THE CRITICAL SPARES INVENTORY SYSTEM

by

Darrin Joel Lipscomb

Project/Report submitted to the Faculty of the

Virginia Polytechnic Institute and State University

in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

in

Systems Engineering

APPROVED:

W. J. Fabrycky

W.J.Fabrycky, Chairman

B.S.Blanchard

B.S.Blanchard

Michael L. Mears

M.L.Mears

May, 1992

TABLE OF CONTENTS

FIGURES

TABLES

APPENDICES

I. INTRODUCTION

This report addresses a long standing logistical support problem of determining, with some degree of precision, the number of spares necessary to keep deployed repairable equipment population systems in operation. Of particular importance is the determination of the quantity of spares to be purchased and stocked, when the spares should be ordered, and the probability that a stockout does not occur. Such a problem exists because of the costs associated with having too many or too few spares in inventory. Having too many spares in inventory is usually responsive to demand but costly in terms of inventory holding costs while having too small a supply of spares increases the risk of a stockout and costs associated with system down-time. This project/report derives its motivation from the latter of the two opposing problems; those in which the costs associated with system down-time are so excessive that they are deemed unacceptable.

It is assumed that a population of repairable equipment items exist in operation having equivalent reliability (MTBF). These items may be repairable components of more complex systems such as motors, pumps, or transmissions. When an item fails it is replaced with a spare from inventory and a repair versus discard decision is made for the failed item. If the item is to be repaired then it is either serviced, provided a server is available, or it waits in the queue until a server becomes available. As items are discarded the inventory of spares is depleted. It is necessary to procure spares at a certain rate in order to keep the inventory at a steady/safe level so that a stockout does not occur. The process mentioned above is referred to as the Critical Spares Inventory System (CSIS). (See Figure 1).

The objective of this report is to formulate a mathematical model which takes the form of a probability mass function for the number of failed items for the system operating in steady-state. The probability mass function can then be used to evaluate the steady-state levels of spares based on the probability of a stockout for each level. CSIS, together with ordering and holding costs, can be used to determine the economic order quantity (EOQ) and the level of spares at which a procurement should take place in order to maintain the steady-state level of spares. Major system parameters involved in this development are

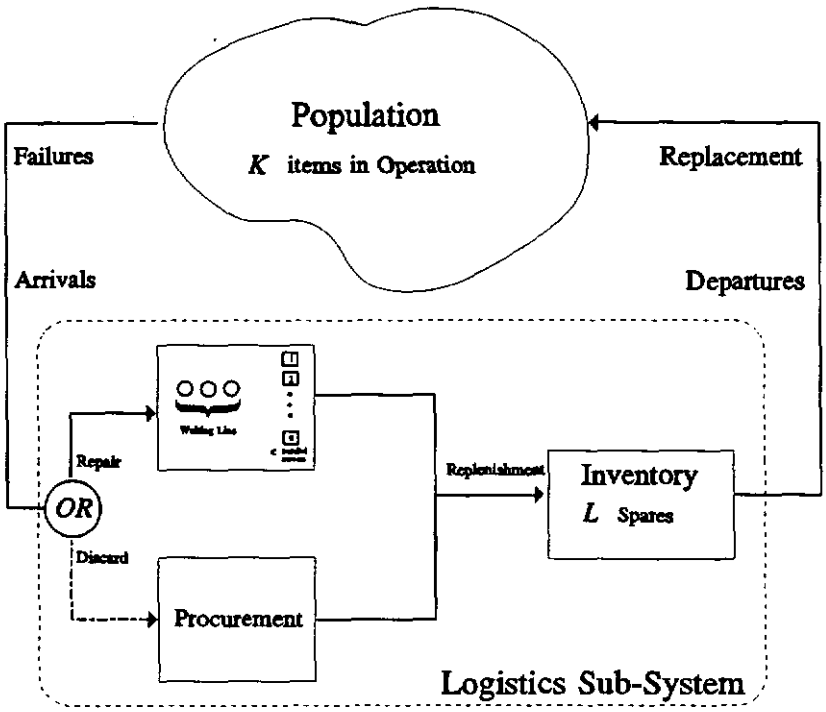


Figure 1. Critical Spares Inventory System

(1) the reliability of the part to be spared, (2) maintainability, (3) discard proportion, (4) number of servers for the repair facility, (5) procurement lead time, and (6) procurement and holding costs.

The first part of this report presents the derivation of the probability mass function through the use of the General Birth-Death Process.[1] This method was chosen, since in its most simplistic form, the system has arrivals of failed items or *deaths* and these items are replaced or *born* either through repair or procurement. The latter part of this report will present general examples utilizing the probability mass function which lend themselves to the development of a computer algorithm. Chapter V discusses the use of the CSIS application in testing alternative designs. Appendix A contains a user's guide for the CSIS software application which accompanies this document. Appendix B contains the CSIS software code listings.

II. THE MATHEMATICAL MODEL

In beginning the formal development of the model, several assumptions about the system being modeled must be stated. In general, there is a population of equipment items to accomplish a necessary objective and a system from which failed items are replaced. The assumptions are:

1. The equipment items in the operational population are finite and homogeneous, i.e., each item may belong to a larger operational unit; however, the failure rate of all of the items are equivalent as is their utilization.

2. There are c parallel repair channels each with equivalent operating characteristics.

3. The interarrival times of failed items are exponentially distributed as are the repair times.

4. The time spent replacing a failed item in the population with a spare from inventory is assumed to be zero.

5. The repair vs. discard decision for a failed item is made immediately when the item fails.

Let,

K = Number of items in operation (population).

L = Steady-state level of spares in inventory.

N = Number of items in operation and inventory = $K + L$.

c = Number of repair channels.

λ = Failure rate of item in operation = 1/MTBF.

μ = Repair rate of a repair channel = 1/MTTR.

δ = Discard proportion of failed items.

λ_n = Failure rate of population when n items have failed.

μ_n = Repair rate when n items have failed.

δ_n = Discard rate when n items have failed.

n = Number of failed items.

P_n = Steady-state probability of n failed items.

The development will follow the general birth-death process presented by White, et al. It is assumed that a sufficiently small interval of time, Δt, can be chosen, such that multiple arrivals, multiple replenishment, and both arrivals and replenishment cannot occur in the interval. So, given n failed parts,

$\lambda_n \Delta t$ = probability of one failed item during Δt

$1 - \lambda_n \Delta t$ = probability of no failed items during Δt

$\mu_n \Delta t$ = probability of one repaired item during Δt

$1 - \mu_n \Delta t$ = probability of no repaired item during Δt

$\delta_n \Delta t$ = probability of one procured item during Δt

$1 - \delta_n \Delta t$ = probability of no procured item during Δt

The only events which can take place in the interval Δt that will produce n failed items in the system at time $t + \Delta t$ are the following:

1. n failed items at time t, no arrival or replenishment during Δt ,

2. n-1 failed items at time t, one arrival and no replenishment during Δt , and

3. $n+1$ failed items at time t, no arrival and one replenishment during Δt ,

where *Replenishment* is defined as either a repaired or procured item. Thus, the probability of n failed items in the system at time $t+\Delta t$ is defined as the sum of the three mutually exclusive events above,

$$\begin{aligned} P_n(t+\Delta t) = \; & P_n(t)\,(1-\lambda_n\Delta t)\,(1-\mu_n\Delta t)\,(1-\delta_n\Delta t) \\ & + P_{n-1}(t)\,\lambda_{n-1}\Delta t\,(1-\mu_{n-1}\Delta t)\,(1-\delta_{n-1}\Delta t) \\ & + P_{n+1}(t)\,(1-\lambda_{n+1}\Delta t)\,(\mu_{n+1}\Delta t+\delta_{n+1}\Delta t) \end{aligned} \qquad (2.1)$$

Expanding the right side of the Equation (2.1), taking $P_n(t)$ to the left-hand side, dividing by Δt , and letting $\Delta t \to 0$ gives the following differential-difference equation:

$$\begin{aligned} \frac{dP_n(t)}{dt} &= -P_n(t)\,\lambda_n - P_n(t)\,\mu_n - P_n(t)\,\delta_n + P_{n-1}(t)\,\lambda_{n-1} \\ &\quad + P_{n+1}(t)\,\mu_{n+1} + P_{n+1}(t)\,\delta_{n+1} \\ &= -P_n(t)\,(\lambda_n+\mu_n+\delta_n) + P_{n-1}(t)\,\lambda_{n-1} \\ &\quad + P_{n+1}(t)\,(\mu_{n+1}+\delta_{n+1}) \end{aligned} \qquad (2.2)$$

The steady-state solution for P_n is sought. Steady-state implies that,

$t \to \infty$ or $dP_n(t)/dt = 0$, so Equation (2.2) becomes

$$0 = -P_n(\lambda_n+\mu_n+\delta_n) + P_{n-1}\lambda_{n-1} + P_{n+1}(\mu_{n+1}+\delta_{n+1}) \quad 0<n<N \qquad (2.3)$$

$$0 = -P_0\lambda_0 + P_1(\mu_1+\delta_1) \quad n=0 \qquad (2.4)$$

$$0 = -P_N(\mu_N+\delta_N) + P_{N-1}\lambda_{N-1} \quad n=N \qquad (2.5)$$

From Equation (2.4),

$$P_1 = \frac{P_0\lambda_0}{(\mu_1+\delta_1)} = \frac{\lambda_0}{(\mu_1+\delta_1)}P_0 \qquad (2.6)$$

From Equation (2.3) when $n=1$,

$$P_2 = \frac{P_1(\lambda_1+\mu_1+\delta_1) - P_0\lambda_0}{(\mu_2+\delta_2)} \qquad (2.7)$$

Substituting Equation (2.6) into Equation (2.7) and solving for P_2 gives

$$P_2 = \frac{\lambda_0\lambda_1}{(\mu_1+\delta_1)(\mu_2+\delta_2)}P_0 \qquad (2.8)$$

The following can be shown using mathematical induction:

$$P_n = \prod_{k=1}^{n} \frac{\lambda_{k-1}}{(\mu_k+\delta_k)}P_0, \qquad 1 \le n \le N \qquad (2.9)$$

To find the value of P_0, note that

$$\sum_{n=0}^{N} P_n = 1 \qquad (2.10)$$

is true. So from (2.9),

$$P_0 + \sum_{n=1}^{N}\prod_{k=1}^{n} \frac{\lambda_{k-1}}{(\mu_k+\delta_k)}P_0 = 1 \qquad (2.11)$$

Solving for P_0 gives

$$P_0 = \left[1 + \sum_{n=1}^{N}\prod_{k=1}^{n} \frac{\lambda_{k-1}}{(\mu_k+\delta_k)}\right]^{-1} \qquad (2.12)$$

The failure rate of an item in the population is given by $\lambda = 1/\text{MTBF}$. It is assumed that a failed item is instantly replaced with a spare from inventory, as long as the level of inventory L is greater than 0. Therefore, the failure rate of the population remains constant at $K\lambda$ where K is the number of operational items. If there does not exist a spare in inventory when an item fails then the failure rate of the population is $(K-(n-L)$ where n is the number of failed items. So the failure rate of the entire population when n items fail can be represented as

$$\lambda_n = \begin{cases} K\lambda & n=0,1,\ldots,L \\ K-(n-L))\lambda & n=L+1,L+2,\ldots,N \end{cases} \qquad (2.13)$$

The proportion of items which arrive at the repair facility on average is $(1-\delta$, thus the repair rate of the repair facility is given by

$$\mu_n = \begin{cases} (1-\delta)\, n\mu & n=0,1,\ldots,c-1 \\ (1-\delta)\, c\mu & n=c,c+1,\ldots,N \end{cases} \qquad (2.14)$$

and the discard rate, when n items have already failed is

$$\delta_n = \delta\lambda_n = \begin{cases} \delta K\lambda & n=0,1,\ldots,L \\ \delta(K-(n-L))\lambda) & n=L+1,L+2,\ldots,N \end{cases} \qquad (2.15)$$

Equations (2.13),(2.14), and (2.15) can be substituted into Equation (2.9) to obtain the probability mass function for the number of failed items. When $c < L$,

$$P_n = \begin{cases} (K\lambda)^n \alpha_n P_0, & n=0,1,\ldots,c \\ \dfrac{(K\lambda)^n}{(c\mu-\delta(c\mu-K\lambda))^{n-c}} \alpha_c P_0, & n=c+1,c+2,\ldots,L \\ \binom{K}{n-L}(n-L)! \dfrac{K^L\lambda^n}{(c\mu-\delta(c\mu-K\lambda))^{L-c}} \alpha_c C_1 P_0, & n=L+1,\ldots,N \end{cases} \qquad (2.16)$$

and when $c > L$,

$$P_n = \begin{cases} (K\lambda)^n \alpha_n P_0 & n=0,1,\ldots,L \\ \binom{K}{n-L} (n-L)!\, K^L \lambda^n \alpha_L B_{n-L} P_0 & n=L+1,L+2,\ldots,c \\ \binom{K}{n-L} (n-L)!\, K^L \lambda^n \alpha_L B_{c-L} C_2 P_0 & n=c+1,\ldots,N \end{cases} \quad (2.17)$$

where,

$$\begin{aligned} \alpha_n &= \prod_{j=1}^{n} ((1-\delta)\, j\mu + \delta K\lambda)^{-1} \\ \beta_n &= \prod_{j=1}^{n} ((1-\delta)\,(L+j)\,\mu + \delta\,(K-j)\,\lambda)^{-1} \\ C_1 &= \prod_{j=1}^{n-L} ((1-\delta)\, c\mu + \delta\,(K-j)\,\lambda)^{-1} \\ C_2 &= \prod_{j=c-L+1}^{n-L} ((1-\delta)\, c\mu + \delta\,(K-j)\,\lambda)^{-1} \end{aligned} \quad (2.18)$$

When $c = L$, the Equations (2.16) and (2.17) are equal and the middle term no longer applies. Thus, the probability mass function for the Critical Spares Inventory System has been derived. Unfortunately, because of the complexity of the equations, the most realistic manner in which to analyze the system is through the use of a computer. An example will be given later, utilizing the CSIS application in solving for the P_n.

III. ECONOMIC ORDER QUANTITY (EOQ)

To determine the EOQ and the level of spares when procurement should take place, we note that attrition of the inventory is due to discarding a proportion of the failed items at a rate equal to the failure rate. So demand on inventory due to discarding (D_δ) is equal to the expected value of the discard rate, or

$$\begin{aligned} D_\delta &= E(\delta_n) \\ &= E(\delta\lambda_n) \\ &= \delta\left(\sum_n \lambda_n P_n\right) \end{aligned} \tag{3.1}$$

and the total demand on inventory, D, is given by,

$$D = \frac{D_\delta}{\delta} = E(\lambda_n) \tag{3.2}$$

To determine the optimal level of spares when procurement should take place, L^*, we only need to look at the lead time demand. Let T be the procurement lead time. Then,

$$L^* = D_\delta T = \delta E(\lambda_n)\, T \tag{3.3}$$

since inventory attrition is due to the discard demand. (See Figure 2).

To determine the EOQ (Q^*), let,

A = ordering cost,

C = item cost, and

i = inventory carrying cost as a proportion of the item cost.

Then,

$$Q^* = \sqrt{\frac{2AD_\delta}{iC}}$$

The time between procurements is equal to $\frac{Q^*}{D_\delta}$.

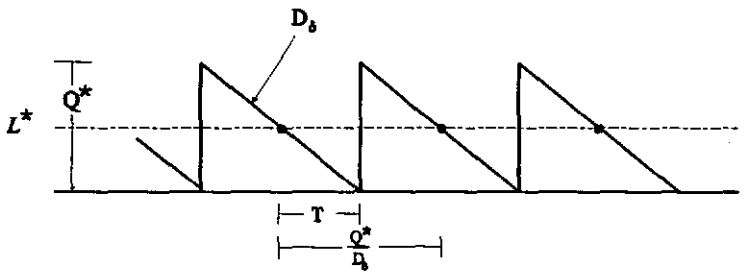


Figure 2. CSIS EOQ Model with Demand D_δ

IV. THE COMPUTER ALGORITHM

Define S as the event that a *stock out occurs*. Then, the probability of a stock out, $P(S)$, is determined using the probability mass function for the number of failed items. Once the probability distribution is determined for all n, it is a matter of summing all P_n for $n=L+1,L+2,\ldots,N$, since a stockout occurs only if $L+1$ or more items fail. So,

$$P(S) = \sum_{n=L+1}^{N} P_n \qquad (4.1)$$

A more likely scenario might be that the user would want to know what level of spares to have on hand in order to keep the probability of a stockout below a given level. So, given a safety factor, which equals $1 - P(S)$, the problem is to determine the level of spares to maintain, L.

The simplest approach is iterative. We let $L=0$ and compute the probability of a stockout. If the computed probability is greater than $P(S)$, then we increment L by 1 and recompute the probability. This process is repeated until the computed probability is less than or equal to $P(S)$. At that point L is the minimum such value so that the required safety factor is met.

Example 1

Let CSIS have the following system parameters:

MTBF = 1440 hours,
MTTR = 80 hours,
δ = 0.50,
c = 3 servers,
K = 20 parts,

and suppose an 85 percent safety factor against stockout is desired. Then, using the CSIS application, the steady-state level of spares, L, is determined to be 3. Thus, over the long

run, 3 spares should be maintained in inventory at any given time. The probability distribution for CSIS with the parameters above is given in Figure 3. Now, suppose we wish to determine the optimal quantity of parts to order and the level of inventory when the order should take place. First, we must determine the discard demand, (D_δ) . From the probability distribution, the expected failure rate is calculated to be 19.8 items for every 1,440 hours. With a discard proportion of 0.50, the discard demand, (D_δ) = 0.50 x 19.8/1440 = 0.00687 parts per hour or approximately 5 items per month. Now let,

T = 720 hours,
A = \$ 100.00,
i = 0.02(as a fraction of item cost on monthly basis), and
C = \$ 1000.00.

Then, the order level is given by,

$$L^* = D_\delta T = 0.0068 \times 720 \approx 5$$

and the optimal quantity to order is thus,

$$Q^* = \sqrt{\frac{2AD_\delta}{iC}} = \sqrt{\frac{2(\$100)(5)}{(0.02)(\$1000)}} \approx 7$$

So, when the level of inventory on hand reaches 5 an order should be placed for 7 parts.

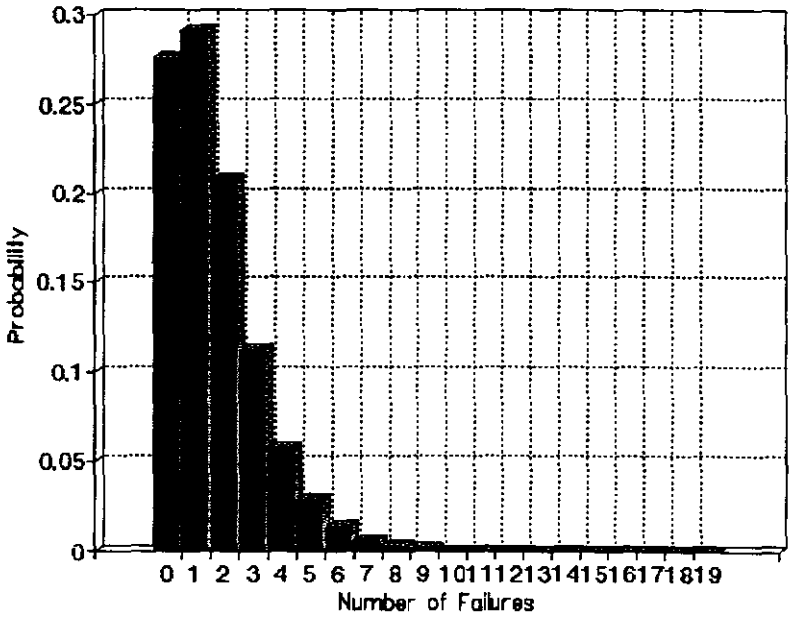


Figure 3. CSIS Probability Distribution for $L = 3$.

Example 2

Suppose the level of inventory, optimal order quantity, and optimal order level are sought for CSIS with an item MTBF of 1000 hours. Assume that all other variables for CSIS is the same as in Example 1 and again an 85 percent safety factor, against stockout, is desired. Using the CSIS application, it is determined that the steady-state level of spares is 5. With a discard proportion of 0.50, the discard demand, (D_δ) = 0.50 x 19.7/1000 = 0.0098 parts per hour or approximately 7.06 parts per month. With,

T = 720 hours,

A = \$ 100.00,

i = 0.02(as a fraction of item cost on **monthly** basis), and

C = \$ 1000.00,

the order level is given by,

$$L^* = D_\delta T = 0.0098 \times 720 \approx 7$$

and the optimal quantity to order is thus,

$$Q^* = \sqrt{\frac{2AD_\delta}{iC}} = \sqrt{\frac{2(\$100)(7.06)}{(0.02)(\$1000)}} \approx 8$$

So, when the level of inventory on hand reaches 7 an order should be placed for 8 items.

V. UTILIZING CSIS In The TESTING Of ALTERNATIVE DESIGNS

This report focuses on a logistics subsystem that is assumed to be in existence. Hence, there are only four variables under the direct control of a decision maker. These are: 1) inventory level, 2) safety factor, 3) procurement level, and 4) procurement quantity. The CSIS model provides a means of testing these four decision variables in the presence of the uncontrollable system parameters. The system parameters in the CSIS model include, MTBF, MTTR, number of repair channels, discard proportion, procurement lead-time, item cost, procurement cost, holding cost, and number of deployed items. (See Table I).

A necessary extension to CSIS is to consider the entire repairable equipment population system. In this case the decision maker would be able to evaluate alternative designs for each set of design dependent parameters. By applying an evaluation measure, such as system cost, the decision maker could generate and evaluate design alternatives with the objective of identifying the design alternative which minimizes total system cost. The decision maker would have control over design dependent parameters such as MTBF and MTTR. Also, the number of repair channels in the repair facility would fall under the direct control of the decision maker as well as inventory level, safety factor, procurement level, and procurement quantity. (See Table II).

The CSIS software application allows the user to test candidate designs through the use of sensitivity graphs and modifications to design dependent parameters. Optimal values for decision variables can be found for each set of design dependent parameters entered. Since no evaluation measure is used, it would be difficult for the decision maker to evaluate the alternative designs in a logical manner. For example: it is likely that there will exist a required safety factor against stockout. The objective would be to find the minimum such inventory level so that the requirement is met. This could be accomplished by choosing a repairable item with a high reliability. (See Figure 4). Since greater reliability usually implies greater acquisition cost, this alternative could be too costly in terms of total system cost. Thus, it is necessary to consider an evaluation measure to rationally determine the best candidate design.

Table I. Decision Variables and System Parameters

Variables/Parameters	Decision Variables	Uncontrollable System Parameters
Inventory Level	X	
Safety Factor	X	
Reorder level	X	
Procurement quantity	X	
MTBF		X
MTTR		X
Number of repair channels		X
Discard proportion		X
Lead-time		X
Item cost		X
Procurement cost		X
Holding cost		X
Demand		X

Table II. Decision Variables and Decision Dependent/Independent System Parameters

Variables/Parameters	Decision Variables	Decision Dependent Parameters	Decision Independent Parameters
Inventory Level	X		
Safety Factor	X		
Reorder level	X		
Procurement quantity	X		
MTBF		X	
MTTR		X	
Number of repair channels	X		
Discard proportion			X
Lead-time			X
Item cost		X	
Procurement cost			X
Holding cost			X
Demand			X

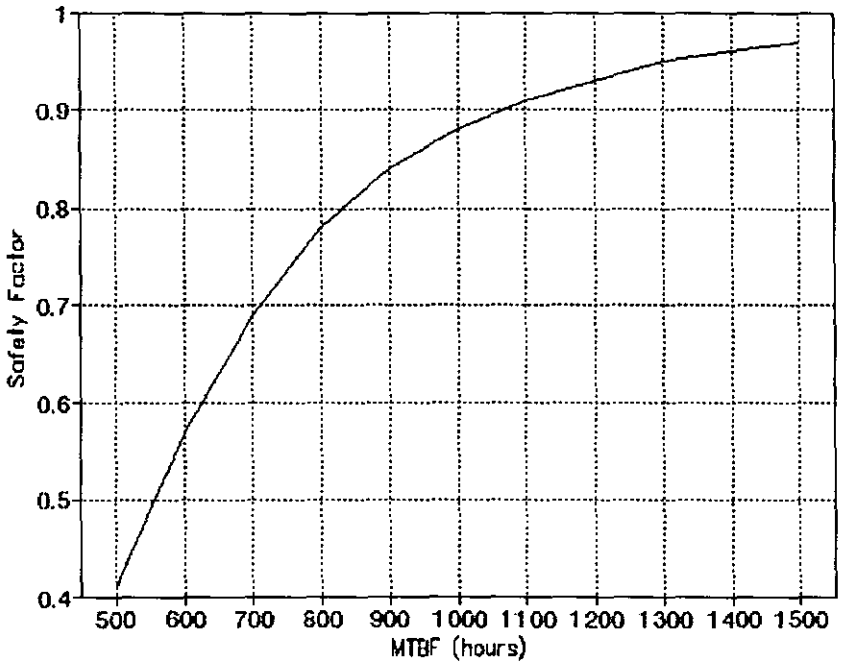


Figure 4. Sensitivity Graph of Safety Factor vs. MTBF

VI. SUMMARY

This report focused on the suboptimization of a logistics support system that is assumed to be in existence. A minimum steady-state level of spares was sought to meet a demand due to the discarding of failed items. Once the sparing level was determined an EOQ model was applied to find optimal values for the decision variables which include the procurement level and the procurement quantity.

This report addresses a concern for determining appropriate sparing levels given fixed system parameters such as the reliability and maintainability of the equipment item in the population. It also discusses how the CSIS application could be used in the testing of alternative designs in the face of decision variables. This model does not address full optimization of the operational system. If certain equipment items are deemed critical to mission success of a deployed system and a stockout is unacceptable then the Critical Spares Inventory System model could be utilized to determine appropriate sparing levels.

A needed extension is to formulate and optimize an evaluation function for the entire system. This evaluation function would simultaneously consider the five decision variables; inventory level, procurement quantity, procurement level, safety factor, and number of repair channels. The evaluation function could then be used to evaluate alternative designs by finding optimum values for the decision variables for each set of design dependent system parameters.

VII. REFERENCES

[1] White, J.A., J.W. Schmidt, and G.K. Bennett, Analysis of Queueing Systems, Academic Press, 1975

[2] Fabrycky, W.J. and B.S. Blanchard, Life-Cycle Cost and Economic Analysis, Prentice-Hall, Inc., 1991

[3] Fabrycky, W. J., "Logistic System Optimization Through Design And Source Dependent Parameters", Proceeding: Fall Conference, Institute of Industrial Engineers, San Antonio, TX, October, 1990

[4] Blanchard, B. S. and W. J. Fabrycky, Systems Engineering and Analysis, 2nd Edition, Prentice-Hall, Inc., 1990

APPENDIX A

CSIS Software User's Guide

INTRODUCTION

The Critical Spares Inventory System Software is an IBM PC based application which utilizes the CSIS algorithm described in this paper. The CSIS application can be used to 1) compare alternative system designs, 2) find the optimal steady-state inventory level for specific system parameters, 3) calculate procurement rates, 4) graph results of the computed probability distribution, and 5) perform sensitivity analysis on the safety factor based on any one of four system variables. This Appendix will provide an overview of the functionality of the CSIS application, henceforth, called CSIS.

HARDWARE/SOFTWARE RECOMMENDATIONS

Processor:	IBM PC, AT 286 or later
Memory:	1 Megabyte RAM
Hard disk:	2.4 megabytes or greater
Video:	EGA,VGA graphics
Software:	MS-DOS 3.3 or later

RUNNING CSIS

Change default to the directory containing the CSIS application. At the MS-DOS prompt enter:

CSIS

After a few seconds of program load, a banner page will be displayed. Hit any key to display the main screen of the CSIS application.

MAIN FUNCTIONS

CSIS contains five main functions located along the main menu bar. These functions are selected by using the right and left arrows provided on extended-101 keyboards and found on the numeric key pad. The five functions are *Input Data, Compute, Graphs, Print,* and *Help*. The currently selected function will be boxed in blue with white lettering. If Input Data, Compute, or Graphs is selected by pressing the return key, <CR>, then a pulldown menu will be displayed. The pulldown menu will contain a list of subfunctions which can be selected using the up and down arrow keys and by pressing <CR>. The Print and Help functions do not have associated pulldown menus. Help can be activated at any time other than when a graph is displayed. However, certain conditions must be met before Print can be selected.

INPUT DATA

Input Data contains four subfunctions, *New, Retrieve, Save,* and *Edit*. These functions all pertain to entering the necessary variables and parameters before any calculations or graphs can be accomplished. *New* allows the user to create a new data set by entering directly from the keyboard. This data can then be saved to a file on the disk by selecting the *Save* subfunction. Likewise, *Retrieve* will allow the user to select and display a previously saved data set, while *Edit* can be used to modify any of the already displayed parameters.

When the user selects *New* from the pulldown menu a list of parameter and variable descriptions is displayed on the left half of the screen like the following:

```
              MTBF:
              MTTR:
   Repair Channels:
Discard Proportion:
        Population:

   Inventory Level:
     Safety Factor:

         Lead Time:
     Ordering Cost:
     Carrying Cost:
         Part Cost:
```

A black box will appear to the right of the MTBF: field. This box indicates that CSIS is expecting an input. A description of this input, including units, is displayed along the bottom of the screen to aide in the input for the current field. The first block or group of fields (MTBF: - Population:) is required input to enable any further computations or graphs. Once the data is input correctly and within the specific range, <CR> is pressed to enter the data and continue with the next field. The black box will move down to the next field in the list. After Population: is entered a white box will appear to the right of the Inventory Level: field. The white box indicates that entering data for this field is optional. However, either the Inventory Level: or the Safety Factor: field must be a non-zero value for further computations. Whichever value is zero or blank, when the Compute/Probability Distribution subfunction is selected, will be automatically calculated and entered. The Lead Time: to Part Cost: fields are also optional input data. Values for these fields only have to be entered if it is desired to compute the optimal procurement rates. A white box is also used for the input of this data.

NOTE The ESCAPE key can be pressed at any time to exit this subfunction.

The *Retrieve* subfunction is used to retrieve data sets from the hard drive which were previously saved using the *Save* subfunction. When *Retrieve* is selected from the

pulldown menu, a box will appear in the middle of the screen for input of the data set name. the name can be entered directly (without file extension) followed by a <CR>. If the data set exists then the list of field descriptions will appear with values supplied. If the data set is not known then a list of all of the data sets can be displayed and selected from by entering <CR> in place of the data set name. The up and down arrow keys are used to scroll through the list of data sets that are displayed. <CR> is used to select the current data set for retrieval.

The *Save* subfunction, when selected, will save all of the data currently displayed. It will not, however, save any computed probability distributions which are also currently displayed. If the list of field descriptions is not currently displayed on the screen, selecting the *Save* subfunction will initiate an error message which will be displayed in red at the bottom right-hand side of screen for a few seconds. The *Save* subfunction, once selected, will prompt for a data set name. This name can be up to eight characters long. The data set will be saved in the same directory that CSIS was executed from and can be immediately retrieved using the *Retrieve* subfunction described above.

The *Edit* subfunction allows the modification of the currently displayed data set. This function will initiate an error message if no data set is currently displayed. If a data set is displayed then a black box will appear to the right of the MTBF: field descriptor. This field can be modified or a <CR> will accept the default value and continue on to the next field. The ESCAPE key can be pressed to exit this function early. However, a <CR> must be selected to enter the value for the current field being edited.

COMPUTE

Compute allows the user to calculate and display textually the CSIS probability distribution, procurement quantity and level, and expected number of failures. If Compute is selected from the main menu bar a pulldown menu will appear containing *Probability Distribution, Procurement Quantity/Level,* and *Expected Number Failures.*

The *Probability Distribution* subfunction, when selected, will calculate and display the CSIS probability distribution for the current data set. The distribution is displayed on the right side of the screen in red. Also, the Inventory Level and Safety Factor fields will be displayed in red on the left side of the screen next to the respective field descriptors. There may be a discrepancy between the Safety Factor before the probability distribution was computed and after due to rounding and the fact that only an integer inventory level is possible. A maximum of ten probabilities for the number of failures will be displayed at any given time. The PgUp and PgDn keys can be used to scroll through all of the probability distribution that was computed, as long as the distribution is displayed in red. Pressing the ESCAPE key will exit out of this mode and the distribution, Inventory Level and Safety Factor fields will be displayed in white. The only way to browse through the distribution is to select this subfunction once again.

The *Procurement Quantity/Level* subfunction can only be selected after the probability distribution has been computed using the *Probability Distribution* subfunction. This subfunction will calculate and display the optimal procurement level and number to procure provided 1) the probability distribution has been computed and 2) the Lead Time: - Part Cost: fields have been entered with valid values. The computed values will be displayed in a box in the center of the screen. Pressing any key will remove the box from view.

The *Expected Number Failures* subfunction will calculate and display the expected number of failures for the current data set. This subfunction can only be selected after the probability distribution has been computed. Like the *Procurement Quant./Level* subfunction this subfunction will display the value in a box in the center of the screen. Pressing any key will remove the box from view.

GRAPHS

The *Graphs* function allows the user to graph the currently displayed probability distribution and a sensitivity graph of the safety factor as a function of one of four system

parameters.

The *Probability Distribution* subfunction, once selected, will display the current textually displayed probability distribution in the form of a bar graph. Pressing any key will remove the graph from view and return to the CSIS main screen. If a value is changed using the *Edit* subfunction then the probability distribution must be recomputed before the probability distribution graph will reflect the change.

The *Sensitivity* subfunction provides a sensitivity graph for the Safety Factor as a function of one of four system parameters. Once this subfunction is selected a box will appear containing the four parameters. These are, MTBF, MTTR, population, and discard proportion. Any one of these parameters can be selected using the up and down arrow keys and pressing the <CR>. The current value of each parameter is used as the middle value for which the variation takes place. That is, if MTBF were selected, then the MTBF value would be varied one half the magnitude of MTBF in the plus or minus direction from the current MTBF. The safety factor will be computed and graphed, in the form of a line graph, for an increment of MTBF equal to one tenth of the MTBF. For example, assume we had a MTBF of 1000 hours, then the safety factor would be computed for an MTBF of 500 to 1500 with an increment of 100. Of course all other parameters stay constant. Only the probability distribution is recomputed for step of the varying MTBF. This same method is used for the other system parameters.

PRINT

The *Print* function is used to print the current data set and the currently displayed probability distribution if one has been computed. This data is routed to a printer connected to the LPT1 port of the computer currently running CSIS. If the printer is offline or is for some reason not accessible via this port then a message will be displayed stating that the printer is not ready.

HELP

The *Help* function will display the help text which accompanies the CSIS application. PgUp, PgDn, and the arrow keys can be used to page through the text. Pressing the F key will prompt for a text string to search in the Help document.

NOTE PRESS THE "X" KEY TO RETURN TO CSIS APPLICATION FROM HELP.

ESCAPE

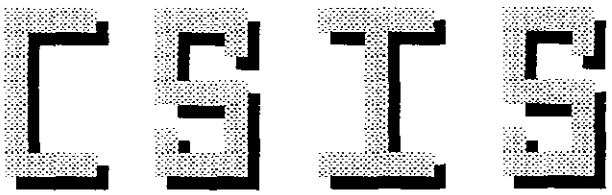
This key, in general, is used to exit the current state of the program. For example, if a pulldown menu is currently displayed and no selection is desired the ESCAPE key can be pressed to return to the main menu bar. Also the ESCAPE key can be used to exit the program. If the menu bar is displayed and, the program is not in the Edit mode, probability distribution is not being browsed, and a pulldown menu is not displayed, then a red box will appear prompting to exit the CSIS application. If "Y" is entered followed by <CR> then the program will return to DOS. If any other key is entered followed by <CR>, the program will return to the main menu bar.

APPENDIX B

CSIS Software Code Listings

```
; Script:     CSIS.SC
; Version:    1.0
; Date:       8 NOV 1991
;
AUTOLIB = "CSIS"                              ; Tell paradox where the
                                              ;   procedures are

ver     = "CSIS Ver 1.0        Critical Spares Inventory System"

PROC CSIS()

   Intro()                                    ; Display the Intro Screen
   Repaint()                                  ; Paint the Canvas Blue
   InitVar()                                  ; Establish subMenu Items

   choice = 1
   retval = 0                                 ; Give retval arbitrary value
   DispMenu(TRUE,choice)                      ; Display the Main Menu

   WHILE 1=1                                  ; Loop forever

     SWITCH
     CASE retval = 101:
       NewData()                              ; Get the input data
     CASE retval = 102:
       Retrieve()                             ; Retrieve a data file
     CASE retval = 103:
       IF (inputFlag) THEN                    ; If user entered data
         SaveData()                           ;   Save data to file
       ELSE
         BEEP BEEP                            ; No data to edit
         @ 24,64
         ?? "No data to save"
         PAINTCANVAS ATTRIBUTE 79 24,64,24,78
         SLEEP 2500
       ENDIF
     CASE retval = 104:
       IF (inputFlag) THEN                    ; If  user entered data
         EditData()                           ;  Edit the existing data
       ELSE
         BEEP BEEP                            ; No data to edit
         @ 24,64
         ?? "No data to edit"
         PAINTCANVAS ATTRIBUTE 79 24,64,24,78
         SLEEP 2500
       ENDIF
     CASE retval = 201:
       CPDist()                               ; Compute the distribution
     CASE retval = 202:
       CProcure()                             ; Compute procurment data
     CASE retval = 203:
       CExpect()                              ; Compute the expected failures
     CASE retval = 301:
       GPDist()                               ; Graph the distribution
     CASE retval = 302:
       GSens()                                ; Graph sensitivity data
     CASE retval = 4:
       Cprint()                               ; Print the data
     CASE retval = 5:
       GetHelp()                              ; Display user's guide
     ENDSWITCH

     IF (retval > 100 AND retval < 105) THEN
       choice = 1                             ; Reset the choice for display
     ELSE IF (retval > 200 AND retval < 204) THEN
       choice = 2                             ; Reset the choice for display
     ELSE IF (retval > 300 AND retval < 303) THEN
       choice = 3                             ; Reset the choice for display
     ELSE IF (retval = 4) THEN
```

```
      choice = 4                                ; Reset the choice for display
    ELSE IF (retval = 5) THEN
      choice = 5                                ; Reset the choice for display
    ELSE IF (retval = 99) THEN
      CheckExit()                               ; Ask if user wants to exit
      IF retval THEN
        QUITLOOP                                ; Quit CSIS
      ELSE
        choice = 1                              ; Reset the menu to first choice
      ENDIF
           ENDIF
         ENDIF
       ENDIF
     ENDIF
    ENDIF

    CURSOR OFF
    DispMenu(FALSE,choice)                      ; Display the Main Menu

  ENDWHILE


ENDPROC ;  CSIS

CSIS()                                          ; Call main procedure
RELEASE VARS ALL


; Script:     INTRO.SC
; Version:    1.0
; Date:       8 NOV 1991
;

PROC Intro()
  Private Y,i,j,x,cnt
  CURSOR OFF
  CANVAS OFF
  CLEARALL
  CLEAR                                 ; Clear all tables on canvas and workspaces
  PAINTCANVAS ATTRIBUTE 112 0, 0, 24, 79
  STYLE ATTRIBUTE 112
  @ 0, 0
  TEXT
```



C R I T I C A L S P A R E S I N V E N T O R Y S Y S T E M

by

Darrin Joel Lipscomb

for

VIRGINIA POLYTECHNIC INSTITUTE AND STATE UNIVERSITY

```
  ENDTEXT
  PAINTCANVAS ATTRIBUTE 113 3,11,3,18
  PAINTCANVAS ATTRIBUTE 113 3,24,3,31
  PAINTCANVAS ATTRIBUTE 113 3,38,3,47
  PAINTCANVAS ATTRIBUTE 113 3,54,3,61
  PAINTCANVAS ATTRIBUTE 113 4,11,4,12
  PAINTCANVAS ATTRIBUTE 113 4,24,4,25
  PAINTCANVAS ATTRIBUTE 113 4,30,4,31
  PAINTCANVAS ATTRIBUTE 113 4,42,4,43
  PAINTCANVAS ATTRIBUTE 113 4,54,4,55
  PAINTCANVAS ATTRIBUTE 113 4,60,4,61
  PAINTCANVAS ATTRIBUTE 113 5,11,5,12
  PAINTCANVAS ATTRIBUTE 113 5,24,5,25
  PAINTCANVAS ATTRIBUTE 113 5,42,5,43
  PAINTCANVAS ATTRIBUTE 113 5,54,5,55
  PAINTCANVAS ATTRIBUTE 113 6,11,6,12
  PAINTCANVAS ATTRIBUTE 113 6,24,6,31
  PAINTCANVAS ATTRIBUTE 113 6,42,6,43
  PAINTCANVAS ATTRIBUTE 113 6,54,6,61
  PAINTCANVAS ATTRIBUTE 113 7,11,7,12
  PAINTCANVAS ATTRIBUTE 113 7,24,7,25
  PAINTCANVAS ATTRIBUTE 113 7,30,7,31
  PAINTCANVAS ATTRIBUTE 113 7,42,7,43
  PAINTCANVAS ATTRIBUTE 113 7,54,7,55
  PAINTCANVAS ATTRIBUTE 113 7,60,7,61
  PAINTCANVAS ATTRIBUTE 113 8,11,8,12
  PAINTCANVAS ATTRIBUTE 113 8,24,8,25
  PAINTCANVAS ATTRIBUTE 113 8,30,8,31
  PAINTCANVAS ATTRIBUTE 113 8,42,8,43
  PAINTCANVAS ATTRIBUTE 113 8,54,8,55
  PAINTCANVAS ATTRIBUTE 113 8,60,8,61
  PAINTCANVAS ATTRIBUTE 113 9,11,9,18
  PAINTCANVAS ATTRIBUTE 113 9,24,9,31
  PAINTCANVAS ATTRIBUTE 113 9,38,9,47
  PAINTCANVAS ATTRIBUTE 113 9,54,9,61

  PAINTCANVAS ATTRIBUTE 116 12,6,12,68
  CANVAS ON

  WHILE NOT CHARWAITING()
    SLEEP 100
  ENDWHILE

  x = GETCHAR()                          ;Clear the buffer

  cnt = 0
  WHILE (cnt < 25)
    @ cnt,0
    ?? FILL(" ",80)
    PAINTCANVAS ATTRIBUTE 16 cnt,0,cnt,79
    cnt = cnt + 1
  ENDWHILE

ENDPROC ;  Intro
```

```
; Script:      REPAINT.SC
; Version:     1.0
; Date:        8 NOV 1991
;

PROC Repaint()
  CANVAS OFF
  CLEARALL
  CLEAR                                  ; Clear all tables on canvas and workspaces
  CURSOR OFF
  PAINTCANVAS ATTRIBUTE 31 0, 0, 24, 79 ; Paint the canvas blue with white letters
  STYLE ATTRIBUTE 31
  CANVAS ON
ENDPROC ; Repaint
```

```
; Script:     INITVAR.SC
; Version:    1.0
; Date:       8 NOV 1991
;

PROC InitVar()

    ARRAY submenu1[4]                       ; Declare submenu's as arrays
    ARRAY submenu2[3]                       ; Declare submenu's as arrays
    ARRAY submenu3[2]                       ; Declare submenu's as arrays
    ARRAY subinfo1[4]
    ARRAY subinfo2[3]
    ARRAY subinfo3[2]

    ARRAY lenmenu[5]                        ; Stores the length of submenu's

    submenu1[1] = "New          "
    submenu1[2] = "Retrieve     "
    submenu1[3] = "Save         "
    submenu1[4] = "Edit         "

    submenu2[1] = "Probability Distribution"
    submenu2[2] = "Procurement Quant./Level"
    submenu2[3] = "Expected Number Failures"

    submenu3[1] = "Probability Distribution"
    submenu3[2] = "Sensitivity"

    subinfo1[1] = "Create a new set of data.                                                    "
    subinfo1[2] = "Retrieve an existing set of data from the disk.                              "
    subinfo1[3] = "Save the current set of data to disk.                                        "
    subinfo1[4] = "Edit the current set of data.                                                "

    subinfo2[1] = "Compute the probability of n failed parts for all n.                         "
    subinfo2[2] = "Compute the optimal procurement quantity and level to order.                 "
    subinfo2[3] = "Compute the expected number of failures.                                     "

    subinfo3[1] = "Graph the probability distribution.                                          "
    subinfo3[2] = "Graph the Safety Factor as a function of a system parameter.                 "

    lenmenu[1] = 4
    lenmenu[2] = 3
    lenmenu[3] = 2
    lenmenu[4] = 0
    lenmenu[5] = 0

    inputflag  = FALSE              ; Set these flags so other routines
    outputflag = FALSE              ; know what is on the canvas
    outputflag1= FALSE
    outputflag2= FALSE
    outputFlag3= FALSE
    edited     = FALSE

    mtbf     = 0                    ; Init all input variables
    mttr     = 0
    chan     = 0
    delta    = 0
    pp      = 0
    level    = 0
    sf       = 0
    ldtime   = 0
    ordcost  = 0
    cc       = 0
    prtcost  = 0

    tablename = ""
ENDPROC ;  InitVar
```

```
; Script:     DISPMENU.SC
; Version:    1.0
; Date:       8 NOV 1991
;

PROC DispMenu(rpflag,choice)

   ARRAY mainmenu[5]
   ARRAY infomenu[5]

   CURSOR OFF
   numitems  = 5
   normcol   = 48
   highltcol = 31
   pdFlag    = FALSE

   mainmenu[1] = "Input Data "
   mainmenu[2] = "   Compute "
   mainmenu[3] = "   Graphs  "
   mainmenu[4] = "  Print    "
   mainmenu[5] = "Help       "

   infomenu[1] = "Enter system parameters and design variables.                                    "
   infomenu[2] = "Probability distribution, Optimal order quantity/level, Expected failures.      "
   infomenu[3] = "Bar graph of probability distribution, or sensitivity analysis.                 "
   infomenu[4] = "Print the input data and results.                                               "
   infomenu[5] = "Display the User's Guide to CSIS.                                              "
   CURSOR OFF

   IF (rpflag) THEN
     CANVAS OFF
     Repaint()                          ; Repaint the PAL canvas
     @ 0,1
     ?? ver
     @ 0,70
     ?? TODAY()                         ; Displays todays date
     PAINTCANVAS ATTRIBUTE 79 0,0,0,79  ; Display first row of screen
   ENDIF

   PAINTCANVAS ATTRIBUTE 51 2,2,4,76
   PAINTCANVAS ATTRIBUTE 0 5,3,5,77
   PAINTCANVAS ATTRIBUTE 0 3,77,4,77

   currow = 3
   STYLE ATTRIBUTE normcol
   FOR i FROM 1 TO numitems
      colmn = 5+(i-1)*12
      @ currow,colmn
      ?? mainmenu[i]
   ENDFOR

   curcol = 5+(choice-1)*12
   PAINTCANVAS ATTRIBUTE highltcol
   currow,curcol,currow,curcol+11      ; Highlight first selectable menu item

   i=choice
   @ 24,0
   ?? infomenu[i]

   CANVAS ON

    WHILE 1=1                          ; LOOP forever

      IF (CHARWAITING() OR pdFlag) THEN
        IF (NOT pdFlag) THEN
          ky = GETCHAR()
        ELSE
```

```
        ky = retval
        pdFlag = FALSE
      ENDIF
      IF (ky = -75) AND NOT curcol = 5 THEN
        PAINTCANVAS ATTRIBUTE normcol currow, curcol, currow, curcol+11
        curcol = curcol - 12
        PAINTCANVAS ATTRIBUTE highltcol currow, curcol, currow, curcol+11
        i = i - 1
      ELSE IF (ky = -75) AND curcol = 5 THEN
        PAINTCANVAS ATTRIBUTE normcol currow, curcol, currow, curcol+11
        curcol = curcol + 4*12
        PAINTCANVAS ATTRIBUTE highltcol currow, curcol, currow, curcol+11
        i = 5
      ELSE IF (ky = -77) AND NOT curcol = 53 THEN
        PAINTCANVAS ATTRIBUTE normcol currow, curcol, currow, curcol+11
        curcol = curcol + 12
        PAINTCANVAS ATTRIBUTE highltcol currow, curcol, currow, curcol+11
        i = i + 1
      ELSE IF (ky = -77) AND curcol = 53 THEN
        PAINTCANVAS ATTRIBUTE normcol currow, curcol, currow, curcol+12
        curcol = curcol - 4*12
        PAINTCANVAS ATTRIBUTE highltcol currow, curcol, currow, curcol+11
        i = 1
      ELSE IF (ky = 27) THEN
        choice = 99
        QUITLOOP                                    ; Exit this while loop
      ELSE IF (ky = 13) THEN
        choice = INT(curcol/12)+1
        IF (choice < 4) THEN
          PullDown(choice)                          ; Show the pulldown menu
        ELSE
          QUITLOOP
        ENDIF
        IF (retval > 100 AND retval < 304) THEN
          choice = retval
          QUITLOOP
        ENDIF
        pdFlag = TRUE
      ELSE
        BEEP
           ENDIF
          ENDIF
         ENDIF
        ENDIF
       ENDIF
      ENDIF
      @ 24,0
      ?? infomenu[i]                                ;  display the info
    ENDIF
  ENDWHILE

  RETURN choice
ENDPROC ;  DispMenu
```

```
; Script:      PULLDOWN.SC
; Version:     1.0
; Date:        8 NOV 1991
;


PROC Pulldown(choice)

    PRIVATE i,maxlen,ky,curline,menulen,templen

    CURSOR OFF
    normcol = 48
    highltcol = 31

    ARRAY tempmenu[lenmenu[choice]]          ; Declare a tempory for menu's
    ARRAY tempinfo[lenmenu[choice]]

    maxlen = 0                               ; maxlen used to get longest element
    menulen = lenmenu[choice]

    SWITCH                                   ; Put array into temporary storage
     CASE choice = 1:
       strtcol = 5
       FOR i FROM 1 TO menulen
        tempmenu[i] = submenu1[i]
        tempinfo[i] = subinfo1[i]
        IF maxlen < LEN(tempmenu[i]) THEN
          maxlen = LEN(tempmenu[i])
        ENDIF
       ENDFOR
     CASE choice = 2:
       strtcol = 17
       FOR i FROM 1 TO menulen
        tempmenu[i] = submenu2[i]
        tempinfo[i] = subinfo2[i]
        IF maxlen < LEN(tempmenu[i]) THEN
          maxlen = LEN(tempmenu[i])
        ENDIF
       ENDFOR
       @ 24,0
       ?? subinfo2[1]
     CASE choice = 3:
       strtcol = 29
       FOR i FROM 1 TO menulen
        tempmenu[i] = submenu3[i]
        tempinfo[i] = subinfo3[i]
        IF maxlen < LEN(tempmenu[i]) THEN
          maxlen = LEN(tempmenu[i])
        ENDIF
       ENDFOR
       @ 24,0
       ?? subinfo3[1]
     CASE choice = 4:
       strtcol = 41
       FOR i FROM 1 TO menulen
        tempmenu[i] = submenu4[i]
        IF maxlen < LEN(tempmenu[i]) THEN
          maxlen = LEN(tempmenu[i])
        ENDIF
       ENDFOR
     CASE choice = 5:
       strtcol = 53
       FOR i FROM 1 TO menulen
        tempmenu[i] = submenu5[i]
        IF maxlen < LEN(tempmenu[i]) THEN
          maxlen = LEN(tempmenu[i])
        ENDIF
```

```
   ENDFOR
ENDSWITCH

FOR i FROM 1 TO menulen            ; Draw the Pull Down Menu
  @ i+4,strtcol
  ?? CHR(179)+SPACES(maxlen)+CHR(179)
  @ i+4,strtcol+1
  ?? tempmenu[i]
ENDFOR
@ menulen+5,strtcol
?? CHR(192)+FILL(CHR(196),maxlen)+CHR(217)

PAINTCANVAS ATTRIBUTE 0 menulen+6,strtcol+1,menulen+6,strtcol+maxlen+1
PAINTCANVAS ATTRIBUTE 0 5,strtcol+maxlen+2,menulen+6,strtcol+maxlen+2

@ 24,0                             ; Put the first instruction on the bottom row
?? tempinfo[1]

endcol = strtcol+maxlen
ky = 1
curline = 5


PAINTCANVAS ATTRIBUTE normcol  5, strtcol, menulen+5, endcol+1


PAINTCANVAS ATTRIBUTE highltcol  5, strtcol+1, 5, endcol

WHILE NOT ky=13
  IF CHARWAITING() THEN
    ky = GETCHAR()
    IF (ky = -80) AND NOT curline = menulen+4 THEN
      PAINTCANVAS ATTRIBUTE normcol curline, strtcol+1, curline, endcol
      curline = curline + 1
      PAINTCANVAS ATTRIBUTE highltcol curline, strtcol+1, curline, endcol
      @ 24,0
      ?? tempinfo[curline-4]
    ELSE IF (ky = -72) AND NOT curline = 5 THEN
      PAINTCANVAS ATTRIBUTE normcol curline, strtcol+1, curline, endcol
      curline = curline - 1
      PAINTCANVAS ATTRIBUTE highltcol curline, strtcol+1, curline, endcol
      @ 24,0
      ?? tempinfo[curline-4]
    ELSE IF (ky = -75) THEN
      choice = -75
      QUITLOOP
    ELSE IF (ky = -77) THEN
      choice = -77
      QUITLOOP
    ELSE IF (ky = 27) THEN
      choice = 99
      QUITLOOP                                    ; Exit this while loop
    ELSE IF (ky = 13) THEN
      choice = choice*100 + curline-4
      QUITLOOP
    ELSE
      BEEP
          ENDIF
         ENDIF
        ENDIF
       ENDIF
      ENDIF
    ENDIF
  ENDIF
ENDWHILE

; Erase the menu
FOR i FROM 1 TO menulen+2
  @ i+4,strtcol
```

```
      ?? SPACES(maxlen+2)
    ENDFOR
    PAINTCANVAS ATTRIBUTE 31 5,strtcol,menulen+6,strtcol+maxlen+3
    PAINTCANVAS ATTRIBUTE 0 5,strtcol,5,strtcol+maxlen+3

    IF (inputFlag OR outputFlag) THEN    ; Replace the canvas
      Replace(inputFlag,outputFlag)
    ENDIF

    CURSOR OFF
    RETURN choice

ENDPROC ;  Pulldown
```

```
; Script:     NEWDATA.SC
; Version:    1.0
; Date:       29 JAN 1992
;

PROC NewData()
  CURSOR NORMAL

  mtbf    = 0             ; Reset all of the parameters
  mttr    = 0
  chan    = 0
  delta   = 0
  pp      = 0
  level   = 0
  sf      = 0
  ldtime  = 0
  ordcost = 0
  cc      = 0
  prtcost = 0

  EntryForm()             ; Draw the form on the canvas
  STYLE ATTRIBUTE 15
  DispMsg(15,"Enter the Mean Time Between Failure for the Part (hours).")
  @ 8,20
  ACCEPT "N" REQUIRED MIN 1 MAX 999999 TO mtbf
  IF NOT retval THEN
    PBlue()
    RETURN 101
  ENDIF
  PAINTCANVAS ATTRIBUTE 31 8,20,8,43
  DispMsg(15,"Enter the Mean Time To Repair for the repair facility (hours).")
  @ 9,20
  ACCEPT "N" REQUIRED MIN 1 MAX 999999 TO mttr
  IF NOT retval THEN
    PBlue()
    RETURN 101
  ENDIF
  PAINTCANVAS ATTRIBUTE 31 9,20,9,43
  DispMsg(15,"Enter the number of repair channels for the repair facility.")
  @ 10,20
  ACCEPT "N" REQUIRED MIN 1 MAX 9999 TO chan
  IF NOT retval THEN
    PBlue()
    RETURN 101
  ENDIF
  PAINTCANVAS ATTRIBUTE 31 10,20,10,43
  DispMsg(15,"Enter the average proportion of discards.")
  @ 11,20
  ACCEPT "N" REQUIRED MIN 0.0 MAX 1.0 PICTURE ".##" TO delta
  IF NOT retval THEN
    PBlue()
    RETURN 101
  ENDIF
  PAINTCANVAS ATTRIBUTE 31 11,20,11,43
  DispMsg(15,"Enter the number of parts in the operational population.")
  @ 12,20
  ACCEPT "N" REQUIRED MIN 1 MAX 9999 TO pp
  IF NOT retval THEN
    PBlue()
    RETURN 101
  ENDIF
  PAINTCANVAS ATTRIBUTE 31 12,20,12,43
  STYLE ATTRIBUTE 112
  DispMsg(112,"Enter the steady-state level of inventory.")
  @ 14,20
  ACCEPT "N" MIN 0 MAX 9999 TO level
  IF (level < 1) THEN
```

```
    level = 0
  ENDIF
  IF NOT retval THEN
    PBlue()
    RETURN 101
  ENDIF
  PAINTCANVAS ATTRIBUTE 31 14,20,14,43
  DispMsg(112,"Enter the probability that a stockout will not occur.")
  @ 15,20
  ACCEPT "N" MIN 0.0 MAX 1.0 PICTURE ".##" TO sf
  IF (sf < 0.01) THEN
    sf = 0.0
  ENDIF
  IF NOT retval THEN
    PBlue()
    RETURN 101
  ENDIF
  PAINTCANVAS ATTRIBUTE 31 15,20,15,43
  DispMsg(112,"Enter the Lead-Time (hours).")
  @ 17,20
  ACCEPT "N" MIN 0 MAX 999999 TO ldtime
  IF NOT retval THEN
    PBlue()
    RETURN 101
  ENDIF
  PAINTCANVAS ATTRIBUTE 31 17,20,17,43
  DispMsg(112,"Enter the order cost for each order. ($)")
  @ 18,20
  ACCEPT "$" MIN 0 MAX 99999 TO ordcost
  IF NOT retval THEN
    PBlue()
    RETURN 101
  ENDIF
  PAINTCANVAS ATTRIBUTE 31 18,20,18,43
  DispMsg(112,"Enter carrying as a fraction of part cost on a monthly basis.")
  @ 19,20
  ACCEPT "N" MIN 0.000001 MAX 1.00 TO cc
  IF NOT retval THEN
    PBlue()
    RETURN 101
  ENDIF
  PAINTCANVAS ATTRIBUTE 31 19,20,19,43
  DispMsg(112,"Enter the cost of a part. ($)")
  @ 20,20
  ACCEPT "$" MIN 0.01 MAX 99999 TO prtcost
  IF NOT retval THEN
    PBlue()
    RETURN 101
  ENDIF
  CURSOR OFF
  PBlue()
  RETURN 101
ENDPROC ; NewData

PROC EntryForm()

  inputflag = TRUE        ; Set this flag so other routines know what is
                          ; on the canvas
  CANVAS OFF
  STYLE ATTRIBUTE 31

  @ 8,0                   ; Clear the canvas
  CLEAR EOS
  PAINTCANVAS ATTRIBUTE 31 8,0,24,79

  @ 8,0
  TEXT
              MTBF:
              MTTR:
```

```
   Repair Channels:
Discard Proportion:
        Population:

   Inventory Level:
     Safety Factor:

         Lead Time:
     Ordering Cost:
     Carrying Cost:
         Part Cost:
  ENDTEXT


  CANVAS ON

ENDPROC ; entryForm

PROC DispMsg(styl,msg)
  @ 24,0
  CLEAR EOL
  PAINTCANVAS ATTRIBUTE 48 24,0,24,79

  STYLE ATTRIBUTE 48
  @ 24,0
  ?? msg
  STYLE ATTRIBUTE styl

ENDPROC

PROC PBlue()
  PAINTCANVAS ATTRIBUTE 31 8,20,20,50
ENDPROC
```

```
; Script:      RETRIEVE.SC
; Version:     1.0
; Date:        29 JAN 1992
;

PROC Retrieve()

  PRIVATE i,sizetbl,ky,curloc
  ARRAY tbl[100]
  CANVAS OFF
  STYLE ATTRIBUTE 48
  @ 24,0
  CLEAR EOL
  @ 24,0
  ?? "Enter the filename or <CR> to see list of files."
  @ 7,29
  ?? SPACES(22)
  @ 8,29
  ?? SPACES(23)
  @ 9,29
  ?? SPACES(23)
  @ 10,29
  ?? SPACES(23)
  @ 11,30
  ?? SPACES(22)
  PAINTCANVAS ATTRIBUTE 48 24,0,24,79
  PAINTCANVAS ATTRIBUTE 48 7,29,10,50
  PAINTCANVAS ATTRIBUTE 0  11,30,11,51
  PAINTCANVAS ATTRIBUTE 0  8,51,11,51
  STYLE ATTRIBUTE 48
  @ 8,32
  ?? "Enter File Name: "
  STYLE ATTRIBUTE 15
  CANVAS ON

  @ 9,36
  ACCEPT "A8" TO tablename

  CANVAS OFF
  IF (SUBSTR(tablename,1,1) = "") THEN
    IF (NOT ISFILE("TABLES.DB")) THEN
      @ 24,51
      ?? "Cannot locate list of tables."
      PAINTCANVAS ATTRIBUTE 79 24,51,24,79
      SLEEP 2500
      RETURN
    ENDIF
    SORT "TABLES" ON "Tablename"
    VIEW "TABLES"
    i = 1
    MOVETO RECORD 1
    WHILE NOT EOT()                      ; Get all of the table names
      MOVETO [tablename]
      tbl[i] = []
      SKIP
      i = i + 1
    ENDWHILE
    sizetbl = i - 1
    IF (sizetbl = 0) THEN
      @ 24,58
      ?? "No files to retrieve."
      PAINTCANVAS ATTRIBUTE 79 24,58,24,79
      CLEARIMAGE
      CANVAS ON
      SLEEP 3000
      CANVAS OFF
      @ 8,32                                ; Clear the writing
```

```
  ?? "                    "
  @ 9,32
  ?? "                    "
  PAINTCANVAS ATTRIBUTE 31 7,29,11,51
  Replace(inputFlag,outputFlag)
  RETURN
ENDIF
CLEARALL
@ 24,0
CLEAR EOL
@ 24,0
?? "Use "+CHR(24)+" and "+CHR(25)+" to scroll through list and <CR> to make selection."
PAINTCANVAS ATTRIBUTE 48 24,0,24,79

@ 8,29
?? FILL(" ",23)
@ 9,29
?? FILL(" ",23)
@ 17,40
?? " "
PAINTCANVAS ATTRIBUTE 48 7,29,16,50
PAINTCANVAS ATTRIBUTE 0  17,30,17,51
PAINTCANVAS ATTRIBUTE 0  8,51,17,51
STYLE ATTRIBUTE 63

NewPage(1)

PAINTCANVAS ATTRIBUTE 15 7,30,7,37
curloc = 1
cnt    = 1
ky     = 0
WHILE (NOT (ky = 13))
  IF CHARWAITING() THEN
    ky = GETCHAR()
    IF (ky = -80 AND curloc+6 < 16 AND cnt < sizetbl) THEN
      PAINTCANVAS ATTRIBUTE 63 6+curloc,30,6+curloc,37
      curloc=curloc+1
      cnt = cnt+1
      PAINTCANVAS ATTRIBUTE 15 6+curloc,30,6+curloc,37
    ELSE IF (ky = -80 AND curloc+6 = 16 AND cnt < sizetbl) THEN
        NewPage(cnt-10)
        PAINTCANVAS ATTRIBUTE 15 6+curloc,30,6+curloc,37
        cnt = cnt+1
    ELSE IF (ky = -80) THEN
        BEEP
    ELSE IF (ky = -72 AND curloc+6 > 7) THEN
        PAINTCANVAS ATTRIBUTE 63 6+curloc,30,6+curloc,37
        curloc=curloc-1
        cnt = cnt-1
        PAINTCANVAS ATTRIBUTE 15 6+curloc,30,6+curloc,37
    ELSE IF (ky = -72 AND curloc+6 = 7 AND cnt > 1) THEN
        NewPage(cnt-1)
        PAINTCANVAS ATTRIBUTE 15 6+curloc,30,6+curloc,37
        cnt = cnt-1
    ELSE IF (ky = -72) THEN
        BEEP
    ELSE IF (ky = 27) THEN
        ClearBx()
        Replace(inputFlag,outputFlag)
        RETURN
    ELSE IF (ky = 13) THEN
        tablename = tbl[cnt]
        QUITLOOP
            ENDIF
           ENDIF
          ENDIF
         ENDIF
        ENDIF
       ENDIF
```

```
        ENDIF
       ENDIF
      ENDIF
    ENDWHILE
ELSE
  IF (NOT ISFILE(tablename+".DB")) THEN
    @ 24,59
    ?? " Cannot locate file."
    PAINTCANVAS ATTRIBUTE 79 24,59,24,79
    CLEARIMAGE
    CANVAS ON
    SLEEP 3000
    CANVAS OFF
    @ 8,32                                  ; Clear the writing
    ?? "                    "
    @ 9,32
    ?? "                    "
    PAINTCANVAS ATTRIBUTE 31 7,29,11,51
    Replace(inputFlag,outputFlag)
    RETURN
  ENDIF
ENDIF
                                            ; Tablename is VALID

EDIT tablename                              ; Get the values
MOVETO RECORD 1
MOVETO [mtbf]
mtbf=[]
MOVETO [mttr]
mttr=[]
MOVETO [chan]
chan=[]
MOVETO [delta]
delta=[]
MOVETO [pp]
pp=[]
MOVETO [level]
level=[]
MOVETO [sf]
sf=[]
MOVETO [ldtime]
ldtime=[]
MOVETO [ordcost]
ordcost=[]
MOVETO [cc]
cc=[]
MOVETO [prtcost]
prtcost=[]
DO_IT!                                  ; Go back to main mode

CLEARIMAGE                              ; Clear the image

ClearBx()                               ; Clear the scroll box
EntryForm()                             ; Put the form on the screen

STYLE ATTRIBUTE 31
  @ 8,20
IF mtbf = 0 THEN
  ?? " "
ELSE
  ?? mtbf
ENDIF
  @ 9,20
IF mttr = 0 THEN
  ?? " "
ELSE
  ?? mttr
ENDIF
  @ 10,20
```

```
  IF chan = 0 THEN
    ?? " "
  ELSE
    ?? chan
  ENDIF
    @ 11,20
  IF delta = 0 THEN
    ?? " "
  ELSE
    ?? delta
  ENDIF
    @ 12,20
  IF pp = 0 THEN
    ?? " "
  ELSE
    ?? pp
  ENDIF
    @ 14,20
  IF level = 0 THEN
    ?? " "
  ELSE
    ?? level
  ENDIF
    @ 15,20
  IF sf = 0 THEN
    ?? " "
  ELSE
    ?? sf
  ENDIF
    @ 17,20
  IF ldtime = 0 THEN
    ?? " "
  ELSE
    ?? ldtime
  ENDIF
    @ 18,20
  IF ordcost = 0 THEN
    ?? " "
  ELSE
    ?? ordcost
  ENDIF
    @ 19,20
  IF cc = 0 THEN
    ?? " "
  ELSE
    ?? cc
  ENDIF
    @ 20,20
  IF prtcost = 0 THEN
    ?? " "
  ELSE
    ?? prtcost
  ENDIF

  CANVAS ON
  outputFlag = FALSE
  RETURN 102
ENDPROC ; Retrieve

PROC NewPage(cntz)
    lnn = 7
    CANVAS OFF
    WHILE (cntz < sizetbl+1 AND lnn < 17)
      @ lnn,30
      ?? FILL(" ",8)
      @ lnn,30
      ?? tbl[cntz]
      PAINTCANVAS ATTRIBUTE 63 lnn,30,lnn,37
      cntz = cntz + 1
```

```
      lnn = lnn + 1
    ENDWHILE
    WHILE (lnn < 17)
      @ lnn,40
      ?? " "
      lnn = lnn+1
    ENDWHILE

    CANVAS ON
ENDPROC

PROC ClearBx()
    lnn = 7
    CANVAS OFF
    WHILE (lnn < 18)
      @ lnn,29
      ?? FILL(" ",22)
      lnn = lnn + 1
    ENDWHILE
    PAINTCANVAS ATTRIBUTE 31 7,29,17,51
    CANVAS ON
ENDPROC
```

```
; Script:      SAVEDATA.SC
; Version:     1.0
; Date:        29 JAN 1992
;

PROC SaveData()

  PRIVATE firstime
  CANVAS OFF
  PAINTCANVAS ATTRIBUTE 48 7,29,10,50
  PAINTCANVAS ATTRIBUTE 0  11,30,11,51
  PAINTCANVAS ATTRIBUTE 0  8,51,11,51
  STYLE ATTRIBUTE 48
  @ 10,40
  ?? " "
  @ 8,32
  ?? "Enter File Name:    "
  STYLE ATTRIBUTE 15
  CANVAS ON
  firstime = TRUE
  WHILE (SUBSTR(tablename,1,1) < "A" OR firstime)
    firstime = FALSE
    @ 9,36
    ACCEPT "A8" REQUIRED DEFAULT tablename TO tablename
    tablename = UPPER(tablename)
    CANVAS OFF
    @ 24,47
    CLEAR EOL
    PAINTCANVAS ATTRIBUTE 48 24,47,24,79
    CANVAS ON
    IF (SUBSTR(tablename,1,1) < "A") THEN
      CANVAS OFF
      @ 24,47
      ?? " Filename must start with letter."
      PAINTCANVAS ATTRIBUTE 79 24,47,24,79
      CANVAS ON
    ENDIF
  ENDWHILE

  CANVAS OFF



  QUERY                                  ; Query the TABLES table for the
                                         ;  tablename
    TABLES |  Tablename         |
           |  Check ~tablename  |
           |                    |
           |                    |

  ENDQUERY
  DO_IT!
  IF (ISEMPTY("ANSWER")) THEN
    EDIT "TABLES"                        ; Insert the tablename into TABLES
    END
    INS
    MOVETO [tablename]
    []=tablename
    DO_IT!
  ENDIF
  CLEARALL

  CREATE tablename                       ; Create the table
    "mtbf"            : "N",
    "mttr"            : "N",
    "chan"            : "N",
    "delta"           : "N",
```

```
    "pp"            : "N",
    "level"         : "N",
    "sf"            : "N",
    "ldtime"        : "N",
    "ordcost"       : "$",
    "cc"            : "N",
    "prtcost"       : "$"

  EDIT tablename                        ; Put the values in the table
  MOVETO RECORD 1
  MOVETO [mtbf]
  []=mtbf
  MOVETO [mttr]
  []=mttr
  MOVETO [chan]
  []=chan
  MOVETO [delta]
  []=delta
  MOVETO [pp]
  []=pp
  MOVETO [level]
  []=level
  MOVETO [sf]
  []=sf
  MOVETO [ldtime]
  []=ldtime
  MOVETO [ordcost]
  []=ordcost
  MOVETO [cc]
  []=cc
  MOVETO [prtcost]
  []=prtcost
  DO_IT!                                ; Go back to main mode

  CLEARIMAGE                            ; Clear the image

  @ 8,32                                ; Clear the writing
  ?? "                  "
  @ 9,32
  ?? "                  "

  PAINTCANVAS ATTRIBUTE 31 7,29,11,51
  Replace(inputFlag,outputFlag)

  CANVAS ON

  RETURN 103
ENDPROC ; SaveData
```

```
; Script:      EDITDATA.SC
; Version:     1.0
; Date:        29 JAN 1992
;

PROC EditData()
  CURSOR NORMAL
  outputFlag2 = FALSE

  STYLE ATTRIBUTE 15
  DispMsg(15,"Enter the Mean Time Between Failure for the Part (hours).")
  @ 8,20
  ACCEPT "N" REQUIRED DEFAULT mtbf MIN 1 MAX 999999 TO mtbf
  Putln(8)
  IF NOT retval THEN
    PBlue()
    RETURN 101
  ENDIF
  PAINTCANVAS ATTRIBUTE 31 8,20,8,43
  DispMsg(15,"Enter the Mean Time To Repair for the repair facility (hours).")
  @ 9,20
  ACCEPT "N" REQUIRED DEFAULT mttr MIN 1 MAX 999999 TO mttr
  Putln(9)
  IF NOT retval THEN
    PBlue()
    RETURN 101
  ENDIF
  PAINTCANVAS ATTRIBUTE 31 9,20,9,43
  DispMsg(15,"Enter the number of repair channels for the repair facility.")
  @ 10,20
  ACCEPT "N" REQUIRED DEFAULT chan MIN 1 MAX 9999 TO chan
  Putln(10)
  IF NOT retval THEN
    PBlue()
    RETURN 101
  ENDIF
  PAINTCANVAS ATTRIBUTE 31 10,20,10,43
  DispMsg(15,"Enter the average proportion of discards.")
  @ 11,20
  ACCEPT "N" REQUIRED DEFAULT delta MIN 0.0 MAX 1.0 TO delta
  Putln(11)
  IF NOT retval THEN
    PBlue()
    RETURN 101
  ENDIF
  PAINTCANVAS ATTRIBUTE 31 11,20,11,43
  DispMsg(15,"Enter the number of parts in the operational population.")
  @ 12,20
  ACCEPT "N" REQUIRED DEFAULT pp MIN 1 MAX 9999 TO pp
  Putln(12)
  IF NOT retval THEN
    PBlue()
    RETURN 101
  ENDIF
  PAINTCANVAS ATTRIBUTE 31 12,20,12,43
  STYLE ATTRIBUTE 112
  DispMsg(112,"Enter the steady-state level of inventory.")
  @ 14,20
  ACCEPT "N" DEFAULT level MIN 0 MAX 9999 TO level
  Putln(14)
  IF (level < 0) THEN
    level = 0
  ENDIF
  IF NOT retval THEN
    PBlue()
    RETURN 101
  ENDIF
  PAINTCANVAS ATTRIBUTE 31 14,20,14,43
```

```
  DispMsg(112,"Enter the probability that a stockout will not occur.")
  @ 15,20
  ACCEPT "N" DEFAULT sf MIN 0.0 MAX .99 TO sf
  Putln(15)
  IF (sf < 0.01) THEN
    sf = 0.0
  ENDIF
  IF NOT retval THEN
    PBlue()
    RETURN 101
  ENDIF
  PAINTCANVAS ATTRIBUTE 31 15,20,15,43
  DispMsg(112,"Enter the Lead-Time (hours).")
  @ 17,20
  ACCEPT "N"  DEFAULT ldtime MIN 1 MAX 99999 TO ldtime
  Putln(17)
  IF NOT retval THEN
    PBlue()
    RETURN 101
  ENDIF
  PAINTCANVAS ATTRIBUTE 31 17,20,17,43
  DispMsg(112,"Enter the order cost for each order. ($)")
  @ 18,20
  ACCEPT "$" DEFAULT ordcost MIN .01 MAX 99999.00 TO ordcost
  Putln(18)
  IF NOT retval THEN
    PBlue()
    RETURN 101
  ENDIF
  PAINTCANVAS ATTRIBUTE 31 18,20,18,43
  DispMsg(112,"Enter carrying cost as a fraction of part cost on a monthly basis.")
  @ 19,20
  ACCEPT "N" DEFAULT cc MIN 0.000001 MAX 1.00 TO cc
  Putln(19)
  IF NOT retval THEN
    PBlue()
    RETURN 101
  ENDIF
  PAINTCANVAS ATTRIBUTE 31 19,20,19,43
  DispMsg(112,"Enter the cost of a part. ($)")
  @ 20,20
  ACCEPT "$" DEFAULT prtcost MIN 0.01 MAX 99999 TO prtcost
  Putln(20)
  IF NOT retval THEN
    PBlue()
    RETURN 101
  ENDIF
  PAINTCANVAS ATTRIBUTE 31 20,20,20,43
  CURSOR OFF
  PBlue()

  RETURN 104
ENDPROC ; NewData

PROC PutLn(tnum)
  IF (outputFlag) THEN
    @ tnum,40
    ?? CHR(179)
  ENDIF
ENDPROC

; Script:     CPDIST.SC
; Version:    1.0
; Date:       29 JAN 1992
;

PROC CPDist()
  PRIVATE lnn
```

```
outputFlag2 = TRUE

IF NOT inputFlag THEN
  BEEP BEEP
  @ 24,63
  ?? "Enter data first."
  PAINTCANVAS ATTRIBUTE 79 24,63,24,79
  SLEEP 2000
  @ 24,63
  ?? SPACES(17)
  PAINTCANVAS ATTRIBUTE 48 24,63,24,79
  RETURN 201
ENDIF

@ 24,57
?? SPACES(23)
@ 24,57
?? "Computing Distribution."
PAINTCANVAS ATTRIBUTE 79 24,57,24,79

ComputePD()                          ; Compute the distribution

outputFlag=TRUE                      ; Will have output on canvas
outputFlag1=TRUE

CANVAS OFF
@ 14,20
?? level
@ 15,20
?? sf
PAINTCANVAS ATTRIBUTE 20 14,20,15,30
lnn = 8                              ; Display the results
STYLE ATTRIBUTE 31
WHILE (lnn < 20)
  @ lnn,40
  ?? CHR(179)
  lnn = lnn + 1
ENDWHILE
STYLE ATTRIBUTE 48
@ 24,0
?? SPACES(80)
@ 24,0
?? "Use PgUp and PgDn keys to move through distribution and ESC to exit."
STYLE ATTRIBUTE 31
@ 8,45
?? "Number Failures     Probability"
lnn = 10
n   = 0
STYLE ATTRIBUTE 20
@ lnn,50
?? FORMAT("W6,AR",n)+FILL(" ",10)+FORMAT("W8.5,AR",p0)
NextPage()

WHILE 1=1
  IF CHARWAITING() THEN
    ky = GETCHAR()
    IF (ky = -81 AND n < pp+level) THEN
      NextPage()
    ELSE IF (ky = -81 AND n = pp+level) THEN
      BEEP
    ELSE IF (ky = -73 AND n > 9) THEN
      PrevPage()
    ELSE IF (ky = -73 AND n = 9) THEN
      BEEP
    ELSE IF (ky = 27) THEN
      STYLE ATTRIBUTE 31
      @ 8,45
      ?? "Number Failures     Probability"
      PAINTCANVAS ATTRIBUTE 31 10,50,19,74
```

```
        PAINTCANVAS ATTRIBUTE 31 14,20,15,30
       QUITLOOP
      ELSE
        BEEP
           ENDIF
          ENDIF
         ENDIF
        ENDIF
       ENDIF
      ENDIF
  ENDWHILE

  RETURN
ENDPROC ; CPDist


PROC NextPage()
  PRIVATE lnn
  IF (n = 0) THEN
    lnn =10
  ELSE
    lnn = 9
  ENDIF
  CANVAS OFF
  WHILE (lnn < 19)
    lnn = lnn + 1
    n = n + 1
    IF (n > pp+level) THEN
      WHILE (lnn < 20)
        @ lnn,50
        ?? SPACES(26)
        lnn = lnn + 1
      ENDWHILE
      n = n-1
      QUITLOOP
    ENDIF
    @ lnn,50
    ?? SPACES(26)
    @ lnn,50
    ?? FORMAT("W6,AR",n)+FILL(" ",10)+FORMAT("W8.5,AR",p[n])
  ENDWHILE
  CANVAS ON
ENDPROC

PROC PrevPage()
  PRIVATE lnn
  n = (INT(n/10)-1)*10
  IF (n = 0) THEN
    @ 10,50
    ?? FORMAT("W6,AR",0)+FILL(" ",10)+FORMAT("W8.5,AR",p0)
    lnn = 10
    n = 1
  ELSE
    lnn = 9
  ENDIF
  CANVAS OFF
  WHILE (lnn < 19)
    lnn = lnn + 1
    @ lnn,50
    ?? SPACES(26)
    @ lnn,50
    ?? FORMAT("W6,AR",n)+FILL(" ",10)+FORMAT("W8.5,AR",p[n])
    n = n + 1
  ENDWHILE
  n = n - 1
  CANVAS ON
ENDPROC
```

```
PROC ComputePD()
  PRIVATE Dataok,c1,sr,dr,k
  Dataok = FALSE
  ARRAY p[10000]
  k = pp+level

  WHILE (NOT Dataok)
    IF (mtbf = 0) THEN
      Edt(1)
    ELSE IF (mttr = 0) THEN
      Edt(2)
    ELSE IF (chan = 0) THEN
      Edt(3)
    ELSE IF (pp = 0) THEN
      Edt(5)
    ELSE IF (level = 0 AND sf = 0) THEN
      Edt(6)
    ELSE IF (k > 10000) THEN
      Edt(7)
    ELSE
      Dataok = TRUE
         ENDIF
        ENDIF
       ENDIF
      ENDIF
     ENDIF
    ENDIF
  ENDWHILE

  fr   = pp*(1.0/mtbf)
  dr   = delta*pp*(1.0/mtbf)
  sr   = (1.0-delta)*(1.0/mttr)
  tsf  = 0.0

  IF (sf > 0.0) THEN
    level = 0
  ENDIF

  WHILE (tsf <= sf)
    tsf = 0.0
    n = 2
    IF (chan <= level) THEN
      p[1] = (fr/(sr+dr))
      sum = p[1]
      WHILE (n < chan+1)
        p[n] = p[n-1]*(fr/(n*sr+dr))
        sum = sum + p[n]
        n = n + 1
      ENDWHILE
      WHILE (n < level+1)
        p[n] = p[n-1]*(fr/(chan*sr+dr))
        sum = sum + p[n]
        n = n + 1
      ENDWHILE
      WHILE (n < pp+level+1)
        c1 = (pp-(n-level))*(1.0/mtbf)
        p[n] = p[n-1]*(c1/(chan*sr+(delta*c1)))
        sum = sum + p[n]
        n = n + 1
      ENDWHILE
    ELSE
      p[1] = (fr/(sr+dr))
      sum = p[1]
      WHILE (n < level+1)
        p[n] = p[n-1]*(fr/(n*sr+dr))
        sum = sum + p[n]
        n = n + 1
      ENDWHILE
      WHILE (n < chan+1)
```

```
      c1 = (pp-(n-level))*(1.0/mtbf)
      p[n] = p[n-1]*(c1/(n*sr+delta*c1))
      sum = sum + p[n]
      n = n + 1
    ENDWHILE
    WHILE (n < pp+level+1)
      c1 = (pp-(n-level))*(1.0/mtbf)
      p[n] = p[n-1]*(c1/(chan*sr+(delta*c1)))
      sum = sum + p[n]
      n = n + 1
    ENDWHILE
  ENDIF

  p0 = 1.0/(1.0+sum)

  n = 1
  tsf = p0

  WHILE (n < pp+level+1)
    p[n]=p[n]*p0
    IF (n < level+1) THEN
      tsf = tsf + p[n]
    ENDIF
    n = n + 1
  ENDWHILE

  IF (tsf < sf) THEN
    level = level + 1
  ENDIF

ENDWHILE

sf = INT(tsf*100)/100


ENDPROC ; ComputPD


PROC Edt(enum)                          ; Display an error and edit data

  CURSOR NORMAL
  SWITCH
    CASE enum = 1:
      @ 24,52
      ?? FILL(" ",28 )
      @ 24,52
      ?? "MTBF must be greater than 0."
      PAINTCANVAS ATTRIBUTE 79 24,52,24,79
      BEEP BEEP
      STYLE ATTRIBUTE 15
      WHILE (mtbf = 0)
        @ 8,20
        ACCEPT "N" REQUIRED DEFAULT mtbf MIN 1 MAX 99999 TO mtbf
      ENDWHILE
      Putln(8)
      @ 8,20
      PAINTCANVAS ATTRIBUTE 31 8,20,8,43
    CASE enum = 2:
      @ 24,52
      ?? FILL(" ",28 )
      @ 24,52
      ?? "MTTR must be greater than 0."
      PAINTCANVAS ATTRIBUTE 79 24,52,24,79
      BEEP BEEP
      STYLE ATTRIBUTE 15
      WHILE (mttr = 0)
        @ 9,20
        ACCEPT "N" REQUIRED DEFAULT mttr MIN 1 MAX 99999 TO mttr
      ENDWHILE
```

```
  Putln(9)
  @ 9,20
  PAINTCANVAS ATTRIBUTE 31 9,20,9,43
CASE enum = 3:
  @ 24,52
  ?? FILL(" ",28 )
  @ 24,52
  ?? "Chan must be greater than 0."
  PAINTCANVAS ATTRIBUTE 79 24,52,24,79
  BEEP BEEP
  STYLE ATTRIBUTE 15
  WHILE (chan = 0)
    @ 10,20
    ACCEPT "N" REQUIRED DEFAULT chan MIN 1 MAX 99999 TO chan
  ENDWHILE
  Putln(10)
  @ 10,20
  PAINTCANVAS ATTRIBUTE 31 10,20,10,43
CASE enum = 5:
  @ 24,52
  ?? FILL(" ",28 )
  @ 24,52
  ?? " pp must be greater than 0."
  PAINTCANVAS ATTRIBUTE 79 24,52,24,79
  BEEP BEEP
  STYLE ATTRIBUTE 15
  WHILE (pp = 0)
    @ 12,20
    ACCEPT "N" REQUIRED DEFAULT pp MIN 1 MAX 9999 TO pp
  ENDWHILE
  Putln(12)
  @ 12,20
  PAINTCANVAS ATTRIBUTE 31 12,20,12,43
CASE enum = 6:
  @ 24,50
  ?? FILL(" ",30 )
  @ 24,50
  ?? "Both Level and Sf cannot be 0."
  PAINTCANVAS ATTRIBUTE 79 24,50,24,79
  BEEP BEEP
  STYLE ATTRIBUTE 15
  @ 14,20
  ACCEPT "N" DEFAULT level MIN 1 MAX 9999 TO level
  Putln(14)
  IF (level < 1) THEN
    level = 0
  ENDIF
  @ 14,20
  PAINTCANVAS ATTRIBUTE 31 14,20,14,43
  @ 15,20
  ACCEPT "N" DEFAULT sf MIN .01 MAX 1.0 PICTURE ".##" TO sf
  Putln(15)
  IF (sf < 0.01) THEN
    sf = 0.0
  ENDIF
  @ 15,20
  PAINTCANVAS ATTRIBUTE 31 15,20,15,43
CASE enum = 7:
  @ 24,56
  ?? FILL(" ",24 )
  @ 24,56
  ?? "Level+pp exceeds 10000."
  PAINTCANVAS ATTRIBUTE 79 24,56,24,79
  BEEP BEEP
  STYLE ATTRIBUTE 15
  @ 12,20
  ACCEPT "N" REQUIRED DEFAULT pp MIN 1 MAX 9999 TO pp
  Putln(12)
  @ 12,20
```

```
  PAINTCANVAS ATTRIBUTE 31 12,20,12,43
  @ 14,20
  ACCEPT "N" DEFAULT level MIN 1 MAX 9999 TO level
  Putln(14)
  @ 14,20
  PAINTCANVAS ATTRIBUTE 31 14,20,14,43
CASE enum = 8:
  @ 24,47
  ?? FILL(" ",33 )
  @ 24,47
  ?? "Lead-Time must be greater than 0."
  PAINTCANVAS ATTRIBUTE 79 24,47,24,79
  BEEP BEEP
  STYLE ATTRIBUTE 15
  WHILE (ldtime = 0)
    @ 17,20
    ACCEPT "N" REQUIRED DEFAULT ldtime MIN 1 MAX 99999 TO ldtime
  ENDWHILE
  Putln(17)
  @ 17,20
  PAINTCANVAS ATTRIBUTE 31 17,20,17,43
CASE enum = 9:
  @ 24,43
  ?? FILL(" ",37 )
  @ 24,43
  ?? "Ordering cost must be greater than 0."
  PAINTCANVAS ATTRIBUTE 79 24,43,24,79
  BEEP BEEP
  STYLE ATTRIBUTE 15
  WHILE (ordcost = 0)
    @ 18,20
    ACCEPT "$" REQUIRED DEFAULT ordcost MIN .01 MAX 99999 TO ordcost
  ENDWHILE
  Putln(18)
  @ 18,20
  PAINTCANVAS ATTRIBUTE 31 18,20,18,43
CASE enum = 10:
  @ 24,43
  ?? FILL(" ",37 )
  @ 24,43
  ?? "Carrying cost must be greater than 0."
  PAINTCANVAS ATTRIBUTE 79 24,43,24,79
  BEEP BEEP
  STYLE ATTRIBUTE 15
  WHILE (cc = 0)
    @ 19,20
    ACCEPT "N" REQUIRED DEFAULT cc MIN .01 MAX 1.00 TO cc
  ENDWHILE
  Putln(19)
  @ 19,20
  PAINTCANVAS ATTRIBUTE 31 19,20,19,43
CASE enum = 11:
  @ 24,47
  ?? FILL(" ",33 )
  @ 24,47
  ?? "Part cost must be greater than 0."
  PAINTCANVAS ATTRIBUTE 79 24,47,24,79
  BEEP BEEP
  STYLE ATTRIBUTE 15
  WHILE (prtcost = 0)
    @ 20,20
    ACCEPT "$" REQUIRED DEFAULT prtcost MIN .01 MAX 999999 TO prtcost
  ENDWHILE
  Putln(20)
  @ 20,20
  PAINTCANVAS ATTRIBUTE 31 20,20,20,43
ENDSWITCH

@ 24,43
```

```
    ?? FILL(" ",37 )
    PAINTCANVAS ATTRIBUTE 48 24,43,24,79

    CURSOR OFF
ENDPROC   ;Edt


; Script:      CPROCURE.SC
; Version:     1.0
; Date:        29 JAN 1992
;

PROC CProcure()
  PRIVATE x
  CheckDist()
  IF NOT retval THEN
    RETURN
  ENDIF

  @ 24,53
  ?? SPACES(27)
  @ 24,53
  ?? "Computing procurement data."
  PAINTCANVAS ATTRIBUTE 79 24,53,24,79

  ComputePR()                          ; Compute the Procurement rates

  CANVAS OFF
  lnn = 7                              ; Clear a box for the window
  STYLE ATTRIBUTE 31
  WHILE (lnn < 12)
    @ lnn,27
    ?? SPACES(26)
    lnn = lnn + 1
  ENDWHILE

  PAINTCANVAS ATTRIBUTE 48 7,27,10,52  ; Draw the window
  PAINTCANVAS ATTRIBUTE 0 11,28,11,53
  PAINTCANVAS ATTRIBUTE 0 8,53,11,53
  STYLE ATTRIBUTE 48
  @ 7,28
  ?? "Optimal Procurement Data"
  @ 9,30
  ?? "Quantity:"
  @ 10,30
  ?? "   Level:"
  @ 9,40
  ?? FORMAT("W4,AL",OptQuant)
  @ 10,41
  ?? OptLevel
  PAINTCANVAS ATTRIBUTE 52 9,40,10,50
  @ 24,0
  ?? Subinfo2[2]
  PAINTCANVAS ATTRIBUTE 48 24,0,24,79
  CANVAS ON
  STYLE ATTRIBUTE 31
  WHILE 1=1
    IF CHARWAITING() THEN
      x = GETCHAR()
      CANVAS OFF
      lnn = 7
      STYLE ATTRIBUTE 31
      WHILE (lnn < 12)
        @ lnn,27
        ?? SPACES(27)
        lnn = lnn + 1
      ENDWHILE
      Replace(inputFlag,outputFlag)
      QUITLOOP
```

```
    ENDIF
  ENDWHILE
  CANVAS ON

  RETURN
ENDPROC ; CProcure

PROC CheckDist()
  IF NOT outputFlag OR NOT outputFlag2 THEN
    @ 24,47
    ?? "Compute Probability Distribution."
    PAINTCANVAS ATTRIBUTE 79 24,47,24,79
    SLEEP 2500
    @ 24,47
    ?? SPACES(33)
    PAINTCANVAS ATTRIBUTE 31 24,47,24,79
    RETURN FALSE
  ELSE
    RETURN TRUE
  ENDIF
ENDPROC

PROC ComputePR()
  PRIVATE n,tsum,lambda,Dataok
  Dataok = FALSE

  WHILE (NOT Dataok)
    IF (ldtime = 0) THEN
      Edt(8)
    ELSE IF (ordcost = 0) THEN
      Edt(9)
    ELSE IF (cc = 0) THEN
      Edt(10)
    ELSE IF (prtcost = 0) THEN
      Edt(11)
    ELSE
      Dataok = TRUE
       ENDIF
      ENDIF
     ENDIF
    ENDIF
  ENDWHILE

  ; Compute the expected value of the failure rate
  n = 1
  lambda = (1.0/mtbf)
  tsum = pp*lambda*p0

  WHILE (n < level+1)
    tsum = tsum + pp*lambda*p[n]
    n = n+1
  ENDWHILE
  WHILE (n < level+pp+1)
    tsum =tsum + (pp-(n-level))*lambda*p[n]
    n = n+1
  ENDWHILE
  demand = tsum

  DscrdDmd = delta*demand

  OptLevel = ROUND((DscrdDmd*ldtime),0)

  carrycost = (cc*prtcost)/720.00            ; Convert carrying cost to hourly

  OptQuant = ROUND(SQRT((2.0*ordcost*DscrdDmd)/(carrycost)),0)

ENDPROC
```

```
; Script:      CEXPECT.SC
; Version:     1.0
; Date:        29 JAN 1992
;

PROC CExpect()
  PRIVATE x
  CheckDist()
  IF NOT retval THEN
    RETURN
  ENDIF


  @ 24,52
  ?? SPACES(28)
  @ 24,52
  ?? "Computing expected failures."
  PAINTCANVAS ATTRIBUTE 79 24,52,24,79

  ComputeEF()                              ; Compute the expected number failures

  CANVAS OFF
  lnn = 7                                  ; Clear a box for the window
  STYLE ATTRIBUTE 31
  WHILE (lnn < 12)
    @ lnn,27
    ?? SPACES(26)
    lnn = lnn + 1
  ENDWHILE

  PAINTCANVAS ATTRIBUTE 48 7,27,10,52  ; Draw the window
  PAINTCANVAS ATTRIBUTE 0 11,28,11,53
  PAINTCANVAS ATTRIBUTE 0 8,53,11,53
  STYLE ATTRIBUTE 48
  @ 7,28
  ?? "Expected Number Failures"
  @ 9,40
  ?? INT(expf)
  PAINTCANVAS ATTRIBUTE 52 9,40,9,50
  @ 24,0
  ?? Subinfo2[3]
  PAINTCANVAS ATTRIBUTE 48 24,0,24,79
  CANVAS ON
  STYLE ATTRIBUTE 31
  WHILE 1=1
    IF CHARWAITING() THEN
      x = GETCHAR()
      CANVAS OFF
      lnn = 7
      STYLE ATTRIBUTE 31
      WHILE (lnn < 12)
        @ lnn,27
        ?? SPACES(27)
        lnn = lnn + 1
      ENDWHILE
      Replace(inputFlag,outputFlag)
      QUITLOOP
    ENDIF
  ENDWHILE
  CANVAS ON

  RETURN
ENDPROC ; CExpect


PROC ComputeEF()
  PRIVATE n,tsum,Dataok
  Dataok = FALSE
```

```
  ; Compute the expected number of failures
  n = 1
  tsum = 0.0

  WHILE (n < level+1)
    tsum = tsum + n*p[n]
    n = n+1
  ENDWHILE
  WHILE (n < level+pp+1)
    tsum =tsum + n*p[n]
    n = n+1
  ENDWHILE

  expf = ROUND(tsum,0)

ENDPROC



; Script:    GPDIST.SC
; Version:   1.0
; Date:      2 FEB 1992

PROC GPdist()

  CheckDist()
  IF NOT retval THEN
    RETURN
  ENDIF


  IF NOT ISFILE("PDIST.G") THEN
    @ 24,58
    ?? "Can't find graph spec."
    PAINTCANVAS ATTRIBUTE 79 24,58,24,79
    SLEEP 2500
    @ 24,58
    ?? SPACES(22)
    PAINTCANVAS ATTRIBUTE 48 24,58,24,79
    RETURN
  ENDIF

  @ 24,66
  ?? "Please Wait..."
  PAINTCANVAS ATTRIBUTE 79 24,66,24,79

  CANVAS OFF
  n = 1
  CREATE "PDIST"
    "Failure"     :  "N",
    "Probability" :  "N"

  EDIT "PDIST"
  WHILE (n < pp)
    DOWN
    INS
    MOVETO [Failure]
    []=n
    MOVETO [Probability]
    []=p[n]
    n = n + 1
  ENDWHILE

  MOVETO RECORD 1
  MOVETO [Failure]
  []=0
  MOVETO [Probability]
  []=p0
```

```
  DO_IT!
  CLEARIMAGE

  CANVAS ON

  CURSOR OFF

  Menu {Image} {Graph} {Load} {PDIST}
  Menu {View} {PDIST} Right Right Menu {Image} {Graph} {ViewGraph} {Screen}

  STYLE ATTRIBUTE 48                ; Put the info back on the info bar
  @ 24,0
  ?? infomenu[3]

  CANVAS OFF
  CLEARALL

  CANVAS ON

  CURSOR NORMAL

  X = GETCHAR()

  CURSOR OFF
  RETURN
ENDPROC ;GPdist
```

```
; Script:      GSENS.SC
; Version:     1.0
; Date:        3 FEB 1992
;

PROC GSens()
  PRIVATE ky
  CheckDist()
  IF NOT retval THEN
    RETURN
  ENDIF

  Clr()

  PAINTCANVAS ATTRIBUTE 48 7,27,13,53   ; Draw the window
  PAINTCANVAS ATTRIBUTE 0 14,28,14,54
  PAINTCANVAS ATTRIBUTE 0 8,54,14,54
  STYLE ATTRIBUTE 48
  @ 7,28
  ?? "Graph Safety Factor Over:"
  @ 9,34
  ?? "MTBF"
  @ 10,34
  ?? "MTTR"
  @ 11,34
  ?? "Population"
  @ 12,34
  ?? "Discard Prop."

  @ 24,0
  CLEAR EOL
  @ 24,0
  ?? "Use "+CHR(24)+" and "+CHR(25)+" to scroll through list and <CR> to make selection."
  PAINTCANVAS ATTRIBUTE 48 24,0,24,79

  PAINTCANVAS ATTRIBUTE 31 9,34,9,47
  curloc = 1
  ky     = 0
  WHILE (NOT (ky = 13))
    IF CHARWAITING() THEN
      ky = GETCHAR()
        IF (ky = -80 AND curloc < 4) THEN
          PAINTCANVAS ATTRIBUTE 48 8+curloc,34,8+curloc,47
          curloc=curloc+1
          PAINTCANVAS ATTRIBUTE 31 8+curloc,34,8+curloc,47
        ELSE IF (ky = -80 AND curloc = 4) THEN
            PAINTCANVAS ATTRIBUTE 48 8+curloc,34,8+curloc,47
            curloc = 1
            PAINTCANVAS ATTRIBUTE 31 8+curloc,34,8+curloc,47
        ELSE IF (ky = -72 AND curloc > 1) THEN
            PAINTCANVAS ATTRIBUTE 48 8+curloc,34,8+curloc,47
            curloc=curloc-1
            PAINTCANVAS ATTRIBUTE 31 8+curloc,34,8+curloc,47
        ELSE IF (ky = -72 AND curloc = 1) THEN
            PAINTCANVAS ATTRIBUTE 48 8+curloc,34,8+curloc,47
            curloc = 4
            PAINTCANVAS ATTRIBUTE 31 8+curloc,34,8+curloc,47
        ELSE IF (ky = 27) THEN
            Clr()
            Replace(inputFlag,outputFlag)
            RETURN
        ELSE IF (ky = 13) THEN
            sel = curloc
            Clr()
            Replace(inputFlag,outputFlag)
            QUITLOOP
        ELSE
            BEEP
```

```
          ENDIF
        ENDIF
      ENDIF
    ENDIF
  ENDIF
 ENDIF
ENDWHILE

@ 24,66
?? "Please Wait..."
PAINTCANVAS ATTRIBUTE 79 24,66,24,79

SAVEVARS mtbf,mttr,chan,delta,pp,level,sf,p0,p

sf = 0    ; make sure safety factor is set to 0

CREATE "SENS"
  "xaxis"          : "N",
  "SF"             : "N"
EDIT "SENS"

SWITCH
CASE sel = 1:
  MinMTBF = INT(mtbf-0.5*mtbf)
  MaxMTBF = INT(mtbf+0.5*mtbf)
  increment = INT(0.1*(MaxMTBF-MinMTBF))
  FOR mtbf FROM MinMTBF TO MaxMTBF STEP increment
     sf = 0.0
     ComputePD()
     ; insert the mtbf and SF into table for graphing
     MOVETO [xaxis]
     []=INT(mtbf)
     MOVETO [SF]
     []=sf
     DOWN
  ENDFOR
  DO_IT!
  Menu {Image} {Graph} {Load} {MTBFsens}
CASE sel = 2:
  MinMTTR = INT(mttr-0.5*mttr)
  MaxMTTR = INT(mttr+0.5*mttr)
  increment = INT(0.1*(MaxMTTR-MinMTTR))
  FOR mttr FROM MinMTTR TO MaxMTTR STEP increment
     sf = 0.0
     ComputePD()
     ; insert the mttr and SF into table for graphing
     MOVETO [xaxis]
     []=INT(mttr)
     MOVETO [SF]
     []=sf
     DOWN
  ENDFOR
  DO_IT!
  Menu {Image} {Graph} {Load} {MTTRsens}
CASE sel = 3:
  MinPOP = INT(pp-0.5*pp)
  MaxPOP = INT(pp+0.5*pp)
  increment = INT(0.1*(MaxPOP-MinPOP))
  FOR pp FROM MinPOP TO MaxPOP STEP increment
     sf = 0.0
     ComputePD()
     ; insert the pp and SF into table for graphing
     MOVETO [xaxis]
     []=INT(pp)
     MOVETO [SF]
     []=sf
     DOWN
  ENDFOR
```

```
    DO_IT!
    Menu {Image} {Graph} {Load} {POPsens}
  CASE sel = 4:
    MinDELTA = 0.0
    MaxDELTA = 1.0
    increment = .10
    FOR delta FROM MinDELTA TO MaxDELTA STEP increment
       sf = 0.0
       ComputePD()
       ; insert the delta and SF into table for graphing
       MOVETO [xaxis]
       []=delta
       MOVETO [SF]
       []=sf
       DOWN
    ENDFOR
    DO_IT!
    Menu {Image} {Graph} {Load} {DPsens}
  ENDSWITCH

  CLEARIMAGE

  Menu {View} {SENS} Right Right Menu {Image} {Graph} {ViewGraph} {Screen}

  CANVAS OFF
  CLEARALL
  STYLE ATTRIBUTE 48
  @ 24,0
  ?? InfoMenu[3]

  PLAY "Savevars"

  CANVAS ON
  CURSOR NORMAL
  x = GETCHAR()

  CURSOR OFF

  RETURN 302
ENDPROC ; GSens

PROC Clr()
  PRIVATE lnn

  CANVAS OFF
  lnn = 7                          ; Clear a box for the window
  STYLE ATTRIBUTE 31
  WHILE (lnn < 15)
    @ lnn,27
    ?? SPACES(27)
    lnn = lnn + 1
  ENDWHILE
  CANVAS ON
ENDPROC


; Script:      CPRINT.SC
; Version:     1.0
; Date:        5 FEB 1992
;

PROC Cprint()

  IF NOT inputFlag THEN
    BEEP BEEP
    @ 24,63
    ?? "Nothing to print."
    PAINTCANVAS ATTRIBUTE 79 24,63,24,79
    SLEEP 2500
```

```
  @ 24,63
  ?? SPACES(17)
  PAINTCANVAS ATTRIBUTE 48 24,63,24,79
  RETURN 4
ENDIF

@ 24,63
?? "Printing data... "
PAINTCANVAS ATTRIBUTE 79 24,63,24,79

IF NOT PRINTERSTATUS() THEN
  BEEP BEEP
  @ 24,63
  ?? "Printer not ready"
  PAINTCANVAS ATTRIBUTE 79 24,63,24,79
  SLEEP 2500
  @ 24,63
  ?? SPACES(17)
  PAINTCANVAS ATTRIBUTE 48 24,63,24,79
  RETURN 4
ENDIF

CANVAS OFF
OPEN PRINTER

PRINT "\n"
PRINT "                                    CSIS DATA\n"
PRINT "\n"
PRINT "          System Parameters\n"
PRINT "          -----------------\n"
PRINT "\n"
PRINT "               MTBF: "+FORMAT("W7,AL",mtbf)+"\n"
PRINT "               MTTR: "+FORMAT("W7,AL",mttr)+"\n"
PRINT "    Repair Channels: "+FORMAT("W7,AL",chan)+"\n"
PRINT " Discard Proportion: "+FORMAT("W4.2,AL",delta)+"\n"
PRINT "         Population: "+FORMAT("W7,AL",pp)+"\n"
PRINT "\n"
PRINT "    Inventory Level: "+FORMAT("W7,AL",level)+"\n"
PRINT "      Safety Factor: "+FORMAT("W4.2,AL",ROUND(sf,2))+"\n"
PRINT "\n"
PRINT "          Lead Time: "+FORMAT("W7,AL",ldtime)+"\n"
PRINT "      Ordering Cost: "+FORMAT("W9.2,AL",ordcost)+"\n"
PRINT "      Carrying Cost: "+FORMAT("W4.2,AL",cc)+"\n"
PRINT "          Part Cost: "+FORMAT("W9.2,AL",prtcost)+"\n"
PRINT "\n"
IF outputFlag THEN

  PRINT "\n"
  PRINT "              Distribution\n"
  PRINT "              ------------\n"
  PRINT "\n"
  PRINT " Number Failures            Probability\n"
  PRINT " ---------------            -----------\n"
  PRINT "         0                   "+FORMAT("W8.5,AR",p0)+"\n"
  n=1
  WHILE (n < pp+level+1 AND outputFlag)
    PRINT "    "+FORMAT("W6,AR",n)+"                    "+FORMAT("W8.5,AR",p[n])+"\n"
    n=n+1
  ENDWHILE
  PRINTER OFF
ENDIF
PRINT CHR(12)
CANVAS ON
@ 24,63
?? SPACES(17)
PAINTCANVAS ATTRIBUTE 48 24,63,24,79

RETURN 4
```

```
ENDPROC ; Cprint
```

```
; Script:      REPLACE.SC
; Version:     1.0
; Date:        29 JAN 1992
;

PROC Replace(in,out)
  PRIVATE tcnt
  CURSOR OFF
  CANVAS OFF
  IF (in) THEN
    @ 8,0
    TEXT
              MTBF:
              MTTR:
   Repair Channels:
    ENDTEXT
    IF (mtbf = 0) THEN
      @ 8,20
      ?? " "
    ELSE
      @ 8,20
      ?? mtbf
    ENDIF
    IF (mttr = 0) THEN
      @ 9,20
      ?? " "
    ELSE
      @ 9,20
      ?? mttr
    ENDIF
    IF (chan = 0) THEN
      @ 10,20
      ?? " "
    ELSE
      @ 10,20
      ?? chan
    ENDIF
    PAINTCANVAS ATTRIBUTE 31 8,0,11,40
  ENDIF

  IF (out) THEN
    tcnt = 7
    STYLE ATTRIBUTE 31
    WHILE (tcnt < 19)
      tcnt= tcnt + 1
      @ tcnt,40
      ?? CHR(179)
    ENDWHILE
    PAINTCANVAS ATTRIBUTE 31 8,40,19,54
    IF (outputFlag1) THEN
      @ 8,45
      ?? "Number Failures"
    ENDIF
  ENDIF
  STYLE ATTRIBUTE 48
  CANVAS ON
ENDPROC ; Replace
```

VITAE

Darrin J. Lipscomb

After attending a year at Old Dominion University, I transfered to Mary Washington College in Fredericksburg where I majored in mathematics. I graduated with a Bachelor of Science degree and honors in mathematics in 1988. I received the Oscar Schultz Award for outstanding mathematics student and was an invited member of Pi Mu Epsilon and Alpha Phi Sigma academic fraternities.

After attending a semester of graduate study in mathematics at Brown University in Providence, R.I., I returned to Fredericksburg and began work at Logicon Corporation as a Systems Analyst. In the Fall of 1989, I began course work, part time, through the Virginia Cooperative Graduate Engineering Program for the Systems Engineering degree at Virginia Polytechnic Institute. In 1990, I went to work for Atlantic Research Corporation where I was primarily involved with data modeling and analysis and software development.

In the spring of 1992, I will complete the requirements for the Master of Science degree in Systems Engineering from Virginia Polytechnic Institute and State University.