

A GENERALIZED SIMULATION MODEL FOR THE
DESIGN OF A CONVEYOR SYSTEM

by

Asif Manzoor Shaikh

Thesis submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of
MASTER OF SCIENCE
in
Industrial Engineering and Operations Research

APPROVED:

Dr. R. A. Wysk, Chairman

Dr. R. P. Davis

Dr. P. M. Ghare

May, 1982

Blacksburg, Virginia

ACKNOWLEDGEMENTS

The author wishes to express his gratefulness to the Manufacturing Engineering faculty of the Industrial Engineering and Operations Research Department of Virginia Polytechnic Institute and State University. The author is especially indebted to Dr. Richard Wysk for his supervision and suggestions. His efforts to motivate me to work hard and fast towards completion of this study is greatly appreciated. I also wish to express my thanks to Dr. Robert P. Davis, and Dr. P. M. Ghare for their time and interest in my work.

I would also like to thank Dr. Tanchoco for his guidance during my graduate studies.

Special thanks are also due to Donna Lovern and Laura Moran-Lopez for preparing this report in a short period of time.

CONTENTS

| | | |
|---|--|-------------|
| ACKNOWLEDGEMENTS | | ii |
| <u>Chapter</u> | | |
| | | <u>Page</u> |
| 1. INTRODUCTION | | 1 |
| Advances in material handling systems | | 4 |
| Conveyors | | 8 |
| Advances in conveyor systems | | 11 |
| Planning for a material handling system .. | | 12 |
| 1. Linear programming | | 14 |
| 2. Dynamic programming | | 15 |
| 3. Queueing theory | | 16 |
| 4. Conveyor theory | | 16 |
| 5. Simulation | | 17 |
| 2. PROBLEM STATEMENT AND OBJECTIVES | | 18 |
| Objectives | | 21 |
| Organization of research | | 22 |
| 3. LITERATURE REVIEW | | 24 |
| Deterministic models | | 25 |
| Probabilistic models | | 27 |
| 4. SOLUTION METHODOLOGY | | 33 |
| Classification of conveyors | | 34 |
| Basic conveyor segments | | 36 |
| Simulation language | | 43 |
| Summary of proposed methods | | 44 |
| 5. SIMULATION MODEL | | 45 |
| Modeling procedure | | 45 |
| Modeling Assumptions and constraints | | 48 |
| Examples | | 49 |
| Control variables | | 70 |
| 6. CONCLUSION | | 72 |
| Recommendations for future research | | 74 |
| BIBLIOGRAPHY | | 76 |
| APPENDIX A. PROGRAM LISTING | | 84 |

APPENDIX B. EXAMPLES: INPUT AND OUTPUT 136

APPENDIX C. USER'S MANUAL 173

VITA 209

ABSTRACT

LIST OF TABLES

| <u>Table</u> | | <u>Page</u> |
|--------------|--|-------------|
| 1 | Example 1 output | 55 |
| 2 | Number of servers | 62 |
| 3 | Length of conveyor sections | 63 |
| 4 | Average number in queues | 64 |
| 5 | Parts through system | 65 |
| 6 | Station statistics | 66 |
| 7 | Variable definition | 179 |
| 8 | Service time and arrival rate distributions | 202 |

LIST OF FIGURES

| <u>Figure</u> | | <u>Page</u> |
|---------------|---|-------------|
| 1 | Relationship of material hndling to production | 2 |
| 2 | A conveyor system | 20 |
| 3a | Straight line conveyor segment | 37 |
| 3b | Two-way merge conveyor segment | 38 |
| 3c | Two-way split conveyor segment | 39 |
| 3d | Three-way merge conveyor segment | 40 |
| 3e | Three-way split conveyor segment | 41 |
| 3f | Closed-loop conveyor segment | 42 |
| 4a | Conveyor system: Examples 1, 2 | 50 |
| 4b | Network: Examples 1, 2 | 51 |
| 5a | Conveyor system: Example 3 | 57 |
| 5b | Network: Example 3 | 57 |
| 6a | Conveyor system: Example 4 | 58 |
| 6b | Network: Example 4 | 59 |
| 7 | Subroutine relationship | 188 |

Chapter 1

INTRODUCTION

A typical manufacturing facility contains work station(s), storage, and receiving and shipping linked by a material handling system. The simplest manufacturing facility consists of a receiving dock, service station, and shipping dock (Figure 1). The movement of parts between any two points in a facility is accomplished using various types of material handling equipment. The time a unit spends in a facility is dependent on the effectiveness of the material handling system being used in that facility.

Material handling is concerned with: motion, time, quantity, and space [5]. Moving material in the most efficient manner, controlling the production time, satisfying the demand for the manufacturing processes, and utilizing the available space efficiently are the basis of efficient material handling.

The material handling system to be used becomes critical as a business grows. The design and selection of a material handling system cannot be overlooked if economic manufacture is desired. Material handling directly affects the production in terms of manufacturing costs, product quality and flow, and product traceability. A properly designed material handling system can reduce costs, increase

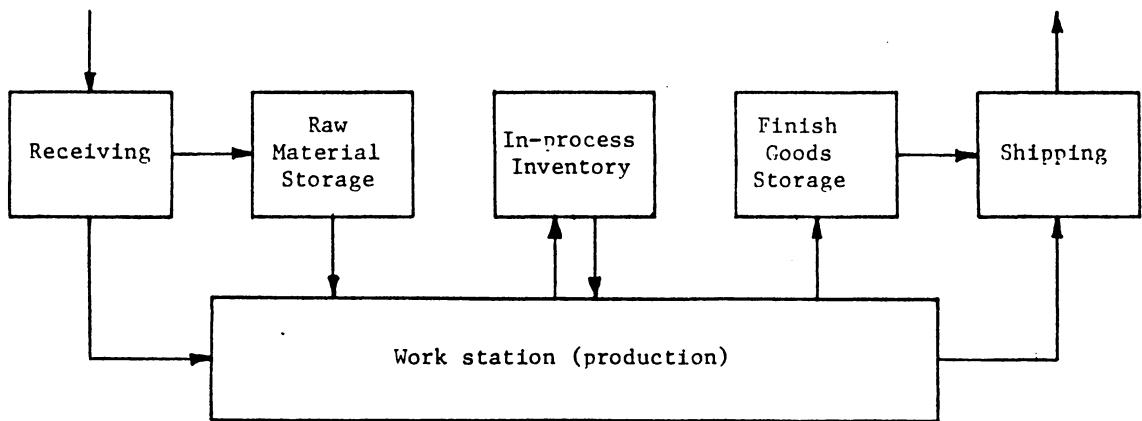


Figure 1. Relationship of Material Handling to Production

capacity, and improve working conditions. Increased capacity may result in better space utilization, improved layout, and/or higher equipment utilization. In the past, material handling costs have been reduced which usually reduced production costs. This trend is declining. In some companies, the material handling cost may be as much as 80-90% of the labor cost [37].

A variety of material handling equipment is used in industry. A combination of several types of this equipment usually make up a material handling system. Apple [5] classifies different handling equipment under three main types as:

1. Conveyors.
2. Cranes and Hoists.
3. Industrial trucks.

There are sub-classifications under each. Material handling systems in operation today are combinations of several types of equipment which are generally categorized as:

1. Fixed-path handling equipment.
2. Limited area handling equipment.
3. Mobile equipment. [52]

Conveyors, storage/retrieval systems, and monorails are a few types of the first class. Examples of limited-area equipment include cranes and hoists while the variable path equipment covers all forms of industrial trucks and perhaps towline systems.

ADVANCES IN MATERIAL HANDLING SYSTEMS-COMPUTER CONTROLS IN MATERIAL HANDLING

During the last two decades, material handling systems have advanced technologically in an extraordinary fashion. These advancements have contributed significantly in warehousing in terms of automation and design. With that trend, attention is now also being given to processing areas (actual assembly areas). It is apparent that some trends have developed. For instance:

1. Automatic and remote controlled equipment,
2. Handling integrated into processing,
3. Handling systems replacing mechanization of individual handling tasks,
4. Communication capabilities integrated into equipment,
5. High speed, large capacity, flexible, and better controlled conveyors,
6. Cranes with remote, and electric or computer control,
7. Unit handling, and
8. Use of robots. [5]

With the trend toward more sophisticated controls, use of computers, and microprocessors that diagnose their own troubles, and progress in other aspects of hardware and

applications, better ways for material handling continue to offer improved operations [22]. The trend is toward more simplified, yet more sophisticated, controls.

Material flow and the optimization of the handling equipment are the primary functions of the controller today. These computers respond very quickly to operation requirements such as the storage and retrieval of parts, sortation, transportation, and tracking of every part in the system.

Computer applications have contributed to the advent of Automatic Storage and Retrieval Systems (ASRS). An ASRS is an automated warehouse with the storage and retrieval of parts done either partially or completely automatic. Storage cells or shelves are used to store parts. Computer controlled cranes, self-guided stock or order selectors are used to accomplish the job of storage and retrieval. The position of each type of part and their quantities are stored in the memory of the computer. Space has always been of major concern, and now narrow aisle concepts and use of guided industrial trucks have shown improvement in space utilization. The hardware consists of two major parts: 1) A mini-computer system with disks and tape data storage, video display terminals and printers, 2) a data terminal that can be mounted on a guided vehicle. A dual processor would offer unique security, reliability and data integrity capabilities of substantial values to any warehousing operation

[62]. By means of advanced telecommunication technology, warehouse personnel can remain in constant communication with the computer system from anywhere in the warehouse.

Carousels are usually used where space is a constraint. Shelves or bins are used to place parts in this "mini-warehouse". These multilevel bins rotate along a fixed path. Each part may have a fixed location of storage. Storage and retrieval of parts is partially or completely automatic. The location and quantities of parts are stored in the memory of the computer. The storage and retrieval process is performed by rotating and bringing the desired bin to a loading/unloading station. Carousels are also being used in manufacturing environments. Work stations are located around the carousel so operators have easy access to the parts.

Several systems are available for part identification and tracking. Magnetic and optical sensors, or decoders, are commonly used for reading identification labels. Printers are used to print the labels. Many systems are also capable of handling inventory and order-entry transactions, which adds a new dimension to inventory control or the status of an order being picked.

In the area of receiving, a software module enables the system to capture all material as it is taken from the receiving dock. A radio data terminal is used to identify

quantity and storage location of the material. This type of receiving handles material that does not need individual tracking [62].

For material flow within the shop, computer controlled conveyor systems are commonly used. Parts are automatically moved from work center to work center throughout the system on a prescribed route. Cartrack systems are also becoming available and very soon will compete with conveyors. Monotonous and dangerous handling tasks are easily performed with this equipment.

Computer controlled, robot-like, variable path industrial vehicles generally identified as Automatic Guided Vehicles (AGV) will be a major part of future material handling systems. Currently, AGV systems are primarily used in warehousing environments [37].

Automated material handling systems have also contributed in the development of Flexible Machining Systems (FMS). An FMS consists of a group of work stations linked by an automated material handling system. Work stations are usually NC machines. This integrated system is computer controlled. An important feature of an FMS is that it can process a variety of different part types simultaneously at the various work stations.

CONVEYORS

There are hundreds of types of material handling equipment and more are being invented every day. It is estimated that presently there are:

240 types of conveyor,
60 types of trucks and vehicles,
100 types of cranes and hoists,
70 types of containers and racks, and
100 types of auxiliary equipment. or

570 types of material handling equipment [5].

In this section some of the common types of conveyors are described. Many definitions have been adopted from Apple [5].

1. Belt Conveyors. An endless fabric, rubber, plastic, leather, or metal belt operating over belt idlers or sliding bed for handling materials, packages, or objects placed directly on the belt. Mesh belts are commonly used in food processing plants. Flat, portable, and troughed are three types of belts. Belt conveyor is used for assembly lines, elevate or lower objects.

2. Bucket Conveyors. Gravity discharged and pivoted are two types of conveyors of this class of conveyors. Buckets attached between two endless chains which operate in suita-

ble guides or casing in horizontal, vertical, inclined, or a combination of these paths over drive, corner, and take-up terminals. Used for heavy and bulk material, and also for wet and greasy material.

3. Cable conveyors. Overhead Tramway is used when space is a constraint. Hooks are attached to wheels which move on a rod.

4. Chain Conveyors. A variety of conveyors fall under this category. Apron, arm, car type, flight, pallet, rolling, trolley are just to name a few, capable of handling almost any material.

5. Gravity Chute. A slide shaped so that it guides objects as they are moved from one location to another. Used for inter-flow and inter level moves.

6. Pneumatic Conveyors. Pipeline, air-activated gravity, and tube are examples of this class of conveyors. Used for dry material, storage of bulky material, and extra safety.

7. Roller Conveyors. Mostly used among all types of conveyors. Supports the load on a series of rollers, turning on fixed bearings, and mounted between side rails at fixed intervals. Distance between rollers is dependent on the size of the object moved. Some example of roller conveyors are:

a. Accordion

b. Gravity

- c. Live
- d. Portable
- e. Spiral

8. Screw Conveyors. Conveyor consists of a continuous or broken-blade helix or screw fastened to a shaft (pipe) and rotated in a trough so that the revolving screw advances the material.

9. Vibrating Conveyors. A trough or a tube flexibly supported and vibrated at a relatively high frequency and small amplitude to convey material (used mostly for hot objects and gaseous materials).

10. Wheel Conveyors. Gravity, live, and spiral are three types of this class of conveyors. These conveyors support the load on a series of skate-like wheels, mounted on common shafts in a frame or on parallel spaced rails, and with the wheels spaced to accomodate the size of the load to be carried (very similar to roller conveyors).

ADVANCES IN CONVEYOR SYSTEMS

Conveyor systems technology has advanced tremendously in the last decade. Photoelectric sensors and programmable controllers of the fiber-optic type are used in conjunction with conveyors for assembly line production. Line shaft conveyors and conveyors with the capability to make 90 degree turns are playing an important role in live roller technology. Different types of mechanisms are available to powerize a conveyor line. Powered accumulating conveyors which can automatically separate products are also available. Accumulating conveyors are used in industry where products must not make any contact with each other. This is the only type of material handling equipment which can be used in a 'super clean' electronic industry where any contact of metals generates enough particals to damage the product. The electronic industry was probably the last industry introduced to automated material handling.

PLANNING FOR A MATERIAL HANDLING SYSTEM

An analytical approach is usually taken in the search for a solution to the material handling problem. It is sometimes called the engineering approach. While systematics may vary, a pattern similar to the following does exist in most approaches:

1. Identify the problem and the objectives.
2. Determine and collect the appropriate data.
3. Develop alternative solutions.
4. Evaluate alternatives and make a decision.
5. Implement the solution and follow-up for improvements.

Many approaches have appeared in various texts and papers; but probably, the best and most widely applied approach is systematic handling analysis (SHA), developed by Muther and Haganas [59]. Basically SHA consists of:

1. A framework of phases which organize the project work.
2. A pattern of procedures to develop material handling plans.
3. A set of conventions.

If these methodologies are applied properly, an effective solution to a material handling problem can be found. Unless properly applied more problems and erroneous results may be obtained.

Some of the more common graphical techniques used for the analysis of material handling problems, and to determine the relationships between operations, criticality of an operation, flow of product etc. are given below:

1. Operation Process Chart
2. Flow Process Chart
3. From-to Chart
4. Flow Diagram
5. Critical Path Method

Details on these and others may be found in texts on plant layout, facilities planning and motion study [7, 50, 53, 83].

Another family of techniques is used for evaluating alternatives and is usually referred to as Quantitative Techniques. These are more mathematical and operations research-oriented procedures. Most of these techniques guarantee a near optimal solution and are divided into these categories:

1. Deterministic
 - A. Linear systems
 - B. Non-linear systems
2. Probabilistic

Some of these techniques will be discussed briefly. It is very important that a proper technique is chosen and used correctly.

1. LINEAR PROGRAMMING

Linear programming is very frequently used to determine the best allocation of resources available to accomplish an objective. The functions expressing the relationships between variables must be linear. Linear programming can be applied to almost all material handling systems. Thompson [85] modeled a scheduling problem as a linear programming problem.

Integer programming, transportation programming, and assignment models are important classes of general linear programs.

A. INTEGER PROGRAMMING. A special case of linear program where variables must be integers. Special cases when only certain specific variables must be integers (called mixed-integer program) can also be handled. The model guarantees an optimum solution. Phillips and Lytton [66] solved a decision problem using integer programming.

B. TRANSPORTATION PROGRAMMING. A class of integer programs is transportation models. Transportation algorithms are used to minimize the cost of distributing a commodity from one point to another point. The supply and demand, at each source and destination are known. This model was developed to support warehouses and find the best schedule.

Restriction of shipment only from a source to destination and not the other way around may not be realistic.

C. ASSIGNMENT MODEL. Problem of assigning n jobs to n machines, usually known as the assignment problem, is another class of integer programs. This is a very useful technique in layout problems. Problems with dependent machines are unsolvable. Problems with unequal jobs or machines can also be handled easily. Aly and Litwhiler [3] developed an allocation model using this technique and applied to the public sector.

D. TRAVELING SALESMAN PROBLEM. A special case of integer program involves finding an optimal route is known the traveling salesman problem. Each location or facility is to be visited only once and route must terminate at the point of origin. No backtracking is allowed. Gensch [27] applied this technique to solve a static time time-constrained scheduling problem.

2. DYNAMIC PROGRAMMING.

Dynamic programming is new among these mathematical techniques and is designed primarily to improve the computational efficiency. The basic idea of the technique is to decompose the problem into smaller subproblems. The dynamic programming approach changes one problem in n variables into n problems, each in one variable. The problem is solved in

stages and a decision is made at each stage. It is applied to network routing problems, production scheduling, etc. It has advantages over conventional procedures in solving multistage decision problems. Rosenman and Gero [78] used a dynamic programming approach to solve design problems.

3. QUEUEING THEORY

Queueing models account for the random flow in a system. Waiting line or queueing theory assumes parts enter the system at some rate, move through different service channels where they are serviced in the fashion in which they are ordered, and then terminate. The arrival and service can be probabilistic. Most of the models make an assumption of Poisson arrival rate and service rate which is not very realistic. Greshwin and Berman [28] analyzed transfer lines using Queueing Theory techniques. Solberg [81] developed "CAN-Q" to analyze flows through material handling systems.

4. CONVEYOR THEORY

Morris [53] classifies conveyors into four categories:

- A. Constant speed, irreversible conveyors (open-loop)
- B. Operator controlled, reversible conveyors (open-loop)
- C. Power and free systems (open loop)
- D. Closed-loop, irreversible, continuous operating system.

Open-loop systems can be modelled mathematically by using previously described techniques. Closed-loop systems are not feasible to be described mathematically. Conveyor theory will be discussed later in the literature review (Chapter 3).

5. SIMULATION

It is always helpful to know how a system is going to work before its implementation. It is normally too expensive and time consuming to experiment with full scale systems. Simulation is the tool most often used in making decisions about the performance of a potential solution. Simulation cannot guarantee optimality, but the best practical solution can be found. Many simulation models have been developed to analyze particular systems [10, 24, 79, 84].

Maxwell [44] criticizes these techniques by, "Simulation has been and will continue to be a prime approach to the problems an industrial engineer, involved in material handling, faces. Optimization techniques based upon linear programming (especially flow networks since they capture inventory relationships) increasingly offer the potential for avoiding a simulation effort. Queueing approaches require further development, and most importantly empirical verification."

Chapter 2

PROBLEM STATEMENT AND OBJECTIVES

Perhaps the most critical step in planning a material handling system is the evaluation of a proposed system design. Evaluation of a facility layout, the routing and scheduling of an AGV system, or the design of a conveyor system are not easy tasks; but it is advantageous to know how a particular design or configuration will operate and perform before it is implemented. Among material handling equipment, conveyor systems are employed universally (perhaps more than any other type of material handling equipment). In evaluating the performance of conveyor systems, design and operational problems are two major areas of interest for an analyst. These two problems are inseparable and should be studied together.

There are a number of evaluation techniques (Chapter 1), and it is very important that a proper methodology is applied to avoid incorrect results and further problems. A conveyor system, if it meets the objectives under the constraints imposed, is said to be operating satisfactorily. An objective can be in terms of cost, profit, space, throughput, etc. Additionally, there may be some constraints, i.e., space, monetary, etc. A mathematical model may seem to be the easiest way to analyze the problem but is gener-

ally difficult to do in the case of a complex conveyor system. The difficulty encountered is that the conveyor is part of a large system. The system includes equipment parameters as well as such considerations as spacing, waiting space allowances, and sequencing of loading and unloading stations etc. It is also extremely difficult to handle the random variables in mathematical models. The layout shown in Figure 2 is very complex. It includes straight line conveyors as well as recirculating or closed loop conveyor sections. To model this layout as a mathematical program and include all details may not be possible. Solving this system to obtain a performance measure may even be a larger problem.

Perhaps the best way to judge the merit of a particular design or configuration is by simulating its performance [9]. Simulation provides feedback on the present concept and can be used to make revisions toward a better design.

Unlike other mathematical programs, it is much easier to include randomness of variables and other detail in a simulation model. Discrete, continuous, or both types of material flows can be handled easily. Recirculation and storage of parts can also be included in the simulation model.

In modeling continuous systems where one is concerned with time-dependent variables, general purpose languages such as BASIC and FORTRAN tend to be used most of the time.

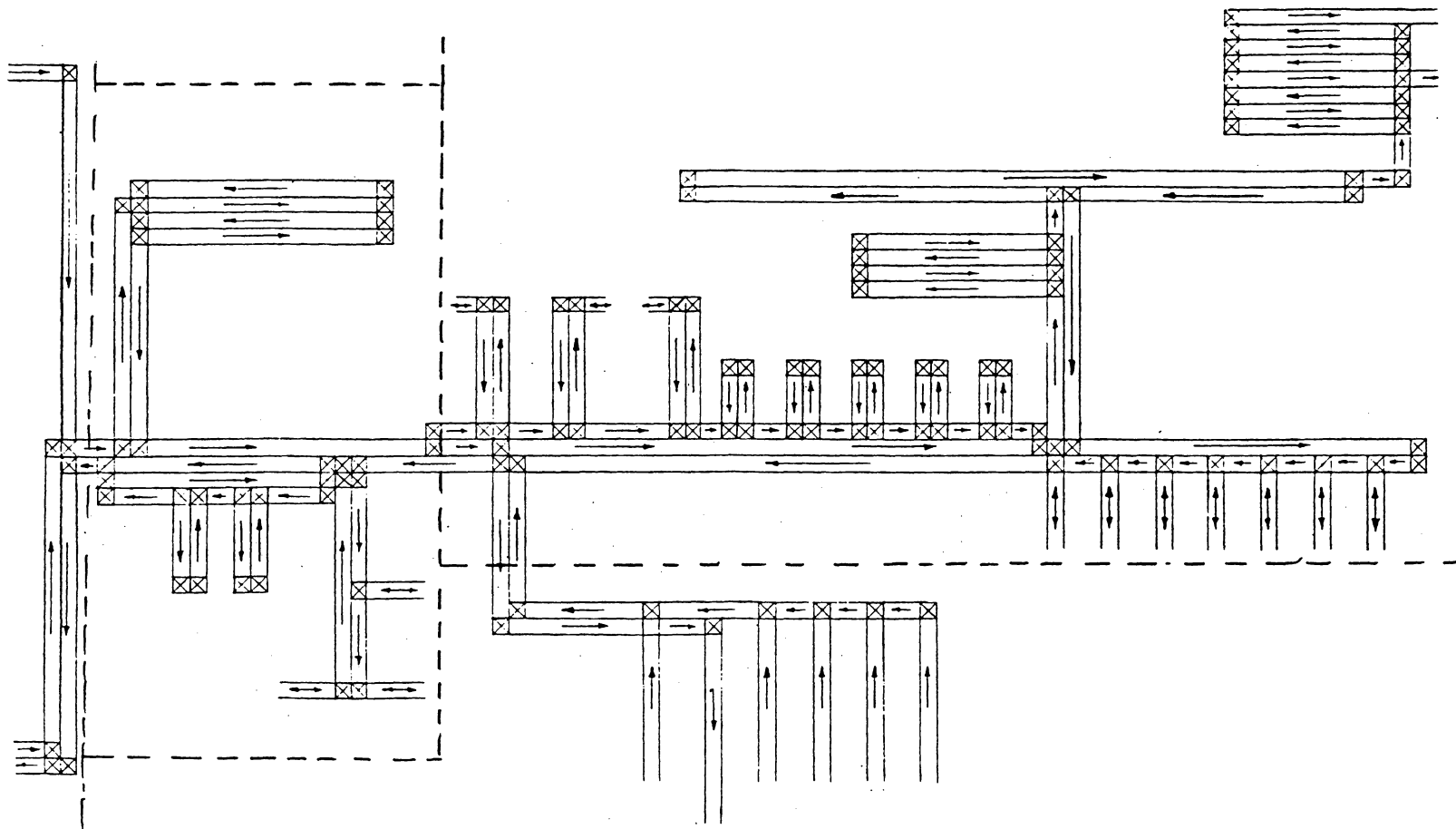


Figure 2. A Conveyor System

An example of a continuous system is a stacker crane, used in an ASRS, where one may account for the speed and velocity of the crane. One often must account for hundreds of entities simultaneously, and monitor changes in their attributes as they progress through the system. A discrete simulation language is best used in such cases. Examples of discrete languages are: GPSS from IBM, SIMSCRIPT from Rand, GASP and SLAM from Pritsker, and ECSL from the University of Birmingham, England [9].

OBJECTIVES

The objective of this research is to develop a simulation package to study and analyze conveyor systems. There is a need for a large-scale, general purpose simulation model that can cope with the complexity of actual conveyor systems. This model should also allow for tradeoff studies in the design stages as well as the analysis of operational problems of existing conveyors [58]. A simulation model will be developed to test the performance of conveyor systems. The objective is two-fold. The first objective is to develop programming modules which model various conveyor types and conveyor segments. The second objective is to compile these modules into a system model to simulate the proposed layout, shown in Figure 2, as well as some variant designs. Different conveyor types, for example: belt and

gravity roller conveyors, have different operational characteristics and cannot be modeled as one type of conveyor. Similarly the model must account for different configurations, like straight line and recirculating conveyors.

ORGANIZATION OF RESEARCH

The layout in Figure 2 is very complex and includes different configurations of conveyor lines. For example: straight line conveyor, recirculating or loop conveyor, intersecting and dividing (splitting) conveyors, etc. Therefore, this layout will be very helpful in testing the simulation model. A part of this layout is used as an example of the application of the simulation model. The simulation study goal is to maximize the flow rates of the product on the assembly line given the conveyor system layout and assembly times that are planned for each work station. Using the example, an analysis to determine any bottlenecks in the conveyor system is performed. Solutions for eliminating the bottlenecks are also discussed. The analysis also includes estimation of average contents and utilization of the conveyor sections, optimization of parts input rate, minimizing delays, estimating the average flow time, etc.

The assembly line process consists of a group of work stations through which a product moves in a predefined sequence. The product is transported via conveyor sections.

If the work station storage area is available, it enters the station. Otherwise, it is recycled on the conveyor sections. When an assembly (subassembly) is complete, the assembled part enters the conveyor system and leaves an opening for another unassembled part to enter that workstation. All parts are on carriers. Empty carriers are returned to the point of origin (entry).

The remainder of this thesis is organized into four major parts. Chapter 3 presents an overview of literature. Some techniques used in analysis of conveyor systems are discussed. This chapter is divided into two areas: 1) Deterministic models, and 2) Probabilistic models.

In Chapter 4, a discussion of how the problem under consideration may be solved is presented. How the problem is formulated and modeled is discussed in Chapter 5. Variables and simulation model are also discussed. Input and output for specific examples is given. A small portion of the layout shown in Figure 2 is simulated to show how the model can be used for planning and design purposes.

Chapter 6 contains a brief summary of the research and a discussion on how the model can be applied to other material handling systems in addition to conveyor systems. Future developments and model expansions are also discussed.

Chapter 3

LITERATURE REVIEW

The design and implementation of a conveyor system is a very complex process. Evaluating proposed designs is very critical. Many papers have been published and texts have been written on the analysis of conveyor systems. In Chapter 1, some techniques for analysis were mentioned. This chapter gives an overview of the literature and discusses some research contributions using different analysis techniques. The chapter is divided into two major areas.

1. Deterministic models
2. Probabilistic models

Under these two topics, models with single and multiple loading and unloading stations (service channels), discrete and continuous flow of material, and storage areas (banks) are discussed.

Since the late 1950's, the birth of conveyor theory, a lot of work has been done in developing mathematical, network, and simulation models for the analysis of conveyor systems. Kwo's work [41] on conveyor theory is probably the earliest work published. He realized the need for analytical approaches in the study of conveyors. After his appeal for analytical approaches, a number of papers were published in the early 1960's [35, 41, 42, 46, 75, 77]. Most of these

early publications were by authors who were also employed as engineers in industry. Thus, conveyor theory developed from a concern within industry to develop analytical models of real world problems [58]. The majority of the work done has concentrated on constant flow-through conveyor systems.

1. Deterministic Models

One of the very first published works on the analysis of conveyor systems concentrated on compatibility of the design of the conveyor with the input and output rates of parts on the conveyor. Kwo's [41,42] model analyzes a conveyor system with one loading station and one unloading station, and time-varying patterns of material flow through the conveyor system. The solution is feasible under some very restrictive input-output patterns. However, it is not a general solution procedure. Kwo's work is considered a milestone and is further studied by others.

Kwo's model led to a mathematical analysis of the problem by Muth [54]. He divided the problem into two separate problems as: 1) continuous loading, and 2) discretely spaced loading. He also extended his results for single loading and unloading stations to the case of multiple loading and unloading stations [55, 56]. A difference equation, to describe material flow along the conveyor, whose solution yields the conditions under which the conveyor is operable for general periodic input and output patterns, is used.

Moodie, Sadowski, and Hill, Jr. [49] developed an integer programming model which can be used to determine optimal (or near optimal) design configurations for unloading of a high speed, mixed product production line. The procedure is a practical application of integer programming. The methodology is applied to a conveyor system with one loading station and multiple unloading stations. Simulation experiments proved that the results were, in fact, good designs.

Morgan [51] analyzed the steady-state behavior of two - link conveyor systems. He considered the systems with intermediate storage, the mean flow, and the number of carriers in queues. For the single carrier in the first link systems, a set of linear equations are used to determine the desired values. An approximation method is used for systems with large number of carriers.

Mitsumori [48] considered a conveyor line with n work stations. He modeled the system as a mathematical programming problem. Optimization of the conveyor-in sequence is shown as maximizing the minimum operation time for all work stations and semi-finished parts.

Ratliff [73] considered a class of production scheduling problems. He discussed how these can be modeled as network flow problems. He assumed that the parts are produced in batches. He also restricted the cost functions to be separable and convex.

Maxwell and Wilson [45] introduced a new methodology for the analysis of material handling systems. They developed a network flow model to analyze the flow dynamics of fixed path material handling systems. The problem is formulated as a cost minimization problem and can also be formulated as a flow maximization problem. Continuous, accumulation, discrete carrier chain, and power and free conveyor systems can be analyzed by this technique. Even though this problem has some special characteristics which do not permit the use of network flow algorithms, it has opened a new area of research in terms of network flow analysis of material handling systems.

2. Probabilistic Models

Perhaps the first probabilistic model was developed by Mayer [46]. His model includes n service channels, closed loop conveyor, and discrete flow of multiple items. All carriers are discretely spaced and are empty when they arrive at the first station. The conveyor is used to transport the units produced from the workstations to the next stage. Units are placed in carriers as they are finished. A nearest available carrier is selected, otherwise, the unit is placed on the floor. He conducted the analysis with carrier capacity of one, and two units. White [86] considered the general case of a carrier with capacity of x units. He

defines the design parameters and indicates how the optimum values of these parameters can be determined. In addition to Mayer's model, Morris [53] includes multiple loading and unloading stations. A conservation of flow approach was employed to develop the performance measures for the system. His model was validated by White and Woodbury [87] using simulation.

Some researchers have concentrated on individual work stations [75, 76, 77]. They have developed probabilistic models of a loading and/or unloading work station. No delays are allowed. Temporary storage areas are assumed to avoid delays in loading/unloading operations. The research focused on the effect of various storage and retrieval disciplines on production. Beightler and Crisp, and Reis, Brennan, and Crisp [8, 74] modeled a single work station as a Markov process to analyze the effect of various storage and retrieval disciplines on the in-process storage requirements. [8] assumed stationary Bernoulli arrivals but Crisp, Skieth, and Barnes [19] later proved using simulation that this assumption could not be validated.

Queueing theory has also been used by many in analyzing conveyor systems. Disney [23] formulated the two unloading stations conveyor system as a multichannel queueing problem. Pritsker [69] generalized Disney's work to m unloading stations. Disney assumed $M/M/m$ queue, but Pritsker considered

M/G/m and D/M/m queues. Pritsker also used simulation to allow recirculation with storage allowed only at the last station. Gregory and Litton [31] formulated the case of m dissimilar work stations and ordered entry with random arrivals as a queueing model and showed that in order to minimize the lost units the work stations should be ordered by descending service rate. Recirculation is not allowed. All the work quoted in this paragraph addresses queueing systems and makes an assumption of non-recirculation, which is very unrealistic [58].

Proctor, Elsayed, and Ragab [71] investigated the steady-state behaviour of a two-channel ordered entry conveyor system. They analyzed the conveyor system both mathematically, using the principles of the queueing theory, and by simulating it. Storage is allowed only at the second service center. Elsayed and Proctor [25] also investigated the steady-state behaviour of two and three channel conveyor systems with n types of Poisson distributed arrivals, and two different queueing disciplines.

Agee and Cullinane [2] developed an economic model to determine the optimum number of loading and unloading stations and conveyor length. The study is based on a single loading and a single unloading point with multiple loading (or unloading) stations allowed at the loading (or unloading) point. A nonstationary Poisson process is assumed and

blocking could occur only when the conveyor is full. A transient analysis is conducted using numerical methods.

Muth [57] analyzed a closed loop conveyor system having a single loading station, a single unloading station, and discrete, time-varying input and output flows. It is shown that the output flow is varied less than the input flow with a suitable decision rule for unloading. Recirculation is also allowed.

Perhaps the only probabilistic analysis which considered not only Poisson but non-Poisson arrivals as well is carried out by Matsui and Shingu [43]. They analyzed and developed an unloading policy in a conveyor system with Poisson and non-Poisson arrivals. The policy minimizes delay time per unit produced. The results also would be useful in designing other queueing systems.

Buzacott [12] analyzed an automatic transfer line with in-process storage consisting of two or three work stations. He performed a Markov chain analysis and studied the effect of buffer capacity on the production. He also studied the line without inventory banks [13] and also with the problem of breakdowns [14].

Phillips and Skeith [67] analyzed a conveyor system using simulation techniques. They included m service channels, and recirculation and storage at each channel. This model was also used to validate the results of Pritsker.

Gourley and Terrell [29, 30] developed a modular general purpose simulation model to study constant-speed, discretely loaded, and recirculating conveyors. The model is extended by Chen and Terrell [17] to include multiple loop conveyor systems which service multiple floors.

GERT and queueing theory is applied together in an analysis of a conveyor system by Ohta [63]. Service time distributions are bounded in the discrete conveyor model. No storage is allowed at the work stations and there is no recirculation. The model is described by states with Markovian property. The model provides important information in the system design.

A queueing network analysis program by Solberg [81], CAN-Q can be used to analyze the network flow in a conveyor system. This evaluation program models the system as a network. Nodes may represent work stations and arcs may represent flow of parts.

Considerable amount of work has been done in developing simulation packages for the analysis of production lines. Possibly the two most significant efforts are by Illinois Institute of Technology Research Institute [1], and by Phillips, et al [68]. [68] developed Generalized Manufacturing Simulator (GEMS). Even though it is not specifically designed for production line analysis, one can represent the system as a network and apply GEMS. GEMS also requires user

to learn modeling in order to use it. On the other hand the Generalized Assembly Line Simulator (GALS) by [1] is specifically developed for the production line analysis and it does not require the user to understand any computer language or network modeling technique. The user only inputs necessary parameters in order to execute. GALS, however, has a drawback that it does not handle material handling component of production line.

Chapter 4

SOLUTION METHODOLOGY

There are about 240 types of conveyors available today. Developing programming modules for each and every conveyor system may not be an easy task and will certainly not be an efficient way to model these systems. Also, organizing and keeping track of these many modules in a model may create problems. A solution to the problem is to categorize the conveyor types to reduce the programming effort as well as the compiling effort of the modules into a system model.

A general-purpose model is applicable to many different conveyor system designs. There are a large number of systems which differ in design, making it impossible to develop simulation modules for each separately. A system is made-up of many conveyor segments. If systems are decomposed into smaller segments (modules), there will probably be very few segments with significantly different characteristics. Developing modules for a few segments and then putting them together to make-up a system is much easier than developing separate modules for individual systems.

To summarize the above, as many conveyor types as possible are classified into as few of classes as possible. This simplifies the problem of developing a programming module for each conveyor type. Conveyor systems are broken

down into smaller conveyor segments, and simulation modules for these segments are developed. This reduces the problem to developing modules for a few conveyor segments rather than the entire systems.

CLASSIFICATION OF CONVEYORS

Conveyors can be classified according to many different criteria. In this study, conveyors are classified according to their operating characteristics. Conveyor types fall into the following categories:

1. Continuous Flow (e.g. Gravity chute)
2. Discretely-spaced (e.g. Belt conveyor)
3. Discretely-spaced Fixed Cycle (e.g. Screw conveyor, drag-line)

Some of the characteristics of these three classes are described.

1. Continuous flow conveyors, like roller conveyors, allow parts to move from one end (loading end) to the other end (unloading end) of conveyor if there is space available. Parts wait at the unloading end of the conveyor to be picked up. As soon as a part is picked up, the next part in the line is available to be unloaded. Parts can queue up until there is no more space available on the conveyor.

2. Discretely-spaced conveyors, like belt conveyors, operate in the same manner as the continuous flow conveyors except that there is some distance between parts. When a part reaches the unloading end of the conveyor and is not unloaded, it stops the conveyor and no other part can advance until the first part is picked up from the unloaded end.

3. Discretely-spaced fixed cycle conveyors, like bucket conveyors, operate like the previous class of conveyors. The only difference is that parts can only be loaded at fixed points on the conveyor.

The types of conveyors described in Chapter 1 may be included in these classes as follows:

| <u>Class</u> | <u>Conveyor Type</u> |
|--------------|----------------------|
| 1 | Gravity Chute |
| | Pneumatic |
| | Roller |
| | Vibrating |
| | Wheel |
| 2 | Belt |
| | Chain |
| 3 | Bucket |
| | Cable |
| | Screw |

BASIC CONVEYOR SEGMENTS

Although many conveyor systems differ in overall design, they all have common basic segments. In other words, most systems have the same basic segments. For example, the layout shown in Figure 2 has three main areas. Each is different from the other in overall design, but all of them have the same basic segments such as straight-line conveyor, intersecting conveyor, etc. In this section the most basic segments are identified.

Perhaps the most common and the most basic segment is a straight line conveyor. Two-way merge and split, three-way merge and split, and closed-loop are other segments. These are shown in Figures 3a through 3f. By arranging these segments properly, one can design an entire system.

Given the three classes and these segments, programming modules are developed for the feasible combinations. For example: class 1 conveyors can be used in designing any segment, but class 3 conveyors cannot be used to design merging, intersecting, or closed loop conveyors. The feasible combinations are as follows:

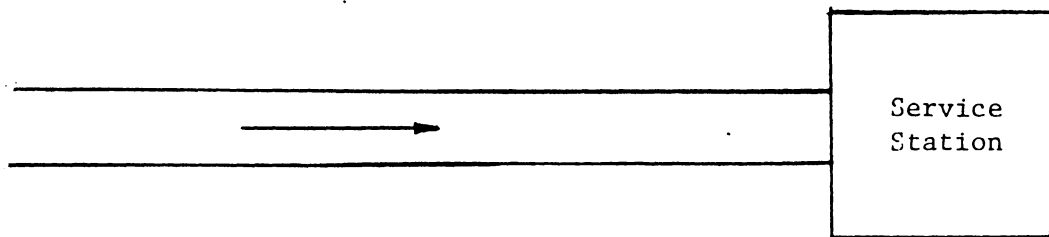


Figure 3a. Straight-line conveyor segment.

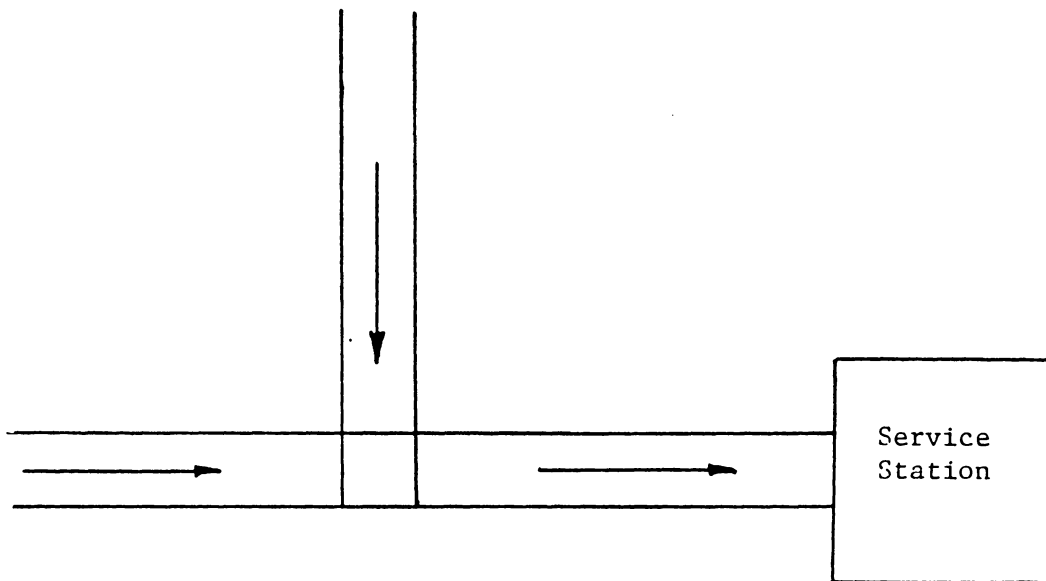


Figure 3b. Two-way Merge Conveyor Segment.

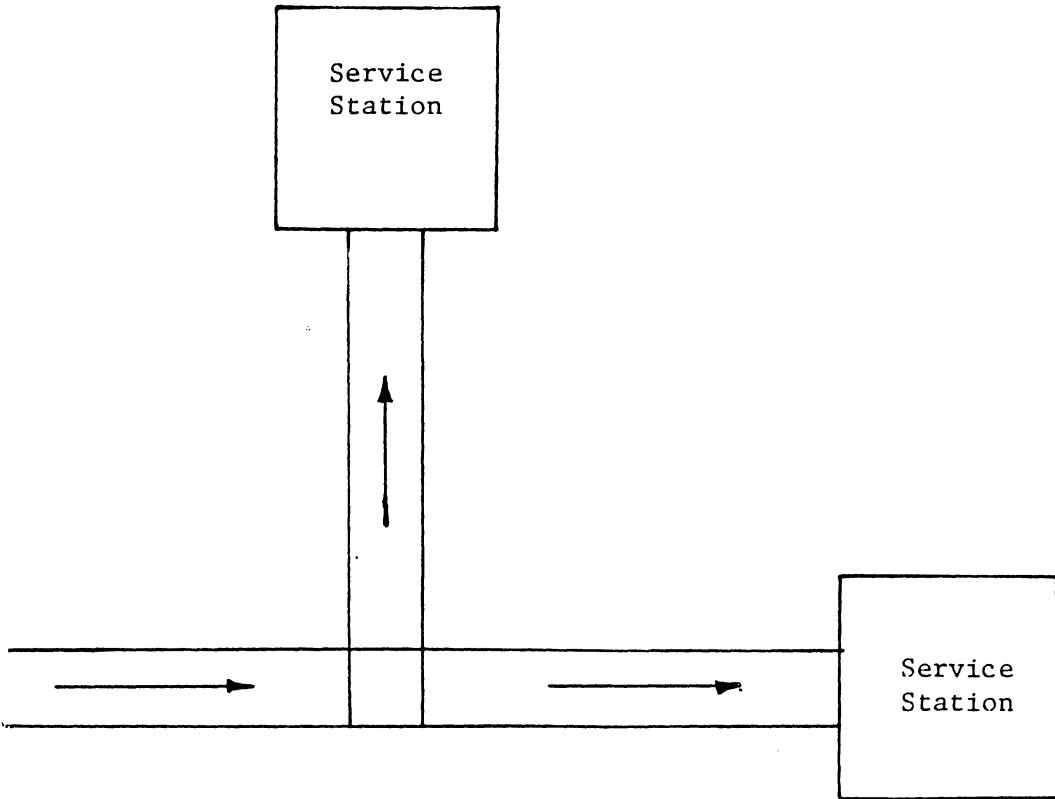


Figure 3c. Two-way split conveyor segment.

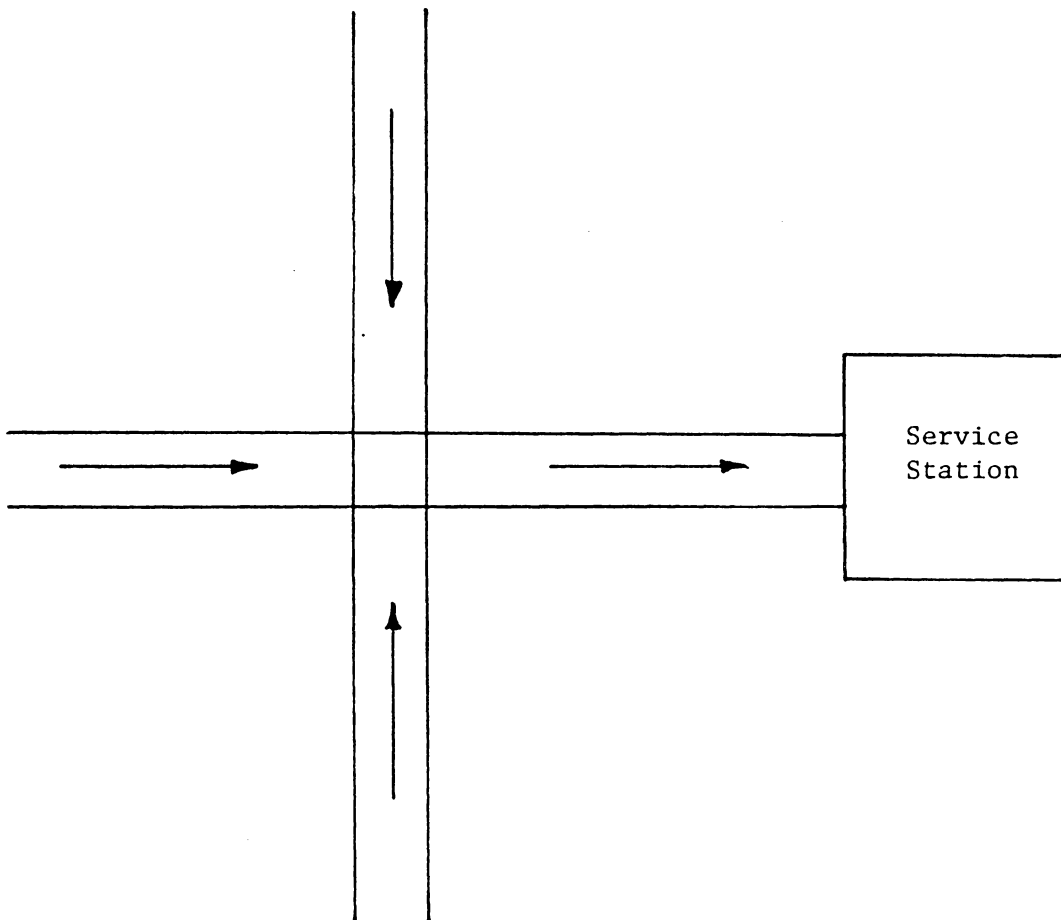


Figure 3d. Three-way merge conveyor segment

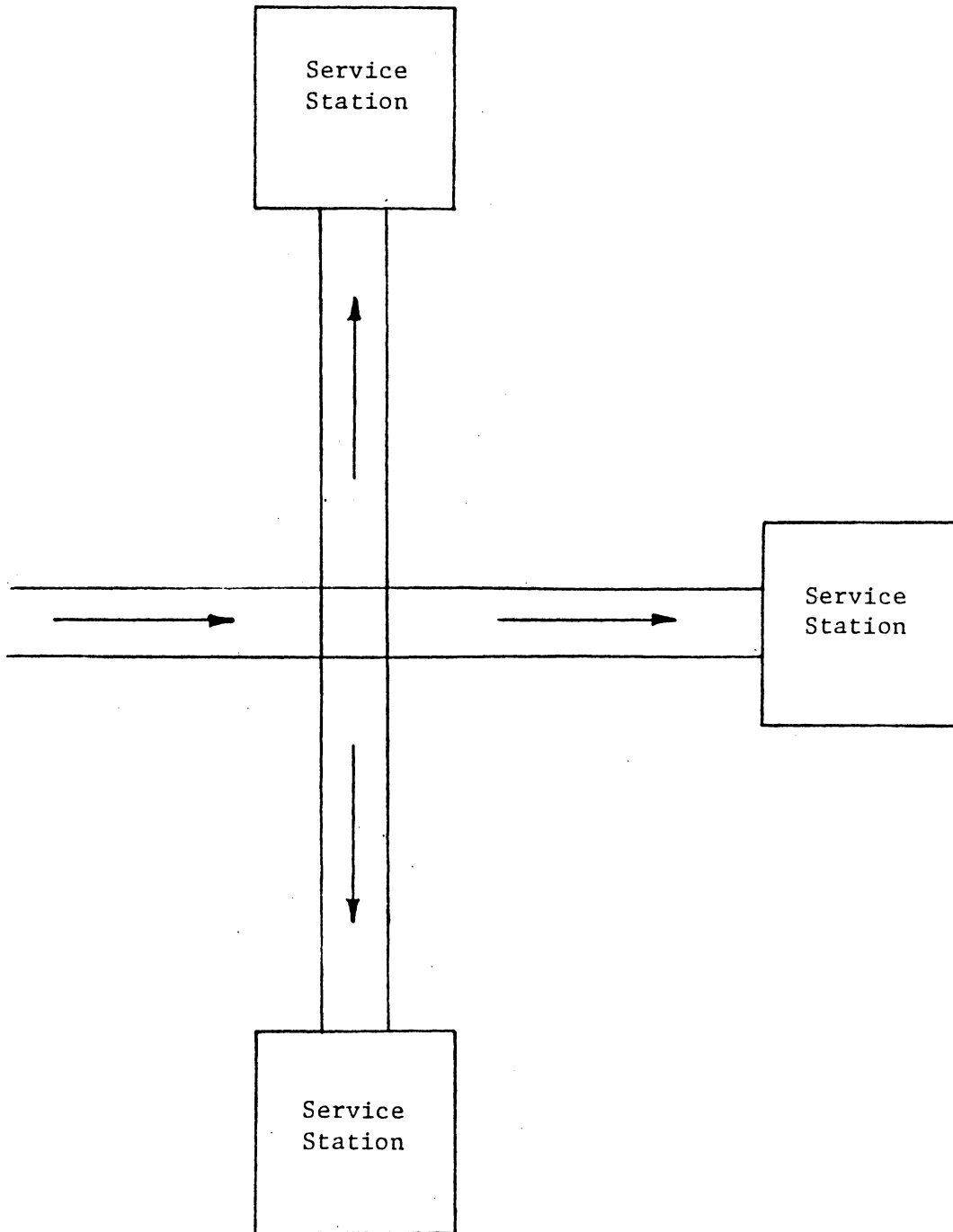


Figure 3e. Three-way split conveyor segment.

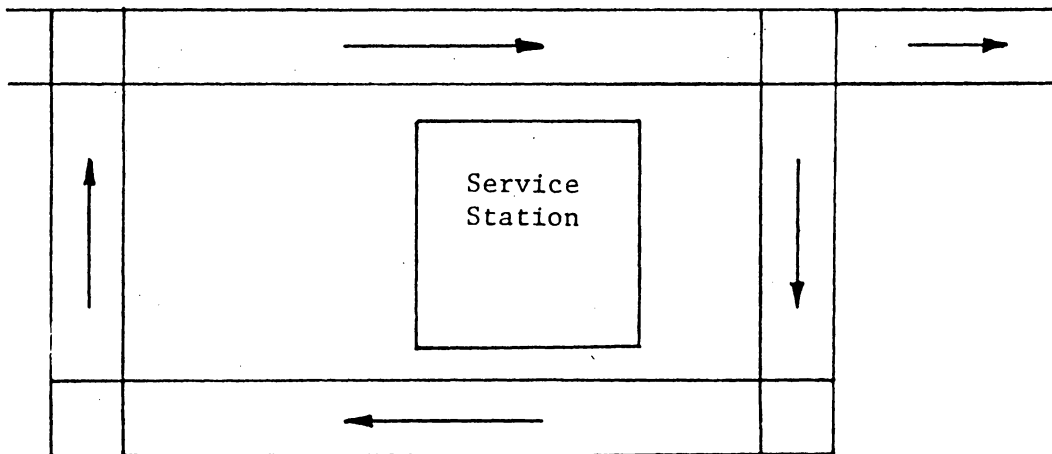


Figure 3f. Closed-loop conveyor segment.

| <u>Conveyor Segments</u> | <u>Conveyor Classes</u> |
|--------------------------|-------------------------|
| Straight line | 1, 2, 3 |
| Two-way merge | 1, 2 |
| Two-way split | 1, 2 |
| Three-way merge | 1, 2 |
| Three-way split | 1, 2 |
| Closed loop | 1, 2 |

Also class 1 and class 2 type conveyors allow recirculation while class 3 type conveyors do not permit parts to be recirculated.

SIMULATION LANGUAGE

As mentioned earlier, there are a number of languages available which can be used to simulate different systems. SLAM II is perhaps the most advanced simulation language and also the most recent. SLAM II is the only available simulation language that supports three different world views of modeling in a single, integrated framework. It permits the use of discrete event, continuous, and network modeling perspectives. It also permits any combination of the three viewpoints to be used in developing simulation models. SLAM II is probably the fastest growing simulation language and is already used by many industries and institutions due to the unique combination of ease and flexibility it provides.

SUMMARY OF PROPOSED METHODS

The objective of this study is to develop a simulation model to study and analyze conveyor systems. Conveyors are categorized into three classes. Six segments which differ in design are the foundations of most of the system designs. Feasible combinations of conveyor classes and segments are modeled within a SLAM II framework. These programming modules are compiled into a system model. This model will be tested by applying it to variant designs.

Chapter 5

SIMULATION MODEL

This problem can be formulated by using the discrete-event or network portions of SLAM. The network portion of SLAM is conceptually easier to use but is not efficient (computationally) when dealing with larger systems. The discrete-event portion of SLAM is not as easy to use but adds flexibility in modeling larger systems. Conveyor systems can also be formulated by using both portions together. A decision on which modeling technique to use was made after testing the ease and flexibility of modeling. It was decided to formulate the problem using discrete-event SLAM.

MODELING PROCEDURE:

The simulation model is structurally divided into three sections, each section corresponding to a different class of conveyors. This simplified the programming efforts and also made it easier to debug the model. Each section is basically the same except for a few minor details to distinguish among the three classes of conveyors.

Conveyor systems are treated as networks. Nodes represent a merging or splitting point (intersection) and also the service areas (stations, machines). Conveyor sections are represented as arcs. When a part, or entity, enters the system, it is assigned some attributes which are modified as

it moves through the system. An entity is assigned its proper route in terms of the service areas it needs to visit. The model transforms this in terms of nodes and arcs in the network.

Dijkstra's shortest path algorithm is used to determine the path between two stations. This path is in terms of nodes and arcs in the network, and is assigned to attributes of an entity.

Basically the model moves parts from conveyor to conveyor, and from station to station. There is no major difference in the concepts of modeling the three classes of conveyors. In all three cases, 'look ahead' methodology is employed. If there is no space on the conveyor, or the station, where a part is supposed to go next, it is stopped in the system. The only difference among the three sub-systems is the detail required before moving a part. Consider the following cases when a part needs to move from a conveyor, or a station to another conveyor or station. The modeling differences in modeling the three classes of conveyors follows.

1. A part has just arrived in the system and needs to go to conveyor 1. If there is no space on conveyor 1, it will not be put onto the conveyor. It will wait in a file (queue). For class 2 conveyors, an additional check is made to see if conveyor 1 is running or not. If not, the part

will wait in a queue. For class 3 conveyors, in addition to the above two conditions, another condition must be met before the part can be placed on the conveyor. It is required that the part must wait for a loading point on the conveyor before a test on the status of conveyor (availability of space and working condition) is made.

2. Now consider the case where a part needs to go from one conveyor 1 to another conveyor 2, which precedes 1. If the part is not able to go to 2, it stays on 1 and also blocks the parts following it on 1. For classes 2 and 3, when this happens, conveyor 1 also stops moving and it delays the movement of parts on it.

3. Further consider the case where a part has just finished service at station 1 and needs to go to conveyor 2, which precedes conveyor 1. When the part cannot leave the station, because of any of conditions given in case 1, it will stay at the station and will block the server from serving any more parts. When there is a space on conveyor 2 and if it is functioning, the part which is blocking a server will have priority over any other parts on conveyor 1 to be loaded onto conveyor 2.

4. Consider the case in which a part has reached a service station and needs to be served there. If the server is busy and recirculation is allowed, it will continue moving if possible, otherwise it will be stopped. If recircu-

lation is not allowed, it will stop there and wait until served. It will stop the parts following it.

Modeling Assumptions and Constraints.

1. Conveyor speed is constant.
2. Different types of parts are allowed in the system.
A part is different from another part if they follow different routes. All parts, same type or different types, have the same dimensions (unit load). It is easier to check how much space is available on a conveyor by just looking at the number of parts on the conveyor.
3. All queues are arranged according to first come first serve priority. Only the priority on service queues can be changed using SLAM input statements (discussed later).
4. There is no pre-emption allowed in the service.
5. At present all yields are 100%. There are no bad parts produced.
6. There may be identical and independent servers at a station. If two servers at a station have different service times, they should be treated as two different stations.
7. A service area, or station, has only one queue, and all servers share that common queue.

8. At present, there is only one location for a service area.
9. At present, there is only one entrance for a part type.
10. A node can have a maximum of 4 arcs emanating and departing. In other words, a node can be linked with a maximum of 4 other nodes.
11. Service times and interarrival times can be any of the following type:
 - a. Constant,
 - b. Random,
 - c. Exponentially distributed,
 - d. Uniformly distributed,
 - e. Triangularly distributed, and
 - f. Normally distributed.

EXAMPLES:

Small conveyor systems are used to discuss input and output for the three classes of conveyors and how one can use the model. Another system is simulated to discuss how the model can help in designing or improving the design of a system.

1). A conveyor system, shown in Figure 4a, has six different stations which serve three different types of parts. Figure 4b represents the network form of the system.

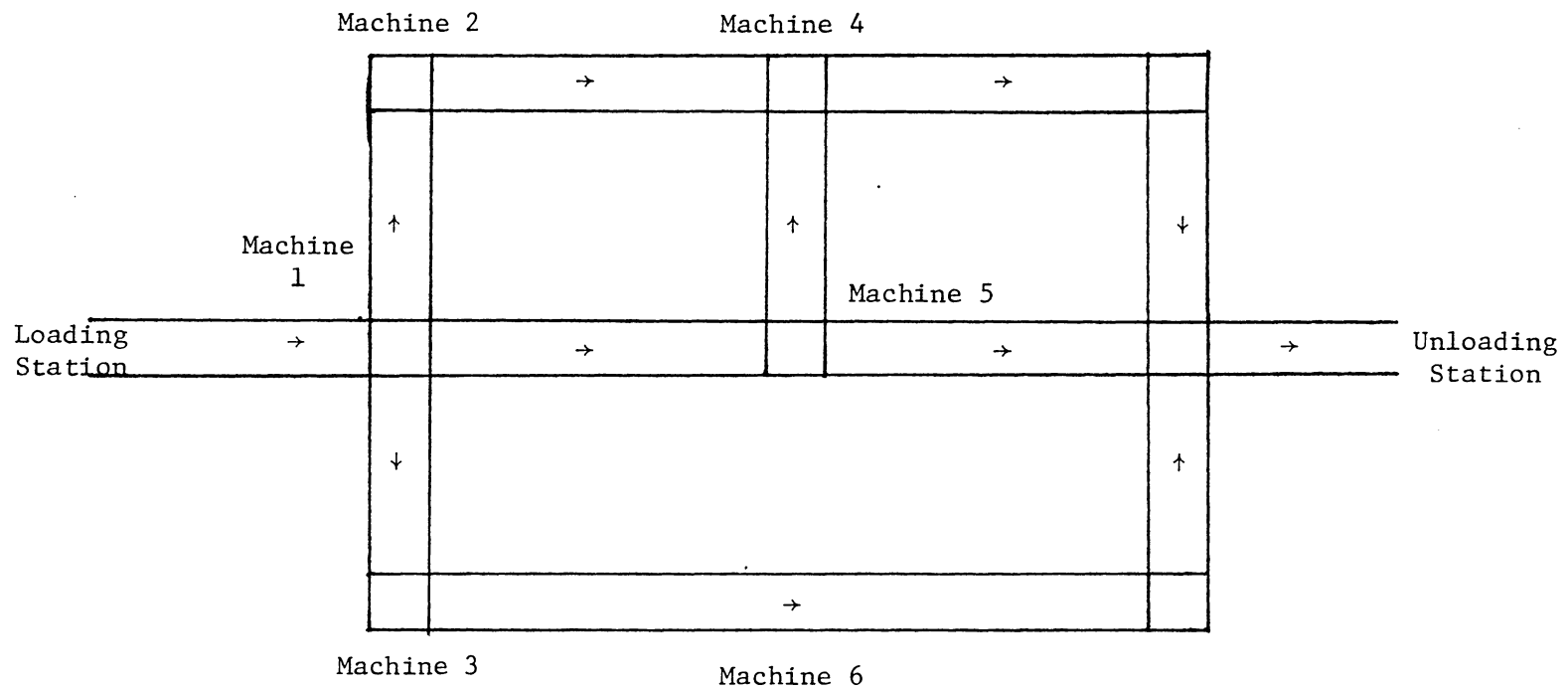


Figure 4a. Conveyor system: Examples 1, 2

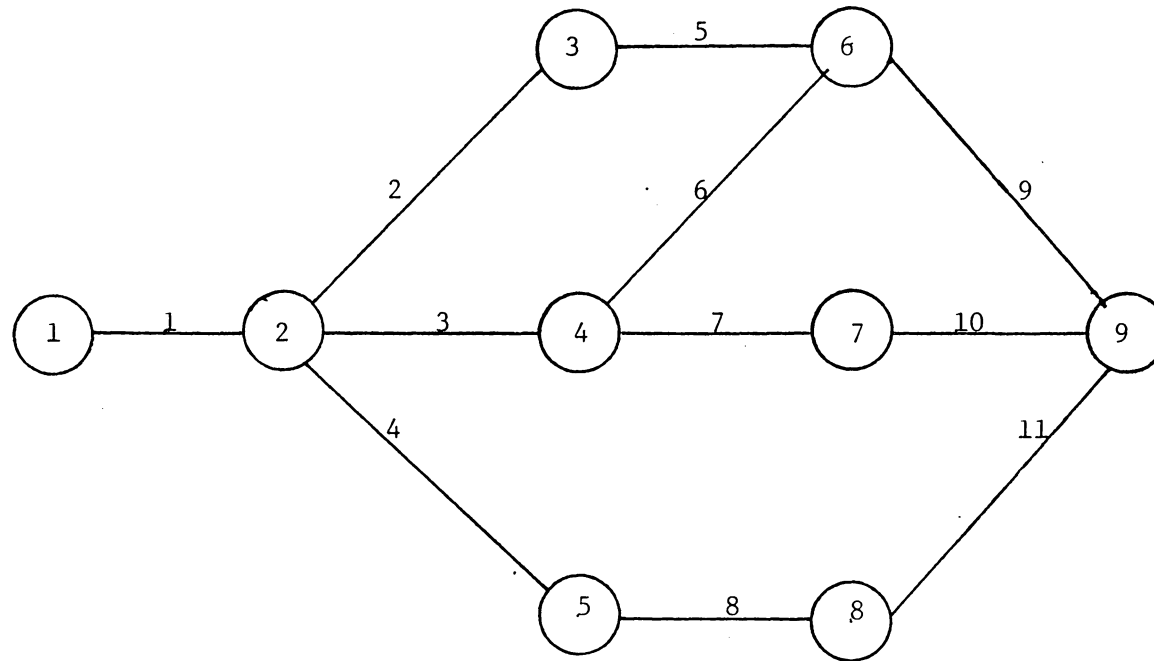


Figure 4b. Network: Examples 1, 2

Arc numbers are placed over each arc. How the conveyor system is converted into a network is described here. Also, information about the system is given below.

Each conveyor section is represented as an arc in the system. Each station is represented as a node. If a station is in the middle of a conveyor section, a node, representing the station, is dividing the two arcs which represent the conveyor. In this case, one conveyor is split into two conveyors. Any junction of two or more conveyors is also represented as a node. The loading and unloading stations are also nodes even if the loading/unloading time is zero.

In Figure 4b, nodes and their description are as follows:

| | |
|--------|-------------------|
| Node 1 | Loading station |
| Node 2 | Machine center 1 |
| Node 3 | Machine center 2 |
| Node 4 | Splitting point |
| Node 5 | Machine center 3 |
| Node 6 | Machine center 4 |
| Node 7 | Machine center 5 |
| Node 8 | Machine center 6 |
| Node 9 | Unloading station |

Only node 4 is not considered a station.

Input:

Appendix B contains input and output for the examples. First, consider the input to example 1.

1. SLAM Input. This will not be discussed here. The user can consult the SLAM manual or text [70]. The variables which are dependent upon the conveyor system are discussed in Appendix C.

2. MODEL Input. Input to the model is unformatted and described in detail in Appendix C. Limitations on some of the variables are also mentioned.

Output:

There are three parts of the output. First is the SLAM echo report. The second is the intermediate results. SLAM summary report is the third part of the output.

SLAM echo report is basically to check the input data and SLAM variables. This is explained in detail in SLAM text [70]. Intermediate results will always give statistics for how much time a unit spends on a conveyor and how many units are on a conveyor. Average, standard deviation, etc. is printed out. Conveyor utilization can be calculated by the formula.

$$\text{Conveyor Utilization} = \frac{\text{Ave. number on Conveyor}}{\text{Capacity of conveyor}} \times 100\%$$

As can be seen from Table 1, conveyor 1 was utilized 71.77% in example 1.

$$\text{Conveyor Utilization} = \frac{2.153}{3.000} \times 100\%$$

Table 1 summarizes the output of example 1. The SLAM summary report provides statistics on how much time each part of a particular type spends in the system. It also gives statistics on how many servers at a station are busy. The user can determine other statistics by simple calculations.

An important part of the SLAM summary report is file statistics. This indicates how many parts are waiting at a station and how long do they wait. Consider example 1; User may neglect statistics on files 1, 2, and 3. File 1 is used when a part is waiting to move from a station to a conveyor, or from another conveyor if the node between the two conveyors is not a station. Files 2 and 3 are not used for class 1 conveyors. File 4 through the maximum number of user defined files are used when a part is waiting for service at a station or while waiting to go to another conveyor which is not available. The last file, file 12, is of no interest to the user, it is the event calender.

All this information can be helpful in making design decisions and evaluating alternatives. This will be discussed later.

Table 1. Example 1 Output

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

SIMULATION PROJECT EXAMPLE

BY ASIF SHAIKH

DATE 1/13/1984

RUN NUMBER 1 OF 1

CURRENT TIME 0.8502E+03

STATISTICAL ARRAYS CLEARED AT TIME 0.1000E+03

STATISTICS FOR VARIABLES BASED ON OBSERVATION

| | MEAN VALUE | STANDARD DEVIATION | COEFF. OF VARIATION | MINIMUM VALUE | MAXIMUM VALUE | NUMBER OF OBSERVATIONS |
|----------------|---------------|-----------------------|------------------------|------------------|------------------|---------------------------|
| JOB1 IN SYSTEM | 0.1521E+03 | 0.9311E+02 | 0.6121E+00 | 0.2590E+02 | 0.2679E+03 | 87 |
| JOB2 IN SYSTEM | 0.9151E+02 | 0.9240E+02 | 0.1010E+01 | 0.1980E+02 | 0.2954E+03 | 67 |
| JOB3 IN SYSTEM | 0.1659E+03 | 0.1006E+03 | 0.6060E+00 | 0.4170E+02 | 0.3042E+03 | 72 |

STATISTICS FOR TIME-PERSISTENT VARIABLES

| | MEAN VALUE | STANDARD DEVIATION | MINIMUM VALUE | MAXIMUM VALUE | TIME INTERVAL | CURRENT VALUE |
|----------------|---------------|-----------------------|------------------|------------------|------------------|------------------|
| STATION 1 BUSY | 0.0 | 0.0 | 0.0 | 0.0 | 0.7502E+03 | 0.0 |
| STATION 2 BUSY | 0.5412E+00 | 0.4983E+00 | 0.0 | 0.1000E+01 | 0.7502E+03 | 0.0 |
| STATION 3 BUSY | 0.0 | 0.0 | 0.0 | 0.0 | 0.7502E+03 | 0.0 |
| STATION 4 BUSY | 0.6774E+00 | 0.4675E+00 | 0.0 | 0.1000E+01 | 0.7502E+03 | 0.0 |
| STATION 5 BUSY | 0.1372E+01 | 0.7991E+00 | 0.0 | 0.2000E+01 | 0.7502E+03 | 0.0 |
| STATION 6 BUSY | 0.9992E+00 | 0.2827E-01 | 0.0 | 0.1000E+01 | 0.7502E+03 | 0.0 |
| STATION 7 BUSY | 0.7591E+00 | 0.4276E+00 | 0.0 | 0.1000E+01 | 0.7502E+03 | 0.0 |
| STATION 8 BUSY | 0.0 | 0.0 | 0.0 | 0.0 | 0.7502E+03 | 0.0 |

**STATISTICS FOR VARIABLES BASED ON OBSERVATION **

| | MEAN VALUE | STANDARD DEVIATION | COEFF. OF VARIATION | MINIMUM VALUE | MA XIMUM VALUE | NUMBER OF OBSERVATIONS |
|---------------------|--------------------|-----------------------|------------------------|------------------|-------------------|---------------------------|
| TIME ON CONVEYOR 1 | 0.7349E+01 | 0.4985E+01 | 0.6784E+00 | 0.0 | 0.2150E+02 | 214 |
| TIME ON CONVEYOR 2 | 0.1014E+01 | 0.1284E+01 | 0.1266E+01 | 0.0 | 0.7200E+01 | 83 |
| TIME ON CONVEYOR 3 | 0.2029E+02 | 0.1953E+01 | 0.9627E-01 | 0.0 | 0.2100E+02 | 69 |
| TIME ON CONVEYOR 4 | 0.7307E+01 | 0.6344E+01 | 0.8682E+00 | 0.0 | 0.2500E+02 | 66 |
| TIME ON CONVEYOR 5 | 0.4612E+01 | 0.4541E+01 | 0.9848E+00 | 0.0 | 0.1200E+02 | 85 |
| TIME ON CONVEYOR 6 | NO VALUES RECORDED | | | | | |
| TIME ON CONVEYOR 7 | 0.2100E+02 | 0.1494E-01 | 0.7114E-03 | 0.0 | 0.2100E+02 | 71 |
| TIME ON CONVEYOR 8 | 0.6682E+01 | 0.5135E+01 | 0.7684E+00 | 0.0 | 0.1700E+02 | 67 |
| TIME ON CONVEYOR 9 | 0.5999E+00 | 0.1836E-02 | 0.3061E-02 | 0.0 | 0.6000E+00 | 87 |
| TIME ON CONVEYOR 10 | 0.5999E+00 | 0.1606E-02 | 0.2677E-02 | 0.0 | 0.6000E+00 | 72 |
| TIME ON CONVEYOR 11 | 0.5999E+00 | 0.1521E-02 | 0.2535E-02 | 0.0 | 0.6000E+00 | 67 |

**STATISTICS FOR TIME-PERSISTENT VARIABLES **

| | MEAN VALUE | STANDARD DEVIATION | MINIMUM VALUE | MAXIMUM VALUE | TIME INTERVAL | CURRENT VALUE |
|--------------------|--------------------|-----------------------|------------------|------------------|------------------|------------------|
| NO. ON CONVEYOR 1 | 0.2153E+01 | 0.9410E+00 | 0.0 | 0.3000E+01 | 0.7502E+03 | 0.0 |
| NO. ON CONVEYOR 2 | 0.1405E+00 | 0.3513E+00 | 0.0 | 0.2000E+01 | 0.7502E+03 | 0.0 |
| NO. ON CONVEYOR 3 | 0.1930E+01 | 0.1550E+00 | 0.0 | 0.2000E+01 | 0.7502E+03 | 0.0 |
| NO. ON CONVEYOR 4 | 0.6573E+00 | 0.8655E+00 | 0.0 | 0.4000E+01 | 0.7502E+03 | 0.0 |
| NO. ON CONVEYOR 5 | 0.6601E+00 | 0.8139E+00 | 0.0 | 0.2000E+01 | 0.7502E+03 | 0.0 |
| NO. ON CONVEYOR 6 | NO VALUES RECORDED | | | | | |
| NO. ON CONVEYOR 7 | 0.2148E+01 | 0.5996E+00 | 0.0 | 0.2000E+01 | 0.7502E+03 | 0.0 |
| NO. ON CONVEYOR 8 | 0.6053E+00 | 0.7070E+00 | 0.0 | 0.2000E+01 | 0.7502E+03 | 0.0 |
| NO. ON CONVEYOR 9 | 0.7997E-01 | 0.2712E+00 | 0.0 | 0.1000E+01 | 0.7502E+03 | 0.0 |
| NO. ON CONVEYOR 10 | 0.6397E-01 | 0.2447E+00 | 0.0 | 0.1000E+01 | 0.7502E+03 | 0.0 |
| NO. ON CONVEYOR 11 | 0.5997E-01 | 0.2374E+00 | 0.0 | 0.1000E+01 | 0.7502E+03 | 0.0 |

FILE STATISTICS

| FILE NUMBER | ASSOCIATED NODE TYPE | AVERAGE LENGTH | STANDARD DEVIATION | MAXIMUM LENGTH | CURRENT LENGTH | AVERAGE WAITING TIME |
|----------------|-------------------------|-------------------|-----------------------|-------------------|-------------------|-------------------------|
| 1 | | 30.9309 | 20.3504 | 61 | 0 | 114.8729 |
| 2 | | 0.0 | 0.0 | 0 | 0 | 0.0 |
| 3 | | 0.0 | 0.0 | 0 | 0 | 0.0 |
| 4 | | 0.0 | 0.0 | 0 | 0 | 0.0 |
| 5 | | 1.8098 | 1.1553 | 3 | 0 | 7.2994 |
| 6 | | 0.0413 | 0.1990 | 2 | 0 | 2.3845 |
| 7 | | 0.5373 | 0.8138 | 4 | 0 | 9.1610 |
| 8 | | 0.4367 | 0.7250 | 2 | 0 | 6.3000 |
| 9 | | 1.8828 | 0.3647 | 2 | 0 | 19.8945 |
| 10 | | 0.5253 | 0.6753 | 2 | 0 | 7.5791 |
| 11 | | 0.0 | 0.0 | 0 | 0 | 0.0 |
| 12 | | 8.3085 | 2.6177 | 18 | 1 | 3.1608 |

2). Same conveyor system is simulated but a class 2 conveyor is used. Output contains the same information as before.

3). Consider a class 3 conveyor system, shown in Figure 5a. There is a little difference in interpreting the output. File 1 is used for parts waiting to enter the system. File 3 is used to hold parts while they are waiting to go from a station to a conveyor. File 2 and 12 are of no interest to user.

4). This example is to explain how the model helps in design and evaluation of alternatives. A part of Figure 2 layout, shown in Figure 6a is used for these purposes. This is transformed into network shown in Figure 6b.

Since the model only allows single location for a station, some adjustments are made in the layout to fit the model. Multi-located stations are those stations which have servers at more than one location. Station 2 has three servers at different places and each serving a different queue. It is assumed that these servers can be combined at one location and can be served through one queue. In making these adjustments, some conveyor sections are oriented differently than the layout shown in Figure 6a, and some have their capacity varied. User must also realize that the accessibility of some servers has also increased since parts do not travel to another location for service.

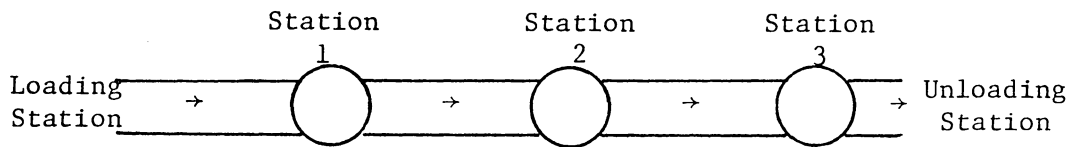


Figure 5a. Conveyor system: Example 3

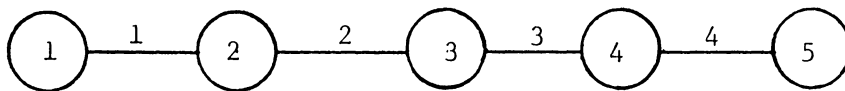


Figure 5b. Network: Example 3

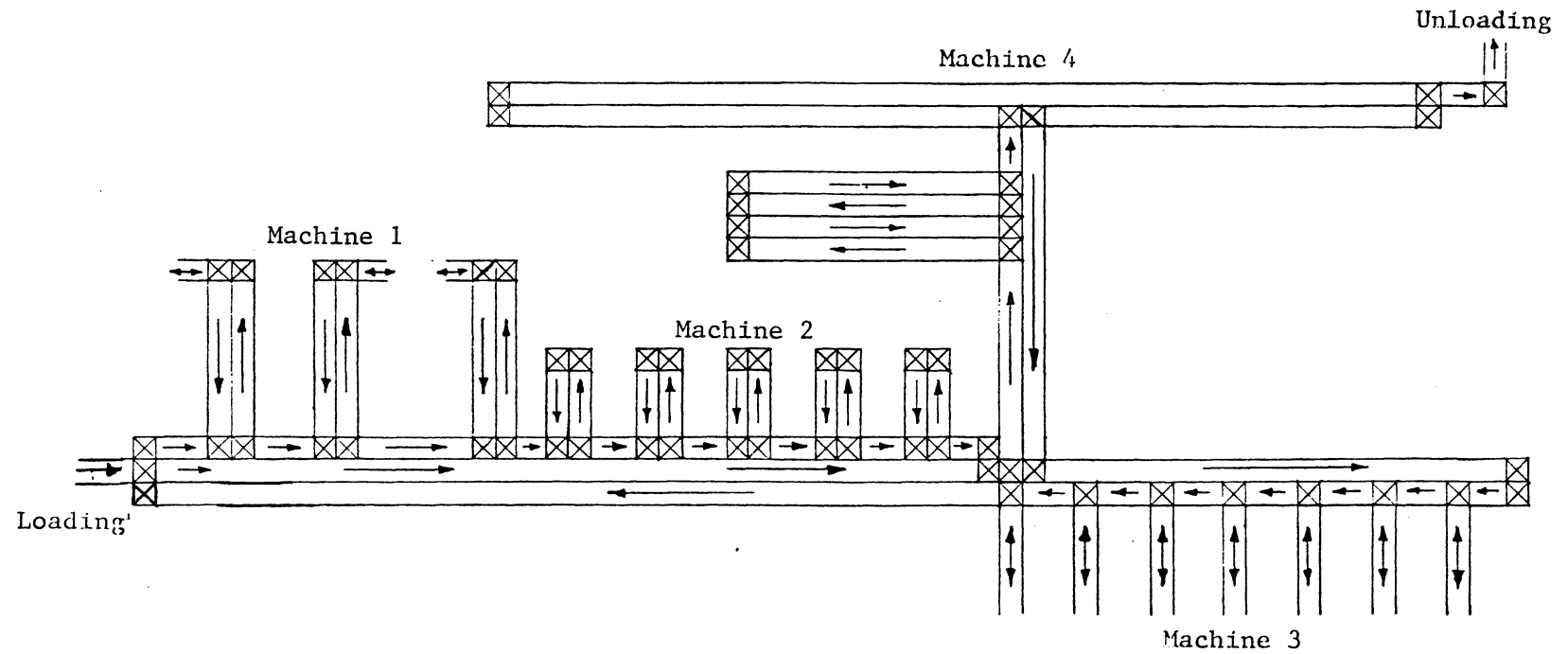


Figure 6a. Conveyor system: Example 4

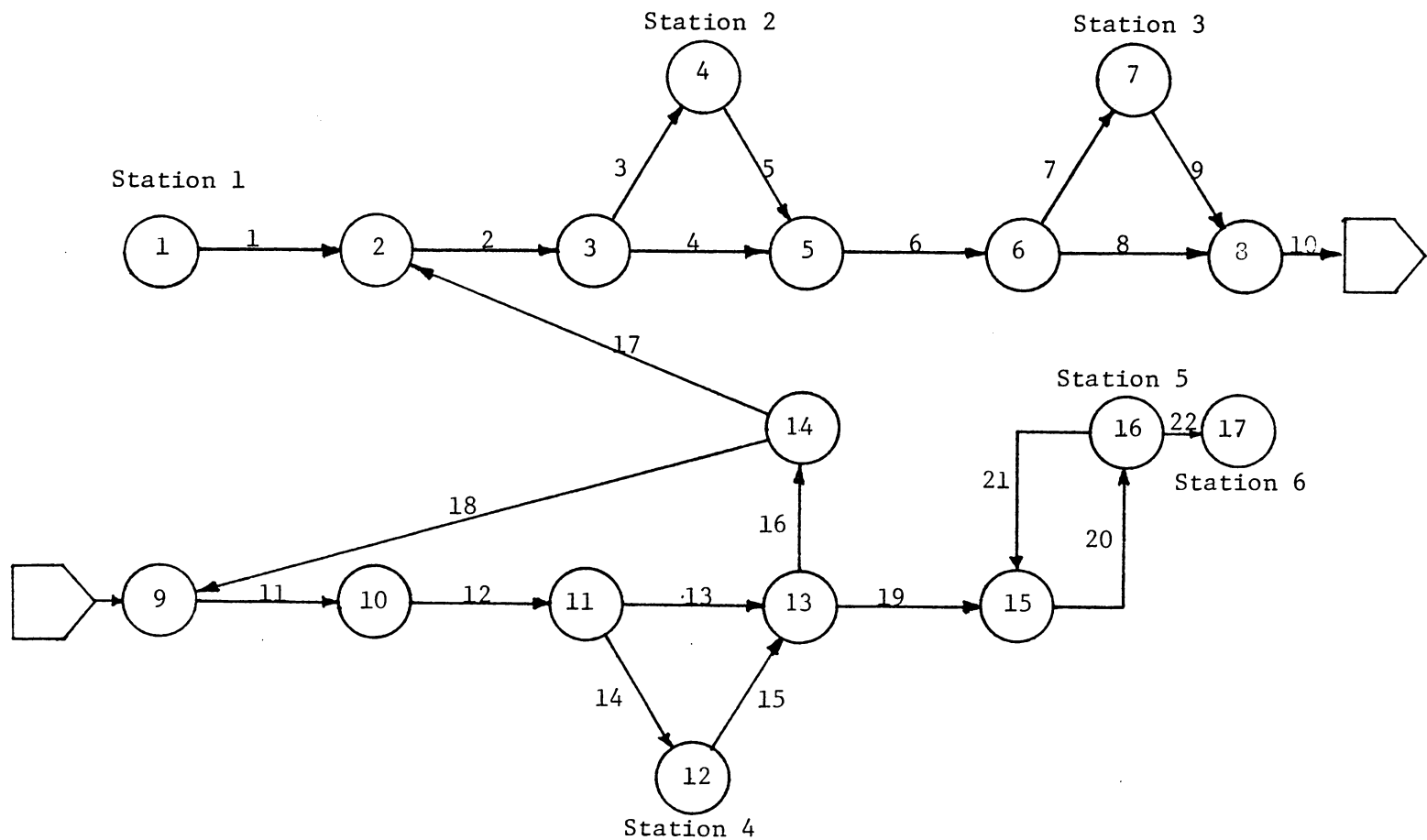


Figure 6b. Network: Example 4

The system is simulated with deterministic and probabilistic inter-arrival and service times. Execution time difference between the two models (deterministic, probabilistic), for this conveyor system, is approximately 2.50 seconds. This may vary with the size of the system being simulated.

Four runs, with probabilistic times, are included in Appendix B. The output and control variables are discussed in this section.

The system is allowing recirculation of parts. When a part cannot be served at a station, there must be a way for the part to get back to the same station. This however is not true for station 6. This problem can be resolved by either having a very small service time or having a large number of servers at that station so that no part has to stay at the station long enough to create any blockages. In the first run the model was changed so that if a part cannot be served at station 6, it stays on the conveyor and waits for service. Thus the user must have a complete recirculatory system or make sure that a non-recirculating station is capable of serving parts without any delay.

Files 4 through 9 are used when parts wait for service at a station. Since this is a recirculating system, no part will wait at a station unless the next conveyor on the path is full. As long as there is space on the path, all parts

will continue moving and no entity will be placed in files. Table 2 lists all the combinations of the number of servers tried. Table 3 lists length of each conveyor section used in four runs. Table 4 summarizes the average number of parts in files. As mentioned, run 1 is the only run where station 6, or file 9, has some entities waiting for service. This system is never completely full, therefore, no part is placed in any file (runs 1, 2, and 3 only). If a part is placed in a file, information on how long it waits in the queue, how many parts are waiting now, average number of parts waiting, and at the most how many parts are waiting, etc. is provided in the output (see Appendix B).

Table 5 indicates how many parts went through different conveyor sections. Information on how long a part stayed on a conveyor section is provided in the output (see Appendix B).

Table 6 indicates average, maximum, and current number of servers busy at each station.

After the first simulation run, it is clear that no part had to wait at any station or conveyor section. Every part went through the system without any delay. Of note, the same results were obtained when this system was simulated using a GPSS simulation model. This is good if the designer wants a system which ,in case of any machine breakdown will still perform without any delays. On the other hand the

Table 2
Number of Servers

| <u>Station Number</u> | Run <u>1</u> | Run <u>2</u> | Run <u>3</u> | Run <u>4</u> |
|----------------------------|-----------------|-----------------|-----------------|-----------------|
| 1 | 1 | 1 | 1 | 1 |
| 2 | 3 | 3 | 2 | 2 |
| 3 | 5 | 3 | 2 | 2 |
| 4 | 7 | 3 | 4 | 3 |
| 5 | 19 | 10 | 15 | 10 |
| 6 | 2 | 1 | 1 | 1 |
| — | — | — | — | — |
| Total Parts Produced | 82 | 82 | 121 | 38 |

Table 3
Length of Conveyor Sections (ft)

| Conveyor Number | Run $\frac{1}{3}$ | Run $\frac{2}{3}$ | Run $\frac{3}{3}$ | Run $\frac{4}{3}$ |
|--------------------|----------------------|----------------------|----------------------|----------------------|
| 1 | 3 | 3 | 3 | 3 |
| 2 | 6 | 6 | 6 | 6 |
| 3 | 6 | 6 | 6 | 6 |
| 4 | 3 | 3 | 3 | 3 |
| 5 | 6 | 6 | 6 | 6 |
| 6 | 6 | 6 | 6 | 6 |
| 7 | 6 | 6 | 6 | 6 |
| 8 | 3 | 3 | 3 | 3 |
| 9 | 6 | 6 | 6 | 6 |
| 10 | 6 | 6 | 6 | 6 |
| 11 | 48 | 48 | 48 | 15 |
| 12 | 9 | 9 | 9 | 9 |
| 13 | 9 | 9 | 9 | 9 |
| 14 | 3 | 3 | 3 | 3 |
| 15 | 9 | 9 | 9 | 9 |
| 16 | 62 | 15 | 15 | 15 |
| 17 | 3 | 3 | 3 | 3 |
| 18 | 62 | 9 | 9 | 9 |
| 19 | 36 | 6 | 6 | 6 |
| 20 | 45 | 15 | 15 | 15 |
| 21 | 45 | 15 | 15 | 15 |
| 22 | 45 | 15 | 15 | 15 |

Table 4
Average Number in Queues

| <u>File Number</u> | Run <u>1</u> | Run <u>2</u> | Run <u>3</u> | Run <u>4</u> |
|----------------------------|-----------------|-----------------|-----------------|-----------------|
| 1 | 0 | 0 | 0 | 14.7567 |
| 2 | 0 | 0 | 0 | 0 |
| 3 | 0 | 0 | 0 | 0 |
| 4 | 0 | 0 | 0 | 0 |
| 5 | 0 | 0 | 0 | 0.8340 |
| 6 | 0 | 0 | 0 | 0.8381 |
| 7 | 0 | 0 | 0 | 1.7208 |
| 8 | 0 | 0 | 0 | 4.0753 |
| 9 | 0.0131 | 0 | 0 | 0 |
| — | — | — | — | — |
| Total Parts Produced | 82 | 82 | 121 | 38 |

Table 5
Parts Through System

| Conveyor Number | Run <u>1</u> 97 | Run <u>2</u> 97 | Run <u>3</u> 142 | Run <u>4</u> 91 |
|--------------------|-----------------------|-----------------------|------------------------|-----------------------|
| 1 | | | | |
| 2 | 97 | 97 | 142 | 640 |
| 3 | 97 | 97 | 142 | 366 |
| 4 | 0 | 0 | 0 | 272 |
| 5 | 96 | 96 | 141 | 362 |
| 6 | 96 | 96 | 141 | 632 |
| 7 | 96 | 96 | 141 | 215 |
| 8 | 0 | 0 | 0 | 415 |
| 9 | 95 | 95 | 139 | 211 |
| 10 | 95 | 95 | 139 | 624 |
| 11 | 95 | 95 | 139 | 619 |
| 12 | 95 | 95 | 139 | 616 |
| 13 | 95 | 95 | 139 | 198 |
| 14 | 0 | 0 | 0 | 415 |
| 15 | 93 | 94 | 136 | 192 |
| 16 | 0 | 0 | 0 | 551 |
| 17 | 0 | 0 | 0 | 551 |
| 18 | 0 | 0 | 0 | 0 |
| 19 | 93 | 94 | 136 | 54 |
| 20 | 93 | 683 | 631 | 215 |
| 21 | 0 | 545 | 495 | 167 |
| 22 | 82 | 82 | 121 | 38 |

Table 6

Station Statistics (no. of servers busy)

| <u>Station Number</u> | <u>Mean</u> | Run 1 <u>Max</u> | <u>Current</u> |
|---------------------------|-------------|---------------------|----------------|
| 1 | 0 | 0 | 0 |
| 2 | 0.9371 | 2 | 1 |
| 3 | 0.9873 | 2 | 1 |
| 4 | 1.6950 | 3 | 2 |
| 5 | 8.9000 | 13 | 11 |
| 6 | 0.5269 | 2 | 1 |

| <u>Station Number</u> | <u>Mean</u> | Run 2 <u>Max</u> | <u>Current</u> |
|---------------------------|-------------|---------------------|----------------|
| 1 | 0 | 0 | 0 |
| 2 | 0.9371 | 2 | 1 |
| 3 | 0.9873 | 2 | 1 |
| 4 | 1.6970 | 3 | 1 |
| 5 | 8.7350 | 10 | 10 |
| 6 | 0 | 0 | 0 |

Table 6 (continued)

Station Statistics (no. of servers busy)

| <u>Station Number</u> | <u>Mean</u> | Run 3 <u>Max</u> | <u>Current</u> |
|---------------------------|-------------|---------------------|----------------|
| 1 | 0 | 0 | 0 |
| 2 | 1.3590 | 2 | 1 |
| 3 | 1.4390 | 2 | 2 |
| 4 | 2.4630 | 4 | 3 |
| 5 | 12.9500 | 15 | 15 |
| 6 | 0 | 0 | 0 |

| <u>Station Number</u> | <u>Mean</u> | Run 4 <u>Max</u> | <u>Current</u> |
|---------------------------|-------------|---------------------|----------------|
| 1 | 0 | 0 | 0 |
| 2 | 1.6830 | 2 | 0 |
| 3 | 1.6250 | 2 | 0 |
| 4 | 2.2420 | 3 | 1 |
| 5 | 8.4600 | 10 | 10 |
| 6 | 0 | 0 | 0 |

designer can certainly make changes in the system, and still have a good system (performance criteria is required to make any judgements). Number of servers can be reduced at some stations. Some conveyor sections may be eliminated, and some may be reduced in length. In terms of changes, the following may be said about any system:

1. vary conveyor speed,
2. vary capacity of conveyor sections,
3. vary number of servers at stations,
4. vary input rate, etc.

In this example, one may want to increase the input rate, reduce the number of servers, and/or reduce the capacity of some conveyor sections to increase the flow through the system.

Consider run 2, the service time at station 6 is zero, so no part has to wait there any more. The number of servers have been changed to 1, 3, 3, 3, 10, 1 at stations 1 through 6 respectively. The results have not changed much, except now there are some parts being recirculated at station 5 on conveyor sections 20 and 21. Some conveyor sections are not being used at all. There may be two reasons for this. The first is that the section is not on the shortest path, and the second is that the section is to be used for recirculation. Sections 4, 8, 14, 16, 17, 18, and 21 are to be used only for recirculation. Since no part is recirculated on 4,

8, 14, 16, 17, and 18, no statistics is obtained on these sections.

Now consider run 3, the input rate is increased and the number of servers have been changed to 1, 2, 2, 4, 15, 1 at respective stations. The results are not different from the previous runs.

Consider run 4, the length of conveyor sections and input rate have been changed. This run is to explain how one can determine if there are any blockages. Conveyor sections have been reduced in length and input rate has been increased to fill the system quickly. Let the length of the sections 1 through 22 be 3, 6, 6, 3, 6, 6, 6, 3, 6, 6, 15, 9, 9, 3, 9, 15, 3, 9, 6, 15, 15, and 15 respectively. The input rate is almost doubled. Number of servers are 1, 2, 2, 3, 10, and 1 at respective stations. Information on where blockages are created can be obtained. For example; section 20 has a capacity of 6 parts. There are 6 parts on conveyor 20. No other part can come onto conveyor 20, thus conveyor 19, which follows conveyor 20, has been blocked. Other sections have also been blocked due to either no space on the next conveyor or no server available at a station the part is at. The sections which are blocked are 1, 2, 3, 5, 6, 7, 9, 11, 12, 13, 15, and 19.

This is a proposed layout and the operational behavior of the system is not known. Some changes are made to fit the

model. It is not possible to predict how the system will behave without a detailed study of the system. Before making decisions about the system based on simulation, several replications are necessary. Based on one run, the following can be said about the system:

1. An increase in input rate increases the flow through the system.

2. An increase in input rate may increase the delay times.

3. A decrease in number of servers at some stations does not affect the performance of the system.

4. A decrease in number of servers at some stations increases flow and waiting time.

5. Some conveyor sections may be eliminated and some may be reduced in length.

The analyst using the model should have performance criteria before any changes can be made in the system.

Control Variables

Some of the control variables are mentioned in the examples above. The variables are discussed here. These variables are:

1. Conveyor speed
2. Conveyor length, or capacity
3. Number of servers
4. service time distribution

5. Input rate to the system

6. Output rate from the system

Operational behavior of any system is controlled by these variables. By varying any one of these variables, one can change the behavior of the system. Production capacity, waiting time, utilization, contents, etc. may be controlled by these variables.

If one is interested in slowing down the output rate from a system, conveyor speed may be reduced, conveyor length may be increased, number of servers may be reduced, etc. Thus the model can provide answers to questions like what happens if this, or how can this be done, etc. by varying the proper variables.

Chapter 6

CONCLUSION

In this study a simulation model is developed in SLAM II to study and analyze conveyor systems. The model is capable of simulating different conveyor systems. Conveyors are classified into three classes which include a majority of conveyor types available today. Conveyors are classified according to their operational characteristics, and are decomposed into smaller segments (subsystems). Programming modules for these segments are easier to develop than modeling the entire systems. These modules help study the characteristics of different configurations or designs. Once the methodology is clearly understood, these modules are compiled into a system.

The model is tested by applying it to different conveyor systems. Some of the examples are included in this report to help explain the mechanism of the model. Input and output of the examples are discussed to help a potential user interpret the results of the model.

Even though the model is a simulation model, it can be used as a design evaluation tool. An example is presented to show how the model can be used for these purposes. When used in a deterministic environment, the simulator is capable of providing system measures very quickly. For small

systems, the compiling and execution time is approximately 3 seconds. This time is about 5 second for large systems of about 50 nodes. The compiling and execution time is comparative against other deterministic models like GALS, and DYNAFLO, which uses 'Out of Kilter' algorithm.

The simualtion model developed in this study is tested on conveyor systems since that branch of material handling is the center of attention here. This does not mean that this model is limited to conveyor systems. With a little modification in the model, it can be applied to any fixed path material handling system. This model may very well be suited for simulating monorail and storage/retrieval systems. Users may have to do some more coding to be able to achieve this, but it certainly is not a very complicated thing to do.

This model can also be used to simulate automatic guided vehicle systems. With a little more work to incorporate, decision making at crossings, this model could be applied to that branch of material handling. When two vehicles cross (or reach) an intersectin at the same time, it is not always true that the same rule is applied to make a decision on which vehicle should go first. This decision is usually based on the status of the system and also on the priorities given to the vehicles. Currently, the model is using a first in first out rule. Parts are always served in

the order of arrival at a station. At an intersection, priority is given to the part which reaches that intersection first. This is one area where improvement is needed since first in first out rule is not applicable to all conveyor systems.

RECOMMENDATIONS FOR FUTURE RESEARCH

The model needs to be improved in many different areas. The following is a description of what can be done in the future to improve the model.

1. As mentioned above, the model needs to have the capability of imposing multiple decision making rules at intersections and also at work stations.
2. The model can be expanded to allow parts of different sizes or different dimensions (no restriction of unit load). This will increase the potential use of the model since a number of real life systems are not restricted to single dimension parts.
3. Multiple entry-points for a part are also needed in the model.
4. The model can also be excluded from the assumption of identical servers at a station. This will allow multiple operations at a station.
5. The model can also include the possibility of a conveyor failure or station breakdown. This will increase the

reliability of the conveyor system designed using the model.

6. All the parts produced at a station are not always good parts. There is a possibility of producing bad parts. This can also be included in the model.
7. Multiple queues for a station are also a possibility and will increase the capabilities of the model.
8. Multiple locations for a station can also be included in the model.
9. Sometimes it is desirable to route parts on a predefined path rather than the shortest path. User should have the capability of overriding the shortest path rule.
10. Visual output can also help the user understand the system. This model can interface with graphics routines and produce a visual description of the system at different stages during a simulation period.

BIBLIOGRAPHY

- [1] Advanced Manufacturing Methods Group, "Introductory User's Manual for GALs", Illinois Institute of Technology Research Institute, Chicago, 1978.
- [2] Agee, M. H. and Cullinane, T. P., "Analyzing Two Link Materials Handling Systems for Short Run Production Jobs", presented at the 41st National Meeting of ORSA, New Orleans, LA (1972).
- [3] Aly, A. A. and Litwhiler, Jr., D. W., "Police Briefing Station: A Location Problem", AIIE Transactions, Vol. 11, No. 1, pp. 12-22.
- [4] Anderson, D. R. and Moodie, C. L., "Optimal Buffer Storage Capacity in Production Line Systems", International Journal of Production Research, Vol. 7, No. 3, pp. 233-240 (1969).
- [5] Apple, J. M., Material Handling Systems Design, The Ronald Press Company, New York (1972).
- [6] Armstrong, R. D., Cook, W. D., and Saipe, A. L., "Optimal Batching in a Semi-Automated Order Picking System", Journal of Operational Research, Vol. 30, No. 8, pp. 711-720 (1979).
- [7] Bedworth, D. D., Industrial Systems: Planning, Analysis, Control, The Ronald Press Company, New York (1973).
- [8] Beightler, C. S. and Crisp, R. M., "A Discrete-Time Queueing Analysis of Conveyor-Serviced Production Station", Operations Research, Vol. 16, No. 5, pp. 986-1001 (1968).
- [9] Barton, J. E., "Try Simulations for Sizing Up Systems--Before It's Too Late", Production Engineering, Vol. 27, No. 12, pp. 42-45.
- [10] Biles, W. E., "Simulation of Material Handling Manufacturing", Unit and Bulk Material Handling, ASME Century 2, Emerging Technology conference, San Francisco, CA., August 19-21, 1980, pp. 241-246.

- [11] Bournazos, M. and Singh, G., "Material Handling Systems", Highway Road Construction International, Vol. 44, No. 1802, pp. 18-19, 22-23, 25 (1976).
- [12] Buzacott, J. A., "Automatic Transfer Lines with Buffer Stocks", International Journal of Production Research, Vol. 5, No. 3, pp. 183-200.
- [13] Buzacott, J. A., "Prediction of the Efficiency of Production Systems without Internal Storage", International Journal of Production Research, Vol. 6, No. 3, pp. 173-188.
- [14] Buzzacott, J. A., "Methods of Reliability Analysis of Production Systems Subject to Breakdowns", Operations Research and Reliability, ed. by D. Grouchko, Gordon and Breach, pp. 211-232 (1971).
- [15] Buzacott, J. A. and Shantikumar, J. G., "Models for Understanding Flexible Manufacturing Systems", AIIE Transactions, Vol. 12, No. 4, pp. 339-349 (1980).
- [16] Carew, V. E., "Economic Considerations for Automating Unit Handling and Storage", ASME Paper n-80-WA/MH-14 presented at the Winter Annual Meeting, Nov. 16-21, 1980, Chicago, IL.
- [17] Chen, T. C. and Terrell, M. P., "A Modular General Purpose Approach to the Simulation of Complex Multi-Loop, Multi-Floor Conveyor System", Presented at the 1st National Computers and Industrial Engineering Conference, Tulsa, OK (1976).
- [18] Conway, R. W., Maxwell, W. L., and Miller, L. W., Theory of Scheduling, Addison-Wesley, Reading, Massachusetts (1967).
- [19] Crisp, R. M., Skeith, R. W., and Barnes, J. W., "A Simulated Study of Conveyor-Serviced Production Stations", International Journal of Production Research, Vol. 7, No. 4, pp. 301-309 (1969).
- [20] Donnenbring, D. G. and Khumawala, B. M., "An Investigation of Branch and Bound Methods for Solving the Segregated Storage Problem", AIIE Transactions, Vol. 5, No. 3, pp. 265-274.
- [21] Dantzig, G. B., Harvey, R. P., Landsdowne, Z. F., Robinson, D. W., and Maier, S. F., "Formulating and Solving the Network Design Problem by Decomposition", Transportation Research-B, Vol. 13B, pp. 5-17 (1979).

- [22] Devore, J. C., "Better Ways for Materials Handling", Industrial Engineering, Vol. 10, No. 9, pp. 31-41 (1978).
- [23] Disney, R. L., "Some Multichannel Queueing Problems with Ordered Entry", Journal of Industrial Engineering, Vol. 13, No. 1, pp. 46-48 (1962).
- [24] Douglas, D. E., Mazzaro, M. T., and Myers, L. J., "A GPSS V Simulation of a Computer Controlled Warehouse Picking System", Proceedings Winter Simulation Conference, Vol. 2, pp. 708-719 (1978), Florida.
- [25] Elsayed, E. A. and Proctor, C. L., "Ordered Entry and Random Choice Conveyors with Multiple Poisson Inputs", International Journal of Production Research, Vol. 15, No. 5, pp. 439-451.
- [26] Francis, R. L. and White, J. A., Facility Layout and Location: An Analytical Approach, Prentice-Hall Inc., Englewood Cliffs, New Jersey (1974).
- [27] Gensch, D. H., "An Industrial Application of the Traveling Salesman's Subtour Problem", AIIE Transactions, Vol. 10, No. 4, pp. 362-370 (1978).
- [28] Greshwin, S. B. and Berman, O. "Analysis of Transfer Lines Consisting of Two Unreliable Machines with Random Processing Times and Finite Storage Buffers", AIIE Transactions, Vol. 13, No. 1, pp. 2-11 (1981).
- [29] Gourley, R. L. and Terrell, M. P., "A Modular Utility Simulation Model Used to Test and Analyze Proposed Conveyor Layouts", Joint ORSA/TIMS National Meeting, San Juan, Puerto Rico (1974).
- [30] Gourley, R. L. and Terrell, M. P., "A Modular General Purpose Approach to the Simulation of Constant Speed Discretely Spaced Recirculating Conveyor Systems", 1974 Systems Engineering Conference, AIIE, Minneapolis, Minnesota.
- [31] Gregory, G. and Litton, C. D., "A Conveyor Model with Exponential Service Times", International Journal of Production Research, Vol. 13, No. 1, pp. 1-7 (1975).
- [32] Groover, M. P., Automation, Production Systems, and Computer-Aided Manufacturing, Prentice-Hall Inc., Englewood Cliffs, New Jersey (1980).

- [33] Hamada, N., "Optimal Scheduling and Evaluation of Computer-Controlled Automated Warehouse System", Electrical Engineering in Japan, Vol. 96, No. 2, pp. 111-120 (1976).
- [34] Hamada, N., "Optimal Scheduling of the Packing System in an Automated Warehouse and Its Evaluation", Electrical Engineering in Japan, Vol. 96, No. 6, pp. 145-153 (1976).
- [35] Helgeson, W. R., "Planning for the Use of Overhead Monorail Non-Reversing Loop Type Conveyor Systems for Storage and Delivery", The Journal of Industrial Engineering, Vol. 11, No. 6, pp. 480-492 (1960).
- [36] Hill, I., "Who Should be Involved in the Material Handling Organization?", Industrial Engineering, Vol. 13, No. 4, p. 44-50 (1981).
- [37] Hill, J. M., "Computers and AS/RS Revolutionize Warehousing", Industrial Engineering, Vol. 12, No. 6, pp. 34-45 (1980).
- [38] Hogg, G. L. and Phillips, D. T., "Scheduling Multi-Cavity Die Casting Machines with Two Levels of Set-up Cost", Proceedings - AIIE 1977 Spring Annual Conference, Dallas, Texas, May 24-27, pp.
- [39] Hutchinson, G. K., "Flexible Manufacturing Systems in the United States", ASME Joint Automatic Control conference, Denver, June 17-20, 1979.
- [40] Kimbler, D. L. and Wysk, R. A., "Applications of Operations Research in Manufacturing Systems", Proceedings - 1980 Fall IE Conference, Minneapolis, Minnesota, Dec. 7-10, pp. 181-189.
- [41] Kwo, T. T., "A Theory of Conveyors", Management Science, Vol. 5, No. 1, pp. 51-71 (1958).
- [42] Kwo, T. T., "A Method for Designing Irreversible Overhead Loop Conveyors", The Journal of Industrial Engineering, Vol. 11, No. 6, pp. 459-466 (1960).
- [43] Matsui, M. and Shingu, T., "A Queueing Analysis of Conveyor - Serviced Production Station and the Optimal Range Strategy", AIIE Transactions, Vol. 10, No. 1, pp. 89-95.

- [44] Maxwell, W. L., "Solving Material Handling Design Problems with OR", Industrial Engineering, Vol. 13, No. 4, pp. 58-66 (1981).
- [45] Maxwell, W. L. and Wilson, R. C., "Dynamic Network Flow Modelling of Fixed Path Material Handling Systems", AIIE Transactions, Vol. 13, No. 1, pp. 12-21 (1981).
- [46] Mayer, H., "Introduction to Conveyor Theory", Western Electric Engineer, Vol. 4, No. 1, pp. 43-47 (1960).
- [47] Mayer, R. J. and Talavage, J. J., "The Optimal Planning of Computerized Manufacturing Systems; Simulation of a Computerized Manufacturing System, Report No. 4, NSF Grant No. APR74 15256, Nov. 1976.
- [48] Mitsumori, S., "Optimum Schedule Control of Conveyor Line", IEEE transactions Automatic Control, Vol. AC14, No. 6, pp. 633-639 (1969).
- [49] Moodie, C. L., Sadowski, R. P., and Hill, Jr., T. W., "Optimal Design for Production Facilities," AIIE Transactions, Vol. 5, No. 3, pp. 230-236.
- [50] Moore, J. Plant Layout and Design, Macmillan Publishing Co., Inc., New York (1962).
- [51] Morgan, J. T., "Two-Link Material Handling System", Operations Research, Vol. 14, No. 1, pp. 16-24.
- [52] Morgan, Jr., F. C., "Design an Optimum Flow In-process Handling System", Industrial Engineering, Vol. 13, No. 4, pp. 94-98 (1981).
- [53] Morris, W. T., Analysis for Materials Handling Management, Richard D. Irwin, Inc. (1962).
- [54] Muth, E. J., "Analysis of Closed-Loop Conveyor Systems", AIE Transactions, Vol. 4, No. 2, pp. 134-143 (1972).
- [55] Muth, E. J., "Analysis of Closed-Loop Conveyor Systems, the Discrete Flow Case", AIIE Transactions, Vol. 6, No. 1., pp. 73-82 (1974).
- [56] Muth, E. J., "Modeling and System Analysis of Multistation Closed-Loop Conveyors", International Journal of Production Research, Vol. 13, No. 6, pp. 559-566 (1975).

- [57] Muth, E. J., "A Model of a Closed-Loop Conveyor with Random Material Flow", AIIE Transactions, Vol. 9, No. 4, pp. 345-351 (1977).
- [58] Muth, E. J. and White, J. A., "Conveyor Theory: A Survey", AIIE Transactions, Vol. 11, No. 4, pp. 270-277 (1979).
- [59] Muther, R. and Haganas, K., Systematic Handling Analysis, Management and Industrial Research Publication, Missouri (1978).
- [60] Neible, B. W., Motion and Time Study, R. D. Irwin, Inc., Illinois (1976).
- [61] Nof, S. Y., Whinston, A. B., and Bullers, W. I., "Control and Decision Support in Automatic Manufacturing Systems", AIIE Transactions, Vol. 12, No. 2, pp. 156-167 (1980).
- [62] Occhipinti, V. M., "The Application of Computer Technology to Distribution", ASME paper n-80-WA/MH-14, presented at the Winter Annual Meeting, Nov. 16-21, 1980, Chicago, IL.
- [63] Ohta, H., "GERT Analysis of a Single Conveyor System", International Journal of Production Research (GB), Vol. 17, No. 4, pp. 405-410.
- [64] Pearman, A. D., "The Structure of the Solution Set to Network Optimization Problems", Transportation Research-B, Vol. 13B, pp. 81-90 (1979).
- [65] Phillips, D. T., "Simulation of Material Handling Systems: When and Which Methodology", Industrial Engineering, Vol. 12, No. 9, pp. 64-77 (1980).
- [66] Phillip, D. T. and Lytton, R. L., "A Resource Constrained Capital Budgeting Model for State Highway Rehabilitation and Maintenance", Proceedings - AIIE 1977 Spring Annual Conference, Dallas, Texas, May 24-27, pp. 297-304.
- [67] Phillips, D. T. and Skeith, R. W., "Ordered Entry Queueing Networks with Multiple Servers and Multiple Queues", AIIE Transactions, Vol. 1, No. 4, pp. 333-342 (1969).
- [68] Phillips, D. T., et al, "GEMS: A Generalized Manufacturing Simulator", Proceedings of the Sixth NSF Gran-

tee's Conference on Production Research and Technology, West Lafayette, IN, September, 1978.

- [69] Pritsker, A. A. B., "Application of Multichannel Queueing Results to the Analysis of Conveyor Systems", Journal of Industrial Engineering, Vol. 17, No. 7, pp. 14-21 (1966).
- [70] Pritsker, A. A. B. and Pegden, C. D., Introduction to Simulation and SLAM, John Wiley and Sons, New York (1979).
- [71] Proctor, C. L., Elsayed, E. L., and Ragab, A. M., "A Two Channel Closed Loop Conveyor with Multiple Poisson Input and Storage at the Second Channel", Modeling and Simulation, Vol. 7, Pt. 1, Pitt. PA., 26-27 April (1976), pp. 195-200.
- [72] Proctor, C. L., El Sayed, E. A., and Elayat, H. A., "A Conveyor System with Homogeneous and Heterogeneous Servers with Dual Input", International Journal of Production Research, Vol. 15, No. 1, pp. 73-85 (1977).
- [73] Ratliff, H. D., "Network Models for Production Scheduling Problems with Convex Cost and Batch Processing", AIIE Transactions, Vol. 10, No. 1, pp. 104-108.
- [74] Reis, I. L., Brennan, J. J., and Crisp, R. M., "A Markovian Analysis for Delay at Conveyor Serviced Production Stations", International Journal of Production Research, Vol. 5, No. 2, pp. 201-211 (1967).
- [75] Reis, I. L., Dunlap, L. L., and Schneider, M. H., "Conveyor Theory: The Individual Station", The Journal of Industrial Engineering, Vol. 14, No. 4, pp. 212-217 (1963).
- [76] Reis, I. L. and Hatcher, J. M., "Probabilistic Conveyor Analysis", International Journal of Production Research, Vol. 2, No. 3, pp. 186-194 (1963).
- [77] Reis, I. L. and Schneider, M. H., "Probabilistic Conveyor Decisions", Kansas State University Experiment Station Special Bulletin No. 19 (1962).
- [78] Rosenman, M. A. and Gero, J. S., "Heuristic Nonserial Dynamic Programming for Large Problems", Engineering Optimization, Vol. 4, pp. 167-178 (1980).

- [79] Runner, J. A. and Leimkuhler, F. F., "The Optimal Planning of Computerized Manufacturing Systems: CAMS A Simulation Analysis Model for Computer-Aided Manufacturing Systems", Report No. 13, NSF Grant No. APR74 15256, Dec. 1978.
- [80] Schruben, L. W. and Moser, P. I., "Study of an Automated Materials Handling System", Proceedings Winter Simulation Conference, Vol. 2, pp. 700-707 (1978), Florida.
- [81] Solberg, J. J., "Quantitative Design Tools for Computerized Manufacturing Systems", Sixth North American Metalworking Research Conference Proceedings, April 1978.
- [82] Steckle, K. E. and Solberg, J. J., "The Optimal Planning of Computerized Manufacturing Systems; Scheduling of Operations in a Computerized Manufacturing System", Report No. 10, NSF Grant No. APR74 15256, Dec. 1977.
- [83] Taha, H. A., Operations Research, Macmillan Publishing Co., Inc., New York (1976).
- [84] Taylor III, B. W., Moore, L. J., and Hammesfahr, R. D., "Global Analysis of a Multi-Product, Multi-Line Production System Using Q-GERT Modeling and Simulation", AIIE Transactions, Vol. 12, No. 2, pp. 145-155 (1980).
- [85] Thompson, J. A., "A Linear Programming Model for Labor Planning", Proceedings - AIIE 1978 Spring Annual Conference, Toronto, Canada, May 25-26, pp. 289-296.
- [86] White, J. A., "Conveyor Theory: A Normative Approach", School of Industrial and Systems Engineering paper, Georgia Institute of Technology (1976).
- [87] White, J. A. and Woodbury, W. C., "Validation of a Conservation of Flow Approach in Modelling a Recirculating Conveyor", School of Industrial and Systems Engineering paper, Georgia Institute of Technology (1976).

APPENDIX A

PROGRAM LISTING

```

C          *      MAIN PROGRAM      *
      DIMENSION NSET(10000)
      COMMON/SCOM1/ATTRIB(100),DD(100),DDL(100),DTNOW,II,MFA,MSTOP,
      $NCLNR,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100),TNEXT,TNOW,
      $XX(100)
      COMMON/UCOM1/NCTYPE,NNODE,LINKN(100,8),NARC,ASIZE(100),
      1LOCATA(100,2),SPEED,USIZE,NSAREA,NSERV(25),LOCATS(25,11),
      2NPART,NVISIT,NROUTE(25,25),NEPT,ARRNT(25),XMEAN(25),ULO(25),
      3UHI(25),LOCATP(25),NCAP(100),
      4TTIME(100),ISHORT(200,100),DIST(100,100),IPATH(100),
      5NPATH(100),CAP(100),NVST(100),IBLOCK(25),MMST(100),TSTOP(100),
      6NRECIR,DISTLD,TMLOAD(100),CC(100),TOTAL,TOTPR
      7,XLO(25),XMODE(25),XMN(25),STD(25),ARINT,AARRNT(25),NMODE(25),
      8XXMEAN(25),UULO(25),XXLO(25),XXHI(25),XXMODE(25),XXMN(25)
      9,SSTD(25),MODE(25),XHI(25),UUHI(25)
      COMMON/UCOM2/SAMP(100,5),TSAMP(100,6),TIMCLR,TIMSTP
      COMMON/UCOM3/MAXPRT(25)
      COMMON QSET(10000)
      EQUIVALENCE (NSET(1),QSET(1))
      NNSET=10000
      NCRDR=5
      NPRNT=6
      NTAPE=7
      CALL SLAM
      STOP
      END

```

```

SUBROUTINE INTLC
C   THIS SUBROUTINE STARTS THE SIMULATION. CALLS FOR OTHER
C   ROUTINES TO INPUT DATA AND TO INITIALIZE VALUES TO VARIABLES.
COMMON/SCOM1/ATRI(100),DD(100),DDL(100),DTNOW,II,MFA,MSTOP,
$NCLNR,NCRDR,NPRNT,NNRUN,NNSSET,NTAPE,SS(100),SSL(100),TNEXT,TNOW,
$XX(100)
COMMON/UCOM1/NCTYPE,NNODE,LINKN(100,8),NARC,ASIZE(100),
1LOCATA(100,2),SPEED,USIZE,NSAREA,NSERV(25),LOCATS(25,11),
2NPART,NVISIT,NROUTE(25,25),NEPT,ARRNT(25),XMEAN(25),ULO(25),
3UHI(25),LOCATP(25),NCAP(100),
4TTIME(100),ISHORT(200,100),DIST(100,100),IPATH(100),
5NPATH(100),CAP(100),NVST(25),IBLOCK(25),MMST(100),TSTOP(100),
6NRECIR,DISTLD,TMLOAD(100),CC(100),TOTAL,TOTPR
7,XLO(25),XMODE(25),XMN(25),STD(25),ARINT,AARRNT(25),NMODE(25),
8XXMEAN(25),UULO(25),XXLO(25),XXHI(25),XXMODE(25),XXMN(25)
9,SSTD(25),MODE(25),XHI(25),UUHI(25)
COMMON/UCOM2/SAMP(100,5),TSAMP(100,6),TIMCLR,TIMSTP
COMMON/UCOM3/MAXPRT(25)
CALL INPUT
CALL CALC
C   SCHEDULE THE FIRST ARRIVAL OF ALL PARTS AND ALSO THE LOADING POINT
C   ON THE CONVEYOR (IF APPROPRIATE).
DO 10 I=1,NPART
  ATRI(4)=I
  CALL ARIVAL(I,ARR)
  CALL SCHDL(1,ARR,ATRI)
10 CONTINUE
  IF(NCTYPE.NE.3) GO TO 30
  DO 20 I=1,NARC
    ATRI(9)=I
    CALL SCHDL(4,ARINT,ATRI)
20 CONTINUE
C   INITIALIZE ALL STATISTICAL VARIABLES
30 CALL CCLCT(-1,0.,NARC)
  CALL MMSTT(-1,TNOW,0.,NARC)
C   CLEAR ALL STATISTICS IF ASKED FOR
  IF(TIMCLR.GT.0) CALL SCHDL(5,TIMCLR,A)
C   SCHEDULE SUMMARY OF RESULTS AFTER STOPPING TIME
  CALL SCHDL(6,TIMSTP,A)
  RETURN
END

```

```

SUBROUTINE INPUT
C   INPUT DATA TO THE SYSTEM.
COMMON/SCOM1/ATRIB(100),DD(100),DDL(100),DTNOW,II,MFA,MSTOP,
SNCLNR,NCRDR,NPRNT,NNRUN,NNSE,NTAPE,SS(100),SSL(100),TNEXT,TNOW,
$XX(100)
COMMON/UCOM1/NCTYPE,NNODE,LINKN(100,8),NARC,ASIZE(100),
1LOCATA(100,2),SPEED,USIZE,NSAREA,NSERV(25),LOCATS(25,11),
2NPART,NVISIT,NROUTE(25,25),NEPT,ARRNT(25),XMEAN(25),ULO(25),
3UHI(25),LOCATP(25),NCAP(100),
4TTIME(100),ISHORT(200,100),DIST(100,100),IPATH(100),
5NPATH(100),CAP(100),NVST(25),IBLOCK(25),MMST(100),TSTOP(100),
6NRECIR,DISTLD,TMLOAD(100),CC(100),TOTAL,TOTPR
7,XLO(25),XMODE(25),XMN(25),STD(25),ARINT,AARRNT(25),NMODE(25),
8XXMEAN(25),UULO(25),XXLO(25),XXHI(25),XXMODE(25),XXMN(25)
9,SSTD(25),MODE(25),XHI(25),UUHI(25)
COMMON/UCOM2/SAMP(100,5),TSAMP(100,6),TIMCLR,TIMSTP
COMMON/UCOM3/MAXPRT(25)
C   CONVEYOR CLASS
READ(5,*) NCTYPE
C   RECIRCULATION INDEX
READ(5,*) NRECIR
C   DISTANCE BETWEEN TWO LOADING POINTS
IF(NCTYPE.EQ.3) READ(5,*) DISTLD
C   NUMBER OF NODES
READ(5,*) NNODE
C   NODE LINKAGES
DO 10 I=1,NNODE
READ(5,*) (LINKN(I,J),J=1,8)
10 CONTINUE
C   NUMBER OF ARCS(CONVEYOR SECTIONS)
READ(5,*) NARC
C   SIZE OF EACH SECTION
READ(5,*) (ASIZE(I),I=1,NARC)
C   EMANATING AND TERMINATING NODE OF EACH ARC
DO 20 I=1,NARC
READ(5,*) (LOCATA(I,J),J=1,2)
20 CONTINUE
C   CONVEYOR SPEED
READ(5,*) SPEED
C   UNIT SIZE
READ(5,*) USIZE
C   NUMBER OF SERVICE AREAS
READ(5,*) NSAREA
C   NUMBER OF SERVERS PER SERVICE AREA
READ(5,*) (NSERV(I),I=1,NSAREA)
C   SERVICE TIME DISTRIBUTION
DO 30 I=1,NSAREA
READ(5,*) MODE(I)
IF(MODE(I).EQ.1) READ(5,*) ARRNT(I)
IF(MODE(I).EQ.3) READ(5,*) XMEAN(I)
IF(MODE(I).EQ.4) READ(5,*) ULO(I),UHI(I)
IF(MODE(I).EQ.5) READ(5,*) XLO(I),XMODE(I),XHI(I)
IF(MODE(I).EQ.6) READ(5,*) XMN(I),STD(I)
30 CONTINUE
C   LOCATION OF SERVICE AREAS

```

```

      DO 40 I=1,NSAREA
      READ(5,*) (LOCATS(I,J),J=1,11)
C 40 CONTINUE
      TYPE OF JOBS OR PARTS
      READ(5,*) NPART
      DO 50 I=1,NPART
C  NUMBER OF SERVICE AREAS TO BE VISITED
      READ(5,*) NVISIT
C  ORDER OF SERVICE AREAS TO BE VISITED
      READ(5,*) (NROUTE(I,J),J=1,NVISIT)
      NVST(I)=NVISIT
C 50 CONTINUE
      DO 60 I=1,NPART
C  NUMBER OF ENTRY POINTS
      READ(5,*) NEPT
C  LOCATION OF ENTRY POINTS
      READ(5,*) LOCATP(I)
C 60 CONTINUE
      DO 70 I=1,NPART
C  INTER-ARRIVAL TIME DISTRIBUTION
      READ(5,*) NMODE(I)
      IF(NMODE(I).EQ.1) READ(5,*) AARRNT(I)
      IF(NMODE(I).EQ.3) READ(5,*) XXMEAN(I)
      IF(NMODE(I).EQ.4) READ(5,*) UULO(I),UUHI(I)
      IF(NMODE(I).EQ.5) READ(5,*) XXLO(I),XXMODE(I),XXHI(I)
      IF(NMODE(I).EQ.6) READ(5,*) XXMN(I),SSTD(I)
C 70 CONTINUE
      MAXIMUM NUMBER OF EACH PARTS ALLOWED IN THE SYSTEM
      READ(5,*) (MAXPRT(I),I=1,NPART)
C  TIME TO CLEAR ALL STATISTICS (IF NOT NEEDED ENTER 0)
      READ(5,*) TIMCLR
C  TIME TO STOP SIMULATION (IF DEPENDENT ON NUMBER OF PARTS
C  COMPLETED ENTER A VERY LARGE NUMBER)
      READ(5,*) TIMSTP
      RETURN
      END

```

```

SUBROUTINE SERVIC(IP,SVT)
C  ASSIGN SERVICE TIMES TO SERVICE STATIONS.
C  SERVICE TIME CAN BE ANY OF THE FOLLOWING:
C      CONSTANT
C      RANDOM
C      EXPONENTIALLY DISTRIBUTED
C      UNIFORMLY DISTRIBUTED
C      TRIANGULARLY DISTRIBUTED
C      NORMALLY DISTRIBUTED
COMMON/SCOM1/ATRI(100),DD(100),DDL(100),DTNOW,II,MFA,MSTOP,
$NCLNR,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100),TNEXT,TNOW,
$XX(100)
COMMON/UCOM1/NCTYPE,NNODE,LINKN(100,8),NARC,ASIZE(100),
1LOCATA(100,2),SPEED,USIZE,NSAREA,NSERV(25),LOCATS(25,11),
2NPART,NVISIT,NROUTE(25,25),NEPT,ARRNT(25),XMEAN(25),ULO(25),
3UHI(25),LOCATP(25),NCAP(100),
4TTIME(100),ISHORT(200,100),DIST(100,100),IPATH(100),
5NPATH(100),CAP(100),NVST(25),IBLOCK(25),MMST(100),TSTOP(100),
6NRECIR,DISTLD,TMLOAD(100),CC(100),TOTAL,TOTPR
7,XLO(25),XMODE(25),XMN(25),STD(25),ARINT,AARRNT(25),NMODE(25),
8XXMEAN(25),UULO(25),XXLO(25),XXHI(25),XXMODE(25),XXMN(25)
9,SSTD(25),MODE(25),XHI(25),UUHI(25)
I=IP
IF(MODE(I).EQ.1) SVT=ARRNT(I)
IF(MODE(I).EQ.2) SVT=DRAND(I)
IF(MODE(I).EQ.3) SVT=EXPON(XMEAN(I),I)
IF(MODE(I).EQ.4) SVT=UNFRM(ULO(I),UHI(I),I)
IF(MODE(I).EQ.5) SVT=TRIAG(XLO(I),XMODE(I),XHI(I),I)
IF(MODE(I).EQ.6) SVT=RNORM(XMN(I),STD(I),I)
RETURN
END

```

```

C      SUBROUTINE ARIVAL(IP,ARR)
C      ASSIGN ARRIVAL RATES TO DIFFERENT TYPES OF PARTS.
C      ARRIVAL RATE CAN BE ANY OF THE FOLLOWING:
C          CONSTANT
C          RANDOM
C          EXPONENTIALLY DISTRIBUTED
C          UNIFORMLY DISTRIBUTED
C          TRIANGULARLY DISTRIBUTED
C          NORMALLY DISTRIBUTED
C      COMMON/SCOM1/ATTRIB(100),DD(100),DDL(100),DTNOW,II,MFA,MSTOP,
$NCLNR,NCRDR,NPRNT,NNRUN,NNSSET,NTAPE,SS(100),SSL(100),TNEXT,TNOW,
$XX(100)
C      COMMON/UCOM1/NCTYPE,NNODE,LINKN(100,8),NARC,ASIZE(100),
1LOCATA(100,2),SPEED,USIZE,NSAREA,NSERV(25),LOCATS(25,11),
2NPART,NVISIT,NROUTE(25,25),NEPT,ARRNT(25),XMEAN(25),ULO(25),
3UHI(25),LOCATP(25),NCAP(100),
4TTIME(100),ISHORT(200,100),DIST(100,100),IPATH(100),
5NPATH(100),CAP(100),NVST(25),IBLOCK(25),MMST(100),TSTOP(100),
6NRECIR,DISTLD,TMLOAD(100),CC(100),TOTAL,TOTPR
7,XLO(25),XMODE(25),XMN(25),STD(25),ARINT,AARRNT(25),NMODE(25),
8XXMEAN(25),UULO(25),XXLO(25),XXHI(25),XXMODE(25),XXMN(25)
9,SSTD(25),MODE(25),XHI(25),UUHI(25)
C      I=IP
C      IF(NMODE(I).EQ.1) ARR=AARRNT(I)
C      IF(NMODE(I).EQ.2) ARR=DRAND(I)
C      IF(NMODE(I).EQ.3) ARR=EXPON(XXMEAN(I),I)
C      IF(NMODE(I).EQ.4) ARR=UNFRM(UULO(I),UUHI(I),I)
C      IF(NMODE(I).EQ.5) ARR=TRIAG(XXLO(I),XXMODE(I),XXHI(I),I)
C      IF(NMODE(I).EQ.6) ARR=RNORM(XXMN(I),SSTD(I),I)
C      RETURN
C      END

```



```

SUBROUTINE CALC
C   INITIALIZE VALUES OF VARIABLES. ALSO CALCULATE OTHER VARIABLES
C   AS CAPACITY OF EACH ARC, TRAVEL TIMES, AND TIME BETWEEN TWO
C   CONSECUTIVE LOADING POINTS, ETC.
COMMON/SCOM1/ATRIB(100),DD(100),DDL(100),DTNOW,II,MFA,MSTOP,
$NCLNR,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100),TNEXT,TNOW,
$XX(100)
COMMON/UCOM1/NCTYPE,NNODE,LINKN(100,8),NARC,ASIZE(100),
1LOCATA(100,2),SPEED,USIZE,NSAREA,NSERV(25),LOCATS(25,11),
2NPART,NVISIT,NROUTE(25,25),NEPT,ARRNT(25),XMEAN(25),ULO(25),
3UHI(25),LOCATP(25),NCAP(100),
4TTIME(100),ISHORT(200,100),DIST(100,100),IPATH(100),
5NPATH(100),CAP(100),NVST(25),IBLOCK(25),MMST(100),TSTOP(100),
6NRECIR,DISTLD,TMLOAD(100),CC(100),TOTAL,TOTPR
7,XLO(25),XMODE(25),XMN(25),STD(25),ARINT,AARRNT(25),NMODE(25),
8XXMEAN(25),UULO(25),XXLO(25),XXHI(25),XXMODE(25),XXMN(25)
9,SSTD(25),MODE(25),XHI(25),UUHI(25)
COMMON/UCOM3/MAXPRT(25)
C   DETERMINE CAPACITIES
DO 10 I=1,NARC
NCAP(I)=ASIZE(I)/USIZE
CAP(I)=NCAP(I)
C   DETERMINE TRAVEL TIMES
TTIME(I)=ASIZE(I)/SPEED
C   INITIALIZE NUMBER ON EACH ARC TO BE ZERO
CC(I)=0.
10 CONTINUE
C   FIND TIME BETWEEN TWO LOADING POINTS
IF(NCTYPE.EQ.3) ARINT=DISTLD/SPEED
DO 20 I=1,NSAREA
C   INITIALIZE EACH SERVER TO BE IDLE, AND THAT NO SERVER IS BLOCKED
C   INITIALLY.
XX(I)=0.
IBLOCK(I)=0
20 CONTINUE
C   CALL FOR ROUTINE TO FIND THE SHORTEST PATHS AMONG ALL NODES.
CALL SHORT
TOTAL=0.
TOTPRT=0.
C   TOTAL NUMBER OF PATS ALLOWED IN THE SYSTEM
DO 30 I=1,NPART
TOTPRT=TOTPRT+MAXPRT(I)
30 CONTINUE
RETURN
END

```

```

SUBROUTINE SHORT
C DETERMINE THE SHORTEST PATHS AMONG ALL NODES.
COMMON/SCOM1/ATRIB(100),DD(100),DDL(100),DTNOW,II,MFA,MSTOP,
$NCLNR,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100),TNEXT,TNOW,
$XX(100)
COMMON/UCOM1/NCTYPE,NNODE,LINKN(100,8),NARC,ASIZE(100),
1LOCATA(100,2),SPEED,USIZE,NSAREA,NSERV(25),LOCATS(25,11),
2NPART,NVISIT,NROUTE(25,25),NEPT,ARRNT(25),XMEAN(25),ULO(25),
3UHI(25),LOCATP(25),NCAP(100),
4TTIME(100),ISHORT(200,100),DIST(100,100),IPATH(100),
5NPATH(100),CAP(100),NVST(25),IBLOCK(25),MMST(100),TSTOP(100),
6NRECIR,DISTLD,TMLOAD(100),CC(100),TOTAL,TOTPR
7,XLO(25),XMODE(25),XMN(25),STD(25),ARINT,AARRNT(25),NMODE(25),
8XXMEAN(25),UULO(25),XXLO(25),XXHI(25),XXMODE(25),XXMN(25)
9,SSTD(25),MODE(25),XHI(25),UUHI(25)
DIMENSION INCID(100,3),NWORK(100,2),TT(100),WW(100),
*PP(100),PL(100),DWORK(100)
INTEGER TT,WW,PP
DATA WW/100*0/,TT/100*0/,PP/100*0/,PL/100*0./,NWORK/200*0/,DWORK/1
100*0./
MNODE=2*NNODE
DO 20 I=1,NNODE
DO 10 J=6,8
K=J-5
INCID(I,K)=LINKN(I,J)
10 CONTINUE
IF(INCID(I,1).GT.0)WW(I)=WW(I)+1
IF(INCID(I,2).GT.0)WW(I)=WW(I)+1
IF(INCID(I,3).GT.0)WW(I)=WW(I)+1
20 CONTINUE
C FIND THE SHORTEST PATH TO ALL THE NODES FROM NODE K
ITOTAL=3*NNODE
K=1
C REMOVE NODE K FROM MATRIX WW
M=1
WW(K)=0
TT(M)=K
PP(M)=0
PL(M)=0
C FIND NODES ADJACENT TO THE NODES THAT CURRENTLY HAS PERMANENT
C LABEL THAT ARE IN MATRIX WW
25 IC=TT(M)
DO 30 MX=1,3
NN=INCID(IC,MX)
C CHECK IF NODE NN EXISTS
IF(NN.EQ.0)GO TO 30
C CALCULATE DISTANCE,SET POINTER
C IC=PRECEEDING NODE; NN=SUCCEEDING NODE.
IPOINT=3*(IC-1)+MX
NWORK(IPOINT,1)=IC
NWORK(IPOINT,2)=NN
C DISTANCE CALCULATION
CALL FINDD(IC,NN)
DWORK(IPOINT)=DIST(IC,NN)+PL(M)
30 CONTINUE

```

```

C      PICK THE NODE CLOSEST FROM NODE K
      IF(M.GE.MNODE)GO TO 60
      AMIN=100000000.
      DO 40 KF=1,ITOTAL
      IF(DWORK(KF).LE.AMIN.AND.DWORK(KF).GT.0.)ISUCC=NWORK(KF,2)
      IF(DWORK(KF).LE.AMIN.AND.DWORK(KF).GT.0.)IPREC=NWORK(KF,1)
      IF(DWORK(KF).LE.AMIN.AND.DWORK(KF).GT.0.)AMIN=DWORK(KF)
40    CONTINUE
C      ASSIGN PERMANENT LABEL TO THE NODE NEAREST FROM K FROM THE SET OF NODES
C      WHOSE DISTANCES ARE COMPUTED AND ARE IN MATRIX WW
      M=M+1
      TT(M)=ISUCC
      PP(M)=IPREC
      PL(M)=AMIN
C      DELETE ALL OTHER PATHS LEADING INTO NODE 'ISUCC' THAT ARE NOT THE
C      SHORTEST PATHS FROM NODE K
      WW(ISUCC)=0
      DO 50 KF=1,ITOTAL
      IF(NWORK(KF,2).NE.ISUCC)GO TO 50
      NWORK(KF,1)=0
      NWORK(KF,2)=0
      DWORK(KF)=0.
50    CONTINUE
C      NOW FIND THE NODES INCIDENT FROM 'ISUCC' THAT ARE IN MATRIX WW
      GO TO 25
C      MINIMUM PATHS HAVE BEEN FOUND FOR ALL NODES FROM NODE K
C      PLACE PATH AND DISTANCE IN THE SHORTEST PATH MATRIX
60    LT=(2*K) - 1
      LP=2*K
      DO 70 IX=1,M
      ISHORT(LT,IX)=TT(IX)
      LINE=TT(IX)
      TT(IX)=0
      ISHORT(LP,IX)=PP(IX)
      PP(IX)=0
      PL(IX)=0.
70    WW(IX)=IX
C      SHORTEST PATH FROM ALL NODES TO ALL OTHER NODES FOUND?
      K=K+1
      IF(K.GT.NNODE)GO TO 80
C      NO. SHORTEST PATH NOT YET FOUND FOR ALL NODES.
C      FIND THE SHORTEST PATH FROM NODE K. RESET M TO 1
      M=1
      TT(M)=K
      WW(K)=0
      GO TO 25
C      ALL SHORTEST PATHS ARE FOUND. PRINT OUT THE STRING OF TRACEABLE PATHS.
80    RETURN
      END

```

```

SUBROUTINE FINDD(IC,NN)
C   FIND THE DISTANT BETWEEN TWO ADJACENT NODES.
COMMON/SCOM1/ATRIB(100),DD(100),DDL(100),DTNOW,II,MFA,MSTOP,
$NCLNR,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100),TNEXT,TNOW,
$XX(100)
COMMON/UCOM1/NCTYPE,NNODE,LINKN(100,8),NARC,ASIZE(100),
1LOCATA(100,2),SPEED,USIZE,NSAREA,NSERVER(25),LOCATS(25,11),
2NPART,NVISIT,NROUTE(25,25),NEPT,ARRNT(25),XMEAN(25),ULO(25),
3UHI(25),LOCATP(25),NCAP(100),
4TTIME(100),ISHORT(200,100),DIST(100,100),IPATH(100),
5NPATH(100),CAP(100),NVST(25),IBLOCK(25),MMST(100),TSTOP(100),
6NRECIR,DISTLD,TMLOAD(100),CC(100),TOTAL,TOTPR
7,XLO(25),XMODE(25),XMN(25),STD(25),ARINT,AARRNT(25),NMODE(25),
8XXMEAN(25),UULO(25),XXLO(25),XXHI(25),XXMODE(25),XXMN(25)
9,SSTD(25),MODE(25),XHI(25),UUHI(25)
DO 10 I=1,NARC
C   FIRST FIND OUT THE ARC NUMBER BETWEEN THE TWO NODES
NSUC=LOCATA(I,1)
IF(NSUC.NE.IC) GO TO 10
IF(LOCATA(I,2).NE.NN) GO TO 10
C   ARC NUMBER FOUND.
NRARC=I
GO TO 15
10 CONTINUE
C   LOOK FOR THE DISTANCE IN THE ARC SIZE MATRIX
15 DIST(IC,NN)=ASIZE(NRARC)
RETURN
END

```

```

SUBROUTINE FINDA(IC,NN,NNARC)
C   FIND OUT THE ARC NUMBER WHICH LINKS NODES IC AND NN
COMMON/SCOM1/ATTRIB(100),DD(100),DDL(100),DTNOW,II,MFA,MSTOP,
$NCLNR,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100),TNEXT,TNOW,
$XX(100)
COMMON/UCOM1/NCTYPE,NNODE,LINKN(100,8),NARC,ASIZE(100),
1LOCATA(100,2),SPEED,USIZE,NSAREA,NSERV(25),LOCATS(25,11),
2NPART,NVISIT,NROUTE(25,25),NEPT,ARRNT(25),XMEAN(25),ULO(25),
3UHI(25),LOCATP(25),NCAP(100),
4TTIME(100),ISHORT(200,100),DIST(100,100),IPATH(100),
5NPATH(100),CAP(100),NVST(25),IBLOCK(25),MMST(100),TSTOP(100),
6NRECIR,DISTLD,TMLOAD(100),CC(100),TOTAL,TOTPR
7,XLO(25),XMODE(25),XMN(25),STD(25),ARINT,AARRNT(25),NMODE(25),
8XXMEAN(25),UULO(25),XXLO(25),XXHI(25),XXMODE(25),XXMN(25)
9,SSTD(25),MODE(25),XHI(25),UUHI(25)
C   SEE IF THE EMINATING NODE CORRESPONDS TO IC
DO 10 I=1,NARC
  NSUC=LOCATA(I,1)
  IF(NSUC.NE.IC) GO TO 10
C   CHECK FOR THE TERMINATING NODE
  IF(LOCATA(I,2).NE.NN) GO TO 10
C   ARC NUMBER IS I
  NNARC=I
  GO TO 15
10 CONTINUE
15 RETURN
END

```

```

SUBROUTINE FINDL(NODFR)
C   FIND THE LAST NODE A PART HAS VISITED
COMMON/SCOM1/ATTRIB(100),DD(100),DDL(100),DTNOW,II,MFA,MSTOP,
$NCLNR,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100),TNEXT,TNOW,
$XX(100)
COMMON/UCOM1/NCTYPE,NNODE,LINKN(100,8),NARC,ASIZE(100),
1LOCATA(100,2),SPEED,USIZE,NSAREA,NSERV(25),LOCATS(25,11),
2NPART,NVISIT,NROUTE(25,25),NEPT,ARRNT(25),XMEAN(25),ULO(25),
3UHI(25),LOCATP(25),NCAP(100),
4TTIME(100),ISHORT(200,100),DIST(100,100),IPATH(100),
5NPAT(100),CAP(100),NVST(25),IBLOCK(25),MMST(100),TSTOP(100),
6NRECIR,DISTLD,TMLOAD(100),CC(100),TOTAL,TOTPR
7,XLO(25),XMODE(25),XMN(25),STD(25),ARINT,AARRNT(25),NMODE(25),
8XXMEAN(25),UULO(25),XXLO(25),XXHI(25),XXMODE(25),XXMN(25)
9,SSTD(25),MODE(25),XHI(25),UUHI(25)
C   FIND THE LAST ARC VISITED AND THE CURRENT NODE
LASTAR=ATTRIB(9)+0.5
LASTND=ATTRIB(11)+0.5
C   FIND THE NODE FROM WHICH THE LAST ARC VISITED EMINATES
C   AND LEADS TO THE CURRENT NODE
DO 10 I=1,NARC
IF(I.NE.LASTAR) GO TO 10
IF(LOCATA(I,2).EQ.LASTND) GO TO 20
10 CONTINUE
CALL ERROR
RETURN
C   REQUIREQ NODE FOUND
20 NODFR=LOCATA(1,1)
RETURN
END

```

```

SUBROUTINE FINDS(NODAT,NSTN,EX)
C   DETERMINE IF A NODE (NODAT) IS A SERVICE STATION (NSTN)?
COMMON/SCOM1/ATTRIB(100),DD(100),DDL(100),DTNOW,II,MFA,MSTOP,
$NCLNR,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100),TNEXT,TNOW,
$XX(100)
COMMON/UCOM1/NCTYPE,NNODE,LINKN(100,8),NARC,ASIZE(100),
1LOCATA(100,2),SPEED,USIZE,NSAREA,NSERV(25),LOCATS(25,11),
2NPART,NVISIT,NROUTE(25,25),NEPT,ARRNT(25),XMEAN(25),ULO(25),
3UHI(25),LOCATP(25),NCAP(100),
4TTIME(100),ISHORT(200,100),DIST(100,100),IPATH(100),
5NPAT(100),CAP(100),NVST(25),IBLOCK(25),MMST(100),TSTOP(100),
6NRECIR,DISTLD,TMLOAD(100),CC(100),TOTAL,TOTPR
7,XLO(25),XMODE(25),XMN(25),STD(25),ARINT,AARRNT(25),NMODE(25),
8XXMEAN(25),UULO(25),XXLO(25),XXHI(25),XXMODE(25),XXMN(25)
9,SSTD(25),MODE(25),XHI(25),UUHI(25)
DO 20 I=1,NSAREA
K=1
J=LOCATS(I,1)
DO 10 JJ=1,J
K=K+1
IF(LOCATS(I,K).EQ.NODAT) GO TO 30
C   NOT A SERVICE STATION
EX=0.
10 CONTINUE
20 CONTINUE
RETURN
C   SERVICE STATION
30 EX=1.
NSTN=I
RETURN
END

```

```

SUBROUTINE FINDS2(NTHIS,NEXT)
C  DETERMINE THE NODE NUMBERS FOR NODES WHICH CORRESPOND TO
C  THE SERVICE STATIONS (NTHIS,NEXT).
COMMON/SCOM1/ATTRIB(100),DD(100),DDL(100),DTNOW,II,MFA,MSTOP,
$NCLNR,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100),TNEXT,TNOW,
$XX(100)
COMMON/UCOM1/NCTYPE,NNODE,LINKN(100,8),NARC,ASIZE(100),
1LOCATA(100,2),SPEED,USIZE,NSAREA,NSERV(25),LOCATS(25,11),
2NPART,NVISIT,NROUTE(25,25),NEPT,ARRNT(25),XMEAN(25),ULO(25),
3UHI(25),LOCATP(25),NCAP(100),
4TTIME(100),ISHORT(200,100),DIST(100,100),IPATH(100),
5NPATH(100),CAP(100),NVST(25),IBLOCK(25),MMST(100),TSTOP(100),
6NRECIR,DISTLD,TMLOAD(100),CC(100),TOTAL,TOTPR
7,XLO(25),XMODE(25),XMN(25),STD(25),ARINT,AARRNT(25),NMODE(25),
8XXMEAN(25),UULO(25),XXLO(25),XXHI(25),XXMODE(25),XXMN(25)
9,SSTD(25),MODE(25),XHI(25),UUHI(25)
J=LOCATS(NTHIS,1)
IF(J.EQ.1) GO TO 20
10 CALL ERROR
RETURN
20 NTHIS=LOCATS(NTHIS,2)
K=LOCATS(NEXT,1)
IF(K.NE.1) GO TO 10
NEXT=LOCATS(NEXT,2)
RETURN
END

```



```
      SUBROUTINE EVENT(I)
C     DIRECTS TO THE PROPER EVENT. SIX EVENTS ARE
C     ARRIVAL TO CONVEYOR
C     END OF CONVEYOR
C     END OF SERVICE
C     ARRIVAL OF A LOADING POINT (ONLY FOR CLASS 3 TYPE CONVEYORS).
C     CLEAR STATISTICS
C     PRINT SUMMARY
      GO TO (1,2,3,4,5,6),I
1     CALL ARRIV(I)
      RETURN
2     CALL CONVR(I)
      RETURN
3     CALL ENDSV(I)
      RETURN
4     CALL LOAD(I)
      RETURN
5     CALL CLRST
      RETURN
6     CALL SMMRY
      RETURN
      END
```

```

SUBROUTINE ARRIV(I)
C   EVENT 1: ARRIVAL TO A CONVEYOR. DIRECT TO THE PROPER ROUTINE
COMMON/SCOM1/ATRIB(100),DD(100),DDL(100),DTNOW,II,MFA,MSTOP,
$NCLNR,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100),TNEXT,TNOW,
$XX(100)
COMMON/UCOM1/NCTYPE,NNODE,LINKN(100,8),NARC,ASIZE(100),
1LOCATA(100,2),SPEED,USIZE,NSAREA,NSERV(25),LOCATS(25,11),
2NPART,NVISIT,NROUTE(25,25),NEPT,ARRNT(25),XMEAN(25),ULO(25),
3UHI(25),LOCATP(25),NCAP(100),
4TTIME(100),ISHORT(200,100),DIST(100,100),IPATH(100),
5NPATH(100),CAP(100),NVST(25),IBLOCK(25),MMST(100),TSTOP(100),
6NRECIR,DISTLD,TMLOAD(100),CC(100),TOTAL,TOTPR
7,XLO(25),XMODE(25),XMN(25),STD(25),ARINT,AARRNT(25),NMODE(25),
8XXMEAN(25),UULO(25),XXLO(25),XXHI(25),XXMODE(25),XXMN(25)
9,SSTD(25),MODE(25),XHI(25),UUHI(25)
GO TO (1,2,3),NCTYPE
C   CONVEYOR CLASS 1
C   IS THIS AN OLD PART ?
1  IF(ATRIB(7).GT.0.) CALL OLD1
C   IS THIS A NEW PART ?
  IF(ATRIB(7).EQ.0.) CALL NEW1
  RETURN
C   CONVEYOR CLASS 2
C   IS THIS AN OLD PART ?
2  IF(ATRIB(7).GT.0.) CALL OLD2
C   IS THIS A NEW PART ?
  IF(ATRIB(7).EQ.0.) CALL NEW2
  RETURN
C   CONVEYOR CLASS 3
C   IS THIS AN OLD PART ?
3  IF(ATRIB(7).GT.0.) CALL OLD3
C   IS THIS A NEW PART ?
  IF(ATRIB(7).EQ.0.) CALL NEW3
  RETURN
END

```

```

SUBROUTINE CONVR( I )
C   EVENT 2: END OF A CONVEYOR. DIRECT TO THE PROPER ROUTINE.
COMMON/SCOM1/ATTRIB(100),DD(100),DDL(100),DTNOW,II,MFA,MSTOP,
$NCLNR,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100),TNEXT,TNOW,
$XX(100)
COMMON/UCOM1/NCTYPE,NNODE,LINKN(100,8),NARC,ASIZE(100),
1LOCATA(100,2),SPEED,USIZE,NSAREA,NSERV(25),LOCATS(25,11),
2NPART,NVISIT,NROUTE(25,25),NEPT,ARRNT(25),XMEAN(25),ULO(25),
3UHI(25),LOCATP(25),NCAP(100),
4TTIME(100),ISHORT(200,100),DIST(100,100),IPATH(100),
5NPATH(100),CAP(100),NVST(25),IBLOCK(25),MMST(100),TSTOP(100),
6NRECIR,DISTLD,TMLOAD(100),CC(100),TOTAL,TOTPRT
7,XLO(25),XMODE(25),XMN(25),STD(25),ARINT,AARRNT(25),NMODE(25),
8XXMEAN(25),UULO(25),XXLO(25),XXHI(25),XXMODE(25),XXMN(25)
9,SSTD(25),MODE(25),XHI(25),UUHI(25)
GO TO (1,2,3),NCTYPE
C   CONVEYOR CLASS 1
1 CALL CONVR1
RETURN
C   CONVEYOR CLASS 2
2 CALL CONVR2
RETURN
C   CONVEYOR CLASS 3
3 CALL CONVR3
RETURN
END

```

```

SUBROUTINE ENDSV(1)
C   EVENT 3: END OF SERVICE. DIRECT TO THE PROPER ROUTINE
COMMON/SCOM1/ATRI(100),DD(100),DDL(100),DTNOW,II,MFA,MSTOP,
$NCLNR,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100),TNEXT,TNOW,
$XX(100)
COMMON/UCOM1/NCTYPE,NNODE,LINKN(100,8),NARC,ASIZE(100),
1LOCATA(100,2),SPEED,USIZE,NSAREA,NSERV(25),LOCATS(25,11),
2NPART,NVISIT,NROUTE(25,25),NEPT,ARRNT(25),XMEAN(25),ULO(25),
3UHI(25),LOCATP(25),NCAP(100),
4TTIME(100),ISHORT(200,100),DIST(100,100),IPATH(100),
5NPATH(100),CAP(100),NVST(25),IBLOCK(25),MMST(100),TSTOP(100),
6NRECIR,DISTLD,TMLOAD(100),CC(100),TOTAL,TOTPR
7,XLO(25),XMODE(25),XMN(25),STD(25),ARINT,AARRNT(25),NMODE(25),
8XXMEAN(25),UULO(25),XXLO(25),XXHI(25),XXMODE(25),XXMN(25)
9,SSTD(25),MODE(25),XHI(25),UUHI(25)
GO TO (1,2,3),NCTYPE
C   CONVEYOR CLASS 1
1 CALL ENDSV1
RETURN
C   CONVEYOR CLASS 2
2 CALL ENDSV2
RETURN
C   CONVEYOR CLASS 3
3 CALL ENDSV3
RETURN
END

```

```

SUBROUTINE NEW1
C   A PART HAS JUST ARRIVED IN THE SYSTEM. THIS ROUTINE ASSIGNS
C   ROUTE, ATTRIBUTE VALUES, ETC.
COMMON/SCOM1/ATTRIB(100),DD(100),DDL(100),DTNOW,II,MFA,MSTOP,
$NCLNR,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100),TNEXT,TNOW,
$XX(100)
COMMON/UCOM1/NCTYPE,NNODE,LINKN(100,8),NARC,ASIZE(100),
1LOCATA(100,2),SPEED,USIZE,NSAREA,NSVR(25),LOCATS(25,11),
2NPART,NVISIT,NROUTE(25,25),NEPT,ARRNT(25),XMEAN(25),ULO(25),
3UHI(25),LOCATP(25),NCAP(100),
4TTIME(100),ISHORT(200,100),D!ST(100,100),IPATH(100),
5NPATH(100),CAP(100),NVST(25),IBLOCK(25),MMST(100),TSTOP(100),
6NRECIR,DISTLD,TMLOAD(100),CC(100),TOTAL,TOTPR
7,XLO(25),XMODE(25),XMN(25),STD(25),ARINT,AARRNT(25),NMODE(25),
8XXMEAN(25),UULO(25),XXLO(25),XXHI(25),XXMODE(25),XXMN(25)
9,SSTD(25),MODE(25),XHI(25),UUHI(25)
COMMON/UCOM2/SAMP(100,5),TSAMP(100,6),TIMCLR,TIMSTP
COMMON/UCOM3/MAXPRT(25)
IP=ATTRIB(4)+0.5
C   CHECK FOR THE MAXIMUM NUMBER OF PARTS IN THE SYSTEM
IF(MAXPRT(IP).EQ.0) RETURN
ATTRIB(4)=IP
C   SCHEDULE NEXT ARRIVAL
CALL ARIVAL(IP,ARR)
CALL SCHDL(1,ARR,ATTRIB)
MAXPRT(IP)=MAXPRT(IP)+1
C   ASSIGN ATTRIBUTE VALUES
ATTRIB(1)=TNOW
ATTRIB(2)=NROUTE(IP,1)
ATTRIB(3)=NROUTE(IP,2)
C   FIND THE SHORTEST PATH
NTHIS=ATTRIB(2)+0.5
NEXT=ATTRIB(3)+0.5
CALL FINDS2(NTHIS,NEXT)
CALL PATH(NTHIS,NEXT,NODPTH)
ATTRIB(4)=IP
ATTRIB(5)=1.
ATTRIB(6)=1.
ATTRIB(7)=NODPTH
ATTRIB(8)=TNOW
ATTRIB(9)=0.
K=11
DO 10 I=1,NODPTH
ATTRIB(K)=NPATHT(I)
K=K+1
10 CONTINUE
NODAT=ATTRIB(11)+0.5
NODTO=ATTRIB(12)+0.5
C   FIND THE NEXT ARC
CALL FINDA(NODAT,NODTO,NNARC)
ATTRIB(10)=NNARC
C   IS THERE ANY SPACE ON THE CONVEYOR ?
IF(CC(NNARC).GE.CAP(NNARC)) GO TO 20
C   YES, INCREMENT NUMBER ON ARC & UPDATE STATS. SCHEDULE NEXT EVENT
CC(NNARC)=CC(NNARC)+1.
CALL MMSTT(0,TNOW,CC(NNARC),NNARC)
CALL SCHDL(2,TTIME(NNARC),ATTRIB)
RETURN
C   NO, PLACE IT IN A QUEUE
20 CALL FILEM(1,ATTRIB)
RETURN
END

```

```

SUBROUTINE NEW2
C   A PART HAS JUST ARRIVED IN THE SYSTEM. THIS ROUTINE ASSIGNS
C   ROUTE, ATTRIBUTE VALUES, ETC. USED INSTEAD OF NEW1 BECAUSE OF
C   CLASS 2 CONVEYOR
COMMON/SCOM1/ATTRIB(100),DD(100),DDL(100),DTNOW,II,MFA,MSTOP,
$NCLNR,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100),TNEXT,TNOW,
$XX(100)
COMMON/UCOM1/NCTYPE,NNODE,LINKN(100,8),NARC,ASIZE(100),
1LOCATA(100,2),SPEED,USIZE,NSAREA,NSERV(25),LOCATS(25,11),
2NPART,NVISIT,NROUTE(25,25),NEPT,ARRNT(25),XMEAN(25),ULO(25),
3UHI(25),LOCATP(25),NCAP(100),
4TIME(100),ISHORT(200,100),DIST(100,100),IPATH(100),
5NPATH(100),CAP(100),NVST(25),IBLOCK(25),MMST(100),TSTOP(100),
6NRECIR,DISTLD,TMLOAD(100),CC(100),TOTAL,TOTPR
7,XLO(25),XMODE(25),XMN(25),STD(25),ARINT,AARRNT(25),NMODE(25),
8XXMEAN(25),UULO(25),XXLO(25),XXHI(25),XXMODE(25),XXMN(25)
9,SSTD(25),MODE(25),XHI(25),UUHI(25)
COMMON/UCOM2/SAMP(100,5),TSAMP(100,6),TIMCLR,TIMSTP
COMMON/UCOM3/MAXPRT(25)
IP=ATTRIB(4)+0.5
C   CHECK FOR THE MAXIMUM NUMBER OF PARTS IN THE SYSTEM
IF(MAXPRT(IP).EQ.0) RETURN
ATTRIB(4)=IP
C   SCHEDULE NEXT ARRIVAL
CALL ARIVAL(IP,ARR)
CALL SCHDL(1,ARR,ATTRIB)
MAXPRT(IP)=MAXPRT(IP)-1
C   ASSIGN ATTRIBUTE VALUES
ATTRIB(1)=TNOW
ATTRIB(2)=NROUTE(IP,1)
ATTRIB(3)=NROUTE(IP,2)
C   FIND THE SHORTEST PATH
NTHIS=ATTRIB(2)+0.5
NEXT=ATTRIB(3)+0.5
CALL FINDS2(NTHIS,NEXT)
CALL PATH(NTHIS,NEXT,NODPTH)
ATTRIB(4)=IP
ATTRIB(5)=1.
ATTRIB(6)=1.
ATTRIB(7)=NODPTH
ATTRIB(8)=TNOW
ATTRIB(9)=0.
K=11
DO 10 I=1,NODPTH
ATTRIB(K)=NPATHT(I)
K=K+1
10 CONTINUE
NODAT=ATTRIB(11)+0.5
NODTO=ATTRIB(12)+0.5
C   FIND THE NEXT ARC
CALL FINDA(NODAT,NODTO,NNARC)
ATTRIB(10)=NNARC
C   IS CONVEYOR RUNNING ?
IF(MMST(NNARC).GT.0) GO TO 20
C   IS THERE ANY SPACE ON THE CONVEYOR ?
IF(CC(NNARC).GE.CAP(NNARC)) GO TO 20
C   YES, INCREMENT NUMBER ON ARC & UPDATE STATS. SCHEDULE NEXT EVENT
CC(NNARC)=CC(NNARC)+1.
CALL MMSTT(0,TNOW,CC(NNARC),NNARC)
C   IS THERE ANOTHER PART AHEAD ON THE CONVEYOR
IF(CC(NNARC).GT.1.) GO TO 30
CALL SCHDL(2,TIME(NNARC),ATTRIB)
RETURN
C   NO, PLACE IT IN A QUEUE
20 CALL FILEM(1,ATTRIB)
RETURN
C   CONVEYOR WILL STOP BECAUSE THERE IS A PART IN FRONT
30 CALL FILEM(2,ATTRIB)
RETURN
END

```

```

SUBROUTINE NEW3
C   A PART HAS JUST ARRIVED IN THE SYSTEM. THIS ROUTINE ASSIGNS
C   ROUTE, ATTRIBUTE VALUES, ETC. USED INSTEAD OF NEW1 OR NEW2 BECAUSE
C   CLASS 3 CONVEYOR
COMMON/SCOM1/ATLIB(100),DD(100),DDL(100),DTNOW,II,MFA,MSTOP,
$NCLNR,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100),TNEXT,TNOW,
$XX(100)
COMMON/UCOM1/NCTYPE,NNODE,LINKN(100,8),NARC,ASIZE(100),
1LOCATA(100,2),SPEED,USIZE,NSAREA,NSERV(25),LOCATS(25,11),
2NPART,NVISIT,NROUTE(25,25),NEPT,ARRNT(25),XMEAN(25),ULO(25),
3UHI(25),LOCATP(25),NCAP(100),
4TTIME(100),ISHORT(200,100),DIST(100,100),IPATH(100),
5NPATH(100),CAP(100),NVST(100),IBLOCK(25),MMST(100),TSTOP(100),
6NRECIR,DISTLD,TMLOAD(100),CC(100),TOTAL,TOTPR
7,XLO(25),XMODE(25),XMN(25),STD(25),ARINT,AARRNT(25),NMODE(25),
8XXMEAN(25),UULO(25),XXLO(25),XXHI(25),XXMODE(25),XXMN(25)
9,SSTD(25),MODE(25),XHI(25),UUHI(25)
COMMON/UCOM2/SAMP(100,5),TSAMP(100,6),TIMCLR,TIMSTP
COMMON/UCOM3/MAXPRT(25)
IP=ATLIB(4)+0.5
C   CHECK FOR THE MAXIMUM NUMBER OF PARTS IN THE SYSTEM
IF(MAXPRT(IP).EQ.0) RETURN
ATLIB(4)=IP
C   SCHEDULE NEXT ARRIVAL
CALL ARIVAL(IP,ARR)
CALL SCHDL(1,ARR,ATLIB)
MAXPRT(IP)=MAXPRT(IP)+1
C   ASSIGN ATTRIBUTE VALUES
ATLIB(1)=TNOW
ATLIB(2)=NROUTE(IP,1)
ATLIB(3)=NROUTE(IP,2)
C   FIND THE SHORTEST PATH
NTHIS=ATLIB(2)+0.5
NEXT=ATLIB(3)+0.5
CALL FINDS2(NTHIS,NEXT)
CALL PATH(NTHIS,NEXT,NODPTH)
ATLIB(4)=IP
ATLIB(5)=1.
ATLIB(6)=1.
ATLIB(7)=NODPTH
ATLIB(9)=0.
K=11
DO 10 I=1,NODPTH
ATLIB(K)=NPATHT(I)
K=K+1
10 CONTINUE
NODAT=ATLIB(11)+0.5
NODTO=ATLIB(12)+0.5
C   FIND THE NEXT ARC
CALL FINDA(NODAT,NODTO,NNARC)
ATLIB(10)=NNARC
C   WAIT FOR THE LOADING POINT
CALL FILEM(1,ATLIB)
RETURN
END

```

```

SUBROUTINE OLD1
C   AN OLD PART HAS ARRIVED AT A NODE & NEEDS TO GO TO THE NEXT ARC
C   ON THE ROUTE.
COMMON/SCOM1/ATRIB(100),DD(100),DDL(100),DTNOW,II,MFA,MSTOP,
$NCLNR,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100),TNEXT,TNOW,
$XX(100)
COMMON/UCOM1/NCTYPE,NNODE,LINKN(100,8),NARC,ASIZE(100),
1LOCATA(100,2),SPEED,USIZE,NSAREA,NSERV(25),LOCATS(25,11),
2NPART,NVISIT,NROUTE(25,25),NEPT,ARRNT(25),XMEAN(25),ULO(25),
3UHI(25),LOCATP(25),NCAP(100),
4TTIME(100),ISHORT(200,100),DIST(100,100),IPATH(100),
5NPATH(100),CAP(100),NVST(25),IBLOCK(25),MMST(100),TSTOP(100),
6NRECIR,DISTLD,TMLOAD(100),CC(100),TOTAL,TOTPR
7,XLO(25),XMODE(25),XMN(25),STD(25),ARINT,AARRNT(25),NMODE(25),
8XXMEAN(25),UULO(25),XXLO(25),XXHI(25),XXMODE(25),XXMN(25)
9,SSTD(25),MODE(25),XHI(25),UUHI(25)
COMMON/UCOM2/SAMP(100,5),TSAMP(100,6),TIMCLR,TIMSTP
COMMON/UCOM3/MAXPRT(25)
C   MODIFY ATTRIBUTES
NTOT=ATRIB(5)+0.5
NODAT=ATRIB(12)+0.5
ATRIB(11)=ATRIB(12)
C   IS THIS NODE A STATION OR NOT ?
CALL FINDS(NODAT,NSTN,EX)
NFL=NSTN+3
J=NTOT+11
ATRIB(12)=ATRIB(J)
NODAT=ATRIB(11)+0.5
NODTO=ATRIB(12)+0.5
C   NEXT ARC
CALL FINDA(NODAT,NODTO,NNARC)
K=ATRIB(9)+0.5
ATRIB(10)=NNARC
C   IS THERE ANOTHER PART AHEAD OF THIS ONE ?
IF(EX.EQ.1..AND.NNQ(NFL).GT.0) GO TO 10
C   NO, IS THERE ANY PLACE ON THE NEXT CONVEYOR
IF(CC(NNARC).GE.CAP(NNARC)) GO TO 10
C   YES, UPDATE SYSTEM VARIABLES & STATS. SCHEDULE NEXT EVENT
CC(K)=CC(K)-1.
TCON=TNOW-ATRIB(8)
CALL CCLCT(0,TCON,K)
CALL MMSTT(0,TNOW,CC(K),K)
CC(NNARC)=CC(NNARC)+1.
ATRIB(8)=TNOW
CALL MMSTT(0,TNOW,CC(NNARC),NNARC)
CALL SCHDL(2,TTIME(NNARC),ATRIB)
C   MOVE ALL THE PARTS FOLLOWING
CALL ADVNC1
RETURN
C   IF THERE IS SOMEBODY AHEAD OR THE NEXT CONVEYOR IS FULL, PLACE
C   PART IN A QUEUE. IF CURRENT NODE IS A STATION, USE STATION QUEUE
C   OTHERWISE, USE FILE 1
C   STOP THE CONVEYOR & NOTE THE TIME
10 IF(EX.EQ.1.) CALL FILEM(NFL,ATRIB)
IF(EX.EQ.0.) CALL FILEM(1,ATRIB)
RETURN
END

```



```

SUBROUTINE OLD2
C   AN OLD PART HAS ARRIVED AT A NODE & NEEDS TO GO TO THE NEXT ARC
C   ON THE ROUTE. USE INSTEAD OF OLD1. CLASS 2 CONVEYOR
COMMON/SCOM1/ATLIB(100),DD(100),DDL(100),DTNOW,II,MFA,MSTOP,
$NCLNR,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100),TNEXT,TNOW,
$XX(100)
COMMON/UCOM1/NCTYPE,NNODE,LINKN(100,8),NARC,ASIZE(100),
1LOCATA(100,2),SPEED,USIZE,NSAREA,NSERV(25),LOCATS(25,11),
2NPART,NVISIT,NROUTE(25,25),NEPT,ARRNT(25),XMEAN(25),ULO(25),
3UHI(25),LOCATP(25),NCAP(100),
4TTIME(100),ISHORT(200,100),DIST(100,100),IPATH(100),
5NPATH(100),CAP(100),NVST(25),IBLOCK(25),MMST(100),TSTOP(100),
6NRECIR,DISTLD,TMLOAD(100),CC(100),TOTAL,TOTPR
7,XLO(25),XMODE(25),XMN(25),STD(25),ARINT,AARRNT(25),NMODE(25),
8XXMEAN(25),UULO(25),XXLO(25),XXHI(25),XXMODE(25),XXMN(25)
9,SSTD(25),MODE(25),XHI(25),UUHI(25)
COMMON/UCOM2/SAMP(100,5),TSAMP(100,6),TIMCLR,TIMSTP
COMMON/UCOM3/MAXPRT(25)
C   MODIFY ATTRIBUTES
NTOT=ATLIB(5)+0.5
NODAT=ATLIB(12)+0.5
ATLIB(11)=ATLIB(12)
C   IS THIS NODE A STATION OR NOT ?
CALL FINDS(NODAT,NSTN,EX)
NFL=NSTN+3
J=NTOT+11
ATLIB(12)=ATLIB(J)
NODAT=ATLIB(11)+0.5
NODTO=ATLIB(12)+0.5
C   NEXT ARC
CALL FINDA(NODAT,NODTO,NNARC)
K=ATLIB(9)+0.5
ATLIB(10)=NNARC
C   IS THERE ANOTHER PART AHEAD OF THIS ONE ?
IF(EX.EQ.1..AND.NNQ(NFL).GT.0) GO TO 10
C   IS THE CONVEYOR RUNNING ?
IF(MMST(NNARC).GT.0) GO TO 10
C   NO, IS THERE ANY PLACE ON THE NEXT CONVEYOR
IF(CC(NNARC).GE.CAP(NNARC)) GO TO 10
C   YES, UPDATE SYSTEM VARIABLES & STATS. SCHEDULE NEXT EVENT
CC(K)=CC(K)-1.
TCON=TNOW-ATLIB(8)
CALL CCLCT(0,TCON,K)
CALL MMSTT(0,TNOW,CC(K),K)
CC(NNARC)=CC(NNARC)+1.
ATLIB(8)=TNOW
CALL MMSTT(0,TNOW,CC(NNARC),NNARC)
C   CHECK IF A PART IS AHEAD
IF(CC(NNARC).GT.1.) GO TO 20
CALL SCHDL(2,TTIME(NNARC),ATLIB)
C   MOVE ALL THE PARTS FOLLOWING
CALL ADVNC2
RETURN
C   IF THE NEXT CONVEYOR IS NOT RUNNING OR
C   IF THERE IS SOMEBODY AHEAD OR THE NEXT CONVEYOR IS FULL, PLACE
C   PART IN A QUEUE. IF CURRENT NODE IS A STATION, USE STATION QUEUE
C   OTHERWISE, USE FILE 1
C   STOP THE CONVEYOR & NOTE THE TIME
10 MMST(K)=1
TSTOP(K)=TNOW
IF(EX.EQ.1.) CALL FILEM(NFL,ATLIB)
IF(EX.EQ.0.) CALL FILEM(1,ATLIB)
RETURN
C   THERE IS A PART AHEAD WHICH WILL STOP THE CONVEYOR
20 CALL FILEM(2,ATLIB)
CALL ADVNC2
RETURN
END

```

```

SUBROUTINE OLD3
C   AN OLD PART HAS ARRIVED AT A NODE & NEEDS TO GO TO THE NEXT ARC
C   ON THE ROUTE. USED INSTEAD OF OLD1 OR OLD2. CLASS 3 CONVEYOR
COMMON/SCOM1/ATRI(100),DD(100),DDL(100),DTNOW,II,MFA,MSTOP,
$NCLNR,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100),TNEXT,TNOW,
$XX(100)
COMMON/UCOM1/NCTYPE,NNODE,LINKN(100,8),NARC,ASIZE(100),
1LOCATA(100,2),SPEED,USIZE,NSAREA,NSRVR(25),LOCATS(25,11),
2NPART,NVISIT,NROUTE(25,25),NEPT,ARRNT(25),XMEAN(25),ULO(25),
3UHI(25),LOCATP(25),NCAP(100),
4TTIME(100),ISHORT(200,100),DIST(100,100),IPATH(100),
5NPATH(100),CAP(100),NVST(100),IBLOCK(25),MMST(100),TSTOP(100),
6NRECIR,DISTLD,TMLOAD(100),CC(100),TOTAL,TOTPR
7,XLO(25),XMODE(25),XMN(25),STD(25),ARINT,AARRNT(25),NMODE(25),
8XXMEAN(25),UULO(25),XXLO(25),XXHI(25),XXMODE(25),XXMN(25)
9,SSTD(25),MODE(25),XHI(25),UUHI(25)
COMMON/UCOM2/SAMP(100,5),TSAMP(100,6),TIMCLR,TIMSTP
COMMON/UCOM3/MAXPRT(25)
C   MODIFY ATTRIBUTES
NTOT=ATRI(5)+0.5
NODAT=ATRI(12)+0.5
ATRI(11)=ATRI(12)
C   FIND OUT THE STATION NUMBER
CALL FINDS(NODAT,NSTN,EX)
NFL=NSTN+3
J=NTOT+11
ATRI(12)=ATRI(J)
NODAT=ATRI(11)+0.5
NODTO=ATRI(12)+0.5
K=ATRI(9)+0.5
C   NEXT ARC
CALL FINDA(NODAT,NODTO,NNARC)
ATRI(10)=NNARC
C   PART MUST WAIT FOR THE NEXT LOADING POINT, PLACE
C   PART IN A QUEUE. CURRENT NODE IS A STATION, USE STATION QUEUE
C   STOP THE CONVEYOR & NOTE THE TIME
MMST(K)=1
TSTOP(K)=TNOW
CALL FILEM(NFL,ATRI)
RETURN
END

```

```

SUBROUTINE CONVR1
C  A PART HAS REACHED END OF A CONVEYOR. DETERMINE WHERE DOES IT GO
C  NEXT ?
COMMON/SCOM1/ATRIB(100),DD(100),DDL(100),DTNOW,II,MFA,MSTOP,
SNCLNR,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100),TNEXT,TNOW,
SXX(100)
COMMON/UCOM1/NCTYPE,NNODE,LINKN(100,8),NARC,ASIZE(100),
1LOCATA(100,2),SPEED,USIZE,NSAREA,NSERV(25),LOCATS(25,11),
2NPART,NVISIT,NROUTE(25,25),NEPT,ARRNT(25),XMEAN(25),ULO(25),
3UHI(25),LOCATP(25),NCAP(100),
4TTIME(100),ISHORT(200,100),DIST(100,100),IPATH(100),
5NPATH(100),CAP(100),NVST(25),IBLOCK(25),MMST(100),TSTOP(100),
6NRECIR,DISTLD,TMLOAD(100),CC(100),TOTAL,TOTPR
7,XLO(25),XMODE(25),XMN(25),STD(25),ARINT,AARRNT(25),NMODE(25),
8XXMEAN(25),UULO(25),XXLO(25),XXHI(25),XXMODE(25),XXMN(25)
9,SSTD(25),MODE(25),XHI(25),UHI(25)
COMMON/UCOM2/SAMP(100,5),TSAMP(100,6),TIMCLR,TIMSTP
COMMON/UCOM3/MAXPRT(25)
ATTRIB(5)=ATTRIB(5)+1.
ATTRIB(9)=ATTRIB(10)
C  HAS IT REACHED THE STATION IT IS SUPPOSED TO GO TO ?
IF(ATTRIB(5).EQ.ATTRIB(7)) GO TO 10
C  NO, SEE IF IT CAN BE ADVANCED IN THE SYSTEM
CALL SCHDL(1,0.,ATTRIB)
RETURN
C  YES, THIS IS THE STATION
10 NSTN=ATTRIB(3)+0.5
J=NSTN+3
C  CHECK THE QUEUE OF THE STATION
IF(NNQ(J).GT.0) GO TO 30
C  CHECK FOR THE AVAILABILITY OF THE SERVERS.
IBUSY=IBLOCK(NSTN)+XX(NSTN)
IF(IBUSY.GE.NSERV(NSTN)) GO TO 20
C  SCHEDULE THE END OF SERVICE, ALSO UPDATE THE SYSTEM
N1=ATTRIB(9)+0.5
XX(NSTN)=XX(NSTN)+1.
CC(N1)=CC(N1)-1.
TCON=TNOW-ATTRIB(8)
CALL CCLCT(0,TCON,N1)
CALL MMSTT(0,TNOW,CC(N1),N1)
XN=N1
I=ATTRIB(3)+0.5
CALL SERVIC(I,SVT)
CALL SCHDL(3,SVT,ATTRIB)
C  ADVANCE ALL THE FOLLOWING PARTS
CALL ADVNC1
RETURN
C  THERE IS NO SERVER AVAILABLE
C  IS RECIRCULATION ALLOWED ?
20 IF(NRECIR.EQ.1) CALL RECIR1
C  NO, PLACE IT IN THE QUEUE
IF(NRECIR.EQ.0) CALL FILEM(J,ATTRIB)
RETURN
C  THERE IS ANOTHER PART AHEAD OF IT
30 CALL FILEM(J,ATTRIB)
RETURN
END

```

```

SUBROUTINE CONVR2
C   A PART HAS REACHED END OF A CONVEYOR. DETERMINE WHERE DOES IT GO
C   NEXT ? USED INSTEAD OF CONVR1----CLASS 2 CONVEYOR
COMMON/SCOM1/ATLIB(100),DD(100),DDL(100),DTNOW,II,MFA,MSTOP,
$NCLNR,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100),TNEXT,TNOW,
$XX(100)
COMMON/UCOM1/NCTYPE,NNODE,LINKN(100,8),NARC,ASIZE(100),
1LOCATA(100,2),SPEED,USIZE,NSAREA,NSRVR(25),LOCATS(25,11),
2NPART,NVISIT,NROUTE(25,25),NEPT,ARRNT(25),XMEAN(25),ULO(25),
3UHI(25),LOCATP(25),NCAP(100),
4TTIME(100),ISHORT(200,100),DIST(100,100),IPATH(100),
5NPATH(100),CAP(100),NVST(25),IBLOCK(25),MMST(100),TSTOP(100),
6NRECIR,DISTLD,TMLOAD(100),CC(100),TOTAL,TOTPR
7,XLO(25),XMODE(25),XMN(25),STD(25),ARINT,AARRNT(25),NMODE(25),
8XXMEAN(25),UULO(25),XXLO(25),XXHI(25),XXMODE(25),XXMN(25)
9,SSTD(25),MODE(25),XHI(25),UUHI(25)
COMMON/UCOM2/SAMP(100,5),TSAMP(100,6),TIMCLR,TIMSTP
COMMON/UCOM3/MAXPRT(25)
ATLIB(5)=ATLIB(5)+1.
ATLIB(9)=ATLIB(10)
C   HAS IT REACHED THE STATION IT IS SUPPOSED TO GO TO ?
C   IF(ATLIB(5).EQ.ATLIB(7)) GO TO 10
C   NO, SEE IF IT CAN BE ADVANCED IN THE SYSTEM
CALL SCHDL(1,0.,ATLIB)
RETURN
C   YES, THIS IS THE STATION
10 NSTN=ATLIB(3)+0.5
J=NSTN+3
C   CHECK THE QUEUE OF THE STATION
IF(NNQ(J).GT.0) GO TO 30
C   CHECK FOR THE AVAILABILITY OF THE SERVERS.
IBUSY=IBLOCK(NSTN)+XX(NSTN)
IF(IBUSY.GE.NSRVR(NSTN)) GO TO 20
C   SCHEDULE THE END OF SERVICE, ALSO UPDATE THE SYSTEM
N1=ATLIB(9)+0.5
XX(NSTN)=XX(NSTN)+1.
CC(N1)=CC(N1)-1.
TCON=TNOW-ATLIB(8)
CALL CCLCT(0,TCON,N1)
CALL MMSTT(0,TNOW,CC(N1),N1)
XN=N1
I=ATLIB(3)+0.5
CALL SERVIC(I,SVT)
CALL SCHDL(3,SVT,ATLIB)
C   ADVANCE ALL THE FOLLOWING PARTS
CALL ADVNC2
RETURN
C   THERE IS NO SERVER AVAILABLE
C   IS RECIRCULATION ALLOWED ?
20 IF(NRECIR.EQ.1) CALL RECIR2
C   NO, PLACE IT IN THE QUEUE
IF(NRECIR.EQ.0) CALL FILEM(J,ATLIB)
IF(NRECIR.EQ.1) RETURN
MMST(N1)=1
TSTOP(N1)=TNOW
RETURN
C   THERE IS ANOTHER PART AHEAD OF IT
30 N1=ATLIB(9)+0.5
MMST(N1)=1
TSTOP(N1)=TNOW
CALL FILEM(J,ATLIB)
RETURN
END

```

```

SUBROUTINE CONVR3
C   A PART HAS REACHED END OF A CONVEYOR. DETERMINE WHERE DOES IT GO
C   NEXT ? USED INSTEAD OF CONVR1, OR CONVR2. CLASS 3 CONVEYOR
COMMON/SCOM1/ATRIB(100),DD(100),DDL(100),DTNOW,II,MFA,MSTOP,
$NCLNR,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100),TNEXT,TNOW,
$XX(100)
COMMON/UCOM1/NCTYPE,NNODE,LINKN(100,8),NARC,ASIZE(100),
1LOCATA(100,2),SPEED,USIZE,NSAREA,NSERV(25),LOCATS(25,11),
2NPART,NVISIT,NROUTE(25,25),NEPT,ARRNT(25),XMEAN(25),ULO(25),
3UHI(25),LOCATP(25),NCAP(100),
4TTIME(100),ISHORT(200,100),DIST(100,100),IPATH(100),
5NPATH(100),CAP(100),NVST(100),IBLOCK(25),MMST(100),TSTOP(100),
6NRECIR,DISTLD,TMLOAD(100),CC(100),TOTAL,TOTPR
7,XLO(25),XMODE(25),XMN(25),STD(25),ARINT,AARRNT(25),NMODE(25),
8XXMEAN(25),UULO(25),XXLO(25),XXHI(25),XXMODE(25),XXMN(25)
9,SSTD(25),MODE(25),XHI(25),UUHI(25)
COMMON/UCOM2/SAMP(100,5),TSAMP(100,6),TIMCLR,TIMSTP
COMMON/UCOM3/MAXPRT(25)
ATTRIB(5)=ATTRIB(5)+1.
ATTRIB(9)=ATTRIB(10)
C   HAS IT REACHED THE STATION IT IS SUPPOSED TO GO TO ?
IF(ATTRIB(5).EQ.ATTRIB(7)) GO TO 10
C   NO, SEE IF IT CAN BE ADVANCED IN THE SYSTEM
CALL SCHDL(1,0.,ATTRIB)
RETURN
C   YES, THIS IS THE STATION
10 NSTN=ATTRIB(3)+0.5
J=NSTN+3
C   CHECK THE QUEUE OF THE STATION
IF(NNQ(J).GT.0) GO TO 20
C   CHECK FOR THE AVAILABILITY OF THE SERVERS.
IBUSY=IBLOCK(NSTN)+XX(NSTN)
IF(IBUSY.GE.NSERVR(NSTN)) GO TO 20
C   SCHEDULE THE END OF SERVICE, ALSO UPDATE THE SYSTEM
N1=ATTRIB(9)+0.5
XX(NSTN)=XX(NSTN)+1.
CC(N1)=CC(N1)-1.
TCON=TNOW-ATTRIB(8)
CALL CCLCT(0,TCON,N1)
CALL MMSTT(0,TNOW,CC(N1),N1)
XN=N1
MMST(N1)=0
NSTN=ATTRIB(3)+0.5
CALL SERVIC(NSTN,SVT)
CALL SCHDL(3,SVT,ATTRIB)
C   ADVANCE ALL THE FOLLOWING PARTS
CALL ADVNC3
RETURN
C   THERE IS NO SERVER AVAILABLE, OR
C   THERE IS ANOTHER PART AHEAD OF IT
20 N1=ATTRIB(9)+0.5
MMST(N1)=1
TSTOP(N1)=TNOW
CALL FILEM(J,ATTRIB)
RETURN
END

```

```

SUBROUTINE RECIR1
C   A PART NEEDS TO BE RECIRCULATED ON THE CONVEYOR
COMMON/SCOM1/ATTRIB(100),DD(100),DDL(100),DTNOW,II,MFA,MSTOP,
$NCLNR,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100),TNEXT,TNOW,
$XX(100)
COMMON/UCOM1/NCTYPE,NNODE,LINKN(100,8),NARC,ASIZE(100),
1LOCATA(100,2),SPEED,USIZE,NSAREA,NSERV(25),LOCATS(25,11),
2NPART,NVISIT,NROUTE(25,25),NEPT,ARRNT(25),XMEAN(25),ULO(25),
3UHI(25),LOCATP(25),NCAP(100),
4TTIME(100),ISHORT(200,100),DIST(100,100),IPATH(100),
5NPATH(100),CAP(100),NVST(25),IBLOCK(25),MMST(100),TSTOP(100),
6NRECIR,DISTLD,TMLOAD(100),CC(100),TOTAL,TOTPR
7,XLO(25),XMODE(25),XMN(25),STD(25),ARINT,AARRNT(25),NMODE(25),
8XXMEAN(25),UULO(25),XXLO(25),XXHI(25),XXMODE(25),XXMN(25)
9,SSTD(25),MODE(25),XHI(25),UUHI(25)
C   FIND THE PATH TO GET BACK TO THIS STATION
NTHIS=ATTRIB(3)+0.5
NEXT=ATTRIB(3)+0.5
J=NTHIS+3
CALL FINDS2(NTHIS,NEXT)
CALL PATH(NTHIS,NEXT,NODPTH)
NODAT=NPAT(1)
NODTO=NPAT(2)
CALL FINDA(NODAT,NODTO,NNARC)
C   IS THERE ANY PLACE ON THE NEXT CONVEYOR ?
IF(CC(NNARC).GE.CAP(NNARC)) GO TO 20
C   YES, RECIRCULATE IT
CC(NNARC)=CC(NNARC)+1.
CALL MMSTT(0,TNOW,CC(NNARC),NNARC)
N1=ATTRIB(9)+0.5
CC(N1)=CC(N1)-1.
CALL MMSTT(0,TNOW,CC(N1),N1)
TCON=TNOW-ATTRIB(8)
CALL CCLCT(0,TCON,N1)
ATTRIB(5)=1.
ATTRIB(7)=NODPTH
ATTRIB(8)=TNOW
ATTRIB(10)=NNARC
K=11
DO 10 I=1,NODPTH
ATTRIB(K)=NPAT(I)
K=K+1
10 CONTINUE
CALL SCHDL(2,TTIME(NNARC),ATTRIB)
CALL ADVNC1
RETURN
C   NO, CANNOT RECIRCULATE, PLACE IT IN THE QUEUE
20 CALL FILEM(J,ATTRIB)
RETURN
END

```

```

SUBROUTINE RECIR2
C   A PART NEEDS TO BE RECIRCULATED ON THE CONVEYOR
COMMON/SCOM1/ATRI(100),DD(100),DDL(100),DTNOW,II,MFA,MSTOP,
$NCLNR,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100),TNEXT,TNOW,
$XX(100)
COMMON/UCOM1/NCTYPE,NNODE,LINKN(100,8),NARC,ASIZE(100),
1LOCATA(100,2),SPEED,USIZE,NSAREA,NSERV(25),LOCATS(25,11),
2NPART,NVISIT,NROUTE(25,25),NEPT,ARRNT(25),XMEAN(25),ULO(25),
3UHI(25),LOCATP(25),NCAP(100),
4TTIME(100),ISHORT(200,100),DIST(100,100),IPATH(100),
5NPATH(100),CAP(100),NVST(25),IBLOCK(25),MMST(100),TSTOP(100),
6NRECIR,DISTLD,TMLOAD(100),CC(100),TOTAL,TOTPR
7,XLO(25),XMODE(25),XMN(25),STD(25),ARINT,AARRNT(25),NMODE(25),
8XXMEAN(25),UULO(25),XXLO(25),XXHI(25),XXMODE(25),XXMN(25)
9,SSTD(25),MODE(25),XHI(25),UUHI(25)
C   FIND THE PATH TO GET BACK TO THIS STATION
NTHIS=ATRI(3)+0.5
NEXT=ATRI(3)+0.5
J=NTHIS+3
CALL FINDS2(NTHIS,NEXT)
CALL PATH(NTHIS,NEXT,NODPTH)
NODAT=NPAT(1)
NODTO=NPAT(2)
CALL FINDA(NODAT,NODTO,NNARC)
N1=ATRI(9)+0.5
C   IS CONVEYOR RUNNING ?
IF(MMST(NNARC).GT.0) GO TO 20
C   IS THERE ANY PLACE ON THE NEXT CONVEYOR ?
IF(CC(NNARC).GE.CAP(NNARC)) GO TO 20
C   YES, RECIRCULATE IT
CC(NNARC)=CC(NNARC)+1.
CALL MMSTT(0,TNOW,CC(NNARC),NNARC)
CC(N1)=CC(N1)-1.
CALL MMSTT(0,TNOW,CC(N1),N1)
TCON=TNOW-ATRI(8)
CALL CCLCT(0,TCON,N1)
ATRI(2)=ATRI(3)
ATRI(5)=1.
ATRI(7)=NODPTH
ATRI(8)=TNOW
ATRI(10)=NNARC
K=11
DO 10 I=1,NODPTH
ATRI(K)=NPAT(I)
K=K+1
10 CONTINUE
C   IS A PART AHEAD OF IT
IF(CC(NNARC).GT.1.) GO TO 30
CALL SCHDL(2,TTIME(NNARC),ATRI)
CALL ADVNC2
RETURN
C   NO, CANNOT RECIRCULATE, PLACE IT IN THE QUEUE
20 MMST(N1)=1
TSTOP(N1)=TNOW
CALL FILEM(J,ATRI)
RETURN
C   A PART IS AHEAD
30 CALL FILEM(2,ATRI)
CALL ADVNC2
END

```

```

SUBROUTINE LOAD(1)
C   A LOADING POINT HAS ARRIVED, CHECK IF THERE IS A PART WAITING
C   TO BE LOADED ONTO THE CONVEYOR
COMMON/SCOM1/ATRI(100),DD(100),DDL(100),DTNOW,II,MFA,MSTOP,
$NCLNR,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100),TNEXT,TNOW,
$XX(100)
COMMON/UCOM1/NCTYPE,NNODE,LINKN(100,8),NARC,ASIZE(100),
1LOCATA(100,2),SPEED,USIZE,NSAREA,NSERV(25),LOCATS(25,11),
2NPART,NVISIT,NROUTE(25,25),NEPT,ARRNT(25),XMEAN(25),ULO(25),
3UHI(25),LOCATP(25),NCAP(100),
4TTIME(100),ISHORT(200,100),DIST(100,100),IPATH(100),
5NPATH(100),CAP(100),NVST(100),IBLOCK(25),MMST(100),TSTOP(100),
6NRECIR,DISTLD,TMLOAD(100),CC(100),TOTAL,TOTPR
7,XLO(25),XMODE(25),XMN(25),STD(25),ARINT,AARRNT(25),NMODE(25),
8XXMEAN(25),UULO(25),XXLO(25),XXHI(25),XXMODE(25),XXMN(25)
9,SSTD(25),MODE(25),XHI(25),UUHI(25)
COMMON/UCOM2/SAMP(100,5),TSAMP(100,6),TIMCLR,TIMSTP
COMMON/UCOM3/MAXPRT(25)
C   FIND OUT WHICH CONVEYOR IT IS AND SEE IF IT IS MOVING NOW
NCON=ATRI(9)+0.5
IF(MMST(NCON).GT.0) RETURN
IF(CC(NCON).GE.CAP(NCON)) RETURN
C   CHECK IF IT IS THE FIRST CONVEYOR, OR THE LOADING STATION
IF(NCON.GT.1) GO TO 20
C   FIRST CONVEYOR
ATRI(9)=NCON
C   SCHEDULE NEXT LOADING POINT
CALL SCHDL(4,ARINT,ATRI)
C   NOTE THE TIME
TMLOAD(NCON)=TNOW
C   CHECK THE QUEUE
IF(NNQ(1).EQ.0) RETURN
C   LOAD THE FIRST PART
CALL RMOVE(1,1,ATRI)
ATRI(8)=TNOW
CC(NCON)=CC(NCON)+1.
CALL MMSTT(0,TNOW,CC(NCON),NCON)
C   CHECK IF A PART IS AHEAD
IF(CC(NCON).GT.1.) GO TO 10
CALL SCHDL(2,TTIME(NCON),ATRI)
RETURN
C   ANOTHER PART IS AHEAD
10 CALL FILEM(2,ATRI)
RETURN
C   THIS IS NOT THE FIRST CONVEYOR, CHECK THE PROPER QUEUE FOR
C   THE PROPER PART TO BE LOADED
20 ATRI(9)=NCON
CALL SCHDL(4,ARINT,ATRI)
TMLOAD(NCON)=TNOW
XN=NCON
NS=NCON+3
C   LOAD THE FIRST PART WAITING AT THE STATION
NRANK1=NFIND(1,3,10,0.,XN,0.)
IF(NRANK1.GT.0) GO TO 40
NRANK1=NFIND(1,NS,10,0.,XN,0.)

```



```

      IF(NRANK1.EQ.0) RETURN
      CALL RMOVE(NRANK1,NS,ATLIB)
      GO TO 60
40  CALL RMOVE(NRANK1,3,ATLIB)
60  N1=ATLIB(9)+0.5
      IF(N1.GT.0) MMST(N1)=0
      CC(NCON)=CC(NCON)+1.
      CALL MMSTT(0,TNOW,CC(NCON),NCON)
      IF(N1.GT.0) CC(N1)=CC(N1)-1.
      TCON=TNOW-ATLIB(8)
      IF(N1.GT.0) CALL CCLCT(0,TCON,N1)
      IF(N1.GT.0) CALL MMSTT(0,TNOW,CC(N1),N1)
      ATLIB(8)=TNOW
C   IS ANOTHER PART AHEAD?
      IF(CC(NCON).GT.1.) GO TO 70
      CALL SCHDL(2,TTIME(NCON),ATLIB)
C   CHECK IF A SERVER WAS BLOCKED
      IF(ATLIB(5).EQ.1.) GO TO 80
      GO TO 90
C   ANOTHER PART IS AHEAD
70  CALL FILEM(2,ATLIB)
C   CHECK IF A SERVER WAS BLOCKED
      IF(ATLIB(5).EQ.1.) GO TO 80
      GO TO 90
C   FREE SERVER
80  NSTN=ATLIB(2)+0.5
      IBLOCK(NSTN)=IBLOCK(NSTN)-1
      J=NSTN+3
C   CHECK THE STATION QUEUE
      IF(NNQ(J).EQ.0) GO TO 100
      CALL COPY(1,J,ATLIB)
C   IS THIS THE STATION IT IS SUPPOSED TO GO TO?
      IF(ATLIB(5).NE.ATLIB(7)) RETURN
C   YES, SCHEDULE THE SERVICE
      CALL RMOVE(1,J,ATLIB)
      N1=ATLIB(9)+0.5
      XX(NSTN)=XX(NSTN)+1.
      CC(N1)=CC(N1)-1.
      TCON=TNOW-ATLIB(8)
      CALL CCLCT(0,TCON,N1)
      CALL MMSTT(0,TNOW,CC(N1),N1)
      MMST(N1)=0
      NSTN=ATLIB(3)+0.5
      CALL SERVIC(NSTN,SVT)
      CALL SCHDL(3,SVT,ATLIB)
C   ADVANCE ALL PARTS FOLLOWING
90  CALL ADVNC3
      RETURN
C   NOBODY IS WAITING AT THE STATION. MOVE THE PARTS ON THE LAST
C   CONVEYOR. REPEAT PROCEDURE UNTIL POSSIBLE
100 N1=NSTN-1
      MMST(N1)=0
      XN=N1
      NRANK=NFIND(1,2,10,0.,XN,0.)
      IF(NRANK.EQ.0) GO TO 110

```

```

      CALL RMOVE(NRANK,2,ATLIB)
      N1=ATLIB(10)+0.5
      TT=TSTOP(N1)-ATLIB(8)
      IF(TT.LE.0.) TT=0.
      RT=TTIME(N1)-TT
      CALL SCHDL(2,RT,ATLIB)
C      START THE CONVEYOR
      110 T1=TSTOP(N1)-TMLOAD(N1)
      IF(T1.LE.0.) T1=0.
      R1=ARINT-T1
      ATLIB(9)=XN
      CALL SCHDL(4,R1,ATLIB)
      TMLOAD(N1)=TNOW
C      WHERE WAS THE LAST PART MOVED FROM ?
C      CHECK THE QUEUE, MOVE THE FIRST PART IF POSSIBLE
      NS=N1+3
      IF(NS.EQ.4) RETURN
      NRANK=NFIND(1,3,10,0.,XN,0.)
      IF(NRANK.GT.0) RETURN
      NSTN=N1
      IF(NNQ(NS).EQ.0) GO TO 100
      RETURN
      END

```

```

SUBROUTINE ENDSV1
C   A PART HAS JUST FINISHED SERVICE. DETERMINE WHERE DOES IT GO NEXT?
COMMON/SCOM1/ATTRIB(100),DD(100),DDL(100),DTNOW,II,MFA,MSTOP,
$NCLNR,NCRDR,NPRNT,NNRUN,NNSE,NTAPE,SS(100),SSL(100),TNEXT,TNOW,
$XX(100)
COMMON/UCOM1/NCTYPE,NNODE,LINKN(100,8),NARC,ASIZE(100),
1LOCATA(100,2),SPEED,USIZE,NSAREA,NSRVR(25),LOCATS(25,11),
2NPART,NVISIT,NROUTE(25,25),NEPT,ARRNT(25),XMEAN(25),ULO(25),
3UHI(25),LOCATP(25),NCAP(100),
4TTIME(100),ISHORT(200,100),DIST(100,100),IPATH(100),
5NPATH(100),CAP(100),NVST(25),IBLOCK(25),MMST(100),TSTOP(100),
6NRECIR,DISTLD,TMLOAD(100),CC(100),TOTAL,TOTPR
7,XLO(25),XMODE(25),XMN(25),STD(25),ARINT,AARRNT(25),NMODE(25),
8XXMEAN(25),UULO(25),XXLO(25),XXHI(25),XXMODE(25),XXMN(25)
9,SSTD(25),MODE(25),XHI(25),UUHI(25)
COMMON/UCOM2/SAMP(100,5),TSAMP(100,6),TIMCLR,TIMSTP
COMMON/UCOM3/MAXPRT(25)
NSTN=ATTRIB(3)+0.5
IP=ATTRIB(4)+0.5
ATTRIB(6)=ATTRIB(6)+1.
NSTT=ATTRIB(6)+0.5
C   CHECK IF THIS IS THE LAST STATION ON THE ROUTE
IF(NSTT.NE.NVST(IP)) GO TO 10
C   YES, COLLECT STATISTICS
TSYS=TNOW-ATTRIB(1)
CALL COLCT(TSYS,IP)
TOTAL=TOTAL+1.
5 J=NSTN+3
IJ=1
LJ=1
C   CHECK THE QUEUE
IF(NNQ(J).GT.0) GO TO 40
C   NOBODY IS WAITING
XX(NSTN)=XX(NSTN)-1.
C   CHECK IF THIS IS THE LAST PART IN THE SYSTEM, PRINT SUMMARY
IF(TOTAL.GE.TOTPR) CALL SCHDL(6,0.,A)
C   ADVANCE ALL PARTS FOLLOWING
CALL ADVNC1
RETURN
C   MOVE THE PART ON TO THE NEXT CONVEYOR IF POSSIBLE
10 ATTRIB(2)=ATTRIB(3)
NS=ATTRIB(6)+0.5
NXS=NS+1
ATTRIB(3)=NROUTE(IP,NXS)
NTHIS=ATTRIB(2)+0.5
NEXT=ATTRIB(3)+0.5
CALL FINDS2(NTHIS,NEXT)
CALL PATH(NTHIS,NEXT,NODPTH)
ATTRIB(5)=1.
ATTRIB(7)=NODPTH
ATTRIB(8)=TNOW
ATTRIB(9)=0.
K=11
DO 20 I=1,NODPTH
ATTRIB(K)=NPATH(I)

```

```

      K=K+1
20  CONTINUE
      NODAT=ATRI(11)+0.5
      NODTO=ATRI(12)+0.5
      CALL FINDA(NODAT,NODTO,NNARC)
      ATRI(10)=NNARC
      IF(CC(NNARC).GE.CAP(NNARC)) GO TO 30
      CC(NNARC)=CC(NNARC)+1.
      CALL MMSTT(0,TNOW,CC(NNARC),NNARC)
      CALL SCHDL(2,TTIME(NNARC),ATRI)
      GO TO 5
C    PART CANNOT ADVANCE, BLOCK THE SERVER
30  CALL FILEM(1,ATRI)
      IBLOCK(NSTN)=IBLOCK(NSTN)+1
      XX(NSTN)=XX(NSTN)-1.
      RETURN
C    CHECK THE FIRST PART IN THE QUEUE, WHERE DOES IT GO ?
40  CALL COPY(1,J,ATRI)
      I=1
      GO TO 60
50  CALL RMOVE(1,J,ATRI)
C    IS THIS THE STATION IT IS SUPPOSED TO GO TO
60  IF(ATRI(5).EQ.ATRI(7)) GO TO 70
C    NO, SEE IF THE NEXT CONVEYOR IS FULL OR NOT
      NODAT=ATRI(11)+0.5
      NODTO=ATRI(12)+0.5
      CALL FINDA(NODAT,NODTO,NNARC)
      K=ATRI(9)+0.5
      IF(CC(NNARC).LT.CAP(NNARC)) GO TO 80
C    PART CANNOT BE ADVANCED
65  IF(LJ.EQ.1) XX(NSTN)=XX(NSTN)-1.
C    IF ANY PART CAN BE ADVANCED IN THE SYSTEM
      IF(IJ.GT.1.OR.LJ.GT.1) CALL ADVNC1
      RETURN
C    THIS IS THE STATION, SCHEDULE THE NEXT EVENT IF IT CAN BE SERVICED
70  IF(LJ.GT.1) GO TO 75
      CALL RMOVE(1,J,ATRI)
      N1=ATRI(9)+0.5
      CC(N1)=CC(N1)-1.
      TCON=TNOW-ATRI(8)
      CALL CCLCT(0,TCON,N1)
      CALL MMSTT(0,TNOW,CC(N1),N1)
      XN=N1
      I=ATRI(3)+0.5
      CALL SERVIC(I,SVT)
      CALL SCHDL(3,SVT,ATRI)
      LJ=LJ+1
C    CHECK THE QUEUE
      IF(NNQ(J).GT.0) GO TO 40
C    ADVANCE ALL PARTS FOLLOWING
75  CALL ADVNC1
      RETURN
80  IF(I.GT.1) GO TO 90
      I=I+1
      GO TO 50
C    THERE IS SPACE ON THE NEXT CONVEYOR, UPDATE STATS, SCHEDULE NEXT
C    EVENT
90  CC(K)=CC(K)-1.
      TCON=TNOW-ATRI(8)
      CALL CCLCT(0,TCON,K)
      CALL MMSTT(0,TNOW,CC(K),K)
      ATRI(10)=NNARC
      CC(NNARC)=CC(NNARC)+1.
      CALL MMSTT(0,TNOW,CC(NNARC),NNARC)
      ATRI(8)=TNOW
      CALL SCHDL(2,TTIME(NNARC),ATRI)
      IJ=IJ+1
C    CHECK THE QUEUE
      IF(NNQ(J).GT.0) GO TO 40
C    ADVANCE ALL PARTS FOLLOWING
      CALL ADVNC1
      END

```

```

SUBROUTINE ENDSV2
C   A PART HAS JUST FINISHED SERVICE. DETERMINE WHERE DOES IT GO NEXT?
COMMON/SCOM1/ATTRIB(100),DD(100),DDL(100),DTNOW,II,MFA,MSTOP,
SNCLNR,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100),TNEXT,TNOW,
SXX(100)
COMMON/UCOM1/NCTYPE,NNODE,LINKN(100,8),NARC,ASIZE(100),
1LOCATA(100,2),SPEED,USIZE,NSAREA,NSERV(25),LOCATS(25,11),
2NPART,NVISIT,NROUTE(25,25),NEPT,ARRNT(25),XMEAN(25),ULO(25),
3UHI(25),LOCATP(25),NCAP(100),
4TTIME(100),ISHORT(200,100),DIST(100,100),IPATH(100),
5NPATH(100),CAP(100),NVST(25),IBLOCK(25),MMST(100),TSTOP(100),
6NRECIR,DISTLD,TMLOAD(100),CC(100),TOTAL,TOTPR
7,XLO(25),XMODE(25),XMN(25),STD(25),ARINT,AARRNT(25),NMODE(25),
8XXMEAN(25),UULO(25),XXLO(25),XXHI(25),XXMODE(25),XXMN(25)
9,SSTD(25),MODE(25),XHI(25),UUHI(25)
COMMON/UCOM2/SAMP(100,5),TSAMP(100,6),TIMCLR,TIMSTP
COMMON/UCOM3/MAXPRT(25)
NSTN=ATTRIB(3)+0.5
IP=ATTRIB(4)+0.5
ATTRIB(6)=ATTRIB(6)+1.
NSTT=ATTRIB(6)+0.5
C   CHECK IF THIS IS THE LAST STATION ON THE ROUTE
IF(NSTT.NE.NVST(IP)) GO TO 10
C   YES, COLLECT STATISTICS
TSYS=TNOW-ATTRIB(1)
CALL COLCT(TSYS,IP)
TOTAL=TOTAL+1.
5 J=NSTN+3
IJ=1
LJ=1
C   CHECK THE QUEUE
IF(NNQ(J).GT.0) GO TO 40
C   NOBODY IS WAITING
XX(NSTN)=XX(NSTN)-1.
C   CHECK IF THIS IS THE LAST PART IN THE SYSTEM, PRINT SUMMARY
IF(TOTAL.GE.TOTPR) CALL SCHDL(6,0.,A)
C   ADVANCE ALL PARTS FOLLOWING
CALL ADVNC2
RETURN
C   MOVE THE PART ON TO THE NEXT CONVEYOR IF POSSIBLE
10 ATTRIB(2)=ATTRIB(3)
NS=ATTRIB(6)+0.5
NXS=NS+1
ATTRIB(3)=NROUTE(IP,NXS)
NTHIS=ATTRIB(2)+0.5
NEXT=ATTRIB(3)+0.5
CALL FINDS2(NTHIS,NEXT)
CALL PATH(NTHIS,NEXT,NODPTH)
ATTRIB(5)=1.
ATTRIB(7)=NODPTH
ATTRIB(8)=TNOW
ATTRIB(9)=0.
K=11
DO 20 I=1,NODPTH
ATTRIB(K)=NPATN(I)

```

```

      K=K+1
20  CONTINUE
      NODAT=ATRI(11)+0.5
      NODTO=ATRI(12)+0.5
      CALL FINDA(NODAT,NODTO,NNARC)
      ATRI(10)=NNARC
      IF(MMST(NNARC).GT.0) GO TO 30
      IF(CC(NNARC).GE.CAP(NNARC)) GO TO 30
      CC(NNARC)=CC(NNARC)+1.
      CALL MMSTT(0,TNOW,CC(NNARC),NNARC)
      IF(CC(NNARC).GT.1.) GO TO 35
      CALL SCHDL(2,TTIME(NNARC),ATRI)
      GO TO 5
C    PART CANNOT ADVANCE, BLOCK THE SERVER
30  CALL FILEM(1,ATRI)
      IBLOCK(NSTN)=IBLOCK(NSTN)+1
      XX(NSTN)=XX(NSTN)-1.
      RETURN
35  CALL FILEM(2,ATRI)
      GO TO 5
C    CHECK THE FIRST PART IN THE QUEUE, WHERE DOES IT GO ?
40  CALL COPY(1,J,ATRI)
      I=1
      GO TO 60
50  CALL RMOVE(1,J,ATRI)
C    IS THIS THE STATION IT IS SUPPOSED TO GO TO
60  IF(ATRI(5).EQ.ATRI(7)) GO TO 70
C    NO, SEE IF THE NEXT CONVEYOR IS FULL OR NOT
      NODAT=ATRI(11)+0.5
      NODTO=ATRI(12)+0.5
      CALL FINDA(NODAT,NODTO,NNARC)
      K=ATRI(9)+0.5
      IF(MMST(NNARC).GT.0) GO TO 65
      IF(CC(NNARC).LT.CAP(NNARC)) GO TO 80
C    PART CANNOT BE ADVANCED
65  IF(LJ.EQ.1) XX(NSTN)=XX(NSTN)-1.
C    IF ANY PART CAN BE ADVANCED IN THE SYSTEM
      IF(IJ.GT.1.OR.LJ.GT.1) CALL ADVNC2
      RETURN
C    THIS IS THE STATION, SCHEDULE THE NEXT EVENT IF IT CAN BE SERVICED
70  IF(LJ.GT.1) GO TO 75
      CALL RMOVE(1,J,ATRI)
      N1=ATRI(9)+0.5
      CC(N1)=CC(N1)-1.
      TCON=TNOW-ATRI(8)
      CALL CCLCT(0,TCON,N1)
      CALL MMSTT(0,TNOW,CC(N1),N1)
      XN=N1
      MMST(N1)=0
      I=ATRI(3)+0.5
      CALL SERVIC(1,SVT)
      CALL SCHDL(3,SVT,ATRI)
      LJ=LJ+1
C    CHECK THE QUEUE
      IF(NNQ(J).GT.0) GO TO 40

```

```

C   ADVANCE ALL PARTS FOLLOWING
75  CALL ADVNC2
    RETURN
80  IF(I.GT.1) GO TO 90
    I=I+1
    GO TO 50
C   THERE IS SPACE ON THE NEXT CONVEYOR, UPDATE STATS, SCHEDULE NEXT
C   EVENT
90  CC(K)=CC(K)-1.
    TCON=TNOW-TRIB(8)
    CALL CCLCT(0,TCON,K)
    CALL MMSTT(0,TNOW,CC(K),K)
    TRIB(10)=NNARC
    CC(NNARC)=CC(NNARC)+1.
    CALL MMSTT(0,TNOW,CC(NNARC),NNARC)
    TRIB(8)=TNOW
    IJ=IJ+1
    MMST(K)=0
    IF(CC(NNARC).LE.1.) GO TO 95
    CALL FILEM(2,TRIB)
    GO TO 96
95  CALL SCHDL(2,TIME(NNARC),TRIB)
C   CHECK THE QUEUE
96  IF(NNQ(J).GT.0) GO TO 40
C   ADVANCE ALL PARTS FOLLOWING
    GO TO 65
    END

```

```

SUBROUTINE ENDSV3
C   A PART HAS JUST FINISHED SERVICE. DETERMINE WHERE DOES IT GO NEXT?
COMMON/SCOM1/ATRI(100),DD(100),DDL(100),DTNOW,II,MFA,MSTOP,
$NCLNR,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100),TNEXT,TNOW,
$XX(100)
COMMON/UCOM1/NCTYPE,NNODE,LINKN(100,8),NARC,ASIZE(100),
1LOCATA(100,2),SPEED,USIZE,NSAREA,NSERV(25),LOCATS(25,11),
2NPART,NVISIT,NROUTE(25,25),NEPT,ARRNT(25),XMEAN(25),ULO(25),
3UHI(25),LOCATP(25),NCAP(100),
4TTIME(100),ISHORT(200,100),DIST(100,100),IPATH(100),
5NPATH(100),CAP(100),NVST(100),IBLOCK(25),MMST(100),TSTOP(100),
6NRECIR,DISTLD,TMLOAD(100),CC(100),TOTAL,TOTPR
7,XLO(25),XMODE(25),XMN(25),STD(25),ARINT,AARRNT(25),NMODE(25),
8XXMEAN(25),UULO(25),XXLO(25),XXHI(25),XXMODE(25),XXMN(25)
9,SSTD(25),MODE(25),XHI(25),UUHI(25)
COMMON/UCOM2/SAMP(100,5),TSAMP(100,6),TIMCLR,TIMSTP
COMMON/UCOM3/MAXPRT(25)
NSTN=ATRI(3)+0.5
IP=ATRI(4)+0.5
N1=ATRI(9)+0.5
ATRI(6)=ATRI(6)+1.
NSTT=ATRI(6)+0.5
C   CHECK IF THIS IS THE LAST STATION ON THE ROUTE
IF(NSTT.NE.NVST(IP)) GO TO 10
C   YES, COLLECT STATISTICS
TSYS=TNOW-ATRI(1)
CALL COLCT(TSYS,IP)
TOTAL=TOTAL+1.
5 J=NSTN+3
C   CHECK THE QUEUE
IF(NNQ(J).GT.0) GO TO 30
C   NOBODY IS WAITING
XX(NSTN)=XX(NSTN)-1.
C   CHECK IF THIS IS THE LAST PART IN THE SYSTEM, PRINT SUMMARY
IF(TOTAL.GE.TOTPR) CALL SCHDL(6,0.,A)
C   ADVANCE ALL PARTS FOLLOWING
MMST(N1)=0
CALL ADVNC3
RETURN
C   MOVE THE PART ON TO THE NEXT CONVEYOR IF POSSIBLE
10 ATRI(2)=ATRI(3)
NS=ATRI(6)+0.5
NXS=NS+1
ATRI(3)=NROUTE(IP,NXS)
NTHIS=ATRI(2)+0.5
NEXT=ATRI(3)+0.5
CALL FINDS2(NTHIS,NEXT)
CALL PATH(NTHIS,NEXT,NODPTH)
ATRI(5)=1.
ATRI(7)=NODPTH
ATRI(9)=0.
K=11
DO 20 I=1,NODPTH
ATRI(K)=NPATH(I)
K=K+1

```



```

20 CONTINUE
   NODAT=ATRIB(11)+0.5
   NODTO=ATRIB(12)+0.5
   CALL FINDA(NODAT,NODTO,NNARC)
   ATRIB(10)=NNARC
C   PART MUST WAIT FOR THE NEXT CONVEYOR LOADING POINT
   CALL FILEM(3,ATRIB)
   IBLOCK(NSTN)=IBLOCK(NSTN)+1
   XX(NSTN)=XX(NSTN)-1.
   RETURN
C   CHECK THE FIRST PART IN THE QUEUE, WHERE DOES IT GO ?
30 CALL COPY(1,J,ATRIB)
C   IS THIS THE STATION IT IS SUPPOSED TO GO TO
   IF(ATRIB(5).EQ.ATRIB(7)) GO TO 40
C   NO
C   PART CANNOT BE ADVANCED
   XX(NSTN)=XX(NSTN)-1.
   RETURN
C   THIS IS THE STATION, SCHEDULE THE NEXT EVENT IF IT CAN BE SERVICED
40 CALL RMOVE(1,J,ATRIB)
   N1=ATRIB(9)+0.5
   CC(N1)=CC(N1)-1.
   TCON=TNOW-ATRIB(8)
   CALL CCLCT(0,TCON,N1)
   CALL MMSTT(0,TNOW,CC(N1),N1)
   MMST(N1)=0
   NSTN=ATRIB(3)+0.5
   CALL SERVIC(NSTN,SVT)
   CALL SCHDL(3,SVT,ATRIB)
C   ADVANCE ALL PARTS FOLLOWING
   CALL ADVNC3
   RETURN
END

```

```

SUBROUTINE ADVNC1
C   A PART AHEAD HAS MOVED IN THE SYSTEM. ALL PARTS FOLLOWING ADVANCE
COMMON/SCOM1/ATTRIB(100),DD(100),DDL(100),DTNOW,II,MFA,MSTOP,
$NCLNR,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100),TNEXT,TNOW,
$XX(100)
COMMON/UCOM1/NCTYPE,NNODE,LINKN(100,8),NARC,ASIZE(100),
1LOCATA(100,2),SPEED,USIZE,NSAREA,NSERV(25),LOCATS(25,11),
2NPART,NVISIT,NROUTE(25,25),NEPT,ARRNT(25),XMEAN(25),ULO(25),
3UHI(25),LOCATP(25),NCAP(100),
4TTIME(100),ISHORT(200,100),DIST(100,100),IPATH(100),
5NPATH(100),CAP(100),NVST(25),IBLOCK(25),MMST(100),TSTOP(100),
6NRECIR,DISTLD,TMLOAD(100),CC(100),TOTAL,TOTPR
7,XLO(25),XMODE(25),XMN(25),STD(25),ARINT,AARRNT(25),NMODE(25),
8XXMEAN(25),UULO(25),XXLO(25),XXHI(25),XXMODE(25),XXMN(25)
9,SSTD(25),MODE(25),XHI(25),UUHI(25)
COMMON/UCOM2/SAMP(100,5),TSAMP(100,6),TIMCLR,TIMSTP
COMMON/UCOM3/MAXPRT(25)
N1=ATTRIB(9)+0.5
XN=N1
C   WHERE WAS THE LAST PART MOVED FROM ?
5  NODFR=ATTRIB(11)+0.5
   IF(ATTRIB(9).NE.0..AND.ATTRIB(9).NE.ATTRIB(10)) CALL FINDL(NODFR)
C   WAS THAT A STATION ?
6  CALL FINDS(NODFR,NSTN,EX)
   IJ=1
C   WAS THAT STATION ON ITS ROUTE ?
   IF(ATTRIB(9).EQ.0.) GO TO 10
   IF(EX.EQ.1.) GO TO 40
C   IT WAS ITS OWN STATION OR IT WAS NOT A STATION
C   CHECK THE QUEUE, MOVE THE FIRST PART IF POSSIBLE
10 NRANK1=NFIND(1,1,10,0.,XN,0.)
   CN=ATTRIB(9)
   NRANK2=NFIND(1,1,9,0.,CN,0.)
   IF(NRANK2.EQ.0) GO TO 15
   IF(NRANK2.LT.NRANK1) RETURN
15 IF(NRANK1.EQ.0) RETURN
C   SEE WHERE DOES THE NEXT PART GO
   CALL COPY(NRANK1,1,ATTRIB)
   I=1
20 NODAT=ATTRIB(11)+0.5
   NODTO=ATTRIB(12)+0.5
   CALL FINDA(NODAT,NODTO,NNARC)
   N1=ATTRIB(9)+0.5
   ATTRIB(10)=NNARC
   IF(I.GT.1) GO TO 30
   IF(CC(NNARC).GE.CAP(NNARC)) RETURN
   CALL RMOVE(NRANK1,1,ATTRIB)
   I=I+1
   GO TO 20
30 CC(NNARC)=CC(NNARC)+1.
   CALL MMSTT(0,TNOW,CC(NNARC),NNARC)
   IF(ATTRIB(9).NE.0.) CC(N1)=CC(N1)-1.
   TCON=TNOW-ATTRIB(8)
   IF(ATTRIB(9).NE.0.) CALL CCLCT(0,TCON,N1)
   IF(ATTRIB(9).NE.0) CALL MMSTT(0,TNOW,CC(N1),N1)

```

```

      ATRIB(8)=TNOW
      CALL SCHDL(2, TTIME(NNARC), ATRIB)
C     AFTER UPDATING, SEE IF IT IS COMING FROM ITS OWN STATION
      XN=N1
      IF(ATRIB(5).GT.1.) GO TO 5
C     SEE IF IT WAS BLOCKING ANY SERVER
      NSTN=ATRIB(2)+0.5
      IF(IBLOCK(NSTN).EQ.0) GO TO 5
C     FREE THE SERVER
      IBLOCK(NSTN)=IBLOCK(NSTN)-1
C     LAST PART MOVED WAS AT A STATION, NOT ITS OWN
40    J=NSTN+3
      NRANK=1
      IF(NNQ(J).EQ.0) GO TO 10
C     SEE WHERE DOES THE NEXT PART IN THE QUEUE GO ?
45    CALL COPY(NRANK,J,ATRIB)
      I=1
      GO TO 60
50    CALL RMOVE(NRANK,J,ATRIB)
C     CHECK IF IT IS AT A STATION ON ITS ROUTE
60    IF(ATRIB(5).EQ.ATRIB(7)) GO TO 90
C     NO, CHECK THE NEXT CONVEYOR
      NODAT=ATRIB(11)+0.5
      NODTO=ATRIB(12)+0.5
      CALL FINDA(NODAT,NODTO,NNARC)
      K=ATRIB(9)+0.5
      ATRIB(10)=NNARC
      IF(CC(NNARC).LT.CAP(NNARC)) GO TO 70
C     THERE IS NO SPACE ON THE NEXT CONVEYOR
      IF(IJ.GT.1) GO TO 5
      GO TO 10
70    IF(I.GT.1) GO TO 80
      I=I+1
      GO TO 50
C     THERE IS SPACE ON THE NEXT CONVEYOR, UPDATE, & SCHEDULE NEXT EVENT
80    CC(K)=CC(K)-1.
      TCON=TNOW-ATRIB(8)
      CALL CCLCT(0,TCON,K)
      CALL MMSTT(0,TNOW,CC(K),K)
      CC(NNARC)=CC(NNARC)+1.
      CALL MMSTT(0,TNOW,CC(NNARC),NNARC)
      ATRIB(8)=TNOW
      CALL SCHDL(2, TTIME(NNARC), ATRIB)
      XN=K
      IJ=IJ+1
C     CHECK THE QUEUE
      IF(NNQ(J).GT.0) GO TO 40
C     ADVANCE ALL PARTS FOLLOWING
      GO TO 10
C     THIS THE STATION, CHECK THE STATUS OF THE SERVERS
C     SCHEDULE SERVICE IF POSSIBLE
90    IBUSY=IBLOCK(NSTN)+XX(NSTN)
      IF(IBUSY.GE.NSERVR(NSTN)) GO TO 10
      CALL RMOVE(NRANK,J,ATRIB)
      N1=ATRIB(9)+0.5
      XN=N1
      CC(N1)=CC(N1)-1.
      CALL MMSTT(0,TNOW,CC(N1),N1)
      TCON=TNOW-ATRIB(8)
      CALL CCLCT(0,TCON,N1)
      XX(NSTN)=XX(NSTN)+1.
      I=ATRIB(3)+0.5
      CALL SERVIC(I,SVT)
      CALL SCHDL(3,SVT,ATRIB)
C     AFTER UPDATING, CHECK THE QUEUE AND ADVANCE ALL PARTS FOLLOWING
      IF(NNQ(J).GT.0) GO TO 40
      GO TO 5
      END

```

```

SUBROUTINE ADVNC2
C   A PART AHEAD HAS MOVED IN THE SYSTEM. ALL PARTS FOLLOWING ADVANCE
COMMON/SCOM1/ATTRIB(100),DD(100),DDL(100),DTNOW,II,MFA,MSTOP,
SNCLNR,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100),TNEXT,TNOW,
SXX(100)
COMMON/UCOM1/NCTYPE,NNODE,LINKN(100,8),NARC,ASIZE(100),
1LOCATA(100,2),SPEED,USIZE,NSAREA,NSERV(25),LOCATS(25,11),
2NPART,NVISIT,NROUTE(25,25),NEPT,ARRNT(25),XMEAN(25),ULO(25),
3UHI(25),LOCATP(25),NCAP(100),
4TTIME(100),ISHORT(200,100),DIST(100,100),IPATH(100),
5NPAT(100),CAP(100),NVST(25),IBLOCK(25),MMST(100),TSTOP(100),
6NRECIR,DISTLD,TMLOAD(100),CC(100),TOTAL,TOTPR
7,XLO(25),XMODE(25),XMN(25),STD(25),ARINT,AARRNT(25),NMODE(25),
8XXMEAN(25),UULO(25),XXLO(25),XXHI(25),XXMODE(25),XXMN(25)
9,SSTD(25),MODE(25),XHI(25),UUHI(25)
COMMON/UCOM2/SAMP(100,5),TSAMP(100,6),TIMCLR,TIMSTP
COMMON/UCOM3/MAXPRT(25)
C   SEE IF A PART IS WAITING ON THE CONVEYOR, IF SO ADVANCE IT
N1=ATTRIB(9)+0.5
XN=N1
NRANK=NFIND(1,2,10,0.,XN,0.)
IF(NRANK.EQ.0) GO TO 5
CALL COPY(NRANK,2,ATTRIB)
IF(MMST(N1).GT.0) GO TO 5
CALL RMOVE(NRANK,2,ATTRIB)
NODFR=ATTRIB(11)+0.5
TT=TSTOP(N1)-ATTRIB(8)
RT=TTIME(N1)-TT
CALL SCHDL(2,RT,ATTRIB)
GO TO 6
C   WHERE WAS THE LAST PART MOVED FROM ?
5 NODFR=ATTRIB(11)+0.5
IF(ATTRIB(2).EQ.0) GO TO 6
IF(ATTRIB(9).NE.0.AND.ATTRIB(10)) CALL FINDL(NODFR)
C   WAS THAT A STATION ?
6 CALL FINDS(NODFR,NSTN,EX)
IJ=1
LJ=1
C   WAS THAT STATION ON ITS ROUTE ?
IF(ATTRIB(9).EQ.0) GO TO 10
IF(EX.EQ.1.) GO TO 40
C   IT WAS ITS OWN STATION OR IT WAS NOT A STATION
C   CHECK THE QUEUE, MOVE THE FIRST PART IF POSSIBLE
10 NRANK1=NFIND(1,1,10,0.,XN,0.)
CN=ATTRIB(9)
NRANK2=NFIND(1,1,9,0.,CN,0.)
IF(NRANK2.EQ.0) GO TO 15
IF(NRANK2.LT.NRANK1) GO TO 110
15 IF(NRANK1.EQ.0) GO TO 110
C   SEE WHERE DOES THE NEXT PART GO
CALL COPY(NRANK1,1,ATTRIB)
I=1
20 NODAT=ATTRIB(11)+0.5
NODTO=ATTRIB(12)+0.5
CALL FINDA(NODAT,NODTO,NNARC)

```

```

N1=ATRI(9)+0.5
ATRI(10)=NNARC
IF(I.GT.1) GO TO 30
IF(MMST(NNARC).GT.0) GO TO 110
IF(CC(NNARC).GE.CAP(NNARC)) GO TO 110
CALL RMOVE(NRANK,1,ATRI)
I=I+1
GO TO 20
30 CC(NNARC)=CC(NNARC)+1.
CALL MMST(0,TNOW,CC(NNARC),NNARC)
IF(ATRI(9).NE.0.) CC(N1)=CC(N1)-1.
TCON=TNOW-ATRI(8)
IF(ATRI(9).NE.0.) CALL CCLCT(0,TCON,N1)
IF(ATRI(9).NE.0) CALL MMST(0,TNOW,CC(N1),N1)
ATRI(8)=TNOW
IF(CC(NNARC).LE.1) GO TO 35
CALL FILEM(2,ATRI)
GO TO 36
35 CALL SCHDL(2,TIME(NNARC),ATRI)
C AFTER UPDATING, SEE IF IT IS COMING FROM ITS OWN STATION
36 XN=N1
MMST(N1)=0
LJ=LJ+1
IF(ATRI(5).GT.1.) GO TO 10
C SEE IF IT WAS BLOCKING ANY SERVER
NSTN=ATRI(2)+0.5
IF(IBLOCK(NSTN).EQ.0) GO TO 10
C FREE THE SERVER
IBLOCK(NSTN)=IBLOCK(NSTN)-1
C LAST PART MOVED WAS AT A STATION, NOT ITS OWN
40 J=NSTN+3
NRANK=1
IF(NNQ(J).EQ.0) GO TO 95
C SEE WHERE DOES THE NEXT PART IN THE QUEUE GO ?
45 CALL COPY(NRANK,J,ATRI)
I=1
GO TO 60
50 CALL RMOVE(NRANK,J,ATRI)
C CHECK IF IT IS AT A STATION ON ITS ROUTE
60 IF(ATRI(5).EQ.ATRI(7)) GO TO 90
C NO, CHECK THE NEXT CONVEYOR
NODAT=ATRI(11)+0.5
NODTO=ATRI(12)+0.5
CALL FINDA(NODAT,NODTO,NNARC)
K=ATRI(9)+0.5
IF(MMST(K).GT.0) GO TO 65
IF(CC(NNARC).LT.CAP(NNARC)) GO TO 70
C THERE IS NO SPACE ON THE NEXT CONVEYOR
65 IF(IJ.GT.1) GO TO 5
GO TO 10
70 IF(I.GT.1) GO TO 80
I=I+1
GO TO 50
C THERE IS SPACE ON THE NEXT CONVEYOR, UPDATE, & SCHEDULE NEXT EVENT
80 CC(K)=CC(K)-1.

```

```

      TCON=TNOW-ATLIB(8)
      CALL CCLCT(0,TCON,K)
      CALL MMSTT(0,TNOW,CC(K),K)
      CC(NNARC)=CC(NNARC)+1.
      CALL MMSTT(0,TNOW,CC(NNARC),NNARC)
      ATLIB(8)=TNOW
      ATLIB(10)=NNARC
      MMST(K)=0
      IF(CC(NNARC).LE.1.) GO TO 85
      CALL FILEM(2,ATLIB)
      GO TO 86
85  CALL SCHDL(2,TTIME(NNARC),ATLIB)
86  XN=K
      IJ=IJ+1
C    CHECK THE QUEUE
      IF(NNQ(J).GT.0) GO TO 40
C    ADVANCE ALL PARTS FOLLOWING
      GO TO 95
C    THIS THE STATION, CHECK THE STATUS OF THE SERVERS
C    SCHEDULE SERVICE IF POSSIBLE
90  IBUSY=IBLOCK(NSTN)+XX(NSTN)
      IF(IBUSY.GE.NSERVR(NSTN)) GO TO 95
      CALL RMOVE(NRANK,J,ATLIB)
      N1=ATLIB(9)+0.5
      XN=N1
      CC(N1)=CC(N1)-1.
      MMST(N1)=0
      CALL MMSTT(0,TNOW,CC(N1),N1)
      TCON=TNOW-ATLIB(8)
      CALL CCLCT(0,TCON,N1)
      XX(NSTN)=XX(NSTN)+1.
      IJ=IJ+1
      IP=ATLIB(3)+0.5
      CALL SERVIC(IP,SVT)
      CALL SCHDL(3,SVT,ATLIB)
C    AFTER UPDATING, CHECK THE QUEUE AND ADVANCE ALL PARTS FOLLOWING
      IF(NNQ(J).GT.0) GO TO 40
95  NRANK=NFIND(1,2,10,0.,XN,0.)
      IF(NRANK.GT.0) GO TO 100
96  IF(IJ.GT.1) GO TO 5
      IF(LJ.LE.1) GO TO 10
      RETURN
100 CALL COPY(NRANK,2,ATLIB)
      N1=ATLIB(10)+0.5
      IF(MMST(N1).GT.0) GO TO 96
      CALL RMOVE(NRANK,2,ATLIB)
      N1=ATLIB(10)+0.5
      NODFR=ATLIB(11)+0.5
      TT=TSTOP(N1)-ATLIB(8)
      RT=TTIME(N1)-TT
      CALL SCHDL(2,RT,ATLIB)
      IF(IJ.GT.1) GO TO 6
      GO TO 10
110 IF(LJ.GT.1) GO TO 95
      RETURN
      END

```

```

SUBROUTINE ADVNC3
C   A PART AHEAD HAS MOVED IN THE SYSTEM. ALL PARTS FOLLOWING ADVANCE
COMMON/SCOM1/ATRI(100),DD(100),DDL(100),DTNOW,II,MFA,MSTOP,
$NCLNR,NCRDR,NPRNT,NNRUN,NNSSET,NTAPE,SS(100),SSL(100),TNEXT,TNOW,
$XX(100)
COMMON/UCOM1/NCTYPE,NNODE,LINKN(100,8),NARC,ASIZE(100),
1LOCATA(100,2),SPEED,USIZE,NSAREA,NSERV(25),LOCATS(25,11),
2NPART,NVISIT,NROUTE(25,25),NEPT,ARRNT(25),XMEAN(25),ULO(25),
3UHI(25),LOCATP(25),NCAP(100),
4TTIME(100),ISHORT(200,100),DIST(100,100),IPATH(100),
5NPATH(100),CAP(100),NVST(100),IBLOCK(25),MMST(100),TSTOP(100),
6NRECIR,DISTLD,TMLOAD(100),CC(100),TOTAL,TOTPR
7,XLO(25),XMODE(25),XMN(25),STD(25),ARINT,AARRNT(25),NMODE(25),
8XXMEAN(25),UULO(25),XXLO(25),XXHI(25),XXMODE(25),XXMN(25)
9,SSTD(25),MODE(25),XHI(25),UUHI(25)
COMMON/UCOM2/SAMP(100,5),TSAMP(100,6),TIMCLR,TIMSTP
COMMON/UCOM3/MAXPRT(25)
C   SEE IF A PART IS WAITING ON THE CONVEYOR, IF SO ADVANCE IT
N1=ATRI(9)+0.5
XN=N1
IF(MMST(N1).GT.0) RETURN
NRANK=NFIND(1,2,10,0.,XN,0.)
IF(NRANK.EQ.0) GO TO 5
CALL RMOVE(NRANK,2,ATRI)
N1=ATRI(10)+0.5
TT=TSTOP(N1)-ATRI(8)
IF(TT.LE.0.) TT=0.
RT=TTIME(N1)-TT
CALL SCHDL(2,RT,ATRI)
C   START THE CONVEYOR
5 T1=TSTOP(N1)-TMLOAD(N1)
IF(T1.LE.0.) T1=0.
R1=ARINT-T1
ATRI(9)=XN
CALL SCHDL(4,R1,ATRI)
TMLOAD(N1)=TNOW
C   WHERE WAS THE LAST PART MOVED FROM ?
C   CHECK THE QUEUE, MOVE THE FIRST PART IF POSSIBLE
6 J=N1+3
IF(J.EQ.4) RETURN
NRANK=1
C   IF NOTHING IS AT STATION, MOVE THE PARTS ON THE CONVEYOR
C   FOLLOWING THE STATION.
IF(NNQ(J).EQ.0) GO TO 30
C   SEE WHERE DOES THE NEXT PART IN THE QUEUE GO ?
CALL COPY(NRANK,J,ATRI)
C   CHECK IF IT IS AT A STATION ON ITS ROUTE
IF(ATRI(5).EQ.ATRI(7)) GO TO 10
C   NO
RETURN
C   THIS IS THE STATION, CHECK THE STATUS OF THE SERVERS
C   SCHEDULE SERVICE IF POSSIBLE
10 IBUSY=IBLOCK(NSTN)+XX(NSTN)
IF(IBUSY.GE.NSERV(NSTN)) RETURN
CALL RMOVE(NRANK,J,ATRI)

```

```

N1=ATRI(9)+0.5
XN=N1
CC(N1)=CC(N1)-1.
MMST(N1)=0
CALL MMSTT(0,TNOW,CC(N1),N1)
TCON=TNOW-ATRI(8)
CALL CCLCT(0,TCON,N1)
XX(NSTN)=XX(NSTN)+1.
NSTN=ATRI(3)+0.5
CALL SERVIC(NSTN,SVT)
CALL SCHDL(3,SVT,ATRI)
C AFTER UPDATING, ADVANCE ALL PARTS FOLLOWING
  IF(MMST(N1).GT.0) RETURN
  NRANK=NFIND(1,2,10,0.,XN,0.)
  IF(NRANK.GT.0) GO TO 20
  GO TO 5
20 CALL RMOVE(NRANK,2,ATRI)
  N1=ATRI(10)+0.5
  TT=TSTOP(N1)-ATRI(8)
  RT=TTIME(N1)-TT
  CALL SCHDL(2,RT,ATRI)
C START THE CONVEYOR
  T1=TSTOP(N1)-TMLOAD(N1)
  IF(T1.LE.0.) T1=0.
  R1=ARINT-T1
  ATRI(9)=N1
  CALL SCHDL(4,R1,ATRI)
  TMLOAD(N1)=TNOW
  GO TO 6
C MOVE PARTS ON PRECEEDING CONVEYORS IF POSSIBLE
30 NSTN=N1
40 N1=NSTN-1
  MMST(N1)=0
  XN=N1
  NRANK=NFIND(1,2,10,0.,XN,0.)
  IF(NRANK.EQ.0) GO TO 50
  CALL RMOVE(NRANK,2,ATRI)
  N1=ATRI(10)+0.5
  TT=TSTOP(N1)-ATRI(8)
  IF(TT.LE.0.) TT=0.
  RT=TTIME(N1)-TT
  CALL SCHDL(2,RT,ATRI)
C START THE CONVEYOR
50 T1=TSTOP(N1)-TMLOAD(N1)
  IF(T1.LE.0.) T1=0.
  R1=ARINT-T1
  ATRI(9)=XN
  CALL SCHDL(4,R1,ATRI)
  TMLOAD(N1)=TNOW
C WHERE WAS THE LAST PART MOVED FROM ?
C CHECK THE QUEUE, MOVE THE FIRST PART IF POSSIBLE
  NS=N1+3
  IF(NS.EQ.4) RETURN
  NRANK=NFIND(1,3,10,0.,XN,0.)
  IF(NRANK.GT.0) RETURN
  NSTN=N1
  IF(NNQ(NS).EQ.0) GO TO 40
  RETURN
END

```



```

SUBROUTINE CLRST
C  CLEAR STATISTICS AFTER THE DESIGNATED TIME
COMMON/SCOM1/ATTRIB(100),DD(100),DDL(100),DTNOW,II,MFA,MSTOP,
$NCLNR,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100),TNEXT,TNOW,
$XX(100)
COMMON/UCOM1/NCTYPE,NNODE,LINKN(100,8),NARC,ASIZE(100),
1LOCATA(100,2),SPEED,USIZE,NSAREA,NSERV(25),LOCATS(25,11),
2NPART,NVISIT,NROUTE(25,25),NEPT,ARRNT(25),XMEAN(25),ULO(25),
3UHI(25),LOCATP(25),NCAP(100),
4TTIME(100),ISHORT(200,100),DIST(100,100),IPATH(100),
5NPATH(100),CAP(100),NVST(25),IBLOCK(25),MMST(100),TSTOP(100),
6NRECIR,DISTLD,TMLOAD(100),CC(100),TOTAL,TOTPR
7,XLO(25),XMODE(25),XMN(25),STD(25),ARINT,AARRNT(25),NMODE(25),
8XXMEAN(25),UULO(25),XXLO(25),XXHI(25),XXMODE(25),XXMN(25)
9,SSTD(25),MODE(25),XHI(25),UUHI(25)
COMMON/UCOM2/SAMP(100,5),TSAMP(100,6),TIMCLR,TIMSTP
COMMON/UCOM3/MAXPRT(25)
CALL CLEAR
CALL CCLCT(-1,X,NARC)
CALL MMSTT(-1,TNOW,CC(NARC),NARC)
C  RE-INITIALIZE THE STATISTICS OF TIME PERSISTENT VARIABLES
DO 10 I=1,NARC
IF(CC(I).GT.0.) CALL MMSTT(0,TNOW,CC(I),I)
10 CONTINUE
RETURN
END

```

```

SUBROUTINE SMMRY
C PRINT SUMMARY
COMMON/SCOM1/ATTRIB(100),DD(100),DDL(100),DTNOW,II,MFA,MSTOP,
$NCLNR,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100),TNEXT,TNOW,
$XX(100)
COMMON/UCOM1/NCTYPE,NNODE,LINKN(100,8),NARC,ASIZE(100),
1LOCATA(100,2),SPEED,USIZE,NSAREA,NSVR(25),LOCATS(25,11),
2NPART,NVISIT,NROUTE(25,25),NEPT,ARRNT(25),XMEAN(25),ULO(25),
3UHI(25),LOCATP(25),NCAP(100),
4TTIME(100),ISHORT(200,100),DIST(100,100),IPATH(100),
5NPATH(100),CAP(100),NVST(25),IBLOCK(25),MMST(100),TSTOP(100),
6NRECIR,DISTLD,TMLOAD(100),CC(100),TOTAL,TOTPR
7,XLO(25),XMODE(25),XMN(25),STD(25),ARINT,AARRNT(25),NMODE(25),
8XXMEAN(25),UULO(25),XXLO(25),XXHI(25),XXMODE(25),XXMN(25)
9,SSTD(25),MODE(25),XHI(25),UUHI(25)
COMMON/UCOM2/SAMP(100,5),TSAMP(100,6),TIMCLR,TIMSTP
COMMON/UCOM3/MAXPRT(25)
WRITE(6,180)
DO 10 I=1,NARC
CALL CCLCT(1,A,I)
10 CONTINUE
WRITE(6,190)
DO 20 I=1,NARC
CALL MMSTT(1,TNOW,CC(I),I)
20 CONTINUE
MSTOP=-1
RETURN
180 FORMAT(///,32X,'**STATISTICS FOR VARIABLES BASED ON OBSERVATION
$**',///,22X,'MEAN',9X,'STANDARD',6X,'COEFF.OF',5X,'MINIMUM',7X,'MA
$XIMUM',6X,'NUMBER OF',/,22X,'VALUE',8X,'DEVIATION',5X,'VARIATION',
$5X,'VALUE',9X,'VALUE',8X,'OBSERVATIONS',/)
190 FORMAT(///,32X,'**STATISTICS FOR TIME-PERSISTENT VARIABLES
$**',///,22X,'MEAN',9X,'STANDARD',6X,'MINIMUM',7X,'MAXIMUM',7X,'TIME
$',10X,'CURRENT',/,22X,'VALUE',8X,'DEVIATION',5X,'VALUE',9X,'VALUE'
$,9X,'INTERVAL',6X,'VALUE',/)
END

```

```

SUBROUTINE CCLCT(IMODE,X,ISTRM)
C   COLLECT STATISTICS FOR VARIABLES BASED ON OBSERVATION
COMMON/SCOM1/ATRIB(100),DD(100),DDL(100),DTNOW,II,MFA,MSTOP,
$NCLNR,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100),TNEXT,TNOW,
$XX(100)
COMMON/UCOM1/NCTYPE,NNODE,LINKN(100,8),NARC,ASIZE(100),
1LOCATA(100,2),SPEED,USIZE,NSAREA,NSERV(25),LOCATS(25,11),
2NPART,NVISIT,NROUTE(25,25),NEPT,ARRNT(25),XMEAN(25),ULO(25),
3UHI(25),LOCATP(25),NCAP(100),
4TTIME(100),ISHORT(200,100),DIST(100,100),IPATH(100),
5NPAT(100),CAP(100),NVST(25),IBLOCK(25),MMST(100),TSTOP(100),
6NRECIR,DISTLD,TMLOAD(100),CC(100),TOTAL,TOTPR
7,XLO(25),XMODE(25),XMN(25),STD(25),ARINT,AARRNT(25),NMODE(25),
8XXMEAN(25),UULO(25),XXLO(25),XXHI(25),XXMODE(25),XXMN(25)
9,SSTD(25),MODE(25),XHI(25),UUHI(25)
COMMON/UCOM2/SAMP(100,5),TSAMP(100,6),TIMCLR,TIMSTP
COMMON/UCOM3/MAXPRT(25)

C   IF(IMODE)10,20,30
C   INITIALIZE
10 DO 15 I=1,ISTRM
DO 14 J=1,3
14 SAMP(I,J)=0.
SAMP(I,4)=0.09E75
SAMP(I,5)=-0.9E75
15 CONTINUE
RETURN
C   UPDATE
20 SAMP(ISTRM,1)=SAMP(ISTRM,1)+1.
SAMP(ISTRM,2)=SAMP(ISTRM,2)+X
SAMP(ISTRM,3)=SAMP(ISTRM,3)+X**2
IF(X.LT.SAMP(ISTRM,4)) SAMP(ISTRM,4)=X
IF(X.GT.SAMP(ISTRM,5)) SAMP(ISTRM,5)=X
RETURN
C   SUMMARIZE
30 IF(SAMP(ISTRM,1).LE.0.) GO TO 100
AVE=SAMP(ISTRM,2)/SAMP(ISTRM,1)
VAR=SAMP(ISTRM,3)-(SAMP(ISTRM,2)**2/SAMP(ISTRM,1))
IF(SAMP(ISTRM,1).LE.1.) VAR=0.0
IF(SAMP(ISTRM,1).GT.1.) VAR=VAR/(SAMP(ISTRM,1)-1.)
SSD=SQRT(ABS(VAR))
CV=9999.
IF(AVE.NE.0.) CV=SSD/AVE
NOB=SAMP(ISTRM,1)+0.5
WRITE(6,200) ISTRM,AVE,SSD,CV,SAMP(ISTRM,4),SAMP(ISTRM,5),NOB
GO TO 160
100 WRITE(6,190) ISTRM
160 CONTINUE
RETURN
190 FORMAT(1X,'TIME ON CONVEYOR',I3,17X,'NO VALUES RECORDED')
200 FORMAT(1X,'TIME ON CONVEYOR',I3,E12.4,4(2X,E12.4),2X,I10)
END

```

```

SUBROUTINE MMSTT(IMODE,T,QTY,ISTRM)
C   COLLECT STATISTICS FOR TIME-PERSISTENT VARIABLES
COMMON/SCOM1/ATTRIB(100),DD(100),DDL(100),DTNOW,II,MFA,MSTOP,
$NCLNR,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100),TNEXT,TNOW,
$XX(100)
COMMON/UCOM1/NCTYPE,NNODE,LINKN(100,8),NARC,ASIZE(100),
1LOCATA(100,2),SPEED,USIZE,NSAREA,NSERV(25),LOCATS(25,11),
2NPART,NVISIT,NROUTE(25,25),NEPT,ARRNT(25),XMEAN(25),ULO(25),
3UHI(25),LOCATP(25),NCAP(100),
4TTIME(100),ISHORT(200,100),DIST(100,100),IPATH(100),
5NPATN(100),CAP(100),NVST(25),IBLOCK(25),MMST(100),TSTOP(100),
6NRECIR,DISTLD,TMLOAD(100),CC(100),TOTAL,TOTPR
7,XLO(25),XMODE(25),XMN(25),STD(25),ARINT,AARRNT(25),NMODE(25),
8XXMEAN(25),UULO(25),XXLO(25),XXHI(25),XXMODE(25),XXMN(25)
9,SSTD(25),MODE(25),XHI(25),UUHI(25)
COMMON/UCOM2/SAMP(100,5),TSAMP(100,6),TIMCLR,TIMSTP
COMMON/UCOM3/MAXPRT(25)

C   IF(IMODE)10,20,30
C   INITIALIZE
10 DO 15 I=1,ISTRM
DO 14 J=1,4
14 SAMP(I,J)=0.
TSAMP(I,5)=0.09E75
TSAMP(I,6)=-0.9E75
15 CONTINUE
RETURN
C   UPDATE
20 TT=T-TSAMP(ISTRM,2)
TSAMP(ISTRM,3)=TSAMP(ISTRM,3)+TSAMP(ISTRM,1)*TT
TSAMP(ISTRM,4)=TSAMP(ISTRM,4)+TSAMP(ISTRM,1)**2*TT
TSAMP(ISTRM,1)=QTY
TSAMP(ISTRM,2)=T
IF(QTY.LT.TSAMP(ISTRM,5)) TSAMP(ISTRM,5)=QTY
IF(QTY.GT.TSAMP(ISTRM,6)) TSAMP(ISTRM,6)=QTY
RETURN
C   SUMMARIZE
30 IF(TSAMP(ISTRM,6).LE.0.) GO TO 100
TSUM=TSAMP(ISTRM,3)+TSAMP(ISTRM,1)*(T-TSAMP(ISTRM,2))
SSDT=TSAMP(ISTRM,4)+TSAMP(ISTRM,1)**2*(T-TSAMP(ISTRM,2))
AVE=TSUM/(T-TIMCLR)
SVAR=(SSDT/(T-TIMCLR))-(AVE**2)
SSD=SIGN(SQRT(ABS(SVAR)),SVAR)
QTYMX=TSAMP(ISTRM,5)
QTYMN=TSAMP(ISTRM,6)
TS=T-TIMCLR
WRITE(6,200) ISTRM,AVE,SSD,TSAMP(ISTRM,5),TSAMP(ISTRM,6),TS,QTY
GO TO 160
100 WRITE(6,190) ISTRM
160 CONTINUE
RETURN
190 FORMAT(1X,'NO. ON CONVEYOR',I3,18X,'NO VALUES RECORDED')
200 FORMAT(1X,'NO. ON CONVEYOR',I3,E12.4,5(2X,E12.4))
END

```

```

C      THIS SUBROUTINE FINDS THE SHORTEST PATHS BETWEEN SERVICE STATIONS
C
      SUBROUTINE PATH(NTHIS,NXTNOD,NODPTH)
      COMMON/SCOM1/ATTRIB(100),DD(100),DDL(100),DTNOW,II,MFA,MSTOP,
      $NCLNR,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100),TNEXT,TNOW,
      $XX(100)
      COMMON/UCOM1/NCTYPE,NNODE,LINKN(100,8),NARC,ASIZE(100),
      1LOCATA(100,2),SPEED,USIZE,NSAREA,NSERV(25),LOCATS(25,11),
      2NPART,NVISIT,NROUTE(25,25),NEPT,ARRNT(25),XMEAN(25),ULO(25),
      3UHI(25),LOCATP(25),NCAP(100),
      4TTIME(100),ISHORT(200,100),DIST(100,100),IPATH(100),
      5NPATH(100),CAP(100),NVST(25),IBLOCK(25),MMST(100),TSTOP(100),
      6NRECIR,DISTLD,TMLOAD(100),CC(100),TOTAL,TOTPRT
      7,XLO(25),XMODE(25),XMN(25),STD(25),ARINT,AARRNT(25),NMODE(25),
      8XXMEAN(25),UULO(25),XXLO(25),XXHI(25),XXMODE(25),XXMN(25)
      9,SSTD(25),MODE(25),XHI(25),UUHI(25)
C      LOCATE THE PORTION OF THE SHORTEST PATH MATRIX THAT STORES SHORTEST PATHS
C      FROM NODE 'NTHIS' TO ALL OTHER NODES.
      MNODE=2*NNODE
      LT=(2*NTHIS)-1
      LP=2*NTHIS
C      FIND THE POSITION OF NODE 'NXTNOD' IN THE STRING OF PATHS
      J=2
10     IF(ISHORT(LT,J).EQ.NXTNOD)GO TO 11
      J=J+1
      GO TO 10
C      NODE 'NXTNOD' IS IN POSITION J OF THE PATH STRING RELATIVE TO NODE 'NTHIS'
11     IX=0
12     IX=IX + 1
      IPATH(IX)=ISHORT(LT,J)
      IPREC=ISHORT(LP,J)
13     J=J-1
      IF(J.EQ.0)GO TO 14
      IF(ISHORT(LT,J).EQ.IPREC)GO TO 12
      GO TO 13
C      THE SHORTEST PATH HAS BEEN FULLY TRACED.
14     K=1
      NODPTH=IX
15     NPATH(K)=IPATH(IX)
      IX=IX-1
      IF(IX.EQ.0)GO TO 16
      K=K+1
      GO TO 15
16     RETURN
      END

```

APPENDIX B

EXAMPLES

EXAMPLE 1

SLAM INPUT

```
GEN, ASIF SHAIKH, EXAMPLE, 1/13/84, 1;  
LIM, 11, 14, 300;  
TIMST, XX(1), STATION 1 BUSY;  
TIMST, XX(2), STATION 2 BUSY;  
TIMST, XX(3), STATION 3 BUSY;  
TIMST, XX(4), STATION 4 BUSY;  
TIMST, XX(5), STATION 5 BUSY;  
TIMST, XX(6), STATION 6 BUSY;  
TIMST, XX(7), STATION 7 BUSY;  
TIMST, XX(8), STATION 8 BUSY;  
STAT, 1, JOB1 IN SYSTEM;  
STAT, 2, JOB2 IN SYSTEM;  
STAT, 3, JOB3 IN SYSTEM;  
SIM;
```

MODEL INPUT

```

1
0
9
0 0 0 0 1 2 0 0
1 1 0 0 3 3 4 5
1 2 0 0 1 6 0 0
1 2 0 0 2 6 7 0
1 2 0 0 1 8 0 0
2 3 4 0 1 9 0 0
1 4 0 0 1 9 0 0
1 5 0 0 1 9 0 0
3 6 7 8 0 0 0 0
11
5. 3. 4. 6. 3. 5. 4. 4. 3. 3. 3.
1 2
2 3
2 4
2 5
3 6
4 6
4 7
5 8
6 9
7 9
8 9
5.
1.5
8
1 1 1 1 2 1 1 2
1
0.
1
5.
1
10.
1
7.7
1
12.
1
10.5
1
8.5
1
0.
1 1 0 0 0 0 1 0 0 0 0
1 2 0 0 0 0 1 0 0 0 0
1 3 0 0 0 0 1 0 0 0 0

```


1 5 0 0 0 0 1 0 0 0 0
1 6 0 0 0 0 2 0 0 0 0
1 7 0 0 0 0 1 0 0 0 0
1 8 0 0 0 0 1 0 0 0 0
1 9 0 0 0 0 2 0 0 0 0
3
4
1 2 5 8
4
1 4 7 8
3
1 6 8
1
1
1
1
1
1
1
1
5.
1
10.
1
7.
100 75 80
200.
3000.

SLAM ECHO REPORT

SIMULATION PROJECT EXAMPLE

BY ASIF SHAIKH

DATE 1/13/1984

RUN NUMBER 1 OF 1

SLAM VERSION JAN 79

GENERAL OPTIONS

| | |
|---|-----|
| PRINT INPUT STATEMENTS (ILIST): | YES |
| PRINT ECHO REPORT (IECHO): | YES |
| EXECUTE SIMULATIONS (IXQT): | YES |
| PRINT INTERMEDIATE RESULTS HEADING (IPIRH): | YES |
| PRINT SUMMARY REPORT (ISMRY): | YES |

LIMITS ON FILES

| | |
|---|-----|
| MAXIMUM NUMBER OF USER FILES (MFILS): | 11 |
| MAXIMUM NUMBER OF USER ATTRIBUTES (MATR): | 14 |
| MAXIMUM NUMBER OF CONCURRENT ENTRIES (MNTRY): | 300 |

FILE SUMMARY

| FILE NUMBER | INITIAL ENTRIES | RANKING CRITERION |
|----------------|--------------------|----------------------|
| 1 | 0 | FIFO |
| 2 | 0 | FIFO |
| 3 | 0 | FIFO |
| 4 | 0 | FIFO |
| 5 | 0 | FIFO |
| 6 | 0 | FIFO |
| 7 | 0 | FIFO |
| 8 | 0 | FIFO |
| 9 | 0 | FIFO |
| 10 | 0 | FIFO |
| 11 | 0 | FIFO |

STATISTICS BASED ON OBSERVATIONS

| COLCT NUMBER | COLLECTION MODE | IDENTIFIER | HISTOGRAM SPECIFICATIONS NCEL HLOW HWID |
|-----------------|--------------------|----------------|--|
| 1 | EVENT | JOB1 IN SYSTEM | |
| 2 | EVENT | JOB2 IN SYSTEM | |
| 3 | EVENT | JOB3 IN SYSTEM | |

STATISTICS FOR TIME PERSISTENT VARIABLES

| TIMST NUMBER | VARIABLE | IDENTIFIER | INITIAL VALUE |
|-----------------|----------|----------------|------------------|
| 1 | XX(1) | STATION 1 BUSY | 0.0 |
| 2 | XX(2) | STATION 2 BUSY | 0.0 |
| 3 | XX(3) | STATION 3 BUSY | 0.0 |
| 4 | XX(4) | STATION 4 BUSY | 0.0 |
| 5 | XX(5) | STATION 5 BUSY | 0.0 |
| 6 | XX(6) | STATION 6 BUSY | 0.0 |
| 7 | XX(7) | STATION 7 BUSY | 0.0 |
| 8 | XX(8) | STATION 8 BUSY | 0.0 |

RANDOM NUMBER STREAMS

| STREAM NUMBER | SEED VALUE | REINITIALIZATION OF STREAM |
|------------------|---------------|-------------------------------|
| 1 | 1274321477 | NO |
| 2 | 2135124613 | NO |
| 3 | 1743251541 | NO |
| 4 | 1624217675 | NO |
| 5 | 2014632579 | NO |
| 6 | 2036774231 | NO |
| 7 | 1452313571 | NO |
| 8 | 1254240657 | NO |
| 9 | 1410143363 | NO |
| 10 | 2135621895 | NO |

INITIALIZATION OPTIONS

| | |
|---------------------------------------|------------|
| BEGINNING TIME OF SIMULATION (TTBEG): | 0.0 |
| ENDING TIME OF SIMULATION (TTFIN): | 0.1000E+21 |
| STATISTICAL ARRAYS CLEARED (JJCLR): | YES |
| VARIABLES INITIALIZED (JJVAR): | YES |
| FILES INITIALIZED (JJFIL): | YES |

NSET/QSET STORAGE ALLOCATION

| | |
|-----------------------------------|-------|
| DIMENSION OF NSET/QSET (NNSET): | 10000 |
| WORDS ALLOCATED TO FILING SYSTEM: | 5400 |
| WORDS ALLOCATED TO NETWORK: | 0 |
| WORDS AVAILABLE FOR PLOTS/TABLES: | 4600 |

INPUT ERRORS DETECTED: 0

EXECUTION WILL BE ATTEMPTED

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

SIMULATION PROJECT EXAMPLE

BY ASIF SHAIKH

DATE 1/13/1984

RUN NUMBER 1 OF 1

CURRENT TIME 0.8502E+03

STATISTICAL ARRAYS CLEARED AT TIME 0.1000E+03

STATISTICS FOR VARIABLES BASED ON OBSERVATION

| | MEAN VALUE | STANDARD DEVIATION | COEFF. OF VARIATION | MINIMUM VALUE | MAXIMUM VALUE | NUMBER OF OBSERVATIONS |
|----------------|---------------|-----------------------|------------------------|------------------|------------------|---------------------------|
| JOB1 IN SYSTEM | 0.1521E+03 | 0.9311E+02 | 0.6121E+00 | 0.2590E+02 | 0.2679E+03 | 87 |
| JOB2 IN SYSTEM | 0.9151E+02 | 0.9240E+02 | 0.1010E+01 | 0.1980E+02 | 0.2954E+03 | 67 |
| JOB3 IN SYSTEM | 0.1659E+03 | 0.1006E+03 | 0.6060E+00 | 0.4170E+02 | 0.3042E+03 | 72 |

STATISTICS FOR TIME-PERSISTENT VARIABLES

| | MEAN VALUE | STANDARD DEVIATION | MINIMUM VALUE | MAXIMUM VALUE | TIME INTERVAL | CURRENT VALUE |
|----------------|---------------|-----------------------|------------------|------------------|------------------|------------------|
| STATION 1 BUSY | 0.0 | 0.0 | 0.0 | 0.0 | 0.7502E+03 | 0.0 |
| STATION 2 BUSY | 0.5412E+00 | 0.4983E+00 | 0.0 | 0.1000E+01 | 0.7502E+03 | 0.0 |
| STATION 3 BUSY | 0.0 | 0.0 | 0.0 | 0.0 | 0.7502E+03 | 0.0 |
| STATION 4 BUSY | 0.6774E+00 | 0.4675E+00 | 0.0 | 0.1000E+01 | 0.7502E+03 | 0.0 |
| STATION 5 BUSY | 0.1372E+01 | 0.7991E+00 | 0.0 | 0.2000E+01 | 0.7502E+03 | 0.0 |
| STATION 6 BUSY | 0.9992E+00 | 0.2827E-01 | 0.0 | 0.1000E+01 | 0.7502E+03 | 0.0 |
| STATION 7 BUSY | 0.7591E+00 | 0.4276E+00 | 0.0 | 0.1000E+01 | 0.7502E+03 | 0.0 |
| STATION 8 BUSY | 0.0 | 0.0 | 0.0 | 0.0 | 0.7502E+03 | 0.0 |

***FILE STATISTICS**

| FILE NUMBER | ASSOCIATED NODE TYPE | AVERAGE LENGTH | STANDARD DEVIATION | MAXIMUM LENGTH | CURRENT LENGTH | AVERAGE WAITING TIME |
|----------------|-------------------------|-------------------|-----------------------|-------------------|-------------------|-------------------------|
| 1 | | 30.9309 | 20.3504 | 61 | 0 | 114.8729 |
| 2 | | 0.0 | 0.0 | 0 | 0 | 0.0 |
| 3 | | 0.0 | 0.0 | 0 | 0 | 0.0 |
| 4 | | 0.0 | 0.0 | 0 | 0 | 0.0 |
| 5 | | 1.8098 | 1.1553 | 3 | 0 | 7.2994 |
| 6 | | 0.0413 | 0.1990 | 2 | 0 | 2.3845 |
| 7 | | 0.5373 | 0.8138 | 4 | 0 | 9.1610 |
| 8 | | 0.4367 | 0.7250 | 2 | 0 | 6.3000 |
| 9 | | 1.8828 | 0.3647 | 2 | 0 | 19.8945 |
| 10 | | 0.5253 | 0.6753 | 2 | 0 | 7.5791 |
| 11 | | 0.0 | 0.0 | 0 | 0 | 0.0 |
| 12 | | 8.3085 | 2.6177 | 18 | 1 | 3.1608 |

INTERMEDIATE RESULTS

**STATISTICS FOR VARIABLES BASED ON OBSERVATION **

| | | MEAN VALUE | STANDARD DEVIATION | COEFF.OF VARIATION | MINIMUM VALUE | MAXIMUM VALUE | NUMBER OF OBSERVATIONS |
|------------------|----|--------------------|-----------------------|-----------------------|------------------|------------------|---------------------------|
| TIME ON CONVEYOR | 1 | 0.7349E+01 | 0.4985E+01 | 0.6784E+00 | 0.0 | 0.2150E+02 | 214 |
| TIME ON CONVEYOR | 2 | 0.1014E+01 | 0.1284E+01 | 0.1266E+01 | 0.0 | 0.7200E+01 | 83 |
| TIME ON CONVEYOR | 3 | 0.2029E+02 | 0.1953E+01 | 0.9627E-01 | 0.0 | 0.2100E+02 | 69 |
| TIME ON CONVEYOR | 4 | 0.7307E+01 | 0.6344E+01 | 0.8682E+00 | 0.0 | 0.2500E+02 | 66 |
| TIME ON CONVEYOR | 5 | 0.4612E+01 | 0.4541E+01 | 0.9848E+00 | 0.0 | 0.1200E+02 | 85 |
| TIME ON CONVEYOR | 6 | NO VALUES RECORDED | | | | | |
| TIME ON CONVEYOR | 7 | 0.2100E+02 | 0.1494E-01 | 0.7114E-03 | 0.0 | 0.2100E+02 | 71 |
| TIME ON CONVEYOR | 8 | 0.6682E+01 | 0.5135E+01 | 0.7684E+00 | 0.0 | 0.1700E+02 | 67 |
| TIME ON CONVEYOR | 9 | 0.5999E+00 | 0.1836E-02 | 0.3061E-02 | 0.0 | 0.6000E+00 | 87 |
| TIME ON CONVEYOR | 10 | 0.5999E+00 | 0.1606E-02 | 0.2677E-02 | 0.0 | 0.6000E+00 | 72 |
| TIME ON CONVEYOR | 11 | 0.5999E+00 | 0.1521E-02 | 0.2535E-02 | 0.0 | 0.6000E+00 | 67 |

**STATISTICS FOR TIME-PERSISTENT VARIABLES **

| | | MEAN VALUE | STANDARD DEVIATION | MINIMUM VALUE | MAXIMUM VALUE | TIME INTERVAL | CURRENT VALUE |
|-----------------|----|--------------------|-----------------------|------------------|------------------|------------------|------------------|
| NO. ON CONVEYOR | 1 | 0.2153E+01 | 0.9410E+00 | 0.0 | 0.3000E+01 | 0.7502E+03 | 0.0 |
| NO. ON CONVEYOR | 2 | 0.1405E+00 | 0.3513E+00 | 0.0 | 0.2000E+01 | 0.7502E+03 | 0.0 |
| NO. ON CONVEYOR | 3 | 0.1930E+01 | 0.1550E+00 | 0.0 | 0.2000E+01 | 0.7502E+03 | 0.0 |
| NO. ON CONVEYOR | 4 | 0.6573E+00 | 0.8655E+00 | 0.0 | 0.4000E+01 | 0.7502E+03 | 0.0 |
| NO. ON CONVEYOR | 5 | 0.6601E+00 | 0.8139E+00 | 0.0 | 0.2000E+01 | 0.7502E+03 | 0.0 |
| NO. ON CONVEYOR | 6 | NO VALUES RECORDED | | | | | |
| NO. ON CONVEYOR | 7 | 0.2148E+01 | 0.5996E+00 | 0.0 | 0.2000E+01 | 0.7502E+03 | 0.0 |
| NO. ON CONVEYOR | 8 | 0.6053E+00 | 0.7070E+00 | 0.0 | 0.2000E+01 | 0.7502E+03 | 0.0 |
| NO. ON CONVEYOR | 9 | 0.7997E-01 | 0.2712E+00 | 0.0 | 0.1000E+01 | 0.7502E+03 | 0.0 |
| NO. ON CONVEYOR | 10 | 0.6397E-01 | 0.2447E+00 | 0.0 | 0.1000E+01 | 0.7502E+03 | 0.0 |
| NO. ON CONVEYOR | 11 | 0.5997E-01 | 0.2374E+00 | 0.0 | 0.1000E+01 | 0.7502E+03 | 0.0 |

EXAMPLE 2

SLAM INPUT

```
GEN,ASIF SHAIKH,EXAMPLE,1/13/84,1;
LIM,11,14,300;
TIMST,XX(1),STATION 1 BUSY;
TIMST,XX(2),STATION 2 BUSY;
TIMST,XX(3),STATION 3 BUSY;
TIMST,XX(4),STATION 4 BUSY;
TIMST,XX(5),STATION 5 BUSY;
TIMST,XX(6),STATION 6 BUSY;
TIMST,XX(7),STATION 7 BUSY;
TIMST,XX(8),STATION 8 BUSY;
STAT,1,JOB1 IN SYSTEM;
STAT,2,JOB2 IN SYSTEM;
STAT,3,JOB3 IN SYSTEM;
SIM;
```


MODEL INPUT

```

2
0
9
0 0 0 0 1 2 0 0
1 1 0 0 3 3 4 5
1 2 0 0 1 6 0 0
1 2 0 0 2 6 7 0
1 2 0 0 1 8 0 0
2 3 4 0 1 9 0 0
1 4 0 0 1 9 0 0
1 5 0 0 1 9 0 0
3 6 7 8 0 0 0 0
11
5. 3. 4. 6. 3. 5. 4. 4. 3. 3. 3.
1 2
2 3
2 4
2 5
3 6
4 6
4 7
5 8
6 9
7 9
8 9
5.
1.5
8
1 1 1 1 2 1 1 2
1
0.
1
5.
1
10.
1
7.7
1
12.
1
10.5
1
8.5
1
0.
1 1 0 0 0 0 1 0 0 0 0
1 2 0 0 0 0 1 0 0 0 0
1 3 0 0 0 0 1 0 0 0 0

```

1 5 0 0 0 0 1 0 0 0 0
1 6 0 0 0 0 2 0 0 0 0
1 7 0 0 0 0 1 0 0 0 0
1 8 0 0 0 0 1 0 0 0 0
1 9 0 0 0 0 2 0 0 0 0
3
4
1 2 5 8
4
1 4 7 8
3
1 6 8
1
1
1
1
1
1
1
1
5.
1
10.
1
7.
100 75 80
200.
3000.

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

SIMULATION PROJECT EXAMPLE

BY ASIF SHAIKH

DATE 1/13/1984

RUN NUMBER 1 OF 1

CURRENT TIME 0.8670E+03

STATISTICAL ARRAYS CLEARED AT TIME 0.1000E+03

STATISTICS FOR VARIABLES BASED ON OBSERVATION

| | MEAN VALUE | STANDARD DEVIATION | COEFF. OF VARIATION | MINIMUM VALUE | MAXIMUM VALUE | NUMBER OF OBSERVATIONS |
|----------------|---------------|-----------------------|------------------------|------------------|------------------|---------------------------|
| JOB1 IN SYSTEM | 0.1406E+03 | 0.1047E+03 | 0.7445E+00 | 0.2693E+02 | 0.3173E+03 | 87 |
| JOB2 IN SYSTEM | 0.1211E+03 | 0.1118E+03 | 0.9232E+00 | 0.2180E+02 | 0.3545E+03 | 68 |
| JOB3 IN SYSTEM | 0.1690E+03 | 0.1069E+03 | 0.6325E+00 | 0.4343E+02 | 0.3094E+03 | 72 |

STATISTICS FOR TIME-PERSISTENT VARIABLES

| | MEAN VALUE | STANDARD DEVIATION | MINIMUM VALUE | MAXIMUM VALUE | TIME INTERVAL | CURRENT VALUE |
|----------------|---------------|-----------------------|------------------|------------------|------------------|------------------|
| STATION 1 BUSY | 0.0 | 0.0 | 0.0 | 0.0 | 0.7670E+03 | 0.0 |
| STATION 2 BUSY | 0.5302E+00 | 0.4991E+00 | 0.0 | 0.1000E+01 | 0.7670E+03 | 0.0 |
| STATION 3 BUSY | 0.0 | 0.0 | 0.0 | 0.0 | 0.7670E+03 | 0.0 |
| STATION 4 BUSY | 0.6644E+00 | 0.4722E+00 | 0.0 | 0.1000E+01 | 0.7670E+03 | 0.0 |
| STATION 5 BUSY | 0.1346E+01 | 0.7700E+00 | 0.0 | 0.2000E+01 | 0.7670E+03 | 0.0 |
| STATION 6 BUSY | 0.9796E+00 | 0.1413E+00 | 0.0 | 0.1000E+01 | 0.7670E+03 | 0.0 |
| STATION 7 BUSY | 0.7441E+00 | 0.4364E+00 | 0.0 | 0.1000E+01 | 0.7670E+03 | 0.0 |
| STATION 8 BUSY | 0.0 | 0.0 | 0.0 | 0.0 | 0.7670E+03 | 0.0 |

FILE STATISTICS

| FILE NUMBER | ASSOCIATED NODE TYPE | AVERAGE LENGTH | STANDARD DEVIATION | MAXIMUM LENGTH | CURRENT LENGTH | AVERAGE WAITING TIME |
|----------------|-------------------------|-------------------|-----------------------|-------------------|-------------------|-------------------------|
| 1 | | 31.6658 | 20.1825 | 65 | 0 | 115.6505 |
| 2 | | 1.8315 | 1.3132 | 6 | 0 | 5.1266 |
| 3 | | 0.0 | 0.0 | 0 | 0 | 0.0 |
| 4 | | 0.0 | 0.0 | 0 | 0 | 0.0 |
| 5 | | 1.7204 | 1.1761 | 3 | 0 | 7.2499 |
| 6 | | 0.0163 | 0.1266 | 1 | 0 | 4.1667 |
| 7 | | 0.7538 | 1.1981 | 4 | 0 | 13.7659 |
| 8 | | 0.1930 | 0.4765 | 2 | 0 | 3.7949 |
| 9 | | 1.6959 | 0.5998 | 2 | 0 | 18.5810 |
| 10 | | 0.5280 | 0.7578 | 2 | 0 | 8.4369 |
| 11 | | 0.0 | 0.0 | 0 | 0 | 0.0 |
| 12 | | 8.6412 | 2.5920 | 17 | 1 | 3.3540 |

INTERMEDIATE RESULTS

**STATISTICS FOR VARIABLES BASED ON OBSERVATION **

| | | | MEAN VALUE | STANDARD DEVIATION | COEFF.OF VARIATION | MINIMUM VALUE | MA XIMUM VALUE | NUMBER OF OBSERVATIONS |
|------------------|----|--|---------------|-----------------------|-----------------------|------------------|-------------------|---------------------------|
| TIME ON CONVEYOR | 1 | | 0.7834E+01 | 0.5404E+01 | 0.6898E+00 | 0.0 | 0.3437E+02 | 214 |
| TIME ON CONVEYOR | 2 | | 0.1179E+01 | 0.9839E+00 | 0.8347E+00 | 0.0 | 0.8000E+01 | 83 |
| TIME ON CONVEYOR | 3 | | 0.1879E+02 | 0.5822E+01 | 0.3100E+00 | 0.0 | 0.2100E+02 | 69 |
| TIME ON CONVEYOR | 4 | | 0.1076E+02 | 0.1030E+02 | 0.9570E+00 | 0.0 | 0.3353E+02 | 66 |
| TIME ON CONVEYOR | 5 | | 0.2855E+01 | 0.3055E+01 | 0.1070E+01 | 0.0 | 0.1200E+02 | 85 |
| TIME ON CONVEYOR | 6 | | | NO VALUES RECORDED | | | | |
| TIME ON CONVEYOR | 7 | | 0.1989E+02 | 0.4038E+01 | 0.2030E+00 | 0.0 | 0.2100E+02 | 71 |
| TIME ON CONVEYOR | 8 | | 0.7378E+01 | 0.6204E+01 | 0.8409E+00 | 0.0 | 0.1700E+02 | 67 |
| TIME ON CONVEYOR | 9 | | 0.5999E+00 | 0.1836E-02 | 0.3061E-02 | 0.0 | 0.6000E+00 | 87 |
| TIME ON CONVEYOR | 10 | | 0.5999E+00 | 0.1606E-02 | 0.2677E-02 | 0.0 | 0.6000E+00 | 72 |
| TIME ON CONVEYOR | 11 | | 0.5999E+00 | 0.1509E-02 | 0.2516E-02 | 0.0 | 0.6000E+00 | 68 |

**STATISTICS FOR TIME-PERSISTENT VARIABLES **

| | | | MEAN VALUE | STANDARD DEVIATION | MINIMUM VALUE | MAXIMUM VALUE | TIME INTERVAL | CURRENT VALUE |
|-----------------|----|--|---------------|-----------------------|------------------|------------------|------------------|------------------|
| NO. ON CONVEYOR | 1 | | 0.2278E+01 | 0.8251E+00 | 0.0 | 0.3000E+01 | 0.7670E+03 | 0.0 |
| NO. ON CONVEYOR | 2 | | 0.1667E+00 | 0.3796E+00 | 0.0 | 0.2000E+01 | 0.7670E+03 | 0.0 |
| NO. ON CONVEYOR | 3 | | 0.1763E+01 | 0.5684E+00 | 0.0 | 0.2000E+01 | 0.7670E+03 | 0.0 |
| NO. ON CONVEYOR | 4 | | 0.9494E+00 | 0.1258E+01 | 0.0 | 0.4000E+01 | 0.7670E+03 | 0.0 |
| NO. ON CONVEYOR | 5 | | 0.4572E+00 | 0.6704E+00 | 0.0 | 0.2000E+01 | 0.7670E+03 | 0.0 |
| NO. ON CONVEYOR | 6 | | | NO VALUES RECORDED | | | | |
| NO. ON CONVEYOR | 7 | | 0.2003E+01 | 0.2877E+00 | 0.0 | 0.2000E+01 | 0.7670E+03 | 0.0 |
| NO. ON CONVEYOR | 8 | | 0.6584E+00 | 0.7941E+00 | 0.0 | 0.2000E+01 | 0.7670E+03 | 0.0 |
| NO. ON CONVEYOR | 9 | | 0.7822E-01 | 0.2685E+00 | 0.0 | 0.1000E+01 | 0.7670E+03 | 0.0 |
| NO. ON CONVEYOR | 10 | | 0.6257E-01 | 0.2422E+00 | 0.0 | 0.1000E+01 | 0.7670E+03 | 0.0 |
| NO. ON CONVEYOR | 11 | | 0.5866E-01 | 0.2350E+00 | 0.0 | 0.1000E+01 | 0.7670E+03 | 0.0 |

EXAMPLE 3

SLAM INPUT

```
GEN,ASIF SHAIKH,EXAMPLE,1/13/84,1;  
LIM,8,14,300;  
TIMST,XX(1),STATION 1 BUSY;  
TIMST,XX(2),STATION 2 BUSY;  
TIMST,XX(3),STATION 3 BUSY;  
TIMST,XX(4),STATION 4 BUSY;  
TIMST,XX(5),STATION 5 BUSY;  
STAT,1,JOB1 IN SYSTEM;  
STAT,2,JOB2 IN SYSTEM;  
STAT,3,JOB3 IN SYSTEM;  
SIM;
```

MODEL INPUT

```

3
0
3.
5
0 0 0 0 1 2 0 0
1 1 0 0 1 3 0 0
1 2 0 0 1 4 0 0
1 3 0 0 1 5 0 0
1 4 0 0 0 0 0 0
4
5. 10. 10. 5.
1 2
2 3
3 4
4 5
5.
1.
5
1 1 1 2 1
1
0.
1
3.
1
5.
1
7.
1
2.
1 1 0 0 0 0 1 0 0 0 0
1 2 0 0 0 0 1 0 0 0 0
1 3 0 0 0 0 1 0 0 0 0
1 4 0 0 0 0 2 0 0 0 0
1 5 0 0 0 0 1 0 0 0 0
3
3
1 2 5
4
1 2 3 5
5
1 2 3 4 5
1
1
1
1
1
1
1
1

```

5.
1
10.
1
7.
50 30 40
0.
1000.

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

SIMULATION PROJECT EXAMPLE

BY ASIF SHAIKH

DATE 1/13/1984

RUN NUMBER 1 OF 1

CURRENT TIME 0.4336E+03

STATISTICAL ARRAYS CLEARED AT TIME 0.0

STATISTICS FOR VARIABLES BASED ON OBSERVATION

| | MEAN VALUE | STANDARD DEVIATION | COEFF. OF VARIATION | MINIMUM VALUE | MAXIMUM VALUE | NUMBER OF OBSERVATIONS |
|----------------|---------------|-----------------------|------------------------|------------------|------------------|---------------------------|
| JOB1 IN SYSTEM | 0.7282E+02 | 0.3489E+02 | 0.4791E+00 | 0.1240E+02 | 0.1302E+03 | 50 |
| JOB2 IN SYSTEM | 0.8652E+02 | 0.3937E+02 | 0.4551E+00 | 0.2060E+02 | 0.1386E+03 | 30 |
| JOB3 IN SYSTEM | 0.9091E+02 | 0.3818E+02 | 0.4200E+00 | 0.2560E+02 | 0.1506E+03 | 40 |

STATISTICS FOR TIME-PERSISTENT VARIABLES

| | MEAN VALUE | STANDARD DEVIATION | MINIMUM VALUE | MAXIMUM VALUE | TIME INTERVAL | CURRENT VALUE |
|----------------|---------------|-----------------------|------------------|------------------|------------------|------------------|
| STATION 1 BUSY | 0.0 | 0.0 | 0.0 | 0.0 | 0.4336E+03 | 0.0 |
| STATION 2 BUSY | 0.8303E+00 | 0.3754E+00 | 0.0 | 0.1000E+01 | 0.4336E+03 | 0.0 |
| STATION 3 BUSY | 0.8072E+00 | 0.3945E+00 | 0.0 | 0.1000E+01 | 0.4336E+03 | 0.0 |
| STATION 4 BUSY | 0.6458E+00 | 0.5278E+00 | 0.0 | 0.2000E+01 | 0.4336E+03 | 0.0 |
| STATION 5 BUSY | 0.5535E+00 | 0.4971E+00 | 0.0 | 0.1000E+01 | 0.4336E+03 | 0.0 |

FILE STATISTICS

| FILE NUMBER | ASSOCIATED NODE TYPE | AVERAGE LENGTH | STANDARD DEVIATION | MAXIMUM LENGTH | CURRENT LENGTH | AVERAGE WAITING TIME |
|----------------|-------------------------|-------------------|-----------------------|-------------------|-------------------|-------------------------|
| 1 | | 15.3042 | 11.1843 | 34 | 0 | 55.2957 |
| 2 | | 1.1380 | 0.9547 | 4 | 0 | 3.1628 |
| 3 | | 0.1663 | 0.3880 | 2 | 0 | 0.3135 |
| 4 | | 0.0 | 0.0 | 0 | 0 | 0.0 |
| 5 | | 1.5120 | 0.6750 | 2 | 0 | 5.5088 |
| 6 | | 0.1527 | 0.3597 | 1 | 0 | 0.8597 |
| 7 | | 0.0498 | 0.2176 | 1 | 0 | 0.2701 |
| 8 | | 0.0830 | 0.2760 | 1 | 0 | 0.6001 |
| 9 | | 31.5126 | 13.0695 | 80 | 79 | 0.7297 |

INTERMEDIATE RESULTS

**STATISTICS FOR VARIABLES BASED ON OBSERVATION **

| | | MEAN VALUE | STANDARD DEVIATION | COEFF.OF VARIATION | MINIMUM VALUE | MA XIMUM VALUE | NUMBER OF OBSERVATIONS |
|------------------|---|---------------|-----------------------|-----------------------|------------------|-------------------|---------------------------|
| TIME ON CONVEYOR | 1 | 0.9978E+01 | 0.3192E+01 | 0.3200E+00 | 0.0 | 0.1680E+02 | 120 |
| TIME ON CONVEYOR | 2 | 0.2552E+01 | 0.9206E+00 | 0.3608E+00 | 0.0 | 0.4800E+01 | 120 |
| TIME ON CONVEYOR | 3 | 0.2262E+01 | 0.2762E+00 | 0.1221E+00 | 0.0 | 0.3195E+01 | 120 |
| TIME ON CONVEYOR | 4 | 0.1335E+01 | 0.5238E+00 | 0.3923E+00 | 0.0 | 0.3400E+01 | 120 |

**STATISTICS FOR TIME-PERSISTENT VARIABLES **

| | | MEAN VALUE | STANDARD DEVIATION | MINIMUM VALUE | MAXIMUM VALUE | TIME INTERVAL | CURRENT VALUE |
|-----------------|---|---------------|-----------------------|------------------|------------------|------------------|------------------|
| NO. ON CONVEYOR | 1 | 0.2762E+01 | 0.1125E+01 | 0.0 | 0.5000E+01 | 0.4336E+03 | 0.0 |
| NO. ON CONVEYOR | 2 | 0.7062E+00 | 0.4555E+00 | 0.0 | 0.1000E+01 | 0.4336E+03 | 0.0 |
| NO. ON CONVEYOR | 3 | 0.6260E+00 | 0.5599E+00 | 0.0 | 0.2000E+01 | 0.4336E+03 | 0.0 |
| NO. ON CONVEYOR | 4 | 0.3695E+00 | 0.5159E+00 | 0.0 | 0.2000E+01 | 0.4336E+03 | 0.0 |

EXAMPLE 4

SLAM INPUT

```
GEN,ASIF SHAIKH,EXAMPLE,1/13/84,1;  
LIM,9,20,300;  
TIMST,XX(1),STATION 1 BUSY;  
TIMST,XX(2),STATION 2 BUSY;  
TIMST,XX(3),STATION 3 BUSY;  
TIMST,XX(4),STATION 4 BUSY;  
TIMST,XX(5),STATION 5 BUSY;  
TIMST,XX(6),STATION 6 BUSY;  
STAT,1,JOB1 IN SYSTEM;  
SIM;
```

MODEL INPUT

1

1

17

0 0 0 0 1 2 0 0

2 1 14 0 1 3 0 0

1 2 0 0 2 4 5 0

1 3 0 0 1 5 0 0

2 3 4 0 1 6 0 0

1 5 0 0 2 7 8 0

1 6 0 0 1 8 0 0

2 6 7 0 1 9 0 0

2 8 14 0 1 10 0 0

1 9 0 0 1 11 0 0

1 10 0 0 2 12 13 0

1 11 0 0 1 13 0 0

2 11 12 0 2 14 15 0

1 13 0 0 2 2 9 0

1 13 0 0 1 16 0 0

1 15 0 0 2 15 17 0

1 16 0 0 0 0 0 0

22

3. 6. 6. 3. 6. 6. 6. 3. 6. 6. 48. 9. 9. 3. 9. 62. 3. 62. 36.

45. 45. 45.

1 2

2 3

3 4

3 5

4 5

5 6

6 7

6 8

7 8

8 9

9 10

10 11

11 12

11 13

12 13

13 14

14 2

14 9

13 15

15 16

16 15

16 17

30.

2.5

6

1 3 5 7 19 2

1

0.

4

8.64 10.56

4

9.72 10.8

4

14.4 21.6

4

80.64 120.96

4

5.83 7.13

1 1 0 0 0 0 1 0 0 0 0

1 4 0 0 0 0 3 0 0 0 0

1 7 0 0 0 0 5 0 0 0 0

1 12 0 0 0 0 7 0 0 0 0

1 16 0 0 0 0 19 0 0 0 0

1 17 0 0 0 0 2 0 0 0 0

1

6

1 2 3 4 5 6

1

1

4

9.72 10.8

300

0.

1000.

Run 1

**STATISTICS FOR VARIABLES BASED ON OBSERVATION **

| | | | MEAN VALUE | STANDARD DEVIATION | COEFF.OF VARIATION | MINIMUM VALUE | MA XIMUM VALUE | NUMBER OF OBSERVATIONS |
|------------------|----|--|---------------|-----------------------|-----------------------|------------------|-------------------|---------------------------|
| TIME ON CONVEYOR | 1 | | 0.9989E-01 | 0.3524E-04 | 0.3528E-03 | 0.0 | 0.1000E+00 | 97 |
| TIME ON CONVEYOR | 2 | | 0.2000E+00 | 0.6063E-03 | 0.3032E-02 | 0.0 | 0.2000E+00 | 97 |
| TIME ON CONVEYOR | 3 | | 0.2000E+00 | 0.6063E-03 | 0.3032E-02 | 0.0 | 0.2000E+00 | 97 |
| TIME ON CONVEYOR | 4 | | | NO VALUES | RECORDED | | | |
| TIME ON CONVEYOR | 5 | | 0.2000E+00 | 0.6176E-03 | 0.3089E-02 | 0.0 | 0.2000E+00 | 96 |
| TIME ON CONVEYOR | 6 | | 0.2000E+00 | 0.6176E-03 | 0.3089E-02 | 0.0 | 0.2000E+00 | 96 |
| TIME ON CONVEYOR | 7 | | 0.2000E+00 | 0.6176E-03 | 0.3089E-02 | 0.0 | 0.2000E+00 | 96 |
| TIME ON CONVEYOR | 8 | | | NO VALUES | RECORDED | | | |
| TIME ON CONVEYOR | 9 | | 0.2000E+00 | 0.6127E-03 | 0.3064E-02 | 0.0 | 0.2000E+00 | 95 |
| TIME ON CONVEYOR | 10 | | 0.2000E+00 | 0.6127E-03 | 0.3064E-02 | 0.0 | 0.2000E+00 | 95 |
| TIME ON CONVEYOR | 11 | | 0.1600E+01 | 0.2417E-02 | 0.1511E-02 | 0.0 | 0.1600E+01 | 95 |
| TIME ON CONVEYOR | 12 | | 0.2998E+00 | 0.0 | 0.0 | 0.0 | 0.3000E+00 | 95 |
| TIME ON CONVEYOR | 13 | | 0.2998E+00 | 0.0 | 0.0 | 0.0 | 0.3000E+00 | 95 |
| TIME ON CONVEYOR | 14 | | | NO VALUES | RECORDED | | | |
| TIME ON CONVEYOR | 15 | | 0.2998E+00 | 0.0 | 0.0 | 0.0 | 0.3000E+00 | 93 |
| TIME ON CONVEYOR | 16 | | | NO VALUES | RECORDED | | | |
| TIME ON CONVEYOR | 17 | | | NO VALUES | RECORDED | | | |
| TIME ON CONVEYOR | 18 | | | NO VALUES | RECORDED | | | |
| TIME ON CONVEYOR | 19 | | 0.1200E+01 | 0.1468E-02 | 0.1224E-02 | 0.0 | 0.1200E+01 | 93 |
| TIME ON CONVEYOR | 20 | | 0.1500E+01 | 0.0 | 0.0 | 0.0 | 0.1500E+01 | 93 |
| TIME ON CONVEYOR | 21 | | | NO VALUES | RECORDED | | | |
| TIME ON CONVEYOR | 22 | | 0.1660E+01 | 0.5949E+00 | 0.3584E+00 | 0.0 | 0.4439E+01 | 82 |

| **STATISTICS FOR TIME-PERSISTENT VARIABLES | | | | | | | | ** | |
|--|----|----------|---------------|-----------------------|------------------|------------------|------------------|------------------|-----|
| | | | MEAN VALUE | STANDARD DEVIATION | MINIMUM VALUE | MAXIMUM VALUE | TIME INTERVAL | CURRENT VALUE | |
| NO. | ON | CONVEYOR | 1 | 0.9689E-02 | 0.9796E-01 | 0.0 | 0.1000E+01 | 0.1000E+04 | 0.0 |
| NO. | ON | CONVEYOR | 2 | 0.1940E-01 | 0.1379E+00 | 0.0 | 0.1000E+01 | 0.1000E+04 | 0.0 |
| NO. | ON | CONVEYOR | 3 | 0.1940E-01 | 0.1379E+00 | 0.0 | 0.1000E+01 | 0.1000E+04 | 0.0 |
| NO. | ON | CONVEYOR | 4 | | NO VALUES | RECORDED | | | |
| NO. | ON | CONVEYOR | 5 | 0.1920E-01 | 0.1372E+00 | 0.0 | 0.1000E+01 | 0.1000E+04 | 0.0 |
| NO. | ON | CONVEYOR | 6 | 0.1920E-01 | 0.1372E+00 | 0.0 | 0.1000E+01 | 0.1000E+04 | 0.0 |
| NO. | ON | CONVEYOR | 7 | 0.1920E-01 | 0.1372E+00 | 0.0 | 0.1000E+01 | 0.1000E+04 | 0.0 |
| NO. | ON | CONVEYOR | 8 | | NO VALUES | RECORDED | | | |
| NO. | ON | CONVEYOR | 9 | 0.1900E-01 | 0.1365E+00 | 0.0 | 0.1000E+01 | 0.1000E+04 | 0.0 |
| NO. | ON | CONVEYOR | 10 | 0.1900E-01 | 0.1365E+00 | 0.0 | 0.1000E+01 | 0.1000E+04 | 0.0 |
| NO. | ON | CONVEYOR | 11 | 0.1520E+00 | 0.3590E+00 | 0.0 | 0.1000E+01 | 0.1000E+04 | 0.0 |
| NO. | ON | CONVEYOR | 12 | 0.2849E-01 | 0.1664E+00 | 0.0 | 0.1000E+01 | 0.1000E+04 | 0.0 |
| NO. | ON | CONVEYOR | 13 | 0.2849E-01 | 0.1664E+00 | 0.0 | 0.1000E+01 | 0.1000E+04 | 0.0 |
| NO. | ON | CONVEYOR | 14 | | NO VALUES | RECORDED | | | |
| NO. | ON | CONVEYOR | 15 | 0.2789E-01 | 0.1646E+00 | 0.0 | 0.1000E+01 | 0.1000E+04 | 0.0 |
| NO. | ON | CONVEYOR | 16 | | NO VALUES | RECORDED | | | |
| NO. | ON | CONVEYOR | 17 | | NO VALUES | RECORDED | | | |
| NO. | ON | CONVEYOR | 18 | | NO VALUES | RECORDED | | | |
| NO. | ON | CONVEYOR | 19 | 0.1116E+00 | 0.3149E+00 | 0.0 | 0.1000E+01 | 0.1000E+04 | 0.0 |
| NO. | ON | CONVEYOR | 20 | 0.1395E+00 | 0.3465E+00 | 0.0 | 0.1000E+01 | 0.1000E+04 | 0.0 |
| NO. | ON | CONVEYOR | 21 | | NO VALUES | RECORDED | | | |
| NO. | ON | CONVEYOR | 22 | 0.1361E+00 | 0.3579E+00 | 0.0 | 0.2000E+01 | 0.1000E+04 | 0.0 |

| | MEAN VALUE | STANDARD DEVIATION | COEFF. OF VARIATION | MINIMUM VALUE | MAXIMUM VALUE | NUMBER OF OBSERVATIONS |
|----------------|---------------|-----------------------|------------------------|------------------|------------------|---------------------------|
| JOB1 IN SYSTEM | 0.1536E+03 | 0.1235E+02 | 0.8042E-01 | 0.1312E+03 | 0.1782E+03 | 81 |

STATISTICS FOR TIME-PERSISTENT VARIABLES

| | MEAN VALUE | STANDARD DEVIATION | MINIMUM VALUE | MAXIMUM VALUE | TIME INTERVAL | CURRENT VALUE |
|----------------|---------------|-----------------------|------------------|------------------|------------------|------------------|
| STATION 1 BUSY | 0.0 | 0.0 | 0.0 | 0.0 | 0.1000E+04 | 0.0 |
| STATION 2 BUSY | 0.9371E+00 | 0.2603E+00 | 0.0 | 0.2000E+01 | 0.1000E+04 | 0.1000E+01 |
| STATION 3 BUSY | 0.9873E+00 | 0.2955E+00 | 0.0 | 0.2000E+01 | 0.1000E+04 | 0.1000E+01 |
| STATION 4 BUSY | 0.1695E+01 | 0.5559E+00 | 0.0 | 0.3000E+01 | 0.1000E+04 | 0.2000E+01 |
| STATION 5 BUSY | 0.8900E+01 | 0.2853E+01 | 0.0 | 0.1300E+02 | 0.1000E+04 | 0.1100E+02 |
| STATION 6 BUSY | 0.5269E+00 | 0.6689E+00 | 0.0 | 0.2000E+01 | 0.1000E+04 | 0.1000E+01 |

FILE STATISTICS

| FILE NUMBER | ASSOCIATED NODE TYPE | AVERAGE LENGTH | STANDARD DEVIATION | MAXIMUM LENGTH | CURRENT LENGTH | AVERAGE WAITING TIME |
|----------------|-------------------------|-------------------|-----------------------|-------------------|-------------------|-------------------------|
| 1 | | 0.0 | 0.0 | 0 | 0 | 0.0 |
| 2 | | 0.0 | 0.0 | 0 | 0 | 0.0 |
| 3 | | 0.0 | 0.0 | 0 | 0 | 0.0 |
| 4 | | 0.0 | 0.0 | 0 | 0 | 0.0 |
| 5 | | 0.0 | 0.0 | 0 | 0 | 0.0 |
| 6 | | 0.0 | 0.0 | 0 | 0 | 0.0 |
| 7 | | 0.0 | 0.0 | 0 | 0 | 0.0 |
| 8 | | 0.0 | 0.0 | 0 | 0 | 0.0 |
| 9 | | 0.0131 | 0.1263 | 2 | 0 | 1.8728 |
| 10 | | 15.7996 | 3.4748 | 20 | 17 | 5.3942 |

Run 2.

**STATISTICS FOR VARIABLES BASED ON OBSERVATION **

| | | | MEAN VALUE | STANDARD DEVIATION | COEFF.OF VARIATION | MINIMUM VALUE | MA XIMUM VALUE | NUMBER OF OBSERVATIONS |
|------------------|----|--|---------------|-----------------------|-----------------------|------------------|-------------------|---------------------------|
| TIME ON CONVEYOR | 1 | | 0.9989E-01 | 0.3524E-04 | 0.3528E-03 | 0.0 | 0.1000E+00 | 97 |
| TIME ON CONVEYOR | 2 | | 0.2000E+00 | 0.6063E-03 | 0.3032E-02 | 0.0 | 0.2000E+00 | 97 |
| TIME ON CONVEYOR | 3 | | 0.2000E+00 | 0.6063E-03 | 0.3032E-02 | 0.0 | 0.2000E+00 | 97 |
| TIME ON CONVEYOR | 4 | | | NO VALUES RECORDED | | | | |
| TIME ON CONVEYOR | 5 | | 0.2000E+00 | 0.6176E-03 | 0.3089E-02 | 0.0 | 0.2000E+00 | 96 |
| TIME ON CONVEYOR | 6 | | 0.2000E+00 | 0.6176E-03 | 0.3089E-02 | 0.0 | 0.2000E+00 | 96 |
| TIME ON CONVEYOR | 7 | | 0.2000E+00 | 0.6176E-03 | 0.3089E-02 | 0.0 | 0.2000E+00 | 96 |
| TIME ON CONVEYOR | 8 | | | NO VALUES RECORDED | | | | |
| TIME ON CONVEYOR | 9 | | 0.2000E+00 | 0.6127E-03 | 0.3064E-02 | 0.0 | 0.2000E+00 | 95 |
| TIME ON CONVEYOR | 10 | | 0.2000E+00 | 0.6127E-03 | 0.3064E-02 | 0.0 | 0.2000E+00 | 95 |
| TIME ON CONVEYOR | 11 | | 0.3999E+00 | 0.4273E-03 | 0.1069E-02 | 0.0 | 0.4000E+00 | 95 |
| TIME ON CONVEYOR | 12 | | 0.2000E+00 | 0.6127E-03 | 0.3064E-02 | 0.0 | 0.2000E+00 | 95 |
| TIME ON CONVEYOR | 13 | | 0.2000E+00 | 0.6127E-03 | 0.3064E-02 | 0.0 | 0.2000E+00 | 95 |
| TIME ON CONVEYOR | 14 | | | NO VALUES RECORDED | | | | |
| TIME ON CONVEYOR | 15 | | 0.2000E+00 | 0.6160E-03 | 0.3080E-02 | 0.0 | 0.2000E+00 | 94 |
| TIME ON CONVEYOR | 16 | | | NO VALUES RECORDED | | | | |
| TIME ON CONVEYOR | 17 | | | NO VALUES RECORDED | | | | |
| TIME ON CONVEYOR | 18 | | | NO VALUES RECORDED | | | | |
| TIME ON CONVEYOR | 19 | | 0.2998E+00 | 0.1013E-03 | 0.3377E-03 | 0.0 | 0.3000E+00 | 94 |
| TIME ON CONVEYOR | 20 | | 0.5000E+00 | 0.4643E-03 | 0.9286E-03 | 0.0 | 0.5000E+00 | 638 |
| TIME ON CONVEYOR | 21 | | 0.5000E+00 | 0.7300E-03 | 0.1460E-02 | 0.0 | 0.5000E+00 | 545 |
| TIME ON CONVEYOR | 22 | | 0.5000E+00 | 0.0 | 0.0 | 0.0 | 0.5000E+00 | 82 |

| **STATISTICS FOR TIME-PERSISTENT VARIABLES | | | | | | | ** | | |
|--|----|----------|---------------|-----------------------|--------------------|------------------|------------------|------------------|------------|
| | | | MEAN VALUE | STANDARD DEVIATION | MINIMUM VALUE | MAXIMUM VALUE | TIME INTERVAL | CURRENT VALUE | |
| NO. | ON | CONVEYOR | 1 | 0.9689E-02 | 0.9796E-01 | 0.0 | 0.1000E+01 | 0.1000E+04 | 0.0 |
| NO. | ON | CONVEYOR | 2 | 0.1940E-01 | 0.1379E+00 | 0.0 | 0.1000E+01 | 0.1000E+04 | 0.0 |
| NO. | ON | CONVEYOR | 3 | 0.1940E-01 | 0.1379E+00 | 0.0 | 0.1000E+01 | 0.1000E+04 | 0.0 |
| NO. | ON | CONVEYOR | 4 | | NO VALUES RECORDED | | | | |
| NO. | ON | CONVEYOR | 5 | 0.1920E-01 | 0.1372E+00 | 0.0 | 0.1000E+01 | 0.1000E+04 | 0.0 |
| NO. | ON | CONVEYOR | 6 | 0.1920E-01 | 0.1372E+00 | 0.0 | 0.1000E+01 | 0.1000E+04 | 0.0 |
| NO. | ON | CONVEYOR | 7 | 0.1920E-01 | 0.1372E+00 | 0.0 | 0.1000E+01 | 0.1000E+04 | 0.0 |
| NO. | ON | CONVEYOR | 8 | | NO VALUES RECORDED | | | | |
| NO. | ON | CONVEYOR | 9 | 0.1900E-01 | 0.1365E+00 | 0.0 | 0.1000E+01 | 0.1000E+04 | 0.0 |
| NO. | ON | CONVEYOR | 10 | 0.1900E-01 | 0.1365E+00 | 0.0 | 0.1000E+01 | 0.1000E+04 | 0.0 |
| NO. | ON | CONVEYOR | 11 | 0.3799E-01 | 0.1912E+00 | 0.0 | 0.1000E+01 | 0.1000E+04 | 0.0 |
| NO. | ON | CONVEYOR | 12 | 0.1900E-01 | 0.1365E+00 | 0.0 | 0.1000E+01 | 0.1000E+04 | 0.0 |
| NO. | ON | CONVEYOR | 13 | 0.1900E-01 | 0.1365E+00 | 0.0 | 0.1000E+01 | 0.1000E+04 | 0.0 |
| NO. | ON | CONVEYOR | 14 | | NO VALUES RECORDED | | | | |
| NO. | ON | CONVEYOR | 15 | 0.1880E-01 | 0.1358E+00 | 0.0 | 0.1000E+01 | 0.1000E+04 | 0.0 |
| NO. | ON | CONVEYOR | 16 | | NO VALUES RECORDED | | | | |
| NO. | ON | CONVEYOR | 17 | | NO VALUES RECORDED | | | | |
| NO. | ON | CONVEYOR | 18 | | NO VALUES RECORDED | | | | |
| NO. | ON | CONVEYOR | 19 | 0.2819E-01 | 0.1655E+00 | 0.0 | 0.1000E+01 | 0.1000E+04 | 0.0 |
| NO. | ON | CONVEYOR | 20 | 0.3193E+00 | 0.6560E+00 | 0.0 | 0.5000E+01 | 0.1000E+04 | 0.1000E+01 |
| NO. | ON | CONVEYOR | 21 | 0.2725E+00 | 0.6276E+00 | 0.0 | 0.5000E+01 | 0.1000E+04 | 0.1000E+01 |
| NO. | ON | CONVEYOR | 22 | 0.4100E-01 | 0.1999E+00 | 0.0 | 0.2000E+01 | 0.1000E+04 | 0.0 |

| | MEAN VALUE | STANDARD DEVIATION | COEFF. OF VARIATION | MINIMUM VALUE | MAXIMUM VALUE | NUMBER OF OBSERVATIONS |
|----------------|---------------|-----------------------|------------------------|------------------|------------------|---------------------------|
| JOB1 IN SYSTEM | 0.1469E+03 | 0.1550E+02 | 0.1056E+00 | 0.1225E+03 | 0.2080E+03 | 82 |

STATISTICS FOR TIME-PERSISTENT VARIABLES

| | MEAN VALUE | STANDARD DEVIATION | MINIMUM VALUE | MAXIMUM VALUE | TIME INTERVAL | CURRENT VALUE |
|----------------|---------------|-----------------------|------------------|------------------|------------------|------------------|
| STATION 1 BUSY | 0.0 | 0.0 | 0.0 | 0.0 | 0.1000E+04 | 0.0 |
| STATION 2 BUSY | 0.9371E+00 | 0.2603E+00 | 0.0 | 0.2000E+01 | 0.1000E+04 | 0.1000E+01 |
| STATION 3 BUSY | 0.9873E+00 | 0.2955E+00 | 0.0 | 0.2000E+01 | 0.1000E+04 | 0.1000E+01 |
| STATION 4 BUSY | 0.1697E+01 | 0.5528E+00 | 0.0 | 0.3000E+01 | 0.1000E+04 | 0.1000E+01 |
| STATION 5 BUSY | 0.8735E+01 | 0.2634E+01 | 0.0 | 0.1000E+02 | 0.1000E+04 | 0.1000E+02 |
| STATION 6 BUSY | 0.0 | 0.0 | 0.0 | 0.0 | 0.1000E+04 | 0.0 |

FILE STATISTICS

| FILE NUMBER | ASSOCIATED NODE TYPE | AVERAGE LENGTH | STANDARD DEVIATION | MAXIMUM LENGTH | CURRENT LENGTH | AVERAGE WAITING TIME |
|----------------|-------------------------|-------------------|-----------------------|-------------------|-------------------|-------------------------|
| 1 | | 0.0 | 0.0 | 0 | 0 | 0.0 |
| 2 | | 0.0 | 0.0 | 0 | 0 | 0.0 |
| 3 | | 0.0 | 0.0 | 0 | 0 | 0.0 |
| 4 | | 0.0 | 0.0 | 0 | 0 | 0.0 |
| 5 | | 0.0 | 0.0 | 0 | 0 | 0.0 |
| 6 | | 0.0 | 0.0 | 0 | 0 | 0.0 |
| 7 | | 0.0 | 0.0 | 0 | 0 | 0.0 |
| 8 | | 0.0 | 0.0 | 0 | 0 | 0.0 |
| 9 | | 0.0 | 0.0 | 0 | 0 | 0.0 |
| 10 | | 15.2539 | 3.3746 | 21 | 16 | 3.3386 |

Run 3

**STATISTICS FOR VARIABLES BASED ON OBSERVATION **

| | | MEAN VALUE | STANDARD DEVIATION | COEFF.OF VARIATION | MINIMUM VALUE | MA XIMUM VALUE | NUMBER OF OBSERVATIONS |
|------------------|----|---------------|-----------------------|-----------------------|------------------|-------------------|---------------------------|
| TIME ON CONVEYOR | 1 | 0.9989E-01 | 0.1424E-03 | 0.1426E-02 | 0.0 | 0.1000E+00 | 142 |
| TIME ON CONVEYOR | 2 | 0.2000E+00 | 0.6476E-03 | 0.3238E-02 | 0.0 | 0.2000E+00 | 142 |
| TIME ON CONVEYOR | 3 | 0.2000E+00 | 0.6631E-03 | 0.3316E-02 | 0.0 | 0.2000E+00 | 142 |
| TIME ON CONVEYOR | 4 | | NO VALUES | RECORDED | | | |
| TIME ON CONVEYOR | 5 | 0.2000E+00 | 0.6705E-03 | 0.3353E-02 | 0.0 | 0.2000E+00 | 141 |
| TIME ON CONVEYOR | 6 | 0.2000E+00 | 0.6705E-03 | 0.3353E-02 | 0.0 | 0.2000E+00 | 141 |
| TIME ON CONVEYOR | 7 | 0.2000E+00 | 0.6705E-03 | 0.3353E-02 | 0.0 | 0.2000E+00 | 141 |
| TIME ON CONVEYOR | 8 | | NO VALUES | RECORDED | | | |
| TIME ON CONVEYOR | 9 | 0.2000E+00 | 0.6650E-03 | 0.3326E-02 | 0.0 | 0.2000E+00 | 139 |
| TIME ON CONVEYOR | 10 | 0.2000E+00 | 0.6650E-03 | 0.3326E-02 | 0.0 | 0.2000E+00 | 139 |
| TIME ON CONVEYOR | 11 | 0.3999E+00 | 0.1663E-02 | 0.4157E-02 | 0.0 | 0.4000E+00 | 139 |
| TIME ON CONVEYOR | 12 | 0.2000E+00 | 0.6650E-03 | 0.3326E-02 | 0.0 | 0.2000E+00 | 139 |
| TIME ON CONVEYOR | 13 | 0.2000E+00 | 0.6650E-03 | 0.3326E-02 | 0.0 | 0.2000E+00 | 139 |
| TIME ON CONVEYOR | 14 | | NO VALUES | RECORDED | | | |
| TIME ON CONVEYOR | 15 | 0.2000E+00 | 0.6671E-03 | 0.3336E-02 | 0.0 | 0.2000E+00 | 136 |
| TIME ON CONVEYOR | 16 | | NO VALUES | RECORDED | | | |
| TIME ON CONVEYOR | 17 | | NO VALUES | RECORDED | | | |
| TIME ON CONVEYOR | 18 | | NO VALUES | RECORDED | | | |
| TIME ON CONVEYOR | 19 | 0.2998E+00 | 0.1189E-03 | 0.3964E-03 | 0.0 | 0.3000E+00 | 136 |
| TIME ON CONVEYOR | 20 | 0.5000E+00 | 0.0 | 0.0 | 0.0 | 0.5000E+00 | 631 |
| TIME ON CONVEYOR | 21 | 0.5000E+00 | 0.0 | 0.0 | 0.0 | 0.5000E+00 | 495 |
| TIME ON CONVEYOR | 22 | 0.5000E+00 | 0.0 | 0.0 | 0.0 | 0.5000E+00 | 121 |

| **STATISTICS FOR TIME-PERSISTENT VARIABLES | | | | | | | ** | | |
|--|----|----------|---------------|-----------------------|------------------|------------------|------------------|------------------|-----|
| | | | MEAN VALUE | STANDARD DEVIATION | MINIMUM VALUE | MAXIMUM VALUE | TIME INTERVAL | CURRENT VALUE | |
| NO. | ON | CONVEYOR | 1 | 0.1418E-01 | 0.1182E+00 | 0.0 | 0.1000E+01 | 0.1000E+04 | 0.0 |
| NO. | ON | CONVEYOR | 2 | 0.2839E-01 | 0.1661E+00 | 0.0 | 0.1000E+01 | 0.1000E+04 | 0.0 |
| NO. | ON | CONVEYOR | 3 | 0.2839E-01 | 0.1661E+00 | 0.0 | 0.1000E+01 | 0.1000E+04 | 0.0 |
| NO. | ON | CONVEYOR | 4 | | NO VALUES | RECORDED | | | |
| NO. | ON | CONVEYOR | 5 | 0.2819E-01 | 0.1655E+00 | 0.0 | 0.1000E+01 | 0.1000E+04 | 0.0 |
| NO. | ON | CONVEYOR | 6 | 0.2819E-01 | 0.1655E+00 | 0.0 | 0.1000E+01 | 0.1000E+04 | 0.0 |
| NO. | ON | CONVEYOR | 7 | 0.2819E-01 | 0.1655E+00 | 0.0 | 0.1000E+01 | 0.1000E+04 | 0.0 |
| NO. | ON | CONVEYOR | 8 | | NO VALUES | RECORDED | | | |
| NO. | ON | CONVEYOR | 9 | 0.2779E-01 | 0.1644E+00 | 0.0 | 0.1000E+01 | 0.1000E+04 | 0.0 |
| NO. | ON | CONVEYOR | 10 | 0.2779E-01 | 0.1644E+00 | 0.0 | 0.1000E+01 | 0.1000E+04 | 0.0 |
| NO. | ON | CONVEYOR | 11 | 0.5559E-01 | 0.2291E+00 | 0.0 | 0.1000E+01 | 0.1000E+04 | 0.0 |
| NO. | ON | CONVEYOR | 12 | 0.2779E-01 | 0.1644E+00 | 0.0 | 0.1000E+01 | 0.1000E+04 | 0.0 |
| NO. | ON | CONVEYOR | 13 | 0.2779E-01 | 0.1644E+00 | 0.0 | 0.1000E+01 | 0.1000E+04 | 0.0 |
| NO. | ON | CONVEYOR | 14 | | NO VALUES | RECORDED | | | |
| NO. | ON | CONVEYOR | 15 | 0.2719E-01 | 0.1630E+00 | 0.0 | 0.2000E+01 | 0.1000E+04 | 0.0 |
| NO. | ON | CONVEYOR | 16 | | NO VALUES | RECORDED | | | |
| NO. | ON | CONVEYOR | 17 | | NO VALUES | RECORDED | | | |
| NO. | ON | CONVEYOR | 18 | | NO VALUES | RECORDED | | | |
| NO. | ON | CONVEYOR | 19 | 0.4078E-01 | 0.1986E+00 | 0.0 | 0.2000E+01 | 0.1000E+04 | 0.0 |
| NO. | ON | CONVEYOR | 20 | 0.3155E+00 | 0.6017E+00 | 0.0 | 0.3000E+01 | 0.1000E+04 | 0.0 |
| NO. | ON | CONVEYOR | 21 | 0.2475E+00 | 0.5580E+00 | 0.0 | 0.3000E+01 | 0.1000E+04 | 0.0 |
| NO. | ON | CONVEYOR | 22 | 0.6050E-01 | 0.2447E+00 | 0.0 | 0.2000E+01 | 0.1000E+04 | 0.0 |

| | MEAN VALUE | STANDARD DEVIATION | COEFF. OF VARIATION | MINIMUM VALUE | MAXIMUM VALUE | NUMBER OF OBSERVATIONS |
|----------------|---------------|-----------------------|------------------------|------------------|------------------|---------------------------|
| JOB1 IN SYSTEM | 0.1460E+03 | 0.1474E+02 | 0.1010E+00 | 0.1205E+03 | 0.1926E+03 | 121 |

STATISTICS FOR TIME-PERSISTENT VARIABLES

| | MEAN VALUE | STANDARD DEVIATION | MINIMUM VALUE | MAXIMUM VALUE | TIME INTERVAL | CURRENT VALUE |
|----------------|---------------|-----------------------|------------------|------------------|------------------|------------------|
| STATION 1 BUSY | 0.0 | 0.0 | 0.0 | 0.0 | 0.1000E+04 | 0.0 |
| STATION 2 BUSY | 0.1359E+01 | 0.4981E+00 | 0.0 | 0.2000E+01 | 0.1000E+04 | 0.1000E+01 |
| STATION 3 BUSY | 0.1439E+01 | 0.5347E+00 | 0.0 | 0.2000E+01 | 0.1000E+04 | 0.2000E+01 |
| STATION 4 BUSY | 0.2463E+01 | 0.6990E+00 | 0.0 | 0.4000E+01 | 0.1000E+04 | 0.3000E+01 |
| STATION 5 BUSY | 0.1295E+02 | 0.3909E+01 | 0.0 | 0.1500E+02 | 0.1000E+04 | 0.1500E+02 |
| STATION 6 BUSY | 0.0 | 0.0 | 0.0 | 0.0 | 0.1000E+04 | 0.0 |

FILE STATISTICS

| FILE NUMBER | ASSOCIATED NODE TYPE | AVERAGE LENGTH | STANDARD DEVIATION | MAXIMUM LENGTH | CURRENT LENGTH | AVERAGE WAITING TIME |
|----------------|-------------------------|-------------------|-----------------------|-------------------|-------------------|-------------------------|
| 1 | | 0.0 | 0.0 | 0 | 0 | 0.0 |
| 2 | | 0.0 | 0.0 | 0 | 0 | 0.0 |
| 3 | | 0.0 | 0.0 | 0 | 0 | 0.0 |
| 4 | | 0.0 | 0.0 | 0 | 0 | 0.0 |
| 5 | | 0.0 | 0.0 | 0 | 0 | 0.0 |
| 6 | | 0.0 | 0.0 | 0 | 0 | 0.0 |
| 7 | | 0.0 | 0.0 | 0 | 0 | 0.0 |
| 8 | | 0.0 | 0.0 | 0 | 0 | 0.0 |
| 9 | | 0.0 | 0.0 | 0 | 0 | 0.0 |
| 10 | | 0.0 | 0.0 | 0 | 0 | 0.0 |
| 11 | | 21.2247 | 4.7641 | 26 | 22 | 3.6753 |

Run 4

**STATISTICS FOR VARIABLES BASED ON OBSERVATION **

| | | MEAN VALUE | STANDARD DEVIATION | COEFF.OF VARIATION | MINIMUM VALUE | MAXIMUM VALUE | NUMBER OF OBSERVATIONS |
|---------------------|------------|--------------------|-----------------------|-----------------------|------------------|------------------|---------------------------|
| TIME ON CONVEYOR 1 | 0.2132E+01 | 0.6283E+01 | 0.2947E+01 | 0.0 | 0.5288E+02 | 91 | |
| TIME ON CONVEYOR 2 | 0.7940E+00 | 0.3594E+01 | 0.4527E+01 | 0.0 | 0.5558E+02 | 640 | |
| TIME ON CONVEYOR 3 | 0.1264E+01 | 0.4719E+01 | 0.3735E+01 | 0.0 | 0.5558E+02 | 366 | |
| TIME ON CONVEYOR 4 | 0.1723E+00 | 0.6698E+00 | 0.3887E+01 | 0.0 | 0.8767E+01 | 272 | |
| TIME ON CONVEYOR 5 | 0.1071E+01 | 0.4877E+01 | 0.4556E+01 | 0.0 | 0.6259E+02 | 362 | |
| TIME ON CONVEYOR 6 | 0.8141E+00 | 0.3703E+01 | 0.4548E+01 | 0.0 | 0.5718E+02 | 632 | |
| TIME ON CONVEYOR 7 | 0.2019E+01 | 0.6286E+01 | 0.3113E+01 | 0.0 | 0.5718E+02 | 215 | |
| TIME ON CONVEYOR 8 | 0.1744E+00 | 0.8408E+00 | 0.4820E+01 | 0.0 | 0.1399E+02 | 415 | |
| TIME ON CONVEYOR 9 | 0.1136E+01 | 0.5946E+01 | 0.5234E+01 | 0.0 | 0.5718E+02 | 211 | |
| TIME ON CONVEYOR 10 | 0.5510E+00 | 0.3743E+01 | 0.6793E+01 | 0.0 | 0.6634E+02 | 624 | |
| TIME ON CONVEYOR 11 | 0.1743E+01 | 0.6370E+01 | 0.3655E+01 | 0.0 | 0.5126E+02 | 619 | |
| TIME ON CONVEYOR 12 | 0.1252E+01 | 0.4749E+01 | 0.3792E+01 | 0.0 | 0.4871E+02 | 616 | |
| TIME ON CONVEYOR 13 | 0.4330E+01 | 0.1012E+02 | 0.2338E+01 | 0.0 | 0.5652E+02 | 198 | |
| TIME ON CONVEYOR 14 | 0.9997E-01 | 0.7103E-03 | 0.7105E-02 | 0.0 | 0.9999E-01 | 415 | |
| TIME ON CONVEYOR 15 | 0.4191E+01 | 0.1020E+02 | 0.2433E+01 | 0.0 | 0.5652E+02 | 192 | |
| TIME ON CONVEYOR 16 | 0.5969E+00 | 0.1137E+01 | 0.1904E+01 | 0.0 | 0.2106E+02 | 551 | |
| TIME ON CONVEYOR 17 | 0.1629E+00 | 0.7408E+00 | 0.4547E+01 | 0.0 | 0.1040E+02 | 551 | |
| TIME ON CONVEYOR 18 | | NO VALUES RECORDED | | | | | |
| TIME ON CONVEYOR 19 | 0.1031E+02 | 0.1136E+02 | 0.1101E+01 | 0.0 | 0.3874E+02 | 54 | |
| TIME ON CONVEYOR 20 | 0.8987E+01 | 0.2059E+02 | 0.2291E+01 | 0.0 | 0.8163E+02 | 215 | |
| TIME ON CONVEYOR 21 | 0.5527E+00 | 0.5703E+00 | 0.1032E+01 | 0.0 | 0.7727E+01 | 167 | |
| TIME ON CONVEYOR 22 | 0.5000E+00 | 0.0 | 0.0 | 0.0 | 0.5000E+00 | 38 | |

**STATISTICS FOR TIME-PERSISTENT VARIABLES

**

| | | | MEAN VALUE | STANDARD DEVIATION | MINIMUM VALUE | MAXIMUM VALUE | TIME INTERVAL | CURRENT VALUE | |
|-----|----|----------|---------------|-----------------------|------------------|------------------|------------------|------------------|------------|
| NO. | ON | CONVEYOR | 1 | 0.4155E+00 | 0.4928E+00 | 0.0 | 0.1000E+01 | 0.5000E+03 | 0.1000E+01 |
| NO. | ON | CONVEYOR | 2 | 0.1073E+01 | 0.9116E+00 | 0.0 | 0.2000E+01 | 0.5000E+03 | 0.2000E+01 |
| NO. | ON | CONVEYOR | 3 | 0.9812E+00 | 0.9379E+00 | 0.0 | 0.2000E+01 | 0.5000E+03 | 0.2000E+01 |
| NO. | ON | CONVEYOR | 4 | 0.9373E-01 | 0.2915E+00 | 0.0 | 0.1000E+01 | 0.5000E+03 | 0.0 |
| NO. | ON | CONVEYOR | 5 | 0.8313E+00 | 0.8928E+00 | 0.0 | 0.2000E+01 | 0.5000E+03 | 0.2000E+01 |
| NO. | ON | CONVEYOR | 6 | 0.1086E+01 | 0.8974E+00 | 0.0 | 0.2000E+01 | 0.5000E+03 | 0.2000E+01 |
| NO. | ON | CONVEYOR | 7 | 0.9249E+00 | 0.9616E+00 | 0.0 | 0.2000E+01 | 0.5000E+03 | 0.2000E+01 |
| NO. | ON | CONVEYOR | 8 | 0.1448E+00 | 0.3519E+00 | 0.0 | 0.1000E+01 | 0.5000E+03 | 0.0 |
| NO. | ON | CONVEYOR | 9 | 0.5360E+00 | 0.7990E+00 | 0.0 | 0.2000E+01 | 0.5000E+03 | 0.2000E+01 |
| NO. | ON | CONVEYOR | 10 | 0.7782E+00 | 0.7972E+00 | 0.0 | 0.2000E+01 | 0.5000E+03 | 0.2000E+01 |
| NO. | ON | CONVEYOR | 11 | 0.2477E+01 | 0.1978E+01 | 0.0 | 0.6000E+01 | 0.5000E+03 | 0.5000E+01 |
| NO. | ON | CONVEYOR | 12 | 0.1670E+01 | 0.1183E+01 | 0.0 | 0.3000E+01 | 0.5000E+03 | 0.3000E+01 |
| NO. | ON | CONVEYOR | 13 | 0.1841E+01 | 0.1393E+01 | 0.0 | 0.3000E+01 | 0.5000E+03 | 0.3000E+01 |
| NO. | ON | CONVEYOR | 14 | 0.8298E-01 | 0.2758E+00 | 0.0 | 0.1000E+01 | 0.5000E+03 | 0.0 |
| NO. | ON | CONVEYOR | 15 | 0.1736E+01 | 0.1359E+01 | 0.0 | 0.3000E+01 | 0.5000E+03 | 0.3000E+01 |
| NO. | ON | CONVEYOR | 16 | 0.6578E+00 | 0.1025E+01 | 0.0 | 0.6000E+01 | 0.5000E+03 | 0.0 |
| NO. | ON | CONVEYOR | 17 | 0.1795E+00 | 0.3838E+00 | 0.0 | 0.1000E+01 | 0.5000E+03 | 0.0 |
| NO. | ON | CONVEYOR | 18 | NO VALUES RECORDED | | | | | |
| NO. | ON | CONVEYOR | 19 | 0.1199E+01 | 0.9519E+00 | 0.0 | 0.2000E+01 | 0.5000E+03 | 0.2000E+01 |
| NO. | ON | CONVEYOR | 20 | 0.4296E+01 | 0.2580E+01 | 0.0 | 0.6000E+01 | 0.5000E+03 | 0.6000E+01 |
| NO. | ON | CONVEYOR | 21 | 0.1846E+00 | 0.6876E+00 | 0.0 | 0.6000E+01 | 0.5000E+03 | 0.0 |
| NO. | ON | CONVEYOR | 22 | 0.3800E-01 | 0.1966E+00 | 0.0 | 0.2000E+01 | 0.5000E+03 | 0.0 |

| **STATISTICS FOR VARIABLES BASED ON OBSERVATION** | | | | | | |
|---|---------------|-----------------------|------------------------|------------------|------------------|---------------------------|
| | MEAN VALUE | STANDARD DEVIATION | COEFF. OF VARIATION | MINIMUM VALUE | MAXIMUM VALUE | NUMBER OF OBSERVATIONS |
| JOB1 IN SYSTEM | 0.2233E+03 | 0.5985E+02 | 0.2680E+00 | 0.1417E+03 | 0.3500E+03 | 38 |

| **STATISTICS FOR TIME-PERSISTENT VARIABLES** | | | | | | |
|--|---------------|-----------------------|------------------|------------------|------------------|------------------|
| | MEAN VALUE | STANDARD DEVIATION | MINIMUM VALUE | MAXIMUM VALUE | TIME INTERVAL | CURRENT VALUE |
| STATION 1 BUSY | 0.0 | 0.0 | 0.0 | 0.0 | 0.5000E+03 | 0.0 |
| STATION 2 BUSY | 0.1683E+01 | 0.6585E+00 | 0.0 | 0.2000E+01 | 0.5000E+03 | 0.0 |
| STATION 3 BUSY | 0.1625E+01 | 0.7213E+00 | 0.0 | 0.2000E+01 | 0.5000E+03 | 0.0 |
| STATION 4 BUSY | 0.2242E+01 | 0.1000E+01 | 0.0 | 0.3000E+01 | 0.5000E+03 | 0.1000E+01 |
| STATION 5 BUSY | 0.8460E+01 | 0.3293E+01 | 0.0 | 0.1000E+02 | 0.5000E+03 | 0.1000E+02 |
| STATION 6 BUSY | 0.0 | 0.0 | 0.0 | 0.0 | 0.5000E+03 | 0.0 |

| **FILE STATISTICS** | | | | | | |
|---------------------|-------------------------|-------------------|-----------------------|-------------------|-------------------|-------------------------|
| FILE NUMBER | ASSOCIATED NODE TYPE | AVERAGE LENGTH | STANDARD DEVIATION | MAXIMUM LENGTH | CURRENT LENGTH | AVERAGE WAITING TIME |
| 1 | | 14.7567 | 15.1183 | 48 | 48 | 9.7211 |
| 2 | | 0.0 | 0.0 | 0 | 0 | 0.0 |
| 3 | | 0.0 | 0.0 | 0 | 0 | 0.0 |
| 4 | | 0.0 | 0.0 | 0 | 0 | 0.0 |
| 5 | | 0.8340 | 0.9780 | 2 | 2 | 11.2698 |
| 6 | | 0.8381 | 0.9802 | 2 | 2 | 13.0960 |
| 7 | | 1.7208 | 1.4644 | 3 | 3 | 23.9006 |
| 8 | | 4.0753 | 2.7493 | 6 | 6 | 47.3871 |
| 9 | | 0.0 | 0.0 | 0 | 0 | 0.0 |
| 10 | | 19.8080 | 6.8594 | 34 | 12 | 0.6905 |

APPENDIX C

USER'S MANUAL

The design of a material handling system is very critical if an economic manufacture is desired. Material handling directly affects the production. Perhaps the most critical step in planning a material handling system is the evaluation of a proposed system design. In evaluating the performance of conveyor systems, design and operational problems should be studied together.

This study is conducted to develop a simulation model in SLAM II to simulate conveyor systems and help the analyst and designer in evaluation and design of conveyor systems. The model is capable of simulating different conveyor systems regardless of what kind of conveyor is being used.

This manual, in conjunction with the report, is to describe the model (program) and help user understand the flow of model and relationship among subroutines used. Even though the program is documented, a brief description of what each subroutine does is also presented.

Model uses a number of attributes, and files. Description of events and variables is also presented.

Attributes: Attributes play an important role in storing information about a part. There are a number of attributes associated with each entity. These attributes identify the part, part route, part path, etc. A description of the attributes is as follows:

Attribute 1 -Arrival time to the system

| | |
|--------------|--|
| Attribute 2 | -Last station visited (Station I.D.) |
| Attribute 3 | -Next station to be visted (Station I.D.) |
| Attribute 4 | -Part type |
| Attribute 5 | -Number of nodes visited on the current path |
| Attribute 6 | -Number of stations visited in the system |
| Attribute 7 | -Total number of nodes on the current path |
| Attribute 8 | -Arrival time on a conveyor section |
| Attribute 9 | -Last arc visted (Arc I.D.) |
| Attribute 10 | -Next arc to be visited (Arc I.D.) |
| Attribute 11 | -Last node visited on the current path (first node) |
| Attribute 12 | -Next node to be visited on the current path (second node) |
| Attribute 13 | -Third node on the path (if any) |
| Attribute 14 | -Fourth node on the path (if any) and so on. |

Attributes 11 and onwards are used to store the current path of a part. For example: if the path between station 2 and 5 is 2, 6, 8, and 15, then

Attribute (11) = 2

Attribute (12) = 6

Attribute (13) = 8

Attribute (14) = 15.

After each node is visited attributes 11 and 12 are assigned to their new values. Thus after the first node, the values will be

Attribute (11) = 6

Attribute (12) = 8

Attribute (13) = 8

Attribute (14) = 15.

The maximum number of attributes associated with a part 'i' is

$$10 + \max (\text{NODPTH})_i$$

Where $(\text{NODPTH})_i$ is the number of nodes on a path between two stations for part type i. There is a limit of 100 on the total number of attributes.

Files:

Files are used in storing entities rather than information about entities. When an entity cannot advance in the system, it is placed in a file. A number of files are used in the model which are used under different conditions and at different places in the system. Each service areas has a file to store entities whily they are waiting for service. When a part is at a node and cannot move in the system it is placed in file 1 or 3 depending on the class of conveyor. Examples will help understand the use of files. At the pre-

sent time, consider files 1,2, and 3 as working files where entities are placed for tracking and for some other checking. Files 2 and 3 may or may not be used depending on the class of conveyor. File 4 and onwards are associated with each service area. The maximum number of files which may be used is

$$3 + \text{NSAREA}$$

where NSAREA is the number of service areas in the system. There is a limit of 100 on the number of files.

Events:

Once a part, or entity, enters the system, it is assigned its attributes. The route, the path, etc. is determined at any stage in the system by looking at the proper attributes. These attributes are modified after each event. The events used are described here.

- | | |
|---------|--|
| Event 1 | -Arrival to a conveyor section. Also includes arrival to the system. |
| Event 2 | -Reaching the end of a conveyor section. |
| Event 3 | -End of service. |
| Event 4 | -Arrival of a loading point on the conveyor (only for class 3 conveyors) |
| Event 5 | -Clear statistics at a specified time, during or at the beginning of |

a simulation period.

Event 6 -End of simulation period.

Global variables, XX(.), are used to define the status of a service area. For example, if XX(2) has a value 5, then there are five servers busy at station 2 at that time. If an XX(.) variable has a value of zero, it means none of the servers are busy at a specific station at that time.

The variables used in the model are consistent in notation throughout the model. These variables are defined in Table 7.

Detail on what each subroutine does and the linkages among subroutines is presented in Appendix C. A program listing is included in Appendix A.

Table 7.

Variable Definition

| | |
|-------------|--|
| ARATE(i) | Arrival rate for part type i |
| ARINT | Time between two consecutive loading points. |
| ASIZE(i) | Length of arc i |
| CAP(i) | Capacity of conveyor (arc) i |
| CC(i) | Number of parts on conveyor i |
| DISTLD | Distance between two successive loading points. |
| IBLOCK(i) | Number of servers blocked at station i |
| ISHORT(i,j) | Matrix to store paths among all nodes |
| LINKN(i,j) | Matrix to store linkages among all nodes |
| LOCATA(i,j) | Matrix to store locations of all arcs in the network |
| LOCATP(i) | Location of entrance i |
| LOCATS(i,j) | Matrix to store location(s) of all stations |
| MAXPRT(i) | Maximum number of part i allowed in the system |
| MMST(i) | { 0- conveyor i running { 1- conveyor i not running |
| NARC | Number of arcs in the network |
| NCTYPE | Conveyor class |
| NEPT | Number of entrances for a part type |
| NNARC | Arc numbers |
| NNODE | Number of nodes in the network |
| NPART | Types of parts entering the system |

Table 7. (continued)

Variable Definition

| | |
|-------------|--|
| NRECIR | { 0- recirculation not allowed 1- recirculation allowed |
| NROUTE(i,j) | Route for part i which visivits j |
| NSAREA | Number of stations in the system |
| NVIST | Number of stations part i visits |
| NVST(i) | Number of stations part i visits |
| NSERVER(i) | Number of servers per station i |
| SPEED | Conveyor speed |
| TIMCLR | Time to clear statistics |
| TIMSTP | Time to stop simulation |
| TMLOAD(i) | Last arrival time of a loading point on conveyor i |
| TSTOP(i) | Last time conveyor i stopped |
| TTIME(i) | Travel time on conveyor i |

List of Subroutines

The following routines are used in the program:

| <u>Subroutine Name</u> | <u>Description</u> |
|------------------------|--|
| Main Program | Calls SLAM |
| INTLC | This subroutine initializes variables and calls for INPUT and CALC. Also schedules events. |
| INPUT | This subroutine reads input. |
| SERVIC | This subroutine determines the service times for stations. |
| ARRIVAL(MODE,I) | Assigns arrival rates to all parts. |
| CALC | Calculates values of variables to be used in the program. |
| SHORT | Determines the shortest path among all nodes. |
| FINDD(IC,NN) | Finds distance between nodes IC and NN. |
| FINDA(IC,NN,NNARC) | Finds arc number, NNARC between nodes IC and NN. |

FINDL(NODFR)

Finds the last node,
NODFR, a part visited.

FINDS(NODAT,NSTN,EX)

Determines if a node
NODAT, is a station,
Finds the station I.D.,
NSTN.

$$Ex = \begin{cases} 0, & \text{not a station} \\ 1, & \text{station} \end{cases}$$

FINDS2(NTHIS,NEXT)

Determines the node
numbers which corres-
pond to stations NTHIS,
and NEXT.

EVENT(I)

Directs to proper event,
I.

$$I = \begin{cases} 1 & \text{Arrival to conveyor} \\ 2 & \text{End of a conveyor} \\ 3 & \text{End of service} \\ 4 & \text{Arrival of a loading} \\ & \text{point} \\ 5 & \text{Clear statistics} \\ 6 & \text{Stop simulation} \end{cases}$$

ARRIV(I)

Event 1, Arrival to a

conveyor. Directs to proper routine. Checks for conveyor class and for old or new part.

CONVR(I)

Event 2, End of a conveyor. Directs to proper routine. Checks for conveyor class.

ENDSV(I)

Event 3, End of service. Directs to proper routine. Checks for conveyor class.

NEW1

Arrival of a new part to the system. This routine assigns values to attributes and initializes the part in the system, class 1 conveyor is used.

NEW2

This routine is used when class 2 conveyor is used. Substitutes for NEW1.

| | |
|--------|--|
| NEW3 | Substitutes for NEW1 and NEW2. Used when class 3 conveyor is used. |
| OLD1 | This routine is used to move old parts in the system. Used with class 1 conveyors. |
| OLD2 | This routine moves old parts in the system. Used with class 2 conveyors. |
| OLD3 | This routine moves old parts in the system. Used with class 3 conveyors. |
| CONVR1 | Event 2, class 1 conveyors. This routine also moves parts properly in the system. |
| CONVR2 | Event 2, class 2 conveyors. This routine is used instead of CONVR1 |

| | |
|--------|---|
| CONVR3 | Event 2, class 3 conveyors. This routine is used instead of CONVR1 and CONVR2. |
| RECIR1 | This routine recirculates parts in the system. Used with class 1 conveyors. |
| RECIR2 | This routine is used instead of RECIR1 with class 2 conveyors. |
| ENDSV1 | Event 3, class 1 conveyors. This routine schedules departure of parts from a station. |
| ENDSV2 | Used instead of ENDSV2 for class 2 conveyors. |
| ENDSV3 | Used instead of ENDSV1 and ENDSV2 for class 3 conveyors. |
| ADVNC1 | This routine advances the entire system after |

a part ahead in the system has moved. Used for class 1 conveyors.

ADVNC2

This routine is used instead of ADVNC1 for class 2 conveyors.

ADVNC3

This routine is used for class 3 conveyors. Used instead of ADVNC1 and ADVNC2.

LOAD(I)

This routine is used only for class 3 conveyors. It loads parts into the conveyors, and also moves them in the system.

CLRST

This routine clears statistical arrays at the time when user want to start collecting statistics.

SMMRY

This routine prints summary report and stops simulation.

CCLCT(MODE,X,ISTRM)

This routine collects statistics for variables based on observation.

MMSTT(MODE,T,QTY,ISTRM)

This routine collects statistics for time-persistent variables.

PATH(NTHIS,NEXT,NODPTH)

This routine finds the shortest path between nodes NTHIS and NEXT. NODPTH is the number of nodes on the path.

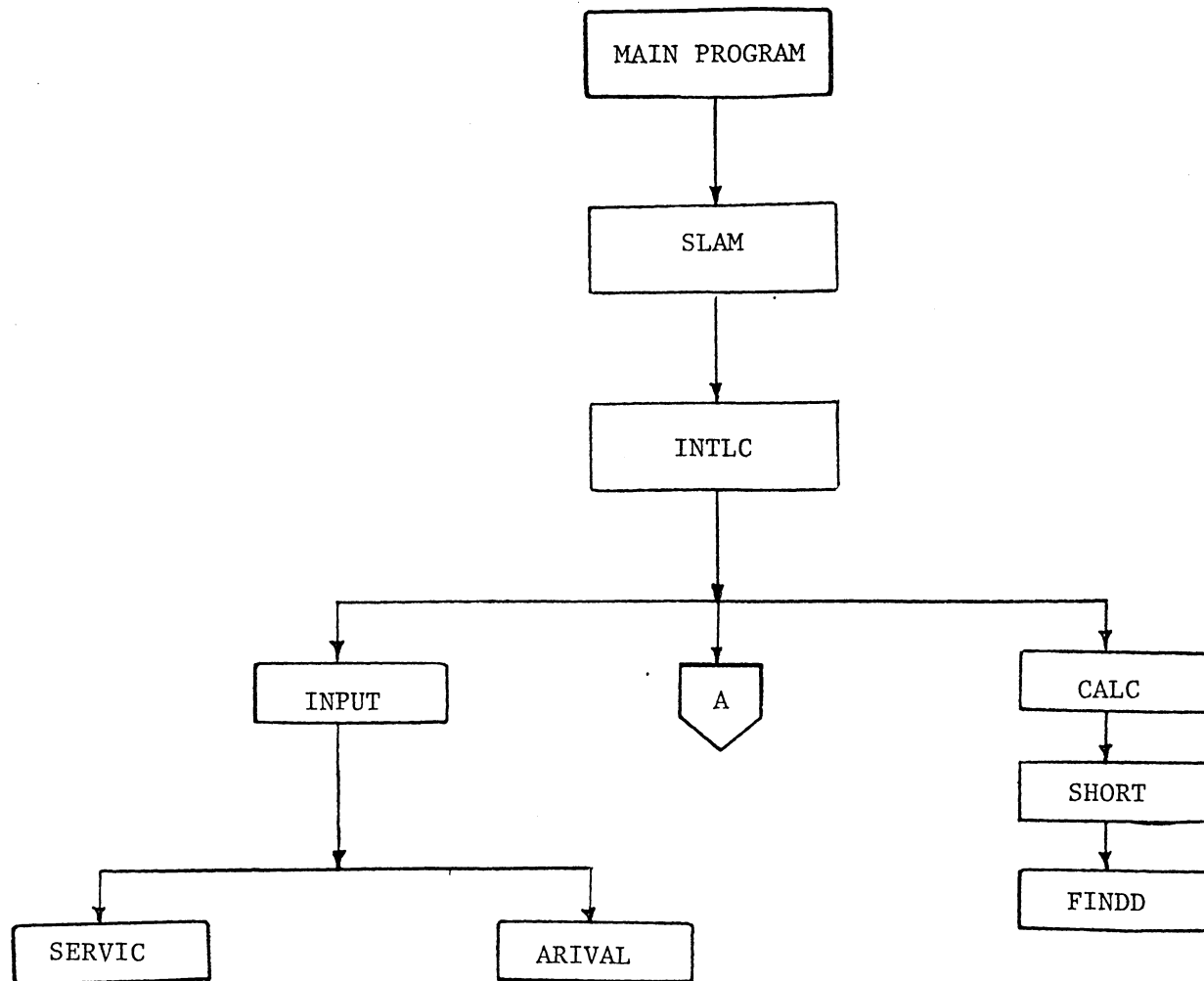


Figure 7. Subroutine Relationship

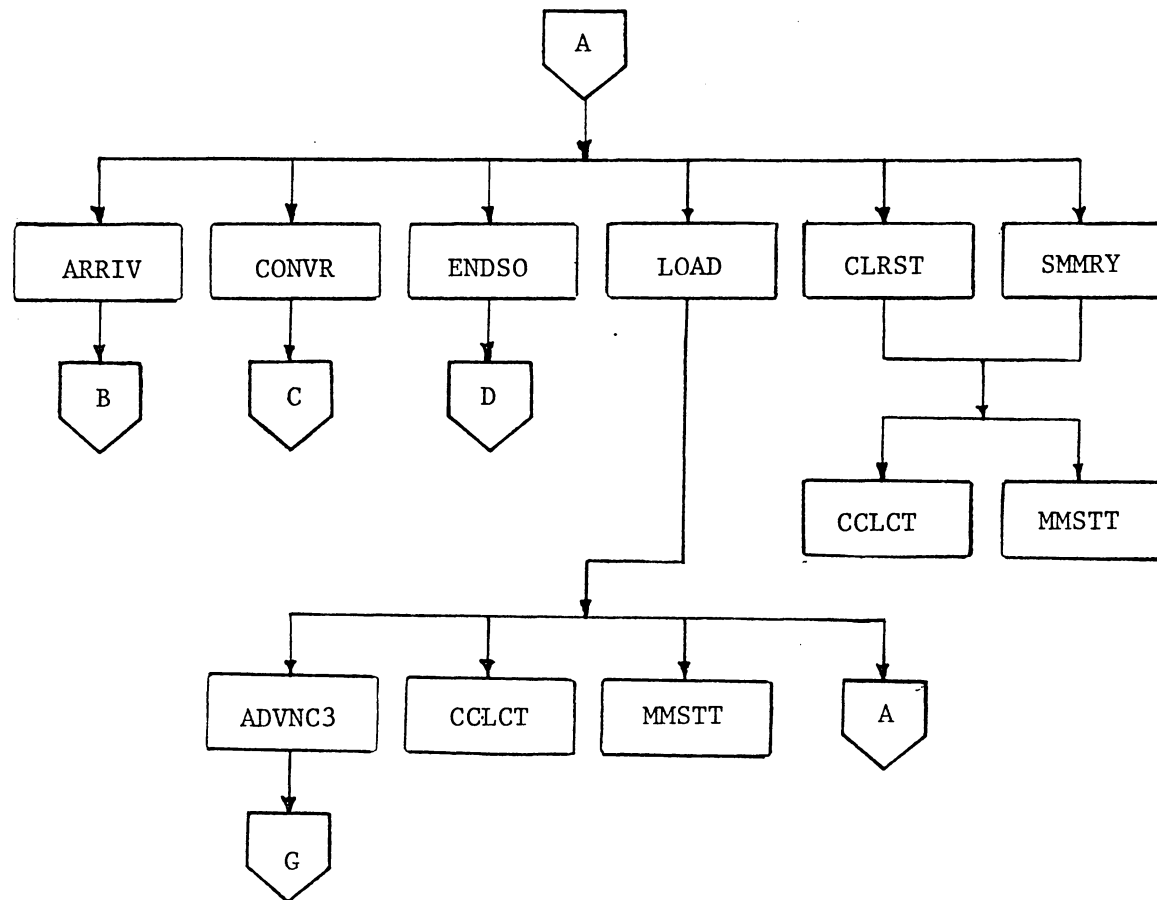


Figure 7. (Continued)

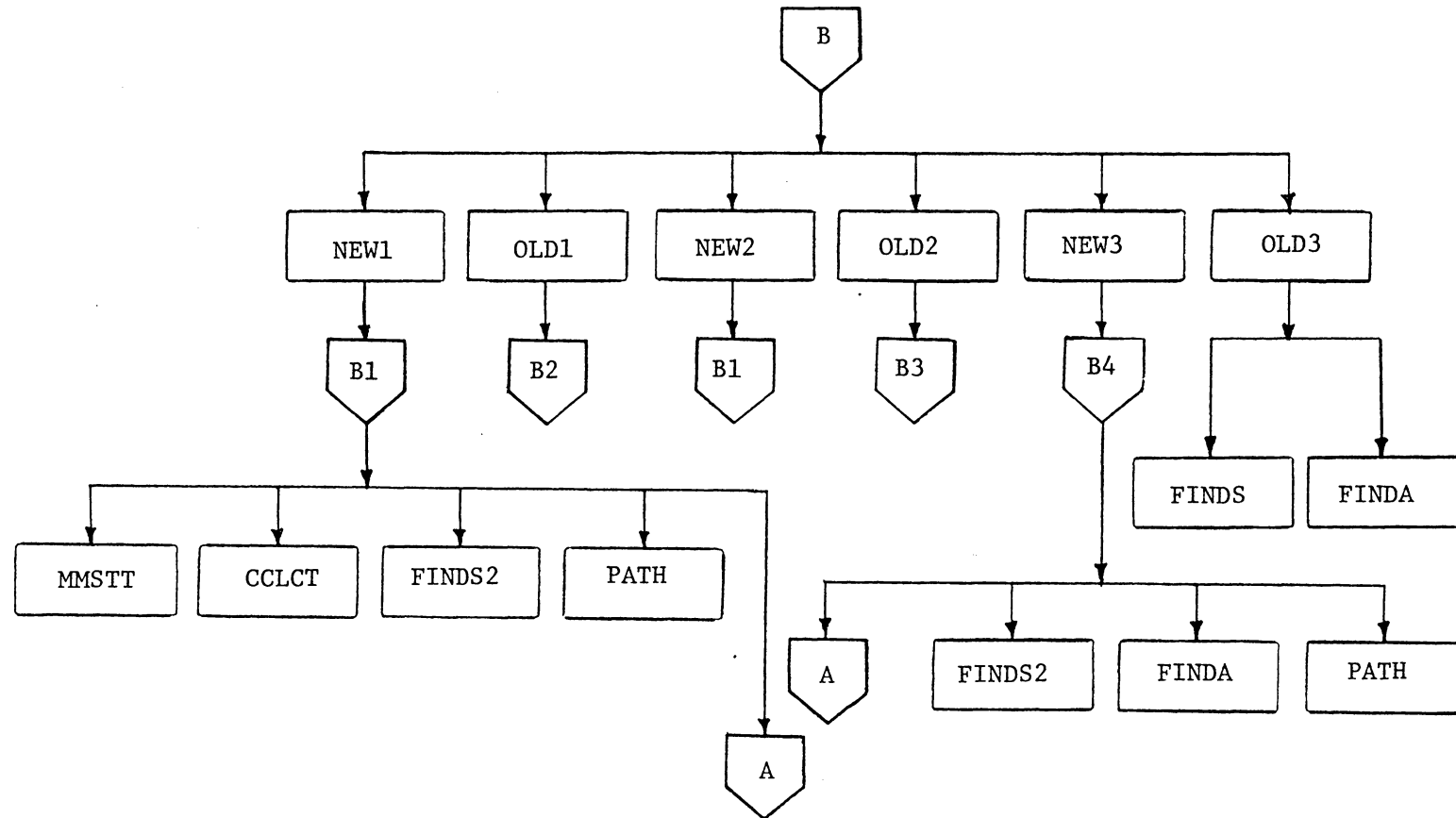


Figure 7. (Continued)

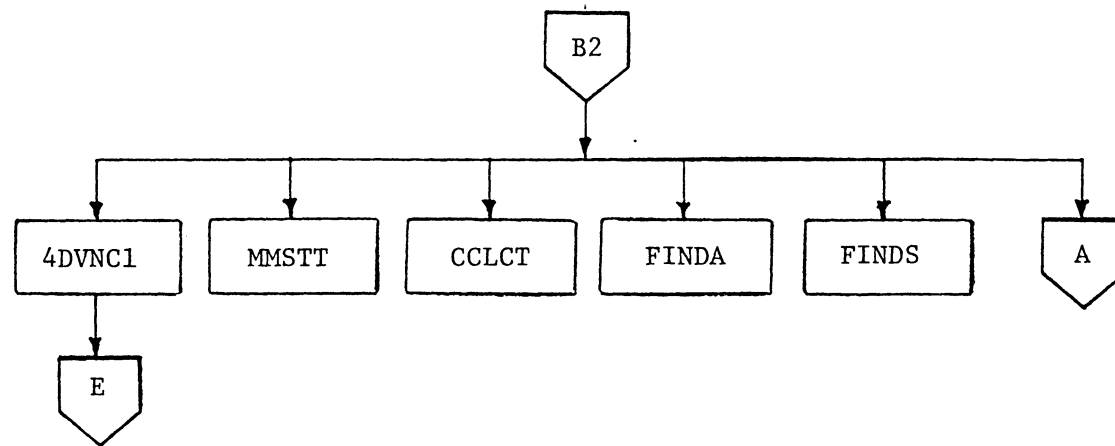


Figure 7. (Continued)

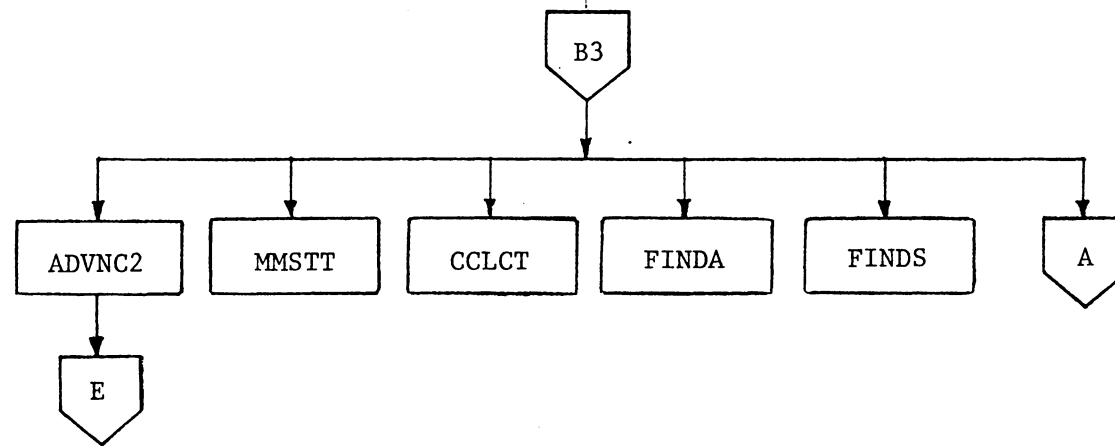


Figure 7. (Continued)

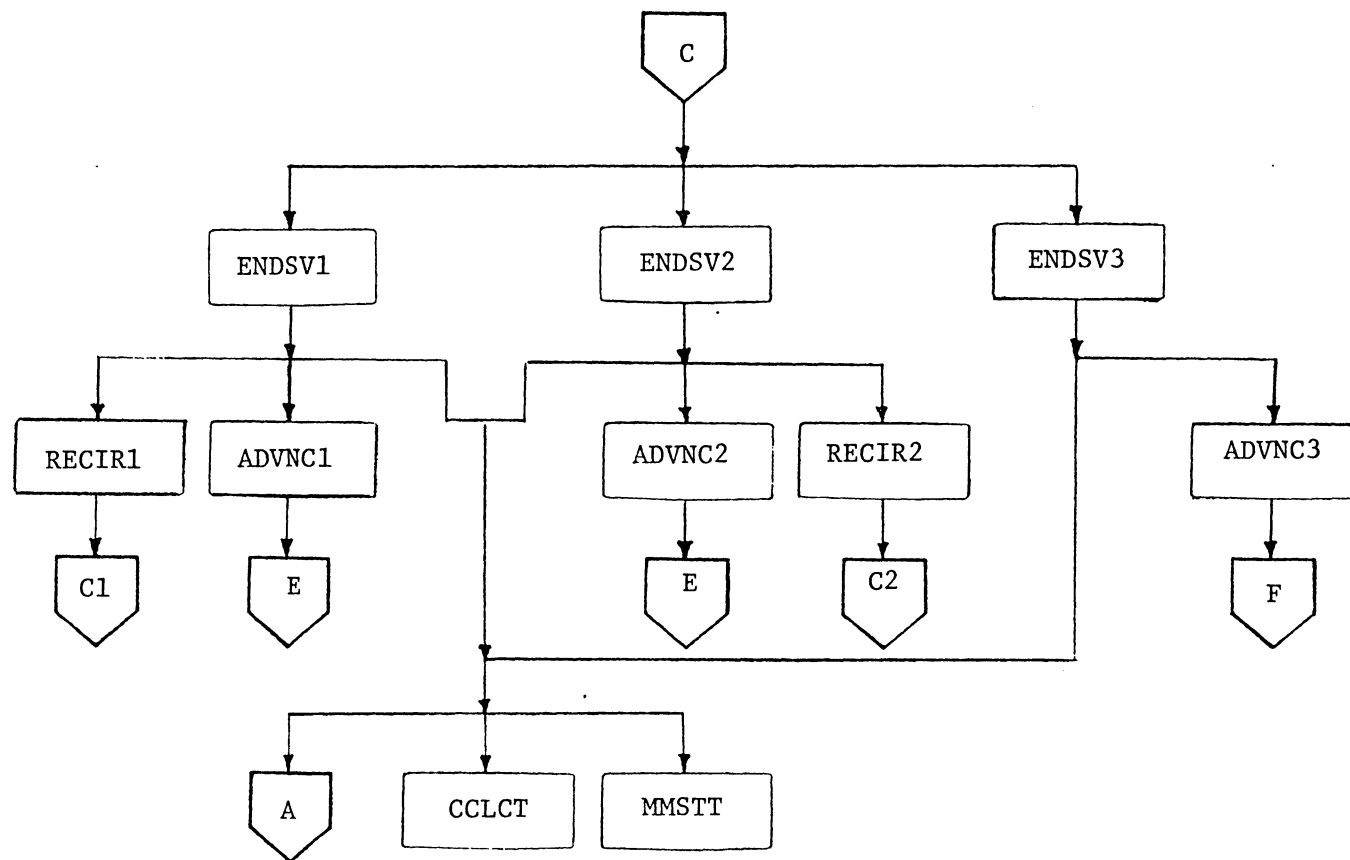


Figure 7. (Continued)

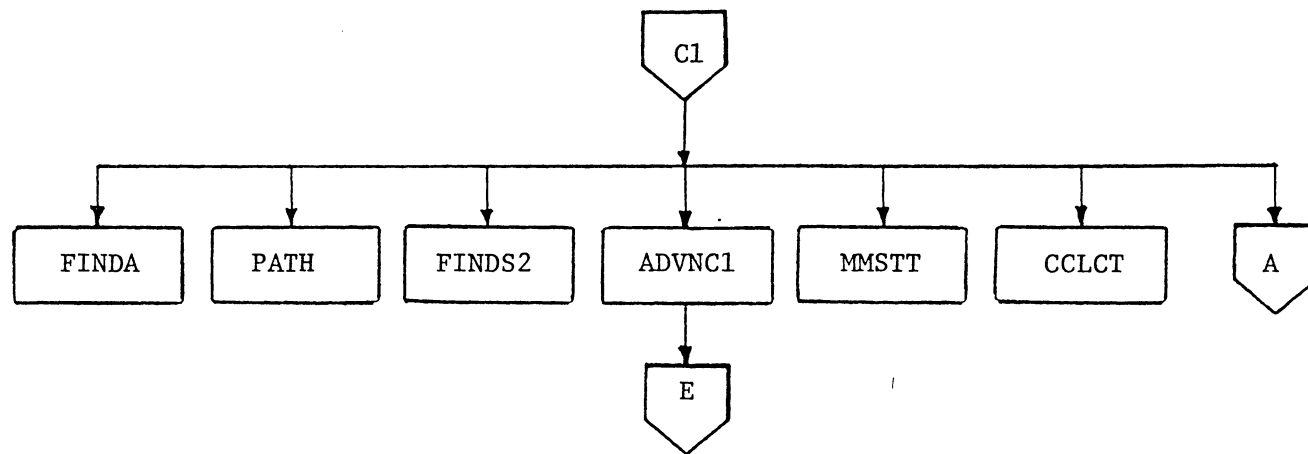


Figure 7. (Continued)

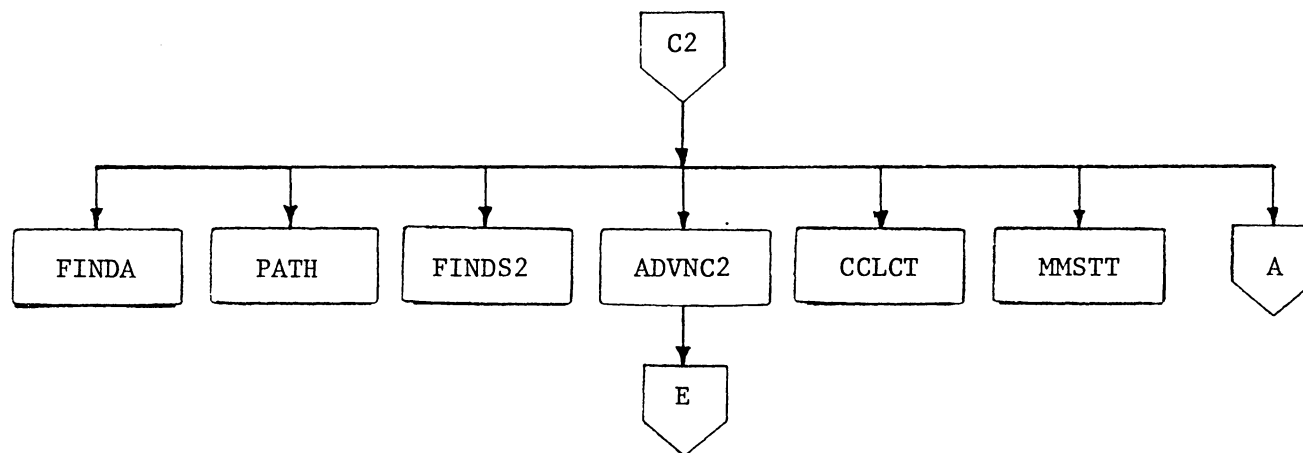


Figure 7. (Continued)

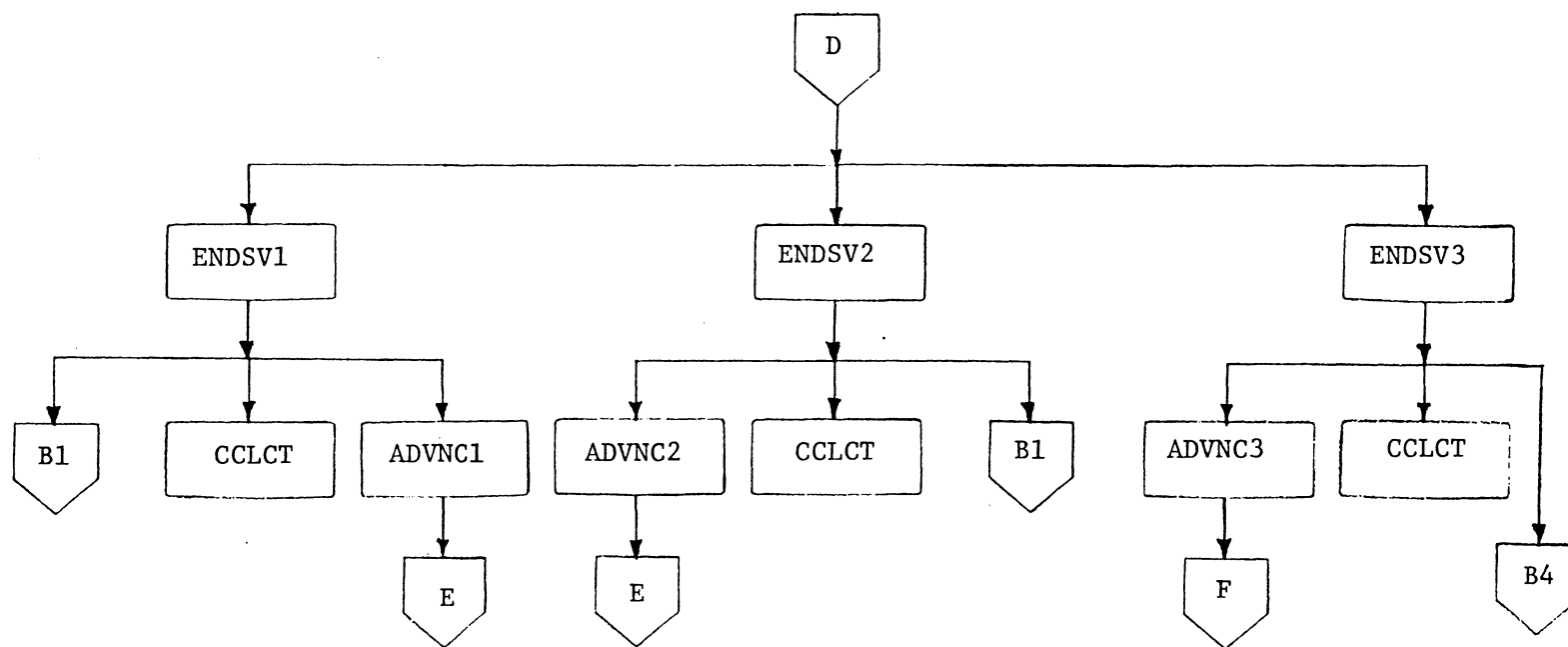


Figure 7. (Continued)

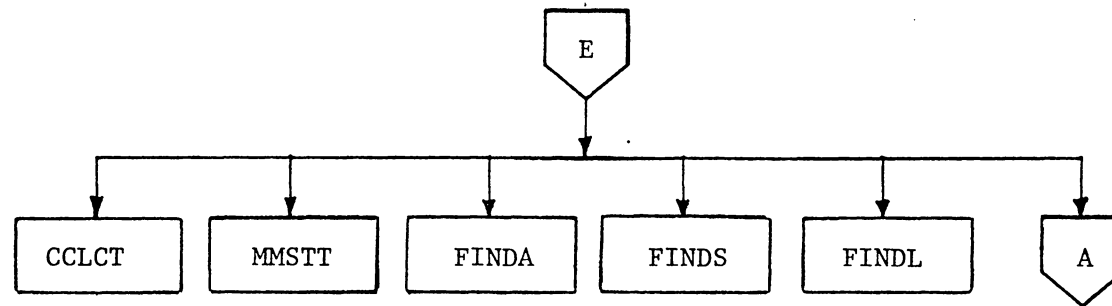


Figure 7. (Continued)

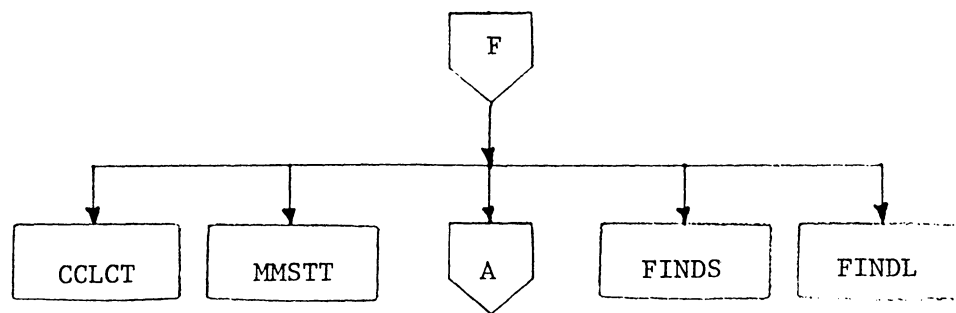


Figure 7. (Continued)

Input:

Appendix B contains input and output of the examples. Consider the input to example 1.

1. SLAM Input. This will not be discussed here. User can consult SLAM manual or text [70]. Only the variables which are dependent upon the conveyor system are discussed.

LIMIT card: The syntax is

LIMIT, MFIL, MATR, MNTRY;

MATR is the number of files, MATR is the number of attributes, and MNTRY is the maximum number of entites allowed in a file. Thus,

$$\begin{aligned} \text{MFIL} &= 3 + \text{number of stations} \\ &= 3 + 8 = 11 \end{aligned}$$

$$\text{MATR} = 10 + \text{maximum number of nodes on a path}$$

let the maximum number be 4.

$$\text{MATR} = 14, \text{ and let}$$

$$\text{MNTRY} = 300$$

TIMST Cards: One card is used for collecting statistics on how many servers are busy at a station. Since there are eight stations, eight of these statements are needed.

STAT Cards: This is used to collect statistics on how much time a part spends in the system. One statement is required for each type of part.

User may also use PRIORITY, INITIALIZE, INTLC, ENTRY, and MONTR statements as needed.

2. MODEL Input. Input to the model is unformatted and described below.

- Conveyor Class; 1, 2, or 3
 - Recirculation Indicator; 0 - not allowed, 1 - allowed
 - Distance between two loading points; only for class 3 conveyors
 - Number of nodes; maximum of 100.
 - Nodes' linkage; one card is required for each node.
- There must be 8 entries on a card. Description of all columns is given below.

Column 1: Total Number of incoming nodes

Columns 2 - 4: I.D. of incoming nodes (maximum of 3 allowed)

Column 5: Total number of outgoing nodes

Columns 6 - 8: I.D. of outgoing nodes (maximum of 3 allowed)

- Number of arcs: maximum of 100
- *Length of each arc;
- Arc location; one card is required for each arc. Each card must have two entries. First entry is the I.D. of the node where the arc is emanating from, and second entry is the I.D. of the terminating node.

- *Conveyor speed; must calculate for gravity chute, etc.
- *Unit size; length of the unit. Length of the side which is parallel to the flow of conveyor.
- Number of stations in the system; maximum of 25
- Number of servers at each station; maximum of 25
- Service time; Two cards may be needed for each station. Service time can be from any of the following distributions:
 - Constant
 - Random
 - Exponential distribution
 - Uniform distribution
 - Triangular distribution
 - Normal distribution

The first card indicates the distribution of service time, and the second card is used to input the parameter values corresponding to the distribution. Service time distribution is indicated by modes. Modes and parameters are listed in Table 8. There is no second card if the distribution is random.

- Station locations; One card is required for each station. There must be 11 entries on each card. Description of all columns is given below.

Table 8.

Service Time and Arrival Rate Distributions

| <u>Distribution</u> | <u>Mode</u> | <u>Parameters</u> |
|---------------------|-------------|-------------------------------|
| Constant | 1 | *constant |
| Random | 2 | ----- |
| Exponential | 3 | *Mean |
| Uniform | 4 | *Low, *Hi values |
| Triangular | 5 | *Low, *Mode, *Hi Values |
| Normal | 6 | *Mean, *Standard deviation |

Column 1: How many different locations are there for that station or station type. This must be 1 at present. Later multiple locations policy will be added in the model.

Columns 2-6: Node I.D. of locations

Column 7-11: Number of servers at each location.

There will be as many as five different locations for each station type. Enter zero if none.

-Part types; maximum of 25

-Part route; Two cards are required for each part type.

First card indicates the number of stations the part will visit. Second card indicates the route of the part in terms of stations.

-Parts input; Two cards are required for each part type.

First card indicates the number of different locations a part can enter the system. Second card indicates the node number of the location. At present, only one entrance is allowed for a part.

-Arrival rate; It is not different from service time.

Two cards may be needed to indicate the mode, and to input the parameter values.

-Limit on parts; Indicate the number of each type of parts allowed in the system.

-*Time to start collecting statistics on variables;

-*Time to stop simulation;

*Must be real. All other input must be integers.

Example 4 input: Consider the layout and network in Figures 6a and 6b. Following is a listing of the model input for this example.

```

1                      Conveyor type 1
1                      Recirculation allowed
17                     Number of nodes
0 0 0 0 1 2 0 0       Nodes' linkage
2 1 14 0 1 3 0 0
1 2 0 0 2 4 5 0
1 3 0 0 1 5 0 0
2 3 4 0 1 6 0 0
1 5 0 0 2 7 8 0
1 6 0 0 1 8 0 0
2 6 7 0 1 9 0 0
2 8 14 0 1 10 0 0
1 9 0 0 1 11 0 0
1 10 0 0 2 12 13 0
1 11 0 0 1 13 0 0
2 11 12 0 2 14 15 0
1 13 0 0 2 2 9 0
1 13 0 0 1 16 0 0
1 15 0 0 2 15 17 0
1 16 0 0 0 0 0 0
22                     Number of arcs
3. 6. 6. 3. 6. 6. 6. 3. 6. 6.    Length of each arc

```

48. 9. 9. 3. 9. 62. 3. 62. 36.

45. 45. 45.

1 2 Arc location

2 3

3 4

3 5

4 5

5 6

6 7

6 8

7 8

8 9

9 10

10 11

11 12

11 13

12 13

13 14

14 2

14 9

13 15

15 16

16 15

16 17

30. Conveyor speed

2.5 Unit size

| | |
|-------------------------|-----------------------------|
| 6 | Number of stations |
| 1 3 5 7 19 2 | Number of servers |
| 1 | Service times |
| 0. | |
| 4 | |
| 8.64 10.56 | |
| 4 | |
| 9.72 10.8 | |
| 4 | |
| 14.4 21.6 | |
| 4 | |
| 80.64 120.96 | |
| 4 | |
| 5.83 7.13 | |
| 1 1 0 0 0 0 1 0 0 0 0 | Location of stations |
| 1 4 0 0 0 0 3 0 0 0 0 | |
| 1 7 0 0 0 0 5 0 0 0 0 | |
| 1 12 0 0 0 0 7 0 0 0 0 | |
| 1 16 0 0 0 0 19 0 0 0 0 | |
| 1 17 0 0 0 0 2 0 0 0 0 | |
| 1 | Part types |
| 6 | Number of machines to visit |
| 1 2 3 4 5 6 | Part route |
| 1 | Number of entry points |
| 1 | Entry point location |
| 4 | Arrival rate |

9.72 10.8

300

Maximum number of parts

0.

Time to start simulation

1000.

Time to stop simulation

Output

There are three parts of the output. First is the SLAM echo report. Second is the intermediate results. SLAM summary report is the third part of the output.

SLAM echo report is basically to check the input data and SLAM variables. This is explained in detail in SLAM text [70]. Intermediate results will always give statistics for how much time a unit spends on a conveyor and how many units are on a conveyor. Average, standard deviation, etc. is printed out.

SLAM summary report provides statistics on how much time each part of a particular type spends in the system. It also gives statistics on how many servers at a station are busy. User can determine other statistics by simple calculations.

An important part of the SLAM summary report is file statistics. This indicates how many parts are waiting at a station and how long do they wait. Consider exaple 1. User may neglect statistics on files 1, 2, and 3. File 1 is used when a part is waiting to move from a station to

a conveyor, or from another conveyor if the node between the two conveyors is not a station. Files 2 and 3 are not used for class 1 conveyors. File 4 through the maximum number of user defined files are used when a part is waiting for service at a station or while waiting to go to another conveyor which is not available. Last file, file 12, is of no interest to the user.

All this information can be helpful in making design decisions and evaluating alternatives. This is discussed in Chapter 5.

**The vita has been removed from
the scanned document**

A GENERALIZED SIMULATION MODEL FOR THE
DESIGN OF A CONVEYOR SYSTEM

by

Asif Manzoor Shaikh

(ABSTRACT)

A simulation model is developed in SLAM II to study and analyze conveyor systems. The adequacy of the model is tested by applying it to a conveyor layout as well as some variant designs. The problem of modeling large systems is reduced by decomposing the system into smaller segments (subsystems). Different types of conveyors are considered. Programming modules are prepared for feasible conveyor types and segments. These simulation modules are integrated into a system.