A GPSS V SIMULATION OF A COMPUTER CONTROLLED WAREHOUSE PICKING SYSTEM

David E. Douglas
Michael T. Mazzaro
Leonard J. Myers

ABSTRACT

This paper presents the model development and results of a simulation of a computer controlled warehouse picking system. The model was programmed in the GPSS V simulation language and run on an IBM 370/155 computer. This model was used to study the problem of ordering products along a conveyor belt and determining the number of men to assign to refill cartridge magazines which hold the product.

STATEMENT OF THE PROBLEM

This presentation describes the development of a simulation model of a computer controlled picking system that will be installed in a manufacturing company. The company produces a large product mix (238 different products) of small items, most of which are packaged in small boxes containing four or six individual items per box. The model is being developed with the cooperation of the company and they are providing actual statistical distributions of the number of packages of each item in an order, the number of orders per day (approximately 1100 per day) plus the package volume for each item. The package volume is necessary for computing the number of packages that will fit into a standard size shipping box for each order.

A sketch of the system to be modeled is shown in Figure 1. The heart of the system is the computer controlled conveyor belt, the cartridge magazines on each side of the conveyor, and some light beams for control. The cartridge magazines hold packaged items to be dumped onto the conveyor belt. Before the start of the picking shift, the information for each order to be processed for the day is entered into the computer. From this order information, each order is separated into the appropriate number of segments based upon the package volume that will loose fill a standard size shipping box. Then, adding the number of segments for each order, the total number of segments for the day is obtained and used to calculate the conveyor speed.

At this point the system is ready to start picking. As the first order starts down the conveyor belt, items that were computed to be in its first segment are dumped from the cartridge magazine onto the moving conveyor belt. These items form a segment. A light sensor determines if a box was actually dumped onto the conveyor belt. If one should have been dumped and is not, then this segment is flagged and will have to have manual intervention after the segment has been dumped into the shipping box at the end of the conveyor. A space is left between segments on the conveyor as the system continues to process each order.

The cartridge magazines can be of two sizes and are correspondingly at different center to center widths along each side of the conveyor. The packaged items are manually placed in the cartridge magazines.

Even an automated system such as the one described above requires some difficult decisions for management. Several of these problems are such that simulation can provide solutions that would otherwise be very difficult. Two major problems will be studied in this simulation model. Since the feeder magazines are manually refilled, the first problem consists of assigning workers to refill magazines. The second problem is determining how to sequence the products along each side of the conveyor belt. These two problems are not independent of each other, which makes for a more challenging study.

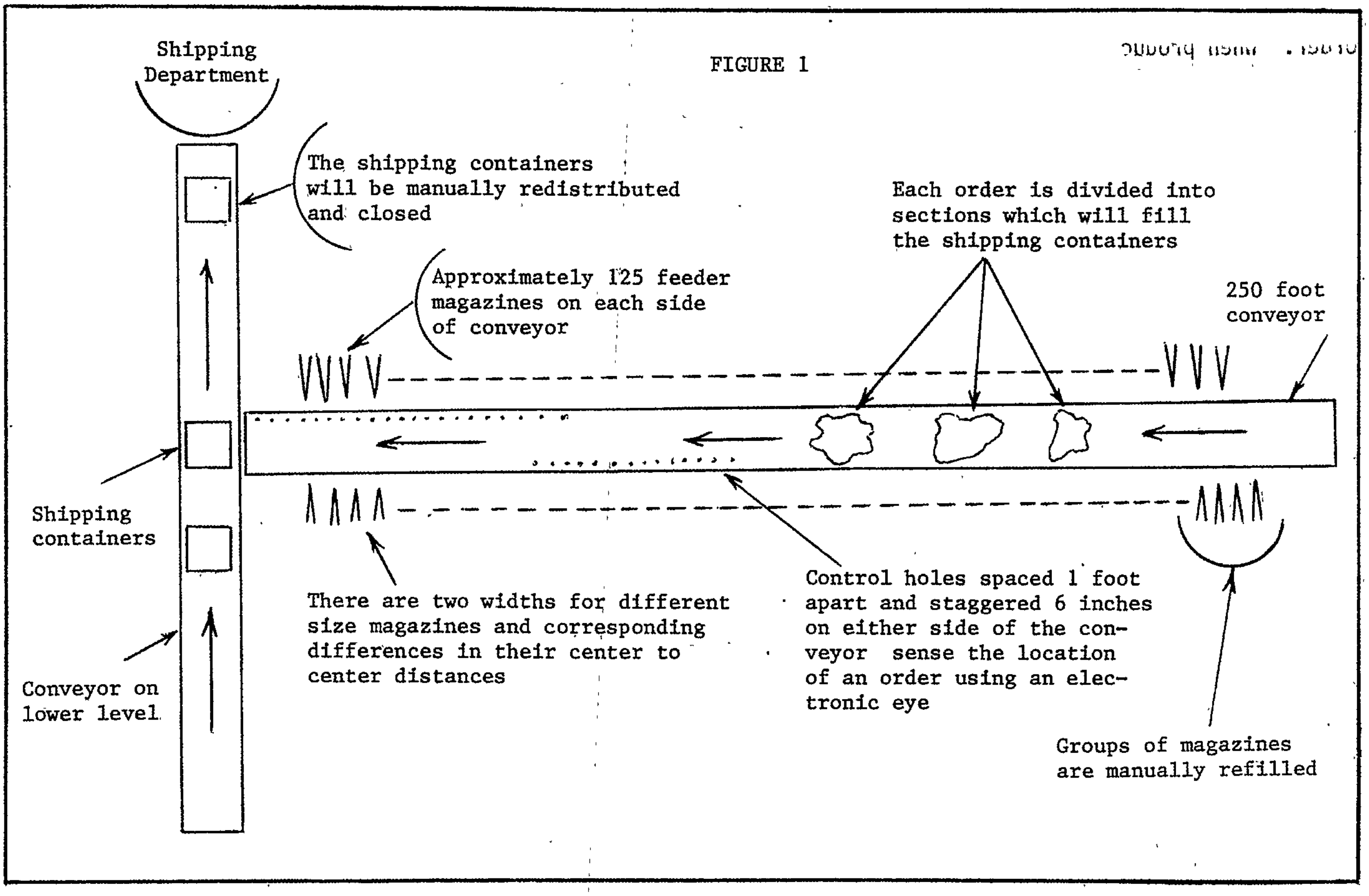
The simulation model to provide solutions for these two problems uses the GPSS V simulation language. It was developed on the University of Arkansas's IBM 370/155 computer.

APPROACH TAKEN IN SETTING UP THE MODEL

In designing the model several questions had to be answered. Among them were how to determine the contents of an order, how to advance the order down the conveyor, restocking the feeder magazines, and providing information on stock outs.

First it was necessary to decide how to determine the contents of an order. As with any model, the accuracy of statistics is essential. The statistical data available from the company on product distribution could not be made to fit a statistical distribution. Hence it was necessary to use 238 functions, one for each product, to determine the number of each product that are in an order. The functions were calculated by taking the number of times a specific quantity of the product appeared in an order during the sample period and dividing by the total number of orders. By starting with the number of times a product was not in an order (probability of zero product) the cumulative

FIGURE 1



frequency for each quantity of a product was calculated. Use of cumulative frequency facilitated the use of discrete functions and a random number generator to determine order content. The order content was kept constant for each different run to facilitate comparison of results.

At this point a question arose as to how an order should be generated and processed. Discussion with the company indicated the shipping container would hold a loose volume of 1410 cubic inches. The question which occurred was either to generate an entire order, assign the quantities of each product, and then split the order into sections, or to generate a section, assign products to it until the section is full, and then continue the assignment of the next section at the point where the last section became full. It should be noted that use of term section refers to the amount of products in an order that will loose fill into the storage container. The question is really one of style since the method used by the computer controlling the actual system was not available. After experimentation with both methods, it was decided to use the system generating by sections.

Specifications indicated the size of a section on the conveyor would be five feet and the speed of the conveyor could range from one to six inches per second. For inclusion in this paper, it was decided to use an average days demand which was a speed of three inches per second, which means that a new section will be generated every 20 seconds (determined by dividing section length by speed). Because of the GPSS simulation clock using whole time units, speed of the conveyor in seconds must be a number which divides evenly into six. However, the model is flexible since, by simply changing time units, other speeds are accomodated.

The determination of order contents is accomplished by generating a section and beginning to assign products. The assignment is accomplished by evaluation of the function number corresponding to the product in question and assigning the result of this evaluation to the half word parameter number equal to the product number (i.e., product 1 quantity is assigned to PH1). Once the quantity of a product in an order is determined the volume of that quantity is calculated by multiplying the value obtained from the volume function by this quantity. The volume function is a discrete function whose values represent the volume of a unit product and the argument of which is based on product number. When the product volume is determined, it is compared with the remaining space on a section (1410 cubic inches minus volume of product already assigned). If all the product will fit on the section the program assigns the next product and this continues until all 238 products are assigned. If a point is reached where all of a product will not fit on a section, the section is assigned as much as will fit and the remaining product is assigned to the next section and the process mentioned above continues through 238 products. A section will never contain products assigned to more than one order. When product 238 has been assigned, the next section begins with the assignment of product 1.

With the assignment of product quantity to a section complete, the model then moves the section down the conveyor and fills the order. Two methods may be used to accomplish this. The first method is the conventional GPSS method where products in a feeder magazine (represented by a storage) are thought of as servers (each server representing one unit of product). As a section reaches a storage and determines it requires a quantity of the product contained therein, it enters the storage with this quantity or the remaining contents of the storage if not enough product is available thus capturing the servers. It then leaves the storage with zero; this simulates removing products from the feeder magazines. This method gives useable data as to how often the storage is not completely full. The standard numerical attribute (SNA) used with this method is R (remaining content). This method also provides an indication of which storages actually ran out of product and how close a storage came to running out through the maximum content statistic. The second or box method considers the storage as containing boxes; as a section reaches a storage containing needed product it enters the storage with zero and leaves with the quantity of product as discussed for the server method. This method uses the S or current content SNA. The advantage of this method is that it shows the average contents of the storage. Disadvantages are the storage must be filled or its contents known at the time the simulation is begun; and how close a storage came to emptying is not known. Both methods were used but, for simplicity, only the server method is shown on the flow diagram (see Figure 2) and discussed through the remainder of this presentation.

The capacity of each storage was determined by dividing magazine height (48 inches) by the height of one unit of product. It should be noted that the storage number does not necessarily correspond to the number of the product contained in the storage (this concept will be discussed later in this section).

Once a section is full, or all products have been assigned, the section begins moving down the conveyor. To allow spacing of the sections each section moves 60 inches from the start of the conveyor before reaching the first storage. At the first storage a test is made to see if the product contained in that storage is in the order. If it is, the storage is entered and its remaining capacity reduced by that quantity. If not, the section checks the next storage and enters it if the product is in the order. Since the magazines are located across from each other, two storages are checked before a clock advancement may occur. To save execution time clock advances for sections moving down the conveyor occur only when the location of the next storage containing a product in the order is located (except when all products have been filled and then the section advances 20 time units and terminates). This 20 second advancement simulates moving the section off the conveyor. A more detailed discussion of the advancement of the section down the conveyor can be found under model logic.

One of the primary purposes of the model is to provide data for decisions concerning the use of men to restock the feeder magazines. Four factors combine to provide the needed data: how many men are needed to refill the feeder magazines, how many magazines should a man be responsible for, is there an arrangement of products along the conveyor which reduces stock outs or is order of products not important, and what were the stock outs for a given line configuration and number of men.

To assist in answering these questions a transaction is generated for each man working on the line. Each of these transactions has 4 halfword parameters and 2 of these parameters are assigned the number of the first and last storage for which a man is responsible. Since a man works on one side of the conveyor he will work only on odd or even number storages between his assigned limits. The section on model logic discusses the operation of the man filling the magazines.

In designing the model, there was much discussion concerning movement of the men restocking the magazines. Initially it was felt the man would select the magazine with the lowest percentage remaining capacity, restock it, and then start the selection process again. This method, however, seems impractical and inefficient as the man probably would spend more time selecting the correct magazine, than just filling a magazine as he walks along and simply walking back and forth between his assigned magazines all day. This back and forth approach is the one used in the model. Accurate time data was not available concerning the restocking procedure. So the model was developed using best estimate figures for movement of the men. Because of the size of the product boxes the estimated time to handle ten or less boxes would be approximately the same. Hence the man checks the current contents (S) of the storage to see if it is greater than zero but less than or equal to ten; if it is he advances ten time units (the time to put ten or less unit products in a magazine), enters the storage with zero units and leaves with the current contents, this simulates restocking the magazine. If the current contents is greater than ten, ten units are restocked and the process is repeated until the storage has a current contents of zero (the magazine is full). To cover a section taking product from a storage while the man is advancing ten time units, each time the man leaves a storage he checks to ensure it is full. When the man finds a storage full he advances five units (the time to move six inches between magazines and see if the next magazine needs to be restocked) and begins the restocking process again. The only difference between the server and box method as far as restocking is concerned is that in the server method SNA "S" is used while SNA "R" is used in the box method; additionally in the box method the man enters with ten or less units and leaves with zero units.

FIGURE 2

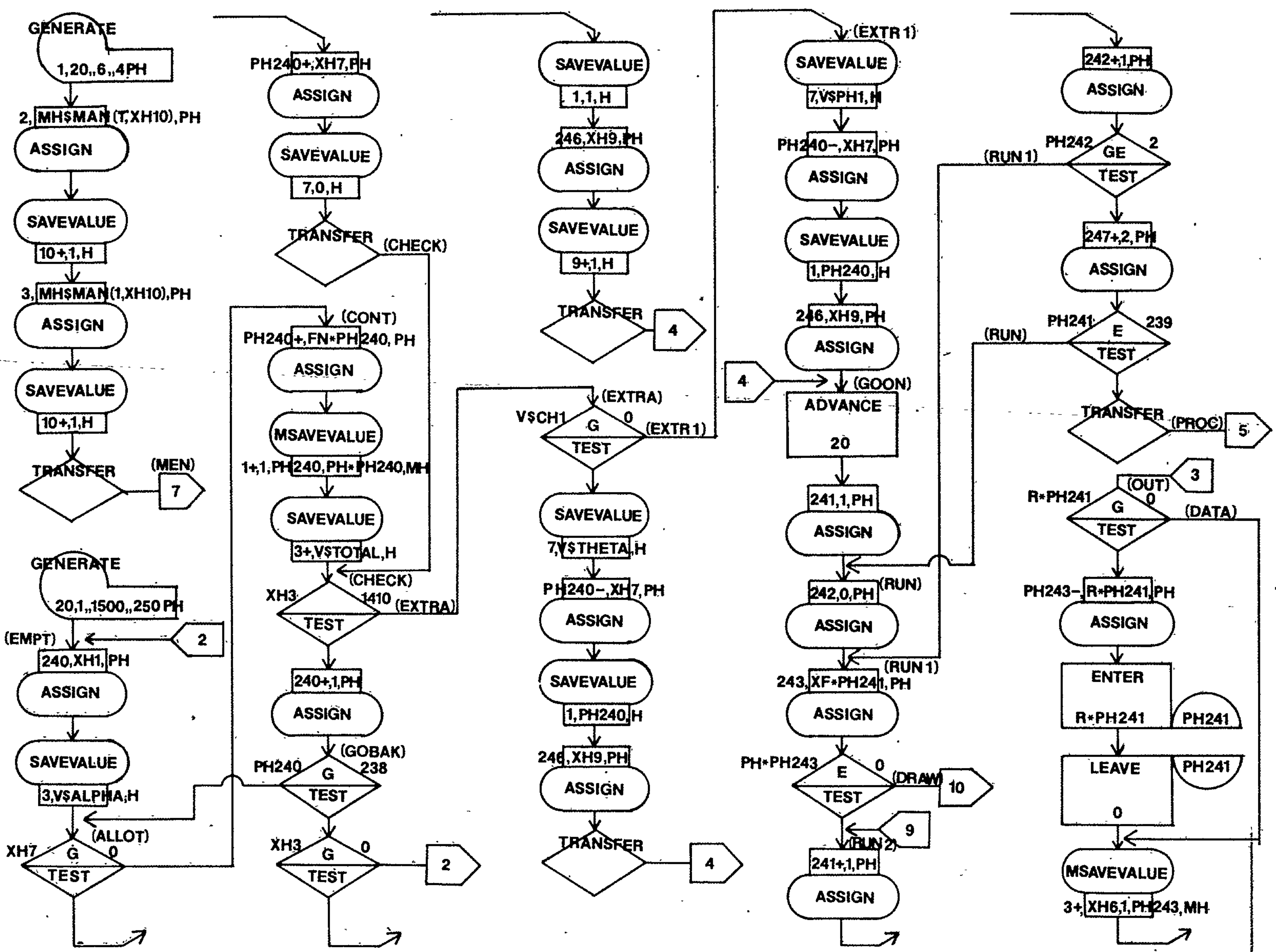
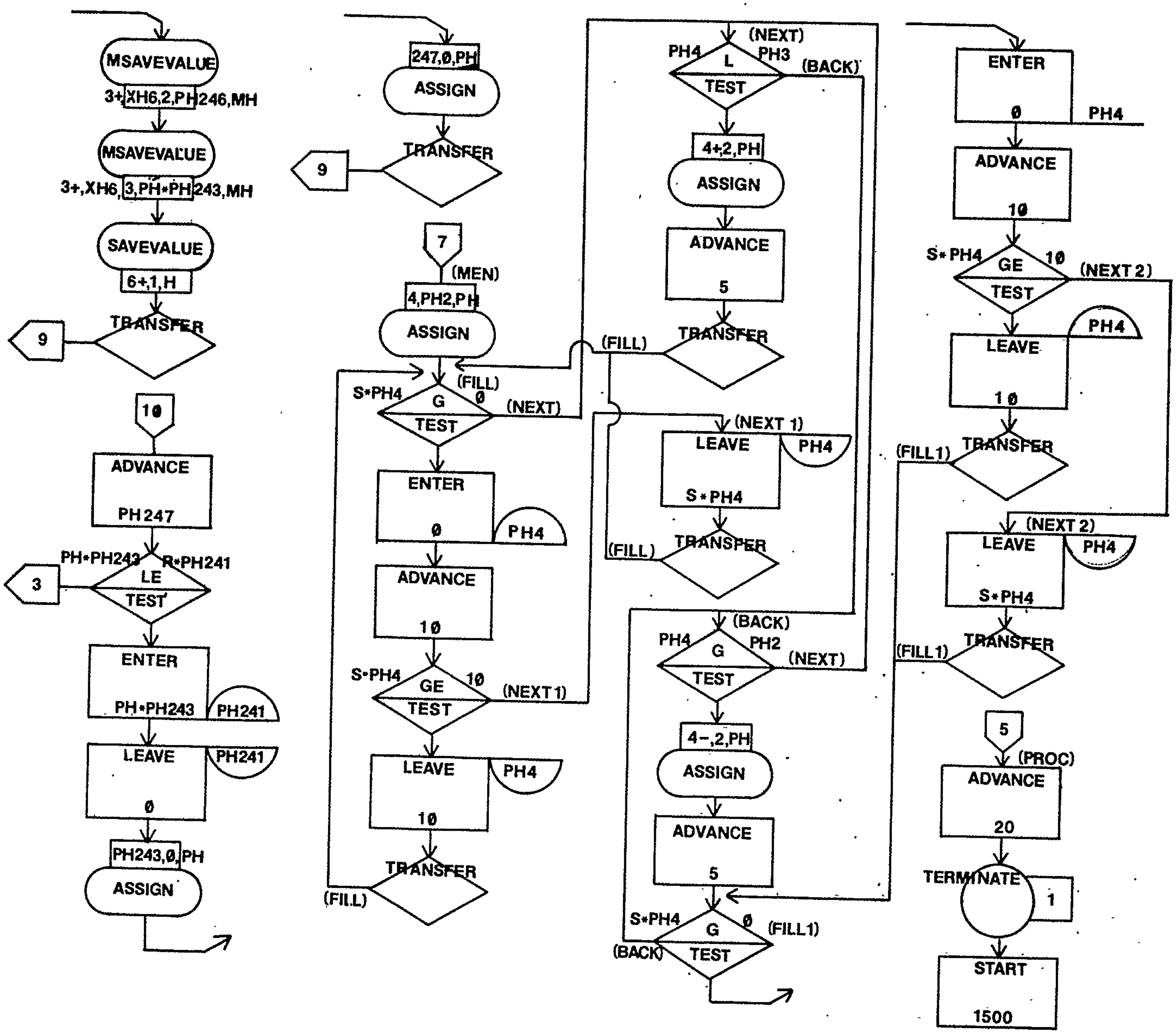

GENERATE
1,20,,6,,4PH
2, MH$MAN(T,XH10),PH
ASSIGN
SAVEVALUE
10+,1,H
3, MH$MAN(1,XH10),PH
ASSIGN
SAVEVALUE
10+,1,H
TRANSFER
(MEN)
7
GENERATE
20,1,,1500,,250 PH
(EMPT)
2
240,XH1,PH
ASSIGN
SAVEVALUE
3,V$ALPHA,H
(ALLOT)
XH7
G
0
TEST
PH240+,XH7,PH
ASSIGN
SAVEVALUE
7,0,H
TRANSFER
(CHECK)
(CONT)
PH240+,FN*PH240, PH
ASSIGN
MSAVEVALUE
1+,1,PH240,PH*PH240,MH
SAVEVALUE
3+,V$TOTAL,H
(CHECK)
XH3
1410
(EXTRA)
TEST
240+,1,PH
ASSIGN
(GOBAK)
PH240
G
238
TEST
XH3
G
0
TEST
2
SAVEVALUE
1,1,H
246,XH9,PH
ASSIGN
SAVEVALUE
9+,1,H
TRANSFER
4
(EXTRA)
V$CH1
G
0
(EXTR1)
TEST
SAVEVALUE
7,V$THETA,H
PH240-,XH7,PH
ASSIGN
SAVEVALUE
1,PH240,H
246,XH9,PH
ASSIGN
TRANSFER
4
(EXTR1)
SAVEVALUE
7,V$PH1,H
PH240-,XH7,PH
ASSIGN
SAVEVALUE
1,PH240,H
246,XH9,PH
ASSIGN
4
(GOON)
ADVANCE
20
241,1,PH
ASSIGN
242,0,PH
(RUN)
ASSIGN
(RUN1)
243,XF*PH241,PH
ASSIGN
PH*PH243
E
0
(DRAW)
10
TEST
9
(RUN2)
241+,1,PH
ASSIGN
242+,1,PH
ASSIGN
PH242
GE
2
(RUN1)
TEST
247+,2,PH
ASSIGN
PH241
E
239
(RUN)
TEST
TRANSFER
(PROC)
5
3
(OUT)
R*PH241
G
0
(DATA)
TEST
PH243-,R*PH241,PH
ASSIGN
ENTER
R*PH241
PH241
LEAVE
0
PH241
MSAVEVALUE
3+,XH6,1,PH243,MH

FIGURE 2 (Continued)



Due to the variance in product frequency and storage capacity, we felt it would be beneficial to design the model to permit arrangement of the products by product number along the conveyor. This allows high volume products (in terms of quantity) to be placed towards the middle of a man's area of responsibility thus reducing the time span between restockings. To add this flexibility to the model, fullword savevalues are given the number of the product contained in the storage (if storage one contains product 140 then XF1 has a value of 140) and storage capacity is changed accordingly. In simulations where product order is sequential, fullword savevalues are not used and the section advancing down the conveyor tests to see if it contains the product in a given storage using direct address rather than indirect address using "XF" as is required for arranged product order.

When a section requires product from a storage and the storage does not have enough servers to provide all the needed product, the quantity of product not available is placed in matrix 3. In this manner stock out information is obtained.

In summary, the model uses transactions to simulate both the men restocking the magazines and the sections, which combine to form orders. Sections may be moved down the conveyor using either the server or box method. Storages are used to represent the feeder magazines which supply the product to a section.

MODEL DEFINITIONS

CLOCK TIME - seconds

TRANSACTIONS - the individual sections of an order; the men refilling the magazines.

VARIABLES:

ALPHA - determines total volume of excess boxes.

CHI - calculates if excess volume is equally divisible by box volume.

DELTA - determines total excess volume.

PHI - number of boxes which make up excess volume.

THETA - number of boxes in excess if volume is not equally divisible.

TOTAL - total volume of particular product in order section.

BLOCK SYMBOLS:

ALLOT - test to see if there was excess product in previous section.

BACK - segment for determining which storage man is filling on his backward pass.

CHECK - determine if section is filled yet.

CONT - assign next product.

DATA - segment which stores information on magazine outages.

EMPT - segment where all sections are sent if no products are assigned to section.

EXTRA - segment subtracts off excess boxes from section.

EXTR 1 - segment subtracts off excess boxes from section.

FILL - man determines if magazine requires filling.

FILL 1 - man determines if magazine requires filling.

FIN - send sections to termination.

GOBAK - go back and try to allocate next product.

MEN - initially assigns each man his first bin to start filling.

NEXT - man checks to see if finished with forward pass of magazines.

NEXT 1 - man fills the bin he is at.

NEXT 2 - man fills the bin he is at.

OUT - this segment supplies each product with required boxes from magazine until the magazine is empty.

PROC - order has been processed through all magazines and leaves conveyor.

RUN - assign order to a right side magazine.

RUN 1 - determine which product is in magazine that order is at.

RUN 2 - go to next magazine.

FUNCTIONS:

1 - 238 - represents cumulative frequency of quantity of the product corresponding to the function number.

VOL - the volume of a unit of product number as a function argument.

SECTION PARAMETERS:

PH1-PH238 - quantity of product corresponding to parameter number assigned to a section.

PH240 - contains the product number currently being assigned.

PH241 - determines which storage is being tested.

PH242 - determines when order section moves to next group of two storages.

PH243 - the product number contained in storage PH241.

PH246 - the order number a section is assigned to.

MAN PARAMETERS:

PH2 - first storage man is responsible for.

PH3 - last storage man is responsible for.

PH4 - storage man is currently filling.

HALFWORD MATRICES:

MAN - list starting and ending storages for each man.

1 - total quantity of each product used during simulation.

3 - product, order number, and quantity of stock out.

FULLWORD SAVEVALUES:

XF1-XF238 - represents the product number contained in the storage corresponding to the savevalue number (if storage 1 contains product 140 then XF1 has a value of 140).

HALFWORD SAVEVALUES:

XH1 - determines the first product to be assigned to a section.

XH3 - totalizes volume of products in a section.

XH6 - counts number of stock outs.

XH7 - stores number of excess boxes in a section.

XH9 - number of orders processed.

XH10 - counter used in assigning magazines to men.

DISCUSSION OF MODEL LOGIC

The program listing, omitting the FUNCTION cards and INITIAL cards, is shown in Figure 3. To make discussion of the model logic easier, the program will be broken down into nine sigments. Generate Men, Section allotment, 'Extra', 'Extra 1', 'Run', 'Draw', 'Out', Forward Pass of men, and Backward Pass of men are the nine separate segments of the program. Each of these segments will be discussed as to its use and its interaction with the other segments.

The first segment generates the transactions which simulate the men who refill the magazines. Each transaction is assigned four halfword parameters. Parameters two and three determines the lowest and highest numbered magazines the man fills. Savevalue ten is used to increment the location number in matrix 'MAN' as each transaction is processed through.

The segment used to generate orders and assign products to each order is the largest and most complex segment. As each transaction is generated it is alloted 250 halfword parameters. The first 238 parameters represent the 238 different possible products in each section. Three savevalues are used in this program segment. Savevalue one keeps track of which product is being assigned, savevalue three stores the total volume of products assigned to a section, and savevalue seven stores the number of boxes which exceed the capacity of a section (1410 cubic inches).

After a transaction is generated it is allotted any excess boxes from the previous section, if there was no excess then the allocation process is started where the last section finished. This may be the first product for the first section of an order or it may be somewhere between product 1 and 238 if it is a subsequent section of an order. If product 238 is assigned before the maximum volume is reached, the section is released to start down the conveyor and the next transaction will start a new order with product one.

The 'Extra' segment is used when the order section has been assigned too much product. This segment tests to see if the excess volume is equally divisible then the excess boxes plus one are taken from the section and saved for the next transaction. If the volume is equally divisible then the 'Extr 1' segment is used and the one extra box is not included.

The 'Run' segment deals with the orders processing down the conveyor. Parameters 241, 242, and 243 are used to identify which magazine, which side of the conveyor, and what product respectively. Each magazine is tested to see if the magazine contains product required by the transaction. Each time a group of two magazines do not contain required product for an order the time it takes to reach the next two magazines is added to parameter 247. When a magazine does contain a section product, the transaction is sent to the 'Draw' segment.

Once a transaction does find a magazine with required product, the 'Draw' segment will advance the clock time according to parameter 247. The transaction will then caputre the number of servers corresponding to the boxes required from the appropriate magazine. The parameter 247 will then be reset at zero. If the storage being entered does not contain enough servers to satisfy the transaction then the transaction is sent to the 'Out' segment.

The 'Out' segment is used primarily for output. This segment caputres what servers are still left in the storage and then stores in a matrix the order, product number, and the number of boxes short.

The forward pass of the men filling magazines is done by starting the man at the magazine described in parameter two. The man then tests each magazine to see if all the servers are available. If all servers are available he then moves five time units to the next magazine he is responsible for. When he finds a magazine where all servers are not available he will make ten or less servers available after a ten second delay. He then tests to see if the magazine has all servers available and proceeds in the manner described above.

The backward pass of the man is started when the test block (PH4 less than PH3) cannot be satisfied. This means the man has reached the last magazine he is responsible for and will start moving back towards the first magazine, testing each one as he goes. The only difference in this segment and the forward pass segment is that the parameter which keeps track of the magazine being tested (PH4) is being reduced by two instead of increased by two each time the man moves.

DISCUSSION OF OUTPUT

The one run made for 1100 sections used fourteen men to restock the unarranged magazines and no stock outs occurred. The real use of this run however was to help determine high volume products to assist in the possible rearrangement of products along the conveyor. To determine the high volume products we took the quantity of product for the run (halfword matrix 1) divided by number of orders (765), and divided this quotient by the capacity of the storage containing the product. Any results of this calculation exceeding .01 were considered high volume products; further results above .02 were considered critical volume functions. In arrangement of the products, critical volume products were distributed as evenly as possible between men and placed at the center of a man's area of responsibility. The other high volume products were also distributed and flank the critical products along the area of responsibility. Additionally, products where the probability was zero or close to zero for being in the order were distributed close to either end of the man's area of responsibility.

Four runs of 800 sections each were made using eight men and six men respectively for both arranged and unarranged products. For the eight men unarranged, stockouts occurred 2.2% of the time (12 orders of 554); for the eight men arranged, .9% (5 of 554); for six men unarranged 7.6% (42 of 554) and for six men arranged 2.5% (14 of 554). The above figures can be compared because all conditions were the same except for number of men and product arrangement. (See Tables 1, 2, and 3.)

With any model, verification is important. Three steps were taken to verify this model. First the expected value of several products was calculated for a sample run of 765 orders; these expected values compared within three or four percent of the product used during the actual simulation. Next, the number of times a product was not in an order was checked and again the values in the simulation compared fovarably with the values obtained from the sample data provided by the company. Finally the anticipated time to process 800 sections was calculated and the value of the simulation clock compared well with the anticipated clock value. The actual clock time can not be calculated because it depends on the location of the products along the line and the contents of the orders.

TABLE 1

MATRIX 3 (8 Men Unarranged)

OUTAGE	PRODUCT	ORDER	BOXES SHORT
1	193	87	2
2	114	174	2
3	114	176	1
4	19	193	2
5	24	249	1
6	184	257	3
7	185	292	8
8	184	301	3
9	22	346	6
10	112	452	1
11	112	453	5
12	112	456	2

TABLE 2

MATRIX 3 (8 Men Arranged)

OUTAGE	PRODUCT	ORDER	BOXES SHORT
1	19	188	2
2	19	189	3
3	185	217	8
4	112	452	1
5	112	453	5

TABLE 3

MATRIX 3 (6 Men Arranged)

OUTAGE	PRODUCT	ORDER	BOXES SHORT
1	22	26	6
2	123	57	1
3	193	93	1
4	19	188	5
5	118	245	1
6	221	265	3
7	221	269	1
8	134	320	1
9	22	348	3
10	22	352	4
11	112	452	2
12	112	453	5
13	119	471	1
14	31	507	2

FIGURE 3

```
          *  BINS FOR WHICH MAN IS RESPONSIBLE  IF COL IS ODD 1ST BIN,IF EVEN,LAST BIN
                   INITIAL     MH$MAN(1,1),1/MH$MAN(1,2),79/MH$MAN(1,3),81
                   INITIAL     MH$MAN(1,4),159/MH$MAN(1,5),161/MH$MAN(1,6),239
                   INITIAL     MH$MAN(1,10),160/MH$MAN(1,11),162/MH$MAN(1,12),240
                   INITIAL     MH$MAN(1,7),2/MH$MAN(1,8),80/MH$MAN(1,9),82
                   INITIAL     XH1,1/XH9,1/XH10,1/XH6,1

          * VARIABLE DEFINITIONS
          *
            ALPHA FVARIABLE   XH7*FN$VOL                   VOLUME OF EXCESS BOXES
            DELTA FVARIABLE   XH3-1410                     TOTAL EXCESS VOLUME
            CHI   VARIABLE    V$DELTA@FN$VOL               CHECK FOR COMPLETE BOXES
            PHI   FVARIABLE   V$DELTA/FN$VOL               BOXES IN EXCESS VOLUME
            THETA FVARIABLE   V$PHI+1.0                    EXCESS BOXES IF NOT WHOLE BOX
            TOTAL FVARIABLE   PH*PH240*FN$VOL+.5           VOLUME OF PRODUCT IN ORDER
                  RMULT       3333
          *
          *  SEGMENT 1  ASSIGN MAN HIS MAGAZINES
          *
 1                GENERATE    1,,,6,,4PH                   GENERATE A MAN                 100
 2                ASSIGN      2,MH$MAN(1,XH10),PH          MAN,S 1ST MAGAZINE             110
 3                SAVEVALUE   10+,1,H                      INCREMENT COLUMN COUNTER       120
 4                ASSIGN      3,MH$MAN(1,XH10),PH          MAN,S LAST MAGAZINE            130
 5                SAVEVALUE   10+,1,H                      INCREMENT COLUMN COUNTER       140
 6                TRANSFER    ,MEN                                                        150
          *
          *  SEGMENT 2  DETERMINING CONTENTS OF AN ORDER
          *
 7                GENERATE    20,,1,1600,,250PH            GENERATE A SECTION             160
 8          EMPT  ASSIGN      240,XH1,PH                   PRODUCT TO BE ALLOCATED        170
 9                SAVEVALUE   3,V$ALPHA,H                  EXCESS VOL PRIOR SECTION       180
10          ALLOT TEST G      XH7,0,CONT                   IS THERE EXCESS PRODUCT        190
11                ASSIGN      PH240+,XH7,PH                ASSIGN EXCESS PRODUCT          200
12                SAVEVALUE   7,0,H                        ZERO OUT EXCESS PRODUCT        210
13                TRANSFER    ,CHECK                                                      220
14          CONT  ASSIGN      PH240+,FN*PH240,PH           ASSIGN PRODUCT TO SECTION      230
15                TEST G      PH*PH240,0,CONT1             IS PROD IN ORDER
16                MSAVEVALUE  PORD+,1,PH240,1,MH           INCREMENT PORD MATRIX
17                MSAVEVALUE  1+,1,PH240,PH*PH240,MH       ADD TO PRODUCT USED MATRIX
18                SAVEVALUE   3+,V$TOTAL,H                 VOL OF PRODUCT IN SECTION      250
19          CHECK TEST LE     XH3,1410,EXTRA               IS SECTION OVERFULL            260
20          CONT1 ASSIGN      240+,1,PH                    MOVE TO NEXT PRODUCT           270
21          GOBAK TEST G      PH240,238,ALLOT              ARE ALL PRODUCTS ASSIGNED      280
22                TEST G      XH3,0,EMPT                   AT LEAST 1 PROD IN SECTION     290
23                SAVEVALUE   1,1,H                        BACK TO 1ST PRODUCT            300
24                ASSIGN      246,XH9,PH                   ORDER NUMBER TO SECTION        310
25                SAVEVALUE   9+,1,H                       INCREMENT ORDER NUMBER         320
26                TRANSFER    ,GOON                        MOVE TO CONVEYOR               330
          * DETERMINE IF EXCESS BOXES ARE WHOLE NUMBER
27          EXTRA TEST G      V$CHI,0,EXTR1                EXCESS BOXES WHOLE NUMBER      340
28                SAVEVALUE   7,V$THETA,H                  NUMBER OF EXCESS BOXES         350
29                ASSIGN      PH240-,XH7,PH                SUBTRACT EXCESS BOXES          360
30                SAVEVALUE   1,PH240,H                    BEGIN NEXT SECTION AT PH240    370
31                ASSIGN      246,XH9,PH                   ASSIGN ORDER NUMBER            380
32                TRANSFER    ,GOON                        MOVE TO CONVEYOR               390
          *  EXCESS BOXES ARE NOT A WHOLE NUMBER
33          EXTR1 SAVEVALUE   7,V$PHI,H                    NUMBER OF EXCESS BOXES         400
34                ASSIGN      PH240-,XH7,PH                SUBTRACT EXCESS BOXES          410
35                SAVEVALUE   1,PH240.H                    BEGIN NEXT SECTION AT PH240    420
36                ASSIGN      246,XH9,PH                   ASSIGN ORDER NUMBER            430
          *
          *  SEGMENT 2  MOVING SECTION DOWN CONVEYOR AND FILLING ORDER
          *
37          GOON  ADVANCE     20                           MOVE TO 1ST MAGAZINE           440
38                MSAVEVALUE  SECTI+,1,PH246,1,MH          INCREMENT SCETION MATRIX
39                ASSIGN      241,1,PH                     1ST MAGAZINE TESTED            450
40          RUN   ASSIGN      242,0,PH                     LEFT SIDE MAGAZINE             460
41          RUN1  ASSIGN      243,XF*PH241,PH              PROD NUMBER IN MAGAZINE        470
42                TEST E      PH*PH243,0,DRAW              IS PRODUCT IN ORDER            480
43          RUN2  ASSIGN      241+,1,PH                    NEXT MAGAZINE                  490
```

FIGURE 3 (Continued)

```
 44          ASSIGN      242+,1,PH                CHANGE SIDES OF CONVEYOR    500
 45          TEST GE     PH242,2,RUN1             LEFT SIDE MAGAZINE          510
 46          ASSIGN      247+,2,PH                INCREMENT TIME COUNTER      520
 47          TEST GE     PH241,240,RUN            LAST MAGAZINE               530
 48          TRANSFER    ,PROC                    MOVE OFF CONVEYOR           540
      *
      *  SEGMENT 3  SUPPLYING THE PRODUCT
      *
 49    DRAW  ADVANCE     PH247                    MOVE TO NEEDED MAGAZINE     550
 50          TEST LE     PH*PH243,R*PH241,OUT     IS THERE ENOUGH PROD        560
 51          ENTER       PH241,PH*PH243           TAKE PRODUCT FROM MAGAZINE  570
 52          LEAVE       PH241,0                  RELEASE MAGAZINE            580
 53          ASSIGN      PH243,0,PH               PRODUCT COMPLETELY FILLED   590
 54          ASSIGN      247,0,PH                 ZERO TIMER COUNTER          600
 55          TRANSFER    ,RUN2                    CONTINUE WITH NEXT PRODUCT  610
      *
      *  SEGMENT 4  MAGAZINE DOES NOT CONTAIN ENOUGH PRODUCT
      *
 56    OUT   TEST G      R*PH241,0,DATA           IS MAGAZINE NOT EMPTY       620
 57          ASSIGN      PH243-,R*PH241,PH        ASSIGN AVAILABLE PRODUCT    630
 58          ENTER       PH241,R*PH241            REMOVE PROD FROM MAGAZINE   640
 59          LEAVE       PH241,0                  RELEASE MAGAZINE            650
      *
      *  MATRIX OF STOCKOUT INFORMATION
 60    DATA  MSAVEVALUE  3+,XH6,1,PH243,MH        PROD NUMBER NOT RECIEVED    660
 61          MSAVEVALUE  3+,XH6,2,PH246,MH        ORDER NUMBER NOT FILLED     670
 62          MSAVEVALUE  3+,XH6,3,PH*PH243,MH     # BOXES NOT RECIEVED        680
 63          SAVEVALUE   6+,1,H                   INCREMENT COUNTER           690
 64          TRANSFER    ,RUN2                    CONTINUE WITH NEXT PRODUCT  700
      *
      *  SEGMENT 5  MAN RESTOCKING THE MAGAZINES
      *
 65    MEN   ASSIGN      4,PH2,PH                 MAGAZINE MAN WORKING WITH   710
 66    FILL  TEST G      S*PH4,0,NEXT             IS MAGAZINE FULL            720
 67          TEST GE     S*PH4,10,NEXT1           < 10 BOXES MISSING          730
 68          ADVANCE     10                       TIME TO LOAD 10 BOXES       740
 69          ENTER       PH4,0                    SEIZE THE MAGAZINE          750
 70          LEAVE       PH4,10                   LOAD 10 BOXES               760
 71          TRANSFER    ,FILL                    RECHECK MAGAZINE            770
 72    NEXT1 ADVANCE     10                       TIME TO LOAD < 10 BOXES     780
 73          ENTER       PH4,0                    SEIZE THE MAGAZINE          790
 74          LEAVE       PH4,S*PH4                LOAD THE MAGAZINE           800
 75          TRANSFER    ,FILL                    RECHECK MAGAZINE            810
 76    NEXT  TEST L      PH4,PH3,BACK             MAN AT HIS LAST MAGAZINE    820
 77          ASSIGN      4+,2,PH                  NEXT MAGAZINE               830
 78          ADVANCE     5                        TIME TO NEXT MAGAZINE       840
 79          TRANSFER    ,FILL                    CHECK MAGAZINE              850
 80    BACK  TEST G      PH4,PH2,NEXT             MAN AT HIS FIRST MAGAZINE   860
 81          ASSIGN      4-,2,PH                  NEXT MAGAZINE               870
 82          ADVANCE     5                        TIME TO NEXT MAGAZINE       880
 83    FILL1 TEST G      S*PH4,0,BACK             IS MAGAZINE FULL            890
 84          TEST GE     S*PH4,10,NEXT2           < 10 BOXES MISSING          900
 85          ADVANCE     10                       TIME TO LOAD 10 BOXES       910
 86          ENTER       PH4,0                    SEIZE THE MAGAZINE          920
 87          LEAVE       PH4,10                   LOAD 10 BOXES               930
 88          TRANSFER    ,FILL1                   RECHECK MAGAZINE            940
 89    NEXT2 ADVANCE     10                       TIME TO LOAD < 10 BOXES     950
 90          ENTER       PH4,0                    SEIZE THE MAGAZINE          960
 91          LEAVE       PH4,S*PH4                LOAD THE MAGAZINE           970
 92          TRANSFER    ,FILL1                   RECHECK THE MAGAZINE        980
 93    PROC  ADVANCE     20                       MOVE SECTION OFF CONVEYOR   990
 94    FIN   TERMINATE   1                        SECTION LOADS SHIPPING BOX  1000
             START       1600
             END
```

BIBLIOGRAPHY

1. Bobillier, P. A., Kuhan, B. C., Probst, A. R., Simulation With GPSS and GPSS V. Englewood Cliffs, NJ: Prentice Hall, 1976.

2. General Purpose Simulation System V User's Manual, SH20-0851-2, IBM, 1977.

3. Gordon, Geoffrey, The Application of GPSS V to Discrete System Simulation. Englewood Cliffs, NJ: Prentice Hall, 1975.

4. Schriber, Thomas J., Simulation Using GPSS. New York: Wiley & Sons, 1974.