SIMULATION OF THE PACKAGING PROCESS FOR A CIGARETTE PACKING LINE

Thesis For the Degree of M. S.

MICHIGAN STATE UNIVERSITY
TSUYOSHI MURAMATSU

1975

ABSTRACT

SIMULATION OF THE PACKAGING PROCESS FOR A CIGARETTE PACKING LINE

By

Tsuyoshi Muramatsu

A packing line can be a complex system with many elements. It has several packing stages such as filling, labeling, cartoning, casing and so on. In each stage, there are particular machines which are called fillers, labelers, cartoners, casers, etc. And, finally, a variety of transfer mechanisms take their places between stages or machines to unite all the elements into an organized formation.

In the organized formation, each element--particularly machines--has to interact in such a way that a breakdown of any one of the elements may result in the shutdown of other elements in the system. This interaction generates the idling time to each component which decreases the production rate of the line system. The analysis of the relation between the idling time and the decrease of the system efficiency is primarily what this discussion is concerned with.

Tsuyoshi Muramatsu

Because of the random nature of an occurrence of machine breakdown, simulation techniques were employed instead of analytical methods. A simulation model was constructed, based on a cigarette packing line, and it was computerized. The model has been proven to work. A designer is able to utilize this model to simulate various line configurations and so choose the one that is most economically effective.

SIMULATION OF THE PACKAGING PROCESS
FOR A CIGARETTE PACKING LINE

By

Tsuyoshi Muramatsu

A THESIS

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

MASTER OF SCIENCE

School of Packaging

1975

ACKNOWLEDGMENTS

I want to express my appreciation to Dr. Wayne H. Clifford for the guidance toward this study.

I am also very grateful for the generous support that The Japan Tobacco & Salt Public Corporation gave me during my course work and research.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1. INTRODUCTION

At this moment, it is found that there are several dozen cigarette brands on the United States market. They are all packaged goods and it is interesting to see that a packet of cigarettes seems to be a good example of a "perfect packing" on terms of the functions of packaging, protection, utility and sales promotion which are the roles in the total marketing strategy to be expected for the package to play. It is also said to be a typical merchandise which has to be produced in the modern mass-production system to meet the large amount of consumption.

In looking into a cigarette packing plant, it is found that a packing line is a complex system with many elements. It starts at the stage of the cigarette making process wherein a pile of shredded tobacco is divided into small amounts (about 1.0 grams each) and rolled up into cigarettes with a piece of thin paper which is usually made of flax pulp. It is not an easy task for a machine to do this cigarette-making job because of the uncertain physical properties of the shredded tobacco that is neither a liquid nor a solid but just an unfamiliar object in terms of physical properties. However, today's cigarette-making machines have the performance of 2,000 outputs per minutes with a prospect of increasing it to 4,000 per minute.

At the next stage, cigarettes are collected into a bundle of 20 pieces, then the bundle is wrapped with a sheet of aluminum foil-paper lamination. This stage is followed by the labeling station where a label is applied to the aluminum foil wrapped bundle and is formed into a deep open-topped box with the bundle inside. Another popular form of package at this stage is the hinged-lid type of hard box which is shaped from a carefully designed boxboard blank. After the open top of the box is sealed with a piece of small paper, something like a stamp, a sheet of cellophane is provided to wrap the box with a string of tear tape attached near the top edges of the box to give customers ease in opening the package.

These three stages--aluminum foil wrapping, boxing, and cellophane wrapping--are usually attended by what might be called a cigarette packing system which looks like an individual machine at a glance but is actually an assembled form of three different machines of which each one serves for its own stage. The production rate of the system can range from 100 to about 400 cycles per minute.

Next comes the stages of cartoning, overwrapping and casing, in this order. At first, usually ten packets are arranged into a carton which is a hard box folded up from a boxboard blank. It is sometimes found in this stage that a sheet of wrapping paper, which is usually made of strong kraft pulp, is employed to wrap a dozen packets into the form of a multi-package, instead of using a carton.

After that, a carton which contains many packets is again wrapped with a sheet of cellophane or other type of plastic film. This is called overwrapping.

The final stage will naturally be the casing. Dozens of cartons are put into a shipping case which is usually a corrugated board box.

Machines which serve for the above three stages --cartoning, overwrapping and casing--will have much in common with those that are used in other packaging business firms. So there should be many standard types of such machines from which to choose.

Thus far, there are up to seven stages in a cigarette packing line, which are as follows:

1. rolling up (cigarette making),
2. aluminum foil wrapping,
3. boxing (labeling),
4. cellophane wrapping,
5. cartoning,
6. overwrapping, and
7. casing.

2. PROBLEM STATEMENT

A packaging line is a complex system with many components, as mentioned in the previous chapter by taking a cigarette packing line as an example. It involves stages, machines which are thought to be elements of a stage, and a variety of transfer equipment. All of these elements are combined, arranged and laid out to complete a packing line or system.

A packing line can be a complex system, but how can the line efficiency be determined? Each machine has its characteristic fraction of the time when it is operating within specifications, which may be called the catalog efficiency. For packing machines, it is reasonable to assume that the catalog efficiency ranges from 70% to 90%. In other words, a machine is out of operation for from 10% to 30% of working hours because of machine breakdowns caused by various reasons, such as jamming.

When those machines are placed in some processing line as components, they begin to interact. This interaction between machines generates the idling time to the component machines as an addition to the breakdown time mentioned before. This fact apparently decreases the operational efficiency of the entire line. The breakdown time which depends upon the catalog efficiency may not be controlled by plant design but the idling time can be, because the idling time is caused by putting a machine in a line. Hence it is thought that there will be a way to improve the efficiency of the entire line in terms of decreasing idling time by designing a proper formation of the line system.

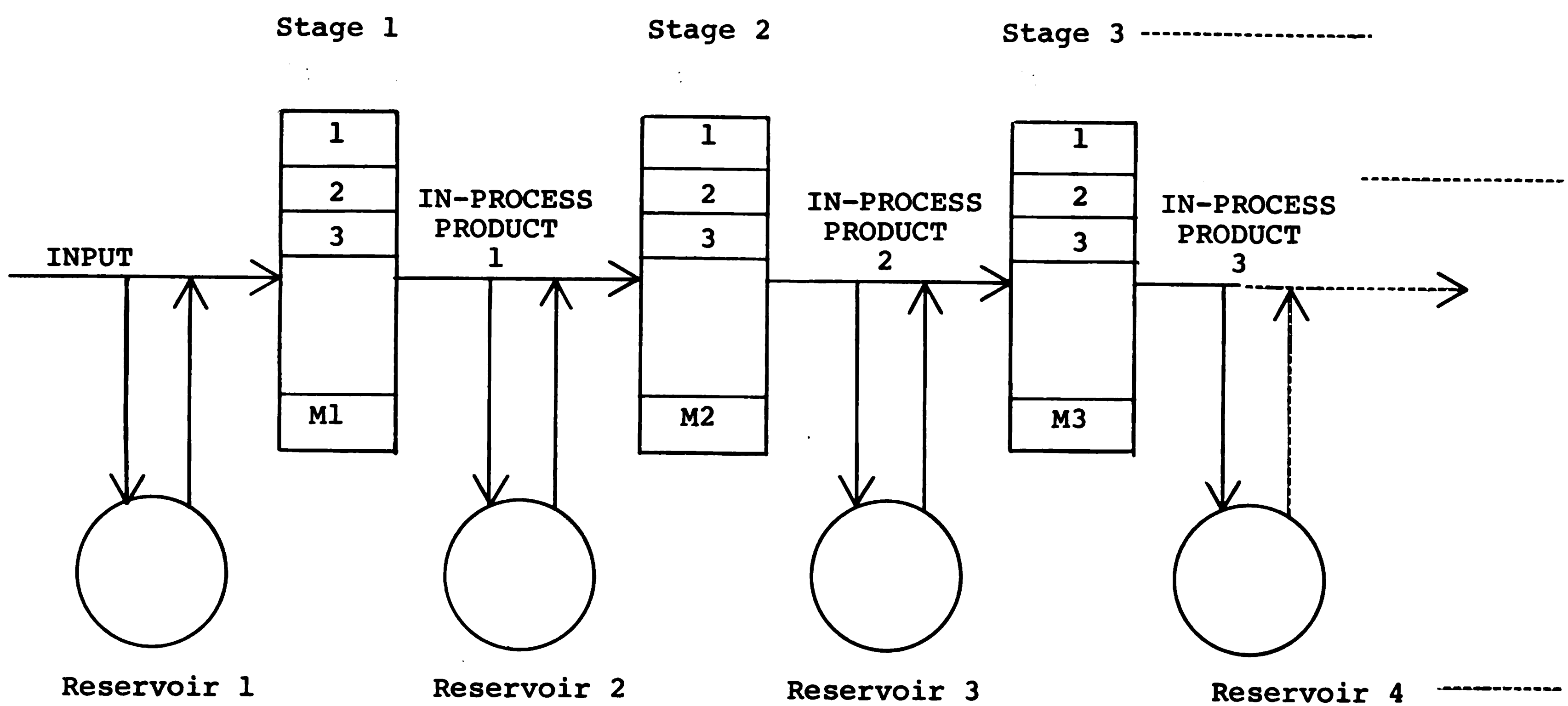
To approach the solution to this problem, analytical methods are not practical, because the occurrence of a machine breakdown or the recovery time from the breakdown has a random tendency. The computer simulation technique is adequate for this purpose.

3. ABSTRACT MODELING

An actual packing line involves many stages such as filling, inner wrapping, outer labeling, boxing, overwrapping, cartoning, casing, etc. Each stage consists of a number of machines which are called fillers, wrappers, labelers, boxers, cartoners, casers and so on. Transfer equipment exists between stages or machines to move the in-process products from one stage to another. Some of them can function as buffers or reservoirs which give a cushion to the interactional movement of stages and machines. A long power-and-free type of transfer conveyor, for example, can hold a limited amount of in-plant products on it in case a machine was shut down ahead. On the other hand, an indexing conveyor cannot hold them at all.

Figure 3.1 visualizes a packing line graphically. In reality, reservoirs in the figure may not be seen as individual equipment. They were introduced there to show that there is a concept of buffers or reservoirs between stages or between machines.

Figure 3.2 is a block diagram which represents the same packing line as Figure 3.1, but is more



1,2,3 .... M: machine number

Figure 3.1.--Packaging Line.

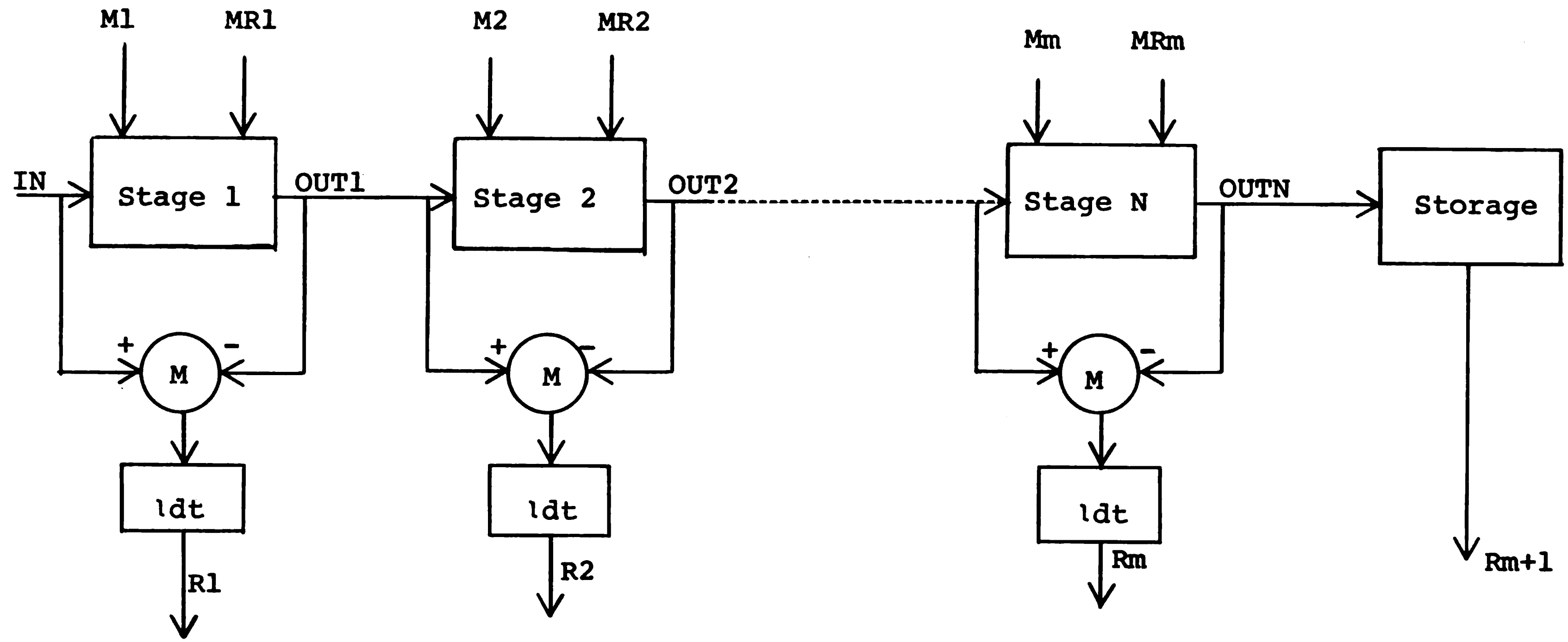


Figure 3.2.--Block Diagram of Packing Line.

mathematically functional. The definition of variables in the figure follows.

1,2,3,...n:	stage number.
IN:	input to the line (#*/min.).
OUT:	output rate of the stage (#*/min.).
R:	amount of accumulation in the reservoir (#*)
M:	number of machines (#).
MR:	number of machines which are running (#).
RMIN:	lower limit of reservoir accumulation (#*).
RMAX:	upper limit of reservoir accumulation (#*).
AF:	aggregation factor between stages, 20 cigarettes in a package, AF = 20 (#).
RPM:	output rate of machine (#/min.).

Following are mathematical notations of the packing system described in Figure 3.2.

1. $0 \leqq MR_{(i)} \leqq M_{(i)}$

2. $OUT_{(i)} = AF_{(1)} \times AF_{(2)} \cdot \cdot \cdot \times AF_{(i)}$

   $\times RPM_{(i)} \times MR_{(i)}$

3. $R_{(i)} = \int [OUT_{(i-1)} - OUT_{(i)}]\ dt$

---

*The unit is counted by the in-process product of the first stage.

4. $RMIN_{(i)} \leqq R_{(i)} \leqq RMAX_{(i)}$

where the subscript (i) identifies the stage number.

4. SIMULATION MODELING

In an attempt to reach the solution of the problem described in the previous section by showing block diagrams and mathematical notations, a simulation model will be formulated on a discrete operation in what follows with Figure 4.1.

At first, variables in Figure 4.1 are defined.

OUTM1: input rate to the stage (#*/min.).

OUT: output rate of the stage (#*/min.).

OUTP1: output rate of the next stage (#*/min.).

RM1: amount of accumulation in the reservoir of the immediately preceding stage (#*).

R: amount of accumulation in the reservoir of the stage (#*).

RP1: amount of accumulation in the reservoir of the next stage (#*).

M: number of machines in the stage (#).

MR: number of machines in the stage which are running (#).

RPM: output rate of machine (#/min.).

RM1MIN, RMIN, RP1MIN: lower limits of the reservoir accumulations, RM1, R, RP1, respectively (#*).

---

*The unit is counted by the in-process product of the first stage.

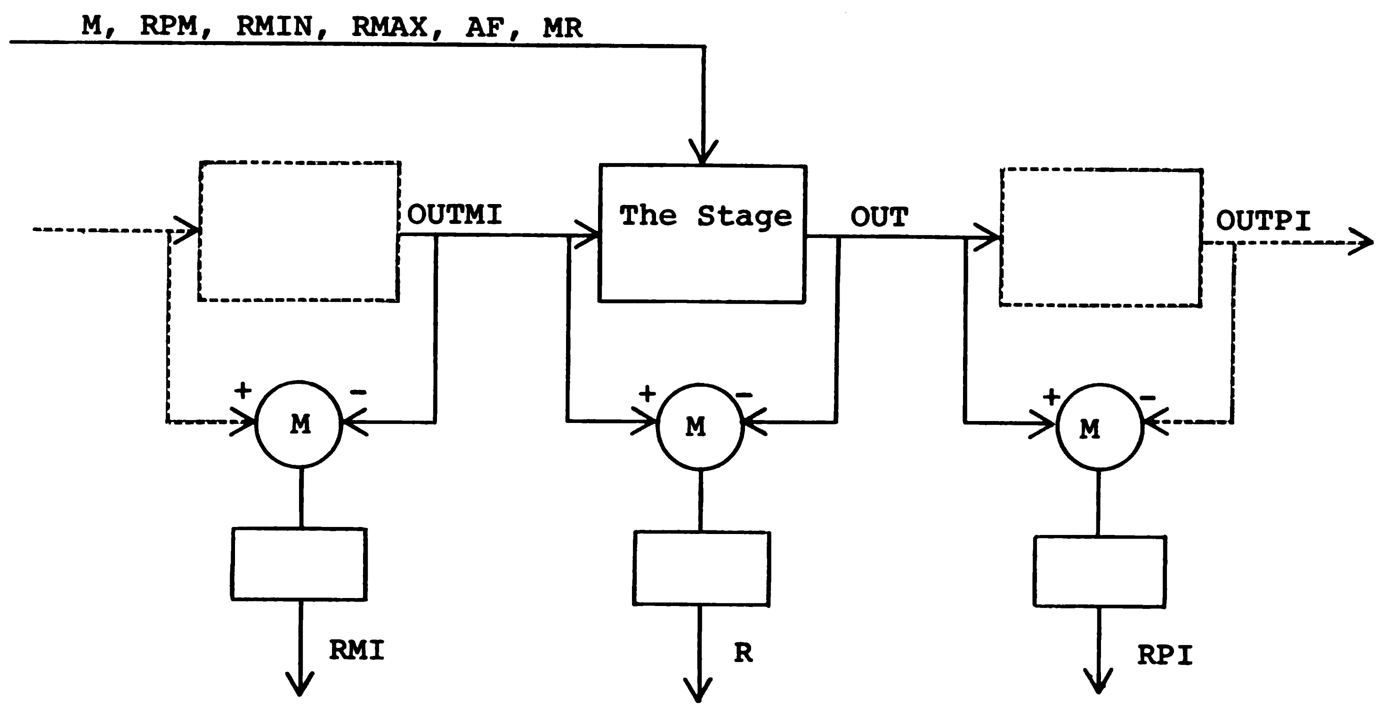


Figure 4.1.--Block Diagram for Simulation Model.

RM1MAX, RMAX, RP1MAX: upper limits of the reservoir accumulations, RM1, R, RP1, respectively (#*).

AF: package aggregation factor, 20 cigarettes in a package, AF = 20.

The simulation procedure starts with determining how many machines in the stage are running or should be allowed to run at any given moment. It is to give MR a particular value which is somewhere between zero and M-number of machines in the stage. This investigation to pick up untroubled machines proceeds this way.

At first,

1. If R < RMIN, MR = 0.

2. If RP1 > RP1MAX, MR = 0.

It means that when the reservoir accumulation of this stage is lower than the minimum level or when the accumulation of the next stage clears the maximum level, in either case or both, all the machines in this stage are halted; in other words, left idled. Eventually MR goes to zero.

The next step is about the other case. That is,

3. If RMIN ≦ R ≦ RMAX and

RP1MIN ≦ RP1 ≦ RP1MAX,

MR = ?

When the above conditions hold, there is no idling in the stage. Each machine is either running or breaking

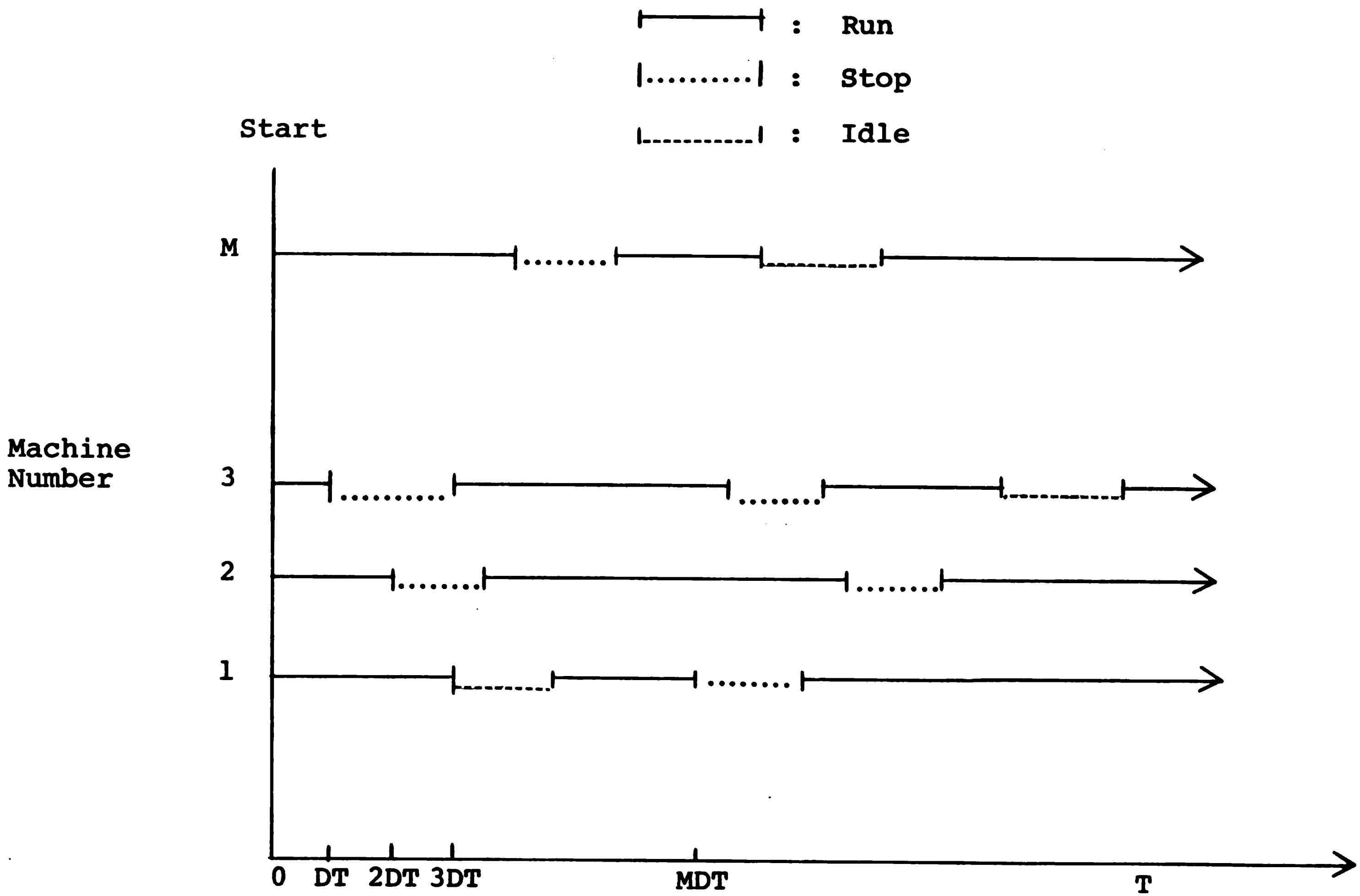
---

*The unit is counted by the in-process product of the first stage.

down and under recovery. MR should be the total number of healthy machines in the stage. The scanning to see which machine is running or breaking down will be done by getting a certain type of random numbers because an occurrence of machine breakdown is thought to be of a random nature. And it is reasonable to assume here that a breakdown is described by an exponential random distribution. More precisely, it can be said that the length of time between two consecutive breakdowns distributes in an exponentially random way, and so does the length of time of a breakdown.

What has to be actually done to this in a computation program is to generate two series of exponentially random numbers, of which one represents the length of time between breakdowns and the other is the breakdown period, and to assign these two kinds of random numbers on a hypothetical time axis one after another (see Figure 4.2). This time axis is thought to be a predetermined time schedule of a machine operation throughout a simulation period. Each machine has its own time schedule assigned by the exponentially random variables. What is more, those time schedules differ from simulation run to simulation run.

Thus the real activity of machines in a line can be put on a paper simulation program, based upon the reasonable assumption that a machine breakdown is



T: Simulation Time (min.)
DT: Unit Length of Simulation Time (min.)

Figure 4.2.--Machine Operation.

an exponentially random phenomenon. Once in a while, the period of idling mentioned before slips in the lineup on the time axis at any arbitrary time. That makes up a complete time schedule of a machine's operation as shown in Figure 4.2.

After completing those time schedules for all the machines in a line, all that is left to be done is to cut all those time axes at any given time and add up the running machines. This will give MR a specific value.

Following are mathematical notations of the simulation model. They are arranged on a discrete form.

1. $OUT(T) = TAF \times MR(T) \times RPM$

2. $R(T + DT) = R(T) + DT \times (OUTM1 - OUT)$

3. $TOUT(T + DT) = TOUT(T) + DT \times OUT(T)$

where

TAF: total aggregation factor, suppose the third stage, $TAF = AF(1) \times AF(2) \times AF(3)$ (#).

T: simulation time (min.).

DT: discrete time increment (min.).

R: amount of accumulation in the reservoir (#*).

TOUT: accumulated output of the stage (#*).

---

*The unit is counted by the in-process product of the first stage.

5. MODEL VALIDATION

Before a model is implemented or after in feedback, the validation of a model is a very important step to analyze and design an actual packing line. It is a process which shows that a model is a real and accurate representation of an actual system. A model at least should produce a set of reasonable outputs from a group of input data which consist of system parameters necessary to specify an actual system.

A practice of model validation involves a step of defining the extent to which the model can be applied to reality. Usually the problem of system designing requires a generalized model to produce many system alternatives from it. However, this is sometimes found to be an overcomplicated and impractical model to manipulate. A simple model is preferable to a complex one.

Figure 5.1 shows what is thought to be a general picture of the packaging line. It involves the concepts of stages, machines, reservoirs, and collector-distributors. Transfer mechanisms are placed between them in order to get the whole system organized.

In the graphic, what seems to be a unit sub-system is the two-stage configuration of

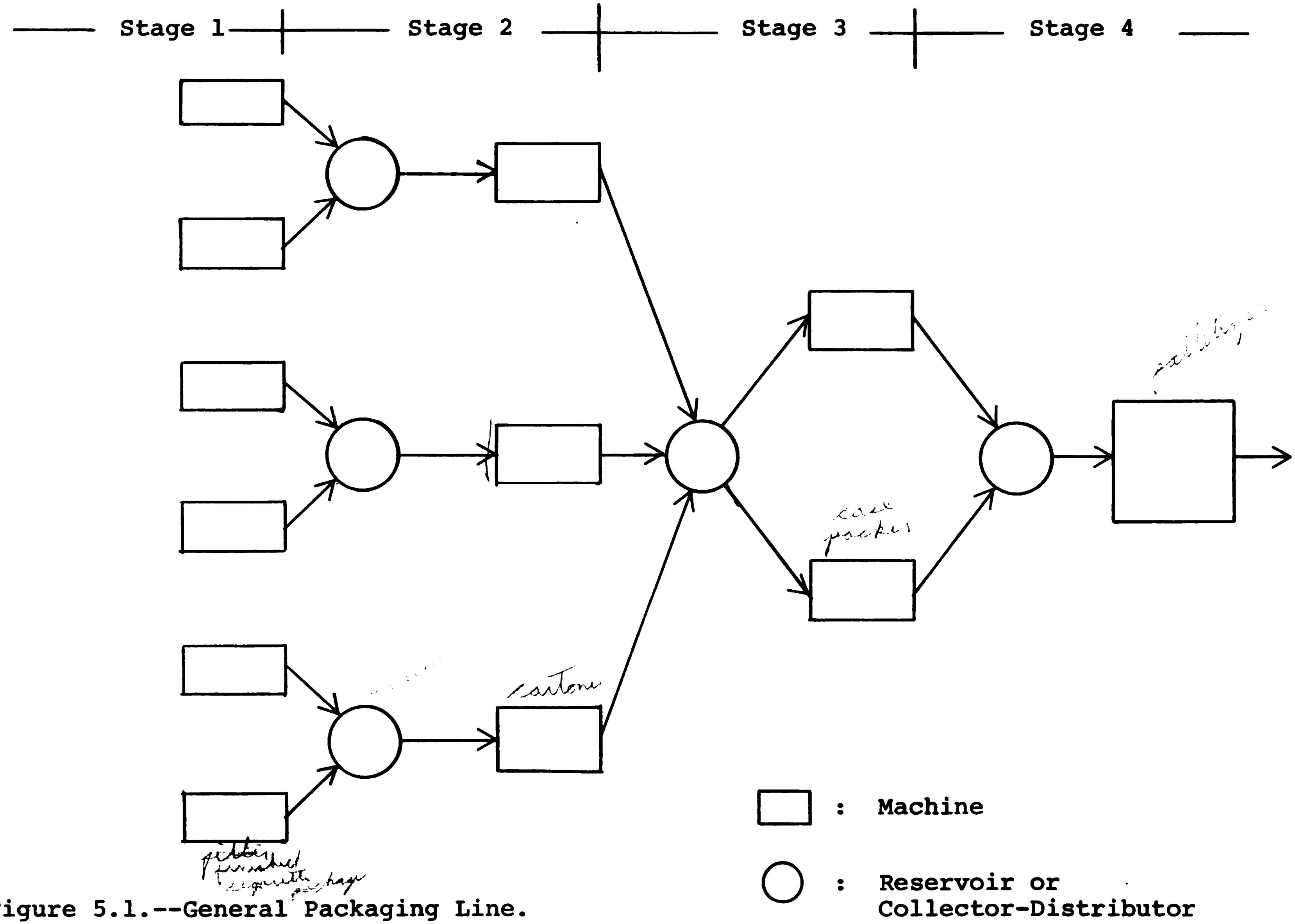


Figure 5.1.--General Packaging Line.

machines-to-machines with a reservoir or distributor-collector inside. It is shown in Figure 5.2. By utilizing the sub-system, the entire system of Figure 5.1 can be interpreted this way. At first, notice the beginning of the convergence structure of Figure 5.1, and replace what are thought to be units there with the unit sub-systems. This decreases the steps of convergence of the whole system by one. Then again apply the sub-system to the beginning of the one-step simplified system. By repeating this process, finally the entire system of Figure 5.1 will become to be describable by only one sub-system of Figure 5.2. This is a way that the simulation model could be utilized.

However, it should be said that although the model presented in this discussion exactly deals with the unit sub-system of Figure 5.2, the problem of simulating the generalized packaging system of Figure 5.1 will be left untouched.

Another thing which should also be taken into consideration is the fact that the model does not involve several other factors which may influence the line efficiency in reality, such as the rate of defects over the total production and the period of machine halts caused by rather routine work such as scheduled inspections for the quality control or periodical material bobbin

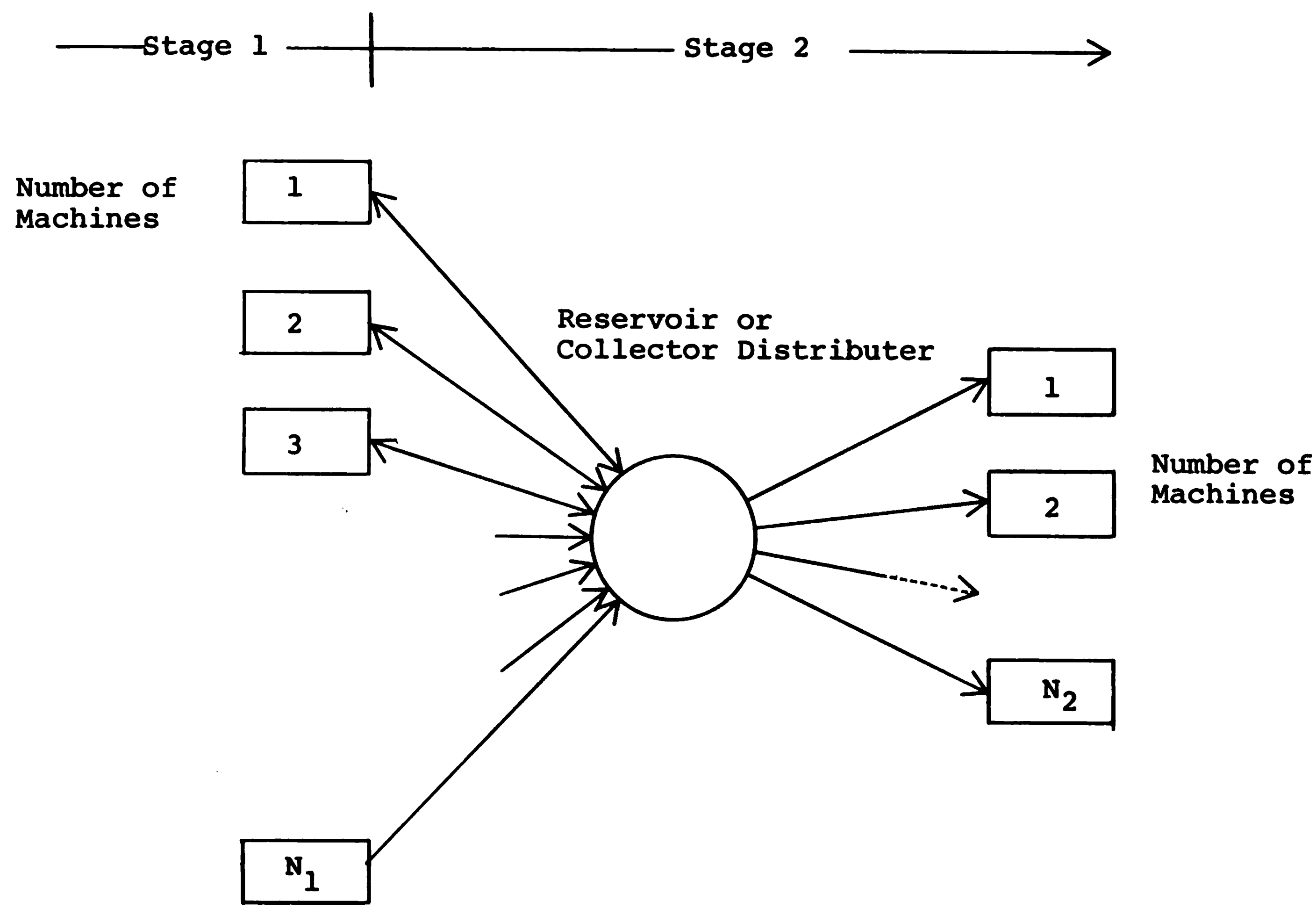


Figure 5.2.--Unit Sub System.

changes. They are thought to be inevitable in reality. However, those items are not supposed to have random natures for which the simulation techniques have to be invoked; they can be handled by analytical methods.

6. COMPUTER PROGRAMMING

6.1. Flow Chart

The computation process involves four major steps. The first step is to detect which machines in the quoted stage are running, out of operation or idling, and to determine how many machines are ready to operate in the stage at any given moment, that is, to give a specific value to MR which was a key variable to reach the solution of the model.

The second step is to go through all the stages from the first to last, utilizing the computation sub-routine explained as the first step in the preceding paragraph with an indexing parameter. Here the production by each stage for the time step is determined, and the respective resources are filled or depleted. The foregoing steps seem to be the essence of the computation process.

What follows next are numerous computation iterations within a simulation run from the starting point (time = 0) to any desired simulation length of time--say, 240 min.--with small time increments--say, 0.1 min. This is the third step which completes one simulation run.

Then the entire computation process up to the third step will be repeated enough times to come up with some statistics to evaluate the computation results.

This is the fourth step and it ends the whole simulation routine.

As mentioned before the essential parts of the computation process are step 1 and step 2--a subroutine and a main routine. Since they are both fairly complex steps, flow charts were drawn to show the process: Figures 6.1-1 and 6.1-2 for step 1, and Figures 6.2-1 and 6.2-2 for step 2. The flow charts correspond to what is mentioned in the previous chapter and the block diagrams, Figures 3.2 and 4.1.

First, variables in step 1 are explained as follows:

R, RMIN: amount of accumulation in the reservoir and its lower limit (#).

RP1, RP1MAX: amount of accumulation in the reservoir of the next stage and its upper limit (#).

T: time axis assigned to each machine. This time proceeds only when the machine is either running (RUN) or broken down (STOP), excluding idling period (IDLE) (min.).

TOT: Take Over Time. This is also a time axis assigned to each machine and determined by the length of time between two consecutive breakdowns (RUN) and the length of time of breakdowns (STOP) one after another. At any notch of the axis, the machine changes its state, RUN→STOP or STOP→RUN (min.).

IRS: Indicator of RUN (1) or STOP (0) of each machine at any time.

RA: uniform RAndom variable distributed between 0 and 1.

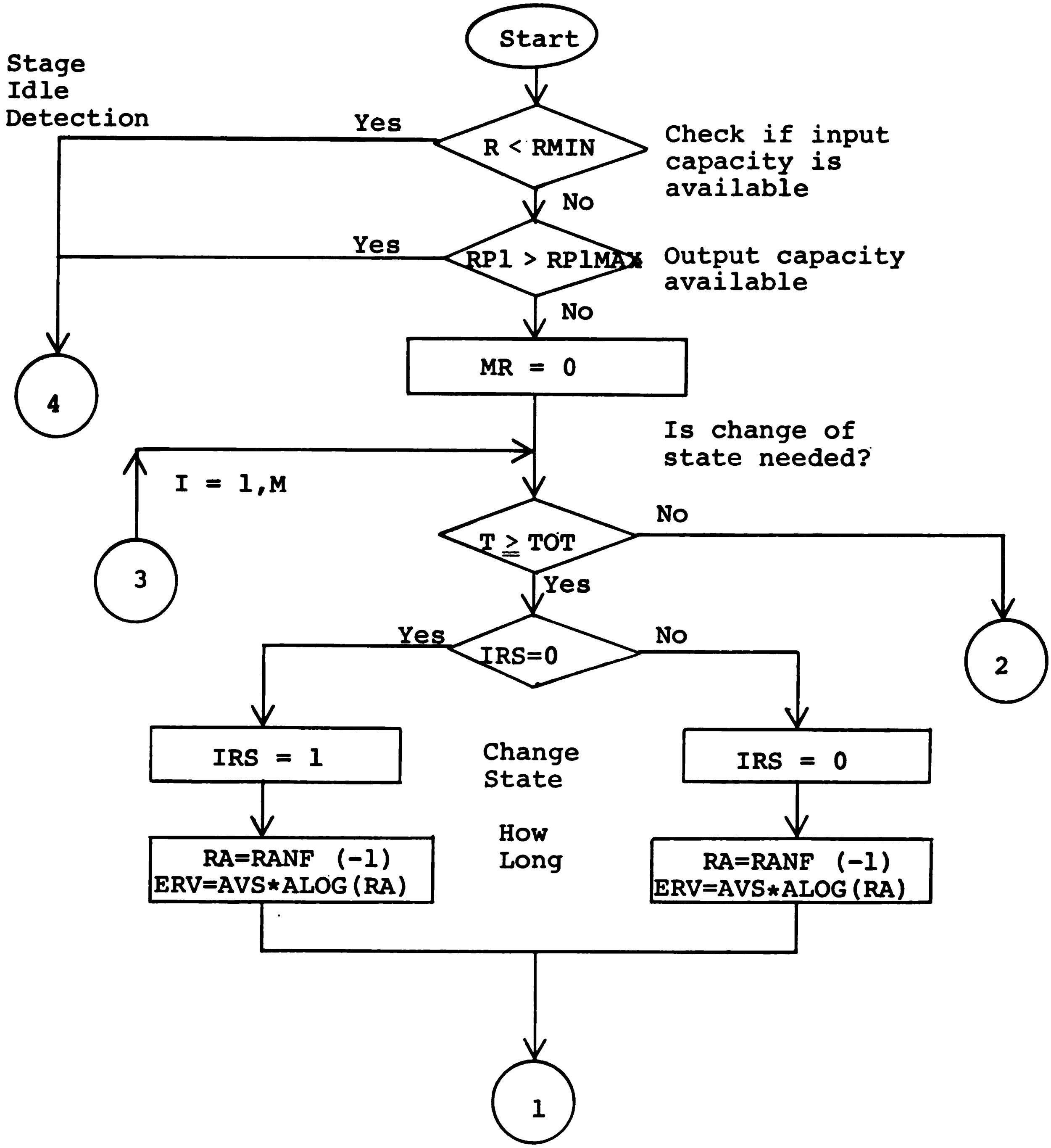


Figure 6.1-1.--Subroutine Runstop, Part 1.

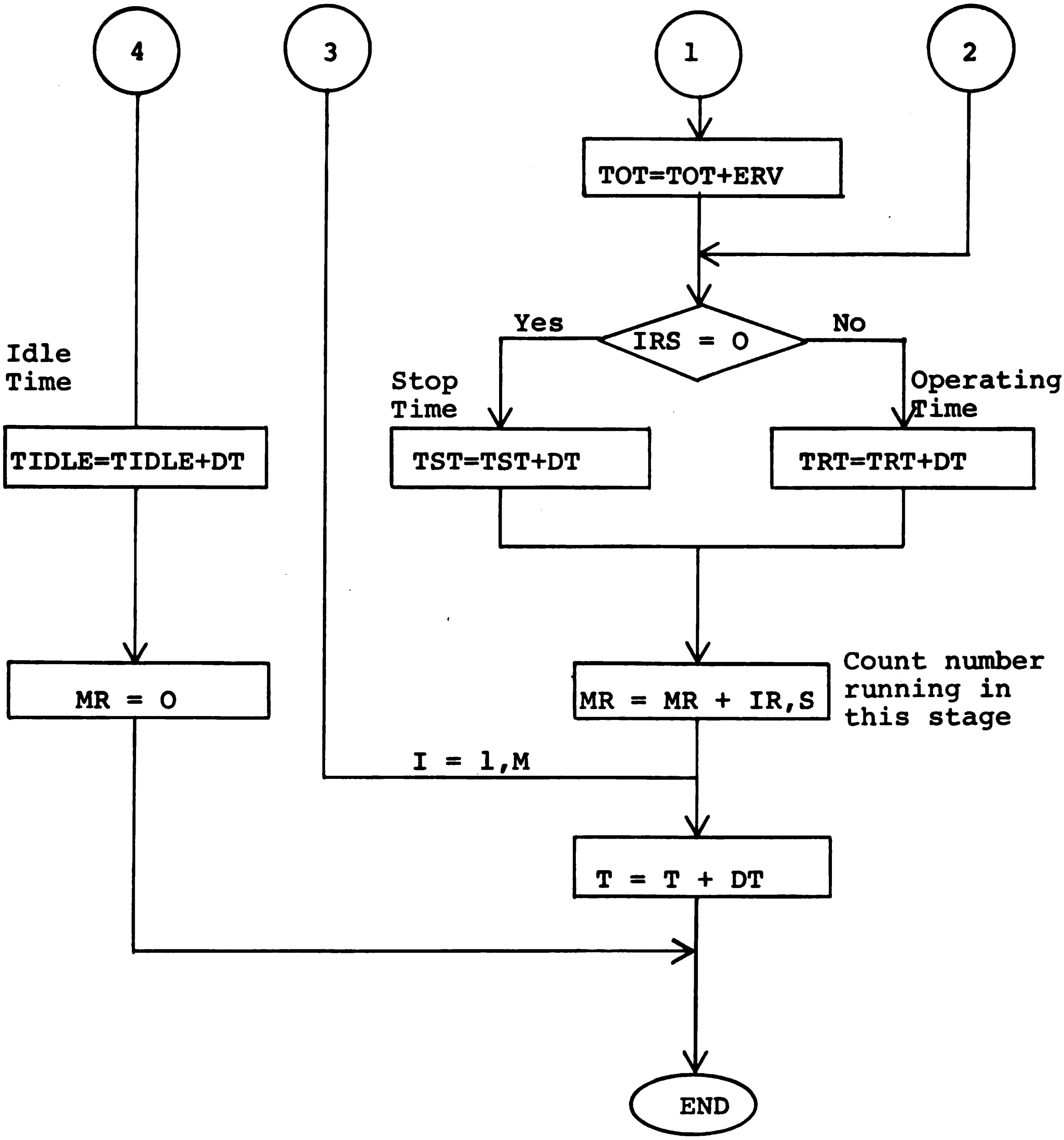


Figure 6.1-2.--Subroutine Runstp, Part 2.

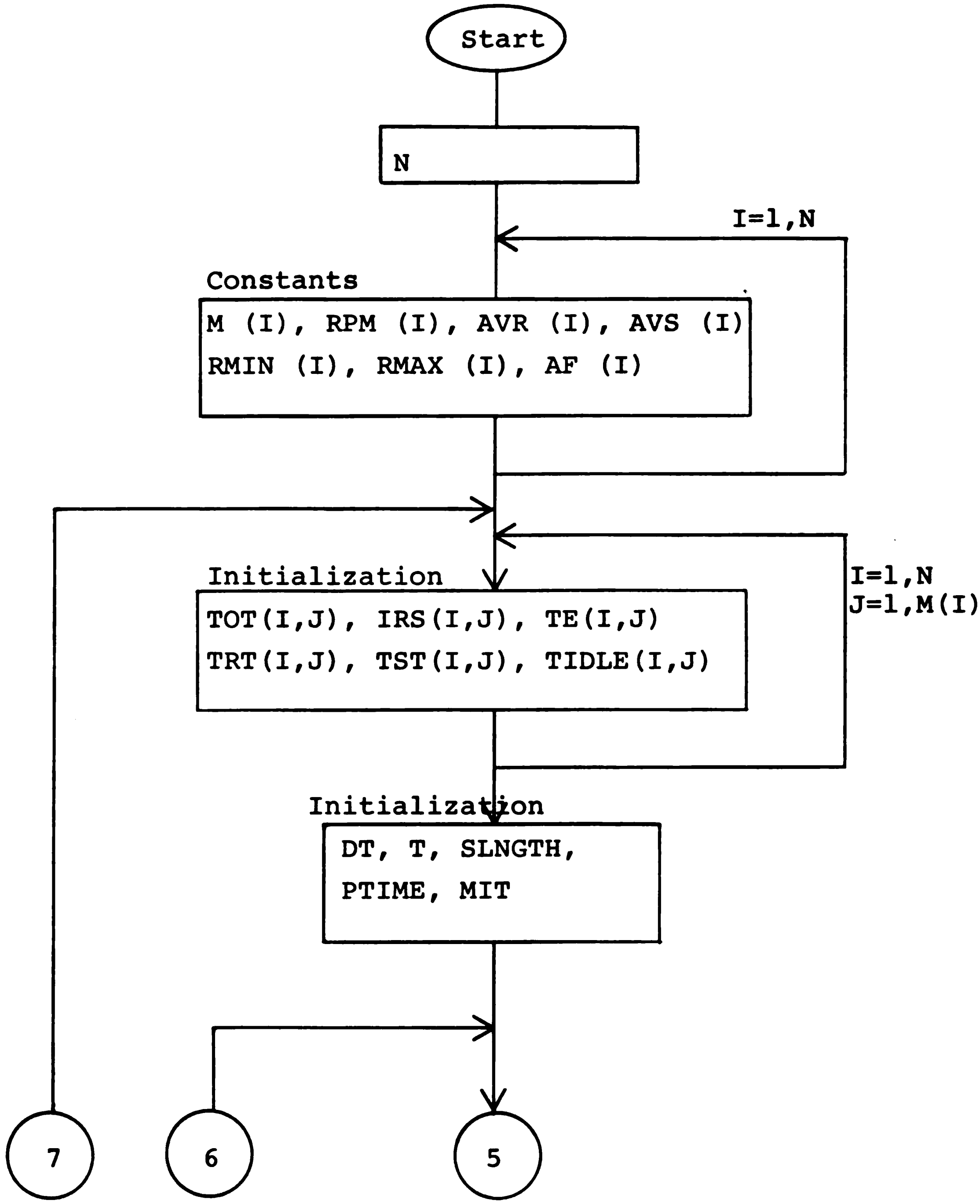


Figure 6.2-1.--Program Pline, Part 1.

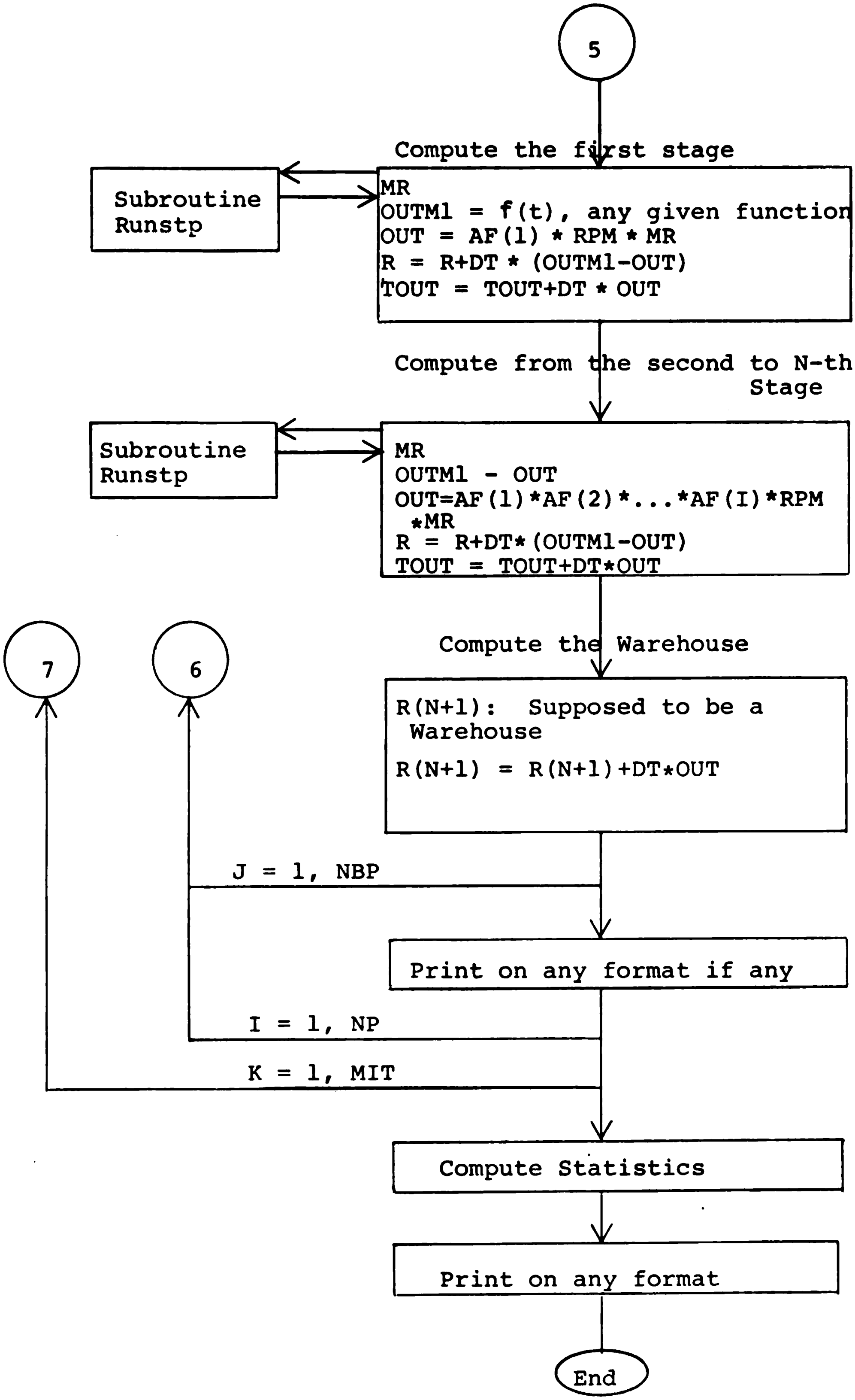


Figure 6.2-2.--Program Pline, Part 2.

AVR: AVerage length of time of RUN (min.).

AVS: AVerage length of time of STOP (min.).

ERV: Exponentially Random Variable which has AVR or AVS as an average.

TRT: Total length of RUN Time of each machine (min.).

TST: Total length of STOP Time of each machine (min.).

TIDLE: Total length of IDLE Time of each stage (min.).

Next, variables for the second step as shown in Figures 6.2-1 and 6.2-2 are defined below. This part is considered as the main routine of the computer program.

N: Number of stages.

$M_{(i)}$: number of Machines of i-th stage (#).

$RPM_{(i)}$: output rate of each machine of the i-th stage (#/min.).

$AVR_{(i)}$, $AVS_{(i)}$, $RMIN_{(i)}$, $RMAX_{(i)}$: same as mentioned in step 1. The subscript (i) indicates the i-th stage (min.).

$AF_{(i)}$: relative package Aggregation Factor across the i-th stage.

$TOT_{(i,j)}$, $IRS_{(i,j)}$, $TRT_{(i,j)}$, $TST_{(i,j)}$, $TIDLE_{(i,j)}$: same as mentioned in step 1. The subscripts (i,j) indicate the (j)-th machine in the (i)-th stage (min.).

$TE_{(i,j)}$: same as T mentioned in the step 1.

DT: Discrete Time increment of simulation (min.).

T: Time axis of simulation (min.).

SLNGTH: length of simulation (min.).

PTIME: In every PTIME minutes of simulation time, one print-out comes out (min.).

NBP: Number of simulation iterations Between two consecutive Print-outs.

NP: Number of times of Print-out.

MIT: Monte Carlo ITeration. Actually a number of iterations of an entire simulation run to come up with statistics.

6.2. An Example of a Program

Tables 6.1 and 6.2 show the listing of the computer program developed here. There are two parts in the program, the PROGRAM PLINE--a main routine--and the SUBROUTINE RUNSTP. It seems that the SUBROUTINE RUNSTP is a more general form which will find a broad range of use in many situations, while the PROGRAM PLINE will just fit the abstract model described in Figure 3.2.

For the purpose of activating the program, some assumptions and decisions have to be made. They are as follows:

(1) The behavior of each machine, that is, a series of repetitions of RUN-STOP, follows the exponentially random distribution.

(2) There are such huge storage spaces both just before the first stage and just after the final stage that the "before" storage cannot be out of stock and the "after" storage also cannot be filled up at any time. This assumption made it possible to consider that the INPUT to a line is always zero. All the

TABLE 6.1.--Program PLINE.

```
      PROGRAM PLINE(INPUT,OUTPUT)
      DIMENSION M(10),RPM(10),AVR(10),AVS(10),RMIN(11),RMAX(11),AF(10)
      DIMENSION R(10),TOUT(10)
      DIMENSION PR(11)
      DIMENSION TRAV(10),TSAV(10),EFF(10)
      DIMENSION EFFS(10),EFFI(10)
      DIMENSION STOREE(100,10),STORER(100,10),STORET(100)
      DIMENSION XBAREE(10),XBARR(10),STDEE(10),STDR(10)
      COMMON TRT(10,10),TST(10,10),TIDLE(10)
      COMMON TOT(10,10),IRS(10,10),TE(10,10)
C
C     TE = TIME AXIS OF EACH MACHINE
C     RPM = OUTPUT RATE OF EACH MACHINE
C     AF = AGGREGATION FACTOR OF EACH STAGE. EX. 20 CIGARETTES
C          INTO ONE PACKAGE.   AF=20
C     TOUT = TOTAL OUTPUT ACCUMULATED
C
C     INITIALIZATION PHASE,   N IS A NUMBER OF STAGES
C
      CALL RANSET(10.)
      READ 100,N
  100 FORMAT(I10)
      READ 110,(M(I),I=1,10),(RPM(I),I=1,10)
  110 FORMAT(5I10/5I10/5F10.0/5F10.0)
      READ 140,(AF(I),I=1,10)
  140 FORMAT(5F10.0/5F10.0)
      READ 120,(AVR(I),I=1,10),(AVS(I),I=1,10)
  120 FORMAT(5F10.2/5F10.2/5F10.2/5F10.2)
      READ 130,(RMIN(I),I=1,10),(RMAX(I),I=1,10)
  130 FORMAT(5E10.2/5F10.2/5F10.2/5F10.2)
      READ 150,(R(I),I=1,10)
  150 FORMAT(5E10.2/5F10.2)
      READ 400,DT,SLNGTH,PTIME
  400 FORMAT(3F10.4)
      READ 160,MIT
  160 FORMAT(I10)
      DO 30 I=1,10
      PR(I)=R(I)
   30 CONTINUE
C
      DO 20 K=1,MIT
      DO 1 J=1,10
      DO 2 I=1,10
      TOT(I,J)=0.0
      IRS(I,J)=0
      TE(I,J)=0.0
      TRT(I,J)=0.0
      TST(I,J)=0.0
```

```
    2 CONTINUE
    1 CONTINUE
      DO 4 I=1,10
      TIDLE(I)=0.0
      TOUT(I)=0.0
    4 CONTINUE
C
      NRP=PTIME/DT+0.1
      NP=SLNGTH/PTIME+0.1
      T=0.0
C
C     DT = DISCRETE TIME INCREMENTE
C     SLNGTH = SIMULATION PERIOD
C     PTIME = TIME INTERVAL FOR PRINTING
C     NRP = SIMULATION ITERATIONS BETWEEN PRINT OUTS
C     NP = PRINTING TIME
C
C     EXECUTION PHASE
C
      DO 5 L=1,NP
      DO 6 J=1,NRP
C     COMPUTE 1-ST STAGE
      I=1
      CALL RUNSTP(I,DT,M(I),AVR(I),AVS(I),R(I),R(I+1),RMIN(I),RMAX(I+1),
     +MR)
      OUTM1=0.0
      AFT=AF(I)
      OUT=AFT*RPM(I)*FLOAT(MR)
      R(I)=R(I)+DT*(OUTM1-OUT)
      TOUT(I)=TOUT(I)+DT*OUT
C     COMPUTE FROM 2-ND TO N-TH STAGE
      DO 7 I=2,N
      CALL RUNSTP(I,DT,M(I),AVR(I),AVS(I),R(I),R(I+1),RMIN(I),RMAX(I+1),
     +MR)
      OUTM1=OUT
      AFT=AFT*AF(I)
      OUT=AFT*RPM(I)*FLOAT(MR)
      R(I)=R(I)+DT*(OUTM1-OUT)
      TOUT(I)=TOUT(I)+DT*OUT
    7 CONTINUE
C
C     COMPUTE AFTER THE FINAL STAGE. ASSUME WEREHOUSE IS SO BIG
C     THAT IT CAN NOT BE FILLED UP
C
      R(N+1)=R(N+1)+DT*OUT
      T=T+DT
    6 CONTINUE
    5 CONTINUE
```

```
C
C     TRAV = AVERAGE RUN TIME OF MACHINES IN A CERTAIN STAGE
C     TSAV = AVERAGE STOP TIME OF MACHINES IN A CERTAIN STAGE
C     EFF = EFFICIENCY OF EACH STAGE
C     TEFF = TOTAL EFFICIENCY OF THE ENTIRE SYSTEM
C
C     COMPUTE SOME DECISION VARIABLES
C
      DO 10 II=1,10
      TRAV(II)=0.0
      TSAV(II)=0.0
      EFF(II)=0.0
      EFFS(II)=0.0
      EFFI(II)=0.0
   10 CONTINUE
      DO 11 II=1,N
      MM=M(II)
      DO 12 JJ=1,MM
      TRAV(II)=TRAV(II)+TRT(II,JJ)
      TSAV(II)=TSAV(II)+TST(II,JJ)
   12 CONTINUE
      TRAV(II)=TRAV(II)/FLOAT(M(II))
      TSAV(II)=TSAV(II)/FLOAT(M(II))
   11 CONTINUE
      DO 13 II=1,N
      EFF(II)=TRAV(II)/T*100.0
   13 CONTINUE
      DO 14 II=1,N
      EFFS(II)=TSAV(II)/(TSAV(II)+TRAV(II))*100.0
      EFFI(II)=TIDLE(II)/T*100.0
   14 CONTINUE
      TEFF=TOUT(N)/(T*RPM(1)*AF(1)*FLOAT(M(1)))*100.0
      NN=N+1
      DO 21 J=1,NN
      STOREF(K,J)=EFF(J)
      STORER(K,J)=R(J)
   21 CONTINUE
      STORET(K)=TEFF
   20 CONTINUE
C
C     COMPUTE MEAN AND STD. DEV.
C
      DO 22 J=1,10
      XBAREF(J)=0.0
      XBARR(J)=0.0
      STDEF(J)=0.0
      STDR(J)=0.0
   22 CONTINUE
```

```
      XBARTE=0.0
      STDTE=0.0
      DO 23 J=1,NN
      DO 24 K=1,MIT
      XBAREF(J)=XBAREF(J)+STOREF(K,J)
      XBARR(J)=XBARR(J)+STORER(K,J)
   24 CONTINUE
      XBAREF(J)=XBAREF(J)/FLOAT(MIT)
      XBARR(J)=XBARR(J)/FLOAT(MIT)
   23 CONTINUE
      DO 25 K=1,MIT
      XBARTE=XBARTE+STORET(K)
   25 CONTINUE
      XBARTE=XBARTE/FLOAT(MIT)
      DO 26 J=1,NN
      DO 27 K=1,MIT
      STDEF(J)=STDEF(J)+(STOREF(K,J)-XBAREF(J))**2
      STDR(J)=STDR(J)+(STORER(K,J)-XBARR(J))**2
   27 CONTINUE
      STDEF(J)=SQRT(STDEF(J)/FLOAT(MIT-1))
      STDR(J)=SQRT(STDR(J)/FLOAT(MIT-1))
   26 CONTINUE
      DO 28 K=1,MIT
      STDTE=STDTE+(STORET(K)-XBARTE)**2
   28 CONTINUE
      STDTE=SQRT(STDTE/FLOAT(MIT-1))
C
C     PRINTING PHASE
C
      DO 29 JJ=1,5
      PRINT 199
      PRINT 200,N
      PRINT 198
      PRINT 210,(MI(I),I=1,10),(RPM(I),I=1,10)
      PRINT 240,(AF(I),I=1,10)
      PRINT 220,(AVR(I),I=1,10),(AVS(I),I=1,10)
      PRINT 230,(RMIN(I),I=1,10),(RMAX(I),I=1,10)
      PRINT 250,(RR(I),I=1,10)
      PRINT 500,DT,SLNGTH,PTIME
      PRINT 260,MIT
      PRINT 299
      PRINT 300,T,(R(II),II=1,10)
      PRINT 340,(TOUT(II),II=1,10)
      PRINT 310,(TRAV(II),II=1,10)
      PRINT 320,(TSAV(II),II=1,10)
      PRINT 330,(TIDLE(II),II=1,10)
      PRINT 370,(EFFS(II),II=1,10)
      PRINT 380,(EFFI(II),II=1,10)
```

```
      PRINT 350,(EFF(II),II=1,10)
      PRINT 390,(IRS(II,1),II=1,10)
      PRINT 360,TEFF
      PRINT 600
      PRINT 550,(XBARR(J),J=1,10)
      PRINT 640,(STRR(J),J=1,10)
      PRINT 610,(XBAREF(J),J=1,10)
      PRINT 620,(STREF(J),J=1,10)
      PRINT 630,XBARTE,STDTE
      PRINT 720
      PRINT 700
  199 FORMAT(*1*,////////16X,*INPUT DATA*)
  200 FORMAT(*0*,18X,*T  *,*      N =*,I2,*        1        2        3
     +    4        5        6        7        8        9       10*)
  198 FORMAT(* *)
  210 FORMAT(* *,20X,*M*,10I9/28X,*RPM*,10F9.0)
  240 FORMAT(* *,28X,*AF*,10F9.0)
  220 FORMAT(* *,27X,*AVR*,10F9.2/28X,*AVS*,10F9.2)
  230 FORMAT(* *,26X,*RMIN*,10F9.3/27X,*RMAX*,10F9.3)
  250 FORMAT(* *,18X,*RESERVE(T=0)*,10F9.3)
  500 FORMAT(* *,28X,*DT*,F9.4/20X,*SIMU.LNGTH.*,F9.1/20X,
     +*PRINT CYCLE*,F9.1)
  260 FORMAT(* *,19X,*MONTE CARLO*,I9)
  299 FORMAT(*0*,15X,*COMPUTATION RESULTS*)
  300 FORMAT(*0*,10X,F10.1,*   RESERVE*,10F9.3)
  340 FORMAT(* *,20X,*     OUTPUT*,10F9.3)
  310 FORMAT(* *,20X,*        RUN*,10F9.1)
  320 FORMAT(* *,20X,*       STOP*,10F9.1)
  330 FORMAT(* *,20X,*       IDLE*,10F9.1)
  370 FORMAT(* *,20X,*  STOP RATE*,10F9.1)
  380 FORMAT(* *,20X,*  IDLE RATE*,10F9.1)
  350 FORMAT(* *,20X,*EFFICIENCY*,10F9.1)
  390 FORMAT(* *,19X,*RUN OR STOP*,10I9)
  760 FORMAT(* *,20X,*LINE EFFI.*,F9.1)
  600 FORMAT(*0*,15X,*MEAN AND STD. DEV. OF EFFICIENCY AND RESERVE*)
  550 FORMAT(*0*,15X,*MEAN OF RESERVE*,10F9.3)
  640 FORMAT(* *,21X,*STD. DEV.*,10F9.3)
  610 FORMAT(* *,17X,*MEAN OF EFFI.*,10F9.1)
  620 FORMAT(* *,21X,*STD. DEV.*,10F9.1)
  630 FORMAT(* *,15X,*MEAN LINE EFFI.*F9.1/22X,*STD. DEV.*,F9.1)
  720 FORMAT(*0*)
  700 FORMAT(* *,20X,*T = TIME (MIN)*/
     +21X,*N = NUMBER OF STAGES*/
     +21X,*M = NUMBER OF MACHINES*/
     +21X,*RPM = OUTPUT RATE OF MACHINE*/
     +21X,*AF = RELATIVE AGGREGATION FACTOR. IF 2 UNITS INTO*,
     +* 1 PACKAGE, AF = 2*/
     +21X,*AVR = MEAN TIME BETWEEN BREAK DOWNS*/
```

```
   +21X,*AVS = MEAN TIME OF BREAK DOWN*/
   +21X,*    AVR/(AVR+AVS) = CATALOGUE EFFICIENCY OF MACHINE*/
   +21X,*RMIN = LOWER LIMIT OF RESERVOIR CAPACITY*)
    PRINT 710
710 FORMAT(21X,*RMAX = UPPER LIMIT OF RESERVOIR CAPACITY*/
   +21X,*RESERVE(T=0) = INITIAL AMOUNT STORED IN RESERVOUR*/
   +21X,*DT = DISCRETE TIME INCREMENTE OF SIMULATION*/
   +21X,*SIMU.LNGTH. = SIMULATION LENGTH (MIN)*/
   +21X,*PRINT CYCLE = WHEN PRINTOUT IN DETAIL REQUIRED,UTILIZED*/
   +21X,*MONTE CARLO = NUMBER OF ITERATIONS FOR STATISTICAL ANALYSIS*/
   +21X,*RESERVE,OUTPUT = COUNTED BY THE UNIT OF STAGE 1*/
   +21X,*RUN = LENGTH OF TIME OF RUN (MIN)
   +*/21X,*STOP = LENGTH OF TIME OF STOP (MIN)*/
   +21X,*IDLE = LENGTH OF TIME OF IDLE (MIN)*/
   +21X,*STOP RATE = STOP/(STOP+RUN)*/
   +21X,*IDLE RATE = IDLE/T
   +*/21X,*EFFICIENCY = RUN/T
   +*/21X,38HLINE EFFI. = OUTPUT(N)/(M(1)*RPM(1)*T))
 29 CONTINUE
    END
```

TABLE 6.2.--Subroutine RUNSTP.

```
      SUBROUTINE RUNSTP(J,DT,M,AVR,AVS,R,RP1,RMIN,RP1MAX,MR)
      COMMON TRT(10,10),TST(10,10),TIDLE(10)
      COMMON TOT(10,10),IRS(10,10),T(10,10)
C
C
C     J = INDICATOR OF STAGE
C     DT = DISCRETE TIME INCREMENTE
C     M = NUMBER OF MACHINES
C     MR = NUMBER OF MACHINES RUNNING
C     AVR = AVERAGE INTERVAL OF ONE CONCECUTIVE RUNNING
C     AVS = AVERAGE INTERVAL OF ONE CONCECUTIVE STOPPING
C     R = RESERV AMOUNT
C     RP1 = RESERVE AMOUNT OF NEXT STAGE
C     RMIN = LOWER LIMIT OF RESERVE
C     RP1MAX = UPPER LIMIT OF RP1
C     TOT = NEXT TURN OVER TIME OF EACH MACHINE. RUN OR STOP
C     IRS = INDICATOR. RUNNING (1) OR STOPPING (0)
C     T = TIME OF EACH MACHINE. IT IS THE SUM OF RUNNING AND
C     STOPPING TIME.
C     TRT = TOTAL RUNNING TIME OF EACH MACHINE
C     TST = TOTAL STOPPING TIME OF EACH MACHINE
C     TIDLE = TOTAL IDLING TIME OF EACH STAGE
C
      IF(R.LT.RMIN) GO TO 5
      IF(RP1.GT.RP1MAX) GO TO 5
      MR=0
      DO 1 I=1,M
      IF(T(J,I).LT.TOT(J,I)) GO TO 4
      IF(IRS(J,I).EQ.0) GO TO 2
      IRS(J,I)=0
C
C     ERV = EXPONENTIALLY RANDOM VARIABLE, THIS PART WILL BE REPLACED,
C     IF OTHER DISTRIBUTIONS THAN EXPONENTIAL ARE REQUIRED
C
      RA=RANF(-1)
      ERV=-AVS*ALOG(RA)
      GO TO 3
    2 IRS(J,I)=1
      RA=RANF(-1)
      ERV=-AVR*ALOG(RA)
    3 CONTINUE
      TOT(J,I)=TOT(J,I)+ERV
    4 CONTINUE
      IF(IRS(J,I).EQ.0) GO TO 7
      TRT(J,I)=TRT(J,I)+DT
      GO TO 8
    7 TST(J,I)=TST(J,I)+DT
    8 CONTINUE
```

```
      MR=MR+TRS(J,I)
      T(J,I)=T(J,I)+DT
    1 CONTINUE
      GO TO 6
    5 CONTINUE
      TIDLE(J)=TIDLE(J)+DT
      MR=0
    6 CONTINUE
      RETURN
      END
```

necessary goods and materials are supplied from the preliminary storage house of infinite stock, as assumed before. And in the same manner, all the products coming out from the final stage will be placed in a warehouse of infinite capacity.

(3) The computation results have to be evaluated statistically. Therefore, in following the Monte Carlo mode, the simulation run is iterated more than several times and, for each iteration, a different series of uniform random numbers or a different part of the same series of uniform random numbers has to be used.

A series of five hundred uniform random numbers introduced to the computation in this discussion is listed in Table 6.3, and was generated by the computer system of Michigan State University.

Table 6.3- (0,1)UNIFORM RANDOM NUMBERS

.13	.63	.09	.85	.01	.66	.58	.63	.17	.50
.30	.98	.02	.07	.15	.37	.15	.37	.30	.71
.91	.41	.57	.52	.14	.20	.61	.51	.39	.01
.47	.52	.57	.86	.89	.02	.20	.79	.20	.08
.44	.35	.32	.83	.07	.94	.85	.64	.70	.04
.53	.71	.13	.13	.89	.93	.36	.59	.23	.71
.11	.41	.95	.27	.94	.79	.92	.61	.41	.08
.39	.98	.66	.14	.13	.74	.27	.84	.87	.74
.29	.31	.03	.87	.12	.03	.31	.76	.81	.63
.13	.34	.39	.81	.71	.06	.27	.26	.52	.80
.38	.19	.71	.32	.55	.25	.71	.16	.29	.81
.16	.57	.45	.83	.21	.12	.28	.01	.33	.04
.44	.08	.12	.74	.93	.54	.65	.30	.28	.69
.63	.77	.53	.58	.03	.00	.04	.47	.30	.60
.13	.02	.81	.91	.10	.23	.36	.43	.51	.43
.58	.18	.92	.77	.04	.42	.84	.80	.94	.02
.56	.31	.99	.73	.15	.44	.79	.78	.99	.03
.49	.94	.45	.45	.06	.90	.77	.32	.67	.68
.34	.94	.69	.46	.84	.95	.72	.16	.44	.55
.33	.68	.33	.98	.15	.30	.16	.47	.36	.99
.39	.64	.54	.51	.10	.49	.33	.09	.84	.08
.32	.47	.79	.73	.27	.55	.25	.80	.13	.53
.10	.23	.38	.03	.75	.71	.05	.09	.91	.07
.08	.02	.28	.32	.89	.58	.82	.34	.37	.33
.45	.42	.28	.04	.72	.13	.32	.91	.52	.11
.77	.55	.28	.44	.63	.46	.07	.78	.42	.38
.55	.41	.32	.57	.82	.34	.86	.87	.18	.43
.70	.67	.31	.94	.72	.17	.76	.63	.51	.73
.72	.06	.20	.34	.44	.53	.74	.44	.54	.50
.33	.97	.93	.45	.29	.54	.40	.90	.63	.23
.00	.64	.45	.28	.35	.30	.19	.04	.83	.33
.41	.62	.19	.12	.65	.09	.26	.99	.61	.32
.25	.62	.62	.45	.75	.35	.42	.36	.08	.82
.27	.67	.75	.45	.50	.37	.16	.17	.46	.21
.30	.67	.25	.46	.31	.28	.96	.84	.73	.80
.40	.38	.75	.43	.41	.37	.15	.21	.98	.45
.79	.22	.14	.55	.97	.34	.21	.17	.22	.86
.12	.52	.61	.65	.07	.71	.75	.06	.77	.47
.71	.55	.52	.04	.44	.37	.22	.40	.44	.12
.31	.62	.20	.29	.39	.77	.87	.73	.20	.06
.10	.60	.81	.07	.55	.43	.41	.34	.90	.03
.22	.64	.18	.07	.64	.54	.91	.45	.37	.57
.01	.70	.44	.57	.59	.99	.68	.44	.94	.77
.52	.24	.34	.43	.22	.19	.11	.26	.41	.57
.39	.21	.01	.30	.72	.42	.51	.07	1.00	.17
.04	.42	.72	.37	.15	.90	.12	.43	.13	.06
.57	.07	.73	.47	.22	.74	.73	.40	.89	.85
.56	.27	.17	.32	.39	.41	1.00	.78	.89	.95
.75	.17	.31	.72	.61	.37	.55	.66	.19	.23
.49	.21	.34	.27	.75	.54	.36	.49	.63	.45

7. PROOF OF THE COMPUTER PROGRAM

7.1. Proof Model 1

This model is a form of chain couplings of nine machines in a series as shown in Figure 7.1. Between machines, there are assumed to be huge reservoirs of infinite capacity. And in all those reservoirs, there are practically infinite amounts of packages available at the starting time of the simulation.

With these assumptions, although they are on a series formation, each machine can operate independently. In other words, there is no idling time in any machine's operation. Therefore, each machine is expected to yield its full catalog efficiency of operation and the line efficiency will equal that of the last machine.

A simulation result of the proof model is shown in Table 7.1. The first part of the computer print-out lists a set of the input data. The middle part is intended to show the computation results of one simulation run which is actually the last iteration of simulation runs in one computer run. The last part presents statistics--means and standard deviations--of several variables picked up from the list of computation results in the middle part of the print-out. The reason to do

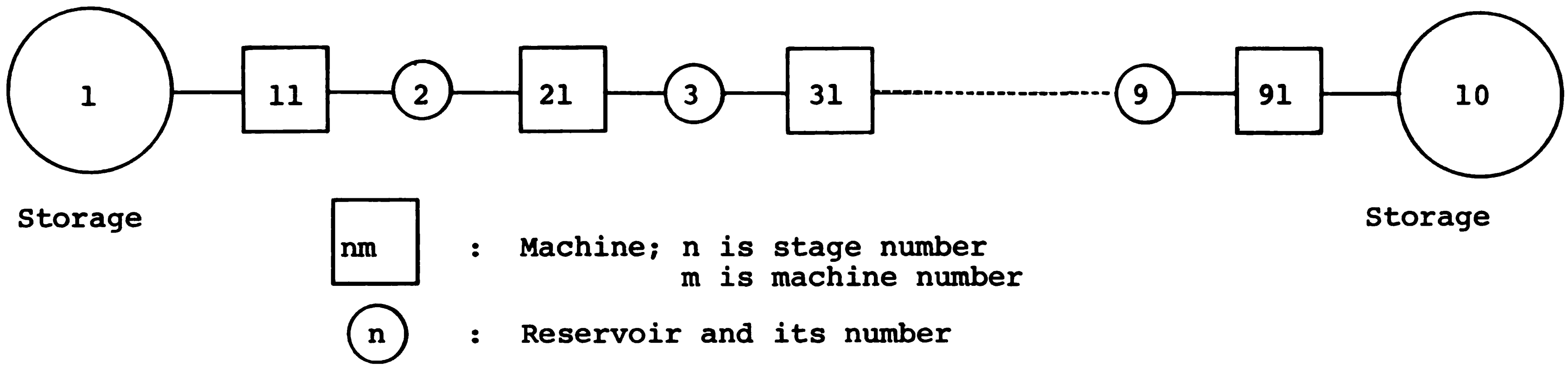


Figure 7.1.--Proof Model 1.

TABLE 7.1.--Proof Model 1.

INPUT DATA

T	N = 9	1	2	3	4	5	6	7	8	9	10
	M	1	1	1	1	1	1	1	1	1	-0
	RPM	320.	160.	160.	80.	80.	40.	40.	20.	20.	-.
	AF	1.	2.	1.	2.	1.	2.	1.	2.	1.	-.
	AVR	4.50	4.50	4.50	4.50	4.50	4.50	4.50	4.50	4.50	-.00
	AVS	.50	.50	.50	.50	.50	.50	.50	.50	.50	-.00
	RMIN	-.0	-.0	-.0	-.0	-.0	-.0	-.0	-.0	-.0	-.0
	RMAX	.100E+13	.100E+13	.100E+13	.100E+13	.100E+13	.100E+13	.100E+13	.100E+13	.100E+13	.100E+13
	RESERVE(T=0)	.100E+12	.100E+06	.100E+06	.100E+06	.100E+06	.100E+06	.100E+06	.100E+06	.100E+06	.100E+06
	DT	.1000									
	SIMU.LNGTH.	240.0									
	PRINT CYCLE	240.0									
	MONTE CARLO	10									

COMPUTATION RESULTS

T		1	2	3	4	5	6	7	8	9	10
240.0	RESERVE	.100E+12	.104E+06	.101E+06	.999E+05	.102E+06	.948E+05	.103E+06	.923E+05	.107E+06	.790E+06
	OUTPUT	.638E+05	.692E+05	.692E+05	.703E+05	.698E+05	.702E+05	.695E+05	.681E+05	.703E+05	.0
	RUN	215.0	216.4	216.1	219.8	218.0	219.5	217.3	212.8	219.6	.0
	STOP	25.0	23.6	23.9	20.2	22.0	20.5	22.7	27.2	20.4	.0
	IDLE	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
	STOP RATE	10.4	9.8	10.0	8.4	9.2	8.5	9.5	11.3	8.5	.0
	IDLE RATE	.0	.0	.0	.0	.0	.0	.0	.0	.0	.0
	EFFICIENCY	89.6	90.2	90.0	91.6	90.8	91.5	90.5	88.7	91.5	.0
	RUN OR STOP	1	1	1	0	1	1	1	1	1	0
	LINE EFFI.	91.5									

MEAN AND STD. DEV. OF EFFICIENCY AND RESERVE

	1	2	3	4	5	6	7	8	9	10
MEAN OF RESERVE	.100E+12	.102E+06	.101E+06	.101E+06	.100E+06	.976E+05	.100E+06	.960E+05	.105E+06	.479E+06
STD. DEV.	.211E+06	.230E+04	.104E+04	.836E+03	.206E+04	.276E+04	.200E+04	.301E+04	.365E+04	.208E+06
MEAN OF EFFI.	90.3	89.8	89.7	89.7	89.4	90.1	89.7	90.7	89.8	.0
STD. DEV.	2.1	1.3	1.3	2.0	1.4	1.0	1.5	1.5	2.0	.0
MEAN LINE EFFI.	89.8									
STD. DEV.	2.0									

T = TIME (MIN)
N = NUMBER OF STAGES
M = NUMBER OF MACHINES
RPM = OUTPUT RATE OF MACHINE
AF = RELATIVE AGGREGATION FACTOR. IF 2 UNITS INTO 1 PACKAGE, AF = 2
AVR = MEAN TIME BETWEEN BREAK DOWNS
AVS = MEAN TIME OF BREAK DOWN
AVR/(AVR+AVS) = CATALOGUE EFFICIENCY OF MACHINE
RMIN = LOWER LIMIT OF RESERVOIR CAPACITY
RMAX = UPPER LIMIT OF RESERVOIR CAPACITY
RESERVE(T=0) = INITIAL AMOUNT STORED IN RESERVOUR
DT = DISCRETE TIME INCREMENTE OF SIMULATION
SIMU.LNGTH. = SIMULATION LENGTH (MIN)
PRINT CYCLE = WHEN PRINTOUT IN DETAIL REQUIRED,UTILIZED
MONTE CARLO = NUMBER OF ITERATIONS FOR STATISTICAL ANALYSIS
RESERVE,OUTPUT = COUNTED BY THE UNIT OF STAGE 1
RUN = LENGTH OF TIME OF RUN (MIN)
STOP = LENGTH OF TIME OF STOP (MIN)
IDLE = LENGTH OF TIME OF IDLE (MIN)
STOP RATE = STOP/(STOP+RUN)
IDLE RATE = IDLE/T
EFFICIENCY = RUN/T
LINE EFFI. = OUTPUT(N)/(M(1)*RPM(1)*T)

this is only its economy. It is very costly to compute statistics of all the output variables.

Following are several comments about Table 7.1:

(1) The catalog efficiency of each machine was assumed to be 90% [AVR/(AVR + AVS) × 100 = 90].

(2) The efficiency of each machine excluding idling time after simulation can be figured out by the formula (100 - STOP RATE). In the computer outcome, this figure equals 90%, as expected.

(3) IDLE's are all zero as they should be.

(4) Standard deviations are small enough for the means to be counted on.

(5) Line efficiency is 90%, as expected.

After the above considerations, the computer program can be said to work properly on this model.

7.2. Proof Model 2

This model is exactly the same as model 1 except that there is no concept of reservoirs between machines at all (see Figure 7.2).

In this model, the interaction of machines is said to be 100%. That is, every breakdown of any one of the machines in the line definitely halts the operation of all other machines and puts them in the state of idling. With this idling time, it is expected that the efficiency of each machine in the line will decrease remarkably and so will the line efficiency.

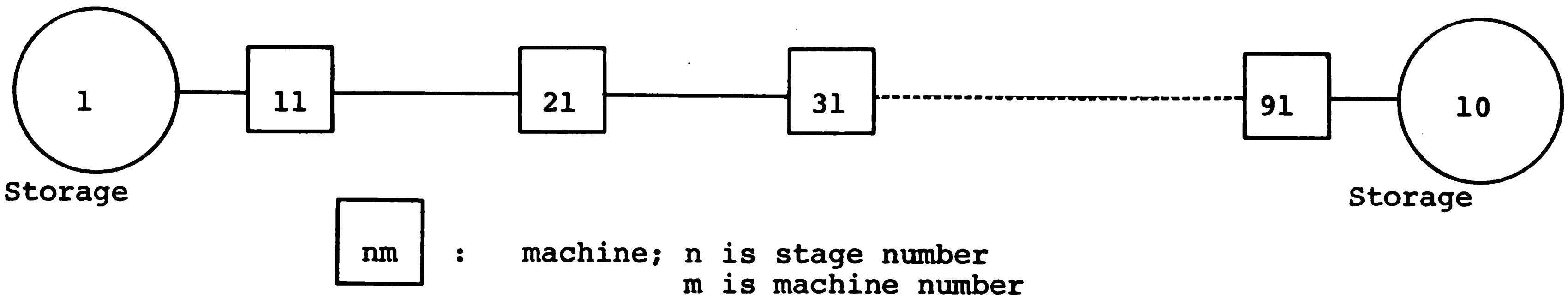


Figure 7.2.--Proof Model 2.

Table 7.2 is the computer print-out about this model. Some comments about the print-out follow.

(1) The catalog efficiency of each machine was assumed to be 90% [AVR/(AVR + AVS) × 100 = 90).

(2) According to the model assumptions, RMIN and RMAX are both zero. However, because of the discrete simulation operation, the following range should be allowed for RMIN and RMAX:

$$-320 \times DT < RMIN < 0$$

$$0 < RMAX < 320 \times DT$$

where 320 means the output rate of the first machine. At any instance when the reservoir accumulation exceeds those ranges, a certain machine is detected to have broken down by the computer program. Therefore, it should be said that there is a time lag DT between an occurrence of breakdown and the resulting halt of the immediate neighborhood machines, instead of halting at the same time as the assumption rules. This could cause the simulation error due to the discrete operation. But fortunately to this system model, this error never adds up. It is all cancelled out with the reverse time lag of restarting.

(3) The efficiency of each machine (100 - STOP RATE) has kept the level of 90% as it should be.

(4) IDLE's are remarkably large.

(5) Standard deviations are small enough.

Table 7.2.--Proof Model 2.

INPUT DATA

T	N = [illegible]	1	2	3	4	5	6	7	8	9	10
	M	1	1	1	1	1	1	1	1	1	-0
	PR	[illegible]	15?.	16?.	[illegible]	30.	4?.	40.	21.	29.	-.
	AF	1.	2.	1.	2.	1.	2.	1.	2.	1.	-.
	AVB	4.50	4.50	4.50	4.50	4.50	4.50	4.50	4.50	4.50	-.00
	AVR	.50	.50	.50	.50	.50	.50	.50	.50	.50	-.00
	RMIN	-.0	-.0	-.0	-.0	-.0	-.0	-.0	-.0	-.0	-.0
	RMAX	.1?0E+?1	.100E+01	.100E+01	.100E+01	.100E+01	.100E+01	.100E+01	.100E+01	.100E+01	.100E+13
	RESERVE(T=0)	.1??E+12	.0	.0	.0	.0	.0	.0	.0	.0	.0
	DT	[illegible]									
	SIMU.LNGTH.	[illegible]									
	PRINT CYCLE	24?.0									
	MONTE CARLO	10									

COMPUTATION RESULTS

T		1	2	3	4	5	6	7	8	9	10
24?.0	RESERVE	.1??E+12	.??E+0?	.0	.0	.0	.0	.0	.0	-.320E+02	.421E+06
	OUTPUT	.42?E+05	.42?E+05	.42?E+05	.428E+05	.428E+05	.428E+05	.428E+05	.428E+05	.428E+05	.0
	RUN	133.?	133.4	133.7	133.6	133.7	133.8	133.7	133.7	133.7	.0
	STOP	1?.7	1?.5	16.1	1?.7	7.9	17.?	13.7	15.9	12.5	.0
	IDLE	9?.9	9?.1	9?.4	9?.1	93.5	93.2	92.6	90.5	93.9	.0
	STOP RATE	11.?	11.6	10.8	9.7	5.5	11.3	9.3	10.6	9.5	.0
	IDLE RATE	3?.4	37.1	37.7	3?.4	41.0	37.2	38.6	37.7	39.1	.0
	EFFICIENCY	55.5	55.6	55.5	55.7	55.7	55.8	55.7	55.7	55.7	.0
	RUN OR STOP	1	1	1	1	1	1	1	1	1	0
	LINE EFFT.	5?.7									

MEAN AND STD. DEV. OF EFFICIENCY AND RESERVE

	1	2	3	4	5	6	7	8	9	10
MEAN OF RESERVE	.1??E+12	.??E+0?	.3??E+01	.5?3E+01	-.320E+01	.643E+01	-.640E+01	-.640E+01	-.150E+02	.228E+06
STD. DEV.	.12?E+06	.15?E+02	.18?E+02	.13?E+02	.236E+02	.203E+02	.232E+02	.202E+02	.226E+02	.129E+06
MEAN OF EFFT.	54.8	54.8	54.8	54.8	54.8	54.8	54.8	54.8	54.8	.0
STD. DEV.	2.0	2.0	2.0	2.0	2.0	2.0	2.0	1.9	1.9	.0
MEAN LINE EFFT.	54.8									
STD. DEV.	1.9									

T = TIME (MIN)
N = NUMBER OF STAGES
M = NUMBER OF MACHINES
PR = OUTPUT RATE OF MACHINE
AF = RELATIVE AGGREGATION FACTOR, IF 2 UNITS INTO 1 PACKAGE, AF = 2
AVB = MEAN TIME BETWEEN BREAK DOWNS
AVR = MEAN TIME OF BREAK DOWN
AVB/(AVB+AVR) = CATALOGUE EFFICIENCY OF MACHINE
RMIN = LOWER LIMIT OF RESERVOIR CAPACITY
RMAX = UPPER LIMIT OF RESERVOIR CAPACITY
RESERVE(T=0) = INITIAL AMOUNT STORED IN RESERVOIR
DT = DISCRETE TIME INCREMENT OF SIMULATION
SIMU.LNGTH. = SIMULATION LENGTH (MIN)
PRINT CYCLE = WHEN PRINTOUT IN DETAIL REQUIRED,UTILIZED
MONTE CARLO = NUMBER OF ITERATIONS FOR STATISTICAL ANALYSIS
RESERVE,OUTPUT = COUNTED BY THE UNIT OF STAGE 1
RUN = LENGTH OF TIME OF RUN (MIN)
STOP = LENGTH OF TIME OF STOP (MIN)
IDLE = LENGTH OF TIME OF IDLE (MIN)
STOP RATE = STOP/(STOP+RUN)
IDLE RATE = IDLE/T
EFFICIENCY = RUN/T
LINE EFFT. = OUTPUT(N)/(M(1)*PR(1)*T)

(6) EFFICIENCY and LINE EFFI are both remarkably low, about 55% with the standard deviation of 2%.

The mathematics procedure which is working in this proof model to calculate the line efficiency is described below.

Let:

T: simulation length,

$t_1, t_2, t_3, \ldots t_9$: length of breakdown time due to the first machine, second, third, . . . ninth machine, respectively, t is time that all machines are up,

$t = T - (t_1 + t_2 + t_3 + \cdots t_9)$,

$E_1, E_2, E_3, \ldots E_9$: efficiency of each machine,

LE: line efficiency.

Then:

$$(1)\ LE = \frac{t}{T}$$

$$(2)\ \frac{t}{t + t_1} = E_1,\ \frac{t}{t + t_2} = E_2,\ \ldots\ \frac{t}{t + t_9} = E_9.$$

Change the form of (2):

$$(3)\ 1 + \frac{t_1}{t} = \frac{1}{E_1},\ 1 + \frac{t_2}{t} = \frac{1}{E_2},$$

$$\ldots,\ 1 + \frac{t_9}{t} = \frac{1}{E_9}\ .$$

Add both sides of (3):

$$9 + \frac{t_1+t_2+\cdots+t_9}{t} = \frac{1}{E_1} + \frac{1}{E_2} + \cdots + \frac{1}{E_9}$$

$$9 + \frac{T-t}{t} = \frac{1}{E_1} + \frac{1}{E_2} + \cdots + \frac{1}{E_9}$$

$$\frac{T}{t} = \frac{1}{E_1} + \frac{1}{E_2} + \cdots + \frac{1}{E_9} - 8$$

Therefore the line efficiency can be written as:

$$(4) \quad LE = \frac{t}{T} = \frac{1}{1/E_1 + 1/E_2 + \cdots + 1/E_9 - 8}$$

This is an expression of line efficiency in terms of the catalog efficiency of each component machine. In this proof model, $E_1 = E_2 = \cdots = E_9 = 0.9$, so

$$LE = \frac{1}{9/0.9 - 8} = 0.5$$

Thus, from the mathematics, the line efficiency is figured out to be 50%.

In comparing these two solutions, it should be said that the analytical result is significantly different from that of simulation, in taking such a small value of standard deviation as 1.9 into consideration.

The reason that made this difference seems to be this. Since the simulation is discrete, there must be a time lag of DT at the smallest between a breakdown of one machine and the resulting halts of the immediate neighboring machines, contrary to the fact that the assumption of the proof model rules the same instant's stop. In the extreme, it can be seen that the time lag gets as large as 8 DT (0.8 min.) from a breakdown of the first machine to the resulting halt of the ninth machine in this model. There is a good possibility that another machine goes into trouble during the time lag of 0.8 minutes. This means that two machines are out of operation at the same time, and it is what the computation result tells. Take the IDLE time of the first machine in the model, for instance. It is 89.8 (min.), while the summation of STOP time from the second machine to the ninth machine is 17.5 + 16.1 + 14.3 + · · · + 12.5 = 114.7 (min.). These two figures were supposed to be equal from the assumption of this proof model, but they were not, in the simulation model.

What made the difference seems to be explainable as follows: Suppose at time T, the first machine broke down. And two DT's later, the fourth through ninth machines are still supposed to be working; the eighth machine was also shut down. One more DT later, the breakdown of the eighth machine puts the seventh and ninth machines on the list of idling, as well as the first machine's breakdown resulting in the idling of the second to fifth machines by that time. And one DT after that, the remaining sixth machine will finally be forced to enter the state of idling. The important thing to say here is that the computer program counts the eighth machine as the STOP, not the IDLE as the proof model rules, in this situation. This increases the length of STOP in the fixed simulation length. The increase of the STOP time also lengthens the RUN period, because the ratio of RUN/(RUN + STOP) is kept at the level of 0.9. The longer RUN time in the fixed operation length means the higher operational efficiency of line.

Therefore, although it would be another story if DT gets very small, it is safer to say here that the simulation program does not work on this kind of model which is characterized by reservoirs of strict zero capacity.

However, this result implies another application of this simulation program to reality. That is, for the case in which the reservoir capacity is larger than zero and smaller than infinite (practically), the program seems to work.

7.3. Proof Model 3

The input data in Table 7.3 specify this model. Features are:

(1) 2-stage line (see Figure 7.3).

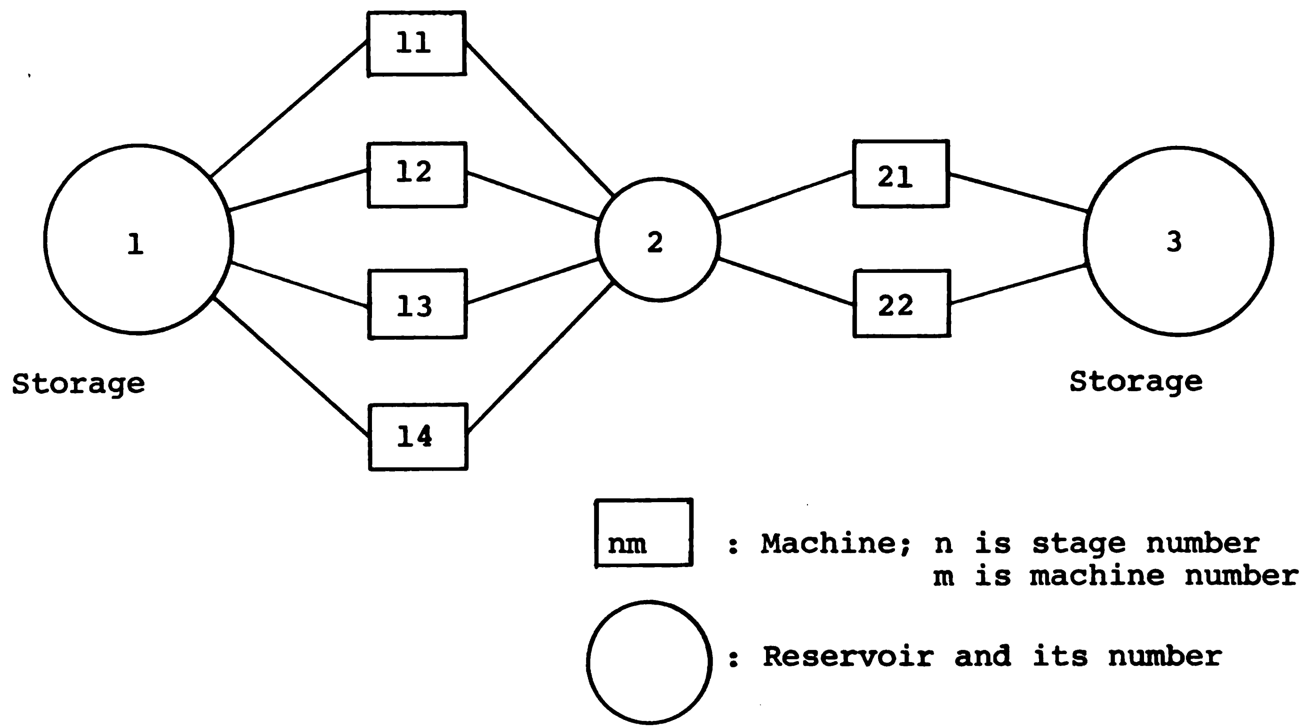


Figure 7.3.--Proof Model 3 and 4.

Table 7.3--Proof Model 3.

INPUT DATA

T	N = 2	1	2	3	4	5	6	7	8	9	10
	M	4	2	-0	-0	-0	-0	-0	-0	-0	-0
	ROM	[illegible]	[illegible]	-0.	-0.	-0.	-0.	-0.	-0.	-0.	-0.
	AF	1.	[illegible]	-0.	-0.	-0.	-0.	-0.	-0.	-0.	-0.
	AVB	2.[illegible]	2.[illegible]	-0.00	-0.00	-0.00	-0.00	-0.00	-0.00	-0.00	-0.00
	AVS	1.[illegible]	1.[illegible]	-0.00	-0.00	-0.00	-0.00	-0.00	-0.00	-0.00	-0.00
	RMIN	-.0	-.0	-.0	-.0	-.0	-.0	-.0	-.0	-.0	
	RMAX	.1[illegible]E+1[illegible]	[illegible]	.[illegible]E+[illegible]	-.0	-.0	-.0	-.0	-.0	-.0	
	RESERVE(T=0)	.1[illegible]E+12	[illegible]	.5[illegible]E+[illegible]	-.0	-.0	-.0	-.0	-.0	-.0	
	DT	.1000									
	SIMU.LNGTH.	240.0									
	PRINT CYCLE	[illegible]									
	MONTE CARLO	[illegible]									

COMPUTATION RESULTS

		1	2	3	4	5	6	7	8	9	10
240.0	RESERVE	.1[illegible]E+12	.[illegible]E+[illegible]	.[illegible]E+[illegible]	-.0	-.0	-.0	-.0	-.0	-.0	
	OUTPUT	.[illegible]E+[illegible]7	.[illegible]E+[illegible]7	.0	.0	.0	.0	.0	.0	.0	
	RUN	2[illegible]4.0	[illegible]	.0	.0	.0	.0	.0	.0	.0	.0
	STOP	[illegible]	[illegible]	.0	.0	.0	.0	.0	.0	.0	.0
	IDLE	[illegible]	[illegible]	.0	.0	.0	.0	.0	.0	.0	.0
	STOP RATE	1[illegible]	[illegible]	.0	.0	.0	.0	.0	.0	.0	.0
	IDLE RATE	[illegible]	[illegible]	.0	.0	.0	.0	.0	.0	.0	.0
	EFFICIENCY	[illegible]	8[illegible]	.0	.0	.0	.0	.0	.0	.0	.0
	RUN OR STOP	1	1	0	0	0	0	0	0	0	0
	LINE EFFI.	[illegible]									

MEAN AND STD. DEV. OF EFFICIENCY AND RESERVE

	1	2	3	4	5	6	7	8	9	10
MEAN OF RESERVE	.1[illegible]E+12	.1[illegible]E+[illegible]	.1[illegible]E+03	.0	.0	.0	.0	.0	.0	
STD. DEV.	.[illegible]E+[illegible]	.[illegible]E+[illegible]	.[illegible]E+[illegible]	.0	.0	.0	.0	.0	.0	
MEAN OF EFFI.	[illegible]	[illegible]	.0	.0	.0	.0	.0	.0	.0	.0
STD. DEV.	1.5	1.[illegible]	.0	.0	.0	.0	.0	.0	.0	.0
MEAN LINE EFFI.	[illegible]									
STD. DEV.	1.[illegible]									

T = TIME (MIN)
N = NUMBER OF STAGES
M = NUMBER OF MACHINES
ROM = OUTPUT RATE OF MACHINE
AF = RELATIVE AGGREGATION FACTOR, IF 2 UNITS INTO 1 PACKAGE, AF = 2
AVB = MEAN TIME BETWEEN BREAK DOWNS
AVS = MEAN TIME OF BREAK DOWN
AVB/(AVB+AVS) = CATALOGIC EFFICIENCY OF MACHINE
RMIN = LOWER LIMIT OF RESERVOIR CAPACITY
RMAX = UPPER LIMIT OF RESERVOIR CAPACITY
RESERVE(T=0) = INITIAL AMOUNT STORED IN RESERVOIR
DT = DISCRETE TIME INCREMENT OF SIMULATION
SIMU.LNGTH. = SIMULATION LENGTH (MIN)
PRINT CYCLE = WHEN PRINTOUT IN DETAIL REQUIRED,UTILIZED
MONTE CARLO = NUMBER OF ITERATIONS FOR STATISTICAL ANALYSIS
RESERVE,OUTPUT = COUNTED BY THE UNIT OF STAGE 1
RUN = LENGTH OF TIME OF RUN (MIN)
STOP = LENGTH OF TIME OF STOP (MIN)
IDLE = LENGTH OF TIME OF IDLE (MIN)
STOP RATE = STOP/(STOP+RUN)
IDLE RATE = IDLE/T
EFFICIENCY = RUN/T
LINE EFFI. = OUTPUT(N)/(M(1)*ROM(1)*T)

(2) 4 machines in the first stage and 2 in the second.

(3) Output rates of machines are 4,000 and 400.

(4) 20 units of the first-stage product is packed into one package by the second stage. AF = 20.

(5) The catalog efficiency of the first-stage machine is 85%, while the second-stage machine claims 90%.

(6) There is a reservoir between stages. Its capacity is 100,000 units which is slightly larger than the production amount of 6 minutes of operation of the line.

(7) At the starting point, the reservoir is half full.

(8) Simulation length is 240 minutes with a time increment of 0.1 minutes.

(9) Iteration for statistics is 10.

Some remarks about the computation result (Table 7.3) follow.

(1) The expected production rate of the first stage is 4 × 4,000 × 0.85 = 13,600 (#/min.), while the second stage has the capacity to produce 2 × 400 × 20 × 0.9 = 14,400 (#/min). Therefore the line efficiency is expected to be 85%--the "bottle-neck" efficiency, at its maximum.

(2) The computer output says that the line efficiency is already at the maximum point--0.85.

(3) The accumulation amount in the reservoir is averaged by 11,240 units with the standard deviation of 8,983 units.

(4) Because of the fact that the total simulation length is long enough--10 × 240 = 2,400 minutes--the line system is thought to be already in steady state. Therefore the above figures in (3) can specify the maximum capacity of the reservoir. That is, 11,240 + 3 × 8,983 = 38,000 units.

(5) In other words, if there is a reservoir which can contain up to 38,000 units, the line yields its maximum production rate.

(6) If the capacity is smaller than 38,000, it gives the first stage the idling time which decreases the line production rate.

7.4. Proof Model 4

This model (see Figure 7.3) is almost the same as model 3. The only difference from it is the reservoir capacity which is assumed to be 1,000 units for this model. The capacity of model 3 was 100,000 units.

With the capacity of 1,000 units, the reservoir can hold 0.25 min. (1,000/4,000) at the maximum, in case any one machine out of four machines in the first stage breaks down. If any one of two machines in the second stage shuts down, the reservoir holds 1,000/(400 × 20) = 0.125 min.

The simulation result for this particular model is seen in Table 7.4. In the table, the line efficiency

Table 7.4.--Proof Model 4.

INPUT DATA

T	N = 2	1	2	3	4	5	6	7	8	9	10
	M	4	2	-0	-0	-0	-0	-0	-0	-0	-0
	RPM	4000.	400.	-.	-.	-.	-.	-.	-.	-.	-.
	AF	1.	20.	-.	-.	-.	-.	-.	-.	-.	-.
	AVT	9.90	9.00	-.00	-.00	-.00	-.00	-.00	-.00	-.00	-.00
	AVS	1.00	1.00	-.00	-.00	-.00	-.00	-.00	-.00	-.00	-.00
	RMIN	-.0	-.0	-.0	-.0	-.0	-.0	-.0	-.0	-.0	-.0
	RMAX	.100E+13	.100E+14	.100E+17	-.0	-.0	-.0	-.0	-.0	-.0	-.0
	RESERVE(T=0)	.100E+12	.100E+03	.100E+06	-.0	-.0	-.0	-.0	-.0	-.0	-.0
	DT	.100									
	SIMU.LNGTH.	240.0									
	PRINT CYCLE	240.0									
	MONTE CARLO	[illegible]									

COMPUTATION RESULTS

T		1	2	3	4	5	6	7	8	9	10
240.0	RESERVE	.100E+12	.101E+03	.270E+08	-.0	-.0	-.0	-.0	-.0	-.0	-.0
	OUTPUT	.264E+07	.264E+07	.0	.0	.0	.0	.0	.0	.0	.0
	RUN	164.8	154.8	.0	.0	.0	.0	.0	.0	.0	.0
	STOP	25.8	23.9	.0	.0	.0	.0	.0	.0	.0	.0
	IDLE	48.4	45.2	.0	.0	.0	.0	.0	.0	.0	.0
	STOP RATE	14.0	14.0	.0	.0	.0	.0	.0	.0	.0	.0
	IDLE RATE	20.2	19.3	.0	.0	.0	.0	.0	.0	.0	.0
	EFFICIENCY	68.7	68.7	.0	.0	.0	.0	.0	.0	.0	.0
	RUN OR STOP	1	1	0	0	0	0	0	0	0	0
	LINE EFFI.	68.7									

MEAN AND STD. DEV. OF EFFICIENCY AND RESERVE

	1	2	3	4	5	6	7	8	9	10
MEAN OF RESERVE	.100E+12	.265E+03	.148E+08	.0	.0	.0	.0	.0	.0	.0
STD. DEV.	.822E+07	.381E+03	.822E+07	.0	.0	.0	.0	.0	.0	.0
MEAN OF EFF.	70.2	70.2	.0	.0	.0	.0	.0	.0	.0	.0
STD. DEV.	2.0	2.0	.0	.0	.0	.0	.0	.0	.0	.0
MEAN LINE EFFI.	70.2									
STD. DEV.	2.0									

T = TIME (MIN)
N = NUMBER OF STAGES
M = NUMBER OF MACHINES
RPM = OUTPUT RATE OF MACHINE
AF = RELATIVE AGGREGATION FACTOR, IF 2 UNITS INTO 1 PACKAGE, AF = 2
AVT = MEAN TIME BETWEEN BREAK DOWNS
AVS = MEAN TIME OF BREAK DOWN
AVT/(AVT+AVS) = CATALOGUE EFFICIENCY OF MACHINE
RMIN = LOWER LIMIT OF RESERVOIR CAPACITY
RMAX = UPPER LIMIT OF RESERVOIR CAPACITY
RESERVE(T=0) = INITIAL AMOUNT STORED IN RESERVOIR
DT = DISCRETE TIME INCREMENTS OF SIMULATION
SIMU.LNGTH. = SIMULATION LENGTH (MIN)
PRINT CYCLE = WHEN PRINTOUT IN DETAIL REQUIRED,UTILIZED
MONTE CARLO = NUMBER OF ITERATIONS FOR STATISTICAL ANALYSIS
RESERVE,OUTPUT = COUNTED BY THE UNIT OF STAGE 1
RUN = LENGTH OF TIME OF RUN (MIN)
STOP = LENGTH OF TIME OF STOP (MIN)
IDLE = LENGTH OF TIME OF IDLE (MIN)
STOP RATE = STOP/(STOP+RUN)
IDLE RATE = IDLE/T
EFFICIENCY = RUN/T
LINE EFFI. = OUTPUT(N)/(M(1)*RPM(1)*T)

is found to be 70.2% with the standard deviation of 2.0%, while in model 3, they were 85.0% and 1.8%, respectively.

8. SUMMARY

It is thought that a packing line is one which is made up of several packing stages or stations which are laid out and spanned toward each other in a certain formation to work as a system. Each stage involves a number of machines which are unique to the stage, and between stages or between machines there must be some sort of transfer mechanisms such as basic transfer conveyors--a power-and-free type or indexing type--or a reservoir mechanism. All those are combined to build a packing line or system.

When a line begins its operation, stages as components of the line and machines as components of the stage also start to play their roles in the line, suffering from the interactional restriction to their movements. What is the interactional restriction? It causes the idling time to machines. In general, each machine as its own operational efficiency--say, catalog efficiency--which is thought to be the expected value of efficiency when the machine is operated alone. But when a machine is placed in a line, it cannot yield the catalog efficiency, because the machine is forced to have a certain length of idling time which is the waiting time for the neighborhood machines in the line recovering from a breakdown. And because of the idling time, the operational efficiency of the entire line has to decrease.

Therefore, on the contrary, if it is possible to cut down on the length of idling time, it can improve the production rate of a line and eliminate excess facilities and equipment from the line. This is exactly what this discussion has been looking for.

In order to reach this goal, it seems to be necessary to start by analyzing the nature of idling time. The length of idling time depends upon the features of line formation which can be parametered by the number of stages, number of machines, output rate of machines, package aggregation factor, catalog efficiency of machine in terms of the mean length of time between breakdowns and mean length of time of a breakdown, and the capacity of reservoirs in terms of the lower limit and upper limit of the product accumulation in them.

Under these considerations, a simulation model has been formulated. It starts with picking up one stage from a line. Many parameters are assigned to this stage. Among them, the concept of a reservoir seems to be important. It is intended to store temporarily the in-process units produced by the immediately previous stage in case this stage is broken down.

The reservoirs are classified into three categories in reality, of which the first one is a power-and-free type of transfer mechanism such as a basic roller conveyor which can be assumed to have a limited volume of temporal storage capacity, the second is any sort of indexing transfer mechanism which is thought to have zero capacity, and the last is a unique facility such as a reservoir which is expected to claim a remarkably large volume of capacity.

The simulation model developed here can deal with the first and the third type of reservoir concept outlined above. But for the second, the reservoir of zero capacity, it requires some modifications in terms of error analysis.

The first thing to be done to the stage in the simulation model is to find out which machines in the stage are running, breaking down or idling at any given moment. The simulation model precedes this scanning process in such a way that the first two states of a machine, running and breaking down, are intepreted by the simulation model as that they are both the phenomena of random occurrence, assumed to be exponentially random; and the third, idling, is detected in such a way that if the reservoir of this stage is empty, or if the reservoir of the next stage is full, in either case or both, all the machines of this stage should be idled.

Thus the number of ready-to-go machines in the stage can be counted.

This scanning process will go through all the stages from the first to the last at the same moment of the simulation time, then advance to the next moment to repeat the same routine all over again, until finally the simulation time gets to any desired length. And this is the end of one simulation run, but not the end of the whole process.

The computation results from a computation run have to involve variances because the random number was introduced in the computation process. Therefore, the simulation run will be iterated enough times to come up with statistics--means and standard deviations.

A computer program for the simulation model has been developed and proven to work as intended. Hence it can be said here that if a packaging line can be described by the diagram (Figure 4.1) or any combined form of it, the simulation model will find an application to the problem to help design a more productive packaging line.

LIST OF REFERENCES

LIST OF REFERENCES

1. Manetsch, Thomas J., and Gerald L. Park. System Analysis and Simulation with Applications to Economic and Social Systems, Part II. Department of Electrical Engineering and System Science, Michigan State University, 1974.

2. Church, Janis. Simulation as a Tool in Assembly Line Design. Dearborn, Michigan: Society of Manufacturing Engineers, 1971.

3. Brayley, M. C. W., et al. "Clearing the Smoke from Tobacco Packaging." Packaging Technology, Vol. 17, No. 117, March 1972.

4. Francis, Richard L., and John A. White. Facility Layout and Location, an Analytical Approach. Englewood Cliffs, New Jersey: Prentice-Hall, Inc., 1974.

5. Blum, Julius R., and Judah I. Rosenblat. Probability and Statistics. Philadelphia, London, Toronto: W. B. Sanders Company, 1972.

6. Davis, G. Glenn. Packaging Converting Machinery Components, 3rd Edition. Washington, D.C.: Packaging Machinery Manufacturers Institute, 2000 K Street, N.W., 1972.