Summer 1983

Analysis of Scheduling for Low Cost Part Task Trainers

William Fellows
University of Central Florida

ANALYSIS OF SCHEDULING
FOR LOW COST PART TASK TRAINERS

BY

WILLIAM FELLOWS
B.S.E.E., State University of New York at Buffalo, 1975

RESEARCH REPORT

Submitted in partial fulfillment of the requirements
for the degree of Master of Science
in the Graduate Studies Program of the College of Engineering
University of Central Florida
Orlando, Florida

Summer Term
1983

ABSTRACT

This study develops a methodology for the analysis of Part Task Trainer (PTT) refresh scheduling used in conjunction with large simulators. A human performance model is defined through the development of descriptive equations and system random variables. PTT scheduling calculations are performed by employing a computer program simulation. The computer algorithm generates a set of random vectors to represent the learning characteristics of a sample group of individual trainees. The relationships between simulator scheduling time, PTT frequency of training, and model variables are demonstrated through a sensitivity analysis. The computer program is designed to be user interactive. This will allow the PTT refresh scheduling program to be used as an analytical tool for the investigator and training planner. A computer summary of the resulting simulator retraining times with PTT refresh is provided to the user.

ACKNOWLEDGEMENTS

I would like to thank the members of my graduate committee, Dr. Harold I. Klee, Dr. Christian S. Bauer, and Dr. Darrell G. Linton for their assistance and dedication in regard to performing this report.

TABLE OF CONTENTS

NOMENCLATURE

β	Frequency Of Part Task Trainer Refreshing
PTT	Part Task Trainer
$R(t)$	Human Performance, Absolute
$r(t)$	Human Performance, Relative
R_N	Relative Human Performance For N Refresh Periods (Model Equation)
R_F	Ultimate Performance In Operational Environment Assuming No Retraining
R_I	Initial Performance In Operational Environment
R_{CR}	Minimum Or Critical Level Of Human Performance
r_{CR}	Critical Relative Performance Value
S.C.	Sensitivity Coefficient
T	PTT Refresh Time Interval
t	Elapsed Time In Operational Environment
$T_{W/O}$	Length Of Time Required Between System Simulator Training Exercises Without PTT Refresh
τ	Exponential Decay Time Constant Or Attenuating Performance Constant
α	Percent Increase In Relative Performance Imparted By PTT Refreshing

μ	Mean
σ	Standard Deviation
R.V.	Random Variable
T_W	Interval Between Simulator Retraining Sessions With PTT Refresh
S.I.	Sample Individual, or Matrix Value Representative of Model Conditions For An Individual Performance Parameters

CHAPTER I

INTRODUCTION

In recent years, the large system simulators used for military training are experiencing heavy loading demands. Training schedules are being stressed to meet the expanding needs for increased operational readiness. Trainer costs for both the development and life cycle operation of the device are increasing. Higher proficiency/skill levels of personnel are being required over greater periods of time. Along with these factors, very sophisticated equipment utilizing the new wave of computers are making their way into the field. The maintenance of high performance levels with the increasing complexity of operational equipment will require implementing refresh training.

In meeting this need for refresh training, a subset of simulation devices called Part Task Trainers (PTT) can be utilized. The proper scheduling of PTT devices will be very important for optimum training and cost effectiveness.

Part Task Trainers are devices that are designed to focus on training personnel in subsystem operations and specific task missions, which can comprise an electrical system, gun sight dynamic operations, or sensor system interpretations. PTTs can also vary in their complexity and size, with some devices nearly realizing the magnitude and scope of training as the large system simulators.

Through the incorporation of new technological advances in the field of microcomputers and computer graphics, the inexpensive Low Cost Part Task Trainer (LCPTT) will be brought into full realization. In Appendix I, such a device is described for a basic passive sonar trainer. Through periodical refresher training by PTTs, the training cycle for the large trainer is extended, reducing costs and schedule impact.

This study will develop a methodology for analysis of Part Task Trainer (PTT) refresh scheduling in conjunction with large simulators. A human performance model will be defined to describe the learning characteristics of trainees. The model equations, random variables, and system parameters are developed for incorporation into a computer algorithm. A computer simulation will be employed to perform the PTT scheduling calculations from the developed algorithm.

The computer program has been designed to be user interactive. This will allow for the analysis over a large spectrum of training conditions through the flexibility in defining system variables and parameters. Essentially, the PTT refresh scheduling program becomes an analytic tool for the investigator or planner.

A sensitivity analysis is performed on the model variables, demonstrating the relationships for calculated scheduling times to the frequency of PTT training. The computer program outputs a summary report on the calculations and values involved in determining the simulator scheduling times with PTT refresh.

CHAPTER II

HUMAN PERFORMANCE MODEL FOR PART TASK TRAINING

An equation for measuring human performance needs to be defined in order to determine the training factors involved in calculation of simulator scheduling times. Considering that there is a decrease in human performance over time, an exponential function can be used for describing this phenomena. The equation describing the attrition of human performance is assumed to be [1]

$$R(t) = R_F + (R_I - R_F)e^{-t/\tau} \tag{1}$$

where

$R(t)$	=	Human Performance
R_I	=	Initial Performance in Operational Environment
R_F	=	Ultimate Performance in Operational Environment Assuming No Retraining
τ	=	Exponential Decay Time Constant

Contained in Equation (1) are the set of random variables R_I, R_F, α, and τ that characterize individual trainees through different performance values. A random number generator is used to produce a set of observations or a random vector from a population of prospective trainees.

This learning retention curve or Human Performance R(t), is shown in Figure 1.

R(t) passes through a designated minimum level of human performance, R_{CR}, which defines the level of proficiency that must be maintained on the operational device, before retraining on the large simulator is once again required. The point where R(t) intersects R_{CR} determines the value of this time interval between initial training and retraining, and is referred to as $T_{W/O}$, or the time interval for retraining without PTT training.

Considering the utilization of PTTs in the overall training cycle, human performance can be periodically increased. This will result in increased time between retraining on the simulator and reduced demand on its scheduled training.

Simplification of the absolute human performance R(t), is achieved through normalizing Equation (1). This results in

$$r(t) = \frac{R(t) - R_F}{R_I - R_F} = e^{-t/\tau} \qquad (2)$$

where r(t) is the relative human performance. As seen in Figure 1, the relative performance r(t) ranges from zero to one.

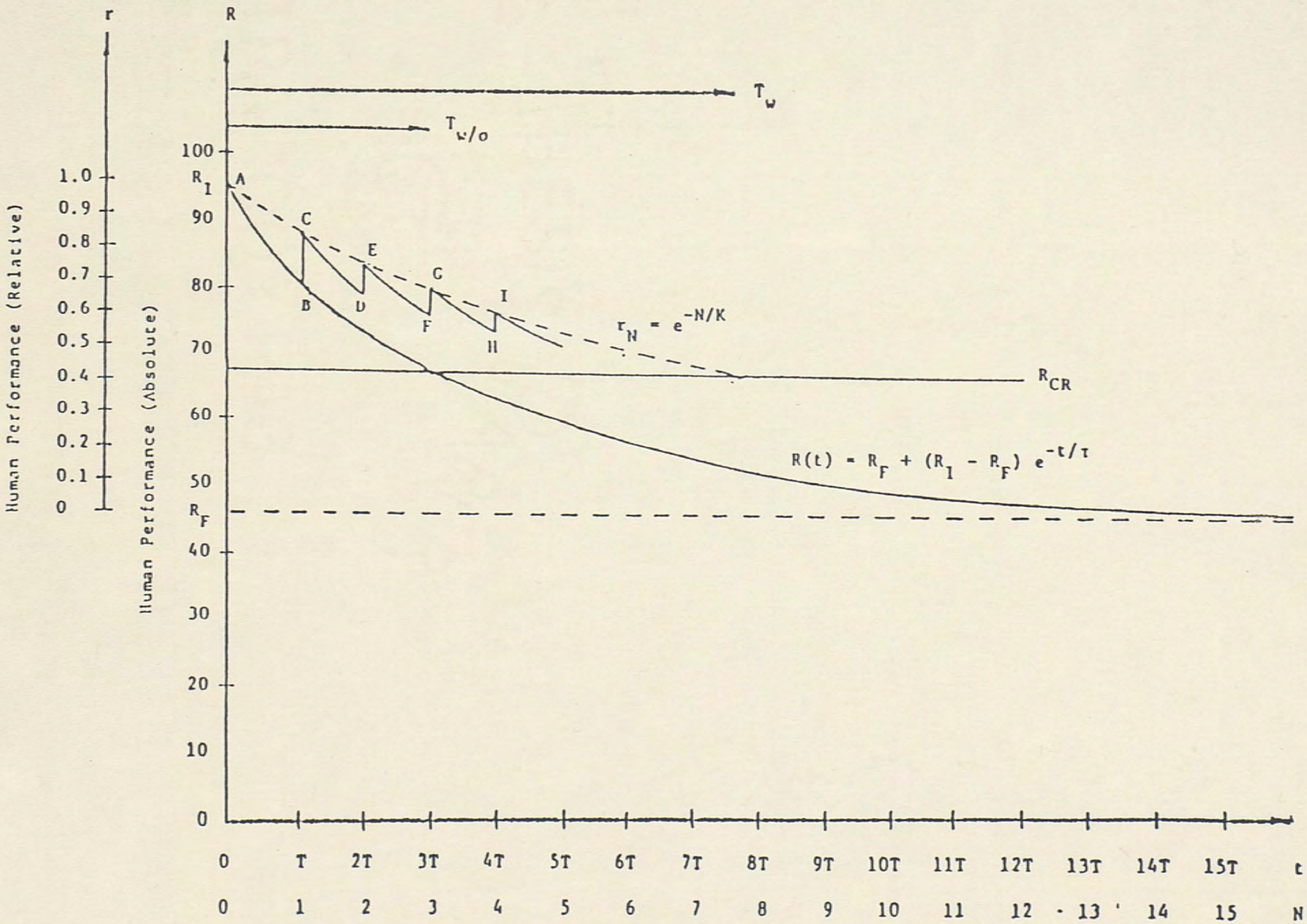


Figure 1. Decline in Human Performance With and Without PTT Refreshing.

The critical relative performance value r_{CR}, is determined by substituting R_{CR} for R(t) in Equation (2), resulting in

$$r_{CR} = \frac{R_{CR} - R_F}{R_I - R_F} \tag{3}$$

Equation (3) implies that the relative critical performance r_{CR} is different for each individual trainee. This is indicative of each trainee having his own unique set of learning parameters. As the values of R_I and R_F change, the relative human performance scale in Figure 1 is effectively shifted, resulting in different values of r_{CR}.

Relative human performance is compared to r_{CR} for determination of simulator retraining time. The interval between initial training on the large simulator (with PTT refresh) and retraining on the simulator is denoted by T_W.

An important parameter in developing the scheduling algorithm is the frequency of PTT refreshing, denoted by β . This parameter is related to the attenuating function time constant τ, and refresh time interval, T, by the expression:

$$T = \beta\tau \qquad 0 \leq \beta \leq 1 \tag{4}$$

$$\text{or} \quad \beta = T/\tau \tag{5}$$

Referring to Figure 1, relative human performance after the first T hours following completion of simulator training is reduced to

$$r_T = 1e^{-T/\tau} \tag{6}$$

Utilizing Equation (5) results in

$$r_T = 1e^{-\beta} \tag{7}$$

Considering an α% increase in relative performance as a consequence of PTT refresh training, the graph switches from point B to C where r=r, given by

$$r_1 = (1 + \alpha)r_T \tag{8}$$

From Equation (7) this becomes

$$r_1 = (1 + \alpha)e^{-\beta} \tag{9}$$

With relative performance exponentially decreasing over the next T interval, relative performance becomes r_{2T}, where

$$r_{2T} = r_1 e^{-T/\tau} \tag{10}$$

Combining Equations (1), (9) and (11) results in

$$r_{2T} = (1 + \alpha)e^{-2\beta} \tag{11}$$

After finishing the second refresh, relative human performance is elevated to $r = r_2$ given by

$$r_2 = (1 + \alpha)r_{2T} \tag{12}$$

$$\text{or} \quad r_2 = (1 + \alpha)^2 e^{-2\beta} \tag{13}$$

The above development assumes an identical α% increase in relative performance over the PTT refresh period. This assumption may not always hold since there are different skill requirements for different devices over the designated periods of time.

Generalizing the results, the relative performance achieved following the Nth refresh, denoted r_N, is given by

$$r_N = (1 + \alpha)^N e^{-\beta N} \tag{14}$$

As seen in Figure 1, a progression of points occur, A, C, E,... corresponding to the discrete times 0, T, 2T, 3T... The sawtooth peaks define an exponential decay shown by the dashed curve. This is demonstrated by expressing Equation (14) as follows:

$$r_N = [(1 + \alpha)e^{-\beta}]^N \tag{15}$$

Introducing the quantity π, where

$$\pi = (1 + \alpha)e^{-\beta} \tag{16}$$

it follows that

$$r_N = \pi^N \qquad N = 1, 2, 3... \tag{17}$$

Equivalently

$$r_N = [e^{(1N\pi)}]^N \qquad N = 1, 2, 3... \tag{18}$$

Defining a discrete time constant K gives

$$r_N = e^{-N/K} \qquad N = 1, 2, 3... \tag{19}$$

where $$K = \frac{1}{1N\pi}$$

Equation (19) is the expression demonstrating the exponential nature of the secondary human performance curve created by PTT refreshing.

The computer algorithm utilizes Equation (14) for performing PTT scheduling determination. An interactive form of the human performance equation is required to describe performance over the PTT refresh period. Equation (14) fulfills the requirement for the computer model equation in describing human performance.

CHAPTER III

METHOD OF ANALYSIS

The objective of this analysis is to determine T_W, the time interval between simulator retraining with PTT refreshing. The use of a computerized algorithm is used to solve this task.

The procedure for determining T_W begins with randomly generating vector sets for the variables R_I, R_F, τ, and α. These random variables are assumed to follow a normal distribution with a known mean and standard deviation. This is performed by first having the computer generate a random number (from a random number generator routine). The interactive computer program uses an inputted mean and standard deviation to describe the normal distribution for the random variable. The computer algorithm uses these values with the randomly generated numbers to form the vector sets. The vector sets represent the performance characteristics for a sample of individual trainees. This will produce for R_I, R_F, α, and τ the following vector sets:

$$
\begin{array}{c}
\text{Sample} \\ 1 \\ \cdot \\ \cdot \\ \cdot \\ n
\end{array}
\quad
\begin{bmatrix} R_{I_1} \\ \cdot \\ \cdot \\ \cdot \\ R_{I_n} \end{bmatrix}
\quad
\begin{bmatrix} R_{F_1} \\ \cdot \\ \cdot \\ \cdot \\ R_{F_n} \end{bmatrix}
\quad
\begin{bmatrix} \alpha_1 \\ \cdot \\ \cdot \\ \cdot \\ \alpha_{1_n} \end{bmatrix}
\quad
\begin{bmatrix} \tau_1 \\ \cdot \\ \cdot \\ \cdot \\ \tau_{1_n} \end{bmatrix}
$$

where n = number of samples. The subprogram description for producing the random number and vector sets is contained in Appendix C.

Each sample row R_I, R_F, α, and τ represents an individual trainee, with his associated learning function or performance characteristic factors.

A chi-square analysis of the resulting sample vector sets is performed for each random variable. The computer algorithm defines seven interval cells where the frequency of occurrence is counted. This would mean that there is six degrees of freedom for the calculated chi-square statistic.

The values of the relative performance r_N in Equation (14) are computed for $.1 \leq \beta \leq 1.0$. Relative performance is calculated for each β value, in increments of 0.1. This provides 10 values of β for each of the twenty generated sample sets.

A comparison of the values of r_N to the value of r_{CR} is then performed. If r_N is less than or equal to r_{CR}, the number of intervals N is saved. The value of the PTT refresh period, T, has been calculated ($T = \beta\tau$) by the program. Taking the value of T, the simulator retraining time, T_W, is calculated by the formula:

$$T_W = T \times N$$

where N = number of periods to reach r_{CR}.

The computer program generates a matrix of T_W values for $.1 \leq \beta \leq 1.0$ and the n samples. Finally, the sample mean of T_W, or $\overline{T}_W$, and T_W sample deviation are calculated for each value of relative to the sum of the sample group.

CHAPTER IV

COMPUTER PROGRAM AND ALGORITHM PTT TRAINING CYCLE INTEGRATION

Description

The PTT Computer Program has the purpose of performing the following functions:

1. The incorporation of an appropriate skill retention model, along with identification of probability distributions for the system random variables through random numerical generation, is performed.

2. An analysis for PTT scheduling is performed for inputted conditions and cases.

3. The program is a simulation that can study the effects of PTT refresh training on system simulator schedule loading.

Program Structure

The computer program has been designed to be an analytic tool, thus it is user interactive. Inputs, parameter, and distribution values are queried to the user for his discretionary inputting, allowing relative flexibility in system analysis. Certain model conditions or functions of realistic operations have been programmed in, but these are either mathematical or real-world restrictions (described in Chapter V).

The program is divided into a main program controlling the sequence of logical operations, and functional subprograms. Remarks and definitions are provided in the listing contained in Appendix F.

The program is in standard BASIC language with some differences in the formatting occurring with the print using statement which in this case are particular to the Wang computer high-level BASIC-2 language.

For general information, the Wang System 2200VP was used, consisting of the following components:

1. Wang CRT display and alphanumerical system keyboard.
2. CPU with user memory and control memory.
3. Floppy disk drive units.
4. Hard disk unit.
5. Hard copy printer unit.

Program Processing

The PTT program consists of a main program which performs the task of integrating the logic required to compute the required outputs in determining T_W, the sample mean, and sensitivity analysis. This process entails calling the appropriate subprogram, initiating their algorithms, utilizing their results, and finally calculating T_W from the system parameters and model equations. The computer program has been structured in a modular form, to provide clear definition of system functions, and a hierarchial sequence of accessing to the required subprogram module.

CHAPTER V

MODEL CONSTRAINTS AND CONDITIONS FOR RANDOM VARIABLES

In attempting to approximate a real world occurrence of a classroom student spectrum, the random variable matrix has been generated to describe the learning factors associated with 20 students.

The constraints on R_I, R_F, α, τ and R_{CR} incorporated in the computer program are:

1. R_I (Initial Performance).

   a. $R_I \leq 100$.

   b. $\mu_{R_I} + 2.5\ \sigma_{R_I} < 100$. The second constraint limits the possibility of obtaining a randomly generated value greater than 100.

   c. Generated values of R_I exceeding 100% are corrected by setting R_I equal to 100%.

2. R_F (Final Performance).

   a. $R_{CR} > \mu_{R_F} - 2.5\ \sigma_{R_F}$.

   The occurrence of R_F being greater than R_{CR} violate the inherent criteria of this model, for the final learning factor can never be greater than the critical learning value. If the above condition does not hold, the program requests that the value of μ_{R_F} and/or σ_{R_F} be changed, preferably μ_{R_F} by the user.

b. The program also checks actual values of R_F to ensure that the $R_F < R_{CR}$. If R_F is greater than R_{CR}, the program interactively redirects the user to input new values for μ_{R_F} and/or σ_{R_F}.

3. α (% Increase Relative Performance).

a. The percent increase in relative human performance must be greater than 0%. This conditional check for α can be expressed as:

If $\mu_\alpha - 2.5\ \sigma_\alpha < 0$, then the program redirects control back to the inputting of μ_α and σ_α for new values to be defined that will meet the model conditions. Individual values of α are checked and discarded if less than zero.

4. τ (Performance Function Time Constraints)

This operational limitation of τ can be developed from the relationship used for determining $T_{W/O}$, because logically, $T_{W/O} > T$ (period for PTT refresh). The formula for $T_{W/O}$, which will be developed in Chapter VII, is

$$T_{W/O} = -\tau\ LN\ \frac{(R_{CR} - R_R)}{(R_I - R_F)}$$

$$\text{If} \quad LN\ \frac{(R_{CR} - R_F)}{(R_I - R_F)} \leq -1 \ ,$$

then $T_{W/O}$ may be equal to or less than τ, which could result in $T_{W/O} \leq T$, which would be contrary to the model. This is shown by

$$T_{W/O} > T = \beta\tau$$

$$T_{W/O} > \beta\tau$$

$$\tau < \frac{T_{W/O}}{\beta} \qquad 0 \leq \beta \leq 1.0$$

Thus, if $T_{W/O}$ is equal to or less than τ, the relationship will not hold for all β . The LN(X) ≥ -1 holds for X $< .368$, where LN(.368) $\simeq -1$. This means that the factor of

$$\frac{(R_{CR} - R_F)}{(R_I - R_F)} < .368$$

must hold in order that $\tau < \frac{T_{W/O}}{\beta}$, and $\frac{T_{W/O}}{\beta} > T$ be true for all values of β .

If this relationship was not upheld, R(t) would effectively reach R_{CR} before the scheduled PTT refresh session, resulting in certain values of β not adhering to $\tau < \frac{T_{W/O}}{\beta}$.

5. $e^{\beta} > (1 + \alpha)$. In the calculation of T_W, the condition $e^{\beta} > (1 + \alpha)$ is tested for the particular value of β and α. If the condition does not hold, then the β value is invalid for that α, thus the T_W value at that β is disregarded. The T_W value is set to 0. Deviation of this conditional restraint is implied from Equation 9 for relative performance $r_1 < 1$.

CHAPTER VI

SENSITIVITY ANALYSIS

Methods For Sensitivity Analysis

Two approaches will be applied in performing a sensitivity analysis of the four random variables, and the important model parameter, R_{CR}.

1. Generation Of Random Operational Base. The first method follows directly in principle the stochastic nature applied to the random variables. It involves choosing one of the random variables, varying its mean while holding the standard deviation constant. The mean and S.D. of the remaining three random variables and R_{CR} will be constant throughout the process. The same principles hold true when R_{CR} is varied. Sample vector sets are randomly generated by initializing the computer program with the appropriately defined variables. This method is performed through the computer program by generating new random vector sets for an operational base condition.

To analyze the effect of R_{CR}, the same μ and σ for defined values of the random variables are maintained, varying the values for R_{CR}.

This process randomly generates new sample sets for each random variable in conjunction with an established mean and standard deviation.

2. Varying One Variable In Relation To A Constant Randomly Generated Base Vector Set. The second method involves invoking Subroutine 1300 for performance of a sensitivity analysis. The subprogram basically sets constant a series of values for a single R.V. or the parameter R_{CR}. In other words, $\sigma = 0$, with μ defined. This process keeps the originally generated values of the Random Variable Matrix, A (I, J). This method does not rigorously follow the dictates of the established stochastic nature of the model variables, but can provide insights into sensitivity variances relative to a controlled base of variables.

Computer Outputs And Sensitivity Coefficient

The computerized algorithm for both methods prints out the computed T_W matrix for each run relative to β $(.1 \leq \beta \leq 1.0)$, along with the sampled mean average and sampled deviation for T_W.

Taking the calculated sample mean, $\bar{T}_W$, and plotting it versus the random variable mean, $\hat{u}_{R.V.}$, will determine the sensitivity coefficient relative to the operational base. This calculation can be accomplished mathematically or graphically from the slope of the tangent to the reference mean $\hat{\mu}$.

The sensitivity coefficient for the model can be expressed as the first order partial derivative

$$\text{Sensitivity Coefficient} = \frac{\partial \bar{T}_W}{\partial \mu_{R.V.}} \tag{20}$$

$$\text{for R.V.} \Rightarrow \begin{cases} \hat{\mu}_{R_I} \\ \hat{\mu}_{R_F} \\ \hat{\mu}_{\alpha} \\ \hat{\mu}_{\tau} \end{cases}$$

This analysis also applied for R_{CR} as a variable factor where μ or R_{CR} is the point around which the sensitivity measurement is made. The sensitivity coefficients are calculated by taking the slope of the line tangent preceding the reference point, and the slope of the tangent after the reference point, summing them and dividing by two. This can be expressed mathematically by Equation (21).

$$S.C. = \frac{\dfrac{Y_1 - Y_2}{X_1 - X_2} + \dfrac{Y_2 - Y_3}{X_2 - X_3}}{2} \tag{21}$$

Thus, the final equational form that is applicable for the PTT model sensitivity coefficient would be

$$S.C. = \frac{\dfrac{\bar{T}_{W1} - \bar{T}_{W2}}{\mu_{R.V.1} - \mu_{R.V.2}} + \dfrac{\bar{T}_{W2} - \bar{T}_{W3}}{\mu_{R.V.2} - \mu_{R.V.3}}}{2} \tag{22}$$

Finally, the S.C. would describe the significance of each variable relative to the operational or base condition:

$$\left[\hat{\mu}_{R_I},\ \hat{\mu}_{R_F},\ \hat{\mu}_{\alpha},\ \hat{\mu}_{\tau},\ R_{CR}\right]$$

CHAPTER VII

AN ECONOMIC ANALYSIS OF PTT REFRESH TRAINING CYCLE

An economic analysis of PTT refresh training can be performed by determining the difference in time between the calculated T_W and interval time for retraining on the large simulator without the implementation of PTT refresh training, or $T_{W/O}$. The economic relationship can be expressed as:

$$\text{PTT Time Saving} = T_W - T_{W/O}$$

or

$$\Delta T = T_W - T_{W/O}. \tag{23}$$

There are many factors involved in attaching a dollar value to the savings incurred by ΔT, let alone the qualitative implications of reducing stress time and training impact on the large simulator.

Significant savings could be achieved in a broad spectrum of areas, such as operational costs, personnel effectiveness, longer proficiency period of performance, logistics, and reduction of training time on the simulator. This economic value relationship, expressed in a dollar value, multiplied times ΔT would express the utility value of PTT, or utility value of PTT = ΔT_W X (Economic Value Relationship).

In determining the value of $T_{W/O}$ that would be used in conjunction with $T_{W/O}$, usage of the human performance relationship is once again employed. Beginning with Equation (1), the absolute human performance, the derivation of $T_{W/O}$ proceeds in the following manner:

1. $R(t) = R_F + (R_I - R_F)e^{-t/\tau}$.

2. Set $R(t) = R_{CR}$, which will represent the solution of R(t) at the intersection point on R_{CR}.

3. $R_{CR} = R_F + (R_I - R_F)e^{-t/\tau}$.

4. Solve for t, the time when human performance has reached a point when retraining on the large simulator is deemed necessary.

5. $$e^{-t/\tau} = \frac{(R_{CR} - R_F)}{(R_I - R_F)} \tag{24}$$

6. $$t = -\tau \, LN \, \frac{(R_{CR} - R_F)}{(R_I - R_F)} \tag{25}$$

Thus

$$T_{W/O} = -\tau \, LN \, \frac{(R_{CR} - R_F)}{(R_I - R_F)} \tag{26}$$

Using this relationship, ΔT in Equation (23) can be calculated, leading to determination of the PTT utility value.

Example Calculations Of ΔT.

For the data set defined as:

$R_I = 91.3$

$R_F = 44$

$R_{CR} = 54$

$\alpha = 10.5$

$\tau = 87.8$

the value of $T_{W/O}$ is calculated as:

$$T_{W/O} = -\tau \text{ LN } \frac{(R_{CR} - R_F)}{(R_I - R_F)}$$

$$= -(87.8)\text{LN } \frac{(54 - 44)}{(91.3 - 44}$$

$$= -(87.8)\text{LN } (.2114)$$

$$T_{W/O} = 135.8$$

Employing the computer algorithm to solve for T_W with PTT refresh over the range of $.1 \leq \beta \leq 1.0$, at the established data set, results in:

β	T_W	$T = T_W - T_{W/O}$
.1	658	572
.2	281	145
.3	210	74
.4	210	74
.5	175	39
.6	210	74
.7	184	48
.8	210	74
.9	158	22
1.0	175	39

The data implies that the best economic value of training and most realistic frequency of PTT refresh would occur at $.2 \leq \beta \leq .4$, with $\beta = .2$ probably representing the best frequency of training, considering real world constraints. T_W does not reach the value of $T_{W/O}$ because $\alpha > 0$, which would imply that some training value would be imparted, during PTT refresh. This is verified by the computed data for the random variable sets. The calculated time period at $\beta = .2$ before simulator retraining has more than doubled. Considerable savings in scheduling and resources could result.

In analyzing the economic factors for PTT refresh training, the planner must consider certain tradeoffs in determining the best value for training. There is an allowable threshold to the frequency of PTT refresh. One cannot train excessive amounts of time on the PTT. Costs alone would be prohibitive, along with scheduling problems and very little productivity resulting. Also, various skill levels are involved in determining the frequency of training. The planner needs to consider the whole situation to arrive at the best training schedule.

CHAPTER VIII

RESULTS

A sensitivity analysis is performed to analyze the relationship between simulator scheduling time (T_W), PTT refreshing frequency (β), and the system variables. Data tables are developed for $\overline{T}_W$ and β for each model variable through a series of computer generated evaluations. The design operational base conditions defined for the model variables are:

R_I Mean = 90%	R_I Sigma = 3
R_F Mean = 45%	R_F Sigma = 3
Alpha Mean = 10%	Alpha Sigma = 3
Tau Mean = 100%	Tau Sigma = 10

$$R_{CR} = 54\%$$

As a part of the analysis, the histograms for the generated R.V.s are plotted, depicting their frequency distributions. The graphs of the computer calculated sample T_W mean ($\overline{T}_W$), will be plotted for the frequency of PTT refresh training. This is performed for each random variable. The graph of $\overline{T}_W$ versus the model variables is also done for specific values of β . The calculated sensitivity coefficients are

calculated from the data tables, determining the most influential variable relative to the operational base condition.

Histogram Plots for Random Variables

Histograms of the model random variables are graphed from computer generated values. In presenting a graphical perspective on the four random variables, the depicted nature of the normal distribution is shown. The data plotted using bar graph or histogram representation visually demonstrates to what accuracy the computer program's random number generator is performing. The histograms are set up into segments or sector cells established by the computerized chi-square analysis in Subroutine 2800. Along with each histogram is provided the numerically calculated chi-square statistic. This statistical value indicates to what degree or confidence level the sample set of generated random variables approach a normal distribution. The histogram plots for several random variables at a defined mean and standard deviation are presented in Figures 2 through 9.

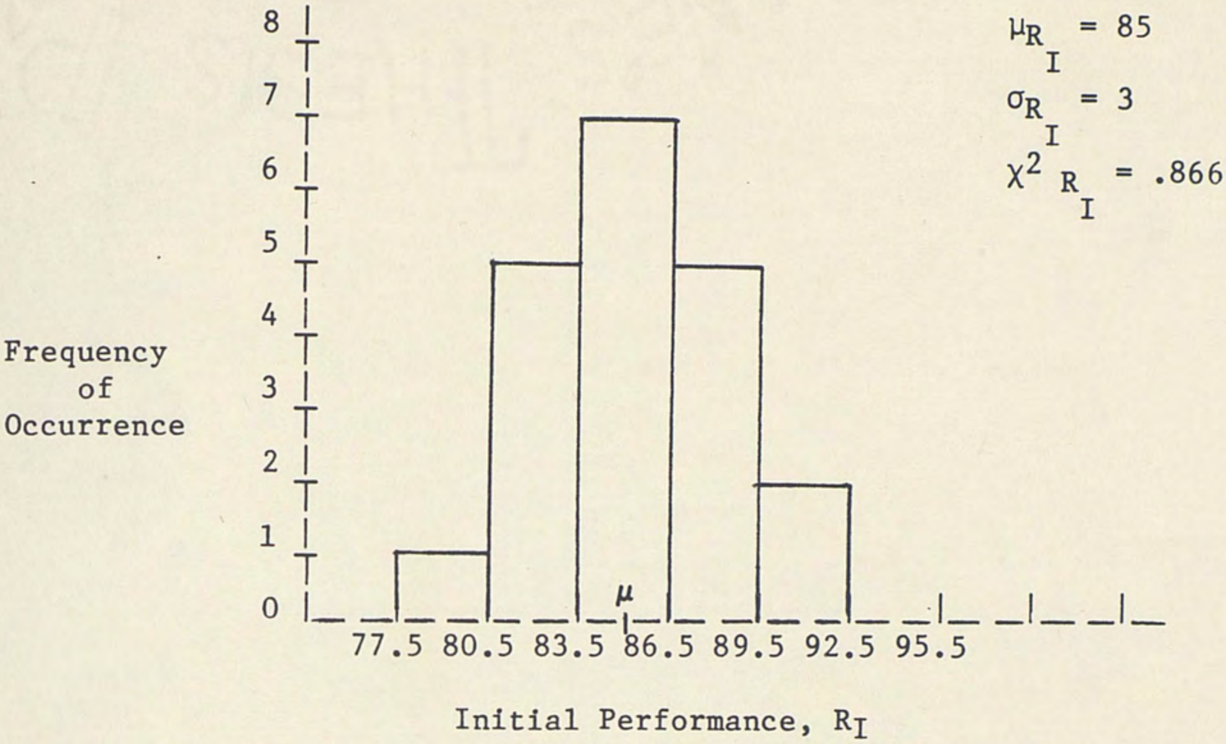


Figure 2. Histogram of Computer Generated Random Variable R_I.

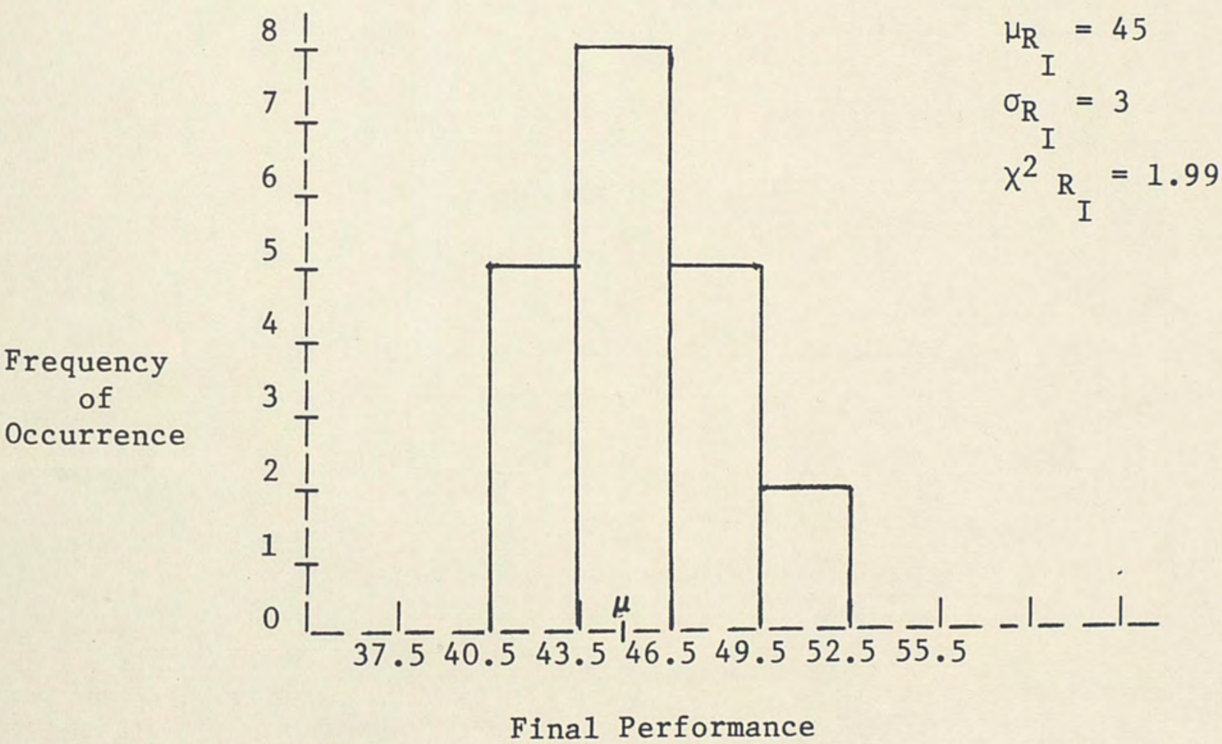


Figure 3. Histogram of Computer Generated Random Variable R_F.

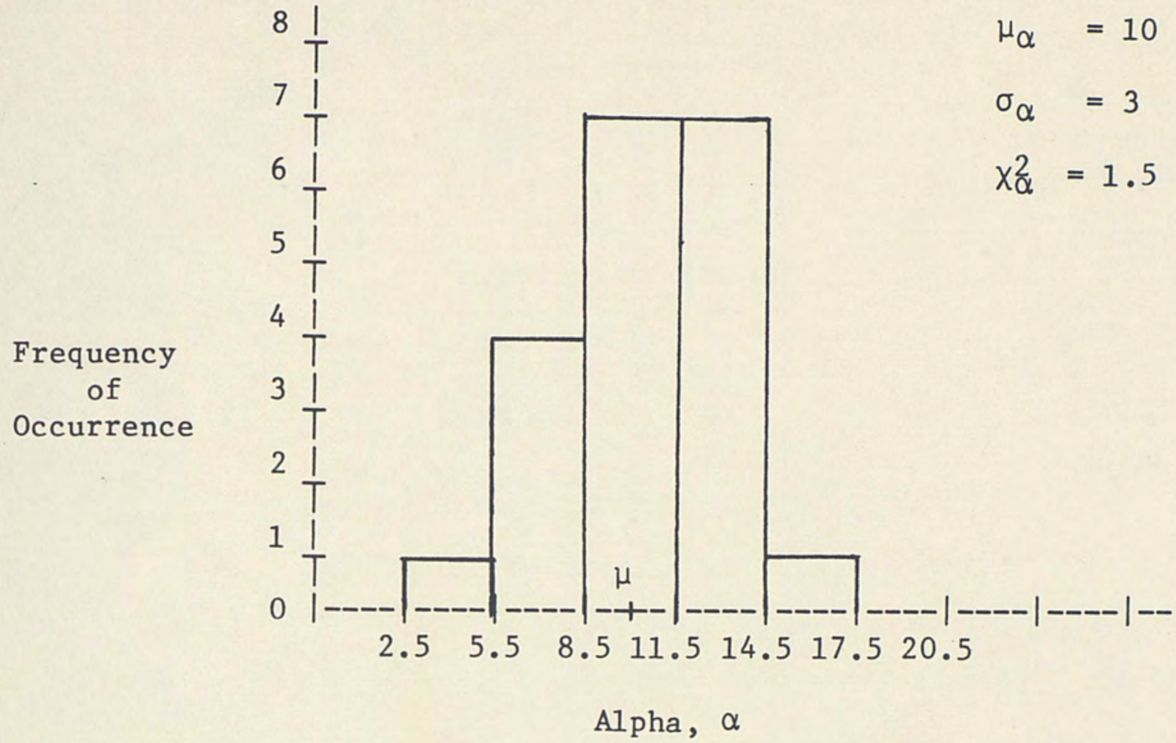


Figure 4. Histogram of Random Variable Alpha.

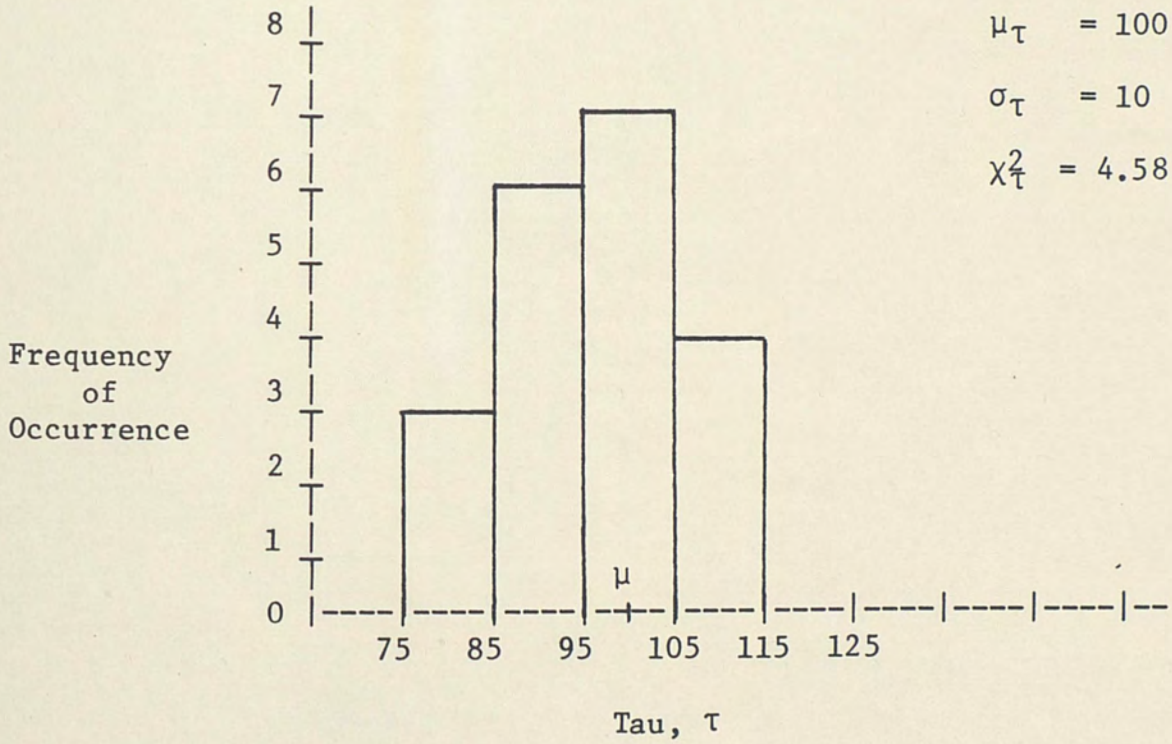
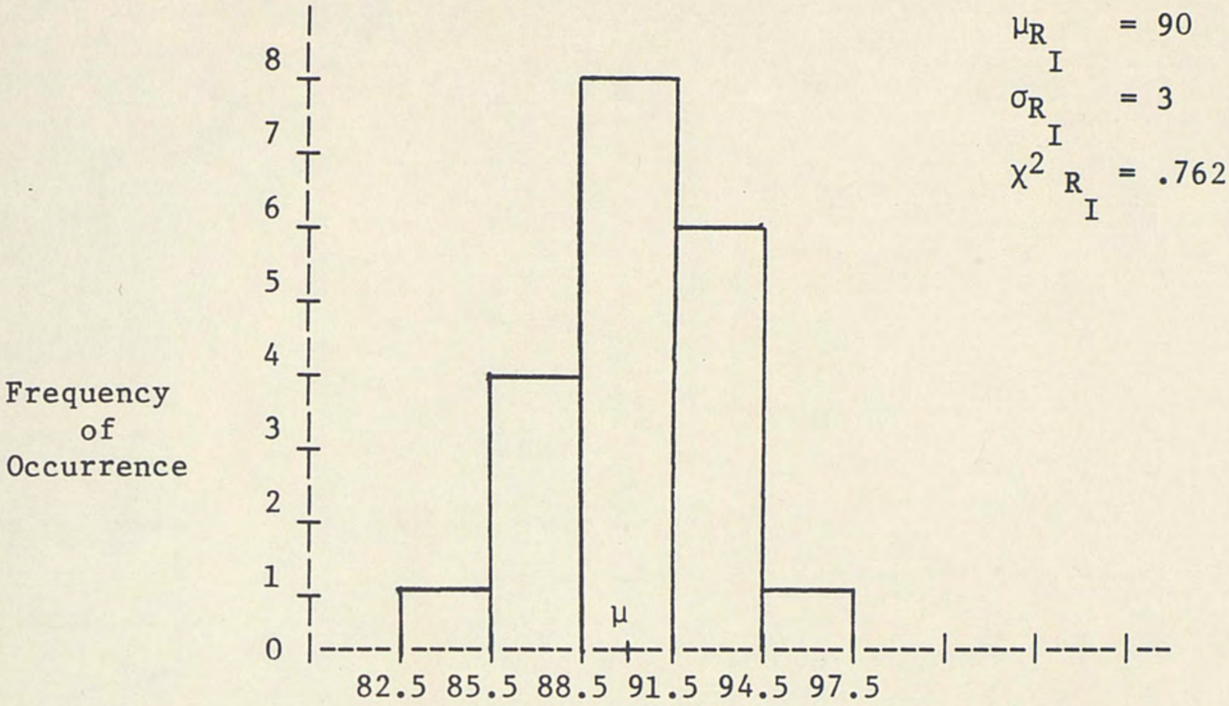
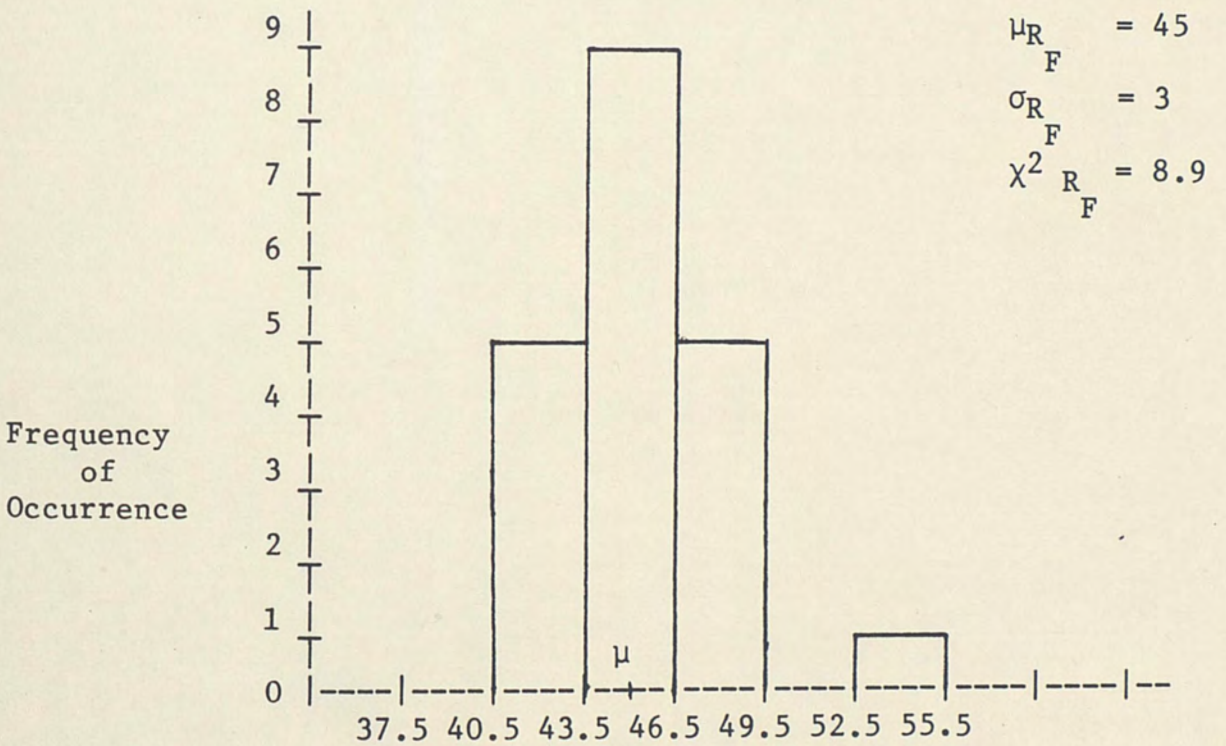


Figure 5. Histogram of Random Variable Tau.



Initial Performance, R_I

Figure 6. Histogram of Random Variable R_I.



Final Performance, R_F

Figure 7. Histogram of Random Variable R_F.

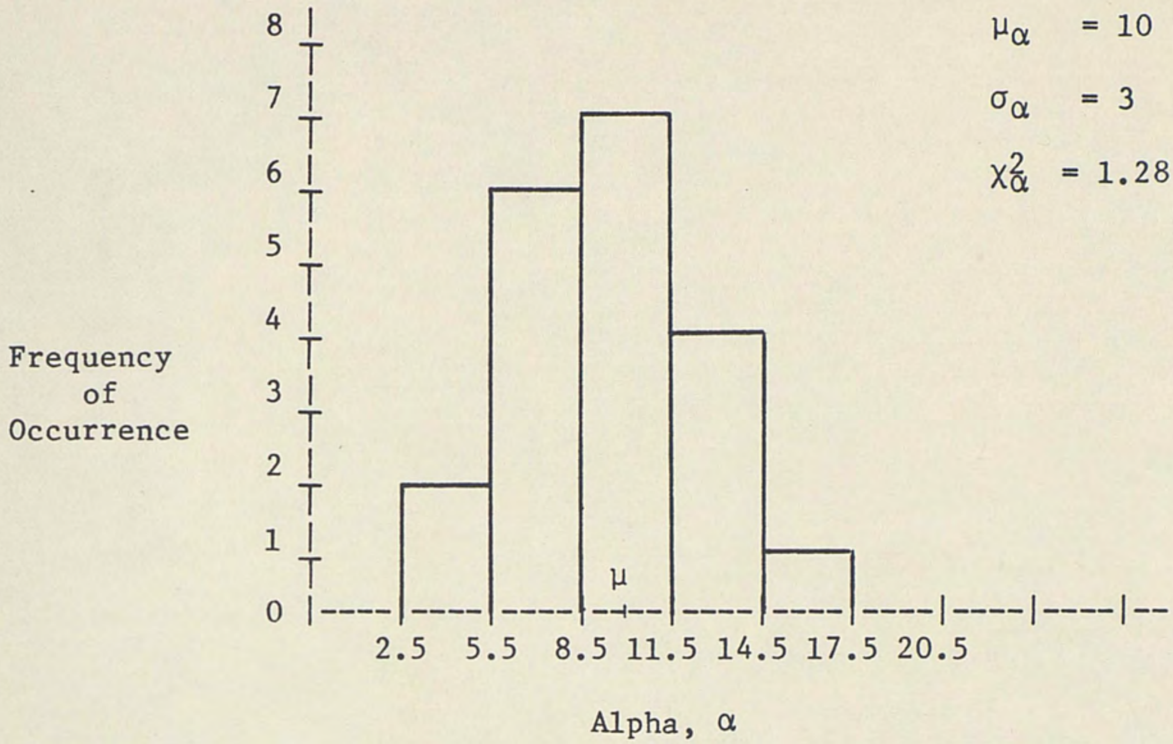


Figure 8. Histogram of Computer Generated Random Variable Alpha.

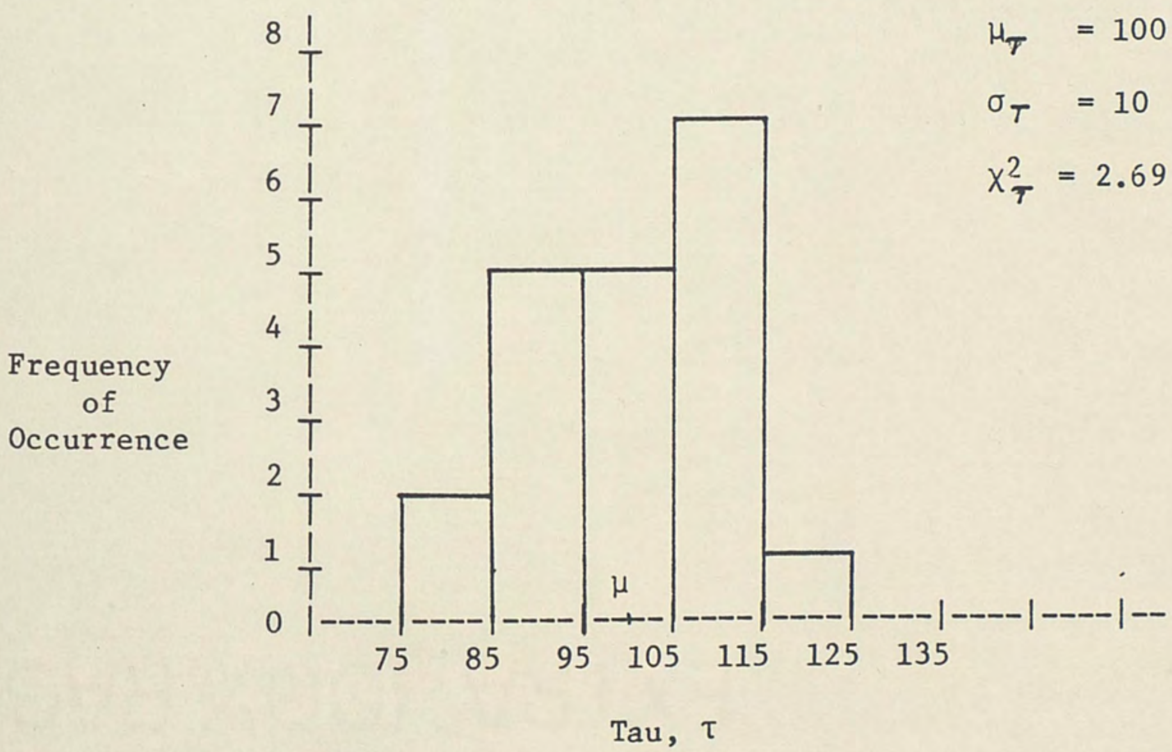


Figure 9. Histogram of Computer Generated Random Variable τ.

Sensitivity Analysis Results

A sensitivity analysis was performed utilizing the defined operational base for the model variables. By employing the interactive computer program, a series of random vector sets were generated, along with their associated scheduling time calculations. This process was performed by varying the mean of one variable, and holding all other values constant. The mean and standard deviation for the remaining variables were thus held constant. R_{CR} was maintained at a constant value, except when it was varied for a sensitivity analysis in relation to the operational base condition. The resulting sample means of T_W, or $\overline{T}_W$ relative to each β are defined for the model variables in Tables 1 through 4. The tables contain the computer generated values of $\overline{T}_W$ used in the analysis and graphs. The sensitivity coefficients are calculated from these tables.

TABLE 1. SAMPLE AVERAGE VALUES OF $\bar{T}_W$ FOR THE RANDOM VARIABLE MEAN μ_{R_I}

$\bar{T}_W$ Values

μ_{R_I} \ β	.1	.2	.3	.4	.5	.6	.7	.8	.9	1.0
80	488	288	214	200	191	199	187	198	192	203
85	482	332	248	224	218	213	215	218	207	211
88	522	311	237	220	210	208	216	217	194	215
90	588	336	265	243	231	230	235	231	229	232
92	559	356	262	237	238	229	217	232	240	221
95	569	312	252	230	230	228	223	226	228	216
98	552	438	315	282	266	261	263	258	272	271

TABLE 2. SAMPLE AVERAGE VALUES OF $\bar{T}_W$ FOR THE RANDOM VARIABLE MEAN μ_{R_F}

$\bar{T}_W$ Values

μ_{R_F} \ β	.1	.2	.3	.4	.5	.6	.7	.8	.9	1.0
35	455	257	180	165	164	161	163	160	179	190
40	492	260	207	196	190	183	195	184	179	199
42	495	296	224	204	204	195	203	197	204	201
45	504	400	256	236	213	222	212	224	203	212
48	628	377	295	271	262	267	245	259	258	245
50	678	507	377	346	330	322	316	314	309	318

TABLE 3. SAMPLE AVERAGE VALUES OF $\bar{T}_W$
FOR THE RANDOM VARIABLE MEAN μ_α

$\bar{T}_W$ Values

μ_α \ β	.1	.2	.3	.4	.5	.6	.7	.8	.9	1.0
5	360	235	215	209	209	214	216	213	221	224
8	422	318	252	238	231	223	237	222	228	235
10	440	358	264	243	222	223	225	233	209	227
12	419	455	298	268	254	238	243	244	231	241
15	419	669	332	273	244	230	241	243	229	219
20	419	838	504	350	312	284	276	261	272	270

TABLE 4. SAMPLE AVERAGE VALUES OF $\bar{T}_W$
FOR THE RANDOM VARIABLE MEAN μ_τ

$\bar{T}_W$ Values

μ_τ \ β	.1	.2	.3	.4	.5	.6	.7	.8	.9	1.0
50	340	179	133	125	121	116	117	123	120	114
80	488	294	209	192	187	182	177	180	182	178
90	431	348	258	232	228	224	222	225	217	223
100	412	340	256	229	223	217	220	229	215	220
110	494	462	312	286	273	263	264	271	264	259
130	784	556	396	351	339	324	333	316	316	321
150	811	475	364	340	326	320	329	328	308	319
400	1770	1214	961	897	871	853	882	880	865	880

Using the values obtained in the data tables, several plots and graphs can be derived. Information pertaining to model trends and characteristics are determined from the graphs. The four graphs contained in Figures 10 through 13 are formulated at the model operational base of $R_I = 90\%$, $R_F = 45\%$, $\alpha = 10\%$, and $\tau = 100$.

The characteristic nature reflected by the operational base point curves were virtually identical. This would be an expected result since each graph represents a mean condition for the operational base. Values of $\overline{T}_W$ were numerically very close for these curves. This demonstrates that the computer model is producing the proper consistency in calculated outputs for the randomly generated samples. Each curve exhibited a rapidly declining exponential feature between $0 \leqslant \beta \leqslant .4$. The curves then inflect into a constancy of values approximately around $T_W = 225$, for $.5 \leqslant \beta \leqslant 1.0$.

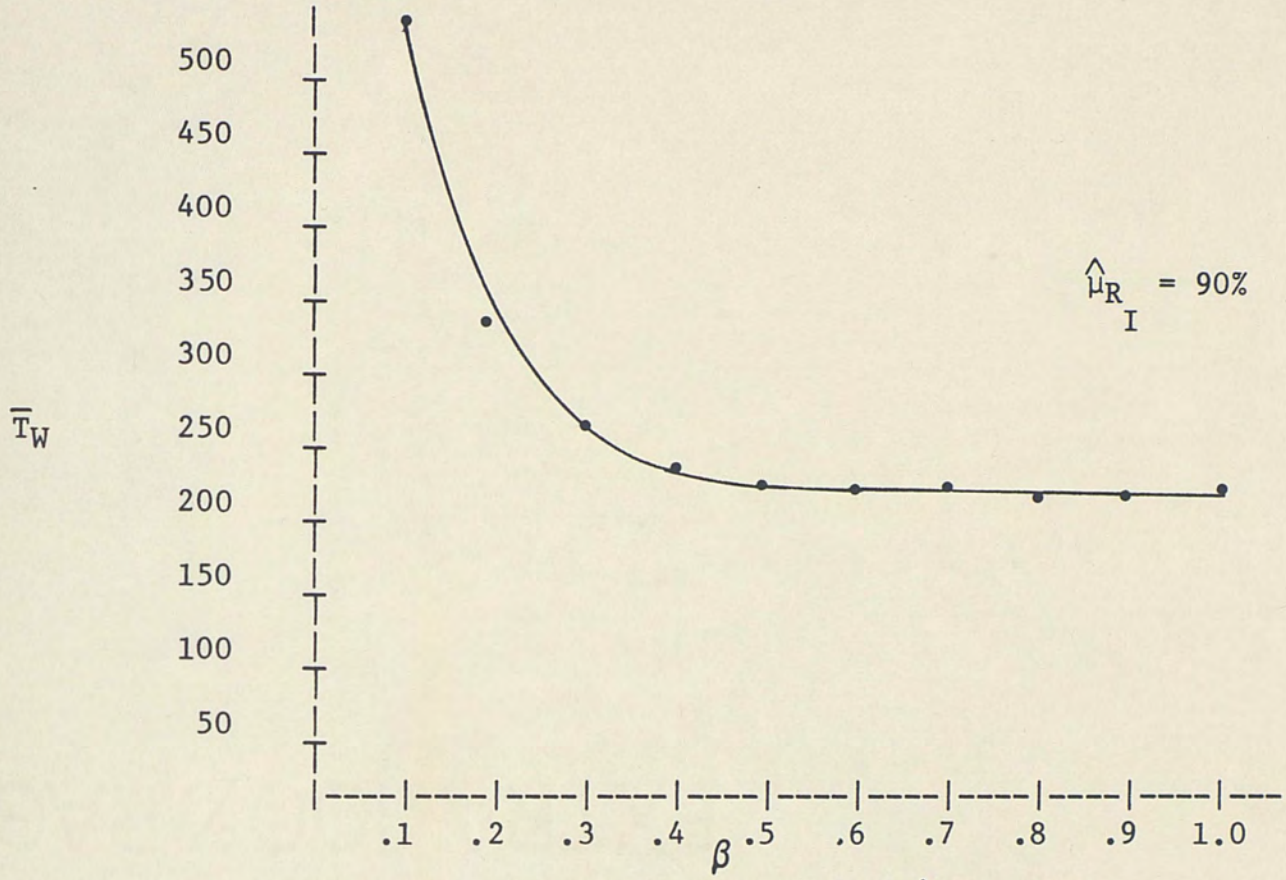


Figure 10. Plot of $\overline{T}_W$ versus β for $\hat{\mu}_{R_I}$ = 90%

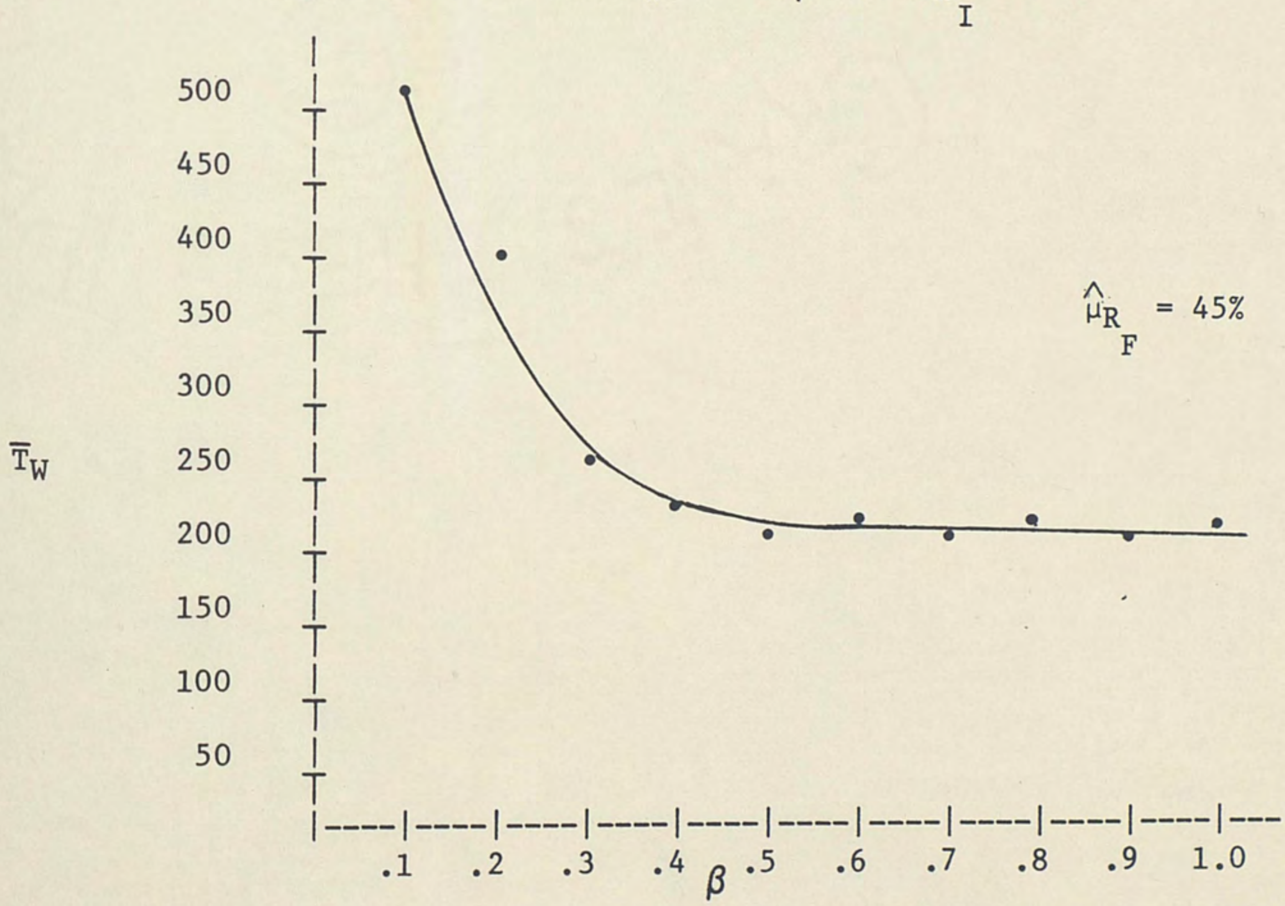


Figure 11. Plot of $\overline{T}_W$ versus β for $\hat{\mu}_{R_F}$ = 45%

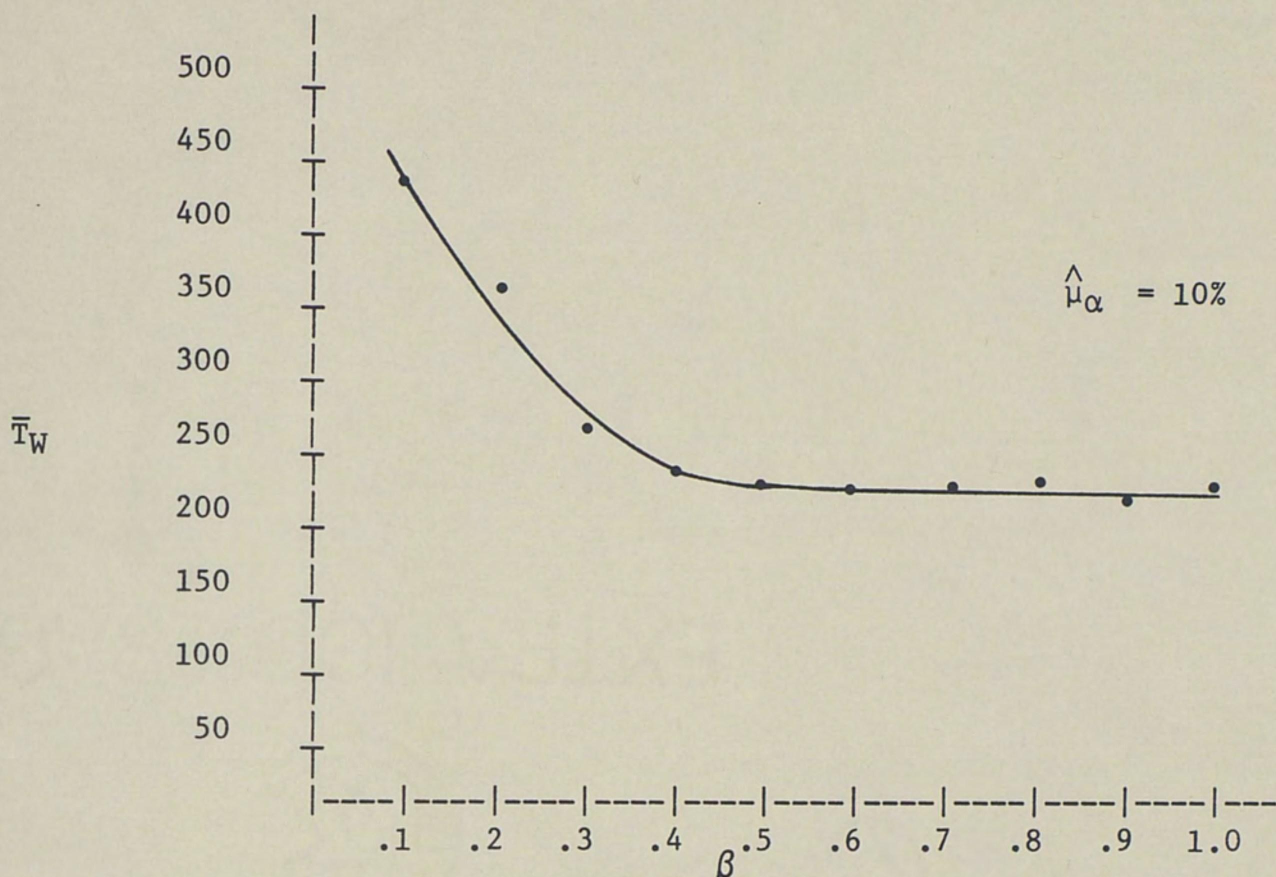


Figure 12. Plot of $\bar{T}_W$ versus β for $\hat{\mu}_\alpha$ = 10%

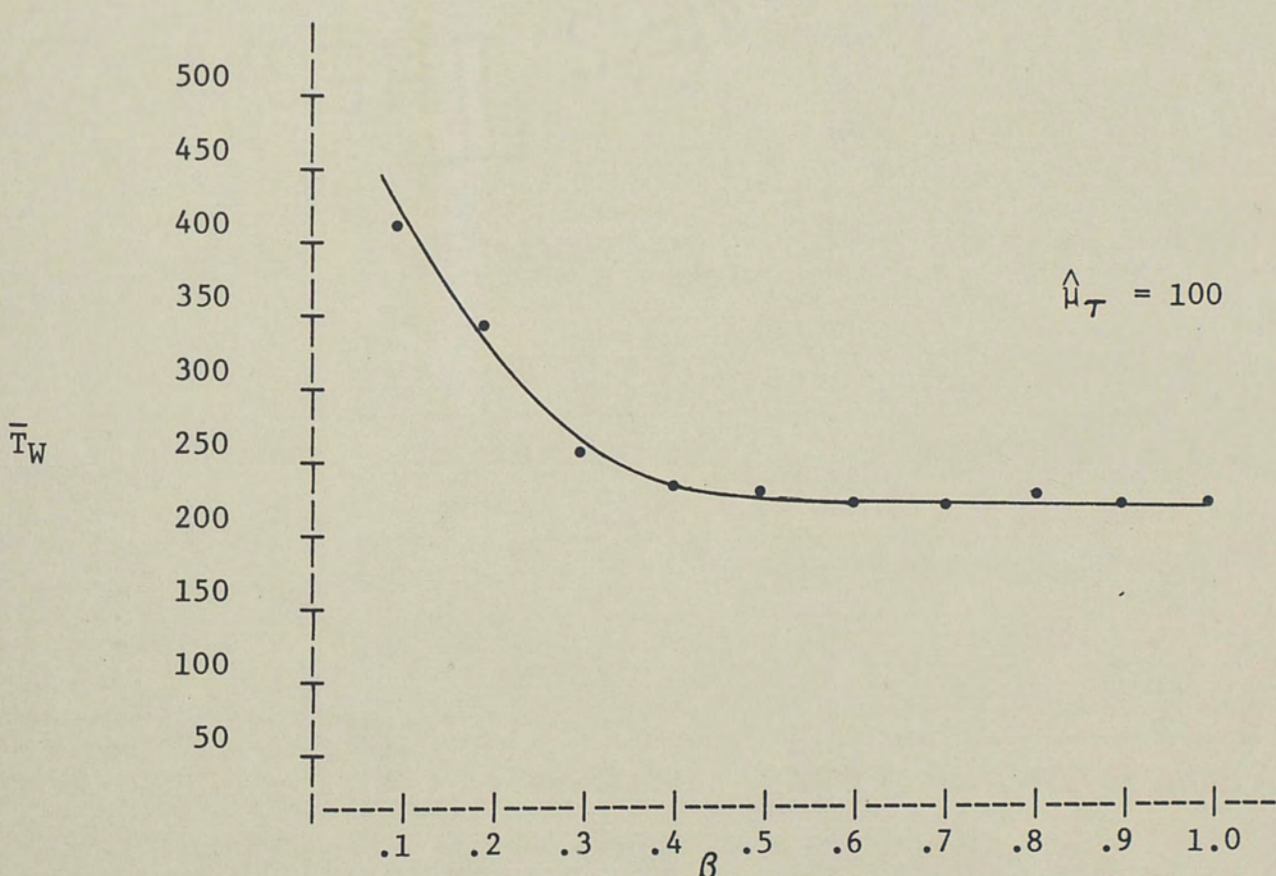


Figure 13. Plot of $\bar{T}_W$ versus β for $\hat{\mu}_\tau$ = 100

This constancy of value in relation to $\bar{T}_W$ may be due to the fact that with $\beta > .4$, the duration between PTT training interval refresh is greater than a threshold level. This would mean that the decreasing exponential nature of the human performance relationship R(t) overtakes or dampens the imparted training value of the PTT. For, as β increases, the time duration between PTT refresh increases, or T becomes larger. This constancy of value in $\bar{T}_W$ for $\beta \geqslant .5$ is exhibited in all $\bar{T}_W$ values for a defined R.V. mean. This would imply that PTT refresh scheduling should be performed for the training frequency interval of $.1 \leqslant \beta \leqslant .5$. The optimum values and best training effectiveness would probably occur for $\beta = .2$ or $\beta = .3$. In having the largest effective values of T_W experienced at these values, training time on the large simulator can be reduced. This concurs with the economic analysis in Chapter VII.

The graphs in Figures 14 through 19 are plotted for R_I, R_F, and α sensitivity mean values versus the corresponding sample mean $\bar{T}_W$, relative to a defined β. The graphs are for $\beta = .2$, and $\beta = .3$. These values were chosen to reflect the optimum training frequency for the PTT. A family of curves for a defined β occurs for each R.V. From corresponding data, the sensitivity coefficient can be determined. The graph of μ_{R_F} in Figure 16 portrays a non-linear portrait, with an increasing slope for larger values μ_{R_F}. This characteristic feature may be attributed to the fact that the larger values of R_F are approaching R_{CR}, resulting in an interaction of these two model variables.

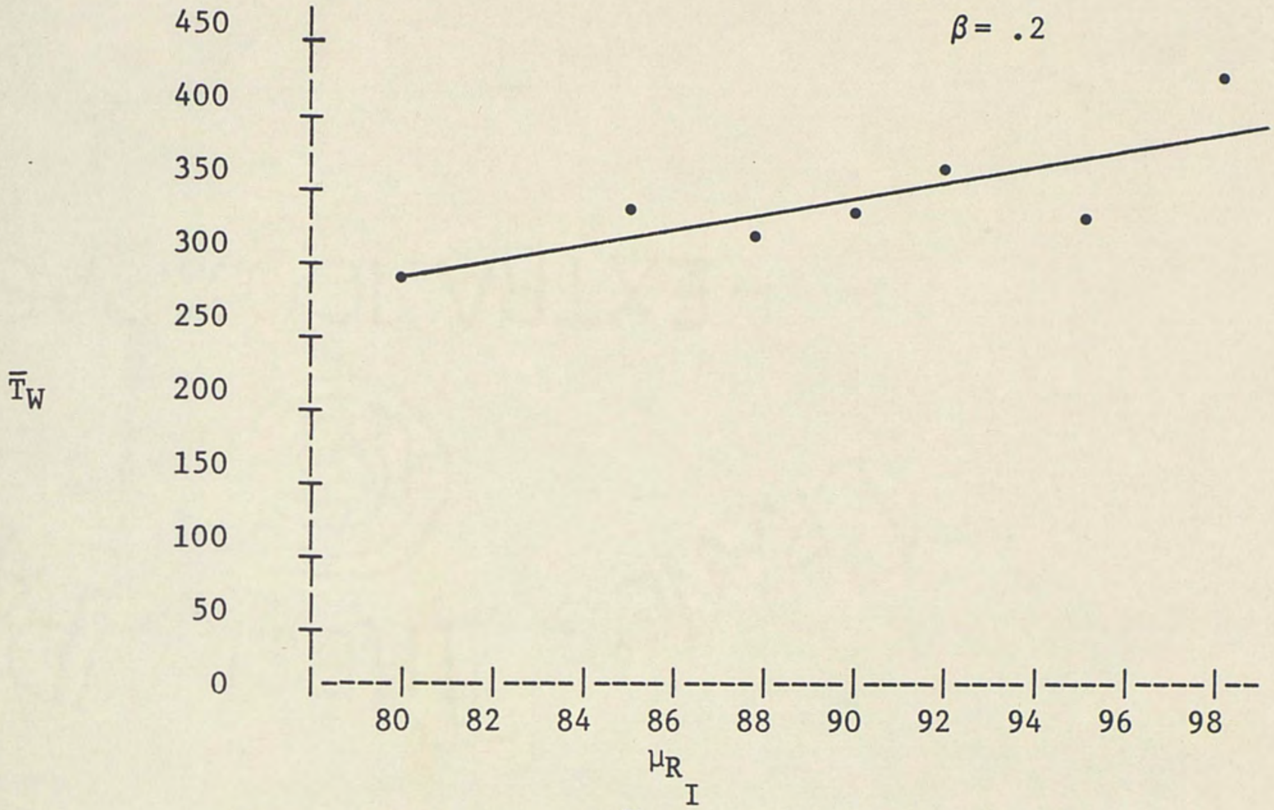


Figure 14. Graph of $\overline{T}_W$ versus μ_{R_I} for $\beta = .2$.

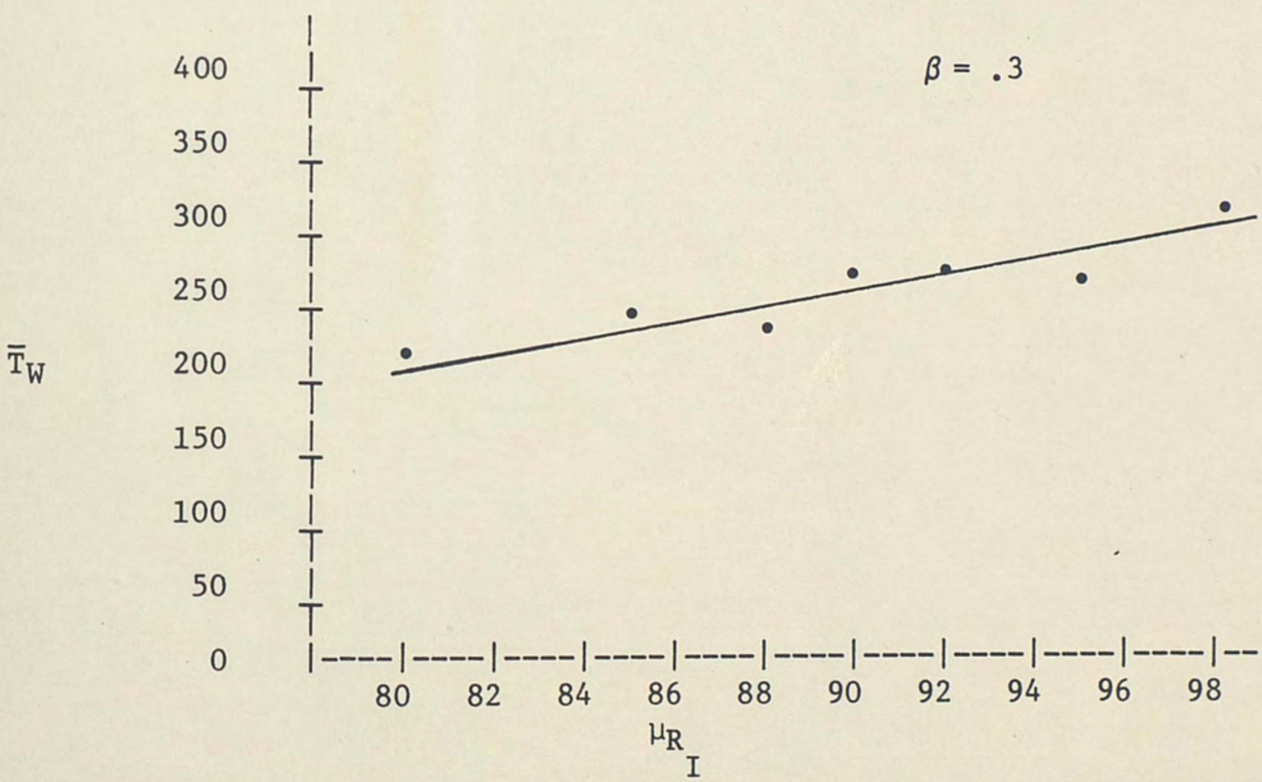


Figure 14. Graph of $\overline{T}_W$ versus μ_{R_I} for $\beta = .3$.

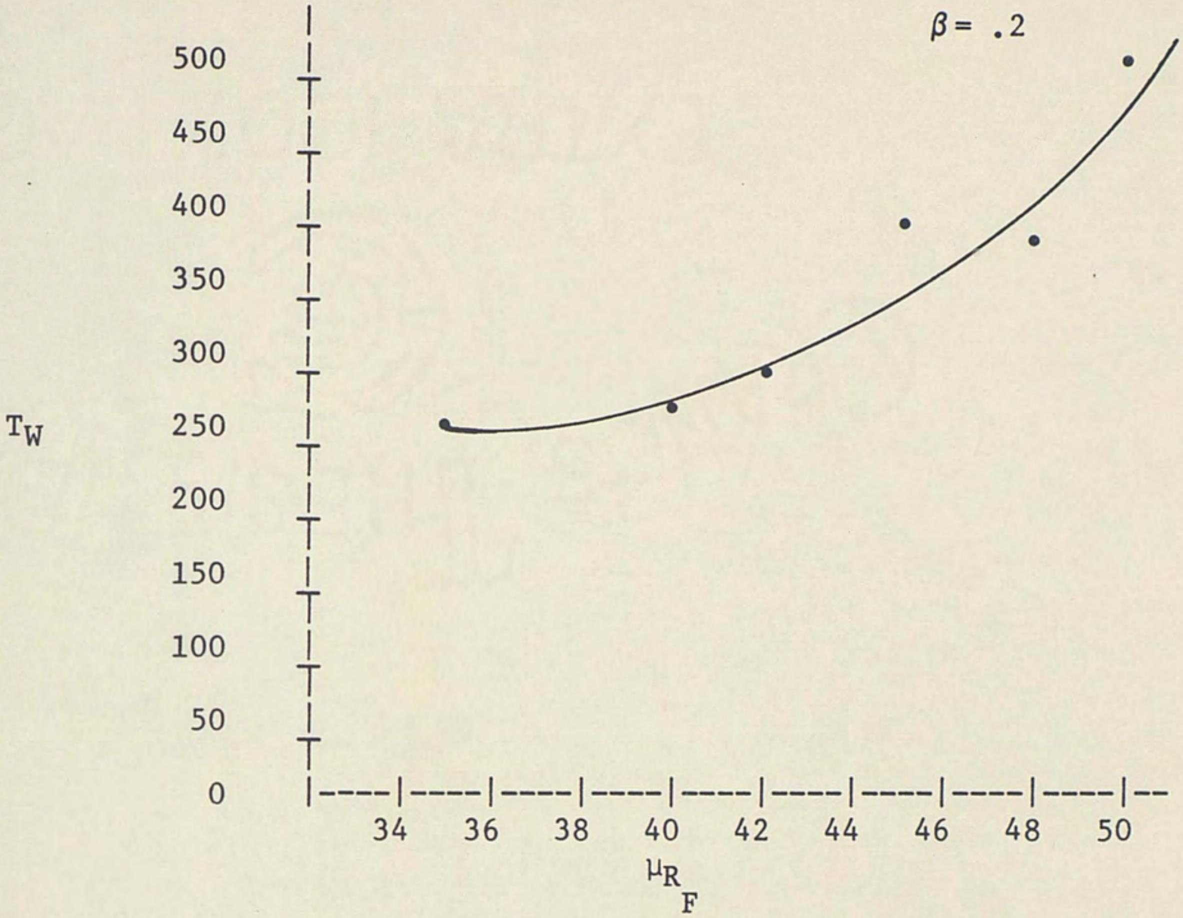


Figure 16. Graph of T_W versus μ_{R_F} for $\beta = .2$.

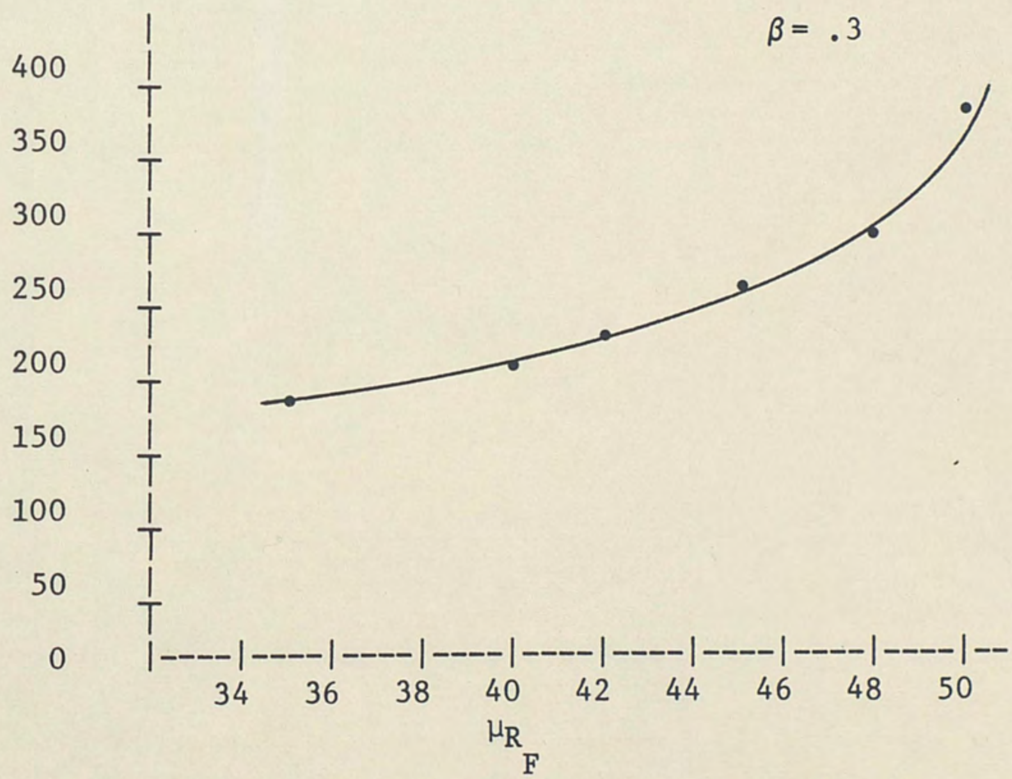


Figure 16. Graph of T_W versus μ_{R_F} for $\beta = .3$.

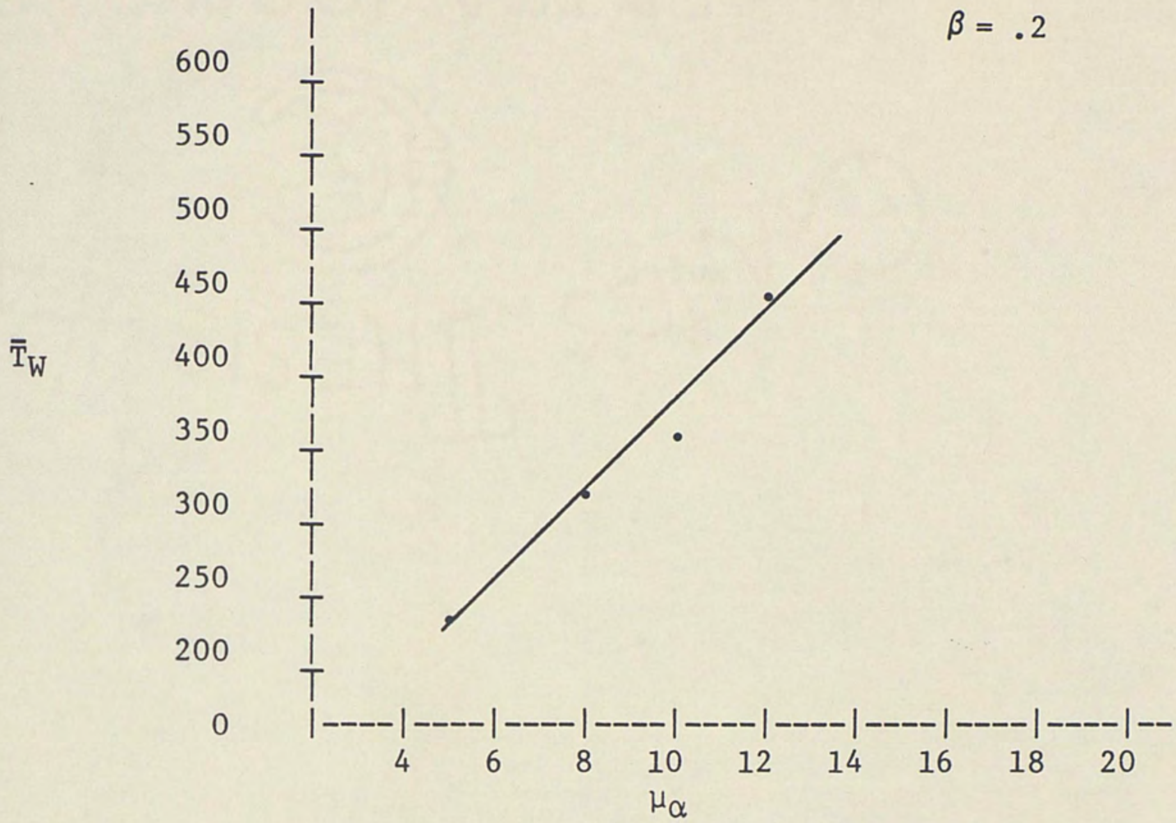


Figure 18. Graph of $\bar{T}_W$ versus μ_{R_I} for β = .2.

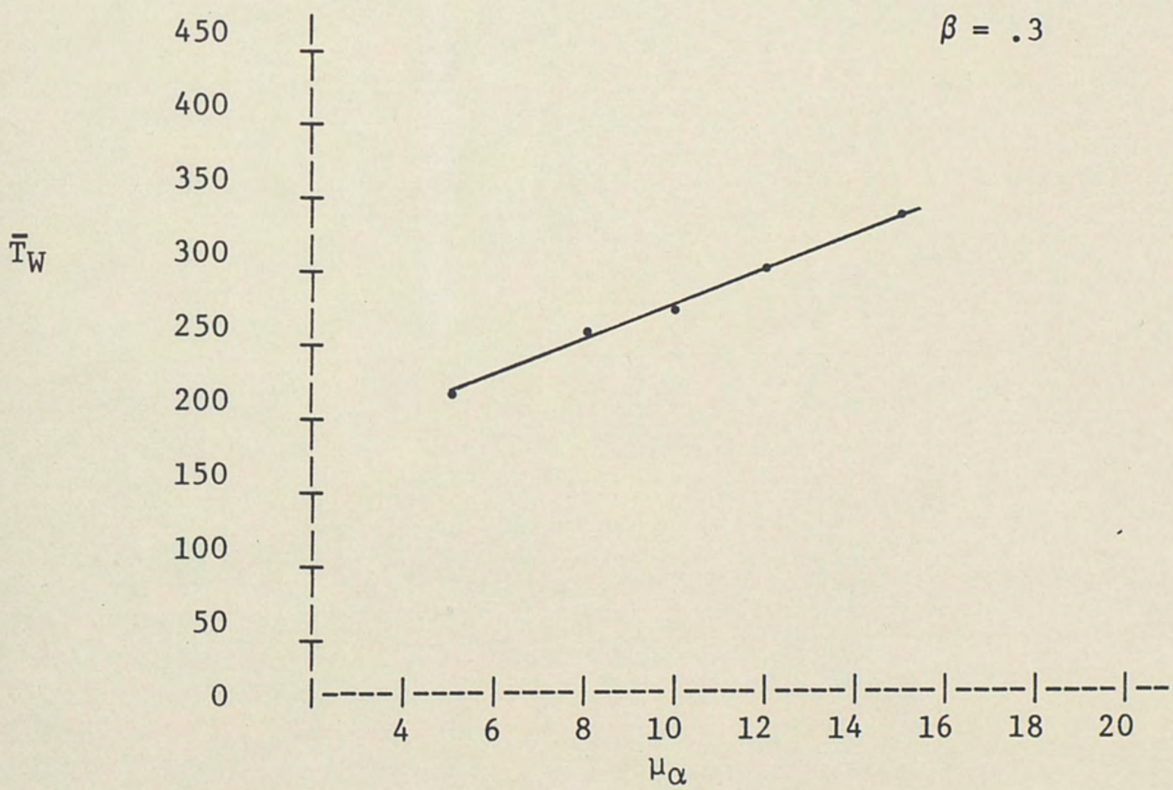


Figure 19. Graph of $\bar{T}_W$ versus μ_{R_I} for β = .3.

Calculation of the sensitivity coefficients (S.C.) is performed by using Equation (14) in relation to the data tables. The resulting S.C.s for the variables are contained in Table 5 and were calculated for $\beta = .2$, .3, and .4.

TABLE 5. SENSITIVITY COEFFICIENTS

Variable	$\beta = .2$	$\beta = .3$	$\beta = .4$	$\hat{\mu}$	Initial Value	Final Value
R_I	11.25	6.25	4.25	90	88	92
R_F	13.5	11.83	11.1	45	42	48
α	34.25	11.5	7.5	10	8	12
τ	5.2	3.98	2.7	100	90	110

Considering the sensitivity data, the most significant random variable was Alpha for the defined operational base at $\beta = .2$. For $\beta = .3$, R_F became the most sensitive R.V. As noted previously, R_F becomes highly sensitive or incurs greater rates of change as its value approaches R_{CR}. Thus R_F could be very significant at values near R_{CR}.

Observing the graph of μ_α versus $\bar{T}_W$ in Figures 18 and 19, it can be seen that the slope for $\beta = .2$ is less than the slope for $\beta = .3$. This occurrence correlates to the S.C. information contained in Table 5. For as β increases, the S.C. decreases.

The trends observed for the variables from the computer generated tables can be described as follows:

1. R_I.

a. As μ_{R_I} increases, $\overline{T}_W$ increases.

b. For a defined μ_{R_I}, as β increases, T_W decreases to a relatively constant set of values for $\beta \geqslant .5$.

2. R_F.

a. As μ_{R_F} increases, $\overline{T}_W$ increases.

b. For a defined μ_{R_F}, as β increases, $\overline{T}_W$ decreases to a constancy of values for $\beta \geqslant .5$.

3. Alpha.

a. As μ_α increases, $\overline{T}_W$ increases.

b. For a defined μ_α, $\overline{T}_W$ decreases as β increases, becoming relatively constant for $\beta \geqslant .5$.

4. Tau.

a. As μ_τ increases, $\overline{T}_W$ increases.

b. For a defined μ_τ, as β increases, $\overline{T}_W$ decreases to a relative level of constancy for $\beta \geqslant .5$.

5. All the random variables exhibited the same trends. R_{CR} will be shown to exhibit some different characteristics.

6. From Table 5 for the sensitivity coefficients, it is observed that as β increases, the S.C. decreases. This decrease occurs non-linearly in a less significant or smaller relative rate of change for the designated variable with β increasing. This again is

reflective of the exponential nature of R(t), for as β increases, T increases, moving or increasing the refresh interval period further along the exponential curve of R(t), resulting in less sensitivity.

A less extensive sensitivity analysis of the model variables was performed by exercising Subroutine 1300, which varies one system variable relative to the initially generated random variable data base. Utilization of this subprogram was performed to demonstrate its investigative operations and to provide collaboration of previous results. A comparative analysis was performed to the results of the preceding method, which adhered explicitly to the defined stochastic generation of random variables.

The subprogram was run for R_I, R_F, α, R_{CR}, with the sensitivity data from the computer outputs synthesized and compiled in Tables 6 through 9. The tables depict the random variable means and R_{CR} values employed to obtain T_W for each β value. The base or operational conditions for these sensitivity runs were the same defined means and standard deviations utilized in the preceding section analysis. This was done in order to promote compatibility for comparative analysis. The means and standard deviations for the random variable distributions were:

The graphs of $\mu_{R_{CR}}$ versus T_W and μ_α versus T_W for $\beta = .2$, $\beta = .3$ are shown in Figures 20 and 21. The curve's characteristic nature, along with the data in Tables 6 through 9 are virtually identical to the previous analysis. This shows a correlation in both methods.

TABLE 6. $\bar{T}_W$ VALUES FOR μ_{R_I} and β

$\bar{T}_W$ Values

μ_{R_I} \ β	.1	.2	.3	.4	.5	.6	.7	.8	.9	1.0
80	667	328	241	221	212	207	212	202	204	217
85	735	359	261	232	228	220	224	220	219	227
90	794	386	282	254	240	239	231	252	233	232

TABLE 7. $\bar{T}_W$ VALUES FOR μ_{R_F} and β

$\bar{T}_W$ Values

μ_{R_F} \ β	.1	.2	.3	.4	.5	.6	.7	.8	.9	1.0
40	689	297	220	196	197	186	206	165	186	207
45	732	374	272	242	241	242	217	248	241	207
50	946	528	385	343	328	313	300	331	293	310

TABLE 8. $\bar{T}_W$ VALUES FOR μ_α and β

$\bar{T}_W$ Values

μ_α \ β	.1	.2	.3	.4	.5	.6	.7	.8	.9	1.0
3	245	210	203	200	206	213	216	216	218	217
6	412	252	229	223	217	216	224	220	227	226
9	918	309	254	237	229	226	231	235	227	232
12	918	402	288	260	248	239	235	243	238	232
15	917	574	331	283	261	257	252	252	242	247

TABLE 9. $\bar{T}_W$ VALUES FOR $\mu_{R_{CR}}$ and β

$\bar{T}_W$ Values

R_{CR} \ β	.1	.2	.3	.4	.5	.6	.7	.8	.9	1.0
50	814	525	373	336	317	314	316	304	302	314
55	794	356	264	231	225	217	224	228	218	222
60	641	258	193	176	171	174	167	169	186	197

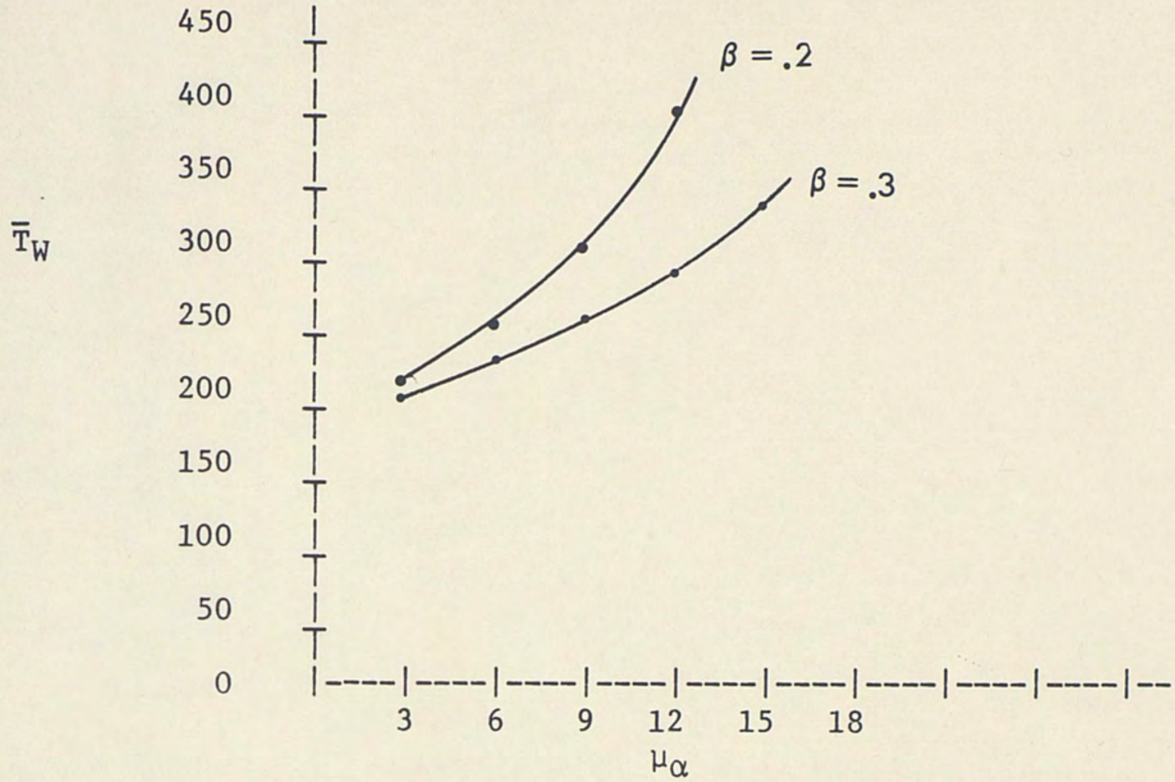


Figure 20. Graph of $\overline{T}_W$ versus μ_α for β values.

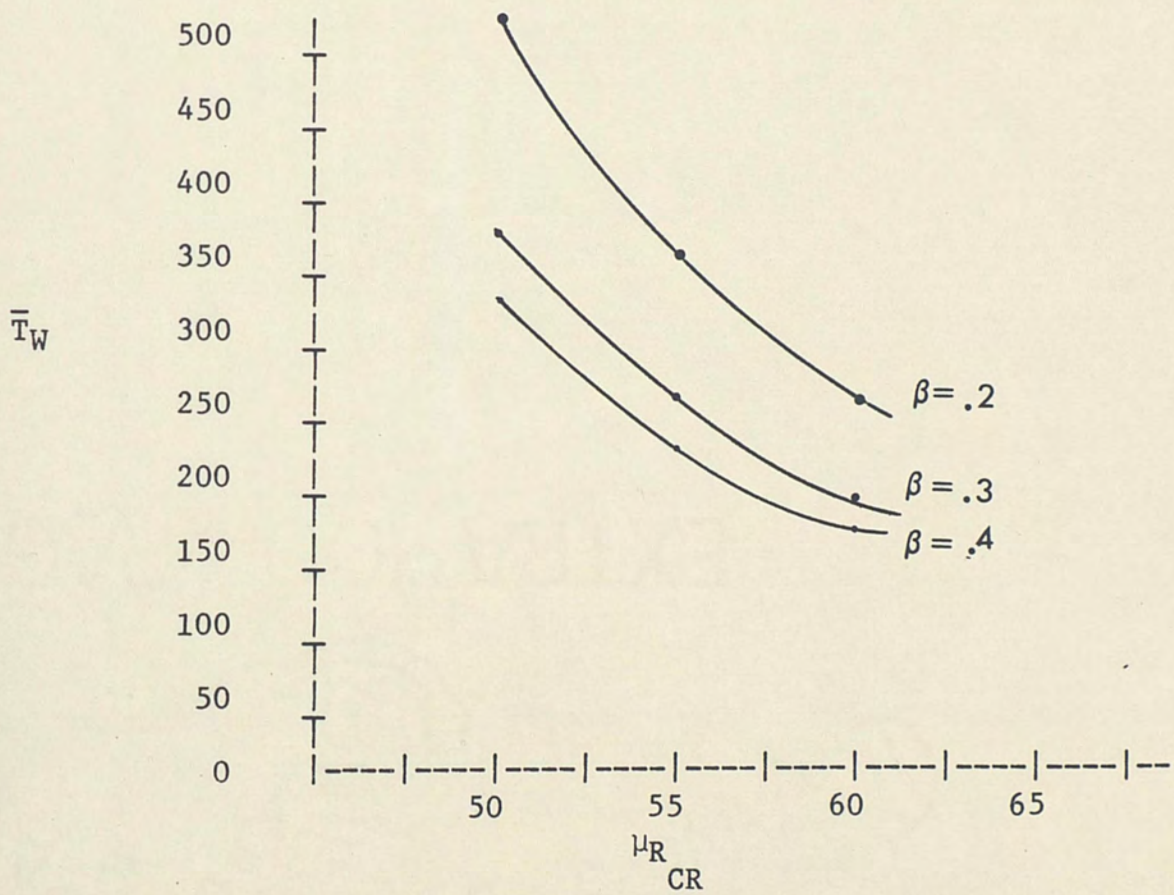


Figure 21. Graph of $\overline{T}_W$ versus $\mu_{R_{CR}}$ for β values.

The graphs of R_{CR} show a decreasing, non-linear type of function. The trends associated with R_{CR} can be categorized as:

1. As R_{CR} increases, $\overline{T}_W$ decreases. This deduction holds true intuitively, because as R_{CR} increases, the sooner R(t) will intersect R_{CR}.

2. As β increases for a value of R_{CR}, $\overline{T}_W$ decreases; very rapidly for small values of β, and less rapidly as β increases.

The sensitivity coefficients for the operational data base are contained in Table 10.

In observing the S.C. of each variable, for β = .2, R_{CR} is the most significant variable for the operational base, with this holding true for all calculated values of β.

TABLE 10. SENSITIVITY COEFFICIENTS

Variable	β = .2	β = .3	β = .4	$\hat{\mu}_2$	μ_1	μ_3
R_I	5.8	4.1	3.3	85	80	90
R_F	23.1	16.5	14.7	45	40	50
α	25.0	9.8	6.16	9	6	12
R_{CR}	26.7	18.0	16.0	55	50	60

For β = .2, S.C. for α is greater than S.C. for R_F, but as β increases, or for β = .3 and β = .4, R_F becomes more significant than α for the operational base. R_{CR} is still the most significant. This correlation for R_F and α also holds when observing the values in Table 5 of the previous method.

In performing a comparative analysis of the two methods for determining sensitivity, there is strong correlation in the data, providing confirmation of the results. Both methods coordinately demonstrate the same associated characteristics and trends for the PTT model.

CHAPTER IX

CONCLUSION

In performing an analysis of PTT refresh scheduling, the characteristics and nature of the human performance model have been presented. Development of a computer program provided a means for calculating the large system simulator training cycle times in conjunction with PTT refresh training. The computer program and model algorithm have considered and simulated real world conditions through random generation of variables, along with implementing operational system constraints. Through interactive computer programming, the user will be able to make effective decisions in regard to scheduling times for PTT refresh training.

By investigating the human performance model and system parameters for PTT scheduling, the most significant factor or variable was identified for a defined operational base. The generated sensitivity data allowed the graphing of the model parameters. From these results, a further understanding of the factors involved in determining simulator scheduling times were developed.

A survey of the economical principles associated with PTT refresh training provided some insights into the savings which could be realized. Further definition in the field of identifying the economic

factors and their associated utility or dollar value in employing PTTs is needed. This analysis would allow for the prediction of actual savings. The utilization of low-cost Part Task Trainers in the training cycle of large system simulators can be an integral factor in maintaining effective skill levels of personnel. Reduction in the scheduling impact on the full-up simulator will occur. The PTT computer program can become a practical tool in the analysis of training situations. Planners will thus have the capability to institute effective training through PTT refresh scheduling.

APPENDIX A

LOW COST PART TASK TRAINER DESIGNS

To obtain an understanding and appreciation for Low Cost Part Task Trainers (LCPTT), this section will briefly describe some Part Task Trainers, their applications, and designs.

A main feature of Low Cost PTTs are that they are inherently suited to using modern state-of-the-art microprocess systems, digital hardware, and graphics processor equipment. As stated previously, PTTs in general can range across a whole spectrum of functional training. Training can encompass a system maintenance training, operational flight emergency procedures, engine simulators, ASW models and sonar trainers, and other procedural trainers for various sensor/weapon systems.

With the advent of microcomputer systems and graphic processors, generic or variable custom made training can be accomplished. What this means, through the power of developed software and software packages, the formats for any display can be graphically represented. By calling the appropriate control function, access to the customized file for the simulated system is achieved, displaying the particular format required. This format can be the console display of a radar, sonar, or cockpit system. Within the simulation, problem control provides on the screen designated targets and dynamic conditions of operation to

realistically portray the actual equipment. The trainee can either proceed through a predetermined set of steps, interactively inputting responses to the conditions, or more sophisticated simulation can be real-time in nature, dynamically responding to the trainee's inputs from a joystick or a mockup throttle. Real-time inputs are transmitted to the simulation algorithm, providing perceptual realism to the operating trainee.

Software And Graphic Processor Application For A LCPTT

An example of a Low Cost Part Task Trainer that was delivered by the author was a device that simulated the passive sonar operation of the AN/SQR-18. The Interim SQR-18 Trainer, as it was designated, consisted of two Tektronix graphic computers that were interconnected to provide:

1. A simulation of an ASW Passive Sonar Display, SQR-18.

2. The control of tactical sonar problems, including the history of the exercise.

The trainer system consisted of two graphic processors, interfaces, and associated peripherals. One graphic processor allows control of ship heading/speed and other factors to create a given acoustic environment. It is essentially the problem control. The other processor displays the SQR-18 format and targets. Interaction is allowed at both displays.

Target vessel history and track of the problem development are stored in the computer memory. These problems can be utilized as a library of demonstration problems, providing problem control use in training, or this capability can allow critique analysis of the

trainee's performance in operational procedures and problem evaluation. Since sonar employs analysis of sound waves in the ocean medium, problem development in real-time can be slow. Thus, computerized simulation allows that the complete problem history that is retained and can be replayed for evaluation at rates up to 240 times real-time.

The Interim SQR-18 Trainer can perform training in the independent or stand alone mode of operation. This LCPTT can also operate through a compatible interface as a generic sonar in a team training situation, with the large simulator providing target inputs and control.

Firmware Application In Low Cost Part Task Trainers

The example above illustrates how employment of microcomputer software contained on floppy disks/tape cartridges coupled with the power of graphic processing can provide cost effective and learning effective part task training. Another application of digital systems and microprocessors using firmware in PROM/ROM memory is going to be applied for an aircraft electrical systems trainer. For this particular part task trainer, the training parameters are predominantly well defined through an operational procedure. This means there exists defined output values on the gauges and equipment for each training step, which are effected by insertion of system faults through the microprocessor.

The microprocessor system and the associated PROM/ROM firmware are contained in the Instructor Console. The microprocessor system will perform the trainer-fault insertion, trainer procedures/routines and overall system control. The type of fault inserted produces defined

malfunction symptoms and system values. A proper maintenance analysis by the trainee will lead one to the cause of the fault. Faults can be a simulated broken wire (switch or relay), a simulated trainer peculiar aircraft electronic equipment subassembly failure, and a failed relay, sensor, or other aircraft electrical component.

In summary, the microprocessor system for this part task trainer controls the trainer operational logic procedures.

APPENDIX B

LIST OF SUBPROGRAMS

The following list of subroutines/subprograms are designated by a remark or title line in the computer listing (Appendix F). A brief description of the subroutine is contained in the remark. The following list provides the titles and brief descriptions of each subprogram.

1. Subroutine 300 - Random Variable Generation. This program generates the random variables R_I, R_F, α, and τ. They are outputted in the form of matrix, A(I,J). Lines 300-799.

2. Subroutine 800 - Subroutine for Calculating T_W, R(K,J), T2(K,J), T3(K,J), and R1(K,J). Calculates and outputs the system models matrix values. Lines 800-899.

3. Subroutine 900 - Subroutine for Printing Random Variable Matrix Labels. Lines 900-999.

4. Subprogram 1000 - Subroutine for Printing Output Matrices: R(K,J), U(K,J), T2(K,J), and T3(K,J). Lines 1000-1168.

5. Subprogram 1200 - Routine for Output Matrix Format.

6. Subprogram 1300 - Subroutine for Performing Sensitivity Analysis of Model R.V.s and Parameters. Lines 1300-1616.

7. Subprogram 1860 - Subroutine for Calculating Sample Mean and Standard Deviation of T_W Matrix.

8. Subprogram 2130 - Subroutine for Calculating Simpson Numerical Value for Normal Distribution.

9. Subprogram 2800 - Subroutine for Calculating Chi-Square Analysis of R.V. Matrix.

10. Subprogram 3100 - Subroutine for Printing Interval Value Matrix B3(I,L) and Probability Value Matrix B3(I,L).

APPENDIX C

SUBPROGRAMS

Random Variable Generation

Generation of the four random variables are contained in Subprogram 300. The process utilized establishes the random variable vector sets and overall program random variable matrix, A(I,J).

A matrix 4 X 20, A(4,20), is created for the inputting of randomly generated vector values, using the defined mean and S.D. values for the normal distribution of R_F, R_I, α, and τ.

Random Number Function And Generation

The Wang computer system has an internal random number function called RND or RND(N), which produces random numbers between 0 and 1. The routine for producing the random number z, which is used in relation to the defined R.V. μ and σ to generate the random variable value is:

340 For I = 1 to 4	Calls out specific R.V.
350 For J = 1 to 20	Generates the representative sample set for 20 individuals
360 X = 0	Initializes value
370 For N = 1 to 20	Calls for 20 values between 0 and 1

380	X = X + RND(N)	Calls RND
390	Next N	Loops through 20 values
400	z = 0	Initializes random numbers
410	z = (x-10)/(20/12) 0.5)	Set formula for R.V. value

Then for each specific R.V., the program uses the equation in line 470, or A(I,J) = μ + (σ X z), which fill the R.V. matrix A(I,J) as the computer steps through the logic loops.

The main program calls on subroutine 800 for the calculation of the following values:

T_W (or T2(K,J)	-	Scheduling time for retraining or simulation with PTT refresh
R(K,J)	-	Model equation performance value that is less than relative r_{CR}
R2(J)	-	Relative critical performance value
T3(K,J)	-	Time period value for each
U(K,T)	-	Number of T time periods to reach r_{CR}

The subroutine first calculates r_{CR}, which becomes the control variable in the comparison relationship to the human performance value. The computer defines r_{CR} in the array R2(J), and is expressed as:

$$r_{CR} = R2(J) = \frac{(R_{CR} - R_F)}{(R_I - R_F)}$$

The model equation is calculated R(K,J) and is compared to r_{CR} by the statement:

IF R(K,J) $\leqslant$ R2(J).

When this condition is fulfilled, the computer will print the number of T periods or intervals it took to reach this value. The computer program limits the number of loops or interactive T calculations to 100, which is well beyond usual normal operational and scheduling periods of time. To determine T_W, or T2(K,J), the U(K,J) matrix number is multipled times T3(K,J), the value of T, arriving at T_W which is outputted as T2(K,J).

Subprogram For Calculating The Sample Mean And Standard Deviation Of T_W

Subroutine 1860 performs the analysis on T_W of calculating T_W's sample mean and standard deviation. The routine sorts out any T_W values that are equal to 0 or would be equal to or greater than 100 times T3(K,J), the time T value, which would thus could exceed the 100 interaction limit described in subsection 2. A counter, C = C + 1, with C initialized to 0, is employed to determine the number of values in the calculation loop are used in figuring the value of T_W and σ_{T_W}. The sample mean of T_W is calculated in the standard manner of:

$$\mu_{T_W} = \bar{T}_W = \frac{\Sigma\, T_W}{C}$$

C = number of samples values.

T_W's sample standard deviation is calculated using the standard formula:

$$\sigma_{T_W} = \frac{(\Sigma T_W - \mu_{T_W})^2}{C}$$

$\overline{T}_W$ and $\sigma\ T_W$ values are then executed by the program to be outputted in vector form.

APPENDIX D

DESCRIPTION OF SUMMARY REPORT

The computer summary output contains the following information for PTT time scheduling:

1. Listing of Inputted Mean and Standard Deviation for R_I, R_F, Alpha, and Tau.

2. Random Variable Data Matrix, A(I,J). A matrix format of the four R.V.s for the sample set of 20 represented trainees. The matrix rows are the R.V.s, and the columns represent individual trainees. A(4,20).

3. Interval Value Matrix B3(I,L). These are the calculated sector or interval values for the R.V.s between 0 and ± 2.5 standard deviations. There are six sector boundaries, comprising a total of seven sections (cells) where the frequency of R.V. occurrence is counted by the computer for the chi-square analysis.

4. Area Probability Value Matrix F3(4,7). These values represent the area contained in each designated sector for a normal distribution. There are seven sectors, which would give six degrees of freedom, or ν = # Sectors - 1; thus $\nu = 6$ for this case.

5. Chi-Square Analysis - Frequency of Data in Intervals for R.V. Matrix. The computer program performs a counting routine on each sector to determine the frequency of R.V.s occurrence. These values are used in calculations of the chi-square statistic.

6. Chi-Square Statistic Reference Number C1(4). The chi-square statistic is printed out for each R.V. . These values are then coordinated for 6 degrees of freedom in the chi-square table to determine the confidence level of the sample population to the normal distribution.

7. Value of Inputted Absolute Critical Human Performance.

8. Relative Critical Human Performance Values of Equation 8 for each of the 20 Sample Population.

9. Outputs of values of β and individual sample which does not meet model criteria, or R(Beta, Sample). Coding Line 817.

10. T3(Beta, S.I.) Matrix, Time Value for Each Tau Times Beta. This matrix provides the T period value for each interval of PTT refresh training. It is a 10 X 20 matrix. As β increases, the T period for PTT refresh increases.

11. R(K,J) Matrix, Performance Value for Training Cycle Below R_{CR}. The calculated human performance value for the interation for $R_N(t) < r_{CR}(J)$. K = Beta, J = Sample Individual.

12. U(K,J) Matrix, Number of Cycles or T-Time Periods to Reach $R_N(t) < r_{CR}$. This matrix provides the number of T-periods to reach r_{CR}.

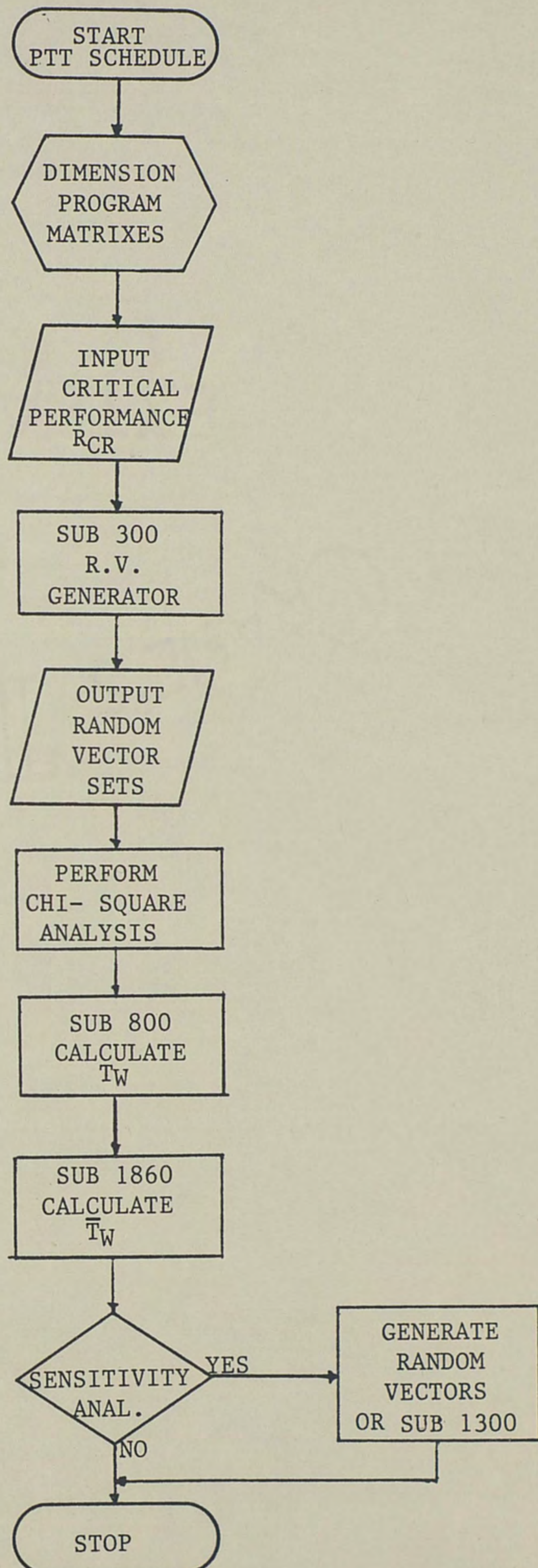
13. T2(K,J) or Tw Matrix, Time Value for Retraining on System Simulator. Derived values of Tw for the generated sample population.

14. Sample $\overline{T}_W$ Mean for each β .

15. Sample $\overline{T}_W$ Standard Deviation for each β .

APPENDIX E

MAIN PROGRAM LOGIC FLOW DIAGRAM


START
PTT SCHEDULE
DIMENSION
PROGRAM
MATRIXES
INPUT
CRITICAL
PERFORMANCE
R_{CR}
SUB 300
R.V.
GENERATOR
OUTPUT
RANDOM
VECTOR
SETS
PERFORM
CHI- SQUARE
ANALYSIS
SUB 800
CALCULATE
T_W
SUB 1860
CALCULATE
$\overline{T}_W$
SENSITIVITY
ANAL.
YES
NO
GENERATE
RANDOM
VECTORS
OR SUB 1300
STOP

APPENDIX F

PROGRAM LISTINGS

PROGRAM LISTING

```
  1 % SCRATCH  F/B10, "PTT4": SAVE F/B10,()"PTT4"
  2      REM  WILLIAM FELLOWS
  3      REM PROGRAM FOR CALCULATRING PART TASK TRAINING CYCLE INTEGRAT
    ION(PTT)
  5 PRINT TAB(30); "PART TASK TRAINER CLCLE INTEGRATION"
  8 DIM R(10,20), T1(100), T2(10,20), A(4,20), T3(10,20), U(10,20), A1(
    4,20),           F(4,10),F3(4,10), R2(20),R1(10,20),C1(4),S8(10,20
    ),T(3)
  9 DIM U2(10,20), U3(10,20), V1(10,20), S(4,20),S1(10), S2(10), S3(10)
    , M1(10),    M2(10), M3(10),S5(4),B3(4,10), M(4), H1(4,10),H3(4,3)
    ,R8(3)
 10 INPUT "RCR=R1= ",R1                                :REM RCR IS MININUM LE
    VEL OF ACCEPTABLE HUMAN                                           PERFORM
    ANCE
 12 GOSUB  300
 14 PRINT
 15 PRINT TAB(30); "RANDOM VARIABLE DATA MATRIX A(I,J)"
 16 PRINT
 19      REM TN=T1 , TW=TRAINING CYCLE TIME=T2
 20 PRINT "      ";
 21 FOR J= 1  TO 20
 22 PRINTUSING "   ## ",J;
 23 NEXT J
 25 PRINT
 26 PRINT
 28 FOR I = 1  TO 4
 29 GOSUB  900
 32 FOR J = 1  TO 20
 34 PRINTUSING   35 , A(I,J);
 35 %####.#
 38 NEXT J
 40 PRINT
 41 PRINT
 42 NEXT I
 44 PRINT
 45      REM SAVE ROUTINE FOR RANDOM VARIABLE DATA MATRIX
 46 FOR I = 1  TO 4
 48 FOR J = 1  TO 20
 49  LET S(I,J) = A(I,J)
 50 NEXT J
 51 NEXT I
 52 GOSUB 2800
 54 PRINT "INPUT R-CRITICAL, RCR= R1 = ";R1
 55 GOSUB 2000
 56 GOSUB  800
 58 GOSUB 2000
 60 GOSUB 1860
130      REM PERFORMANCE OF CHI SQUARE ANALYSIS ON TW
139      REM THIS PART OF THE ALOGRITHIM WILL EITHER RUN ANOTHER DATA B
    ASE THROUGH      TO OBTAIN NEW VALUES OR PERFORM A SENSITIVITY ANA
    LYSIS OF THE RV'S, OR STOP.
```

```
140 INPUT " SENSITIVITY ANALYSIS = 1 ,  GENERATE NEW DATA = 2 , STOP =
    3 ",W
141 PRINT " SENSITIVITY ANALYSIS = 1 ,  GENERATE NEW DATA = 2 , STOP =
    3 ",";W;"
142 IF W=1  THEN GOSUB 1300
144 IF W=2  THEN    3
146 IF W=3  THEN  158
148 IF W=0  THEN  140
150 IF W<0  THEN  140
152 GOTO  140
156 GOTO  158
158 STOP
300      REM RANDOM VARIABLE GENERATION(RVG)
305 PRINT
310 PRINT TAB(30);"RANDOM VARIABLE"
315      REM SETTING UP DATA MATRIX FOR RV'S
320 DIM A(4,20)
325      REM A(I,J) WILL BE RV DATA MATRIX
330      REM A(1,J)= RI, INITIAL PERFORMANCE IN OPERATIONAL ENVIRANMENT
331      REM A(2,J)= RF, ULTIMATE PERFORMANCE IN OPERATIONAL ENVIRONMEN
    T
332      REM A(3,J)= ALPHA, RELATIVE PERFORMANCE INCREASE
333      REM A(I=4,J)= TAU, EXPONENTIAL DECAY CONSTANT
340 FOR I=1  TO 4
350 FOR J= 1  TO 20
360 LET X=0
370 FOR N= 1  TO 20
380 X= X + RND(N)
390 NEXT N
400 LET Z=0
410 Z = (X-10)/((20/12)↑0.5)
420 IF I=1  THEN  440
425 IF I=2  THEN  500
426 IF I=3  THEN  550
427 IF I=4  THEN  600
440      REM CALC RI
441 IF J >= 2  THEN  470
442 PRINT
455 INPUT "RI MEAN IS = ",M
465 INPUT "RI SIGMA IS = ",S
466 PRINT TAB(30);"RI MEAN IS = ";M; TAB(45); "RI SIGMA IS = ";S
470 LET A(1,J)= M + (S*Z)
471 LET M(1) = M
    :S5(1) = S
472 IF A(1,J)>100  THEN  475
473 GOTO  485
475 LET A(1,J)=100
485 GOTO  672
500      REM CALC RF
503 IF J>=2  THEN  525
504 PRINT
```

```
505 INPUT "RF MEAN IS = ",M
510 INPUT "RF SIGMA IS = ",S
515 IF R1< M-(2.5*S)  THEN  517
516 GOTO  520
517 PRINT " RCR <(RF MEAN - 2.5*RF SIGMA), POSSIBLE GENERATION OF INVAL
    ID VALUE                 OF RF, WHERE RF>RCR "
518 GOSUB 2000
519 GOTO  505
520 PRINT TAB(30); "RF MEAN IS = ";M; TAB(45); "RF SIGMA IS = ";S
525 LET A(2,J) = M+(S*Z)
530 LET M(2) = M
    :S5(2) = S
535 IF A(2,J)>R1  THEN  545
540 GOTO  672
542 GOTO  550
545 PRINT "INVALID CRITERIA, RF CANNOT BE GREATER THAN R-CRITICAL, RCR,
                           RECOMMEND REDUCTION IN VALUE OF RF MEAN (CAN
    REDUCE SIGMA OR BOTH)"
546 GOTO  505
547 GOSUB 2000
548 LET J=1
549 GOTO  505
550      REM CALC ALPHA
553 IF J>=2  THEN  575
554 PRINT
555 INPUT "ALPHA MEAN IS  = ",M
560 INPUT "ALPHA SIGMA IS = ",S
562 IF (M-2.5*S)>0  THEN  570
564 PRINT " VALUE LESS THAN OR EQUAL TO ZERO POSSIBLE, INCREASE ALPHA M
    EAN"
565 GOSUB 2000
568 GOTO  555
570 PRINT TAB(30); "ALPHA MEAN IS = ";M; TAB(45);"ALPHA SIGMA IS = ";S
575 LET A(3,J)=M+(S*Z)
588 LET M(3) = M
    :S5(3) = S
589 IF A(3,J)<=0  THEN  591
590 GOTO  672
591 PRINT
    :PRINT "VALUE OF ALPHA <=0, CHANGE VALUE OF MEAN/SIGMA"
592 GOTO  554
600      REM CALC TAU, TRAINING FREQUENCY
610 IF J >=2  THEN  635
615 PRINT
620 INPUT "TAU MEAN = ",M
625 INPUT " TAU SIGMA IS = ",S
630 PRINT TAB(30); "TAU MEAN IS = ";M; TAB(45); "TAU SIGMA IS = ";S
635 LET A(4,J)= M+(S*Z)
640 LET M(4) = M
    :S5(4) = S
655 FOR K =1  TO 10
```

```
660 LET T3(K,J)=A(4,J)*(K*.1)
662 LET S8(K,J)=T3(K,J)
665 NEXT K
672 NEXT J
673 NEXT I
675 RETURN
800      REM SUBROUTINE FOR CALCULATING TW, R(K,J),T2(K,J),T3(K,J),& R1
     (K,J)
802      REM DETERMINATION OF RELATIVE RCRITICAL(RCR), R1(J)
804 FOR J = 1  TO 20
808 R2(J) = (R1-A(2,J))/(A(1,J)-A(2,J))
810 NEXT J
811 GOSUB 2000
     :PRINT TAB(30);"RELATIVE RCR R2(J)"
     : PRINT "  J=  ";
     :              FOR J=1  TO 20
     :PRINTUSING " ##   ",J;
     :NEXT J
812 GOSUB 2000
     :PRINT "      ";
     :FOR J=1  TO 20
     :PRINTUSING  813 ,R2(J);
813 %.####
814 NEXT J
815 GOSUB 2000
816 FOR J = 1  TO 20
     :FOR K=1  TO 10
817 IF EXP((.1)*K)<(1 + ((.01)*A(3,J)))  THEN  848
818 FOR U = 1  TO 100
820 LET R(K,J)= ((1+(.01)*A(3,J))↑U)*EXP(-((.1)*K)*U)
822 LET R1(K,J)= EXP(-(N*(((.1)*K)-LOG(1+(.01)*A(3,J)))))
824 IF R(K,J)<R2(J)  THEN  828
826 NEXT U
828 LET U(K,J)=U
830 IF U = 100  THEN  834
832 LET T2(K,J) = U*T3(K,J)
833 GOTO  836
834 LET T2(K,J) = 0
836 NEXT K
840 NEXT J
844 GOTO  854
848 PRINT TAB(30);"R(";K;",";J;") DOES NOT MEET CRITERIA"
852 GOTO  836
854 GOSUB 1000
856 PRINT
860 RETURN
900      REM SUBROUTINE FOR PRINTING RV MATRIX LABELS
905 IF I=1  THEN  940
910 IF I=2  THEN  945
915 IF I=3  THEN  950
920 IF I=4  THEN  955
```

```
 940 PRINTUSING "  RI   ";
 942 GOTO  960
 945 PRINTUSING "  RF   ";
 948 GOTO  960
 950 PRINTUSING "ALPHA  ";
 952 GOTO  960
 955 PRINTUSING " TAU   ";
 960 RETURN
1000      REM SUBROUTINE FOR PRINTING OUTPUT MATRIXES
1005      REM R(K,J) = PERFORMANCE VALUE FOR TIME CYCLE BELOW R-CRITICAL
1006      REM U(K,J) = U, TIME CYCLES OR INTERATIONS TO REACH RCR ↑
1007      REM T2(K,J) = U*T(K,J) = TW MATRIX , THE TIME VALUE FOR LEARNI
    NG                  PERFORMANCE BELOW RCR
1008      REM T3(K,J) =TIME VALUE FOR EACH TAU AS A FUNCTION OF BETA, B=
     .1 - 1.0
1009      REM K = BETA * .1 ; BETA = FREQUENCY OF TRAINING FACTOR
1020 PRINT
1022 PRINT
1025 PRINT TAB(40);"T3(K,J) MATRIX, TIME VALUE FOR EACH TAU / BETA "
1028 PRINT "       ";
1030 GOSUB 1200
1035 FOR K = 1  TO 10
1038 PRINTUSING "B=#.# ",K*.1;
1040 FOR J = 1  TO 20
1042 PRINTUSING  1045 , T3(K,J);
1045 %######
1048 NEXT J
1050 PRINT
1052 NEXT K
1055 PRINT
1058 PRINT
1060 PRINT TAB(35);"R(K,J) MATRIX, PERFORMANCE VALUE FOR TRAINING CYCLE
     BELOW        RCR"
1065 PRINT "       ";
1068 GOSUB 1200
1072 FOR K = 1  TO 10
1076 PRINTUSING "B=#.# ",K*.1;
1080 FOR J = 1  TO 20
1082 PRINTUSING 1085 , R(K,J);
1085 %##.###
1088 NEXT J
1092 PRINT
1095 NEXT K
1100 GOSUB 2000
1108 PRINT TAB(40); "U(K,J) MATRIX, NUMBER OF CYCLES OR TIME PERIODS TO
     REACH  RCR"
1110 PRINT "       ";
1115 GOSUB 1200
1118 FOR K = 1  TO 10
1120 PRINTUSING "B=#.# ",K*.1;
1122 FOR J= 1  TO 20
```

```
1123 PRINTUSING 1124 , U(K,J);
1124 %######
1126 NEXT J
1128 PRINT
1130 NEXT K
1132 PRINT
1133 PRINT
1135 PRINT TAB(40); "T2(K,J) OR TW MATRIX, TIME VALUE FOR FULL RE-TRAINI
     NG"
1138 PRINT
1140 PRINT
1144 GOSUB 1200
1146 FOR K = 1  TO 10
1148 PRINTUSING "B=#.# ",K*.1;
1150 FOR J = 1  TO 20
1152 PRINTUSING 1155 , T2(K,J);
1155 %######
1160 NEXT J
1162 PRINT
1165 NEXT K
1168 RETURN
1200      REM SUB-SUBROUTINE FOR PRINTING MATRIX FORMAT
1205 GOSUB 2000
1212 PRINT "       ";
1215 FOR J = 1  TO 20
1220 PRINTUSING "   ## ",J;
1225 NEXT J
1230 PRINT
1232 PRINT
1233 RETURN
1300      REM SUBROUTINE FOR PERFORMING SENSITIVITY ANALYSIS OF RV'S
1301 GOSUB 1600
1302      REM THIS SUBROUTINE WILL VARY A SPECIFIED RANDOM VARIABLE AND
      MAINTAIN          THE STATUS OF THE REMAINING RV'S FOR THE PREVIOU
      SLY GENERATED DATA BASE.
1305 INPUT " TO VARY RI, ENTER 1 ; TO VARY RF, ENTER 2 ; TO VARY ALPHA,
      ENTER 3       ; TO VARY TAU, ENTER 4 ; TO VARY RCR, ENTER 5; ",G
1308 PRINT " TO VARY RF, ENTER 1 ; TO VARY RI, ENTER 2 ; TO VARY ALPHA,
      ENTER 3       ; TO VARY TAU, ENTER 4 ; TO VARY R-CRITICAL, ENTER 5;
       ";G
1310 IF G=1  THEN 1350
1315 IF G=2  THEN 1400
1320 IF G=3  THEN 1450
1325 IF G=4  THEN 1500
      :IF G=5  THEN 1550
1326 IF G>5  THEN  139
1327 IF G<=0  THEN  139
1350      REM SENSITIVITY ANALYSIS FOR RI
1351 GOSUB 2000
1354 PRINT "SENSITIVITY ANALYSIS FOR RANDOM VARIABLE RI"
1355 GOSUB 1600
```

```
1360 INPUT "ENTER FIRST VALUE FOR RI IN ANALYSIS ?",X
1362 INPUT " ENTER FINAL VALUE TO BE CONSIDERED FOR RI?",Y
     :INPUT "ENTER                   INCREMENT STEP FOR RI SENSITIVITY ANAL
     YSIS ",B1
1363 PRINT TAB(26);"FIRST VALUE OF RI = ";X ; TAB(50); "FINAL VALUE OF R
     I = ";Y       ; TAB(80); "INCREMENT STEP OF RI = ";B1
1364 FOR R3 = X  TO Y  STEP B1
1365 FOR J=1  TO 20
     :LET A(1,J)=R3
     :NEXT J
1366 GOSUB 2000
1367 PRINT TAB(35); " SENSITIVITY VALUE OF RI  = ";R3
     :GOSUB 2000
1368 GOSUB  800
1372 GOSUB 1860
1378 NEXT R3
1380 GOTO  140
1400      REM SENSITIVITY ANALYSIS FOR RF
1403 GOSUB 2000
1408 PRINT "SENSITIVITY ANALYSIS FOR RANDOM VARIABLE RF"
1410 GOSUB 1600
1415 INPUT "ENTER FIRST VALUE FOR RF IN ANALYSIS ?",X
1418 INPUT " ENTER FINAL VALUE TO BE CONSIDERED FOR RF?",Y
     :INPUT " ENTER                      INCREMENT STEP FOR RF SENSITIVITY
     ANALYSIS ",B2
1419 PRINT TAB(26);"FIRST VALUE OF RF = ";X ; TAB(50); "FINAL VALUE OF R
     F = ";Y       ; TAB(80); " INCREMENT STEP OF RF= ";B2
1420 FOR R3 = X  TO Y  STEP B2
1421 FOR J=1  TO 20
     :LET A(2,J)=R3
     :NEXT J
1422 GOSUB 2000
1423 PRINT TAB(35); " SENSITIVITY VALUE OF RF  = ";R3
     :GOSUB 2000
1424 GOSUB  800
1425 GOSUB 1860
1426 NEXT R3
1430 GOTO  140
1450      REM SENSITIVITY ANALYSIS FOR ALPHA
1453 GOSUB 2000
1458 PRINT "SENSITIVITY ANALYSIS FOR RANDOM VARIABLE ALPHA"
1460 GOSUB 1600
1465 INPUT "ENTER FIRST VALUE FOR ALPHA IN ANALYSIS ?",X
1468 INPUT " ENTER FINAL VALUE TO BE CONSIDERED FOR ALPHA?",Y
     :INPUT " ENTER                    INCREMENT STEP FOR ALPHA SENSITIVITY
/     ANALYSIS ",B3
1469 PRINT TAB(26);"FIRST VALUE OF ALPHA= ";X ; TAB(50); "FINAL VALUE OF
      ALPHA = ";Y; TAB(80); " INCREMENT STEP OF ALPHA = ";B3
1471 FOR R3 = X  TO Y  STEP B3
1472 FOR J= 1  TO 20
     :LET A(3,J)=R3
```

```
     :NEXT J
     :GOSUB 2000
1473 PRINT TAB(35); " SENSITIVITY VALUE OF ALPHA  = ";R3
     :GOSUB 2000
1474 GOSUB  800
1478 GOSUB 1860
1480 NEXT R3
1486 GOTO  140
1500      REM SENSITIVITY ANALYSIS FOR TAU
1503 GOSUB 2000
1508 PRINT "SENSITIVITY ANALYSIS FOR RANDOM VARIABLE TAU"
1510 GOSUB 1600
1515 INPUT "ENTER FIRST VALUE FOR TAU IN ANALYSIS ?",X
1518 INPUT " ENTER FINAL VALUE TO BE CONSIDERED FOR TAU?",Y
     :INPUT " ENTER                  INCREMENT STEP FOR TAU SENSITIVITY
      ANALYSIS ",B4
1519 PRINT TAB(26);"FIRST VALUE OF TAU = ";X ; TAB(50); "FINAL VALUE OF
     TAU = "      ;Y; TAB(80); " INCREMENT STEP OF TAU = ";B4
1520 FOR R3 = X  TO Y  STEP B4
1521 FOR J=1  TO 20
     :LET A(4,J)=R3
     :NEXT J
1522 GOSUB 2000
1523 PRINT TAB(35); " SENSITIVITY VALUE OF TAU  = ";R3
     :GOSUB 2000
1524 FOR K=1  TO 10
     :FOR J=1  TO 20
     :T3(K,J)=R3*(K*.1)
     :NEXT J
     :NEXT K
1525 GOSUB  800
1526 GOSUB 1860
1527 NEXT R3
1528  FOR K=1  TO 10
     :FOR J=1  TO 10
     :T3(K,J)=S8(K,J)
     :NEXT J
     :NEXT K
1530 GOTO  140
1550      REM SENSITIVITY ANALYSIS OF R-CRITICAL
1553 GOSUB 2000
1558 PRINT "SENSITIVITY ANALYSIS FOR RANDOM VARIABLE RCR"
1560 GOSUB 1600
1565 INPUT "ENTER FIRST VALUE FOR RCR IN ANALYSIS ?",X
1568 INPUT " ENTER FINAL VALUE TO BE CONSIDERED FOR RCR?",Y
     :INPUT "ENTER                  INCREMENT STEP VALUE FOR RCR SENS
     ITIVITY ANALYSIS ",B5
1569 PRINT TAB(26);"FIRST VALUE OF RCR = ";X ; TAB(50); "FINAL VALUE OF
     RCR = ";Y; TAB(80); " INCREMENT STEP OF RCR = ";B5
1570 FOR R3 = X  TO Y  STEP B5
1571 LET R1 = R3
```

PROGRAM LISTING

```
1572 GOSUB 2000
1573 PRINT TAB(35); " SENSITIVITY VALUE OF RCR  = ";R3
     :GOSUB 2000
1574 GOSUB  800
1575 GOSUB 1860
1576 NEXT R3
1578 GOTO  140
1598 RETURN
1600      REM SUBROUTINE FOR RE-INITIALIZING RANDOM VARIABLE DATA MATRIX
      FOR              SENSITIVITY ANALYSIS
1602 FOR I=1  TO 4
1605 FOR J=1  TO 20
1608 LET A(I,J) = S(I,J)
1610 NEXT J
1612 NEXT I
1616 RETURN
1860      REM SUBROUTINE FOR CALCULATING SAMPLE MEAN AND SAMPLE STD. DEV
     . OF TW OR         T2(K,J)
1861      REM THE NUMBER OF SAMPLE POINTS FOR CALCULATING THE MEAN WILL
     BE EQUAL TO C
1862 FOR K=1  TO 10
1863 LET C=0
1864 LET M = 0
1867 FOR J=1  TO 20
1868 IF T2(K,J)>= (100*T3(K,J))  THEN  1885
1870 IF T2(K,J) = 0  THEN 1885
1872 LET M = M + T2(K,J)
1878 LET C = C+1
1885 NEXT J
1887 LET M2(K) = M/C
1894 NEXT K
1900      REM ROUTINE FOR CALCULATING SAMPLED STANDARD DEVIATION
1901 FOR K=1  TO 10
1902 LET S = 0
1903 LET C=0
1905 FOR J=1  TO 20
1908 IF T2(K,J)>= (100*T3(K,J))  THEN  1920
1909 IF T2(K,J) = 0  THEN 1920
1914 LET C = C+1
1915 LET S =   S + (T2(K,J)-M2(K))↑2
1920 NEXT J
1921 LET S3(K) = (S/C)↑0.5
1928 NEXT K
1929      REM PRINTING ROUTINE FOR MEAN AND STD.DEV.
1930 PRINT TAB(40); "SAMPLE MEAN FOR EACH BETA"
1932 GOSUB 2000
1936 PRINT "      ";
1938 FOR K=1  TO 10
1940 PRINTUSING "   ## ",K;
1944 NEXT K
1946 GOSUB 2000
```

PROGRAM LISTING

```
1948 PRINTUSING "MEAN= ";
1950 FOR K=1  TO 10
1953 PRINTUSING 1956 ,M2(K);
1956 %####.#
1958 NEXT K
1960 GOSUB 2000
1964 PRINT TAB(40); "SAMPLE STD. DEV. FOR EACH BETA"
1968 GOSUB 2000
1970 PRINT "      ";
1972 FOR K=1  TO 10
1973 PRINTUSING "   ## ",K;
1976 NEXT K
1978 GOSUB 2000
1980 PRINTUSING " SD = ";
1982 FOR K=1  TO 10
1984 PRINTUSING 1986 ,S3(K);
1986 %####.#
1988 NEXT K
1990 GOSUB 2000
1999 RETURN
2000      REM SUBROUTINE PRINT
2001 PRINT
2003 PRINT
2004 RETURN
2200      REM SUBROUTINE FOR CALCULATING SIMPSON NUMERICAL VALUE FOR NOR
       MAL                   DISTRIBUTION
2202      REM NORMAL DISTRIBUTION FUNCTION F(Z) = (1/((2*#PI)↑0.5))* EXP
       (-(Z)↑2)/2)
2208 LET F1 = 1/((2*#PI)↑0.5)
2210 FOR I = 1  TO 4
2211 C=0
2212 FOR H1 = .5  TO 2.5
2215 LET Z1 =0
2218 C=C+1
2220 LET Z1=0
       :FOR X1=1  TO (N1-1)  STEP 2
       :LET Z1=Z1 + (4*(F1*EXP(-((X1*H1/       N1)↑2))/2))
2222 NEXT X1
2224 LET Z2=0
       :FOR X2=2  TO (N1-2)  STEP 2
       :LET Z2=Z2 + (2*(F1*EXP(-((X2*H1/       N1)↑2))/2))
2226 NEXT X2
2228 LET H3(I,C)= ((H1/(3*N1))*(F1+Z1+Z2+(F1*EXP(-(H1↑2)/2))))
2231 NEXT H1
2233 NEXT I
2236 FOR I = 1  TO 4
2237 FOR L= 1  TO 7
2238 IF L=1 OR L=7  THEN 2242
2240 GOTO 2243
2242 LET F3(I,L) = .5 - H3(I,3)
2243 IF L=2 OR L=6  THEN 2248
```

PROGRAM LISTING

```
2244 GOTO 2250
2248 LET F3(I,L) = H3(I,3) - H3(I,2)
2250 IF L=3 OR L=5  THEN 2254
2252 GOTO 2256
2254 LET F3(I,L) = H3(I,2) - H3(I,1)
2256 IF L=4  THEN 2260
2258 GOTO 2262
2260 F3(I,L) = 2*H3(I,1)
2262 NEXT L
2264 NEXT I
2268 RETURN
2800      REM SUBROUTINE FOR CHI SQUARE ANALYSIS OF RESULTS OR TW MATRIX
2815      REM DETERMINATION OF CHI SQUARE ANALYSIS
2818 FOR I = 1  TO 4
2820 LET C3=0
2822 FOR J3 = -2.5  TO 2.5  STEP 1
2844 LET C3 = C3 + 1
2846 LET B3(I,C3) = M(I) + (J3*S5(I))
2848 NEXT J3
2850 NEXT I
2860      REM DETERMINATION OF DISTRIBUTION INTERVALS
2861      REM INITIALIZE FREQUENCY COUNTER AND CHI STATISTIC MATRIX
2862 FOR I = 1  TO 4
2863 FOR N3 = 1  TO 7
2864 F(I,N3) = 0
2866 NEXT N3
2867 NEXT I
2868 FOR I = 1  TO 4
2872 FOR J = 1  TO 20
2874 FOR N3 = 1  TO 6
2880 IF A(I,J) <= B3(I,N3)  THEN 2886
2881 IF A(I,J) > B3(I,6)  THEN 2883
2882 GOTO 2885
2883 LET N3 = 7
      :GOTO 2886
2885 NEXT N3
2886 F(I,N3)= F(I,N3)+ 1
2888 NEXT J
2889 NEXT I
2890  GOSUB 3300
      :GOTO 2898
2891 FOR I=1  TO 4
2892 LET F3(I,1)=.0062
      :F3(I,2)=.0606
      :F3(I,3)=.2417
      :F3(I,4)=.3830
      :F3(I,5)=      .2417
      :F3(I,6)=.0606
      :F3(I,7)=.0062
2894 NEXT I
2895 GOTO 2898
```

PROGRAM LISTING

```
2896 INPUT "NUMBER OF SIMPSON INTERVALS N1 = ",N1
2897 GOSUB 2200
2898 GOSUB 3100
2899 FOR I = 1  TO 4
2900 C1(I) = 0
2904 FOR L = 1  TO 7
2912 C1(I) = C1(I) +     (F(I,L)-20*F3(I,L))↑2/(20*F3(I,L))
2915 NEXT L
2918 NEXT I
2930      REM PRINTING ROUTINES  FOR CHI SQUARE ANALYSIS
2933 GOSUB 2000
2935 PRINT TAB(35); " CHI SQUARE ANALYSIS"
2938 PRINT
2940 PRINT TAB(35); "FREQUENCY OF DATA IN INTERVALS FOR RV MATRIX"
2942 PRINT
2944 PRINT "      ";
2948 FOR L=1  TO 7
2950 PRINTUSING "   ## ",L;
2952 NEXT L
2954 GOSUB 2000
2956 FOR I = 1  TO 4
2960 GOSUB  900
2962 FOR N = 1  TO 7
2964 PRINTUSING 2968 , F(I,N);
2968 %######
2972 NEXT N
2974 GOSUB 2000
2976 NEXT I
2980 GOSUB 2000
2982 PRINT TAB(30); "CHI SQUARE STATISTIC REFERENCE NUMBER C1(I)"
2983 PRINT
2984 GOTO 3000
2985 PRINTUSING "      ";
2986 FOR L=1  TO 7
2988 PRINTUSING "   ## ";L
2989 NEXT L
2990 GOSUB 2000
2991 FOR I= 1  TO 4
2992 GOSUB 2000
2993 FOR L= 1  TO 7
2995 %##.###
2996 NEXT L
2997 GOSUB 2000
2998 NEXT I
2999 RETURN
3000 FOR I =1  TO 4
3003 PRINT TAB(30); " CHI SQUARE VALUE  C1(";I;") =   ";C1(I)
3006 PRINT
3008 NEXT I
3010 RETURN
3100      REM SUBROUTINE FOR PRINTING INTERVAL VALUE MATRIX B3(I,L), AND
```

```
                              AREA PROB. VALUE FOR INTERVAL, F3(I,L)
3118 PRINT TAB(30); " INTERVAL VALUE MATRIX B3(I,L)"
3119 GOSUB 2000
3120 PRINT "          ";
3125 FOR L1=1  TO 6
3130 PRINTUSING "    ##  ",L1;
3133 NEXT L1
3135 GOSUB 2000
3140 FOR I= 1  TO 4
3142 GOSUB  900
3144 FOR L=1  TO 6
3160 PRINTUSING 3162 , B3(I,L);
3162 %########
3164 NEXT L
3168 GOSUB 2000
3170 NEXT I
3180 GOSUB 2000
3215 PRINT TAB(30);"AREA PROB. VALUE MATRIX F3(I,L)"
     :GOSUB 2000
3219 GOSUB 2000
3220 PRINT "          ";
3225 FOR L1=1  TO 7
3230 PRINTUSING "    ##  ",L1;
3233 NEXT L1
3235 GOSUB 2000
3240 FOR I= 1  TO 4
3242 GOSUB  900
3244 FOR L=1  TO 7
3252 PRINTUSING 3253 , F3(I,L);
3253 %###.####
3264 NEXT L
3268 GOSUB 2000
3270 NEXT I
3274 RETURN
3300     REM SUBROUTINE FOR CALCULATING NUMERICAL VALUE FOR NORMAL DIST
     RIBUTION
3302     REM NORMAL DISTRIBUTION FUNCTION F(Z) = (1/((2*#PI)↑0.5))* EXP
     (-(Z)↑2)/2)
3304 FOR I= 1  TO 4
3305 LET C=0
3306 FOR X8 = .5  TO 2.5
3310 C=C+1
3312 R8(C)= EXP(-X8↑2/2)/2.5066282746
3314 T(C)= 1/(1+.33267*ABS(X8))
3316 H3(I,C) = .5-R8(C)*(.4361836*T(C)↑2 + .937298*T(C)↑3)
3318 NEXT X8
3320 NEXT I
3336 FOR I = 1  TO 4
3337 FOR L= 1  TO 7
3338 IF L=1 OR L=7  THEN 3342
3340 GOTO 3343
```

PROGRAM LISTING

```
3342 LET F3(I,L) = .5 - H3(I,3)
3343 IF L=2 OR L=6  THEN 3348
3344 GOTO 3350
3348 LET F3(I,L) = H3(I,3) - H3(I,2)
3350 IF L=3 OR L=5  THEN 3354
3352 GOTO 3356
3354 LET F3(I,L) = H3(I,2) - H3(I,1)
3356 IF L=4  THEN 3360
3358 GOTO 3362
3360 F3(I,L) = 2*H3(I,1)
3362 NEXT L
3364 NEXT I
3368 RETURN
4000 END
4113 SELECT PRINT 215(132)
4890 GOTO 2896
```

REFERENCES

1. Klee, Harold and Sepulveda, Jose A. "Use of Low-Cost Part Task Trainers to Lower Demand on Training Simulators," University of Central Florida, Orlando, FL, 1982.