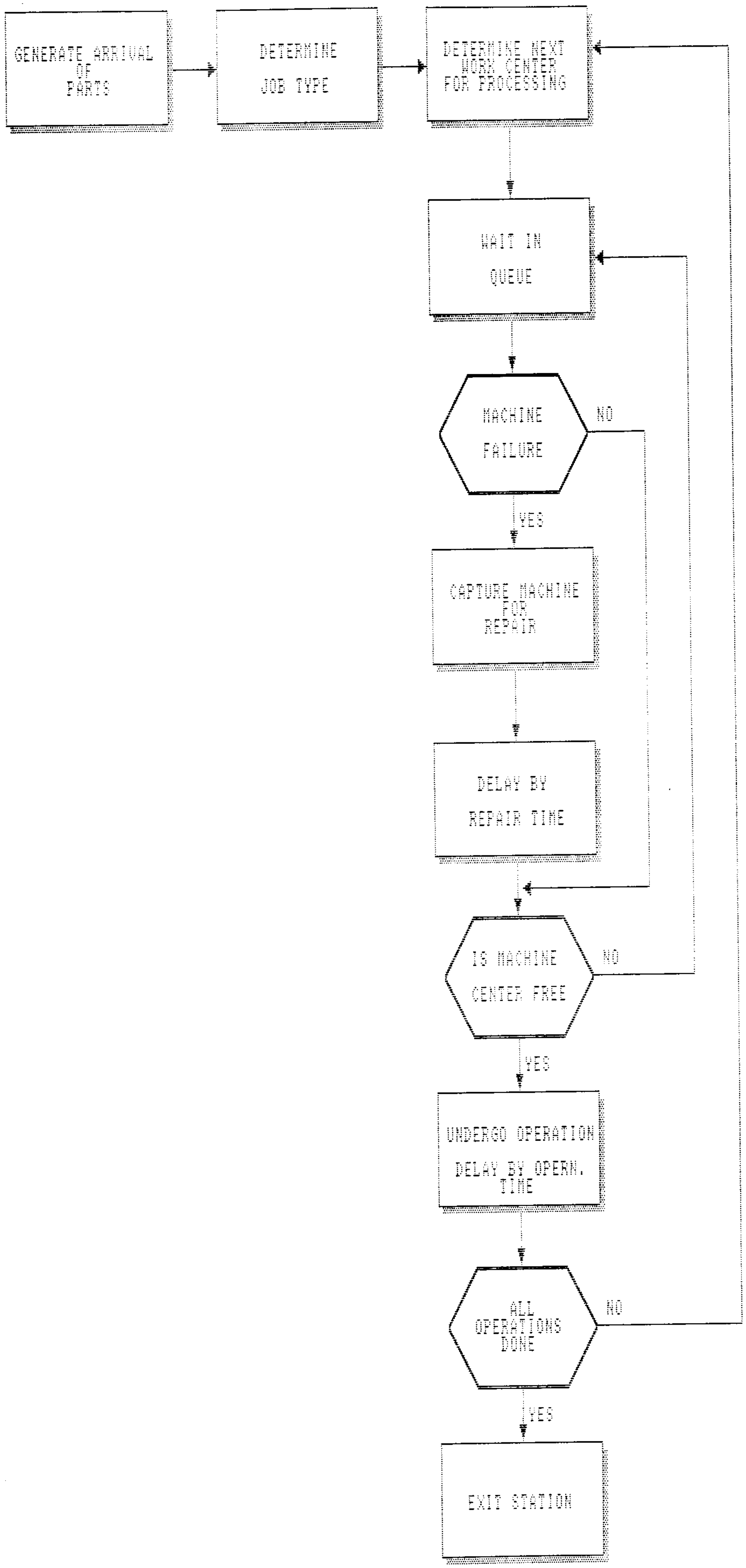


Figure 1: Logic flow diagram for job shop

process or event orientation can be used to describe the model. Continuous change systems are modeled with algebraic or differential equations. A combination of these orientations can be used to model combined discrete-continuous models.

SIMAN is designed around a logical modeling framework in which the simulation program is decomposed into the system model frame and experimental frame. The system model defines the static and dynamic characteristics of the system under consideration, whereas the experimental frame defines the experimental conditions under which the model is run to generate specific output data. For a given model, there can be many experimental frames resulting in many sets of output data. Since the model structure and experimental frame of the program are two distinct elements, different simulation experiments can be performed by changing only the experimental condition in the experimental frame. The system model remains the same.

The SIMAN simulation program when executed, keeps the record of the model state transitions as they occur in specified simulated time and generates output files as desired by the user. The data in the output files can than be subjected to various data analysis such as data truncation and compression, and the formatting and display of histograms, plots, tables, etc. One output file can be subjected to many different data treatments without re-executing the simulation program. Data treatments can also be applied to sets when performing an analysis based on multiple runs of a model or when comparing the response of two or more systems or comparing it to analytical results.

5.2 The Software Structure

As shown in figure 2, a SIMAN simulation is divided into three distinct activities: system model development, experimental frame development, and data analysis. Within these three activities, the SIMAN software consists of five individual processors which interact through four data files.

1. The model processor is used to construct a block diagram model. The data file that is generated is called the model file.
2. The experiment processor is used to define the experimental frame for the system model. The data file that is generated is called the experiment file.
3. The link processor combines the model file and the experiment file to produce the program file.
4. The program file is input to the run processor which executes the simulation runs and writes the results on the output file. If the system model includes an event or continuous model, the user written FORTRAN subroutines are linked to the run processor before the simulation runs are executed.
5. The output processor is used to analyze, format and display the data contained in the output file.

The above mentioned five independent processors within the SIMAN software simplify the framework by separating the distinct functional activities of simulation. The computer memory requirements are reduced due to only one of the five processors is executing at one time.

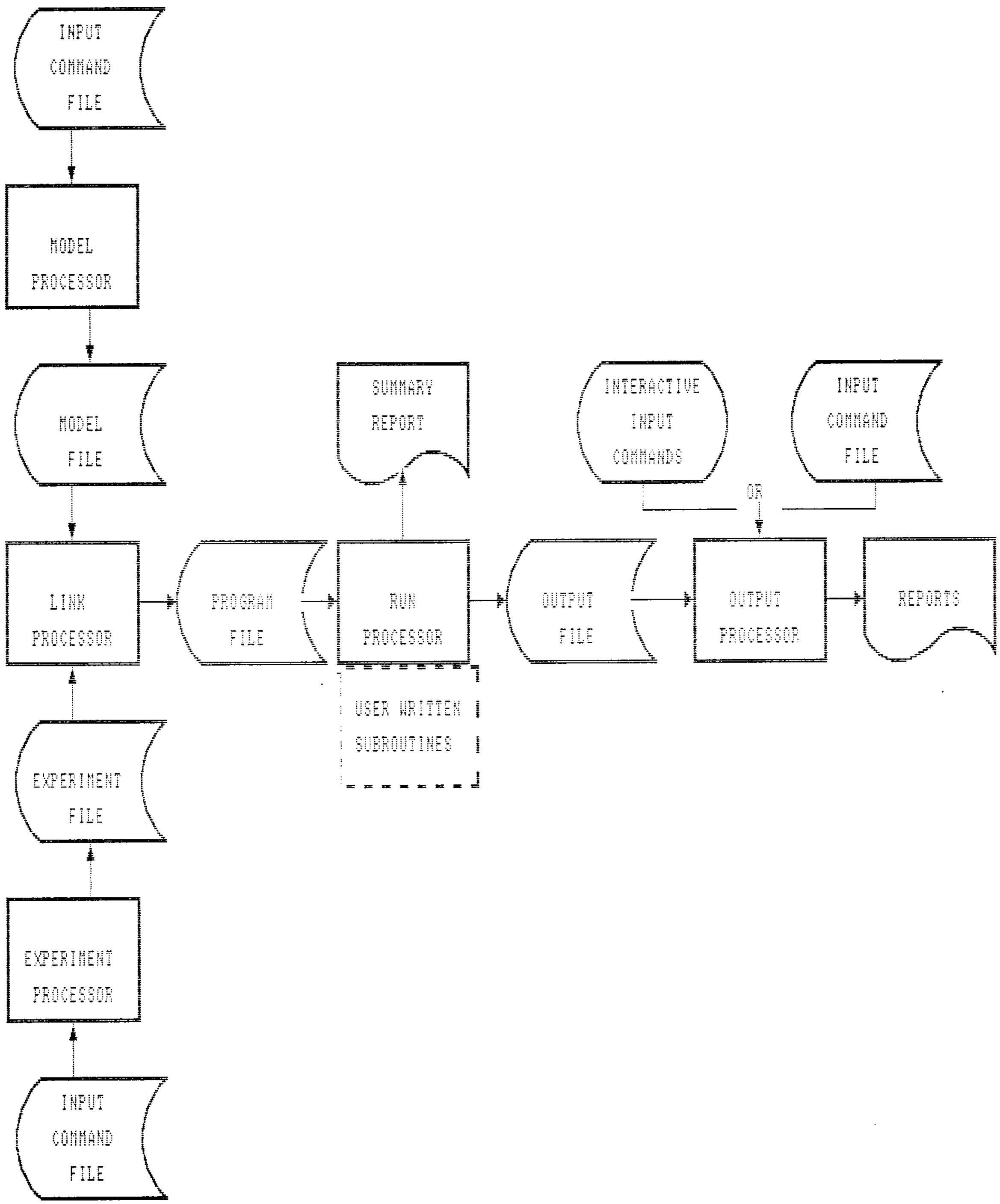


Figure 2: SIMAN Software Structure

5.3 Modeling Manufacturing System

The process orientation is best suited for modeling most manufacturing systems. This uses block diagrams which are linear top-to-down flow graphs that depicts the flow of entities through the system. The block diagram is constructed as a sequence of blocks whose shapes indicate their function. The sequencing of blocks is depicted by arrows which control the flow of entities from block to block through the entire diagram.

These entities are used to represent "things" such as workpiece, information, people, etc. which flows through the real system. Each entity may be individualized by assigning attributes to describe or characterize it. For example, an entity representing product to be manufactured might have attributes corresponding to processing time and transportation time for the workpiece. As the entities flow from block to block, they may be machined, inspected, delayed, disposed, combined with other entities, etc., as determined by the function of each block.

The general purpose attributes of an entity are denoted by the real array A(I). In the above mentioned example, A(1) could be machining time and A(2) the transportation time between stations. Each entity has its own unique attribute array which is "carried along" with the entity as it flows from block to block within a model.

There are ten different basic block types in SIMAN. The symbol and function of each of ten types of blocks are summarized in figure 3 & 4.

The OPERATION, HOLD and TRANSFER blocks are further subdivided into several different block functions depending upon their operation type, hold type or transfer type. These types are specified as the first operand of the block and consist of a verb which is descriptive of the specific function which the block is to perform. For example, the operation type CREATE specifies that the block

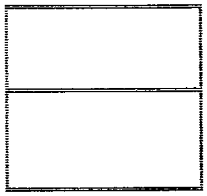
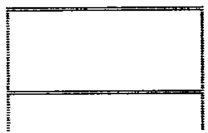
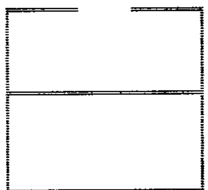
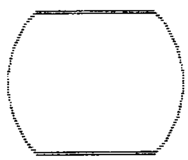
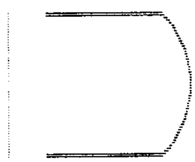
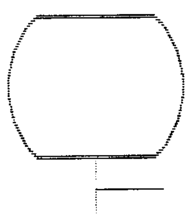
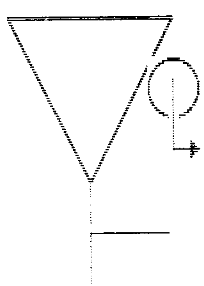
NAME	SYMBOL	FUNCTION
OPERATION		The OPERATION block is used to model a wide range of processe such as time delays, attribute assignment, etc.
TRANSFER		The TRANSFER block is used to model transfers between stations via material handling systems.
HOLD		The HOLD block is used to model situations in which the movement of an entity is delayed based on system status. The HOLD block must be preceded by a queueing facility to provide a waiting space for delayed entities.
QUEUE		The QUEUE block provides a waiting space for entities which are delayed at following HOLD or MATCH block.
STATION		The STATION block defines the interface points between model segments and the material handling systems.
BRANCH		The BRANCH block models the conditional probabilistics and deterministic branching of entities.
PICKQ		The PICKQ block is used to select from a set of following QUEUE blocks.

Figure 3: SIMAN basic block types

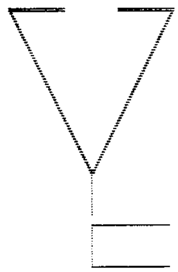
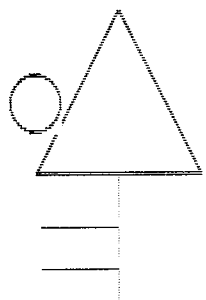
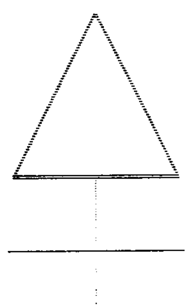
NAME	SYMBOL	FUNCTION
SELECT		The SELECT block is used to select between resources associated with a set of following operation blocks.
QPICK		The QPICK block is used to select from a set of preceding QUEUE blocks
MATCH		The MATCH block delays entities in a set of preceding QUEUE blocks until entities with the same value of a specified attribute resides in each QUEUE.

Figure 3.1: SIMAN basic block types (Cont'd)

is to create entities which flows from block to block; and the operation type ASSIGN specifies that the block is to assign a value to an attribute or variable; and the hold type PREEMPT that hold the entity until one resource unit is allocated to the entity; or the transfer type ROUTE which route the entity to a specified station.

Each block function of SIMAN is referenced by a block function name. In the case of the OPERATION, HOLD and TRANSFER blocks, each basic block type perform several different functions whereas in the use of QUEUE, STATION, BRANCH, PICKQ, QPICK, SELECT, and MATCH blocks, each block type performs only one function and the block function name is the same as the basic block name.

All the basic block types of SIMAN have operands which control the function of the block. For example, the CREATE block has operands, CREATE,NB:TBC,MC:MODIFIERS; which prescribe the number of entities per batch arrival, the time between batch arrivals, and maximum number of batches to create, respectively. Block may also optionally be assigned with a block label, and a comment line. A block label is appended at the lower left side of a block and can consist of up to eight alphanumeric characters. It is used for branching or referencing from other blocks of the model file. The comment line , if entered, must be on the right side of the block. It helps in understanding the actual function of that block.

A block diagram model can be defined in the form of either statement model or diagram model. The statement model is a transcription of the block diagram model into statement form for input to the model processor. The diagram model is a graphic representation of the system using the ten basic block symbols of SIMAN. There is one to one correspondence between blocks in the diagram model and the statement in the statement model.

If the diagram model is used to create a model for the system then BLOCK program automatically generates the corresponding statement of the model for the input to the SIMAN language.

5.4 Characteristics of Manufacturing Systems

Manufacturing systems exhibit a number of unique characteristics which makes it very difficult and time consuming to model within the framework of a general purpose simulation language. The most significant of these characteristics are as follows:

1. Large manufacturing systems are typically comprised of a number of different workcenters or cells. A natural way to model such system is to decompose the large system into its workcenters, modeling each workcenters separately, and then combine the workcenters models into an overall system model. The general purpose simulation languages typically do not provide a logical format for modeling separate workcenters.
2. Often several workcenters within the manufacturing system are functionally equivalent. As a result, it is possible to develop a single functional description which can be used to model all the similar Workcenters within the system. General purpose simulation languages generally do not provide any features to exploit this property.
3. The workbenches which move through a manufacturing system typically each have a unique process plan. This means that each entity in the model must have its own routing sequence through the Workcenters as well as its own setup, processing time, tool requirements, etc., within the Workcenters. Since the general purpose simulation languages do not provide any feature for this, a considerable amount of effort can be consumed in incorporating logic within the model fo r maintaining process plans on each

workpiece and controlling the flow from workcenter to workcenter.

4. In most manufacturing systems, there is an uneven distribution of workload between workcenters. One way that this varying workload is accommodated is through the use of different operating schedules for the different workcenters based upon their workstations. Hence, it is desirable to be able to assign each workcenter an operating schedule which it will follow during the simulation. Again, this is often awkward to do with general purpose simulation languages.
5. An essential element of most manufacturing systems is the material handling component. This includes devices such as robots, AGV's, conveyors, power and free monorail systems, etc. These types of devices can be extremely difficult to model with general purpose simulation languages.

It should be noted that although in theory a general purpose simulation language can given enough effort accurately model these characteristics of manufacturing systems, the modeling effort involved can be enormous.

5.5 Manufacturing Modeling Features of SIMAN

Modeling Workstations

In manufacturing systems, it is frequently desirable to model distinct workcenters within the system. This can be done within SIMAN by employing the STATION block which defines the beginning of a station submodel. An entity is entered into the STATION block using a TRANSFER block. The TRANSFER block is used to represent entity movements between station submodels.

Each station submodel is referenced by a positive integer which is called the station number. This number corresponds to the physical location within the system. The station number is an operand of both the STATION block and the TRANSFER block.

When an entity enters a STATION block, the entity's station attribute, M, is set by SIMAN to the station number of STATION block. The entity carries this special attribute with it as it proceeds through the sequence of blocks which comprises the station submodel. The entity remains within the station submodel until it is disposed, or it is sent to a new station submodel via a TRANSFER block.

The block sequence within a station submodel defines the processes through which the entities flow. The processes normally involve the queueing of entities due to limited resources such as operations, tools, etc.

Macro Submodels

One particularly useful feature for modeling workcenters in SIMAN is the macro submodel. This powerful feature permits the development of a single macro submodel to represent a set of two or more similar yet distinct workcenters. For example, a typical jobshop consists of several different workcenters (lathes, planers, etc.) which are functionally equivalent, and differ only in their number and type of machines, buffer sizes, etc. We can model such a jobshop by constructing a single macro submodel which represents the process encountered by a job at a general jobshop workcenter. This single macro submodel can then be used to model a jobshop of arbitrary size.

The beginning of a macro submodel is defined by a STATION block. The range of station numbers represented by the macro submodel is specified as the operand of the block. An entity is entered into the macro submodel by sending it to the STATION block using a TRANSFER block. All entities sent to a station number in the specified range of the STATION block are processed as arrivals to the block. Upon entering the STATION block, the macro station attribute M of the entity is set by SIMAN to the station number to which the entity was sent.

The station attribute can also be used to specify a resource to be seized or released through the use of indexed resources. An indexed resource allows a single name to be assigned to a set of two or more different resource types, with each resource in the set having its own capacity. The resource types within the set are distinguished by an index appended to the resource name. For example, MACHINE(1) and MACHINE(2) represent two distinct resources with different capacities. These resources are completely independent other than sharing the common name MACHINE.

Visitation Sequences

A workpiece is sent to its next workcenter using a TRANSFER block. However, the TRANSFER block must have some way to determine which workcenter is next in sequence for a particular workpiece. In addition, it may be necessary to update one or more attributes of the workpiece to correspond to the processing parameters at that workcenter. For example, if the general purpose attribute A(1) was used to specify the processing time on the machine, than the additional function of the TRANSFER block is to update attribute A(1) correspond to the processing time at the next workcenter.

The workcenter visitation sequence and corresponding attribute update values are specified in SIMAN using the SEQUENCES element which is included as part of the experimental frame. The following SEQUENCES element defines two different visitation sequences.

```
SEQUENCES:1,5,EX(1,1)/3,UN(2,1):
        2,4,UN(3,1)/2,10.3/6,EX(4,1):
```

Sequence number 1 consists of two workcenter visits. The first visit is to station number 5 and assigns A(1) a sample from an exponential distribution. The second visit is to station number 3 and assigns A(1) a sample from a uniform distribution. Sequence number 2 consists of visits to stations 4 then 2 and then 6 with the assignments to attribute A(1) as shown.

Each workpiece in the system has two special attributes which are used in conjunction with the SEQUENCES element to determine its next station and attribute update values at the TRANSFER block. The first special attribute is NS which specifies the number of the visitation sequence which the workpiece is to follow. This value is typically assigned to the entity when it enters the model. The second special attribute is IS which keeps track of the current index within the sequence. An index value of K means that the workpiece is at the Kth workcenter within its visitation sequence. The index attribute is automatically updated by SIMAN whenever the entity arrives to a TRANSFER block.

Additional attribute could be employed to specify setup times, special tool requirements, etc., which might be part of the process plan. In addition, by resetting the value of IS for a given entity within a workcenter submodel, a portion of a given sequence could be repeated or skipped. Likewise by resetting the value of NS, the sequence which the workpiece follow could be changed.

Resource Schedules

The workcenters within a manufacturing system often operates according to different work schedules as a result of their differing loads. Within, SIMAN this characteristic can be easily modeled through the use of the SCHEDULES element which is included in the experimental frame. The SCHEDULES element is used to define a work schedule by specifying a resource capacity over time. A resource capacity within the model can then be directed to follow a given work schedule. For example, resources in workcenter 1 might be directed to follow schedule number 1 and the resources in workcenter number 2 might be directed to follow schedule number 2.

The following SCHEDULES element defines two different work schedules:

```
SCHEDULES:1,1*8,0*16
        2,1*EX(1,1),0*UN(2,1);
```

In schedule number 1, the capacity is 1 for 8 time units, then 0 for 16 time units, and then this cycle repeats. In schedule number 2, the capacity is 1 for a duration which is sampled from an exponential distribution, and then 0 for a duration which is sampled from a uniform distribution, and then this cycle repeats.

Modeling Material Handling System

Within a manufacturing system, the movement of entities between workcenters is accomplished by the material handling system. In simple terms, the function of material handling is the movement of material from one point to another. There is a large variety of material handling devices which have been developed to support this function.

APPENDIX

REFERENCE GUIDE FOR

SIMULATION OF MANUFACTURING CELL

USING HOOPS GRAPHICS SOFTWARE AND

SIMAN SIMULATION LANGUAGE

INTRODUCTION

This document is a guide to use the graphical software called Simulation of manufacturing cell. It is intended to be use as a reference guide. The design of the manual emphasizes compact descriptions of the features of the software components with only simple supporting examples.

This document provides pertinent information about software features in a format that is well organized and easy to reference. In the following section, it describes the important information about each components of the software to simplify the use of software.

SCREEN LAYOUT

As shown in the figure , the main menu will be at the bottom of the right corner. The panel board which is use for entering the data will be located above the main menu. Depending upon number of products and number of components for each product, there will be three pages at the most, and each page will contain 2 products. Each component of each product will have the following four inputs segments:

1) Quantity to be produce.

2) Machining sequence.

3) Machining time and.

4) Material handling time.

Page up button to move to the next page and Page down button to move to the previous page are located at the top and bottom right corner of the screen, respectively.

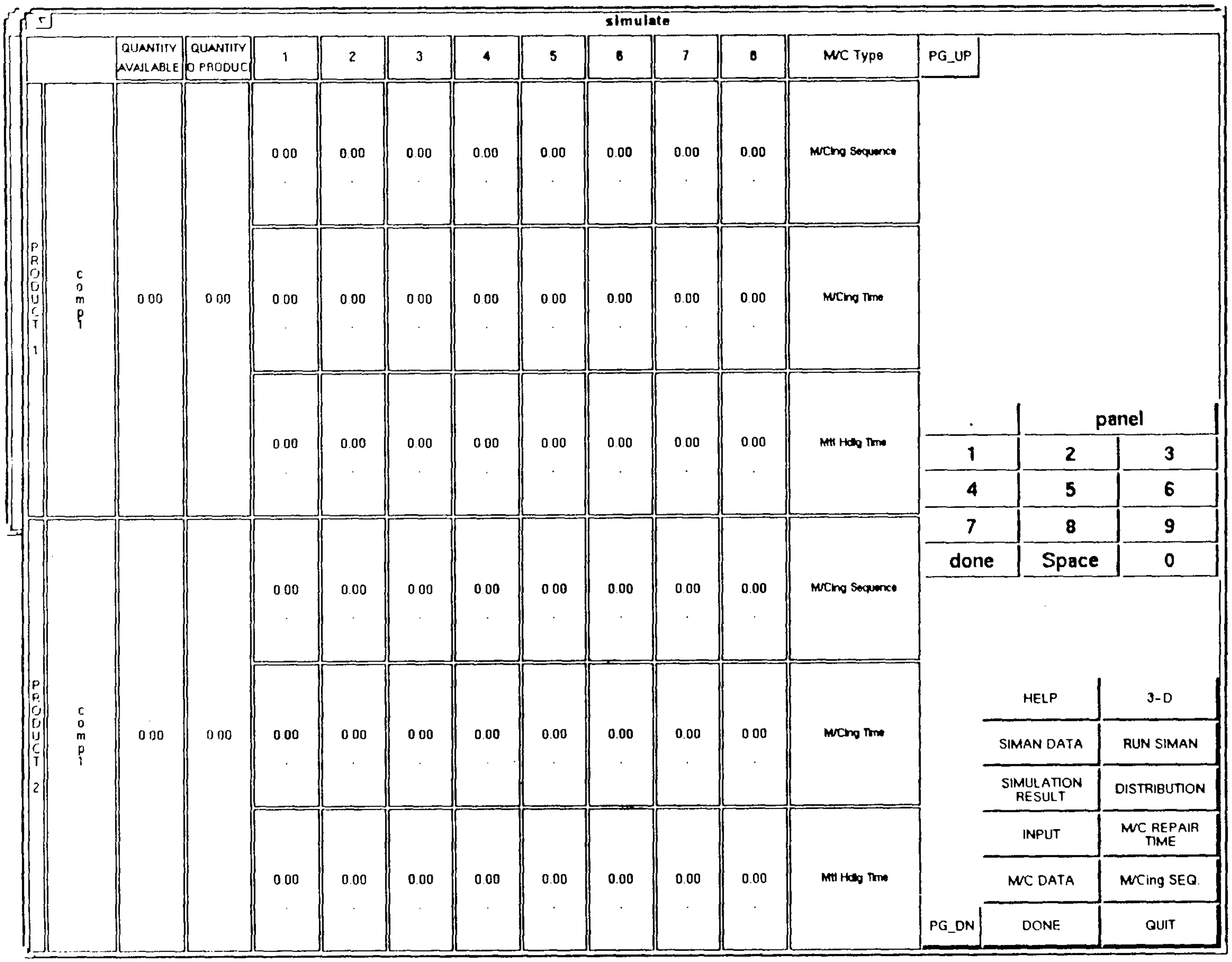


Figure 4: Screen layout of developed software

M/Cing SEQ

Enter the sequence of machining operations to be performed on a component.

DESCRIPTION

When a Button M/Cing SEQ is selected from the main menu the user is asked to enter the machining sequence of a selected component. To enter the sequence of operation, select the corresponding segment under the machine number (marked at the top of the screen)in the Machining Sequence row for a desired component . Using the panel board enter the sequence number for that machine.

Once the particular component is selected user can not enter the sequence of operation for another component. To do so, user must finish entering M/Cing sequence for selected component ,then select DONE and M/Cing SEQ from main menu, respectively.

ERROR MESSAGE

While entering the machining sequence user may get the following two types of Error messages:

1) *Conflicting in machining order*

This error occurs when user enters the same sequence number for the two different machine. The machine which was previously set to that sequence number will be automatically set to zero.

2) *Machining Sequence is incomplete*

This error occurs when user selects DONE from the main menu without entering the complete machining sequence.

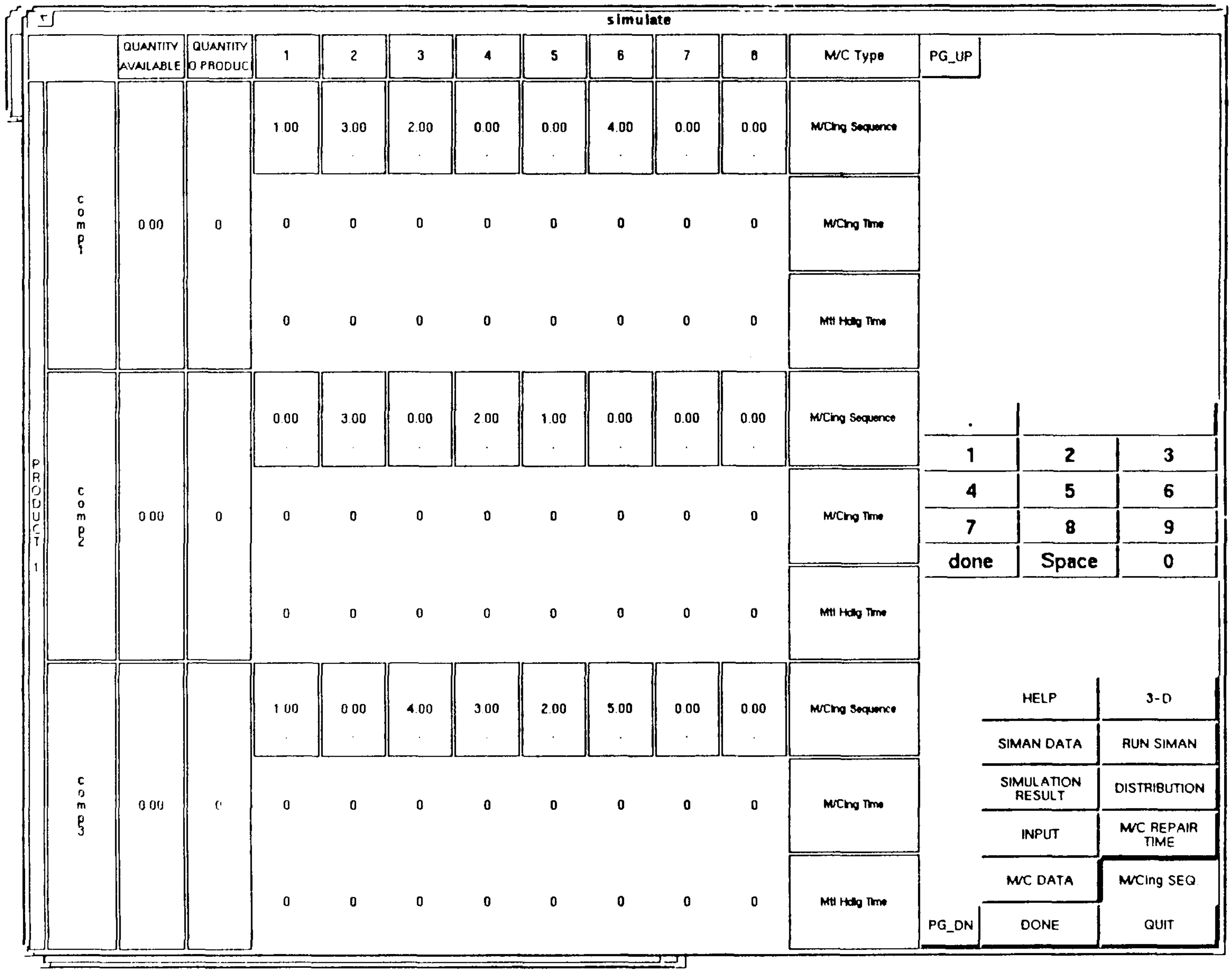


Figure 5: Input of Machine sequence

INPUT

Enter the machining time for each component on that machine. Enter the material handling time of the component from one machine to another machine between the sequence of operation. Enter the quantity of product to be produced.

DESCRIPTION

When a button INPUT is selected the user is asked to enter the machining time, material handling time and quantity to be produced for each component. To enter the machining and material handling time select the corresponding segment under the machine number (marked at the top of the screen) in the M/Cing Time and Mtrl Hdlg Time row for a desired component. Using the panel board enter the machining and material handling time for the component.

If two or more component has the same machining and/or material handling time than the respective segments can be selected simultaneously before entering the time for the component. Thus, facilitate the multiple input of machining and material handling time. The same is true for the quantity to be produced for each component. Select DONE from main menu after entering the machining and material handling time for all the segments respective to sequence of operation for each component.

ERROR MESSAGE

The following error message may occur while entering the machining and material handling time.

1) *Warning: Segment limit has been reached*

This error occurs when the number of segments selected are more than twenty. Enter the value for those segments and then proceed further.

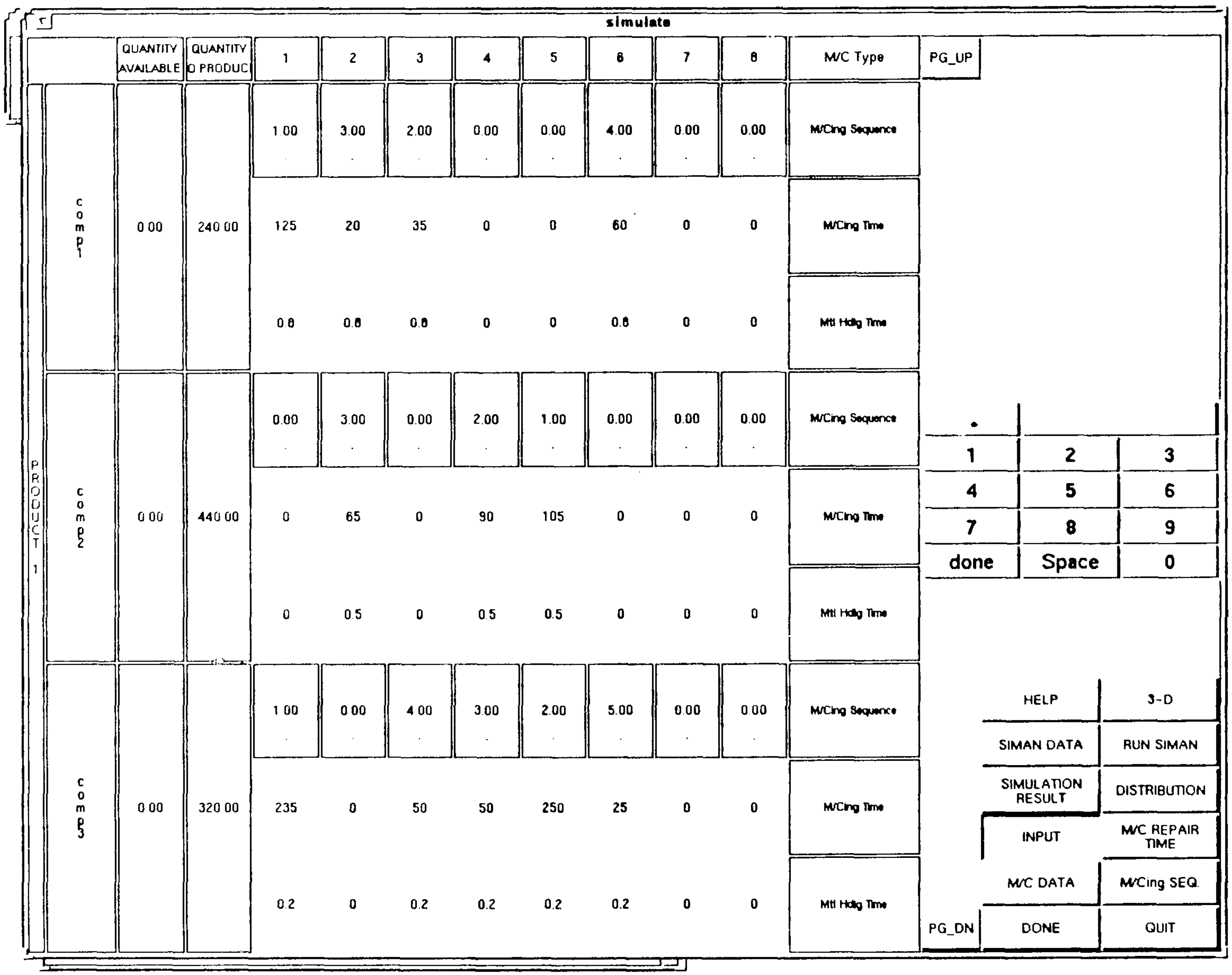


Figure 6: Input of M/Cing and Mtl. Hdlg time

DISTRIBUTION

Enter the distribution (***EX***ponential, ***UN***iform, ***R***a***N***dom, ***CO***nstant) for machining time, material handling time and interarrival time.

DESCRIPTION

Select DISTRIBUTION button from the main menu. The window with the most commonly used four of the above distribution will pop up at the top right corner of the screen. To enter the distribution for machining and material handling time, first select the distribution and than select the segments in M/Cing Time and Mtrl Hdlg Time row that has the respective distribution. If more than one segment has the same distribution, at that time all the segment can be selected simultaneously.

To apply other distribution, Select DONE after completing the previous distribution and only than select the required distribution. To finish distribution click DONE twice.

ERROR MESSAGES

The following error messages may appear while applying the distribution.

1) *This distribution requires only one parameter.*

The ***EX***ponential and ***CO***nstant distributions requires only one parameter. Hence, above stated error occurs when the user tries to assign this distribution to a segment which has two parameters.

2) *This distribution requires two parameters.*

This error occurs when the user tries to assign either the ***NO***rmal or the ***R***a***N***dom distribution to a segment which has only one parameter as an input.

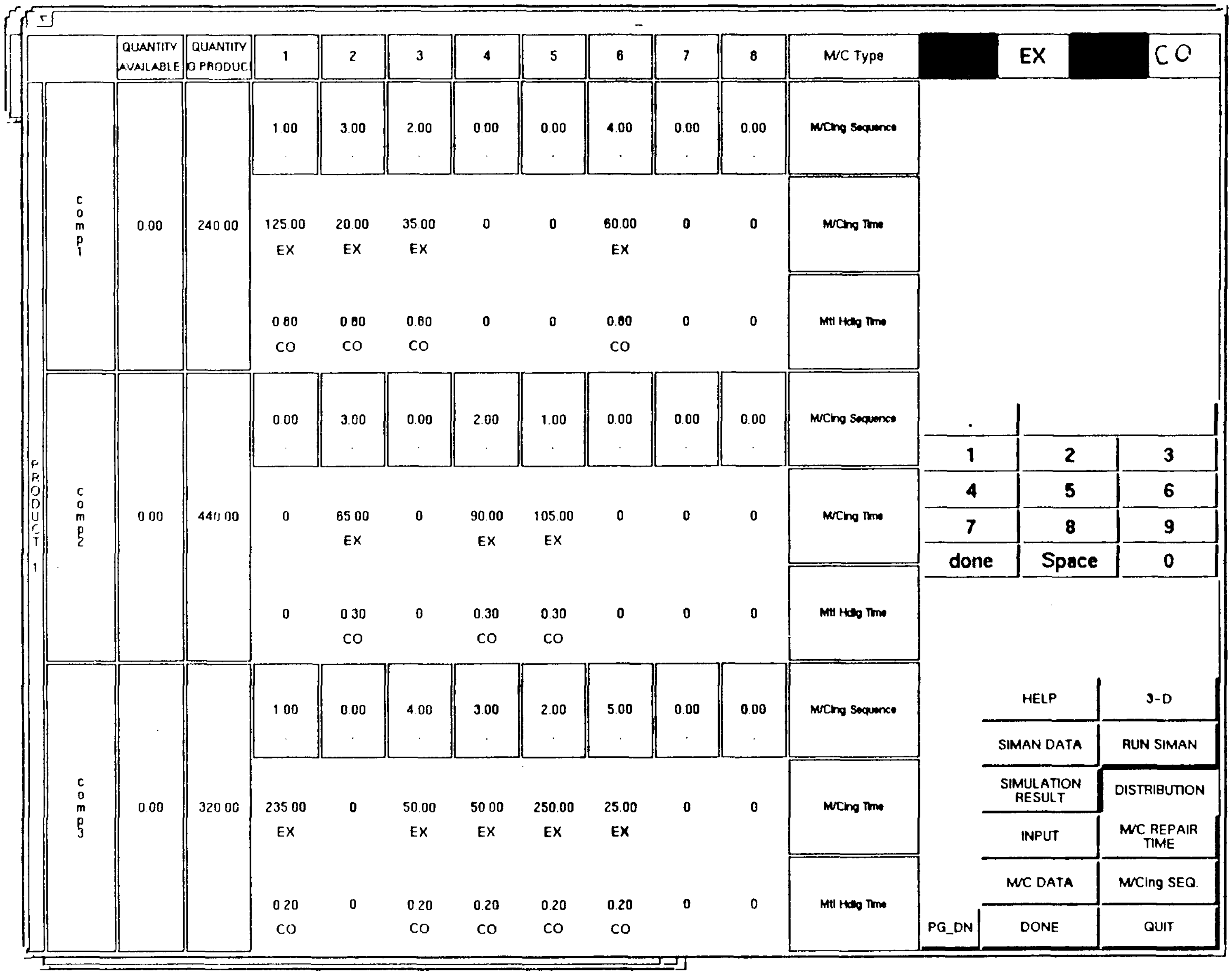


Figure 7: Applying Distribution

M/C DATA

Enter the quantity of each type of machine, its efficiency and its buffer capacity.

DESCRIPTION

When the button M/C DATA is selected, the window displaying the M/C Type, M/C Qty, M/C Eff and Buffer Capacity pops up at the top right corner. In M/C Qty enter how many numbers of each machine type is on the manufacturing cell. Under the M/C Eff heading, enter the efficiency of each type of machine and under Buffer Capacity enter the maximum number of unit that a buffer can hold.

To enter the data, select the segment using the mouse and than using the panel board input the valid data for that segment. If more than one segment has the same value than first select all those segments and than enter the value using the panel board. Select DONE from main manu to end entering the M/C Data. The default value for the machine quantity for each type of machine is 8, the efficiency of each type of machine is 90% and the buffer capacity at each machine is 1000.

ERROR MESSAGE

The following error message may appear on the screen.

1) *Please make an appropriate selection*

This error occurs when user tries to click the button or segment which is not related to the M/C data input.

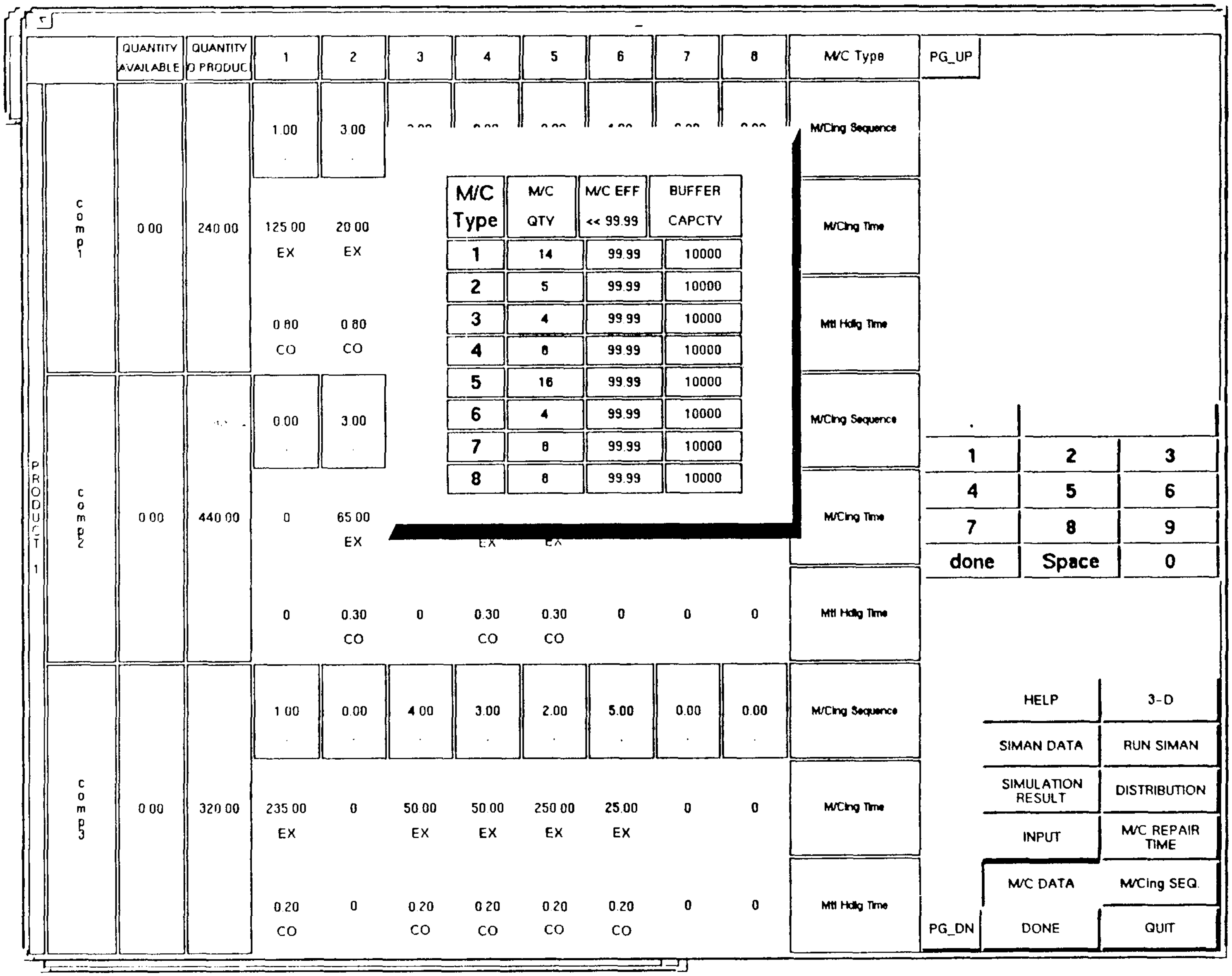


Figure 8: Input of Machine related data

M/C REPAIR TIME

Enter the mean time for repairing each type of machine during the machine breakdown.

DESCRIPTION

When the button M/C REPAIR TIME is selected from main menu window pops up in the center of the screen. Number 1 through 8 specifies each type of machine. To enter the machine repair time select the segments in the M/C Repair time row corresponding to the machine number and than using the panel board enter the value. If more than one machine has the same repair time, the user can select all of those segments simultaneously and than enter the value. Select DONE from main menu to end entering the machine repair time.

The default value for the machine repair time is 10 time units and it's distribution is exponential.

ERROR MESSAGE

While inputing the M/C repair time, the user may get the following error message.

1) *Please make an appropriate selection*

This error occurs when user tries to click the button or segment which is not related to the M/C REPAIR TIME input.

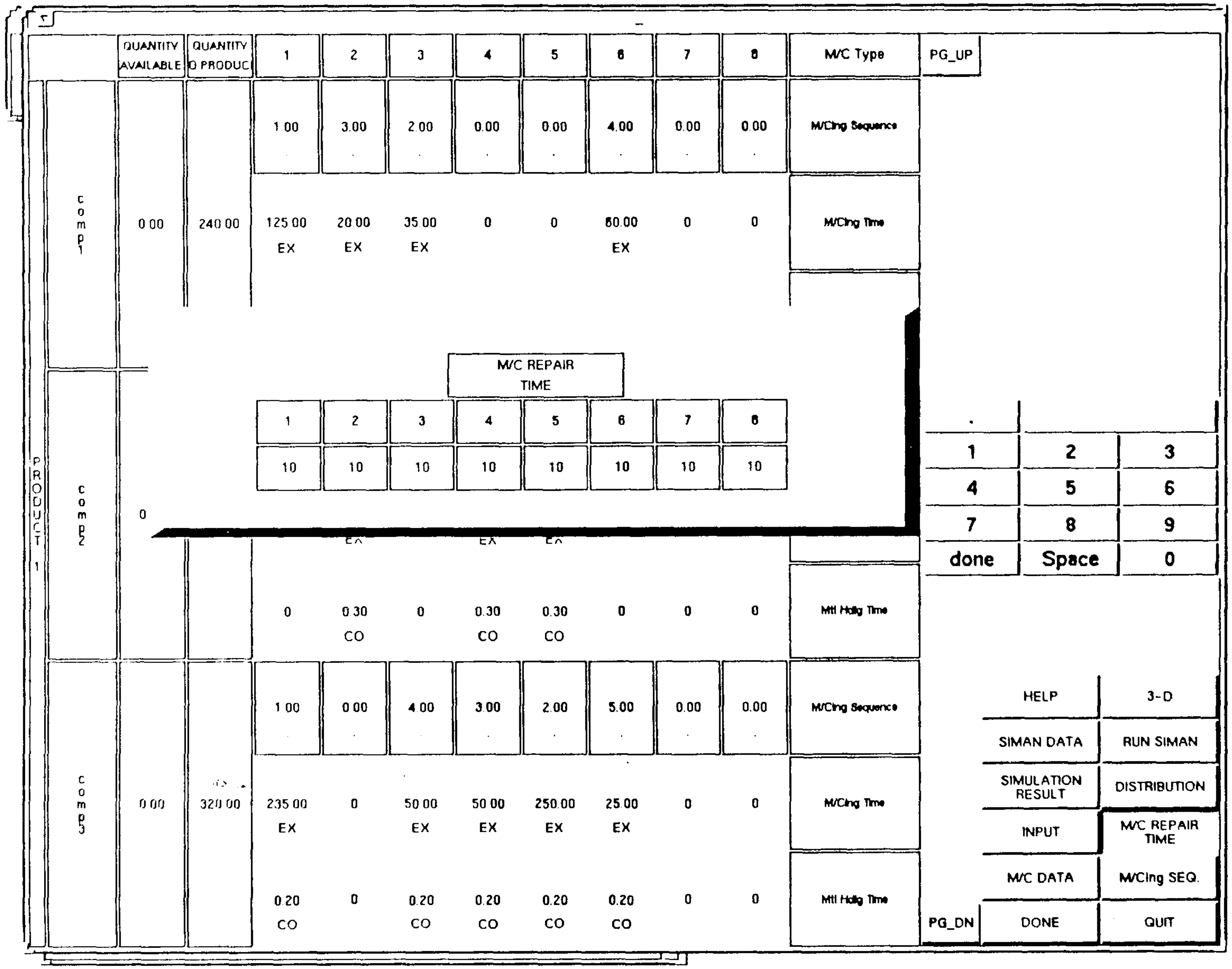


Figure 9: Input of Machine repair time

SIMAN DATA

Enter the data for Run Length, Maximum Entities, No. of Run, Batch Size and Interarrival Time and Distribution, that are necessary to create the experimental file to run the Siman simulation program.

DESCRIPTION

When the button SIMAN DATA is selected, the window with the above Siman Data pops up in the center of the screen. Select the button for which the data is to be entered. Enter the value using panel board. Select DONE from the main menu and than again select the button for which the value is to be entered.

To apply the distribution for the Interarrival time select the DISTRIBUTION from the main menu and follow the procedure explained in DISTRIBUTION.

After finish entering all the data, do not forget to create an EXPERIMENTAL FILE for running the Siman program. To do so click on the EXPERIMENTAL FILE button. Make sure that all the data has been entered correctly , otherwise simulation results generated will be misleading.

NO. OF RUN

Enter the number of times the simulation program t be run. For each replication, the siman will run the program using the different random number. The default value for no. of run is 1.

RUN LENGTH

Enter the maximum length of simulation run. The default value for run length is 4800 time units.

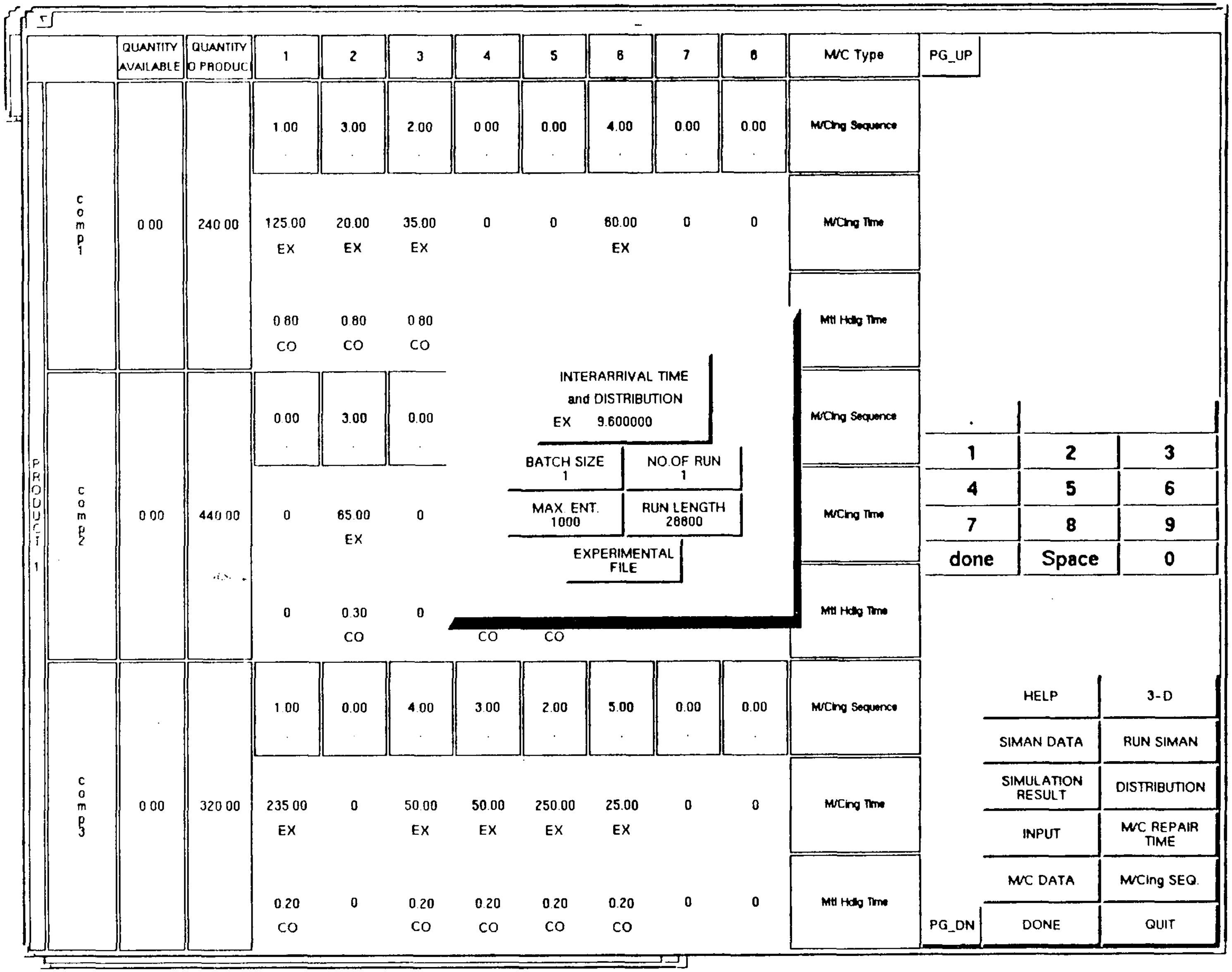


Figure 10: Input of SIMAN related data

BATCH SIZE

Enter the maximum number of entities in each batch. The default value for the batch size is 1.

MAXIMUM ENTITIES

Enter the maximum number of concurrent entities in the system. If you attempt to create an entity when the data pool is empty or not enough, the simulation run will end with the message "MAXIMUM NUMBER OF ENTITIES EXCEEDED". This means that the value specified must be increased. The default value for the discrete element is 1000.

INTER ARRIVAL TIME AND DISTRIBUTION

Enter the time and its distribution between the subsequent arrivals of the batch. The default distribution for interarival time is constant.

EXPERIMENT FILE

This will create the experimental file necessary to run the SIMAN simulation program.

ERROR MESSAGES

1) *Please make an appropriate selection*

This error occurs when the button selected is not within the window that popped up while clicking the button SIMAN DATA.

2) *Warning : Use the correct sequence of entering the Data*

When user follows the wrong sequence of entering the data (i.e. If user try to select the other button within the window without selecting DONE from main menu) , the above error occurs.

The correct sequence is, first user should select the button for which the value is desire to enter than input the value using the panel board and confirm the value entered by clicking on DONE from the main menu. Select the other button to enter the data and follow the same procedure as mentioned.

If any error occurs while applying distribution, refer to DISTRIBUTION.

RUN SIMAN

Simulation program written in SIMAN Simulation language will be compiled using the experimental file which was generated by the user.

DESCRIPTION

When the button RUN SIMAN is selected, the Siman simulation program will run for a specified length of time (mentioned in the experimental file) and generates the output file.

To generate the experimental file, refer to SIMAN DATA.

ERROR MESSAGE

The following error message may appear on the screen.

1) *Experimental file is not present*

The experimental file must be generated (refer to SIMAN DATA to create the experimental file).

SIMAN RESULTS

It will show the output of the simulation results that is generated by running the Siman program.

DESCRIPTION

Simulation result shows the percentage of machine efficiency, machine breakdown and machine starved for component over the time period (Run length) . It will also show the number of each type of component arrived and how many of them are machined, the maximum queue at each type of machine, and the minimum and maximum time for the component in the system.

ERROR MESSAGE

The possible error message that may appear is,

1) *Simulation result is not present*

This error occurs when simulation program is not compiled and user try to view the simulation result.

The program consists of entering the various input related to manufacturing system (using the mouse) as explained in the Appendix. When the user enters all the necessary data, the experimental file as shown in the information on page 77 and 78 at the end of the Appendix will be generated. Than using the general model file on page 75 and 76 for the manufacturing cell, the simulation program will run for a desired period of time. The output result for the problem starts on the page 79 in the Appendix.

```
         BEGIN;
;
         CREATE,X(1):ED(2):MARK(3);           CREATE ARRIVAL OF BATCH
         ASSIGN:NS = DP(1,1);                 CREATE JOB TYPE
         ASSIGN:A(4)=NS;                      ASGN VAL TO FIND TIME IN SYS
         COUNT:A(4),1;                        COUNT NO. OF EACH COMPONENTS
         ASSIGN:A(5)=A(4)+X(50);             X(50)=NO OF PRODUCTS
;
LOOP     ROUTE:A(2),SEQ;                      ROUTE TO PROCESS SEQUENCE
         STATION,1-8;                         ASSIGN STATION NUMBER
         ASSIGN:X(M+9)=0;                     SET STARVE FLAG OFF
         BRANCH,1:                           !CHECK FOR BLOCK CONDITION
           IF,NR(M+9).GE.MR(M+9),BLOCKD:     !MACHINE BLOCKED
           ELSE,QBUF;                         NOT BLOCK
BLOCKD   ASSIGN:X(M+17)=1;                    SET BLOCKED FLAG ON
QBUF     QUEUE,M+8;                             QUEUE FOR MACHINE M
         SEIZE:BUFFER(M);
         ASSIGN:X(M+17)=0;
;
         QUEUE,M;
         SEIZE:MACHINE(M);                    SEIZE MACHINE M
         RELEASE:BUFFER(M);
         ASSIGN:X(M+1)=1;                     SET MACHINE BUSY
         DELAY: A(1);                         DELAY BY MACHINING TIME
         ASSIGN:X(M+1)=0;                     SET MACHINE IDLE
         BRANCH,1:                           !CHECK FOR STARVED CONDITION
           IF,NQ(M).LE 0,STARVE:             !MACHINE STARVED
           ELSE,RELEM;                        MACHINE NOT STARVED
STARVE   ASSIGN:X(M+9)=1;                     SET STARVED CONDITION
RELEM    RELEASE:MACHINE(M):                 !RELEASE MACHINE M
         NEXT(LOOP);                          !ROUTE TO NEXT STATION IN SEQ.
;
         STATION,9;                           EXIT STATION
         COUNT:A(5),1;
         TALLY:A(4),INT(3):DISPOSE;           TALLY TIME FOR THE BATCH ;
;
;        FAILURE LOGIC
;
         CREATE,1:0,8;                      CREATE FAILURE ENTITIES
         ASSIGN:S(50)=S(50)+1;                SET MACHINE COUNTER
         ASSIGN:A(6)=S(50);                   SAVE MACHINE CENTER NUMBER
         ASSIGN:M=A(6)+17;                    SET STATION NUMBER
         STATION,18-25;                       FAILURE STATIONS
;
UPTIME   ASSIGN:                             !CALCULATE MTBF
         P(6,1)=(P(3,A(6))/(1-(P(2,A(6))/
         100.0)))-P(3,A(6));                  FOR PARMETER SET
         DELAY:ED(3);                         DELAY FOR UPTIME
         QUEUE,M;                             QUEUE FOR BREAKDOWN
         SEIZE:MACHINE(M-17);                 GET MACHINE FOR BREAKDOWN
         ASSIGN:X(M-8)=0;                     RESET STARVED FLAG
         ASSIGN:X(M)=0;                       RESET BLOCKED FLAG
         ALTER:MACHINE(M-17),-1;              SET MACHINE TO FAIL MODE
         RELEASE:MACHINE(M-17);               SET MACHINE TO ACTIVE
         ASSIGN:P(7,1) = P(3,A(6));           CALCULATE MTR
         ASSIGN:X(M+8) = 1;                   SET FAIL FLAG
         DELAY:ED(4);                         DELAY FOR REPAIR
         ASSIGN:X(M+8)=0;                     RESET FAIL FLAG
         BRANCH,1:
         IF,NR(M-8).GE.MR(M-8),BLOK:
         ELSE,NOBLOK;
;
BLOK     ASSIGN:X(M)=1;
;
NOBLOK   BRANCH,1:                           !CHECK FOR STARVED
           IF,NQ(M-17).LE.0,STRV:            !STARVED
```

```
              ELSE,NOSTRV;                          NOT STARVED
;
STRV        ASSIGN:X(M-8)=1;                        RESET STARVED FLAG
;
NOSTRV      ALTER:                                 !TURN MACHINE BACK ON AND
              MACHINE(M-17),+1:
                NEXT(UPTIME);                      SEND BACK TO START OF LOOP
;
END;
```

```
BEGIN;
PROJECT,MACHINE CELL,VSP,8/12/93;
DISCRETE,1000,10,61,61;
INITIALIZE,X(1)=1,X(50)=3;
RESOURCES:1-8,MACHINE,14,5,4,8,16,4,8,8:
          9,EXIT STATION:
          10-17,BUFFER,10000,10000,10000,10000,10000,10000,10000,10000;
COUNTERS:
1,COMP1 OF PRD1 ARR,9999:
2,COMP2 OF PRD1 ARR,9999:
3,COMP3 OF PRD1 ARR,9999:
4,COMP1 OF PRD1 MED,9999:
5,COMP2 OF PRD1 MED,9999:
6,COMP3 OF PRD1 MED,9999;
SEQUENCES:
!
!
1,1,EX(8,1),CO(9)&
3,EX(10,1),CO(11)&
2,EX(12,1),CO(13)&
6,EX(14,1),CO(15)&
9:
2,5,EX(16,1),CO(17)&
4,EX(18,1),CO(19)&
2,EX(20,1),CO(21)&
9:
3,1,EX(22,1),CO(23)&
5,EX(24,1),CO(25)&
4,EX(26,1),CO(27)&
3,EX(28,1),CO(29)&
6,EX(30,1),CO(31)&
9;
;
DISTRIBUTIONS:1,CO(4):2,EX(5,1):3,EX(6,1):4,EX(7,1);
;
PARAMETERS:
!
1,
0.240000,1,
0.680000,2,
1.000000,3:
2,99.99,99.99,99.99,99.99,99.99,99.99,99.99,99.99:
3,10,10,10,10,10,10,10,10:
4,9:
5,9.600000:
6,0:
7,0:
8,125.00:
9,0.0:
10,35.00:
11,0.0:
12,20.00:
13,0.0:
14,60.00:
15,0.0:
16,105.00:
17,0.0:
18,90.00:
19,0.0:
20,65.00:
21,0.0:
22,235.00:
23,0.0:
24,250.00:
25,0.0:
26,50.00:
```

```
27,0.0:
28,50.00:
29,0.0:
30,25.00:
31,0.0;
TALLIES: 1,TIME IN SYSTEM FOR COMP1:
       2, TIME IN SYSTEM FOR COMP2:
       3, TIME IN SYSTEM FOR COMP3;
;
DSTAT:      1,NQ(1),MACHINE 1 QUEUE:
     2,NQ(2),MACHINE 2 QUEUE:
     3,NQ(3),MACHINE 3 QUEUE:
     4,NQ(4),MACHINE 4 QUEUE:
     5,NQ(5),MACHINE 5 QUEUE:
     6,NQ(6),MACHINE 6 QUEUE:
     7,NQ(7),MACHINE 7 QUEUE:
     8,NQ(8),MACHINE 8 QUEUE:
     9,X(2),MACHINE 1 EFFI:
     10,X(3),MACHINE 2 EFFI:
     11,X(4),MACHINE 3 EFFI:
     12,X(5),MACHINE 4 EFFI:
     13,X(6),MACHINE 5 EFFI:
     14,X(7),MACHINE 6 EFFI:
     15,X(8),MACHINE 7 EFFI:
     16,X(9),MACHINE 8 EFFI:
     17,X(26),MACHINE 1 FAIL:
     18,X(27),MACHINE 2 FAIL:
     19,X(28),MACHINE 3 FAIL:
     20,X(29),MACHINE 4 FAIL:
     21,X(30),MACHINE 5 FAIL:
     22,X(31),MACHINE 6 FAIL:
     23,X(32),MACHINE 7 FAIL:
     24,X(33),MACHINE 8 FAIL:
     25,X(10),MACHINE 1 STARVE:
     26,X(11),MACHINE 2 STARVE:
     27,X(12),MACHINE 3 STARVE:
     28,X(13),MACHINE 4 STARVE:
     29,X(14),MACHINE 5 STARVE:
     30,X(15),MACHINE 6 STARVE:
     31,X(16),MACHINE 7 STARVE:
     32,X(17),MACHINE 8 STARVE:
     33,X(18),MACHINE 1 BLOCKED:
     34,X(19),MACHINE 2 BLOCKED:
     35,X(20),MACHINE 3 BLOCKED:
     36,X(21),MACHINE 4 BLOCKED:
     37,X(22),MACHINE 5 BLOCKED:
     38,X(23),MACHINE 6 BLOCKED:
     39,X(24),MACHINE 7 BLOCKED:
     40,X(25),MACHINE 8 BLOCKED:
     41,NR(10),BUFFER 1:
     42,NR(11),BUFFER 2:
     43,NR(12),BUFFER 3:
     44,NR(13),BUFFER 4:
     45,NR(14),BUFFER 5:
     46,NR(15),BUFFER 6:
     47,NR(16),BUFFER 7:
     48,NR(17),BUFFER 8;
;
REPLICATE,1,0,28800;
END;
```

SIMAN IV - License #8810506
New Jersey Institute of Tech.

Summary for Replication 1 of 3

Project: MACHINE CELL — Run execution date : 8/17/1993
Analyst: VSP — Model revision date: 8/12/1993

Replication ended at time : 28800.0

TALLY VARIABLES

Identifier	Average	Variation	Minimum	Maximum	Observations
TIME IN SYSTEM FOR COM	297.29	.56765	33.686	1149.6	762
TIME IN SYSTEM FOR COM	294.75	.53723	18.343	1056.4	1302
TIME IN SYSTEM FOR COM	672.50	.52615	87.740	2571.2	885

DISCRETE-CHANGE VARIABLES

Identifier	Average	Variation	Minimum	Maximum	Final Value
MACHINE 1 QUEUE	1.4480	2.3139	.00000	23.000	.00000
MACHINE 2 QUEUE	.57305	2.6535	.00000	12.000	.00000
MACHINE 3 QUEUE	.62789	2.3623	.00000	10.000	.00000
MACHINE 4 QUEUE	1.0756	2.4509	.00000	15.000	.00000
MACHINE 5 QUEUE	.94700	2.5303	.00000	17.000	5.0000
MACHINE 6 QUEUE	.45187	2.6818	.00000	10.000	.00000
MACHINE 7 QUEUE	.00000	--	.00000	.00000	.00000
MACHINE 8 QUEUE	.00000	--	.00000	.00000	.00000
MACHINE 1 EFFI	.64144	.74766	.00000	1.0000	1.0000
MACHINE 2 EFFI	.59191	.83033	.00000	1.0000	.00000
MACHINE 3 EFFI	.60808	.80282	.00000	1.0000	.00000
MACHINE 4 EFFI	.60340	.81073	.00000	1.0000	1.0000
MACHINE 5 EFFI	.62137	.78061	.00000	1.0000	1.0000
MACHINE 6 EFFI	.56540	.87673	.00000	1.0000	1.0000
MACHINE 7 EFFI	.00000	--	.00000	.00000	.00000
MACHINE 8 EFFI	.00000	--	.00000	.00000	.00000
MACHINE 1 FAIL	.00000	--	.00000	.00000	.00000
MACHINE 2 FAIL	.00000	--	.00000	.00000	.00000
MACHINE 3 FAIL	.00000	--	.00000	.00000	.00000
MACHINE 4 FAIL	.00000	--	.00000	.00000	.00000
MACHINE 5 FAIL	.98521E-04	100.74	.00000	1.0000	.00000
MACHINE 6 FAIL	.00000	--	.00000	.00000	.00000
MACHINE 7 FAIL	.00000	--	.00000	.00000	.00000
MACHINE 8 FAIL	.00000	--	.00000	.00000	.00000
MACHINE 1 STARVE	.35784	1.3396	.00000	1.0000	.00000
MACHINE 2 STARVE	.40093	1.2224	.00000	1.0000	1.0000
MACHINE 3 STARVE	.38314	1.2689	.00000	1.0000	1.0000
MACHINE 4 STARVE	.39297	1.2429	.00000	1.0000	.00000
MACHINE 5 STARVE	.37867	1.2809	.00000	1.0000	.00000
MACHINE 6 STARVE	.42538	1.1623	.00000	1.0000	.00000
MACHINE 7 STARVE	.00000	--	.00000	.00000	.00000
MACHINE 8 STARVE	.00000	--	.00000	.00000	.00000
MACHINE 1 BLOCKED	.00000	--	.00000	.00000	.00000
MACHINE 2 BLOCKED	.00000	--	.00000	.00000	.00000
MACHINE 3 BLOCKED	.00000	--	.00000	.00000	.00000
MACHINE 4 BLOCKED	.00000	--	.00000	.00000	.00000
MACHINE 5 BLOCKED	.00000	--	.00000	.00000	.00000
MACHINE 6 BLOCKED	.00000	--	.00000	.00000	.00000
MACHINE 7 BLOCKED	.00000	--	.00000	.00000	.00000

MACHINE 8 BLOCKED	.00000	--	.00000	.00000	.00000
BUFFER 1	1.4480	2.3139	.00000	23.000	.00000
BUFFER 2	.57305	2.6535	.00000	12.000	.00000
BUFFER 3	.62789	2.3623	.00000	10.000	.00000
BUFFER 4	1.0756	2.4509	.00000	15.000	.00000
BUFFER 5	.94699	2.5303	.00000	17.000	5.0000
BUFFER 6	.45187	2.6818	.00000	10.000	.00000
BUFFER 7	.00000	--	.00000	.00000	.00000
BUFFER 8	.00000	--	.00000	.00000	.00000

COUNTERS

Identifier	Count	Limit
COMP1 OF PRD1 ARR	768	9999
COMP2 OF PRD1 ARR	1317	9999
COMP3 OF PRD1 ARR	908	9999
COMP1 OF PRD1 MED	762	9999
COMP2 OF PRD1 MED	1302	9999
COMP3 OF PRD1 MED	885	9999

SIMAN IV - License #8810506
New Jersey Institute of Tech.

Summary for Replication 2 of 3

Project: MACHINE CELL　　Run execution date : 8/17/1993
Analyst: VSP　　Model revision date: 8/12/1993

Replication ended at time : 28800.0

TALLY VARIABLES

Identifier	Average	Variation	Minimum	Maximum	Observations
TIME IN SYSTEM FOR COM	302.70	.54496	12.605	1094.5	672
TIME IN SYSTEM FOR COM	316.22	.53867	25.312	1241.9	1327
TIME IN SYSTEM FOR COM	710.20	.53823	82.904	3031.8	937

DISCRETE-CHANGE VARIABLES

Identifier	Average	Variation	Minimum	Maximum	Final Value
MACHINE 1 QUEUE	1.9512	1.9450	.00000	18.000	5.0000
MACHINE 2 QUEUE	1.0638	2.4546	.00000	19.000	5.0000
MACHINE 3 QUEUE	.33620	3.0398	.00000	8.0000	.00000
MACHINE 4 QUEUE	1.0263	2.5896	.00000	20.000	4.0000
MACHINE 5 QUEUE	2.5716	1.9143	.00000	28.000	.00000
MACHINE 6 QUEUE	.41222	2.9071	.00000	9.0000	.00000
MACHINE 7 QUEUE	.00000	--	.00000	.00000	.00000
MACHINE 8 QUEUE	.00000	--	.00000	.00000	.00000
MACHINE 1 EFFI	.68210	.68269	.00000	1.0000	1.0000
MACHINE 2 EFFI	.58601	.84051	.00000	1.0000	1.0000
MACHINE 3 EFFI	.53122	.93940	.00000	1.0000	.00000
MACHINE 4 EFFI	.60453	.80881	.00000	1.0000	1.0000
MACHINE 5 EFFI	.66340	.71231	.00000	1.0000	.00000
MACHINE 6 EFFI	.53527	.93178	.00000	1.0000	.00000
MACHINE 7 EFFI	.00000	--	.00000	.00000	.00000
MACHINE 8 EFFI	.00000	--	.00000	.00000	.00000
MACHINE 1 FAIL	.00000	--	.00000	.00000	.00000
MACHINE 2 FAIL	.00000	--	.00000	.00000	.00000

MACHINE 3 FAIL	.00000	--	.00000	.00000	.00000
MACHINE 4 FAIL	.00000	--	.00000	.00000	.00000
MACHINE 5 FAIL	.00000	--	.00000	.00000	.00000
MACHINE 6 FAIL	.00000	--	.00000	.00000	.00000
MACHINE 7 FAIL	.00000	--	.00000	.00000	.00000
MACHINE 8 FAIL	.00000	--	.00000	.00000	.00000
MACHINE 1 STARVE	.31790	1.4648	.00000	1.0000	.00000
MACHINE 2 STARVE	.40777	1.2051	.00000	1.0000	.00000
MACHINE 3 STARVE	.46238	1.0783	.00000	1.0000	1.0000
MACHINE 4 STARVE	.39158	1.2465	.00000	1.0000	.00000
MACHINE 5 STARVE	.33550	1.4073	.00000	1.0000	1.0000
MACHINE 6 STARVE	.45606	1.0921	.00000	1.0000	1.0000
MACHINE 7 STARVE	.00000	--	.00000	.00000	.00000
MACHINE 8 STARVE	.00000	--	.00000	.00000	.00000
MACHINE 1 BLOCKED	.00000	--	.00000	.00000	.00000
MACHINE 2 BLOCKED	.00000	--	.00000	.00000	.00000
MACHINE 3 BLOCKED	.00000	--	.00000	.00000	.00000
MACHINE 4 BLOCKED	.00000	--	.00000	.00000	.00000
MACHINE 5 BLOCKED	.00000	--	.00000	.00000	.00000
MACHINE 6 BLOCKED	.00000	--	.00000	.00000	.00000
MACHINE 7 BLOCKED	.00000	--	.00000	.00000	.00000
MACHINE 8 BLOCKED	.00000	--	.00000	.00000	.00000
BUFFER 1	1.9512	1.9450	.00000	18.000	5.0000
BUFFER 2	1.0688	2.4546	.00000	19.000	5.0000
BUFFER 3	.33620	3.0398	.00000	8.0000	.00000
BUFFER 4	1.0263	2.5896	.00000	20.000	4.0000
BUFFER 5	2.5716	1.9143	.00000	28.000	.00000
BUFFER 6	.41222	2.9071	.00000	9.0000	.00000
BUFFER 7	.00000	--	.00000	.00000	.00000
BUFFER 8	.00000	--	.00000	.00000	.00000

COUNTERS

Identifier	Count	Limit
COMP1 OF PRD1 ARR	683	9999
COMP2 OF PRD1 ARR	1346	9999
COMP3 OF PRD1 ARR	962	9999
COMP1 OF PRD1 MED	672	9999
COMP2 OF PRD1 MED	1327	9999
COMP3 OF PRD1 MED	937	9999

SIMAN IV - License #8810506
New Jersey Institute of Tech.

Summary for Replication 3 of 3

Project: MACHINE CELL Run execution date : 8/17/1993
Analyst: VSP Model revision date: 8/12/1993

Replication ended at time : 28800.0

TALLY VARIABLES

Identifier	Average	Variation	Minimum	Maximum	Observations
TIME IN SYSTEM FOR COM	315.01	.53881	34.303	1176.1	720
TIME IN SYSTEM FOR COM	333.58	.52233	32.938	1100.8	1351
TIME IN SYSTEM FOR COM	724.93	.51909	92.469	2597.1	983

DISCRETE-CHANGE VARIABLES

Identifier	Average	Variation	Minimum	Maximum	Final Value
MACHINE 1 QUEUE	2.4838	1.9468	.00000	24.000	.00000
MACHINE 2 QUEUE	1.5164	1.9263	.00000	19.000	.00000
MACHINE 3 QUEUE	.58514	2.3291	.00000	11.000	.00000
MACHINE 4 QUEUE	1.6523	2.6652	.00000	30.000	6.0000
MACHINE 5 QUEUE	2.1980	2.1610	.00000	27.000	.00000
MACHINE 6 QUEUE	.41500	2.7215	.00000	8.0000	.00000
MACHINE 7 QUEUE	.00000	--	.00000	.00000	.00000
MACHINE 8 QUEUE	.00000	--	.00000	.00000	.00000
MACHINE 1 EFFI	.67938	.68697	.00000	1.0000	.00000
MACHINE 2 EFFI	.67692	.69085	.00000	1.0000	1.0000
MACHINE 3 EFFI	.60083	.81509	.00000	1.0000	1.0000
MACHINE 4 EFFI	.66032	.71722	.00000	1.0000	1.0000
MACHINE 5 EFFI	.67423	.69510	.00000	1.0000	1.0000
MACHINE 6 EFFI	.55798	.89004	.00000	1.0000	1.0000
MACHINE 7 EFFI	.00000	--	.00000	.00000	.00000
MACHINE 8 EFFI	.00000	--	.00000	.00000	.00000
MACHINE 1 FAIL	.96168E-03	32.231	.00000	1.0000	.00000
MACHINE 2 FAIL	.00000	--	.00000	.00000	.00000
MACHINE 3 FAIL	.00000	--	.00000	.00000	.00000
MACHINE 4 FAIL	.00000	--	.00000	.00000	.00000
MACHINE 5 FAIL	.00000	--	.00000	.00000	.00000
MACHINE 6 FAIL	.00000	--	.00000	.00000	.00000
MACHINE 7 FAIL	.00000	--	.00000	.00000	.00000
MACHINE 8 FAIL	.13919E-03	84.754	.00000	1.0000	.00000
MACHINE 1 STARVE	.31994	1.4579	.00000	1.0000	1.0000
MACHINE 2 STARVE	.31738	1.4666	.00000	1.0000	.00000
MACHINE 3 STARVE	.39380	1.2407	.00000	1.0000	.00000
MACHINE 4 STARVE	.33578	1.4065	.00000	1.0000	.00000
MACHINE 5 STARVE	.32306	1.4475	.00000	1.0000	.00000
MACHINE 6 STARVE	.43468	1.1404	.00000	1.0000	.00000
MACHINE 7 STARVE	.00000	--	.00000	.00000	.00000
MACHINE 8 STARVE	.48159	1.0375	.00000	1.0000	1.0000
MACHINE 1 BLOCKED	.00000	--	.00000	.00000	.00000
MACHINE 2 BLOCKED	.00000	--	.00000	.00000	.00000
MACHINE 3 BLOCKED	.00000	--	.00000	.00000	.00000
MACHINE 4 BLOCKED	.00000	--	.00000	.00000	.00000
MACHINE 5 BLOCKED	.00000	--	.00000	.00000	.00000
MACHINE 6 BLOCKED	.00000	--	.00000	.00000	.00000
MACHINE 7 BLOCKED	.00000	--	.00000	.00000	.00000
MACHINE 8 BLOCKED	.00000	--	.00000	.00000	.00000
BUFFER 1	2.4838	1.9468	.00000	24.000	.00000
BUFFER 2	1.5164	1.9263	.00000	19.000	.00000
BUFFER 3	.58514	2.3291	.00000	11.000	.00000
BUFFER 4	1.6523	2.6652	.00000	30.000	6.0000
BUFFER 5	2.1980	2.1610	.00000	27.000	.00000
BUFFER 6	.41500	2.7215	.00000	8.0000	.00000
BUFFER 7	.00000	--	.00000	.00000	.00000
BUFFER 8	.00000	--	.00000	.00000	.00000

COUNTERS

Identifier	Count	Limit
COMP1 OF PRD1 ARR	728	9999
COMP2 OF PRD1 ARR	1368	9999
COMP3 OF PRD1 ARR	1010	9999
COMP1 OF PRD1 MED	720	9999
COMP2 OF PRD1 MED	1351	9999
COMP3 OF PRD1 MED	983	9999

Run Time: 3 min(s) 4 sec(s)
Simulation run complete.

BIBLIOGRAPHY

1. Findlay, R. C., FMS concepts realised, in *Proceedings of 3rd International Conference on FMS*, West Germany, 1984.

2. Naylor, T. H., Balintfy, J. L., Burdick, D. S. and Chu, K. *Computer Simulation Techniques*, John Wiley, New York, 1966.

3. Pidd, M., *Computer Simulation in Management Science*, John Wiley, London, 1984.

4. Poole, T. and Szymankiewicz, J., *Using Simulation to Solve Problems*, McGraw-Hill, 1977.

5. Carrie, A., *Simulation of Manufacturing Systems*, John Wiley, 1988.

6. Browne, R., Dubois, D., Rathmill, K., Sethi, S. and Stecke, K., Classification of Flexible manufacturing systems, *THE FMS Magazine*, April 1984.

7. Dupont-Gatelmand, C., A survey of flexible manufacturing systems, *J. of Manufacturing Systems*, 1982.

8. Suri, R. and Hildebrandt, R. R., Modelling flexible manufacturing systems using mean value analysis, *Journal of manufacturing Systems*, 1984.

9. Kellock, B., Why this team said : we'll do it our way, *Computer and Manufacturing Technology*, June 1986.

10. Talavage, J., Simulation analysis of an operational computerised manufacturing system, *Proc. CAM-78, International Conference on Computer - aided Manufacture*, June 1978.

11. Lenz, J.E., *Proc. SIM-1, 1st international conference on Simulation in Manufacturing*, IFS Ltd., England, 1985.

12. Mills, R.I., Simulation for manufacturing systems, *Proc FMS-5, 5th Int. Conf. on Flexible Manufacturing Systems*, England, 1986

13. Shannon, R.E., *Systems Simulation: The Art and The Science*, Prentice-Hall, Englewood Cliffs, New Jersey, 1975.

14. Pritsker, A.A.B., *Introduction to simulation and SLAM II*, Halsted Press and Systems Publishing Corporation, 1984.

15. Pritsker, A.A.B., *Introduction to simulation and SLAM*, John Wiley, New York, 1979.

16. Pegden, C.D., *Introduction to SIMAN*, System Modeling Corporation, 1988.