THE ECONOMIC DESIGN OF DYNAMIC \bar{X} -CONTROL CHARTS

Ву

BEHROOZ PARKHIDEH

Bachelor of Industrial Engineering Iran College of Science and Technology Tehran, Iran 1977

> Master of Science Oklahoma State University Stillwater, Oklahoma 1980

Submitted to the Faculty of the Graduate College of the Oklahoma State University in partial fulfillment of the requirements for the Degree of DOCTOR OF PHILOSOPHY May, 1984

Thesis 19840 P246e cop.2



THE ECONOMIC DESIGN OF DYNAMIC \bar{X} -CONTROL CHARTS

Kenneth E Case Thesis Adviser	
(Valente Toelles	
M. Palmin Tunell	
Don Holbert	
Marman Markan	
'Dean of the Graduate College	

Thesis Approved:

PREFACE

This research is concerned with control charting which comprises an important part of the statistical quality control. The purpose of this research is to originate a dynamic control charting approach, in which the control chart parameters are varying over time, in order to best design an \bar{X} -control chart having a generalized process failure mechanism.

A generalized dynamic model for the \bar{X} -control chart is developed. A special control chart methodology is introduced and incorporated into this model along with a Weibull distribution employed to represent the process failure mechanism. An optimization procedure is employed to economically design the parameters of this dynamic control chart. The dynamic chart designs are then compared with Duncan's \bar{X} -chart, for the situation in which the true process failure mechanism is given by a Weibull distribution

I wish to express my special appreciation to my major advisor and the chairman of my Ph.D. committee, Dr. Kenneth E. Case, for his constant encouragement, guidance, and assistance throughout this research and during my master and doctoral program. Dr. Case's high standard of excellency in academic areas and in leadership and his warm and outstanding personality has benefited me and many other students at this school. They have also influenced my philosophy both professionally and personally. His belief in my abilities has led me to have confidence in myself. I also appreciate his personal concern for my

career although in no way can his valuable advice, guidance, and care be adequately recognized.

I extend my sincere thanks to my committee members Dr. Donald Holbert, Dr. C. Patrick Koelling, and Dr. M. Palmer Terrell, Head of the School of Industrial Engineering and Management, for their interest and assistance during this work and my stay at Oklahoma State University. I am also grateful to Dr. Lyle D. Broemeling, currently on leave, and Dr. Joe H. Mize, currently at Arizona State University, who have served as members of my committee before leaving Oklahoma State University.

Thanks and acknowledgment must be given to the faculty and Head of the School of Industrial Engineering and Management who have committed themselves to keep an excellent academic and research standard which has provided national recognition and reputation for Oklahoma State University. Their efforts have also produced a challenging, but inspiring and very rewarding environment for me and other students at this university.

Mrs. Marcia Hickman is to be thanked for her virtually faultless typing and suggestions concerning this dissertation.

Also thanks to my parents who always encouraged me in my academic endeavors. To my mother, sister, and brothers I wish to express my heartfelt thanks for many ways in which they provided support.

TABLE OF CONTENTS

Chapter	Page
I. THE RESEARCH PROBLEM	. 1
Purpose	. 1 . 2 . 3 . 4
Introduction	• 2
Concept, Background and Importance	. 3
Statistically Based_X-Control Chart	. 4
Economically Based X-Control Chart	. 6
Dynamic X-Control Chart	. 7
Process Failure Mechanisms and	
X-Control Chart	. 9
Background and Importance	. 9
General Process Failure Mechanism	
Dynamic \bar{X} -Control Chart Introduction and Importance	. 12
Introduction and Importance	. 12
Concept and Contribution	. 13
Summary of Research Objectives	. 14
II. LITERATURE REVIEW	. 16
Introduction	. 16
Statistical Quality Control and	1/
Control Charts	. 16
Economic Modeling and Optimization	. 17
of Control Charts	. 17
Background	• 1/
X-Control Chart	. 19
Conclusions	
Process Failure Mechanism and	•
Control Charts	. 20
Dynamic X-Control Chart	•
Summary	•
Jummury	•
III. ECONOMIC DESIGN OF A DYNAMIC X-CONTROL CHART WITH A GENERALIZED PROCESS	
FAILURE MECHANISM	. 24
Introduction	. 24
Assumptions	•
Notation	27

Chapter	Page
Approach to Model Formulation	29
Model Components and Cycle Time	29
Dynamic \bar{X} -Control Chart Operation	30
Economic Model Formulation	32
Some Probability Definitions	32
Average In-Control, Out-of-Control,	
and Cycle Times	34
ATOWIN; Expression	41
Expected Number of False Alarms	44
Cost of Looking for False Alarms	45
Cost of Finding the Assignable Cause	45
Cost of Sampling and Inspection	45
Cost Formulation	50
Summary	52
IV. ECONOMIC OPTIMIZATION OF A DYNAMIC X-CONTROL CHART; ECONOMIC COMPARISON WITH DUNCAN'S	
X-CHART	53
Introduction	53
Notation	55
A Dynamic \bar{X} -Control Chart Methodology	57
Model Implementation; Computational	3,
Considerations	60
Summations of Series Involving $P(OOC_i)$	60
Summations Involving Products of Q's	63
	64
Simultaneous Restrictions on n _i and k _i Economic Optimization of the Dynamic	
X-Control Chart	67
General Strategy	67
Techniques Used in Optimization Algorithm	68
Optimization Search Algorithm	69
Testing the Search Algorithm	74
Finding a "Good" Starting Point	76
Duncan's Model Optimization	77
Economic Comparisons Between the Dynamic	• •
\bar{X} -Control Chart and Duncan's \bar{X} -Control Chart	79
Examples for Comparison	79
Analysis of Examples	81
Summary	85
V. USING THE INTERACTIVE COMPUTER PROGRAM	87
Introduction	87
Overview	87
Getting Started	88
Economic Design of the Dynamic \bar{X} -Chart	89
Economic Evaluation of the Dynamic X-Chart	92
Economic Design of Duncan's X-Chart	98
Economic Evaluation of Duncan's X-Chart	99
Summary	102

Chapter	Page
VI. SUMMARY AND CONCLUSION	103
BIBLIOGRAPHY	105
APPENDICES	111
APPENDIX A - Duncan's Model as Special Case of the Dynamic Model	112
APPENDIX B - Fortran Program Listing	120

LIST OF TABLES

Table		Page
4.1.	Loss-Cost Values for Exponential Process Failure Mechanism	62
4.2.	P Values for Duncan's Examples	65
4.3.	Examples Chosen for Economic Comparison	80
4.4.	Optimal Economic Designs of Duncan's and the Dynamic \bar{X} -Chart and Their Comparisons	82

LIST OF FIGURES

Figur	e	Page
1.1.	Statistically Based \bar{X} -Control Chart	5
1.2.	Economically Based \bar{X} -Control Chart	8
1.3.	Dynamic \bar{X} -Control Chart	10
3.1.	Cycle Time	31
3.2.	Out-of-Control Time Before the Detecting Sample is Plotted for Case (1)	36
3.3.	Out-of-Control Time Before the Detecting Sample is Plotted for Case (2)	38
3.4.	Out-of-Control Time Before the Detecting Sample is Plotted for Case (3)	39
3.5.	Average Time of Occurrence of Assignable Cause Within the i th Interval	42
3.6.	Cycle Length for Case (1)	47

CHAPTER I

THE RESEARCH PROBLEM

Purpose

In recent years, the scope and importance of quality control in business and industry has increased rapidly. Statistical quality control, an important set of quality tools, contains some of the best recognized quantitative techniques for improving productivity. One of the major areas of statistical quality control is process control, in which control charts are employed for analyzing process capability and for establishment and maintenance of statistical control of the process. The most famous and widely used control chart in industry is the \bar{X} -control chart, based upon statistical as well as economic design principles [64].

This research extends the state of the art in process control charting by:

- 1. defining and developing an economically based dynamic \bar{X} -control chart in which sample size, control limit width, and interval between samples are dynamic.
- 2. employing this new methodology to model, investigate, and compensate for the effects of different process failure mechanisms on the operation of \bar{X} -control charts (the exponential time to failure mechanism is by far the most popular distribution employed by researchers to date).

Introduction

In recent years, the scope and importance of quality control in business and industry has broadened as never before. Today, a company's reputation depends primarily on its ability to deliver a product of satisfactory quality to its customers. In fact, industrial leaders are now emphasizing the importance of quality in successful operation of a company in today's competitive market [37].

One of the factors that contributes to this focused attention on quality is a growing awareness of the needs and demands of the customers. This trend, which might be called consumerism, acknowledges the importance of customer satisfaction and recognizes that the consumer should expect to purchase safe, reliable products at fair prices [37]. This concept has been further supported by the creation of the Consumer Product Safety Commission (1972).

The field of quality can be divided into several areas, one of which is statistical quality control. Statistical quality control techniques can be used to achieve the quality objectives with the least cost possible.

An important part of statistical quality control is control charting. Control charts are used for one or more of the following purposes:

- 1. to bring a process under control,
- 2. to help establish process capability,
- to maintain control of a process.

This research will concentrate on the latter purpose. Some of the more popular control charts used to maintain current control of a process include:

1. \bar{X} -Chart (Sample mean control chart),

- 2. R-Chart (Sample range control chart),
- 3. P-Chart (Percent defective control chart),
- 4. CuSum Chart (Cumulative Sum control chart),
- 5. Moving Average Chart,
- 6. Median and Midrange Charts,
- 7. T^2 -Chart (Multivariate average control chart). This research is concerned with only the first of these.

X-Control Chart

Concept, Background and Importance

The theory of control charts was formally introduced by Walter Shewhart [70]. This theory is based on a differentiation of the cause of variation in quality. One source of variation called chance (inherent) variation is the sum of the effects of the whole complex of chance causes about which little can be done [22]. The other source of variation called "assignable causes" produces relatively large variations that are attributable to special causes such as differences between operators, equipment, and materials. Chance (inherent) variations behave in a random manner and follow statistical laws. Large variations due to assignable causes exhibit classic nonrandom behavior. Therefore, it is possible to detect assignable cause variations using statistical procedures. Control charts provide such a statistical vehicle.

Among many different control charts and procedures developed for monitoring of a process, the $\bar{\lambda}$ -control chart for averages is the most widely used technique [35]. A scientific survey of many firms in the United States in 1976 shows that the use of $\bar{\lambda}$ -control charts dominates

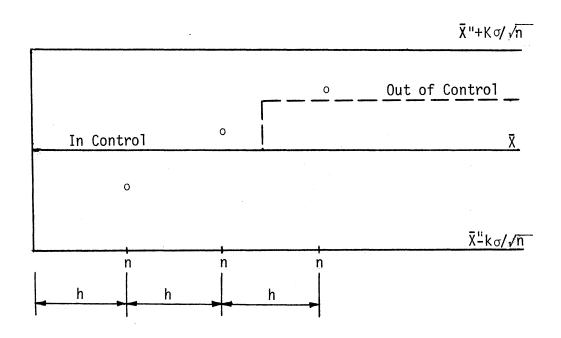
the use of any other control chart techniques in practice [64]. More recent encouragement from such notable consultants as J. M. Juran and W. E. Deming have further increased their use. Summing up the previous and current trends in the theoretical development and application of \bar{X} -control chart indicates that in the future the \bar{X} -chart will continue to receive further attention because of its fundamental importance in scientific quality control [31].

Statistically Based X-Control Chart

Traditionally \bar{X} -control charts are designed statistically. This concept was introduced by Shewhart [70] who suggested that samples of size n=4 or 5 be taken at intervals of h hours and the samples averages be plotted on a chart with control limits $k \circ \bar{\chi}$ above and below the mean such as in Figure 1.1. If a sample average falls outside the control limits, an action should be taken to find the assignable cause.

The control limits commonly used in the United States are .00135 probability limits or set at k=3 standard deviations of the sample average $(\pm 3\sigma_X)$. A .00135 probability limit implies that if chance (inherent) causes alone are at work, a point will fall above the upper limit with a .135% probability. Also, the probability of a point falling below the lower limit is only .135%. That is, the chance of a point falling outside the control limits, when the process is in control, is very small--less than three out of a thousand. Therefore, if a point falls outside these control limits, it can almost assuredly be said that the variation is produced by an assignable cause.

In general, any multiple of sigma other than the usual 3-sigma can be used to establish the control limits. This choice depends upon the



n = Constant Sample Size of 4 or 5.

h = Constant Sampling Interval. k = Constant Control Limit Spread of 3.

Figure 1.1 Statistically Based \bar{X} -Control Chart

risk that management of the quality function is willing to tolerate; tighter control limits achieved using a smaller multiple of sigma will increase the probability of concluding the process is out of control when it is really in control.

It is also noted that under this traditional statistical \bar{X} -control chart design, the value of the interval between samples, h, is left to be specified using some rule of thumb.

In summary, the introduction of the statistical design of the \bar{X} -control chart sets a scientific basis for the design and application of process control techniques. However, it fails to provide the practitioner with anything more than qualitative, rather than quantitative, guidelines for deciding the value of the interval between samples (h). More importantly, the use of suggested values of sample size of n=4 or 5, and the usual multiple of sigma, k=3, might well result in a control plan which is far from optimum in a cost sense.

Economically Based X-Control Chart

The design and operation of a control chart has economic consequences. The cost of sampling and testing, the cost of searching for assignable cause signals and possibly correcting them, and the cost of producing defective products are all affected by the selection of the control chart parameters—n, h, and k [50]. Therefore, it is logical to design control charts based upon an economic measure of performance.

In 1956, Duncan [20] formulated an economic model of an \bar{X} -control chart based on the maximum income criterion. This maximum income criterion is a natural one to consider since it relates to the financial aspects of operating a business. Since the publication of Duncan's

paper, many different formulations of the economic design of control charts have appeared. His assumptions and approach have proved to be most practical and appealing, and his work has become a classic in the field.

Duncan assumes that the process starts in-control and is subject to assignable causes which occur at random and shift the process mean to an out-of-control state. It is assumed that the transition between incontrol and out-of-control states is instantaneous.* Furthermore, Duncan assumes that the time from the start of the process in-control until it goes out-of-control follows an exponential distribution. This provides considerable simplification in the formulation of the cost model.

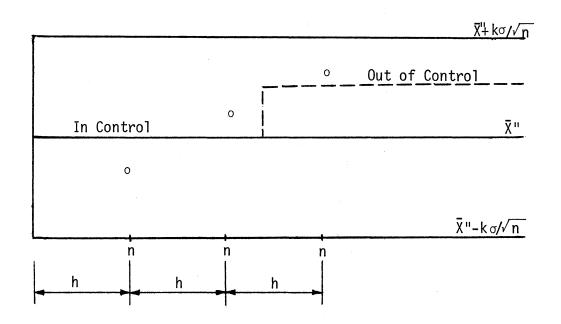
Duncan [20] applies formal optimization methodology to the economic cost model in determining the control chart parameters n, h, and k, which result in the optimum net income per unit of time. The economically-based control chart is illustrated in Figure 1.2. Note the similarity to the statistically-based chart shown in Figure 1.1.

Dynamic** \bar{X} -Control Chart

In almost all formulations of economically based \bar{X} -control charts, as well as economic design of other control charts, it has been assumed that the control chart parameters n, h, and k are fixed throughout the

^{*}Processes that "drift" slowly from an in-control state, such as in the case of tool wear, is not the subject of this research.

^{**&}quot;Dynamic" in conjunction with the \bar{X} -control chart, is a term used for the first time in this research. The word "dynamic" is chosen to indicate the varying (dynamic) nature of any or all of the control chart parameters--n_i, h_i, and k_i--as functions of time.



 $\ensuremath{\mathsf{n}}$, $\ensuremath{\mathsf{h}}$, and $\ensuremath{\mathsf{k}}$ are constant. However, they are found by minimization of the cost function.

Figure 1.2. Economically Based \bar{X} -Control Chart

operation of the chart. In fact, this practice has been so common that after reading the definitions of control charts in current books and journal papers, it is difficult to perceive a control chart in which sample sizes (n_i) , sampling intervals (h_i) , and/or control limit widths (k_i) are changing throughout the operation of the control chart.

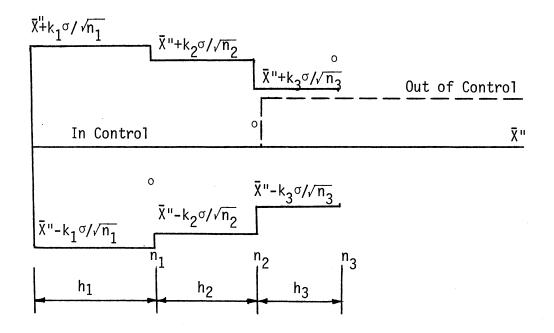
A new control chart methodology in which the control chart parameters n_i , h_i , and k_i are dynamic might be needed in the optimal design of control chart models which better reflect reality. That is, if some of the simplistic assumptions used in the classical economic design of $\bar{\chi}$ -control chart are changed to be more realistic, then the use of a dynamic control chart methodology might be necessary for correct modeling and optimization. For example, this is the case when the distributional assumption of time to process mean shift is changed from the exponential to a more generalized distribution.

The concept of a dynamic \bar{X} -control chart is illustrated in Figure 1.3. Note the difference between this chart and the state of the art control charts shown in Figures 1.1 and 1.2.

Process Failure Mechanisms and \bar{X} -Control Chart

Background and Importance

Certain assumptions about the behavior of the production process are required to formulate an economic model for the design of an \bar{X} -control chart. One required fundamental assumption is that pertaining to the mechanism governing the occurrence of the assignable causes which shift the process from an in-control state to an out-of-control state.



 $\mathbf{n_i},~\mathbf{h_i},~\mathrm{and}~\mathbf{k_i}$ are functions of time--dynamic-- and are found by minimization of a general cost function.

Figure 1.3. Dynamic \bar{X} -Control Chart

It is usually assumed that the assignable causes occur during an interval of time according to a Poisson process. That is, the length of time the process remains in-control before it shifts out-of-control is an exponential random variable. This assumption implies a Markovian process failure (shift) mechanism and allows considerable simplification in the cost model development.

It can be argued that, in the presentation of these models, it is not always so clear whether this Markov property results from insight into the physical nature of the production process or from preference for the mathematical convenience it provides [2]. Furthermore, as Baker [2] has suggested, the optimal economic control chart design is relatively sensitive to the choice of process failure mechanism. This is an important consideration because substantial cost penalties may occur in practice as the result of assuming a process failure mechanism in the economic model which is not compatible to the reality of the process.

General Process Failure Mechanism

The exponential distribution of time to shift is the most commonly used process failure (shift) mechanism in the economic design of the \bar{X} -control chart. In reliability engineering, the exponential distribution is referred to as the constant failure rate* (CFR) distribution because of its memoryless property. This property implies that the probability that a device (or a process) will not fail in a future time

^{*}Failure rate is the rate at which failures occur in a designated time interval.

interval, given that it has not failed until the present time, is independent of the length of the time it has been working in the past.

Physically, a CFR distribution represents the life distribution of many electronic components. Also the life distribution of a total system composed of many components that have different failure distributions may approach the exponential. In practice, however, there are many mechanical processes for which only an increasing failure rate (IFR) distribution is representative. For example, the Weibull distribution is widely used in reliability engineering [42] and well represents IFR mechanical systems; it can also be used to represent CFR (and even DFR--decreasing failure rate) situations. To avoid incorrect modeling, it is desirable to economically design an \bar{X} -control chart in which the failure process is governed by a more generalized distribution. To this end, the Weibull distribution is proposed rather than the exponential.

Dynamic X-Control Chart

Introduction and Importance

Fixed sample sizes and intervals between samples are used in the optimal economic designs of the \overline{X} -control chart. In Duncan's economic design of the \overline{X} -control chart, fixed sample sizes and intervals between samples are optimum because of his choice of the memoryless process shift (failure) mechanism. On the other hand, the use of varying sample sizes and sampling intervals, which is in fact necessary for non-Markovian processes, makes the mathematical modeling and optimal design of the \overline{X} -control chart a complicated task. Therefore, the trend of following Duncan's paper and the avoidance of mathematical complications

have resulted in the use of fixed sample sizes, sampling intervals, and control limit widths.

The use of varying sample sizes, sampling intervals, and control limit widths is indispensible to the optimum design of control charts in which process shift (failure) mechanism is non-Markovian. For example, when the failure rate of a production process increases over time, it might be more economical to reduce the interval between samples. Also, Taylor [72] considered the problem of minimizing the running and repair costs for a production process. Dynamic programming is used in his work to find the optimum sequence of time intervals for inspecting the produced items. He shows an example of a process for which the use of fixed inspection intervals is not optimum. Ignoring these facts can result in the design of uneconomical control charts. Thus, as suggested by Baker [2], if careless modeling is the price of convenience and acceptance, then the price may indeed be very high.

Concept and Contribution

The concept of a dynamic control chart is previously defined and is illustrated in Figure 1.3. In this new approach to the design of the \bar{X} -control chart, the sample size n_i , sampling interval h_i , and the control limit width k_i are dynamic over time. There is no documentation in the literature which considers such a general methodology to the optimum economic design of a control chart.

The concept of the dynamic control chart seems essential for the optimal economic design of \bar{X} -control charts having a non-Markovian process shift mechanism. Furthermore, the use of this new concept and methodology will provide a means for the thorough investigation of the

importance of the process failure mechanism assumption and its effect on the economic design of \bar{X} -control charts.

Summary of Research Objectives

Based on the above discussions, the primary objective of this research is stated as follows:

<u>Objective</u>:

To originate, develop, seek favorable solutions for, and investigate the effects of appropriate dynamic \bar{X} -control chart methodology under the non-Markovian process shift (failure) mechanism.

In order to accomplish this objective, several subobjectives must be met.

<u>Subobjectives:</u>

- 1. To originate and develop dynamic \bar{X} -control chart methodology in which sample sizes, intervals between samples, and/or control limit widths are dynamic; varying over time.
- 2. To formulate the generalized dynamic version of Duncan's economically-based \bar{X} -control chart model in which the process failure mechanism can be of any form while incorporating the dynamic \bar{X} -control chart methodology.
- 3. To develop a general strategy, together with a computer program, to select appropriate values of the decision variables n_i , h_i , and k_i for the economically based dynamic \bar{X} -control chart.

- 4. To investigate and summarize the effects of different process shift mechanisms on the operation of \bar{X} -control charts.
- 5. To economically compare the dynamic \bar{X} -control chart and Duncan's \bar{X} -control chart plans when the actual underlying process shift mechanism is not Markovian.

CHAPTER II

LITERATURE REVIEW

Introduction

This chapter reviews developments in the literature pertaining to the objectives of this research. Support for this research has been documented in Chapter I. This chapter elaborates on this support and presents other sources which discuss concepts and techniques relating to this study.

This chapter is divided into four areas.

- 1. Statistical quality control and control charts
- 2. Economic modeling and optimization of control charts
- 3. Process failure mechanism and control charts
- 4. Dynamic \bar{X} -control chart.

Statistical Quality Control and Control Charts

Shewhart [70] first introduced the concept of statistical quality control in 1931. The concept can be used in many different ways ranging from manufacturing of goods to delivery of services [37]. Accordingly, the use of statistical quality control has spread throughout the world. Duncan [22] states that almost all industrialized nations use statistical quality control.

Two major areas of statistical quality control are acceptance sampling and control charting. In control charting, important developments include [31]:

- 1. Shewhart Control Charts and Their Ramifications-- \overline{X} ; R; p; c; u; X chart
- 2. Modifications of Shewhart Control Charts--moving average and range; median and midrange; geometric moving average
 - 3. Cumulative Sum Control Charts
 - 4. Acceptance Control Charts
 - 5. Multi-Characteristic Control Charts--Hotelling T²; Q chart.

More recently, considerable attention is given to the economic design of these control charts. Because of its importance, the economic design of control charts is elaborated upon in the next section.

According to a 1976 scientific survey of 173 firms, representing all geographical areas of the United States, the most popular control chart in practice is the \bar{X} -control chart [64]. Further, the \bar{X} -control chart is recognized to be of fundamental importance in quality control. Gibra [31] states that the \bar{X} -control chart will continue to receive further attention in the future. For these reasons, the \bar{X} -control chart is a sound topic for further research.

Economic Modeling and Optimization of Control Charts

Background

Shewhart's original design of control charts is based on "empirical-economic" considerations. Naturally, there have been many situations for which control charting has been found to be uneconomical

[50]. As a result, several techniques have been proposed to improve economic performance of the chart.

^ Early remedies included alternatives of Shewhart's control method, such as the use of warning limits [56, 75], and/or runs tests [74, 52]. Another early concern over the Shewhart \bar{X} -control charts involved the assumption of normality. Burr [7] found that Shewhart's control chart design is quite robust relative to non-normality.

A pioneering theoretical work in the area of cost modeling of quality control systems is that of Girshick and Rubin [32]. Their results along with those of other researchers including Bather [5], Ross [63], Savage [65], and White [76] are primarily of theoretical interests and do not lead to simple process control rules. Most of these works along with those of Aroian et al. [1], Barish et al. [3], Cowden [17], and Weiler [73, 74] can be referred to as "semi-economic" [50] design procedures.

The "optimal economic" design of the \bar{X} -control chart is introduced by Duncan [20]. His paper is the first to deal with a fully economic design of a Shewhart-type control chart. Duncan considers the cost of taking and inspecting a sample, the cost of maintaining the control chart, the average cost of looking for an assignable cause when either none exists, or when it has occurred, and the cost per hour of producing defective items. The decision variables for Duncan's model are n, h, and k, as previously defined, and are found by maximizing the expected net income per hour of operation, or by minimizing the loss cost incurred.

Optimum Economic Design of the \bar{X} -Control Chart

Duncan's assumptions and approach have proved to be most practical and appealing [14]. Accordingly, his model for the \bar{X} -control chart has received much attention and has become a classic in the field.

Several authors have elaborated on optimization methods of Duncan's model. Goel et al. [33] propose a method to find the exact optimum of Duncan's model. This procedure is superior to Duncan's approximate optimization technique.

Several other models are developed in connection with the economic design of \bar{X} -control charts. Gibra [30] has developed an economic model of the \bar{X} -chart similar to Duncan's model. However, he assumes that the time required to take and inspect a sample, interpret the results, and to search for and eliminate the assignable cause is an Erlang random variable [50]. Gibra [29] has also developed the optimum economic design of \bar{X} -control charts associated with the situation when the mean of the quality characteristic exhibits a linear trend over time. This model would be suitable for processes involving tool wear [50].

Duncan [21] has developed an economic model of a situation in which there are multiple assignable causes rather than just one assignable cause. Direct search methods are used to find the optimum control chart parameters. Chiu [11], however, shows that some of the numerical results in Duncan's paper are wrong. Knappenberger et al. [43] have also proposed a model for the economic design of the \bar{X} -control chart when there are multiple assignable causes. In this paper, the expected cost per unit produced is optimized rather than the expected cost per unit time in [21].

It is noteworthy that both Duncan [21] and Knappenberger et al.

[43] report that a single assignable cause model, matching the true multiple assignable cause system in certain ways, produces very good results. Furthermore, Montgomery [50] states that sensitivity analyses of these economic models show that multiple assignable cause processes can usually be approximated well by an appropriately chosen single assignable cause model. These observations and the complexity of the multiple assignable cause models have contributed to the fact that these models have not received much attention in the literature.

Conclusions

Clearly, economic design of the \bar{X} -control chart is receiving much attention. Among many different economic models of the \bar{X} -control chart, Duncan's model [20] is practical and has received much attention. Furthermore, Duncan's work has stimulated much further work. That is, many researchers have developed economic designs of other control charts including the R chart, p chart, and CuSum chart by following Duncan's model and approach.

Process Failure Mechanism and Control Charts

Duncan [20] assumes that assignable causes occur during an interval of time according to a Poisson process. That is, the time to failure is an exponential random variable. This assumption allows considerable simplification in the development of the economic model. The nature of the occurrence of assignable causes is called the "process failure mechanism" [50].

Gibra [31] and Montgomery [50] consider the particular choice of the process failure mechanism a critical assumption. Baker [2] has proposed a simple process model that allows the effect of this assumption to be investigated. His illustrative models are simple discrete-time versions of Duncan's continuous time model. Specifically, Baker compares two models. The first model is a discrete-time analog of Duncan's model when the process failure mechanism has the memoryless property. Baker's second model allows the use of any discrete probability function to model the process failure mechanism.

For a specific choice of a non-Markovian process failure mechanism in the second model, smaller sample sizes and narrower control limits compared to the first model are outcomes of the optimization procedure. This is possible because the run length in control in the second model does not have the memoryless property and a false alarm can postpone a true shift. Baker [2] concludes that the optimal economic control chart design is relatively sensitive to the choice of process failure mechanism. Therefore, substantial cost penalties may be incurred if an incorrect process failure mechanism is assumed [50].

In a recent paper [49], Montgomery and Heikes investigate the robustness of the process failure mechanism assumption for the fraction defective (p) control chart. They consider simple discrete-time models similar to those of Baker [2]. They conclude that the choice of process failure mechanism is important and the incorrect specification of this property can result in significant cost penalties.

Dynamic X-Control Chart

In Duncan's economic formulation of the \bar{X} -control chart, as well as most other economically-based control charts, a memoryless process failure mechanism is assumed. For this specific assumption, the use of fixed sample sizes and fixed interval between samples are optimum.

On the other hand, in order to develop and correctly optimize an economic model of the \bar{X} -control chart in which the process failure mechanism does not have the memoryless property, fixed sample sizes, sampling frequency, and control limit spread should be avoided. For example, Taylor [72] considered the problem of minimizing the running and repair costs of a production process. He shows an example of a process for which the use of fixed intervals between inspections is not optimum. These observations have led to the origination and development of dynamic \bar{X} -control chart methodology in which control chart parameters (sample sizes, interval between samples, and control limit widths) are dynamic--varying over time. There is no documentation in the literature describing or using this new concept.

Summary

A literature survey of the problems, contributions, and needs related to the objectives of this research is presented. This survey demonstrates an increasing interest in the economic design of the \bar{X} -control chart. It is emphasized that the choice of the process failure mechanism used in the economically-based \bar{X} -control chart is a critical one. However, in most of the economic designs of the \bar{X} -control charts a Markovian process is employed to model the failure mechanism.

There is very little work done in the economic design of the \bar{X} -control charts having a non-Markovian or a general process failure mechanism.

This survey indicates that a need exists for the following:

- 1. To provide a generalized economically-based \bar{X} -control chart model in which different process failure mechanisms can be used.
- 2. To develop appropriate procedures for the optimum design of this generalized economically-based \bar{X} -control chart.
- 3. To investigate the effects of different process failure mechanism assumptions on the economic design of the \bar{X} -control chart.

CHAPTER III

ECONOMIC DESIGN OF A DYNAMIC X-CONTROL CHART WITH A GENERALIZED PROCESS FAILURE MECHANISM

Introduction

The purpose of this chapter is to develop an economic model of a dynamic \bar{X} -control chart that will optimize the design of \bar{X} -control charts when the underlying process failure mechanism is of a generalized type. The economic design of \bar{X} -control charts is introduced by Duncan [20]. The acceptance and popularity of Duncan's approach to cost modeling is presented in Chapter II.

The economic model developed in this research uses a cost structure which is similar to Duncan's "classic" \bar{X} -chart cost model but improves on the process failure mechanism assumption by employing a generalized distribution of time to failure (Duncan uses the memoryless exponential distribution to represent time to failure). This provides a model in which a choice can be made as to the distribution which best represents the process environment. A proof that Duncan's model [20] is a special case of the generalized model of this chapter is given in Appendix A.

Optimization of this generalized dynamic \bar{X} -control chart can make excellent use of a methodology in which sample sizes, intervals between samples, and control limit widths are allowed to vary over time. The actual optimization of the dynamic \bar{X} -control chart using this methodology is discussed in Chapter IV.

Assumptions

In order to develop the modeling of dynamic \bar{X} -control charts, the following assumptions are employed:

- 1. The \bar{X} -control chart is used to maintain the statistical control of a production process.
- 2. The production process is characterized by a single in-control state. That is, the in-control state corresponds to the mean of a measurable quality characteristic when no assignable cause is present.
- 3. The occurrence of an assignable cause shifts the process mean to a known value.
- 4. The process standard deviation is assumed to be known. The assignable cause does not affect the process standard deviation.
- 5. The shift in the process average is instantaneous. That is, the process does not drift slowly from the in-control state, such as is the case with tool wear.
- 6. The occurrence time for the assignable causes are independently, identically distributed random variables with a density function f(t), t > 0. Note that f(t) is not restricted to the exponential case, but can be of any form. For example, it can be a Weibull density function.
- 7. The process is not self correcting. That is, after an assignable cause has occurred, the process can only be brought back to the incontrol state by management intervention.
- 8. The process is not shut down while the search for the assignable cause is in progress.
- 9. Sampling is continued during the search for the assignable cause.

- 10. Sampling inspection is not subject to measurement error.
- 11. The rate of production is sufficiently high so that the possibility of a change in the process occurring during the time a sample is taken can be neglected.
- 12. Action will be taken when a sample point falls outside the control limits.
- 13. The cost of adjustment or repair (including possible shutting down of the process) and the cost of bringing the process back to a state of statistical control subsequent to the discovery of an assignable cause are not considered.
- 14. The time required to take, inspect, and chart a sample is proportional to the sample size.
- 15. The average time required to find an assignable cause is a constant value.
- 16. Sample sizes, intervals between samples, and control limit spreads are dynamic, thus being permitted to change over time.

Note that Duncan's use of the exponential time to failure is a special case of assumption number 6. Also, Duncan's use of constant sample sizes, constant interval between samples, and constant control limit width is a special case of assumption number 16. Other assumptions are either explicitly or implicitly employed in Duncan's economic \bar{X} -control chart model.

The special model formulation of this research makes it possible to easily change any or all of assumptions 8, 9, 14, and 15. This provides an opportunity to further investigate the effect of different assumptions on the cost model and/or to tailor the model to fit a specific process environment.

Notation

The following symbols are employed to facilitate model development and presentation:

- n_i number of individual measurements making up the ith sample; that is; the ith sample size.
- h; length of the ith interval; the interval between the (i-1)th and ith samples.
- $k_{\,i}$ a factor used in determining the width of the control limits on the \bar{x} -control chart corresponding to the i^{th} sample. It represents the number of i^{th} sample average standard deviations separating each control limit and the center line.
- t i the time from the start of the process in-control until the i th sample is taken; t i = $\sum_{j=1}^{h} h_j$.
- the scale parameter of a Weibull distribution. See also the definition for η .
- η the shape parameter of a Weibull distribution; density function $f\left(t\right)$ is Weibull if:

$$f(t) = \theta \eta (\theta t)^{\eta-1} e^{-(\theta t)^{\eta}}, t > 0.$$

- λ the rate of occurrence per hour of assignable causes when the process failure mechanism is governed by the exponential distribution; that is, θ = λ when η = 1.
- \bar{X} " standard or desired process mean.
- $\boldsymbol{\sigma}$ standard or true process standard deviation.
- δ magnitude of the out of control shift in the process mean in multiples of $\sigma.$ The shift is $\delta\sigma.$
- Φ Φ is the cumulative probability function of the standard normal distribution;

$$\Phi(X) = \int_{-\infty}^{X} \frac{e^{-\frac{Z^2}{2}}}{\sqrt{2\pi}} dZ.$$

P_j - probability of detecting a shift on the ith sample, when there is an assignable cause;

$$P_{i} = \Phi(-k_{i} - \delta \sqrt{n_{i}}) + 1 - \Phi(k_{i} - \delta \sqrt{n_{i}}).$$

- Q_i probability of failing to detect a shift on the ith sample, when there is an assignable cause; $Q_i = 1 P_i$.
- α_i probability of a false alarm on the ith sample when there is no assignable cause; $\alpha_i = 2 \Phi (-k_i)$.
- 00C; an abbreviation for out-of-control in the ith interval.
- $P(00C_1)$ probability that the process shifts to the out-of-control condition in the i^{th} interval.
- $\Gamma(a)$ Gamma integral; $\int_{0}^{\infty} Z^{a-1} e^{-Z} dZ$.
- $\gamma(a,x)$ the unnormalized incomplete Gamma integral; $\int_0^x Z^{a-1} e^{-Z} dZ$.
- the rate at which the average sampling, testing, and charting time for a sample increases with the sample size.
- D the average search time for an assignable cause.
- V_0 the hourly income from operation in the in-control condition.
- V₁ the hourly income from operation in the out-of-control condition.
- M the reduction in process hourly income due to the occurrence of the assignable cause; M = V_0 V_1 .
- T the average cost per occasion of looking for an assignable cause when no assignable cause exists.
- W the average cost per occasion of finding the assignable cause, when it exists.
- the cost per sample of sampling, testing, and charting that is fixed and independent of the sample size.
- the unit cost of sampling, testing, and charting that is related to the sample size. The relationship is assumed to be linear.
- ACT the average cycle time.
- AIC the average time for the occurrence of an assignable cause.

- ATOWIN $_{i}$ the average time of the occurrence of the shift within the ith interval, given that the shift has occurred in the ith interval.
- TOOC the time that the process is operating in the out-of-control condition before the detecting sample is plotted on the chart.
- the average time the process is operating in the out-of-control condition before the detecting sample is plotted on the chart; AOOC = E[TOOC].
- the proportion of time that the process is in-control.
- ENFALS the expected number of false alarms during the average cycle time.
- AHCS the average hourly cost of sampling, testing, and charting; C_3 .
- C₁ the average hourly cost of looking for false alarms.
- C₂ the average hourly cost of finding the assignable cause.
- the average hourly cost of sampling, testing, and charting;
 AHCS.
- the loss-cost per hour of operation. Minimizing L corresponds to maximizing the average net profit per hour of operation.

Approach to Model Formulation

Model Components and Cycle Time

The components of this model are (i) the cost of an out-of-control condition, (ii) the cost of false alarms, (iii) the cost of finding an assignable cause, and (iv) the cost of sampling and inspection.

One key element in these components is the average cycle time. Cycle time is defined to be the total time from which the process starts in an in-control condition, shifts to an out-of-control condition, the out-of-control condition is detected, and the assignable cause is found. That is, cycle time is composed of the time the process is in-control, the time the process is out-of-control before a detecting sample is

taken, the time to evaluate and chart that sample, and the average time taken to then find the assignable cause. Cycle time is illustrated in Figure 3.1.

When the average cycle time is determined, then the cost components can be converted to a "per hour of operation" basis. The sequence of production cycles with accumulation of costs over a cycle belong to a class of stochastic processes called renewal reward processes [50]. Ross [60] shows that for a renewal reward process the average time cost is given by the expected cost during a cycle divided by the expected cycle time.

Dynamic X-Control Chart Operation

A major task of model development in this research involves the generalized formulation of Duncan's cost model to allow the model to represent different process failure mechanisms. This is important since there are many processes for which only an Increasing Failure Rate (IFR) distribution can represent the time to failure. Decision variables selected using Duncan's model, which employs a Constant Failure Rate (CFR) distribution, can result in substantial cost penalties when used in an IFR environment.

In order to correctly optimize the general \bar{X} -control chart model, dynamic sample sizes--n_i, intervals between samples--h_i, and control limit widths--k_i, should be considered. That is, the first sample of size n₁ is taken after the process has been operationg for h₁ hours, while the control limits are set at \bar{X} " \pm k₁ $\frac{\sigma}{\sqrt{n_1}}$. This is followed by a second sample of size n₂ taken at time h₁ + h₂, while the control

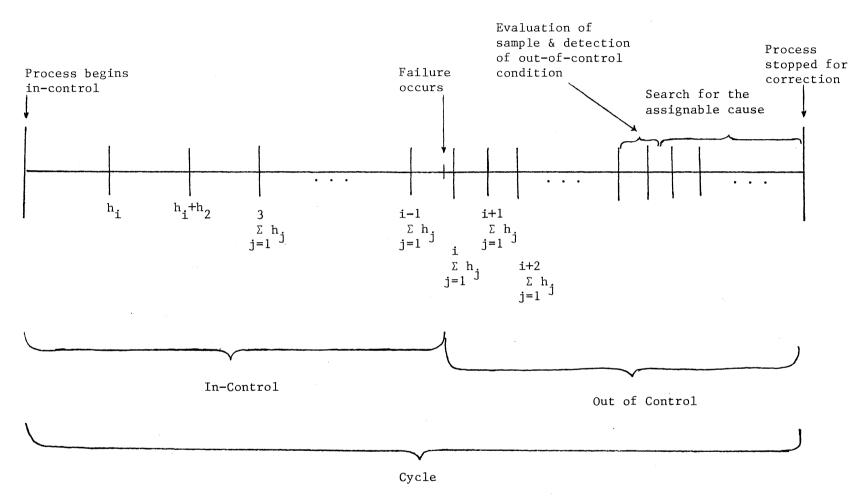


Figure 3.1. Cycle Time

limits, for plotting this sample, are set at $\bar{X}''\pm k_2\frac{\sigma}{\sqrt{n_2}}$, and so on. That is, in general, the ith sample of size n_i is taken at time $t_i = \sum\limits_{j=1}^i h_j$ and is plotted on a chart with control limits at

$$\bar{X}$$
"± $k_i \frac{\sigma}{\sqrt{n_i}}$.

No restriction has been set on the relationship between n_{i-1} and n_i , h_{i-1} and h_i , and k_{i-1} and k_i in the model formulation. However, for the purpose of optimizing the cost model, specific relationships are assumed between n_{i-1} and n_i , h_{i-1} and h_i , and k_{i-1} and k_i . The nature of this relationship is explained later in a discussion on dynamic \bar{X} -control chart methodology, in Chapter IV.

Economic Model Formulation

Some Probability Definitions

In the model formulation of this chapter, several probabilities are frequently used. These are (i) probability of detecting a shift on the i^{th} sample when there is an assignable cause, P_i , (ii) probability of failing to detect a shift on the i^{th} sample when there is an assignable cause, Q_i , (iii) probability of a false alarm on the i^{th} sample when there is no assignable cause, α_i , and (iv) probability of the process going out-of-control during the i^{th} interval, $P(OOC_i)$. The expressions for these probabilities are discussed below.

 P_i - Probability that the assignable cause will be detected on the i^{th} sample taken from the process, given that an assignable cause has occurred before the i^{th} sample. In accordance with

assumption number 12, P_i is the probability that the ith sample point falls outside the control limit, when the process mean has shifted from \bar{X}'' to $\bar{X}'' + \delta \sigma$. Therefore,

$$P_{i} = \int_{-\infty}^{-k_{i}-\delta \sqrt{n_{i}}} \frac{\frac{z^{2}}{2}}{\frac{e}{\sqrt{2\pi}}} dZ + \int_{k_{i}-\delta \sqrt{n_{i}}}^{\infty} \frac{\frac{z^{2}}{2}}{\sqrt{2\pi}} dZ$$

$$(3.1)$$

$$= \Phi(-k_{1} - \delta \sqrt{n_{1}}) + 1 - \Phi(k_{1} - \delta \sqrt{n_{1}})$$
 (3.2)

Q_i - Probability of not detecting a shift on the ith sample, when there is an assignable cause. Therefore,

$$Q_{i} = 1 - P_{i}$$
 (3.3)

 α_i - Probability of a false alarm on the ith sample. That is, the probability that the ith sample value falls outside the control limits, when the process mean is in-control. Therefore,

$$\alpha_{i} = 2 \int_{k_{i}}^{\infty} \frac{e^{-\frac{z^{2}}{2}}}{\sqrt{2\pi}} dZ \qquad (3.4)$$

=
$$2 \Phi (-k_i)$$
 (3.5)

 $P(00C_i)$ - Probability that the process shifts to the out-of-control condition during the i^{th} interval, that is during interval t_{i-1} to t_i . If the time to failure is distributed as f(t), t > 0, then,

$$P(00C_i) = \int_{t_{i-1}}^{t_i} f(t)dt, t > 0$$
 (3.6)

Note that P_i and α_i are changing from sample to sample, based on the values of sample size and/or control limit spread, while in the classical economic model of \bar{X} -chart P and α are constant.

Average In-Control, Out-of-Control, and Cycle Times

In this research, average cycle time is expressed as follows:

or,

$$ACT = AIC + AOOC + D (3.7)$$

where:

ACT - the average cycle time.

AIC - the average time for the occurrence of an assignable cause.

A00C - the average time that the process is out-of-control before the detecting sample is taken, evaluated, and plotted on the chart.

D - the average search time to find the assignable cause while the process is operating in the out-of-control condition.

AIC, AOOC, and D will be examined in turn.

The term AIC is equal to the mean of the distribution governing the process failure (shift) mechanism. For example, consider the following:

- (1) The length of time the process remains in the in-control state, given that it begins in control, is an exponential random variable with mean $\frac{1}{\lambda}$. That is, the process failure mechanism is governed by a Poisson process with intensity of λ occurrences per hour. In this case, AIC is equal to $\frac{1}{\lambda}$.
- (2) The length of time the process remains in the in-control state, given that it begins in control, is a Weibull random variable with parameters θ and η . In this case, AIC is equal to $\frac{1}{\theta}$ Γ $(1+\frac{1}{n})$.

The term AOOC is the average time that the process is out-of-control before the detecting sample is taken, evaluated, and plotted on the chart. The expression for AOOC is derived as follows.

In any time interval, the process has a chance of shifting to the out-of-control state. $P(00C_{\dot{1}})$ denotes this probability of a shift to the out-of-control condition in the $i^{\dot{t}h}$ interval.

First, assume that the process has shifted to the out-of-control state in an arbitrary interval, e.g., the ith interval. Now, given this assumption, consider the following cases:

(1) The shift is detected on the very first sample taken after the shift. In this case, the expected time the process operates in the out-of-control state before the detecting sample is plotted on the chart is as follows (see Figure 3.2):

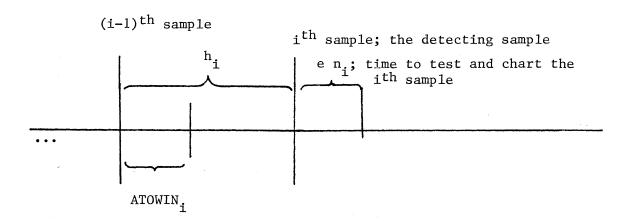


Figure 3.2 Out-of-Control Time Before the Detecting Sample is Plotted for Case (1)

$$h_i + e n_i - ATOWIN_i$$
 (3.8)

where:

h; - the length of the ith interval.

 $e n_i$ - the time to inspect and plot the i^{th} sample.

- ${\sf ATOWIN}_i$ the average time of the occurrence of the shift within the ith interval, given that the shift occurs in the ith interval. ${\sf Development}$ of the expression for ${\sf ATOWIN}_i$ will be deferred until later.
- (2) The first sample taken after the shift fails to detect the shift but the second sample taken after the shift, the (i+1)th sample, detects the shift. In this case, before the sample point is plotted on the chart, the process operates in the out-of-control state for the following time period (see Figure 3.3):

$$h_i + h_{i+1} + e n_{i+1} - ATOWIN_i$$
 (3.9)

(3) The first and second samples taken after the shift fail to detect the shift but the third sample taken after the shift, the (i+2)th sample, detects the shift. In this case, before the detecting sample point is plotted on the chart, the process operates in the out-of-control state for the following time period (see Figure 3.4):

$$h_i + h_{i+1} + h_{i+2} + e n_{i+2} - ATOWIN_i$$
 (3.10)

•

(n) Subsequent cases follow in an analogous manner.

Note that, given the original condition that the shift has occurred in the i^{th} interval, the probability of realizing case (1) is P_i , the

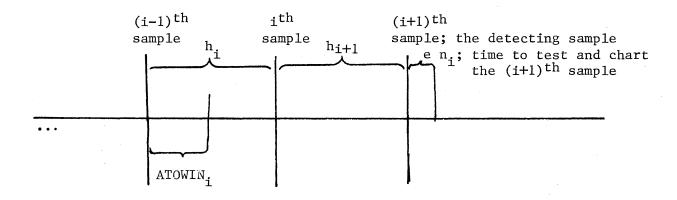


Figure 3.3. Out-of-Control Time Before the Detecting Sample is Plotted for Case (2).

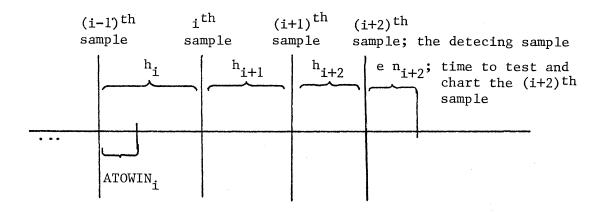


Figure 3.4. Out-of-Control Time Before the Detecting Sample is Plotted for Case (3)

probability of realizing case (2) is Q_i P_{i+1} , the probability of realizing case (3) is Q_i Q_{i+1} P_{i+2} , and so on. Therefore, given that the process shifts to an out-of-control condition in the i^{th} interval, the expected value of the time out-of-control before the detecting sample is plotted on the chart, TOOC, is:

$$= P_{i} \left[h_{i} + e \ n_{i} - ATOWIN_{i} \right]$$

$$+ \sum_{j=i+1}^{\infty} \begin{bmatrix} j-1 \\ \pi \\ k=1 \end{bmatrix} P_{j} \begin{bmatrix} j \\ \Sigma \\ k=1 \end{bmatrix} h_{k} + e \ n_{j} - ATOWIN_{i}$$
(3.12)

Now, the unconditioned expected value of TOOC can be obtained by taking the expectation of the conditional expectation of TOOC over all possible intervals. That is:

A00C = E[TOOC] =
$$\sum_{i=1}^{\infty} E[TOOC|OOC_i] P(OOC_i)$$
 (3.13)

The term D is the average search time to find the assignable cause after a point plotted on the chart falls outside the control limits.

Note that during this search time the process is operating in the out-of-control condition.

The proportion of time that the process is operating in the incontrol state is:

$$\beta = \frac{AIC}{ACT} \tag{3.14}$$

Similarly, the proportion of the time that the process is operating in the out-of-control state is:

$$(1-\beta) = \frac{(A00C + D)}{ACT}$$
 (3.15)

ATOWIN; Expression

ATOWIN $_{i}$ is the average time of the occurrence of the shift within the ith interval, given that the shift occurs in this interval. This is illustrated in Figure 3.5. The expression for ATOWIN $_{i}$ follows.

Let f(t), t > 0 represent the distribution of time to failure, then:

ATOWIN_i =
$$\frac{\int_{t_{i-1}}^{t_i} (t - t_{i-1}) f(t) dt}{\int_{t_{i-1}}^{t_i} f(t) dt}$$
(3.16)

This can be simplified as follows:

ATOWIN_i =
$$\frac{\int_{t_{i-1}}^{t_i} t f(t) dt - \int_{t_{i-1}}^{t_i} t_{i-1} f(t) dt}{\int_{t_{i-1}}^{t_i} f(t) dt}$$

$$(3.17)$$

$$= \frac{\int_{t_{i-1}}^{t_{i}} t f(t) dt}{\int_{t_{i-1}}^{t_{i}} f(t) dt} - t_{i-1}$$
 (3.18)

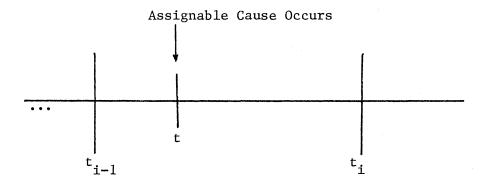


Figure 3.5. Average Time of Occurrence of Assignable Cause Within the $i^{\mbox{th}}$ Interval

For example, when time to failure is Weibull distributed, then:

$$f(t) = \begin{cases} \theta n & (\theta t)^{n-1} e^{-(\theta t)^n}, & t > 0 \\ 0, & \text{otherwise} \end{cases}$$
 (3.19)

 $ATOWIN_i$ is as follows:

ATOWIN_i =
$$\frac{\int_{t_{i-1}}^{t_{i}} t \, \theta n \, (\theta t)^{n-1} \, e^{-(\theta t)^{n}} \, dt}{\int_{t_{i-1}}^{t_{i}} \theta n \, (\theta t)^{n-1} \, e^{-(\theta t)^{n}} \, dt} - \sum_{j=1}^{t-1} h_{j}$$
(3.20)

Let $(\theta t)^n = u$, then $n \theta (\theta t)^{n-1} dt = d u$. Also $(\theta t)^n = u$ implies that $\frac{1}{\theta t} = u^n$. Therefore,

ATOWIN_i =
$$\frac{\int_{(\theta t_{i-1})^{n}}^{(\theta t_{i})^{n}} \frac{1}{\theta} u^{\frac{1}{n}} e^{-u} du}{\int_{(\theta t_{i})^{n}}^{(\theta t_{i})^{n}} e^{-u} du} - \sum_{j=1}^{i-1} h_{j}$$
(3.21)

$$= \frac{\frac{1}{\theta} \left[\gamma \left((\theta t_{i})^{n}, \frac{1}{n} + 1 \right) - \gamma \left((\theta t_{i-1})^{n}, \frac{1}{n} + 1 \right) \right]}{\theta \left[e^{-(\theta t_{i})^{n}} - e^{-(\theta t_{i-1})^{n}} \right]} - \sum_{j=1}^{i-1} h_{j}$$
(3.22)

where:

 $\gamma(a,x)$ - the unnormalized incomplete Gamma integral; $\int_{0}^{x} e^{-t} t^{a-1} dt$.

Expected Number of False Alarms

A false alarm occurs when a sample value falls outside the control limits, while the process is actually in-control. The false alarm results in searching for the non-existent assignable cause.

The expected number of false alarms during the average cycle time can be determined as follows.

First, assume that the process goes out-of-control in the ith interval. Given this assumption, the expected number of false alarms is:

E [number of |
$$OOC_i$$
] = $\alpha_1 + \alpha_2 + \dots + \alpha_{i-1}$ (3.23)

Equation (3.23) simply states that on the first sample taken from this process the chance of a false alarm is α_1 , on the second sample taken from this process the chance of a false alarm is α_2 , and so on until and including the (i-1)th sample. Any point which falls outside the control limits after the (i-1)th sample is a true alarm.

The expected number of false alarms during a cycle can be determined by summing the expected number of false alarms during a given interval over all possible intervals while weighting each of them by their corresponding probabilities. So,

ENFALS =
$$\sum_{j=1}^{\infty} P(00C_{j}) \begin{bmatrix} \sum_{j=1}^{j-1} \alpha_{j} \end{bmatrix}$$
 (3.24)

The expected number of false alarms per hour of operation is equal to $\frac{\mathsf{ENFALS}}{\mathsf{ACT}} \; \boldsymbol{\cdot}$

Cost of Looking for False Alarms

If T is the average cost of looking for an assignable cause when the process is in-control, then the expected cost per hour of looking for false alarms, C_1 , is:

$$C_1 = T \frac{ENFALS}{ACT}$$
 (3.25)

Cost of Finding the Assignable Cause

In this research, in accordance with Duncan's model, a cycle is defined so that there is only one assignable cause per cycle. Therefore, if the average cost of finding the assignable cause when it occurs is W, the average cost per hour on this account, C_2 , is:

$$C_2 = \frac{W}{ACT} \tag{3.26}$$

Cost of Sampling and Inspection

Duncan [20] assumes that the cost of sampling and inspection is composed of two components. One of the components, b, is the fixed cost of taking, testing, and plotting the sample that is independent of the sample size. The other component, c, is the variable cost per item of sampling, testing, and charting. Furthermore, Duncan assumes that a sample of size n is taken every h hours. Therefore, the cost per hour of sampling, testing, and charting is simply given by $\frac{b+cn}{h}$.

In this research, the sample sizes and the intervals between samples (and the control limit spreads) are allowed to vary over time. However, it is desirable to develop an economic model of the generalized

dynamic \bar{X} -control chart which closely follows Duncan's cost model. Therefore, the following expression is derived to describe the equivalent of Duncan's average hourly sampling and charting cost for a dynamic \bar{X} -chart.

First, assume that the process is known to go out-of-control in a specific interval, e.g., the $i^{\mbox{th}}$ interval. Given this assumption, consider the following cases:

(1) The shift which occurred in the ith interval is detected on the very first sample. In this case, the sampling and charting cost per cycle is equal to the sum of the costs of all samples taken so far; the first i samples, plus the cost of all future samples which are to be taken while the detecting sample is plotted, plus the subsequent search time to find the assignable cause. That is

CI1 =
$$\sum_{j=1}^{i} (b + cn_j) + \sum_{k=i+1}^{m1} (b + cn_k)$$
 (3.27)

where:

Note that the cycle length in this case, as is illustrated in Figure 3.6, is equal to:

LI1 =
$$\sum_{j=1}^{i} h_j + e n_i + D$$
 (3.28)

So, the average hourly cost of sampling and charting for this case is equal to $\frac{\text{CI1}}{\text{III}}$.

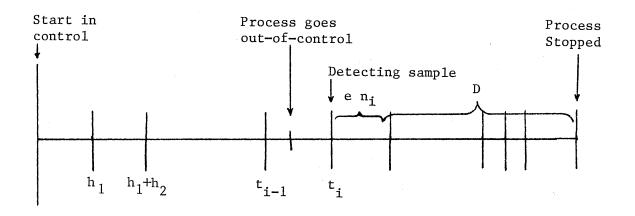


Figure 3.6. Cycle Length for Case (1)

(2) The shift which occurred in the i^{th} interval goes undetected on the first sample after the shift, the i^{th} sample, but is detected on the second sample after the shift, the $(i+1)^{th}$ sample. Following the same argument as in case (1), the sampling and charting cost per cycle is:

CI2 =
$$\sum_{i=1}^{i+1} (b + cn_i) + \sum_{k=i+2}^{m2} (b + cn_k)$$
 (3.29)

where

m2 = min
$$\ell$$
2 $\Rightarrow \sum_{j=i+2}^{\ell} h_j > e n_{i+1} + D$

The cycle length in this case is:

LI2 =
$$\sum_{j=1}^{i+1} h_j + e n_{i+1} + D$$
 (3.30)

So, the average hourly cost of sampling and charting for this case is equal to $\frac{\text{CI2}}{\text{II2}}$.

(3) The shift which occurred in the i^{th} interval goes undetected on the first and second samples taken after the shift, but is detected on the third sample after the shift, the $(i+2)^{th}$ sample. Following the same argument as in case (1), the sampling and charting cost per cycle in this case is:

CI3 =
$$\sum_{j=1}^{i+2}$$
 (b + cn_j) + $\sum_{k=i+3}^{m3}$ (b + cn_k) (3.31)

where:

m3 = min
$$\ell$$
3 $\Rightarrow \sum_{j=i+3}^{\ell 3} h_j > e n_{i+2} + D$

The cycle length in this case is:

LI3 =
$$\sum_{j=1}^{i+2} h_j + e n_{i+2} + D$$
 (3.32)

So, the average hourly cost of sampling and charting for this case is equal to $\frac{\text{CI}\,3}{\text{LI}\,3}$.

•

(n) Subsequent cases follow in an analogous manner.

Therefore, given the original condition that the shift occurs in the i^{th} interval, the probability of case (1) happening is P_i , the probability of case (2) happening is Q_i P_{i+1} , the probability of case (3) happening is Q_i Q_{i+1} P_{i+2} , and so on. Therefore, the average hourly cost of sampling and charting given that the process goes out-of-control in the i^{th} interval is:

E
$$\begin{bmatrix} \text{hourly cost of } \\ \text{sampling and } | \text{ OOC}_i \end{bmatrix} = P_i \frac{\text{CI1}}{\text{LI1}}$$

+
$$Q_i P_{i+1} \frac{CI2}{LI2} + Q_i Q_{i+1} P_{i+2} \frac{CI3}{LI3} + ...$$
 (3.33)

Now, the overall hourly cost of sampling and testing can be determined by finding the expectation of this conditional expectation:

$$C_3 = AHCS = \sum_{i=1}^{\infty} E \begin{bmatrix} hourly cost of \\ sampling and \\ charting \end{bmatrix} P(00C_i)$$
 (3.34)

Cost Formulation

Based on the above derivation of different cost components, the average hourly net income of the process under the surveillance of a dynamic \overline{X} -control chart is developed as follows.

where

Average hourly cost of false alarms is:

$$C_1 = T \frac{ENFALS}{ACT}$$
 (3.38)

Average hourly cost of finding the assignable cause is:

$$C_2 = \frac{W}{ACT} \tag{3.39}$$

Average hourly cost of sampling, testing, and charting is:

$$C_3 = AHCS (3.40)$$

Therefore,

Process average hourly
$$= V_0 \beta + V_1(1-\beta) + C_1 + C_2 + C_3$$
 net income
$$= V_0 \beta + V_1(1-\beta) - T \frac{ENFALS}{ACT}$$

$$- \frac{W}{ACT} - AHCS$$
 (3.41)

=
$$V_0 - (1-\beta) M - T \frac{ENFALS}{ACT} - \frac{W}{ACT} - AHCS$$
 (3.42)

where:

$$M = V_0 - V_1 (3.43)$$

$$= V_0 - L$$
 (3.44)

where:

$$L = (1-\beta) M + T \frac{ENFALS}{ACT} + \frac{W}{ACT} + AHCS$$
 (3.45)

The objective of this economic formulation is to maximize average hourly net income which is equivalent to minimzing the loss-cost L.

Summary

An economic model is developed to determine the design of a generalized dynamic \bar{X} -control chart. This model is developed using Duncan's approach to the economic design of control charts. The mathematical development and derivation of the net income per hour for this dynamic \bar{X} -control chart is discussed. A proof that Duncan's model is a special case of this generalized model is given in Appendix A.

The model developed in this research has the advantageous capability of representing different process failure mechanisms while Duncan's model applies only to the exponential time to failure mechanisms. Also, it allows the incorporation of any dynamic control charting philosophy in which sample sizes, intervals between samples, and control limit widths are free to change as functions of time, or the process history, throughout the chart's operation.

In the next chapter, a special dynamic control charting methodology is specified, together with an optimization procedure to find the minimum loss-cost design. The minimum loss-cost design is equivalent to the design which maximizes net profit per hour.

CHAPTER IV

ECONOMIC OPTIMIZATION OF A DYNAMIC \bar{X} -CONTROL CHART; ECONOMIC COMPARISON WITH DUNCAN'S \bar{X} -CHART

Introduction

The purpose of this chapter is: (1) to introduce a dynamic methodology employed to optimize the economic model of the \bar{X} -control chart developed in Chapter III, (2) to discuss the computational aspects of implementing the theoretical model of Chapter III on a computer, (3) to present a computer search algorithm developed to carry out the optimization of the dynamic \bar{X} -control chart, and (4) to provide an economic comparison and analysis between Duncan's optimal plans and the plans obtained by employing the dynamic \bar{X} -control chart when the actual underlying process failure mechanism is not memoryless.

The economic formulation of a generalized dynamic \bar{X} -control chart has been discussed in Chapter III. In order to optimize this dynamic model, a relationship between n_{i-1} and n_i , h_{i-1} and h_i , and k_{i-1} and k_i must be assumed. The nature of this relationship is determined by the choice of the dynamic control chart methodology selected, one such methodology being presented in this chapter. This methodology is incorporated into the dynamic \bar{X} -control chart model through a computer program.

The model of Chapter III is a complex model involving several summations of series with infinitely many terms. Since, ultimately, this model is to be implemented in the form of a computer program, it is necessary to consider the computational implications and feasibilities of these summations of series calculations. Methods and procedures are found and successfully employed to approximate these summations of series within a reasonable number of computer calculations.

The optimum of a dynamic \bar{X} -control chart is obtained when the values for decision variables— n_i 's, h_i 's, and k_i 's—result in the minimum cost of operating the process subject to a specified shift in the process mean and for a specific set of costs and process failure distribution. This optimum would also be dictated by the specific choice of the control chart methodology. The approach to optimization of the dynamic \bar{X} -control chart consists of the use of a "good" starting solution and a computer direct search technique developed specifically for the loss-cost function used.

The effect of different process failure mechanisms on the optimum economic design of the \bar{X} -control chart is examined. Based on several representative examples, the total costs of operating the process are compared for the optimal designs given by Duncan's \bar{X} -chart model and the corresponding designs generated by the dynamic \bar{X} -chart model. The economic comparison is performed assuming that the true process failure mechanism at work is an Increasing Failure Rate (IFR) Weibull distribution rather than the exponential distribution employed by Duncan.

Notation

The following terms are employed to facilitate this chapter's presentation. Some of the terms introduced here are not defined previously. Some other cost and distribution parameters are introduced in Chapter III. They are repeated here for clarity, because they are used extensively in the following economic comparisons.

- n_i number of individual measurements making up the i^{th} sample; that is, the i^{th} sample size.
- In size of the first sample to be taken; n_1 .
- nf a factor used to change the sample size as a function of the sample number i.
- h_1 time interval elapsed between taking the $(i-1)^{\mbox{th}}$ and the $i^{\mbox{th}}$ samples.
- Ih time interval elapsed from the start of the process until the first sample is taken.
- t i time elapsed from startup of the process in-control until the i th sample is taken; t i = $\sum_{j=1}^{n} h_j$.
- a factor used to change the sampling intervals throughout the control charts's operation as a function of the sample number
 i.

- k_i number of ith sample average standard deviations separating each control limit and the original process mean. That is, the ith sample average is plotted on a chart having the upper and lower control limits of $\overline{X}'' + k_i \frac{\sigma}{\sqrt{n_i}}$ and $\overline{X}'' k_i \frac{\sigma}{\sqrt{n_i}}$, respectively.
- Ik value of k; when the first sample is being plotted.
- kf a factor used to change the control limit spread as a function of the sample number i.
- θ , η parameters of a Weibull distribution; density function f(t) is Weibull if: $f(t) = \theta \eta (\theta t)^{\eta-1} e^{-(\theta t)^{\eta}}$, t > 0.
- λ parameter of an exponential distribution. Note that θ = λ when η = 1.
- μ mean of the process failure distribution.
- δ magnitude of the out-of-control shift in the process mean in multiples of the process standard deviation.
- the rate at which the average sampling, testing, and charting time for a sample increases with the sample size.
- D the average search time for an assignable cause.
- M the reduction in process hourly income due to the occurrence of the assignable cause.
- T the average cost per occasion of looking for an assignable cause when no assignable cause exists.
- the average cost per occasion of finding the assignable cause, when it exists.
- b cost per sample of sampling, testing, and charting that is fixed and independent of the sample size.

c - unit cost of sampling, testing, and charting that is related to the sample size. The relationship is assumed to be linear.

n* - close-to-optimum real-value sample size used in Duncan's
model optimization.

n* - the optimal integer-valued sample size for Duncan's model.

 $L_{n'}^*$, $L_{n''}^*$ - local optimums for Duncan's model when sample size is fixed at n' and n", respectively.

A Dynamic X-Control Chart Methodology

The dynamic \bar{X} -control chart model developed in Chapter III is a generalized model in that there are no restrictions set on the relationships between n_{i-1} and n_i , h_{i-1} and h_i , and k_{i-1} and k_i . For the purpose of optimizing this model, the following relationships for n_i , h_i , and k_i are established. They are then incorporated into the generalized dynamic \bar{X} -control chart model in a computer program.

$$n_i = In (nf)^{i-1}$$
 (4.1)

$$h_i = Ih (hf)^{i-1}$$
 (4.2)

$$k_i = Ik (kf)^{i-1}$$
 (4.3)

where

 n_{i} - the size of the i^{th} sample to be taken. That is, the size of the sample to be taken at time t_{i} = $\sum\limits_{j=1}^{h} h_{j}$.

In - the size of the initial sample to be taken.

- of a factor used to change the sample size throughout the control chart operation. For all practical purposes .8 < nf < 1.2, thus allowing the sample size to increase as a function of sample number when nf > 1, or to decrease as a function of sample number when nf < 1. Note that when nf = 1, then the sample sizes stay the same throughout the control chart's operation.
- h_i the size of the i^{th} interval between samples. That is, the time interval elapsed between taking the $(i-1)^{th}$ and the i^{th} samples.
- Ih the time interval elapsed from the start of the process until the first sample is taken.
- hf a factor used to change the sampling intervals throughout the control chart's operation. The same comments mentioned for nf values between .8 and 1.2 apply for hf.
- k_i the size of the distance in multiples of the sample average standard deviation between each control limit and the original process mean. That is, the upper and lower control limits at time $t_i = \sum\limits_{j=1}^{i} h_j$ are $\overline{X}'' + k_i \frac{\sigma}{\sqrt{n_i}}$ and $\overline{X}'' k_i \frac{\sigma}{\sqrt{n_i}}$, respectively.
- Ik the value of k_1 when the first sample is being plotted.
- kf a factor used to change the control limit spread throughout the control chart's operation. The same comments mentioned for nf values between .8 and 1.2 apply for kf.

The empirical justification for employing equations (4.1), (4.2), and (4.3) to represent a specific dynamic relationship is based on logical observations. Many process failure mechanisms are characterized

by IFR distributions such as the Weibull. For these processes, as time of process operation increases, the probability of it shifting to an out-of-control state increases. Thus, more frequent sampling, increasing sample sizes, and/or decreasing control limit widths might be economically justified by their detecting the shift earlier. For some other processes, it might be more economical to have one or more of the sampling intervals, sample sizes, and control limit widths treated as constants throughout the chart's operation, while the rest are either increasing or decreasing over time.

The use of equations (4.1), (4.2), and (4.3) in conjunction with a computer optimization technique help assure that the desired combinations of the values of nf, hf, and kf can be determined, resulting in the least total cost of operation. Thus, whether any or all of the sample sizes, sampling intervals, and/or control limit widths should be increasing, constant, or decreasing to best suit a specific process environment is determined by optimization of the loss-cost model.

It should be noted that the dynamic relationship specified by equations (4.1), (4.2), and (4.3) is only one special set of many possible relationships. Ideally, no such predetermined relationships between $\mathbf{n_{i-1}}$ and $\mathbf{n_i}$, $\mathbf{h_{i-1}}$ and $\mathbf{h_i}$, and $\mathbf{k_{i-1}}$ and $\mathbf{k_i}$ is desirable. Rather, an optimization procedure should determine $\mathbf{n_i}$, $\mathbf{h_i}$, and $\mathbf{k_i}$ such that they result in the least total cost. However, this does not seem practically possible because the extremely large dimensionality of the problem makes the solution of the problem by any optimization procedures computationally infeasible. On the other hand, it is believed that the dynamic relationships of equations (4.1), (4.2), and (4.3) provide a versatile dynamic control chart methodology with enough feasibility

in decision variables so that not much is sacrificed in loss-cost improvement.

Model Implementation; Computational Considerations

The model developed in Chapter III employs several summations of series where each consists of the sum of infinitely many terms. Based on the nature of the terms that are comprising these summations of series, they can be divided into two categories. One type is the summation of series in which the ith term of the series involved $P(00C_i)$, the probability that the process shifts to the out-of-control condition in the ith interval. The summations of series given by equations (3.13), (3.24), and (3.34) are of this type. The other type is the summation of series in which the jth element involves $\prod_{k=i}^{g} Q_k$, where j assumes values from (i+1) to infinity. The summations of series given by equations (3.12) and (3.33) belong to the latter category.

It is obvious that when actually computing any of the above summations of series, the summation of terms cannot be carried out for all the infinitely many terms. The following observations are intended to make the mathematical formulations of Chapter III a computationally feasible reality.

Summations of Series Involving $P(OOC_i)$

In order to compute the summations of series involving $P(00C_i)$, first note that $P(00C_i)$, as given by equation (3.6), is the area under the failure density function from time t_{i-1} until time t_i . It is obvious that as the value of i, and consequently the values of t_{i-1} and t_i become very large, the value of $P(00C_i)$ approaches zero for any of

the realistic failure density functions of interest. In other words, as the cumulative distribution function of the random variable representing the process failure mechanism approaches one, $P(00C_i)$ approaches zero. Therefore, there is a quantile point for the cumulative distribution function after which $P(00C_i)$ values and their corresponding terms of the summation of series are very small, resulting in contributions to the summations of series which are practically insignificant.

This fact is advantageously used to approximate the summations of series involving $P(00C_{\dot{1}})$. This is accomplished by computing the terms of the series and carrying out their accumulation only until a reasonably high quantile of the cumulative distribution function of the failure process is reached.

Further justification for the use of this approximation approach is given in Table 4.1. This Table gives the values of the total loss-cost of the model of Chapter III, as given by equation (3.45), assuming that n_i , h_i , and k_i are constant throughout the operation and the process failure density is exponential. In other words, the model of Chapter III is simplified to Duncan's \bar{X} -control chart model. Three different cases each of four examples are then considered, stopping the summation of series calculations at quantiles of .99, .9999, and .999999. Exact values of the loss-costs are also obtained using the exact version of Duncan's model. In Duncan's model, the loss-cost is expressed in terms of a simple mathematical equation and can be calculated very accurately.

A comparison of the results obtained using the model of Chapter III and Duncan's exact model provides good evidence that this approximation approach works quite well (see Table 4.1). Note that the higher the value of quantile used to approximate the summations of series, the more

TABLE 4.1 LOSS-COST VALUES FOR EXPONENTIAL PROCESS FAILURE MECHANISM

Selected Design n h k			λ	.99 Quantile Value Loss-Cost†		.9999 Quantile Value Loss-Cost		.999999 Quantile Value Loss-Cost		Duncan's Exact* Model Loss-Cost
 5.	1.41	3.08	.01	460.52	3.99170	921.03	4.01254	1381.55	4.01278	4.01278
4	0.78	2.94	.03	153.51	9.55198	307.01	9.59190	460.51	9.59238	9.59239
6	1.4	3.7	.01	460.52	6.34565	921.03	6.36819	1381.55	6.36845	6.36845
8	12.	1.9	.01	460.52	2.39604	921.03	2.42093	1381.55	2.42128	2.42128
8	12.	1.9	.01	460.52	2.39604	921.03	2.42093	1	381.55	381.55 2.42128

[†]Loss-cost is in terms of dollars per hour. *Duncan's exact cost model implemented in a double-precision computer program.

accurate is the computation. At the limit, as the quantile approaches one, the summations of series computations become equal to their theoretical values within the computer's accuracy. However, the results obtained using a very high value for the quantile, e.g., .999999, which at times necessitate a large number of calculations on the computer, do not differ significantly from the results obtained using the reasonable quantile values of .99 to .9999, which require relatively much smaller numbers of calculations. For practical purposes enough accuracy is maintained when a quantile of .99 is used. However, as a precaution against limiting the application of the model, when the higher computing cost can be justified by the higher accuracy desired, the specification of the desired quantile is left to the user in the interactive computer program which implements the model.

Summations Involving Products of Q's

In order to compute the summations of series involving $\prod_{k=1}^{J-1} Q_k$, first note that Q_k , as given by equation (3.3), is the probability that a shift goes undetected on the k^{th} sample. It is obvious then that the j-1 value of $\prod_{k=1}^{J-1} Q_k$ is going to approach zero as (j-i-1) becomes a very large number. Therefore, there is a point in computation of the series where the contribution of the terms of the summation to the total sum ceases to be significant. Therefore, a check in the computer program can identify when this point is reached, the terms of the summation become negligible, and the calculation of the series is concluded.

Experimentation with this type of summation of series, using Duncan's optimal design values, shows that in fact this terminating point of calculation is approached very fast. This implies that the

approximation of this type of summation of series, especially in the proximity of the optimal design, requires few computations and is quite efficient on the computer.

Simultaneous Restrictions on n_i and k_i

Note that for the summations of series involving products of Q_k terms, the approximation of the series becomes more efficient as the values of the Q_k terms become smaller. For example, if all the consecutive Q_k values are smaller than .1, then after only ten iterations, the next term of the series is multiplied by a product of Q_k values which does not exceed $(.1)^{10} = 1$. $\times 10^{-10}$. It is obvious then that the terminating point of calculations for approximating this series is reached very quickly.

Furthermore, it is observed that for all of Duncan's model optimal designs [20] [33] the value of P (corresponding to P_i in the dynamic model), the probability of detecting the shift on a given sample, is always greater than .7, with only two examples in which P is still greater than .55. Actually, in the majority of cases P is greater than .8. This fact is illustrated in Table 4.2 where P is calculated for each of the 25 examples given in [20]. This point has also been recognized by Chiu, et al. [12] and Montgomery [51] who have developed a scheme and computer program for simplified economic design of Duncan's \bar{X} -chart by prespecifying a high value of .9 or .95 for P. Furthermore, Chiu, et al. [12] observe that the loss-cost function is robust with respect to P. They state that in most circumstances the difference between the restricted minimum, when P is set at .9 or .95, and the exact minimum value of the loss-cost is very small.

TABLE 4.2 P VALUES FOR DUNCAN'S EXAMPLES

Optimal Design									
Ex. No.*	Desc.†	n	h	k	k-6√ <u>n</u>	Р			
1	δ=2,G	5	1.41	3.08	-1.3921	.9181			
2	$\delta = 2$,G	5	1.02	3.08	-1.3921	.9181			
3	δ=2,G	4	0.78	2.94	-1.0600	.8554			
4	$\delta = 2$,D	5	1.3	3.2	-1.2721	.8983			
5	δ=2,G	4	0.41	2.95	-1.05	.8531			
6	δ=2,D	5	0.1	3.2	-1.2721	.8983			
7	δ=2,G	2	0.94	2.69	1384	.5550			
8	$\delta = 2,G$	5	1.62	3.05	-1.4221	.9225			
9	$\delta = 2, D$	4	1.3	2.4	-1.6000	.9452			
10	δ=2,G	6	1.45	3.67	-1.2290	.8905			
11	δ=2,D	7	1.3	4.4	8915	.8137			
12	δ=2,G	6	3.47	2.88	-2.0190	.9783			
13	$\delta = 2, D$	3	2.6	2.4	-1.0641	.8564			
14	δ=2,G	1	4.66	1.46	5400	.7054			
15	$\delta = 2, D$	3	.8	2.4	-1.0610	.8564			
16	$\delta = 1,G$	14	5.47	2.68	-1.0617	.8558			
17	$\delta = 1, D$	12	1.6	2.6	8641	.8062			
18	$\delta = 1$,G	21	7.23	3.39	-1.1926	.8835			
19	$\delta = 1$,G	18	11.02	2.56	-1.6826	.9538			
20	$\delta = 1, D$	8	12.	1.9	9284	.8234			
21	δ=.5,G	38	23.45	2.21	8722	.8085			
22	$\delta = .5,G$	21	1.3	2.11	1813	.5719			
23	$\delta = .5, D$	55	30.	2.3	-1.4081	.9204			
24	$\delta = .5, D$	55	30.	2.3	-1.4081	.9204			
25	δ=.5,G	12	54.32	1.13	6021	.7264			

^{*}All example numbers are the same as those used in Duncan's paper. $^\dagger D$ = Duncan's optimal design G = Goel's optimal design

⁽Choice between D and G is based on the minimum loss-cost criterion.)

Fortunately, this property is very desirable in terms of computational efficiency, since a high value of P implies a low value for Q (corresponding to Q_k in the dynamic model). Based on the above discussion, it is then logical to restrict P_i values in the dynamic model to be at least greater than .5, without expecting to cause significant restrictions, if any at all, on the optimal solution. This constraint is expressed as follows

$$P_i > .5$$
 (4.4)

Substitute in equation (3.2) for P_i results in:

$$\Phi(-\delta \sqrt{n_i} - k_i) + \Phi(\delta \sqrt{n_i} - k_i) > .5$$
 (4.5)

For simplicity, assume that only the positive shift (i.e., $\delta > 0$) actually occurs. The first term on the left of the inequality is then practically zero. (If $\delta < 0$, a similar constraint can be obtained noting that the second term on the left of the inequality is zero.) Therefore,

$$\Phi \ (\delta \sqrt{n_i} - k_i) > .5$$
 (4.6)

or

$$\delta \sqrt{n_i} - k_i > 0 \tag{4.7}$$

Experimentation with the dynamic \bar{X} -chart model shows that the above constraint is not a binding constraint at the optimal solution. However, its inclusion ensures the computational efficiency of the computer model evaluations during the optimization of the model. That is, it will guard against some computationally undesirable combinations of the decision variables which might otherwise be tried in the search toward

an optimum. This constraint is incorporated in the objective function in the form of a barrier and/or a penalty method.

Economic Optimization of the Dynamic \bar{X} -Control Chart

General Strategy

The goal of economic optimization of the dynamic \bar{X} -control chart is to find the optimal combination of the values of the decision variables --Ih (or ISTEPS), hf, Ik, kf, In, and nf--which result in the minimum loss-cost. This minimum loss-cost corresponds to the maximum average hourly net income obtained from the process. Because of the complexity of the dynamic \bar{X} -control chart model developed in Chapter III, there exists no analytically explicit optimal solution. Therefore, multi-dimensional computer search techniques must be used for optimization of the model.

A special direct search algorithm is developed for optimization of the dynamic \bar{X} -chart model. This algorithm is designed based upon much experimentation with the loss-cost function of the dynamic model so that it takes advantage of the special landscape of this function. Furthermore, the algorithm is designed based on the observation that all the six decision variables of the model cannot be simultaneously optimized. This is due to the fact that In and ISTEPS are integer variables.

This optimization algorithm makes use of a modified Coggins search technique [44], the ideas of Davies, Swann, and Campey [39], Powell's algorithm [39], and the basic philosophy behind direct search algorithms [40]. An important factor in employing this optimization algorithm, as well as most other computer optimization search routines, is the

selection of the initial starting conditions for the decision variables of the model. A "good" initial starting condition contributes to a fast and reliable determination of the optimum solution. In this research, rather than selecting a starting point haphazardly, a logical method is used to determine a good starting point.

Techniques Used in Optimization Algorithm

Before describing the optimization search algorithm in detail, the following methods which are extensively used in the search algorithm are very briefly discussed.

The Davies, Swann, and Campey (DSC) technique [39] is an efficient unidimensional search algorithm. In the DSC search, steps of increasing size are taken until the unique optimum of the function is bracketed. Then the best three values of the decision variable which are bracketing the minimum are used to fit a quadratic function to them. The fitted quadratic is considered to provide a good approximation to the objective function over an interval close to its minimum. This is based on the observation that many objective functions behave as a quadratic in the proximity of their minimum. The DSC technique then approximates the minimum of the objective function by analytically calculating the minimum of the fitted quadratic.

Powell's technique [39] is another efficient unidimensional search algorithm. In this technique a quadratic approximation is carried out using the first three points obtained in the direction of search. The value of the decision variable corresponding to the minimum of this fitted quadratic is determined analytically. The set of the three values of the decision variable is updated and the quadratic

approximation is repeated until the minimum of the objective function is located within the required precision. Thus, Powell's technique is different from DSC in that it does not first bracket the minimum and in that it employs several quadratic fits until the required precision is obtained.

Coggins' technique [44] is based on the observation that a combination of the DSC and Powell algorithms work better than either of the individual algorithms [39]. This technique then employs the DSC technique to bracket the minimum first and Powell's technique to approximate the minimum. Powell's technique is repeated until the required precision on the minimum is reached or other convergence criteria are satisfied [44]. The specific implementation of Coggins' technique in [44] is modified to make it more efficient. Experimentation with several test functions shows that the modified Coggins' technique works 15% to 20% more efficiently, in terms of the number of objective function evaluations, than the implementation in [44] which performs some redundant objective function evaluations. Furthermore, the original implementation [44] is written for maximization problems. This is also changed so that the modified Coggins' technique used in this research searches for the minimum.

Optimization Search Algorithm

A special search algorithm is developed to optimize the loss-cost function of the dynamic \bar{X} -control chart. It is designed based upon much experimentation with the loss-cost function so that it is efficient in finding an optimum or near-optimum solution for this specific function. Efficiency of the algorithm, in terms of the number of objective

function evaluations, is important because of the complexity of the loss-cost function. In no way is this algorithm meant to be a general purpose optimization search algorithm.

The central logic of the algorithm is developed for a two-at-a-time search. Thus, the algorithm first optimizes the loss-cost over hf and Ih (or preferably ISTEPS) variables. It then proceeds to optimize over kf and Ik, and finally concludes one pass of the search by optimizing over nf and In.

The detailed structure of the search routine for optimizing the loss-cost over the hf and ISTEPS variables is as follows:

- With a good starting solution, do a line search employing Coggins technique along variable hf to find the local minimum along this direction. Much experimentation with the loss-cost function has shown this initial line search to be very effective in terms of its contribution to objective function improvement.
- 2. Starting with the best minimum obtained thus far, move along the other variable, ISTEPS. If this new move is a success, then double the step size for the next move in this direction along the variable ISTEPS. If this new point is a failure, then either:
 - (i) Reduce the step size to its minimum acceptable value given that the current step size is greater than its minimum acceptable value, or
 - (ii) Switch the direction of the search for the next move along this variable.
- 3. Follow the logic of step 2 for the other variable, hf.

4. Iterate between steps 2 and 3 until either of the following conditions is satisfied.

ŝ

- (i) Successive failures are encountered in both directions along only one of the variables. In this case, go to step 5.
- (ii) Successive failures are encountered in both directions along both of the variables. In this case, go to step 7.
- (iii) The user-specified limit on the number of loss-cost function evaluations is reached. In this case, stop the search.
- 5. Since successive failures are encountered in both directions along one of the variables, the minimum is bracketed along that variable. Therefore, proceed to Powell's method and repeatedly employ quadratic fits to estimate the minimum more accurately until either of the following is satisfied:
 - (i) The required precision for the decision variable is attained.
 - (ii) A specified number of quadratic fits, depending on the previous search history, is performed. Note that in the case of a decision variable for which the required precision and the minimum step size value are the same, Powell's method is skipped. This assures that only integer values are tried for the integer variables, where the required precision and the minimum step size values are equal to one.

- 6. Go to either of steps 2 or 3 to search along the other direction.
- 7. Try an additional point by moving further ahead with both variables in the direction of the last successful moves for each. Then, similarly, try an additional such point. This strategy is to some degree helpful in guarding against small bumps that might exist in the objective function landscape. If either of these trial points provides an improvement in the loss-cost, then proceed to step 2. If no improvement is observed in either trial, conclude the two-dimensional search.

The accuracy of the search algorithm results can be increased, if the following step is appended at the beginning of step 7.

7A. Perform a line search using Coggins' method for each of the variables.

For the loss-cost function of this research, this additional level of accuracy is not judged to compensate for the additional computational burden. The minimum obtained by concluding the two-dimensional search without step 7A seems quite satisfactory.

The detailed structure of the search routine for optimizing kf and Ik variables is as follows:

- Start with the optimal solution obtained in the conclusion of the search over the hf and Ih (or ISTEPS) variables or the best solution found thus far.
- Move along the kf variable. Follow the same logic as that of step 2 of the search over hf and ISTEPS.
- 3. Follow the remaining steps, 3 to 7, of the search over hf and ISTEPS while replacing these variables with kf and Ik.

The detailed structure of the search routine for optimizing the nf and In variables is as follows:

- Start with the optimal solution obtained in the conclusion of the search over the kf and Ik variables or the best solution found thusfar.
- Move along the nf variable. Follow the same logic as that of step 2 of the search over hf and ISTEPS.
- 3. Follow the remaining steps, 3 to 7, of the search over hf and ISTEPS while replacing these variables with nf and In.

Note that the search over kf and Ik, and nf and In differs from the search over hf and Ih (or ISTEPS) in that Coggins' method of line search is not employed in step 1. Certainly the use of this method in step 1 would not be detrimental to the accuracy of the optimum solution. However, experimentation with the loss-cost function indicates that the use of a very accurate line search, Coggins' method, at the beginning of the search over kf and Ik or nf and In slightly deducts from the efficiency of the search.

After one pass of the special search algorithm through all six decision variables is complete, the user can then decide to employ the search algorithm to further optimize the loss-cost function. However, experimentation shows that when a "good" starting point is employed, not much cost improvement is obtained after the first pass.

Note that the proper specification of step sizes, their associated maximum and minimum values, and the required precisions for the six directions of the search can play an important role in the efficiency and reliability of the search algorithm. Recommended values of the above parameters are set in the computer program of Chapter V. However,

these values are submitted to the user for verification. The change of these parameters requires a good understanding of the search routine. Careless specification of these parameters might cause the search to stop short of the optimum and/or a decrease in the search technique's efficiency.

A similar point of caution applies to the specification of the initial starting point. The user is advised to start with a "good" initial point suggested by the program. However, when in doubt, several subjectively proper initial starting points can be tried to increase the reliability of the optimum solution obtained.

Testing the Search Algorithm

The central algorithm of the special search routine is a two-dimensional search technique. Although it is a special search technique, developed only for the loss-cost function of interest, it is tested on three general test functions.

The first test problem is the following two-dimensional function used to test several of the algorithms in [44].

$$F_1 = -3803.84 - 138.08 X_1 - 232.92 X_2 + 123.08 X_1^2$$

$$+ 203.64 X_2^2 + 182.25 X_1 X_2$$
(4.8)

This function has a minimum value of approximately -3873.9 at X_1 = .20609, X_2 = .4796 according to [44]. Using the same starting point as is used in Kuester and Mize [44], the algorithm locates a minimum value of -3873.22 at X_1 = .2054, X_2 = .480 in 60 iterations. Note that a

Rosenbrock direct search algorithm implemented in [44] took about the same number of iterations, 62.

The second test problem is due to Rosenbrock [61]. It has the following functional expression:

$$F_2 = 100 \cdot (X_2 - X_1^2)^2 + (1 - X_1)^2$$
 (4.9)

This function has a minimum value of zero at $X_1 = 1$., $X_2 = 1.0$. Starting from $X_1 = .2$, $X_2 = 1.0$, the search technique successfully located a minimum value of .284 x 10^{-7} at $X_1 = .99999$, $X_2 = 1.0$ after 23 iterations. However, when starting at a point far from the minimum, $X_1 = -1$, $X_2 = -1$, the search technique failed to find the minimum. This is not considered as a problem for the search technique because Rosenbrock's function has a very narrow and steep curved valley which causes even many of the more generalized optimization techniques to fail in locating its minimum.

The third test is a problem due to Beale [6]. This problem has the following functional expression:

$$F_{3} = (1.5 - X_{1}(1 - X_{2}))^{2} + (2.25 - X_{1}(1 - X_{2}^{2}))^{2} + (2.625 - X_{1}(1 - X_{2}^{3}))^{2}$$

$$+ (2.625 - X_{1}(1 - X_{2}^{3}))^{2}$$

$$(4.10)$$

This function has a minimum value of zero at $X_1 = 3$, $X_2 = .5$. Using the suggested starting point of $X_1 = 0$, $X_2 = 0$, the search technique successfully located a minimum of .7 x 10^{-7} at $X_1 = 2.99$, $X_2 = .498$ after 529 iterations.

In summary, the central algorithm of the special optimization technique works properly. It is not, however, intended as a general optimization algorithm as it might perform poorly or inefficiently for some of the general optimization problems.

Finding a "Good" Starting Point

The importance of a good set of initial values of the decision variables to be used in a computer optimization technique cannot be overemphasized. Based on the choice of the starting conditions, a computer search technique may find the desired optimum either efficiently or inefficiently, and either realiably or far from the global optimum. In this work, the starting conditions having the properties of efficiency and reliability are called a "good" starting point. Since the objective function (loss-cost function) of interest is complex, it is especially important to employ a good starting point to enhance the efficiency of the search technique.

Much experimentation with the optimization of this dynamic \bar{X} -control chart suggest that a good starting point is given by the optimum design of Duncan's corresponding \bar{X} -control chart. Therefore, rather than selecting the starting point for a dynamic \bar{X} -control chart optimization at random, the following approach is used to determine good initial starting conditions.

1. Construct a Duncan \bar{X} -control chart model which simplistically corresponds to the dynamic X-control chart of interest. That is, enter the same cost values and equivalent (as far as possible) distribution parameters into Duncan's model. For example, for a Weibull distribution with a mean of $\mu=100$ as used in the dynamic model, use an exponential parameter of $\lambda=1/\mu=.01$ in Duncan's model.

2. Optimize Duncan's model. The optimal design of his model constitutes a good starting point for optimization of the dynamic \bar{X} -chart.

Duncan's Model Optimization

The optimal design of Duncan's model is needed primarily to provide a good starting point for dynamic \bar{X} -control chart optimization. It is also required for the economic comparisons between dynamic and Duncan's \bar{X} -control chart optimal designs. The following optimization strategy, which makes use of a commercially available computer search technique ZXMIN [77], is designed to find the optimal design of Duncan's \bar{X} -control chart.

- 1. Using ZXMIN, optmize Duncan's (exact) cost model over all three variables--sampling interval, control limit spread, and sample size. That is, sample size is treated as a real-valued variable. The optimal real-valued sample size found in this stage, n_R^* , is close to the optimal integer-valued sample size, n^* .
- 2. Treat sample size as an integer variable. Start with an integer sample size, n', equal to $\lfloor n_R^* \rfloor 2$. (For this optimization strategy n' could have been set to $\lfloor n_R^* \rfloor 1$, $\lfloor n_R^* \rfloor$, etc.) If n' is less than 2, then set n' equal to 2, since a sample size smaller than 2 is not practically correct.
- 3. Fixing the integer sample size at n', use the ZXMIN routine to optimize Duncan's model over two variables--sampling interval and control limit spread. Call this optimum L_{n}^{\star} .
- 4. Fix the integer sample size at n'' = n' + IDIRC, where IDIRC is either +1 or -1 indicating whether the sample size is incremented or

decremented, respectively. (Originally IDIRC is set to +1.) Using the ZXMIN routine, optimize Duncan's model for this given sample size. Call this optimum L_{n}^{\star} .

- 5. Compare L_n^* , with $L_{n''}^*$. Note that Duncan's loss-cost function is a convex-like function and is unimodal. Therefore,
 - (i) If L_{n}^* , L_{n}^* , then n' is smaller than the optimal integer sample size, n*. Replace n' by n" and L_{n}^* , by L_{n}^* and go to step 4.
 - (ii) If $L_{n'}^*$ < $L_{n''}^*$, then n' is either greater than or equal to the optimal sample size, n*. In the latter case, stop the search and conclude that n* = n'. In the former case, set IDIRC = -1 so that the sample size will be decremented in all future steps, and go to step 4.

This optimization strategy is quite efficient in that it does not enumerate all integer values of sample size, starting from a sample size of 2, in order to find the minimum cost design of Duncan's model. This optimization strategy is implemented as part of the interactive computer program in Chapter V.

The ZXMIN computer optimization routine used in the above optimization strategy is based on a quasi-Newton method [23] and finds the minimum of a function of several variables. Quasi-Newton methods are a class of methods which use line search techniques in conjunction with a symmetric positive definite matrix approximating the inverse of the Hessian matrix of the function to be minimized [24]. ZXMIN is selected for use in the above optimization strategy because of reliable performance on all 25 of Duncan's examples. Since ZXMIN is a general optimization routine, it is necessary in the actual optimization program to

restrict the sample size to be greater than two (2), the control limit spread between 0.0 and 12.0 standard deviations of the sample average, and the sampling interval between 0.0 and 100.0. For practical purposes, these constraints do not put any limitations on the optimization of Duncan's \bar{X} -chart. In any case, these constraints can easily be modified.

In the case that the ZXMIN optimization routine is not available, the use of a reliable optimization routine such as the simplex method of Nelder and Mead [53] is recommended. Ready to use FORTRAN computer codes for the Nelder and Mead simplex method are available in [44] and [54].

Economic Comparisons Between the Dynamic \bar{X} -Control Chart and Duncan's \bar{X} -Control Chart

Examples for Comparison

To provide a comprehensive economic comparison between the dynamic \bar{X} -control chart and Duncan's \bar{X} -control chart, sixteen representative examples are considered, as shown in Table 4.3. Most of these examples are chosen from Duncan's paper [20]. The values of the costs and distribution parameters in this table cover a wide range of possibilities.

These sixteen examples are divided into four groups: 1 to 12, 16 to 20, 22, and 1b to 22b. In group 1 (δ = 2), example 1 is the base case, and the rest are its variations. In group 2 (δ = 1), example 16 is the base case, and example 20 is its variation. In group 3 (δ = .5) only example 22 is employed. In group 4 (η = 6), example 1b is the base case, and the rest are its variations. For all the examples in groups

TABLE 4.3

EXAMPLES CHOSEN FOR ECONOMIC COMPARISON

No.*	δ	λ	M	е	D	T	W	b	С	η	. θ	Characteristics	
1	2	.01	100	.05	2	50	25	.5	.1	3	.00892975	Basis for 1 to 12	
3		.03								3	.02678939	λ increases 3 times	
3a		.05								3	.04464898	λ increases 5 times	
5			1000							3	.00892975	M increases 10 times	
7				.50						3	.00892975	e increases 10 times	
8					20			1.0		3	.00892975	D increases 10 times	
10						500	250			3	.00892975	T and W increase 10 times	
11						5000	2500			3	.00892975	T and W increase 100 times	
12								5		3	.00892975	b increases 10 times	
16	1	.01	12.87	.05	2	50	25	.5	.1	3	.00892975	Basis for 16 and 20	
20									1	3	.00892975	c increases 20 times	
22	.5	.01	225	.05	2	50	25	.5	.1	3	.00892975	Basis; δ is .5	
1b	2	.01	100	.05	2	50	25	.5	.1	6	.00927719	Basis for 1b to 22b; n is 6	
3b		.03								6	.02783158	Same as 3 but n is 6	
16b	1	.01	12.87							6	.00927719	Same as 16 but n is 6	
22b	.5		225							6	.00927719	Same as 22 but n is 6	

^{*}All example numbers are the same as those used in Duncan's paper, with the exception of 1b, 3a, 3b, 16b, and 22b.

-

1, 2, and 3, the n parameter of the Weibull distribution used in the dynamic \bar{X} -chart model has a value of 3. For group 4 examples, the n parameter is set at 6. Note that in any case the 0 parameter of the Weibull is calculated so that both the Weibull distribution used in the dynamic \bar{X} -control chart and the exponential distribution used in Duncan's \bar{X} -control chart have equal means.

Analysis of Examples

For each of the examples in Table 4.3, two cases are investigated.

- Duncan's model optimum design is evaluated in the dynamic model to calculate the loss-cost incurred in the real environment as the result of incorrectly employing Duncan's model.
- 2. The dynamic model optimum design and its associated cost is calculated. The results are shown in Table 4.4.

To assure proper comparisons between dynamic optimal designs and Duncan's optimal designs, the following procedures are followed to obtain the results given in Table 4.4

- Exactly the same set of cost parameters are used in both the dynamic and Duncan's models.
- 2. Equivalent distributional parameters are used in both models. For example, assume that the parameter of the exponential distribution used in Duncan's model is λ = .01. If the n parameter of the corresponding Weibull distribution used in the dynamic model is equal to 3, then the θ parameter of the Weibull is calculated such that the mean of the Weibull becomes equal to the mean of the exponential; μ = $1/\lambda$ = 100. Therefore,

TABLE 4.4

OPTIMAL ECONOMIC DESIGNS OF DUNCAN'S AND THE DYNAMIC X-CHART AND THEIR COMPARISONS

No.	A*	In	nf	Ih	hf	Ik	kf	100L*	Percent Difference
1	DG DY	5 5	1.0 0.9989854	1.41 2.5400	1.0 0.9865	3.08 3.0960	1.0	399.260 376.428	-5.72
3	DG DY	4	1.0 0.9979963	0.78 1.4184	1.0 0.9776391	2.94 3.029461	1.0 0.999015	955.491 918.915	-3.83
3a	DP DY	4	1.0 0.9968417	0.63832 1.15525	1.0 0.9698815	2.9277 3.0189	1.0 0.998556	1430.602 1388.029	-2.98
5	DG DY	4 4	1.0 0.9995658	0.41 0.7099	1.0 0.9964308	2.95 3.05521	1.0 0.999783	2689.968 2613.273	-2.85
7	DG DY	2 2	1.0 1.0	0.94 1.6721	1.0 0.9912492	2.69 2.79	1.0 0.9996	536.531 505.313	-5.82
8	DG DY	5 5	1.0 0.9987293	1.62 2.96845	1.0 0.984931	3.05 3.14318	1.0 0.9991233	1835.502 1820.074	84
10	DD DY	6 6	1.0 0.9993572	1.3 2.5005	1.0 0.98691	3.80 3.730206	1.0 0.9998007	636.175 610.325	-4.06
11	DD DY	7 7	1.0 0.999458	1.30 2.8226	1.0 0.984977	4.4 4.27563	1.0 0.9998309	2833.764 2805.439	-1.0
12	DG DY	6	1.0 0.9978	3.47 6.23243	1.0 0.966713	2.88 2.902155	1.0 1.0	582.072 532.605	-8.50

Table 4.4 (Continued)

No.	A*	In	nf	Ih	hf	Ik	kf	100L*	Percent Difference
16	DG DY	14 14	1.0 0.9990475	5.47 9.89156	1.0 0.9477137	2.68 2.7631	1.0 0.997823	140.380 128.503	-8.46
20	DD DY	8	1.0 0.9945819	12. 0.22345	1.0 0.88224	1.9 2.0	1.0 0.990086	239.278 212.092	-11.36
22	DG DY	21 20	1.0 0.99980	1.3 2.3210	1.0 0.987893	2.11 2.1407	1.0 0.99981	1345.458 1252.671	-6.90
1b	DG DY	5 5	1.0 0.998341	1.41 3.6878	1.0 0.973497	3.08 3.2062	1.0 0.999039	399.300 358.790	10.14
3b	DG DY	4 4	1.0 0.9966622	0.78 2.1236	1.0 0.954179	2.94 3.0762	1.0 0.9978896	955.634 890.204	-6.80
16b	DG DY	14 14	1.0 0.9978122	5.47 14.3257	1.0 0.89696	2.68 2.8155	1.0 0.99545	140.404 118.973	-15.26
22b	DG DY	21 20	1.0 0.999894	1.3 3.4670	1.0 0.975064	2.11 2.21	1.0 0.998282	1346.700 1171.91	-12.98

A*: DD = Duncan's optimal design obtained using Duncan's model

DG = Goel's optimal design obtained using Duncan's model

DP = This research's optimal design obtained using Duncan's model

DY = The dynamic model's true optimal design

⁽The choice between DD and DG is based on the minimum cost criterion.

DP is used only when DD and DG are not available.)

¹⁰⁰L*: The true environment loss-cost in terms of dollars per 100 hours of operation.

$$1/\theta \Gamma (1 + 1/\eta) = 1/\lambda$$
 (4.11)

$$1/\theta \ \Gamma \ (1 + 1/3) = 100$$
 (4.12)

$$\theta = .00892975$$
 (4.13)

- 3. Duncan's optimal design is always implemented in the actual environment in which the process failure mechanism is governed by a Weibull distribution.
- 4. The same quantile value of .99 is used to calculate the loss-cost incurred in the real environment for both Duncan's and the dynamic \bar{X} -chart optimum design.

The economic comparisons summarized in Table 4.4 show that the dynamic \bar{X} -control chart design is always superior to Duncan's \bar{X} -chart design. The cost reduction provided by the dynamic model compared to Duncan's model varies depending on the particular situation at work. Note that as the mean of the process failure distribution decreases, the cost reduction becomes smaller. Similarly, when D is relatively large, the cost improvement is less significant. On the other hand, as the sampling cost increases, more significant cost improvement appears possible by the use of the dynamic model.

A comparison of group 4 examples, in which η is 6, with the examples of the first 3 groups, in which η is 3, shows that as the underlying real process failure mechanism differs more significantly from the exponential distribution, the cost improvement provided by the dynamic \bar{X} -chart design becomes more significant. It is interesting to note that in the above examples when η is doubled from 3 to 6, the percentage cost improvement is doubled.

In short, the significance of the process failure mechanism on the economic design of the \bar{X} -chart is well illustrated by this economic comparison. The optimal economic design of the \bar{X} -chart obtained using Duncan's model can be far from the true minimum cost design when the underlying process failure mechanism is not exponential. In this situation, a good knowledge of the environment and the use of the dynamic \bar{X} -chart design can provide significant cost savings.

Summary

Computational aspects of the model of Chapter III are discussed along with a special optimization algorithm developed for the optimal economic design of a dynamic \bar{X} -chart. The special search algorithm is based on the ideas of Davies, Swann, and Campey, Powell, and Coggins' procedure and the basic philosophy behind direct search techniques. It also makes use of a "good" starting point which is found by another strategy developed to optimize a corresponding Duncan's model.

Economic comparison between Duncan's and the dynamic \bar{X} -chart is performed. Sixteen representative examples covering a wide range of situations are selected. A majority of the examples are from Duncan's paper; other examples are employed to better investigate the cost implications of different process failure mechanisms. The results of this comparison are shown in Table 4.4. An analysis of these results shows that the minimum cost design obtained using Duncan's model can be far from the true minimum cost design when the true process failure mechanism is not exponential. Furthermore, as the true process failure mechanism differs more significantly from the exponential distribution,

the cost improvement provided by the dynamic $\overline{\textbf{X}}\text{-chart}$ design becomes more significant.

Ę,

CHAPTER V

USING THE INTERACTIVE COMPUTER PROGRAM

Introduction

Overview

ź.

This chapter presents the use of an interactive computer program which primarily implements the economic design of the dynamic \bar{X} -chart as is presented in previous chapters. It has the additional features of economic evaluation in a dynamic environment and economic design and evaluation of Duncan's \bar{X} -chart. The computer program provides the user with a versatile tool for economic design of \bar{X} -charts whether the process failure mechanism is exponential or Weibull.

The entire program is interactive in that the computer prompts the user for all necessary inputs. Care is taken to reduce the user's task of entering the parameters. Thus, almost all often-used values of inputs including the optimization technique parameters are automatically calculated in the program. These values are presented to the user for either verification or change. In addition, all the user's inputs are extensively checked for their appropriateness and the user is prompted to correct probable errors or inconsistencies. Only when a set of input has been checked by the program and verified by the user does the program continue.

When several values are to be entered, they only need be separated by a space or a comma. Integer values are entered without a decimal point. The input mechanism is virtually self-explanatory, as long as the user understands the terms being input as well as their mathematically feasible range. The latter is also extensively checked by the program for correctness. Thus any person, without previous familiarity with a computer and/or statistics, can easily use this program to economically design and/or evaluate a dynamic $\bar{\lambda}$ -control chart and compare it with the corresponding Duncan's $\bar{\lambda}$ -control chart.

In the remainder of this chapter, actual interactive output is interspersed with comments and explanation. The tasks performed by the program will be illustrated in depth. All computer outputs shown are automatically generated by the computer except for the input values which follow a question mark (?). These input values are entered by the user.

Getting Started

The interactive program performs: (1) Economic design of the dynamic \bar{X} -chart, (2) Economic evaluation of the dynamic \bar{X} -chart, (3) Economic design of Duncan's \bar{X} -chart, and (4) Economic evaluation of Duncan's \bar{X} -chart.

The program begins by presenting the main options menu (M.1). The selection of "1" from this menu indicates that the dynamic \bar{X} -control chart (Weibull process failure) is to be pursued.

```
##* ENTER OPTION NUMBER
1 = THE DYNAMIC X-BAR CHART(WEIBULL ENVIRONMENT)
2 = DUNCAN'S X-BAR CHART(EXPONENTIAL ENVIRONMENT)
3 = EXIT SYSTEM
?
1
```

Economic Design of the Dynamic \bar{X} -Chart

After the dynamic \bar{X} -chart is selected, the program prompts the user to enter the parameters of the Weibull distribution which represents the process failure mechanism. Note that only after the user confirms the validity of the input does the program ask about the cost and shift values. Then the major dynamic \bar{X} -chart options menu (M.2) is presented. A selection of "1" from this menu leads to the economic design of the dynamic \bar{X} -chart.

```
>> WEIBULL ENVIRONMENT <<
*** FOR THE DYNAMIC ECONOMIC X-BAR CHART.
    ENTER VALUES: THETA, ETA
.0089298
           3
VALUES ENTERED ARE:
DISTRIBUTION INFORMATION;
WEIBULL W/ THETA= 0.008930 AND ETA= 3.000000 => MEAN=
CORRECT ? 1=YES 2=NO 3=RETURN TO THE MAIN MENU
    ENTER VALUES: DELTA, B, C, D, E, M, T, W
   .5 .1 2. .05 12.87 50. 25.
                                                                           (M.2)
    COST AND OTHER INFORMATION:
      DELTA=
               1.0000 B=
                             0.5000 C=
                                               0.1000 D=
                                                              2.0000
                 0.0500 M=
                               12.8700 T=
                                              50.0000
                                                             25.0000
CORRECT ? 1=YES 2=NO 3=RETURN TO THE MAIN MENU
*** ENTER OPTION NUMBER
    1 = ECON, DESIGN OF THE DYNAMIC X-BAR CHART (OPTIMIZATION)
    2 = ECON. EVALUATION IN THE WEIBULL ENVIRONMENT
    3 = RETURN TO REVISE WEIBULL PARAMETERS
    4 = RETURN TO REVISE COST PARAMETERS
    5 = RETURN TO THE MAIN MENU
```

In the economic design of the dynamic \bar{X} -chart the program first automatically calculates a good starting point, as is described in Chapter IV, and prompts the user for acceptance or rejection of the point. Here, the user desires to input a starting point. Note how the program checks the user's input and prompts the user with helpful error messages. Then the program prompts the user with the calculated values of the optimization parameters and the preprogrammed values of the maximum numbers of loss-cost evaluations allowed in the three stages of the optimization. A menu is then presented so that any of these values can be changed to those of the user's preference. Here, the user wants to change the maximum number of loss-cost evaluations allowed.

```
*** FOR ECON. OPTIMIZATION OF THE DYNAMIC X-BAR CHART,
    THE FOLLOWING STARTING POINT IS SUGGESTED:
                    IH=
                                           2.6734
      IN= 14
                           5.4828 IK=
      NF=1.000000 HF=1.000000
                                    KF=1.0000000
    YOU ACCEPT THIS POINT.
CORRECT ?
           1=YES 2=NO 3=RETURN TO THE PREVIOUS MENU
FOR YOUR DESIRED STARTING POINT FOR OPTIMIZATION, ENTER:
    IN, IH, IK, NF, HF, KF
14 5.5 -2.7 1. 10.
!?! ERROR -- IK SHOULD BE BETWEEN O.O AND 12.
!?! ERROR -- NF, NH, AND NK SHOULD BE BETWEEN O.O AND 2.
DO IT OVER !
FOR YOUR DESIRED STARTING POINT FOR OPTIMIZATION, ENTER:
    IN, IH, IK, NF, HF, KF
14 5.5 2.7 1. 1. 1.
VALUES ENTERED: IN=
                      14
                                IH=
                                      5.5000
                                                         2.7000
                                                  TK=
                NF=1.0000000 HF=1.0000000
                                                 KF = 1.0000000
CORRECT ?
            1=YES 2=NO 3=RETURN TO THE PREVIOUS MENU
    QUANTILE VALUE OF 0.990000000 IS USED.
    YOU ACCEPT THIS.
CORRECT ?
          1=YES 2=NO 3=RETURN TO THE PREVIOUS MENU
```

```
OPTIMIZATION PARAMETERS:
                  HF
                        ISTEPS
                                   ΙK
                                              KF
                                                     IN
                                                              NF
                          2.
                                0.1000
STEP SIZE:
                                                           0.0069976
               0.001000
                                         0.0069976
                                                      1.
MIN STEP SIZE: 0.000100
                                0.1000
                                                           0.0017494
                          1.
                                         0.0017494
                                                      1.
MAX STEP SIZE: 0.004000
                                                           0.0069976
                                0.2000
                                         0.0069976
                                                      2.
                         32.
REQ PRECISION: 0.000050
                          1.
                                0.0500
                                         0.0011663
                                                           0.0011663
MAX. NUMBER OF ITERATIONS (LOSS-COST EVALUATIONS):
    ITRMX1= 45
                 ITRMX2= 25
                               ITRMX3= 25
***ENTER OPTION NUMBER:
    1 = ALL CORRECT, NO REVISION NEEDED
    2 = NEED TO REVISE OPTIMIZATION PARAMETERS
    3 = NEED TO REVISE MAX. NUMBER OF ITERATIONS
    4 = RETURN TO THE PREVIOUS MENU
```

Note that the user's input is checked, commented, and then presented for verification. Then, the optimization output follows. All the distribution, cost, and other appropriate information entered before is summarized in the optimization output for easy reference. Finally, the optimum design and its associated loss-cost per 100 hours of operations are printed.

```
ENTER VALUES: ITRMX1, ITRMX2, ITRMX3
?
95
    25 -25
!?! ERROR -- THE MAX. NUMBER OF ITERATIONS SHOULD BE AT LEAST 1.
DO IT OVER !
    ENTER VALUES: ITRMX1, ITRMX2, ITRMX3
          25
95
    OPTIMIZATION PARAMETERS:
                  HF
                        ISTEPS
                                   ΙK
                                              KF
                                                     IN
                                                              NF
STEP SIZE:
               0.001000
                          2.
                                0.1000
                                          0.0069976
                                                      1.
                                                           0.0069976
MIN STEP SIZE: 0.000100
                                 0.1000
                                          0.0017494
                                                           0.0017494
                          1.
                                                      1.
MAX STEP SIZE: 0.004000
                                          0.0069976
                                                           0.0069976
                         32.
                                 0.2000
                                                      2.
REQ PRECISION: 0.000050
                                 0.0500
                                          0.0011663
                                                           0.0011663
MAX. NUMBER OF ITERATIONS (LOSS-COST EVALUATIONS):
    ITRMX1= 95
                 ITRMX2= 25
***ENTER OPTION NUMBER:
    1 = ALL CORRECT, NO REVISION NEEDED
    2 = NEED TO REVISE OPTIMIZATION PARAMETERS
    3 = NEED TO REVISE MAX. NUMBER OF ITERATIONS
    4 = RETURN TO THE PREVIOUS MENU
?
```

```
DISTRIBUTION INFORMATION;
     WEIBULL W/ THETA= 0.008930 AND ETA= 3.000000 => MEAN=
   COST AND OTHER INFORMATION:
             1.0000 B=
                         0.5000 C=
                                    0.1000 D=
                                                 2.0000
     DELTA=
                        12.8700 T=
             0.0500 M=
                                    50.0000 W=
        F=
   STARTING POINT FOR OPTIMIZATION IS:
     IN= 14
                IH=
                      5.5000 IK=
                                   2.7000
     NF=1.0000000 HF=1.0000000
                             KF=1.0000000
   QUANTILE VALUE IS: 0.99000000
   MAX. NUMBER OF ITERATIONS (LOSS-COST EVALUATIONS):
     ITRMX1= 95
               ITRMX2= 25
                           ITRMX3= 25
CHECK THE ABOVE INFORMATION.
EVERYTHING IS CORRECT ? 1=YES 2=NO
               ** THE OPTIMAL DYNAMIC DESIGN IS: **
     IN= 14
                IH=
                     11.6400 IK=
                                  2.7000
     NF=0.9987366 HF=0.9483984
                             KF=0.9991310
****** LOSS-COST PER 100 HOURS = $
                                   128.507
********************
DO YOU WANT TO EMPLOY ANOTHER PASS OF OPTIMIZATION.
STARTING WITH THE BEST SOLUTION FOUND SO FAR?
1=YES 2=NO, RETURN TO THE PREVIOUS MENU
2
```

Economic Evaluation of the Dynamic \bar{X} -Chart

A selection of "2" from the dynamic \bar{X} -chart menu (M.2) leads to the economic evaluation of this chart. Note that nf, hf, and kf values are equal to one ensuring that sample sizes, sampling intervals, and control limits stay constant throughout the chart's operation. In fact, this design to be evaluated is the optimal design obtained using Duncan's model (see example 16 in [33]). The final loss-cost for this design using a quantile of .999 is a number in the range of \$141.246 to \$141.279. Note that the exact cost figure cannot be given because the dynamic model is unable to simultaneously satisfy both of the user's requirements of maintaining the initial sampling interval and the quantile exactly at 5.5 and .999, respectively. Note that if the same quantile value of .99, as in the optimization were used, this economic

evaluation of the Duncan's optimal design implemented in the Weibull environment could be correctly compared against the dynamic \bar{X} -chart design for the cost saving provided by the dynamic design. This discussion is true if it is known that the real process failure mechanism is characterized by the Weibull rather than the exponential distribution.

```
*** ENTER OPTION NUMBER
   1 = ECON. DESIGN OF THE DYNAMIC X-BAR CHART (OPTIMIZATION)
   2 = ECON. EVALUATION IN THE WEIBULL ENVIRONMENT
   3 = RETURN TO REVISE WEIBULL PARAMETERS
   4 = RETURN TO REVISE COST PARAMETERS
   5 = RETURN TO THE MAIN MENU
*** FOR ECON. EVALUATION IN THE WEIBULL ENVIRONMENT,
   ENTER:
   IN, IH, IK, NF, HF, KF
14 5.5 2.7
             1.
                 1.
VALUES ENTERED: IN=
                                   5.5000
                    14
                                               TK=
                                                     2.7000
                             IH=
               NF=1.0000000 HF=1.0000000
                                              KF=1.0000000
CORRECT ?
           1=YES 2=NO 3=RETURN TO THE PREVIOUS MENU
!?! ERROR -- DO IT OVER !
CORRECT ? 1=YES 2=NO 3=RETURN TO THE PREVIOUS MENU
QUANTILE VALUE OF 0.99000000 IS USED.
YOU ACCEPT THIS.
CORRECT ? 1=YES 2=NO 3=RETURN TO THE PREVIOUS MENU
ENTER YOUR DESIRED QUANTILE:
.999
QUANTILE VALUE OF 0.99900000 IS USED.
YOU ACCEPT THIS.
CORRECT ? 1=YES 2=NO 3=RETURN TO THE PREVIOUS MENU
******* ECON. EVALUATION IN WEIBULL ENVIRONMENT *********
   DISTRIBUTION INFORMATION;
     WEIBULL W/ THETA= 0.008930 AND ETA= 3.000000 => MEAN= 99.9999
   COST AND OTHER INFORMATION:
     DELTA=
               1.0000 B=
                            0.5000 C=
                                          0.1000 D=
                                                        2.0000
               0.0500 M=
                            12.8700 T=
        E =
                                        50.0000 W=
                                                       25.0000
   *** THE DESIGN TO BE EVALUATED IS: ***
                   IH=
                         5.5000 IK=
     NF=1.0000000 HF=1.0000000
                                 KF=1.0000000
   QUANTILE VALUE IS:0.999000000
CHECK THE ABOVE INFORMATION.
EVERYTHING IS CORRECT ? 1=YES 2=NO
```

```
*** FOR THE FOLLOWING DESIGN QUANTILE IS FIXED AT 0.99900000
    IN= 14 IH= 5.6125 IK=
                               2.7000
    NF=1.0000000 HF=1.0000000 KF=1.0000000
  LOSS-COST PER 100 HOURS = $
                          141,279
  *** FOR THE FOLLOWING DESIGN QUANTILE IS FIXED AT 0.99900000
    LOSS-COST PER 100 HOURS = $
                          141,246
  *** FOR THE FOLLOWING DESIGN, THE ACTUAL QUANTILE IS 0.99949054
    IN= 14 IH= 5.5000 IK= 2.7000
NF=1.0000000 HF=1.0000000 KF=1.0000000
  LOSS-COST PER 100 HOURS = $
                           141.295
*******************
*******************
```

The following interactive procedure and output illustrates the use of options 3 and 4 of the dynamic \bar{X} -chart menu and the convenience they provide in updating the distribution and cost information. Finally, the selection of option 5 leads to the main menu.

```
*** ENTER OPTION NUMBER
    1 * ECON. DESIGN OF THE DYNAMIC X-BAR CHART (OPTIMIZATION)
    2 = ECON. EVALUATION IN THE WEIBULL ENVIRONMENT
3 = RETURN TO REVISE WEIBULL PARAMETERS
    4 = RETURN TO REVISE COST PARAMETERS
    5 = RETURN TO THE MAIN MENU
   ENTER VALUES: THETA, ETA
.0092772 6.
VALUES ENTERED ARE:
    DISTRIBUTION INFORMATION;
      WEIBULL W/ THETA= 0.009277 AND ETA= 6.000000 => MEAN= 99.9999
CORRECT ? 1=YES 2=NO 3=RETURN TO THE MAIN MENU
*** ENTER OPTION NUMBER
    1 = ECON. DESIGN OF THE DYNAMIC X-BAR CHART (OPTIMIZATION)
    2 = ECON. EVALUATION IN THE WEIBULL ENVIRONMENT
    3 * RETURN TO REVISE WEIBULL PARAMETERS
    4 = RETURN TO REVISE COST PARAMETERS
   5 = RETURN TO THE MAIN MENU
*** FOR ECON. OPTIMIZATION OF THE DYNAMIC X-BAR CHART,
    THE FOLLOWING STARTING POINT IS SUGGESTED:
               IH= 5.4817 IK= 2.6736
     IN= 14
     NF=1.0000000 HF=1.0000000 KF=1.0000000
    YOU ACCEPT THIS POINT.
CORRECT ? 1=YES 2=NO 3=RETURN TO THE PREVIOUS MENU
1
   QUANTILE VALUE OF 0.990000000 IS USED.
    YOU ACCEPT THIS.
```

```
CORRECT ? 1=YES 2=NO 3=RETURN TO THE PREVIOUS MENU
1
   OPTIMIZATION PARAMETERS:
                HF ISTEPS
                               ΙK
                                         KF.
                                                IN
STEP SIZE:
             0.001000
                            0.1000 0.0093410
                                                1. 0.0093410
                       2.
                       1. 0.1000 0.0023353
32. 0.2000 0.0093410
MIN STEP SIZE: 0.000100
                                                1.
2.
                                                      0.0023353
MAX STEP SIZE: 0.004000 32.
                                                      0.0093410
                            0.0500 0.0015568 1.
REQ PRECISION: 0.000050 1.
                                                      0.0015568
MAX. NUMBER OF ITERATIONS (LOSS-COST EVALUATIONS):
   ***ENTER OPTION NUMBER:
   1 = ALL CORRECT, NO REVISION NEEDED
   2 = NEED TO REVISE OPTIMIZATION PARAMETERS
   3 = NEED TO REVISE MAX. NUMBER OF ITERATIONS
   4 = RETURN TO THE PREVIOUS MENU
******* ECON. DESIGN OF THE DYNAMIC X-BAR CHART ***********
   DISTRIBUTION INFORMATION;
     WEIBULL W/ THETA= 0.009277 AND ETA= 6.000000 => MEAN= 99.9999
    COST AND OTHER INFORMATION:
     DELTA= 1.0000 B= 0.5000 C=
E= 0.0500 M= 12.8700 T=
                                         O.1000 D=
                                                      2.0000
                                       50.0000 W=
   STARTING POINT FOR OPTIMIZATION IS:
     IN= 14 IH= 5.4817 IK= 2.6736
NF=1.0000000 HF=1.0000000 KF=1.0000000
                                       2.6736
   QUANTILE VALUE IS: 0.99000000
   MAX. NUMBER OF ITERATIONS (LOSS-COST EVALUATIONS):
     CHECK THE ABOVE INFORMATION.
EVERYTHING IS CORRECT ? 1=YES 2=NO
?
1
                 ** THE OPTIMAL DYNAMIC DESIGN IS: **
     IN= 14 IH= 15.3832 IK= 2.8736
NF=0.9976232 HF=0.8965787 KF=0.9928866
******* LOSS-COST PER 100 HOURS = $
                                      118.898
************************
DO YOU WANT TO EMPLOY ANOTHER PASS OF OPTIMIZATION,
STARTING WITH THE BEST SOLUTION FOUND SO FAR?
1=YES 2=NO, RETURN TO THE PREVIOUS MENU
******** ECON. DESIGN OF THE DYNAMIC X-BAR CHART **********
   DISTRIBUTION INFORMATION;
     WEIBULL W/ THETA= 0.009277 AND ETA= 6.000000 => MEAN= 99.9999
    COST AND OTHER INFORMATION:
    DELTA= 1.0000 B= 0.5000 C=
E= 0.0500 M= 12.8700 T=
                                       O.1000 D=
50.0000 W=
                                                      2.0000
                                                     25.0000
   STARTING POINT FOR OPTIMIZATION IS:
     IN= 14
                 IH= 15.3832 IK=
                                     2.8736
```

```
NF=0.9976232 HF=0.8965787 KF=0.9928866
    QUANTILE VALUE IS: 0.99000000
    MAX. NUMBER OF ITERATIONS (LOSS-COST EVALUATIONS):
     CHECK THE ABOVE INFORMATION.
EVERYTHING IS CORRECT ? 1=YES 2=NO
10
!?! ERROR -- DO IT OVER !
CHECK THE ABOVE INFORMATION.
EVERYTHING IS CORRECT ? 1=YES 2=NO
1
                 ** THE OPTIMAL DYNAMIC DESIGN IS: **
      IN= 14 IH= 15.3832 IK= 2.8736
NF=0.9976232 HF=0.8965787 KF=0.9926570
****** LOSS-COST PER 100 HOURS = $
                                      118.897
************************
*************************
DO YOU WANT TO EMPLOY ANOTHER PASS OF OPTIMIZATION,
STARTING WITH THE BEST SOLUTION FOUND SO FAR?
1=YES 2=NO, RETURN TO THE PREVIOUS MENU
*** ENTER OPTION NUMBER
    1 = ECON. DESIGN OF THE DYNAMIC X-BAR CHART (OPTIMIZATION)
    2 = ECON. EVALUATION IN THE WEIBULL ENVIRONMENT
    3 = RETURN TO REVISE WEIBULL PARAMETERS
4 = RETURN TO REVISE COST PARAMETERS
    5 = RETURN TO THE MAIN MENU
    ENTER VALUES: DELTA, B, C, D, E, M, T, W
?
    .5 1. 2. .05 12.87 50. 25.
    COST AND OTHER INFORMATION:
     DELTA= 1.0000 B=
                            0.5000 C=
                                         1.0000 D=
                                                       2.0000
E= 0.0500 M= 12.8700 T= 50.0000 CORRECT ? 1=YES 2=NO 3=RETURN TO THE MAIN MENU
                                        50.0000 W=
                                                      25.0000
*** ENTER OPTION NUMBER
   1 = ECON. DESIGN OF THE DYNAMIC X-BAR CHART (OPTIMIZATION)
   2 = ECON. EVALUATION IN THE WEIBULL ENVIRONMENT
   3 = RETURN TO REVISE WEIBULL PARAMETERS
   4 = RETURN TO REVISE COST PARAMETERS
   5 = RETURN TO THE MAIN MENU
```

```
*** FOR ECON. EVALUATION IN THE WEIBULL ENVIRONMENT.
     ENTER:
     IN, IH, IK, NF, HF, KF
 14 5.5 2.7 1. 1. 1. VALUES ENTERED: IN= 14
                    IN= 14 IH= 5.5000
NF=1.000000 HF=1.0000000
                                                      IK= 2.7000
KF=1.0000000
              1=YES 2=NO 3=RETURN TO THE PREVIOUS MENU
 ?
 QUANTILE VALUE OF 0.990000000 IS USED.
 YOU ACCEPT THIS.
 CORRECT ? 1=YES 2=NO 3=RETURN TO THE PREVIOUS MENU
DISTRIBUTION INFORMATION;
WEIBULL W/ THETA= 0.009277 AND ETA= 6.000000 => MEAN= 99.9999
     COST AND OTHER INFORMATION:
    DELTA= 1.0000 B= 0.5000 C= 1.0000 D=
E= 0.0500 M= 12.8700 T= 50.0000 W=

*** THE DESIGN TO BE EVALUATED IS: ***
IN= 14 IH= 5.5000 IK= 2.7000
NF=1.0000000 HF=1.0000000 KF=1.0000000
                                                                  2.0000
                                                                 25,0000
    QUANTILE VALUE IS:0.99000000
CHECK THE ABOVE INFORMATION.
EVERYTHING IS CORRECT ? 1=YES 2=NO
?
1
    *** FOR THE FOLLOWING DESIGN QUANTILE IS FIXED AT 0.99000000
      IN= 14 IH= 5.5614 IK= 2.7000
NF=1.0000000 HF=1.0000000 KF=1.0000000
                                      364.735
     LOSS-COST PER 100 HOURS = $
    *** FOR THE FOLLOWING DESIGN QUANTILE IS FIXED AT 0.99000000
       IN= 14 IH= 5.3475 IK= 2.7000
NF=1.0000000 HF=1.0000000 KF=1.0000000
     LOSS-COST PER 100 HOURS = $
                                       373.708
    *** FOR THE FOLLOWING DESIGN, THE ACTUAL QUANTILE IS 0.99892659
       IN= 14 IH= 5.5000 IK= 2.7000
NF=1.0000000 HF=1.0000000 KF=1.0000000
     LOSS-COST PER 100 HOURS = $ 370.088
*** ENTER OPTION NUMBER
    1 * ECON. DESIGN OF THE DYNAMIC X-BAR CHART (OPTIMIZATION)
    2 = ECON. EVALUATION IN THE WEIBULL ENVIRONMENT
    3 = RETURN TO REVISE WEIBULL PARAMETERS
    4 = RETURN TO REVISE COST PARAMETERS
    5 = RETURN TO THE MAIN MENU
5
```

Economic Design of Duncan's X-Chart

The selection of "2" from the main menu indicates that Duncan's \bar{X} -chart (exponential process failure) is to be pursued. Once accesssed, the user is first prompted for the values of the distribution, shift, and cost parameters used in Duncan's economic \bar{X} -chart. After proper verification, Duncan's \bar{X} -chart menu (M.3) is presented. A selection of "1" from this menu leads to the economic \bar{X} -chart design.

```
==> MAIN MENU <==
*** ENTER OPTION NUMBER
    1 = THE DYNAMIC X-BAR CHART(WEIBULL ENVIRONMENT)
    2 = DUNCAN'S X-BAR CHART(EXPONENTIAL ENVIRONMENT)
    3 = EXIT SYSTEM
     >> EXPONENTIAL ENVIRONMENT <<
*** FOR DUNCAN'S ECONOMIC X-BAR CHART, ENTER VALUES:
   LAMBDA, DELTA, B, C, D, E, M, T, W
VALUES ENTERED ARE:
                                                                        (M.3)
.01 1. .5 .1 2.
                    .05 12.87 50. 25.
    DISTRIBUTION INFORMATION;
     EXPONENTIAL W/ LAMBDA=
                                 0.0100 => MEAN= 100.0000
    COST AND OTHER INFORMATION:
               1.0000 B=
     DELTA=
                             0.5000 C=
                                            0.1000 D=
                                                          2,0000
         E=
                0.0500 M=
                             12.8700 T=
                                           50.0000 W=
                                                         25.0000
*** ENTER OPTION NUMBER
    1 = ECON. DESIGN OF DUNCAN, S X-BAR CHART (OPTIMIZATION)
   2 = ECON. EVALUATION IN THE EXPONENTIAL ENVIRONMENT
   3 * RETURN TO REVISE COST AND DISTRIBUTION PARAMETERS
   4 = RETURN TO THE MAIN MENU
?
```

Then the user is prompted with the values of a starting point suggested by the program. These values can and are changed by the user's request to those of his preference. After proper verification, the optimization is performed and the optimal design of Duncan's \bar{X} -chart and its associated loss-cost per 100 hours of operation are printed.

```
*** FOR ECON. OPTIMIZATION OF DUNCAN'S X-BAR CHART,
    THE FOLLOWING STARTING POINT IS SUGGESTED:
    N= 5 H= 1.0000 K=
YOU ACCEPT THIS POINT.
          1=YES 2=NO 3=RETURN TO THE PREVIOUS MENU
!?! ERROR -- DO IT OVER !
CORRECT ? 1=YES 2=NO 3=RETURN TO THE PREVIOUS MENU
    FOR YOUR DESIRED STARTING POINT FOR OPTIMIZATION, ENTER:
   1. 3.
    VALUES ENTERED: N=
                        4
                                               K= 3.0000
                               H= 1.0000
CORRECT ? 1=YES 2=NO 3=RETURN TO THE PREVIOUS MENU
****** ECON. DESIGN OF DUNCAN'S X-BAR CHART *********
VALUES ENTERED ARE:
    DISTRIBUTION INFORMATION;
      EXPONENTIAL W/ LAMBDA=
                                0.0100 => MEAN= 100.0000
    COST AND OTHER INFORMATION:
      DELTA= 1.0000 B= 0.5000 C=
                                          0.1000 D=
                                                        2.0000
                                         50.0000 W=
         E≖
               0.0500 M=
                            12.8700 T=
                                                       25.0000
   STARTING POINT FOR OPTIMIZATION IS:
    N= 4 H= 1.0000 K= 3.0000
CHECK THE ABOVE INFORMATION.
EVERYTHING IS CORRECT ? 1=YES 2=NO
        ** THE OPTIMAL DUNCAN'S DESIGN IS: **
         N= 14 H= 5.4813 K= 2.6723
****** THE MIN. LOSS-COST PER 100 HOURS = $ 141.593
```

Economic Evaluation of Duncan's \bar{X} -Chart

A selection of "2" from menu (M.3) leads to the economic evaluation of Duncan's \bar{X} -chart. The program carries the cost information entered before and proceeds to prompt the user for the values of the design to be evaluated. Then the program prints a summary of the distribution and cost information along with the design to be evaluated. Upon verification of this information, the economic evaluation of Duncan's \bar{X} -chart is performed and the resulting loss-cost is printed.

```
*** ENTER OPTION NUMBER
    1 * ECON. DESIGN OF DUNCAN, S X-BAR CHART (OPTIMIZATION)
    2 = ECON. EVALUATION IN THE EXPONENTIAL ENVIRONMENT
    3 = RETURN TO REVISE COST AND DISTRIBUTION PARAMETERS
    4 = RETURN TO THE MAIN MENU
*** FOR ECON. EVALUATION IN THE EXPONENTIALENVIRONMENT,
    ENTER VALUES: N, H, K
VALUES ENTERED: N= 4 H= 10.0000 K= 3
CORRECT ? 1=YES 2=NO 3=RETURN TO THE PREVIOUS MENU
                                                 K= 3.0000
*** FOR ECON. EVALUATION IN THE EXPONENTIALENVIRONMENT.
    ENTER VALUES: N, H, K
  1. 3.
    VALUES ENTERED: N= 4
                                 H≃ 1.0000
                                                 K= 3.0000
CORRECT ? 1=YES 2=NO 3=RETURN TO THE PREVIOUS MENU
DISTRIBUTION INFORMATION;
      EXPONENTIAL W/ LAMBDA=
                                 0.0100 => MEAN= 100.0000
    COST AND OTHER INFORMATION:
    DELTA= 1.0000 B= 0.5000 C= 0.1000 D= 2.0000 E= 0.0500 M= 12.8700 T= 50.0000 W= 25.0000 *** THE DESIGN TO BE EVALUATED IS: ***
     N= 4 H= 1.0000 K= 3.0000
CHECK THE ABOVE INFORMATION.
EVERYTHING IS CORRECT ? 1=YES 2=NO
****** LOSS-CAST PER 100 HOURS= $ 220.959
```

The following interactive procedure and output illustrates the use of option 3 of Duncan's \bar{X} -chart menu. This option is employed to change the distribution and cost information for conveniently repeating any of the options 1 and 2 of the menu. After the user has performed all the desired economic designs and evaluations, option 4 of menu (M.3) is selected to return to the main menu. In the main menu, a selection of "3" ends the execution of the interactive computer program.

```
*** ENTER OPTION NUMBER

1 = ECON. DESIGN OF DUNCAN,S X-BAR CHART (OPTIMIZATION)

2 = ECON. EVALUATION IN THE EXPONENTIAL ENVIRONMENT

3 = RETURN TO REVISE COST AND DISTRIBUTION PARAMETERS

4 = RETURN TO THE MAIN MENU

?
```

```
>> EXPONENTIAL ENVIRONMENT <<
*** FOR DUNCAN'S ECONOMIC X-BAR CHART, ENTER VALUES:
   LAMBDA, DELTA, B, C, D, E, M, T, W
.01 1. .5 1. 2. .05 12.87 50. 25.
VALUES ENTERED ARE:
   DISTRIBUTION INFORMATION:
                               0.0100 => MEAN= 100.0000
     EXPONENTIAL W/ LAMBDA=
   COST AND OTHER INFORMATION:
     DELTA= 1.0000 B= 0.5000 C= 1.0000 D=
                                                       2.0000
              0.0500 M= 12.8700 T= 50.0000 W= 25.0000
*** ENTER OPTION NUMBER
   1 = ECON. DESIGN OF DUNCAN, S X-BAR CHART (OPTIMIZATION)
   2 = ECON. EVALUATION IN THE EXPONENTIAL ENVIRONMENT
   3 = RETURN TO REVISE COST AND DISTRIBUTION PARAMETERS
   4 = RETURN TO THE MAIN MENU
*** FOR ECON. OPTIMIZATION OF DUNCAN'S X-BAR CHART.
   THE FOLLOWING STARTING POINT IS SUGGESTED:
   N= 5 H= 1.0000 K= 3.0000 YOU ACCEPT THIS POINT.
CORRECT ? 1=YES 2=NO 3=RETURN TO THE PREVIOUS MENU
******* ECON. DESIGN OF DUNCAN'S X-BAR CHART *********
VALUES ENTERED ARE:
   DISTRIBUTION INFORMATION;
     EXPONENTIAL W/ LAMBDA=
                              0.0100 => MEAN= 100.0000
   COST AND OTHER INFORMATION:
            1.0000 B= 0.5000 C= 1.0000 D= 0.0500 M= 12.8700 T= 50.0000 W=
     DELTA=
                                                       2.0000
               0.0500 M=
        F=
                                                      25.0000
   STARTING POINT FOR OPTIMIZATION IS:
    N= 5 H= 1.0000 K= 3.0000
CHECK THE ABOVE INFORMATION.
EVERYTHING IS CORRECT ? 1=YES 2=NO
?
        ** THE OPTIMAL DUNCAN'S DESIGN IS: **
         N= 8 H= 12.3805 K= 1.8827
****** THE MIN. LOSS-COST PER 100 HOURS = $ 242.071
*****************
*** ENTER OPTION NUMBER
   1 = ECON. DESIGN OF DUNCAN, S X-BAR CHART (OPTIMIZATION)
   2 * ECON. EVALUATION IN THE EXPONENTIAL ENVIRONMENT
   3 = RETURN TO REVISE COST AND DISTRIBUTION PARAMETERS
    4 = RETURN TO THE MAIN MENU
   ==> MAIN MENU <==
*** ENTER OPTION NUMBER
    1 = THE DYNAMIC X-BAR CHART(WEIBULL ENVIRONMENT)
   2 = DUNCAN'S X-BAR CHART(EXPONENTIAL ENVIRONMENT)
   3 = EXIT SYSTEM
```

Summary

Almost all the features of the interactive computer program are illustrated in this chapter. Several examples are given which describe the capabilities of this computer program. The interactive and user-oriented features of this program make it a flexible and convenient tool for the economic design of an \bar{X} -chart whether the process failure mechanism is exponential or Weibull. It allows any person without even previous familiarity with a computer and/or statistics to practically use and benefit from the results of this research. As such it will help the faster implementation of the dynamic \bar{X} -chart in practice, as well as the broader applications of \bar{X} -control charts.

CHAPTER VI

SUMMARY AND CONCLUSION

ħ

Control charting is an important part of statistical quality control which can be used to achieve the quality objectives with the least possible cost. This research extends the state of the art in control charting by fulfilling the objective and all the subobjectives of Chapter I. That is:

- 1. A dynamic \bar{X} -control chart methodology in which sample sizes, intervals between samples, and control limit widths are dynamic has been originated.
- 2. A generalized dynamic version of Duncan's \bar{X} -chart model in which the process failure mechanism can be of any form has been formulated. This formulation follows the same cost structure as in Duncan's classic economic \bar{X} -chart model.
- 3. A special process failure mechanism represented by the Weibull distribution, an important distribution in reliability engineering, has been assumed and incorporated into the generalized dynamic model along with a special control chart methodology.
- 4. A general strategy together with a special computer search technique has been developed to decide on the appropriate values of the sample sizes, intervals between samples, and control limit widths for the dynamic \bar{X} -chart. This

optimization strategy is based on the use of a "good" initial starting point in conjunction with a search routine which makes use of the ideas of Davies, Swann, and Campey, Powell, and Coggin's procedures and the basic philosophy behind direct search techniques.

- 5. Economic design of the dynamic and Duncan's \bar{X} -charts have been compared under a variety of situations. The effect of process failure mechanisms which are characterized by the Weibull distribution, rather than the exponential, have been investigated.
- 6. A versatile, comprehensive, interactive computer program has been developed and described. This program implements the economic design and evaluation of (1) the dynamic \bar{X} -chart; Weibull process failure, and (2) Duncan's \bar{X} -control chart; exponential process failure.

Based on the results obtained in this research, the dynamic \bar{X} -control chart design is always superior to Duncan's \bar{X} -control chart design when the true process failure is Weibull. The cost reduction provided by the dynamic model compared to Duncan's model varies depending on the particular situation at work. It is observed that as the mean of the process failure distribution decreases, the cost reduction becomes smaller. On the other hand, as the underlying process failure distribution differs more significantly from the exponential distribution, this cost reduction becomes larger.

BIBLIOGRAPHY

- [1] Aroian, L. A. and H. Levane. "The Effectiveness of Quality Control Charts." Journal of the American Statistical Association, 45, 252 (Dec., 1950), 520-529.
- [2] Baker, K. R. "Two Process Models in the Economic Design of an X-Chart." <u>AIIE Transactions</u>, 3, 4 (Dec., 1971), 257-263.
- [3] Barish, N. N. and N. Hauser. "Economic Design for Control Decisions." <u>Journal of Industrial Engineering</u>, 14, 3 (1963), 125-134.
- [4] Barnard, G. A. "Control Charts and Stochastic Processes."

 Journal of the Royal Statistical Society, Series B, 21, 2

 (1959), 239-271.
- [5] Bather, J. A. "Control Charts and the Minimization of Costs."

 Journal of the Royal Statistical Society, Series B, 25, 1

 (1963), 49-80.
- [6] Beale, E. M. L., "On an Iterative Method of Finding a Local Minimum of a Function of More Than One Variable," Princeton University Research Group Technical Report 25, (Nov., 1958).
- [7] Burr, I. W. "The Effect of Non-Normality on Constants for \bar{X} and R Charts." Industrial Quality Control, 23, 11 (May, 1967), 563-569.
- [8] Case, K. E., J. W. Schmidt and G. K. Bennett. "A Discrete Economic Multiattribute Acceptance Sampling." AIIE Transactions, 7, 4 (Dec., 1975), 363-369.
- [9] Case, K. E. and G. K. Bennett. "The Economic Effect of Measurement Error on Variables Acceptance Sampling." <u>International</u> <u>Journal of Production Research</u>, 15, 2 (1977), 117-128.
- [10] Case, K. E. and L. L. Jones. <u>Profit Through Quality-A Quality Assurance Program for Manufacturers</u>. AIIE Monograph Series. Norcross, Georgia: American Institute of Industrial Engineers, 1978.
- [11] Chiu, W. K. "Comments on the Economic Design of X-Charts."

 Journal of the American Statistical Association, 68, 344

 (Dec., 1973), 919-921.

- [12] Chiu, W. K. and G. B. Wetherill. "A Simplified Scheme for the Economic Design of X-Charts." <u>Journal of Quality Technology</u>, 6, 2 (April, 1974), 63-69.
- [13] Chiu, W. K. and G. B. Wetherill. "Quality Control Practices."

 <u>International Journal of Production Research</u>, 13, 2 (1975),

 175-182.
- [14] Chiu, W. K. "Economic Design of Attribute Control Charts." Technometrics, 17, 1 (Feb., 1975), 81-87.
- [15] Chiu, W. K. "On the Estimation of Data Parameters for Economic Optimum X-Charts." Metrika, 23, 3 (Sept., 1976), 135-147.
- [16] Chiu, W. K. and C. H. Cheung. "An Economic Study of X-Charts with Warning Limits." <u>Journal of Quality Technology</u>, 9, 4 (Oct., 1977), 166-171.
- [17] Cowden, D. J. <u>Statistical Methods in Quality Control</u>. Englewood Cliffs: Prentice Hall, Inc., 1957.
- [18] Craig, C. C. "The \bar{X} and R-Charts and Its Competitors." <u>Journal</u> of Quality Technology, 1, 2 (April, 1969), 102-104.
- [19] Drury, C. G. and J. G. Fox. <u>Human Reliability in Quality Control</u>. London: Taylor and Francis Ltd., 1975.
- [20] Duncan, A. J. "The Economic Design of X-Charts Used to Maintain Current Control of a Process." <u>Journal of the American Statistical Association</u>, 11, 274 (June, 1956), 228-242.
- [21] Duncan, A. J. "The Economic Design of \bar{X} -Charts When There is a Multiplicity of Assignable Causes." <u>Journal of the American</u> Statistical Association, 66, 333 (March, 1971), 107-121.
- [22] Duncan, A. J. <u>Quality Control and Industrial Statistics</u>. 4th Ed. Homewood, Ill.: Richard D. Irwin, Inc., 1974.
- [23] Fletcher, R. "Fortran Subroutines for Minimization by Quasi-Newton Methods." Report R7125 AERE, Harwell, England, (June 1972).
- [24] Fletcher, R. <u>Practical Methods of Optimization</u>. Vol. 1: Unconstrained Optimization, John Wiley & Sons, Ltd., 1980.
- [25] Fruend, R. A. "Acceptance Control Charts." <u>Industrial Quality</u> <u>Control</u>, 14, 4 (Oct., 1957), 13-23.
- [26] Fruend, R. A. "A Reconsideration of the Variables Control Chart."

 Industrial Quality Control, 16, 11 (May, 1960), 35-41.
- [27] Fruend, R. A. "Graphical Process Control." <u>Industrial Quality</u> <u>Control</u>, 18, 7 (Jan., 1962), 15-22.

- [28] Ghare, P. M. and P. E. Torgersen. "The Multicharacteristic Control Chart." <u>Journal of Industrial Engineering</u>, 19, 6 (June, 1968), 269-272.
- [29] Gibra, I. N. "Optimal Control of Processes Subject to Linear Trends." <u>Journal of Industrial Engineering</u>, 18, 1 (1967), 35-41.
- [30] Gibra, I. N. "Economically Optimal Determination of the Parameters of X-Control Chart." Management Science, 17, 9 (May, 1971), 635-646.
- [31] Gibra, I. N. "Recent Developments in Control Chart Techniques."

 <u>Journal of Quality Technology</u>, 7, 4 (Oct., 1975), 183-192.
- [32] Girshick, M. A. and H. Rubin. "A Bayes' Approach to a Quality Control Model." <u>Annals of Mathematical Statistics</u>, 23 (1952), 114-125.
- [33] Goel, A. L., S. C. Jain and S. M. Wu. "An Algorithm for the Determination of the Economic Design of X-Charts Based on Duncan's Model." Journal of the American Statistical Association, 63, 321 (March, 1968), 304-320.
- [34] Goel, A. L. and S. M. Wu. "Economically Optimum Design of CuSum Charts." Management Science, 19, 11 (1973), 1271-1282.
- [35] Gordon, G. R. and J. I. Weindling. "A Cost Model for Economic Design of Warning Limit Control Chart Schemes." AIIE Transactions, 7, 3 (Sept., 1975), 319-329.
- [36] Grant, E. L. and R. S. Leavenworth. <u>Statistical Quality Control.</u> 5th Ed. New York: McGraw-Hill Book Company, 1980.
- [37] Hayes, G. E. and H. G. Romig. <u>Modern Quality Control</u>. Encino, Cal.: Bruce, 1977.
- [38] Heikes, R. G., D. C. Montgomery and J. Y. H. Yeung. "Alternative Process Models in the Economic Design of T² Control Charts." AIIE Transactions, 6, 1 (1974), 55-61.
- [39] Himmelblau, D. M. <u>Applied Nonlinear Programming</u>. New York: McGraw-Hill, 1972.
- [40] Hooke, R., and T. A. Jeeves. "Direct Search Solution of Numerical and Statistical Problems." Journal of Association of Computing Machinery, 8 (1961), 212-229.
- [41] Juran, J. M. Quality Control Handbook. 3rd Ed. New York: McGraw-Hill, 1974.
- [42] Kapur, K. C. and L. R. Lamberson. Reliability in Engineering
 Design. New York: John Wiley and Sons, Inc., 1979.

- [43] Knappenberger, H. A. and A. H. E. Grandage. "Minimum Cost Quality Control Tests." <u>AIIE Transactions</u>, 1, 1 (March, 1969), 24-32.
- [44] Kuester, J. L. and J. H. Mize. Optimization Techniques with Fortran. New York: McGraw-Hill, 1973.
- [45] Ladany, S. P. "Optimal Use of Control Charts for Controlling Current Production." Management Science, 19, 7 (1973), 763-772.
- [46] Ladany, S. P. and Y. Alperovitch. "An Optimal Set-up Policy for Control Charts." Omega, 3, 1 (1975), 113-118.
- [47] Montgomery, D. C. and P. J. Klatt. "Economic Design of T² Control Charts to Maintain Control of a Process." Management Science, 19, 1 (Sept., 1972), 76-89.
- [48] Montgomery, D. C. and P. J. Klatt. "Minimum Cost Multivariate Quality Control Tests." <u>AIIE Transactions</u>, 4, 2 (1972), 103-110.
- [49] Montgomery, D. C. and R. G. Heikes. "Process Failure Mechanism and Optimal Design of Fraction Defective Control Charts."

 AIIE Transactions, 8, 4 (1976), 467-472.
- [50] Montgomery, D. C. "The Economic Design of Control Charts: A Review and Literature Survey." <u>Journal of Quality Technology</u>, 12, 2 (April, 1980), 75-87.
- [51] Montgomery, D. C. "Economic Design of an X-Control Chart."

 Computer Programs, <u>Journal of Quality Technology</u>, 14, 1,

 (Jan. 1982), 40-43.
- [52] Moore, P. G. "Some Properties of Runs in Quality Control Procedures." <u>Biometrika</u>, 45 (1958), 89-95.
- [53] Nelder, J. A., and R. Mead. "A Simplex Method for Function Minimization." The Computer Journal, 7 (1965), 308-313.
- [54] Olsson, D. M. "A Sequential Simplex Program for Solving Minimization Problems." <u>Journal of Quality Technology</u>, 6, 1, (Jan. 1974), 53-57.
- [55] Page, E. S. "Cumulative Sum Charts." <u>Technometrics</u>, 3, 1 (Feb., 1961), 1-9.
- [56] Page, E. S. "A Modified Control Chart with Warning Limits."
 Biometrika, 49 (June, 1962), 171-176.
- [57] Page, E. S. "Comparison of Process Inspection Schemes."

 <u>Industrial Quality Control</u>, 21, 5 (Nov., 1964), 245-249.

- [58] Parzen, E. <u>Stochastic Processes</u>. San Francisco: Holden-Day, Inc., 1962.
- [59] Roberts, S. W. "Control Charts Based on Geometric Moving Averages." <u>Technometrics</u>, 1, 3 (Aug., 1959), 239-250.
- [60] Roberts, S. W. "A Comparison of Some Control Chart Procedures." Technometrics, 8, 3 (Aug., 1966), 411-43.
- [61] Rosenbrock, H. H. "An Automatic Method for Finding the Greatest or Least Value of a Function." <u>Computer Journal</u>, 3, (1960), 175-184.
- [62] Ross, S. M. Applied Probability Models with Optimizations. San Francisco: Holden-Day, Inc., 1970.
- [63] Ross, S. M. "Quality Control Under Markovian Deterioration." Management Science, 17, 5 (1971), 587-596.
- [64] Saniga, E. M. and L. E. Shirland. "Quality Control in Practice--A Survey." Quality Process, 10, 5 (1977), 30-33.
- [65] Saniga, E. M. "Joint Economically Optimal Design of \bar{X} and R Control Charts." Management Science, 24, 4 (1978), 420-431.
- [66] Savage, I. R. "Surveillance Problems." Naval Research Logistic Quarterly, 9, 2 (1962), 187-209.
- [67] Scheffe, H. "Operating Characteristics of Average and Range Charts." <u>Industrial Quality Control</u>, 5, 6 (May, 1949), 13-18.
- [68] Shewhart, W. A. "Quality Control Charts." Bell System Technical Journal, 5, 4 (Oct., 1926), 593-603.
- [69] Shewhart, W. A. "Quality Control." Bell System Technical Journal, 6, 4 (Oct., 1927), 722-735.
- [70] Shewhart, W. A. Economic Control of Quality of Manufactured Product. New York: D. Van Nostrand Co., Inc., 1931.
- [71] Taylor, H. M. "Markovian Sequential Replacement Processes."

 Annals of Mathematical Statistics, 36 (1965), 1677-1694.
- [72] Taylor, H. M. "Statistical Control of a Gaussian Process." <u>Technometrics</u>, 9, 1 (1967), 29-41.
- [73] Weiler, H. "On the Most Economical Sample Size for Controlling the Mean of a Population." <u>Annals of Mathematical</u> Statistics, 23 (1952), 247-254.
- [74] Weiler, H. "The Use of Runs to Control the Mean in Quality Control." <u>Journal of the American Statistical Association</u>, 48, 264 (1953), 816-825.

(

- [75] Weindling, J. L., S. B. Littauer and J. Tiago de Oliveria. "Mean Action Time of the X-Control Chart with Warning Limits."

 Journal of Quality Technology, 2, 2 (April, 1970), 79-85.
- [76] White, C. C. "A Markov Quality Control Process Subject to Partial Observation." <u>Management Science</u>, 23, 8 (1974), 843-852.
- [77] "ZXMIN," <u>International Mathematical and Satistical Library</u>, (June 1982), 1-4.

APPENDICES

APPENDIX A

DUNCAN'S MODEL AS SPECIAL CASE OF THE DYNAMIC MODEL

INTRODUCTION

This appendix is concerned with deriving Duncan's loss-cost model as a special case of the dynamic loss-cost expression for the situation in which the process failure mechanism is exponential and the control chart parameters-- n_i , h_i , and k_i --are constant throughout the chart's operation. This situation is the one employed in Duncan's model.

In the remainder of this appendix, different terms and components of the dynamic loss-cost expression are considered and each is simplified for Duncan's model situation. These components are then implemented in the dynamic loss-cost expression which is then seen to be the same as Duncan's loss-cost expression. The notation used in this appendix follows exactly the same convention as introduced in Chapter III.

Duncan's Model Situation

Duncan's model is based on the assumption that the process failure is given by the exponential distribution and that the control chart parameters— n_i , h_i , and k_i —are constant throughout the chart's operation. The equations which follow represent the immediate simplification of some of the terms used in the dynamic model for this situation.

$$n_i = n, \forall i$$
 (A.1)

$$h_i = h, \forall i$$
 (A.2)

$$k_i = k$$
, $\forall i$ (A.3)

$$t_{i} = \sum_{j=1}^{i} h_{j} = (i) h$$
 (A.4)

$$P_i = P = \Phi (\delta \sqrt{n} - k), \text{ for } \delta > 0$$
 (A.5)

$$Q_i = Q = 1 - P$$
 (A.6)

$$\alpha_{i} = \alpha = 2 \Phi (-k)$$
 (A.7)

Also, equation (3.6) is simplified to:

$$P(00C_i) = P(00C) = \int_{t_{i-1}}^{t_i} \lambda e^{-\lambda t} dt = e^{-\lambda (i-1)h} - e^{-\lambda ih}$$
 (A.8)

ATOWIN--Average Time of the Occurrence of the Shift Within an Interval

The expression for $ATOWIN_i$ is given by equation (3.16) of Chapter III. For Duncan's situation this expression can be simplified to the following:

ATOWIN; = ATOWIN =
$$\frac{1 - (1 + \lambda h)e^{-\lambda h}}{\lambda (1 - e^{-\lambda h})} \simeq \frac{h}{2} - \frac{\lambda h^2}{12}$$
 (A.9)

A00C--Average Out-of-Control Time Before the Detecting Sample is Charted

The expression for AOOC is given by equation (3.13) of Chapter III. This can be written as:

$$A00C = \sum_{i=1}^{\infty} P(00C_{i}) \left\{ \left[P_{i} (h_{i} + en_{i}) + Q_{i}P_{i+1}(h_{i} + h_{i+1} + en_{i+1}) + \prod_{j=i}^{i+1} Q_{j}P_{i+2} \left(\sum_{j=i}^{i+2} h_{j} + en_{i+2} \right) + \dots + \prod_{j=1}^{r-1} Q_{j}P_{r} \left(\sum_{j=i}^{r} h_{j} + en_{r} \right) + \dots \right] - ATOWIN_{i} \right\}$$

$$(A.10)$$

Substituting for the terms in (A.10) using equations (A.1) to (A.9), results in the following:

ADOC =
$$(1 - e^{-\lambda h}) \left\{ \left[P(h + en) + (1 - P)P(2h + en) + (1 - P)^2 P(3h + en) + ... \right] - ATOWIN \right\}$$

+ $(e^{-\lambda h} - e^{-2\lambda h}) \left\{ \left[P(h + en) + (1 - P)P(2h + en) + (1 - P)^2 P(3h + en) + ... \right] - ATOWIN \right\} + ... (A.11)$

After the cancellation of the similar terms, equation (A.11) can be written as:

A00C =
$$\left[P(h + en) + (1 - P)P(2h + en) + (1 - P)^{2}P(3h + en) + ... \right] - ATOWIN$$
 (A.12)

Expanding and rearranging the right hand side of this equation results in:

A00C = Pen
$$\left[1 + (1 - P) + (1 - P)^2 + ...\right]$$

+ Ph $\left[1 + 2 (1 - P) + 3(1 - P)^2 + ...\right]$ - ATOWIN (A.13)

Now, let (1 - P) = x and note that

$$1 + x + x^2 + \dots = \frac{1}{1 - x}$$
, for x < 1

and

$$1 + 2x + 3x^2 + \dots = \frac{1}{(1 - x)^2}$$
, for x < 1.

Therefore, equation (A.13) can be written as:

A00C =
$$Ph\frac{1}{p^2} + P = n\frac{1}{P} - ATOWIN$$

= $h/P + en - ATOWIN$
= $h/P + en - h/2 + \lambda h^2/2$ (A.14)

AIC--Average Cycle Length

The expression for the average cycle length is given by equation (3.7) of Chapter III. That is:

$$ACT = AIC + AOOC + D \tag{A.15}$$

Note that for the exponential distribution AIC, the average time in-control before the process goes out-of-control is equal to $1/\lambda$. Substituting $1/\lambda$ for AIC and expression (A.14) for AOOC in the ACT expression results in:

ACT =
$$1/\lambda + h/P + en - h/2 + \lambda h^2/2$$
 (A.16)

Notice that the average cycle length as given by equation (A.16) is equal to the average cycle length derived by Duncan.

ENFALS--Expected Number of False
Alarms Per Cycle

The expression for ENFALS is given by equation (3.24) of Chapter III. That is:

ENFALS =
$$\sum_{i=1}^{\infty} P(00C_i) \begin{bmatrix} i-1 \\ \sum_{j=1}^{\infty} \alpha_j \end{bmatrix}$$
 (A.17)

Substituting for $P(00C_{i})$ from equation (A.8) results in the following:

ENFALS = 0 +
$$(e^{-\lambda h} - e^{-2\lambda h})(\alpha)$$

+
$$(e^{-2\lambda h} - e^{-3\lambda h})(2\alpha) + (e^{-3\lambda h} - e^{-4\lambda h})(3\alpha) + ...$$
 (A.18)

After simplification, the above equation can be written as:

ENFALS =
$$\alpha e^{-\lambda h} + \alpha e^{-2\lambda h} + \alpha e^{-3\lambda h} + \dots = \frac{\alpha e^{-\lambda h}}{1 - e^{-\lambda h}}$$
 (A.19)

AHCS--Average Hourly Cost of Sampling and Charting

The expression for AHCS is given by equations (3.33) and (3.34) of Chapter III. The following is obtained by substituting equations (3.33) in (3.34).

AHCS =
$$\sum_{i=1}^{\infty} \left(P_i \frac{CI1}{LI1} + Q_i P_{i+1} \frac{CI2}{LI2} \right)$$

$$+ Q_i Q_{i+1} P_{i+2} \frac{CI3}{LI3} + ... P(00C_i)$$
 (A.20)

Note that for the Duncan's model situation, the general form $\frac{\text{CIr}}{\text{LIr}}$ can be written as:

$$\frac{CIr}{LIr} = \frac{r(b+cn) + (\frac{en+D}{h})(b+cn)}{h+en+D}$$

$$= \frac{(b + cn) (rh + en + D)}{h(rh + en + D)} = \frac{b + cn}{h}$$
 (A.21)

Substituting (A.5), (A.6), (A.8), and (A.21) in equation (A.20) results in:

AHCS =
$$(1 - e^{-\lambda h}) \left[P\left(\frac{b + cn}{h}\right) + (1 - P)P\left(\frac{b + cn}{h}\right) + (1 - P)P\left(\frac{b + cn}{h}\right) + (1 - P)^{2}P\left(\frac{b + cn}{h}\right) + \dots \right] + (e^{-\lambda h} - e^{-2\lambda h}) \left[P\left(\frac{b + cn}{h}\right) + (1 - P) P\left(\frac{b + cn}{h}\right) + \dots \right] + \dots$$

$$= P\left(\frac{b + cn}{h}\right) + (1 - P)P\left(\frac{b + cn}{h}\right) + (1 - P)^{2}P\left(\frac{b + cn}{h}\right) + \dots$$

$$= P\left(\frac{b + cn}{h}\right) \left[1 + (1 - P) + (1 - P)^{2} + \dots \right]$$

$$= \frac{b + cn}{h} \tag{A.22}$$

The Average Loss-Cost Per Hour

The expression for the average loss-cost per hour for the dynamic model is given by equation (3.45). That is:

$$L = (1 - \beta)M + T \frac{ENFALS}{ACT} + \frac{W}{ACT} + AHCS$$
 (A.23)

Substitute for ENFALS and AHCS using equations (A.19) and (A.22), respectively. Therefore,

$$L = (1 - \beta)M + T \left[\left(\frac{\alpha e^{-\lambda h}}{1 - e^{-\lambda h}} \right) / ACT \right] + W/ACT$$

$$+ (b + cn)/h \qquad (A.24)$$

Observing that Duncan approximates $\frac{\alpha}{1-e^{-\lambda h}}$ by $\alpha/\lambda h$ and represents $(1-\beta)$ by γ , shows that the loss-cost given by equation (A.24) is equal to Duncan's loss-cost expression.

APPENDIX B

FORTRAN PROGRAM LISTING

		· · · · · · · · · · · · · · · · · · ·	
	*******	*************	
C*			00000200
C*	THIS INTRACTIVE	PROGRAM PERFORMS	00000300
C*		DESIGN AND EVALUATION OF THE DYNAMIC X-BAR CHART	00000400
C*	(2) ECONOMIC	DESIGN AND EVALUATION OF DUNCAN'S X-BAR CHART	00000500
C*			00000600
C*	BY BEHROOZ PARKH		00000700
C*		MANAGEMENT	00000800
C*		OKLAHOMA STATE UNIVERSITY	00000900
C*	DISSERTATION ADV	OKLAHOMA STATE UNIVERSITY VISOR: DR. KENNETH E. CASE	00001000
C*			00001100
C*			00001200
	******	***********	
C*			00001400
C*	GENERAL STRUCTUR	RE AND INPUT REQUIREMENTS	00001500
C*			00001600
C*			00001700
C*	SUBROUTINE	FUNCTION	00001800
C*			00001900
C*	DYNM	PROMPTS THE USER FOR INFORMATION NEEDE FOR ECON. DESIGN AND/OR ECON. EVALUATION OF THE DYNAMIC X-BAR CHART.	00002000
C*		ECON. DESIGN AND/OR ECON. EVALUATION OF THE	00002100
C*		DYNAMIC X-BAR CHART.	00002200
C*	D1111G	BROWNER WIR HOLD BOD THRODY MICH NEEDS BOD	00002300
C*	DUNC	PROMPTS THE USER FOR INFORMATION NEEDE FOR	00002400
. C*		PROMPTS THE USER FOR INFORMATION NEEDE FOR ECON. DESIGN AND/OR ECON. EVALUATION OF DUNCAN'S X-BAR CHART.	00002500 00002600
C*		DUNCAN'S A-BAR CHART.	00002800
C*	DUNOPT	OPTIMIZES DUNCAN'S MODEL.	00002700
C*	DUNOFI	OFTIMIZES DONCAN'S MODELL.	00002000
C*	FUNCT	DUNCAN'S COST MODEL USED FOR 3-DIMENSIONAL	00003000
C*	ronci	OPTIMIZATION.	00003000
Č*		Of ITMI BATTON,	00003100
Č*	FUNCT2	DUNCAN'S COST MODEL USED FOR 2-DIMENSIONAL	00003300
Č*	10015	OPTIMIZATION OVER H AND K.	00003400
Č*			00003500
Č*	PROBD	CALCUATES PROBABILITY OF DETECTING THE SHIFT	00003600
C*		FOR DUNCAN'S MODEL.	00003700
C*			00003800
C*	PROFA	CALCUATES PROBABILITY OF FALSE ALARMS FOR	00003900
C*		DUNCAN'S MODEL.	00004000
C*			00004100
C*	DYNOPT	DUNCAN'S MODEL. OPTIMIZES THE DYNAMIC MODEL.	00004200
C*			00004300
C*	TWOSCH	TWO-AT-A-TIME SEARCH ROUTINE.	00004400
C*	A		00004500
C*	OMYSCH	USED IN CONJUNCTION WITH TWOSCH TO PERFORM A	00004600
C*		TWO-DIMENSIONAL SEARCH.	00004610
C*	COCCTY	DEDEADNG & DDECTOR LINE CENDON MOTIO	00004700
C*	COGGIN	PERFORMS A PRECISE LINE SEARCH USING THE	00004800
C*		METHOD OF COGGINS.	00004900 00005000
C*	CSTPW	CALCULATES THE VALUE OF ISTEPS FOR GIVEN VALUES	00005000
C*	COLEM	OF HF, IH, AND WEIBULL DISTRIBUTION PARAMETERS.	00005100
C*		or ar, in, and merbond distribution parameters.	00005300
Č*	FTCR	CALCULATES THE VALUE OF NF (OR KF) FOR GIVEN	00005400
Č*		VALUES OF IN (OR IK) AND IN ENDING (OR IK ENDING	
Č*		The second secon	00005600
Č*	SETDEL	CALAULATES THE VALUE OF INITIAL STEP SIZE FOR	00005700
Č*		KF (OR NF) FOR THE SEARCH ROUTINE.	00005800
Č*			00005900
C*	APP	CALCULATES THE AVERGE TIME THE PROCESS IS	00005910
C*		IN OUT-OF-CONTROL CONDITION BEFORE THE DETECTING	00005920
C*		SAMPLE IS PLOTTED ON THE CHART.	00005930

```
C*
                                                                       00005940
C*
      CMAINT
                    CALCULATES THE AVERAGE HOURLY COST OF MAINTAINING 00005950
C*
                    THE CONTROL CHART ( FOR THE DYNAMIC MODEL.)
                                                                       00005960
C*
                                                                       00005970
Č*
                    CALCULATES THE AVERAGE NUMBER OF FALSE ALARMS PER 00005980
Č*
                    CYCLE ( FOR THE DYNAMIC MODEL.)
                                                                       00005991
C*
                                                                       00005992
                                                                       00005993
C*
      PROOCW
                    CALCULATES THE AREA UNDER WEIBULL DENSITY BETWEEN
C*
                    A TO B ( FOR THE DYNAMIC MODEL.)
                                                                       00005994
C*
                                                                       00005995
C*
                                                                       00007100
C***
      ********************
C*
                                                                       00007300
C*
                                                                       00007400
    EXTERNAL FUNCTIONS REQUIRED:
C*
      (1) REGULAR SYSTEM SUPPLIED FORTRAN FUNCTIONS
                                                                       00007500
      (2) FOUR IMSL SUBROUTINES
C*
                                                                       00007600
C*
          MDNORD-- CALCULATES NORMAL DENSITY INTEGRAL.
                                                                       00007700
C*
          MDGAM -- CALCULATES THE INCOMPLETE GAMA INTEGRAL.
MDGAMMA-- CALCULATES THE GAMMA FUNCTION.
                                                                       00007800
C*
                                                                       00007810
C*
           ZXMIN -- PERFOMS FUNCTION MINIMIZATION USING A QUASI-NEWTON 00007900
C*
                    METHOD.
                                                                       00008000
C*
                                                                       00008100
C*
                                                                       00008200
C*
                                                                       00008300
C**********************
                                                                       00008400
C*
                                                                       00008500
C*
                                                                       00008600
C*
                                                                       00008700
C*
                                                                       00008800
C*
                                                                       00008900
C*************
                                           ******* 00009000
C** MAIN PROGRAM
                                                                     * 00009100
C**
                                                                     * 00009200
C** THIS PROGRAM CALLS SUBROUTINES DYNM AND DUNC TO PERFORM THE
                                                                     * 00009300
C** FOLLOWING TASKS:
                                                                     * 00009400
C**
                                                                     * 00009500
       (1) ECONOMIC DESIGN AND EVALUATION OF THE DYNAMIC X-BAR CHART
C**
       (2) ECONOMIC DESIGN AND EVALUATION OF DUNCAN'S X-BAR CHART
                                                                     * 00009600
C**
                                                                     * 00009700
C*************************
                                                                       00009800
C*
                                                                       00009900
C*
                                                                       00010000
     IMPLICIT REAL*8(A-H,O-Z)
                                                                       00010100
      COMMON / MAIN1 / LUR, LUW
                                                                       00010200
C*
                                                                       00010300
С
                                                                       00010400
C
                                                                       00010500
C**
                                                                       00010600
C**LUR IS THE LOGICAL UNIT NUMBER OF THE READER C**LUW IS THE LOGICAL UNIT NUMBER OF THE PRINTER
                                                                       00010700
                                                                       00010800
C**
                                                                       00010900
      LUR=5
                                                                       00011000
      LUW=6
                                                                       00011100
C**
                                                                       00011200
C**PROMPT THE USER WITH THE MAIN MENU
                                                                       00011300
C**
                                                                       00011400
   10 WRITE(LUW.11)
                                                                       00011500
   11 FORMAT(/,/,T5,' ==> MAIN MENU <==',/,/,

* '*** ENTER OPTION NUMBER',/,
                                                                       00011600
                                                                       00011700
             T5,' 1 = THE DYNAMIC X-BAR CHART (WEIBULL ENVIRONMENT)',/,
                                                                       00011800
         T5,55H 2 = DUNCAN'S X-BAR CHART(EXPONENTIAL ENVIRONMENT)
                                                                       00011900
      read(Lur,*)Menul
                                                                       00012000
                                                                       00012100
      GO TO (100,200,300), MENU1
                                                                       00012200
      WRITE(LUW, 20)
                                                                       00012300
   20 FORMAT(' !?! ERROR -- DO IT OVER !')
                                                                       00012400
      GO TO 10
                                                                       00012500
```

```
100 CALL DYNM
                                                                                                0.0012600
                                                                                                00012700
        GO TO 10
   200 CALL DUNC
                                                                                                00012800
        GO TO 10
                                                                                                00012900
                                                                                                 00013000
C**EXIT SYSTEM
                                                                                                00013100
C**
                                                                                                 00013200
  300 STOP
                                                                                                 00013300
        END
                                                                                                 00013400
C*
                                                                                                 00013500
C*
                                                                                                 00013600
C*
                                                                                                 00013700
C*
                                                                                                 00013800
                                                                                 ******** 00013900
SUBROUTINE DYNM
                                                                                                 00014200
                                                                                             ** 00014300
C**
                                                                                              * 00014400
C** THIS ROUTINE PROMPTS THE USER FOR THE NECESSARY INFORMATION
                                                                                              * 00014500
                                                                                              * 00014600
C** NEEDED FOR THE DYNAMIC X-BAR CHART DESIGN OR EVALUATION.
C**
                                                                                              * 00014700
C** THIS ROUTINE CALLS THE FOLLOWING SUBROUTINES:
                                                                                              * 00014800
            DUNOPT-- TO OPTIMZE DUNCAN'S MODEL EQUIVALENT TO THE DYNAMIC * 00014900 MODEL IN ORDER TO GET A GOOD STARTING POINT FOR * 00015000
C**
C**
                                                                                              * 00015000
C**
                        DYNOPT ROUTINE.
                                                                                              * 00015100
C**
                                                                                              * 00015200
            DYNOPT-- OPTIMIZE DYNAMIC COST MODEL
C**
                                                                                              * 00015300
            DYMEVA-- EVALUATE DYNAMIC COST MODEL FOR A GIVEN DESIGN
C**
            DUNOPT-- TO OPTIMZE DUNCAN'S MODEL EQUIVALENT TO THE DYNAMIC * 00015400
C**
                                                                                              * 00015500
                        MODEL IN ORDER TO GET A GOOD STARTING POINT FOR
                        DYNOPT ROUTINE.
C**
                                                                                              * 00015600
C**
                                                                                              * 00015700
C**
                                                                                                 00015900
C*
                                                                                                 00016000
        IMPLICIT REAL*8(A-H,O-Z)
                                                                                                 00016100
       REAL*8 LAMBDA 00016200
REAL*8 NF, IH, HF, IK, KF 00016300
COMMON / MAIN1 /LUR, LUW 00016400
COMMON / DCDY1 / DELTA, B,C,DD,E,VZMV1,T,W 00016500
COMMON / DUNC1 / LAMBDA 00016600
COMMON / DUNC4 / N,H,RK 00016700
COMMON / DUNC5 / NDCOPT, HDCOPT, RKDCOP, FDCOPT 00016800
COMMON / DYNM1 / THETA,ETA 00016900
COMMON / DYNM2 / ISTEPS 00017000
COMMON / DYNM3 / IN, NF, IH, HF, IK, KF 00017100
COMMON / DYNM4 / PROBPT 00017200
COMMON / DYNM5 / ITRMX1, ITRMX2, ITRMX3 00017300
COMMON / DYNM6 / DEL(6), DELMN(6), DELMX(6), XQLIM(6) 00017400
COMMON / DYNM8 / ISTPP 00017500
COMMON / DYNM8 / ISTPP 00017700
COMMON / DYNM8 / ISTPP 00017700
COMMON / DYNM8 / ISTPP 00017800
COMMON / DYOPT4 / NWOPT, HWOPT, FNWOPT, FHWOPT, FKWOPT, YFWOPT 00017800
COMMON / CMN1 / CUPROX 00017900
        REAL*8 LAMBDA
                                                                                                 00016200
                                                                                                 00017900
C*
             ENTER DISTRIBUTION, COST, AND OTHER PARAMETERS
                                                                                                 00018000
                                                                                                 00018100
      5 WRITE(LUW,6)
                                                                                                 00018200
     6 FORMAT(/,T5,' >> WEIBULL ENVIRONMENT <<',/,
* ' *** FOR THE DYNAMIC ECONOMIC X-BAR CHART,')
                                                                                                 00018300
                                                                                                 00018400
                                                                                                 00018500
C**ENTER DISTRIBUTION INFROMATION
                                                                                                 00018600
                                                                                                 00018700
C*.IREV=1 INDICATES ORDINARY INITIALIZATION OF PARAMETERS
                                                                                                 00018800
C*.IREV=2 INDICATES THE PASS AFTER 3 ( DIST. PARMS. REVISION) IS
                                                                                                 00018900
C*.SELECTED FROM THE FOLLOWING MENU.
                                                                                                 00019000
                                                                                                 00019100
```

```
IREV=0
                                                                                         00019200
 6001 IREV=IREV+1
                                                                                          00019300
                                                                                          00019400
     7 WRITE(LUW,8)
     8 FORMAT(T5,' ENTER VALUES: THETA, ETA')
READ(LUR,*)THETA, ETA
                                                                                          00019500
                                                                                          00019600
                                                                                          00019700
C*.CALCULATE MEAN OF THE WEIBULL DISTRIBUTION
                                                                                          00019800
                                                                                          00019900
        USEWMN=1.D0+1.D0/ETA
                                                                                          00020000
        WBMEAN=1.D0/THETA*DGAMMA(USEWMN)
                                                                                          00020100
       WRITE(LUW, 12) THETA, ETA, WBMEAN
                                                                                          00020200
   12 FORMAT(' VALUES ENTERED ARE:',/.
                                                                                          00020300
     * T5,' DISTRIBUTION INFORMATION;',/, 00020400
*T7,' WEIBULL W/ THETA=',F10.6,' AND ETA=',F10.6,' => MEAN=',F10.4)00020500
   13 WRITE(LUW,14)
                                                                                          00020600
   14 FORMAT(' CORRECT ? 1=YES 2=NO 3=RETURN TO THE MAIN MENU')
                                                                                          00020700
       READ(LUR, *) I YN1
                                                                                          00020800
       GO TO ( 15, 7, 600), IYN1
WRITE(LUW, 22)
                                                                                          00020900
                                                                                          00021000
                                                                                          00021100
       GO TO 13
C**
                                                                                          00021200
C**ENTER COST AND SHIFT PARAMETERS
                                                                                          00021300
C**
                                                                                          00021400
   15 IF(IREV.GT.1) GO TO 20
                                                                                          00021500
   WRITE(LUW,16)

16 FORMAT(T5,' ENTER VALUES: DELTA, B, C, D, E, M, T, W')
READ(LUR,*) DELTA, B,C,DD,E,VZMV1, T, W
WRITE(LUW,17)DELTA,B,C,DD,E,VZMV1,T,W
                                                                                          00021600
                                                                                          00021700
                                                                                          00021800
                                                                                          00021900
   17 FORMAT(T5,' COST AND OTHER INFORMATION:',/,
*T7,' DELTA=',F10.4,' B=',F10.4,' C=',F10.4,' D=',F10.4,/,
*T7,' E=',F10.4,' M=',F10.4,' T=',F10.4,' W=',F10.4)
                                                                                          00022000
                                                                                          00022100
                                                                                          00022200
    18 WRITE(LUW, 19)
                                                                                          00022300
    19 FORMAT(' CORRECT ? 1=YES 2=NO 3=RETURN TO THE MAIN MENU')
                                                                                          00022400
       READ(LUR, *) IYN2
                                                                                          00022500
       GO TO ( 20, 15, 600), IYN2
                                                                                          00022600
       WRITE(LUW, 22)
                                                                                          00022700
       GO TO 18
                                                                                          00022800
                                                                                          00022900
C*
            SELECTION FOR DESIGN, EVALUATION , ETC.
                                                                                          00023000
                                                                                          00023100
    20 IREV=1
                                                                                          00023200
    WRITE(LUW,21)

21 FORMAT(/,' *** ENTER OPTION NUMBER',/,

* T5,' 1 = ECON. DESIGN OF THE DYNAMIC X-BAR CHART (OPTIMIZATION'00023500
           T5,' 2 = ECON. EVALUATION IN THE WEIBULL ENVIRONMENT',/,
                                                                                          00023600
                                                                                          00023700
           T5, '3 = RETURN TO REVISE WEIBULL PARAMETERS',/,
T5,' 4 = RETURN TO REVISE COST PARAMETERS',/,
T5,' 5 = RETURN TO THE MAIN MENU')
                                                                                          00023800
                                                                                          00023900
                                                                                          00024000
       READ(LUR, *)MENU2
                                                                                          00024100
       GO TO ( 100, 390 , 6001, 15, 600), MENU2
                                                                                          00024200
       WRITE(LUW, 22)
                                                                                          00024300
    22 FORMAT(' !?! ERROR -- DO IT OVER !')
                                                                                          00024400
       GO TO 18
                                                                                          00024500
C*
                                                                                          00024600
C
            ECON. DESIGN (OPTIMIZATION) OF THE DYNAMIC X-BAR CHART
                                                                                          00024700
C*
                                                                                          00024900
C**
                                                                                          00025000
C**INITIALIZATION OF STARTING POINT FOR OPTIMIZATION
                                                                                          00025100
                                                                                          00025200
C*.THE FOLLOWING N , H , AND K ARE USED AS STARTING POINT FOR OPTIMIZING00025300 C*.DUNCAN'S MODEL WHICH PROVIDES THE STARTING POINT FOR OPTIMIZATION OF 00025400
C*.THE DYNAMIC MODEL.
                                                                                          00025500
C*.
                                                                                          00025600
   100 N=5
                                                                                          00025700
```

```
H=1.D0
                                                                                     00025800
       RK=3.D0
                                                                                     00025900
C**
                                                                                     00026000
C**CALCULATE THE CORRESPONDING EXPONENTIAL PARAMETER USED IN DUNCAN'S
                                                                                     00026100
C**COST MODEL.
                                                                                     00026200
C**
                                                                                     00026300
       LAMBDA=1.D0/WBMEAN
                                                                                     00026400
C
                                                                                     00026500
C*.
                                                                                     00026600
       CALL DUNOPT
                                                                                     00026700
C*.
                                                                                     00026800
С
                                                                                     00028900
       IN=NDCOPT
                                                                                     00027000
       NF=1.D0
                                                                                     00027100
       IH=HDCOPT
                                                                                     00027200
       HF=1.0
                                                                                     00027300
       IK=RKDCOP
                                                                                     00027400
                                                                                     00027500
       KF=1.0
                                                                                     00027600
                                                                                     00027700
  101 WRITE(LUW, 102) IN, IH, IK, NF, HF, KF
  102 FORMAT(' *** FOR ECON. OPTIMIZATION OF THE DYNAMIC X-BAR CHART,'
                                                                                     00027800
     * ,/,T5,' THE FOLLOWING STARTING POINT IS SUGGESTED:',/,

* T6,' IN=',I4,5X,' IH=',F10.4,' IK=',F10.4,/,

* T6,' NF=',F9.7,' HF=',F9.7,1X,' KF=',F9.7,/,

* T5,' YOU ACCEPT THIS POINT.')
                                                                                     00027900
                                                                                     00028000
                                                                                     00028100
                                                                                     00028200
  103 WRITE(LUW, 104)
                                                                                     00028300
  104 FORMAT(' CORRECT ?
                              1=YES 2=NO 3=RETURN TO THE PREVIOUS MENU') 00028400
       READ(LUR, *) IYN1
                                                                                     00028500
       GO TO (180,110,20),1YN1
                                                                                     00028600
       WRITE(LUW, 106)
                                                                                     00028700
  106 FORMAT(' !?! ERROR -- DO IT OVER !')
                                                                                     00028800
       GO TO 103
                                                                                     00028900
C*
                                                                                     00029000
C....IF THE SUGGESTED STARTING POINT IS NOT ACCEPTED...
                                                                                     00029100
                                                                                     00029200
  110 WRITE(LUW,111)
                                                                                     00029300
  111 FORMAT(' FOR YOUR DESIRED STARTING POINT FOR OPTIMIZATION, ',

* 'ENTER:',/,5X,'IN, IH, IK, NF, HF, KF')
READ(LUR,*)IN,IH,IK,NF,HF,KF
                                                                                     00029400
                                                                                     00029500
                                                                                     00029600
C**
                                                                                     00029700
C**CHECK TO SEE IF THESE ARE IN THE ACCEPTABLE RANGE
                                                                                     00029800
C**
                                                                                     00029900
       IF(IN.LT.1000 .AND.IN.GE.2 ) GO TO 118
    WRITE(LUW,115)
                                                                                     00030000
                                                                                     00030100
  115
           FORMAT(' !?! ERROR -- IN SHOULD BE BETWEEN 2 AND 1000')
                                                                                     00030200
           GO TO 135
                                                                                     00030300
  118 IF(IH.GT.0.0 .AND.IH.LT.100. ) GO TO 125
                                                                                     00030400
           WRITE(LUW, 120)
                                                                                     00030500
           FORMAT(' !?! ERROR -- IH SHOULD BE BETWEEN 0.0 AND 100.')
  120
                                                                                     00030600
           GO TO 135
                                                                                     00030700
  125 IF( IK.GT.O.O .AND. IK.LT.12. ) GO TO 129
                                                                                     00030800
           WRITE(LUW, 127)
                                                                                     00030900
           FORMAT(' !?! ERROR -- IK SHOULD BE BETWEEN 0.0 AND 12.')
                                                                                     00031000
  129 IF( NF.GT.2.D0 .OR. NF.LT.0.D0 ) GO TO 132
                                                                                     00031100
       IF( HF.GT.2.D0 .OR. HF.LT.0.D0 ) GO TO 132
IF( KF.GT.2.D0 .OR. KF.LT.0.D0 ) GO TO 132
                                                                                     00031200
                                                                                     00031300
       GO TO 155
                                                                                     00031400
                                                                                     00031500
           WRITE(LUW, 133)
  132
                                                                                     00031600
           FORMAT(' !?! ERROR -- NF, NH, AND NK SHOULD BE BETWEEN ', '0.0 AND 2.',/)
  133
                                                                                     00031700
                                                                                     00031800
                                                                                     00031900
  135
           WRITE(LUW, 136)
                                                                                     00032000
           FORMAT(' DO IT OVER !')
  136
                                                                                     00032100
           GO TO 110
                                                                                     00032200
C**
                                                                                     00032300
```

```
C**ECHOPRINT THE VALUES FOR CHECK
                                                                              00032400
C**
                                                                              00032500
  155 WRITE(LUW, 158) IN, IH, IK, NF, HF, KF
                                                                              00032600
  158 FORMAT(' VALUES ENTERED: IN=',14,5x,' IH=',F9.4,4x,' IK='

* ,F9.4,/,T17,' NF=',F9.7,' HF=',F9.7,4x,' KF=',F9.7
                                                                              00032700
                                                             KF=',F9.7)
                                                                              00032800
  159 WRITE(LUW, 160)
                                                                              00032900
  160 FORMAT(' CORRECT ?
                           1=YES 2=NO 3=RETURN TO THE PREVIOUS MENU') 00033000
      READ(LUR, *)IYN3
                                                                              00033100
      GO TO (180,110,20),1YN3
                                                                              00033200
      WRITE(LUW, 161)
                                                                              00033300
  161 FORMAT(' !?! ERROR -- DO IT OVER !')
                                                                              00033400
      GO TO 159
                                                                              00033500
C*
                                                                              00033600
C*
                                                                              00033700
   _._. WHEN THE STARTING POINT IS ACCEPTED_._.
                                                                              00033800
                                                                              00033900
C....THEN SUGGEST THE QUANTILE.....
                                                                              00034000
                                                                              00034100
  180 PROBPT=0.99D0
                                                                              00034200
      WRITE(LUW, 184)PROBPT
                                                                              00034300
  184 FORMAT(T5, ' QUANTILE VALUE OF ',F11.9,' IS USED.',

* /,T5,' YOU ACCEPT THIS.')
                                                                              00034400
                                                                              00034500
  185 WRITE(LUW, 186)
                                                                              00034600
  186 FORMAT(' CORRECT ? 1=YES 2=NO 3=RETURN TO THE PREVIOUS MENU') 00034700
      READ(LUR, *) IYN5
                                                                              00034800
      GO TO (200,190,18),1YN5
                                                                              00034900
      WRITE(LUW, 188)
                                                                              00035000
  188 FORMAT(' !?! ERROR -- DO IT OVER !')
                                                                              00035100
      GO TO 185
                                                                              00035200
   ______IF QUANTILE VALUE IS NOT ACCEPTED_._...
                                                                              00035300
                                                                              00035400
C*
                                                                              00035500
  190 WRITE(LUW,191)
                                                                              00035600
  191 FORMAT(' ENTER YOUR DESIRED VALUE FOR QUANTILE:')
                                                                              00035700
      READ(LUR, *) PROBPT
                                                                              00035800
                                                                              00035900
C**CHECK TO SEE IF IT IS IN THE ACCEPTABLE RANGE
                                                                              00036000
                                                                              00036100
      IF( PROBPT.LE.1.0D0 .AND. PROBPT.GT.0.D0 ) GO TO 200
                                                                              00036200
      WRITE(LUW, 195)
                                                                              00036300
  195 FORMAT(' !?! ERROR -- QUANTILE SHOULD BE BETWEEN 0.0 AND 1.0',

* /,' DO IT OVER !')
                                                                              00036400
                                                                              00036500
      GO TO 190
                                                                              00036600
C*
                                                                              00036700
C.....IF QUANTILE VALUE IS ACCEPTED......CTHEN SUGGEST OPTIMIZATION PARAMETERS, ETC.....
                                                                              00036800
                                                                              00036900
                                                                              00037000
C**NOTE THAT VARIABLES:
                           1
                                  2
                                         3
                                                                              00037100
C**CORRESPOND TO
                               ISTEPS IK
                            HF
                                               KF
                                                      IN
                                                            NF , RESPECTIVELY00037200
                                                                              00037300
C*.CALCULATE ISTEPS FOR THE GIVEN VALUES OF HF, IH, PROBPT, AND
                                                                              00037400
C*.DISTRIBUTIONA INFORMATION
                                                                              00037500
C*.
                                                                              00037600
  200 ISTEPS=CSTPW(HF,IH)
                                                                              00037700
C*.
                                                                              00037800
C*.ARRAY DEL CONTAINS THE INITIAL STEP SIZES
                                                                              00037900
                                                                              00038000
      DEL(1)=.001
                                                                              00038100
      DEL(2)=2.
                                                                              00038200
      DEL(3)=.1
                                                                              00038300
С.,
                                                                              00038400
C..SET THE LIMIT ON KF AUTOMATICALLY BY SPECIFYING AN INCREMENT ON THE 00038500
C..IK-ENDING, DELIME.
                                                                              00038600
                                                                              00038700
      DELIKE=.25D0
                                                                              00038800
      DEL(4)=SETDEL(DELIKE)
                                                                              00038900
```

```
DEL(5)=1.
                                                                                   00039000
с..
                                                                                   00039100
C..SET THE LIMIT ON NF AUTOMATICALLY BY SPECIFYING AN INCREMENT ON THE
                                                                                   00039200
C.. IN-ENDING, DELIME.
                                                                                   00039300
с..
                                                                                   00039400
       DELINE=.25D0
                                                                                   00039500
      DEL(6) = SETDEL(DELINE)
                                                                                   00039600
C*.
                                                                                   00039700
C*.INITIALIZE MINIMUNM LIMITS ON STEP SIZES
                                                                                   00039800
C*.
                                                                                   00039900
       DELMN(1) = .0001
                                                                                   00040000
       DELMN(2)=1.D0
                                                                                   00040100
       DELMN(3) = .1D0
                                                                                   00040200
       DELMN(4) = DEL(4)/4.D0
                                                                                   00040300
       DELMN(5)=1.D0
                                                                                   00040400
       DELMN(6) = DEL(6)/4.D0
                                                                                   00040500
C*.
                                                                                   00040600
C*.INITIALIZE MAXIMUM LIMITS ON STEP SIZES
                                                                                   00040700
C*.
                                                                                   00040800
                                                                                   00040900
       DELMX(1) = .004D0
       DELMX(2)=32.D0
                                                                                    00041000
       DELMX(3) = .2D0
                                                                                   00041100
       DELMX(4)=DEL(4)
                                                                                    00041200
       DELMX(5)=2.D0
                                                                                   00041300
       DELMX(6)=DEL(6)
                                                                                   00041400
C*.
                                                                                   00041500
C*.INITIALIZE THE REQUIRED PRECISION
                                                                                   00041600
C*.
                                                                                   00041700
       XQLIM(1) = .00005
                                                                                   00041800
       XQLIM(2)=1.D0
                                                                                   00041900
       XQLIM(3) = .05D0
                                                                                   00042000
       XQLIM(4) = DEL(4)/6.D0
                                                                                    00042100
       XQLIM(5)=1.D0
                                                                                   00042200
       XQLIM(6) = DEL(6)/6.D0
                                                                                    00042300
C*.
                                                                                    00042400
C*.INITIALIZE THE LIMITS ON THE MAXIMUM NUMBER OF LOSS-COST EVALUATIONS 00042500
C*.FOR THE THREE STAGS OF THE SEARCH.
                                                                                    00042600
C*.
                                                                                    00042700
       ITRMX1=45
                                                                                    00042800
       ITRMX2=25
                                                                                    00042900
       ITRMX3=25
                                                                                   00043000
C*.
                                                                                    00043100
C*.SUGGEST THESE TO THE USER
                                                                                    00043200
C*.
                                                                                    00043300
                                                                                    00043400
  205 WRITE(LUW,210)
210 FORMAT(T5,' OPTIMIZATION PARAMETERS:',/,
                                                                                    00043500
                                                                                    00043600
      * T18,' HF
      ' T18,' HF | ISTEPS',T36,'IK
WRITE(LUW,213) (DEL(I),I=1,6)
                                                                        NF',/)
                                                   KF', T54, 'IN
                                                                                   00043700
                                                                                    00043800
  213 FORMAT(' STEP SIZE:',T17,F8.6,T27,F3.0,T33,F7.4,T43,F9.7,T54,
* F3.0,T60,F9.7)
                                                                                    00043900
                                                                                    00044000
      WRITE(LUW, 215) (DELMN(I), I=1,6)
                                                                                   00044100
  215 FORMAT(' MIN STEP SIZE:',T17,F8.6,T27,F3.0,T33,F7.4,T43,F9.7,T54,
                                                                                   00044200
      * F3.0,T60,F9.7)
WRITE(LUW,217) (DELMX(I),I=1,6)
                                                                                    00044300
                                                                                    00044400
  217 FORMAT(' MAX STEP SIZE:',T17,F8.6,T27,F3.0,T33,F7.4,T43,F9.7,T54,
                                                                                   00044500
           F3.0,T60,F9.7)
                                                                                    00044600
  WRITE(LUW,219) (XQLIM(I),I=1,6) 00044700
219 FORMAT(' REQ PRECISION:',T17,F8.6,T27,F3.0,T33,F7.4,T43,F9.7,T54, 00044800
           F3.0,T60,F9.7,/)
                                                                                    00044900
C
                                                                                    00045000
       WRITE(LUW, 222) ITRMX1, ITRMX2, ITRMX3
                                                                                   00045100
  222 FORMAT(' MAX. NUMBER OF ITERATIONS (LOSS-COST EVALUATIONS):',/,

* T5,' ITRMX1=',14,' ITRMX2=',14,' ITRMX3=',14,/)
                                                                                   00045200
                                                                                   00045300
  225 WRITE(LUW, 226)
226 FORMAT(' ***ENTER OPTION NUMBER:',/,
                                                                                   00045400
                                                                                   00045500
```

```
T5,' 1 = ALL CORRECT, NO REVISION NEEDED',/,
                                                                                              00045600
            T5, 1 = ALL CORRECT, NO REVISION NEEDED',/,
T5,' 2 = NEED TO REVISE OPTIMIZATION PARAMETERS',/,
T5,' 3 = NEED TO REVISE MAX. NUMBER OF ITERATIONS',/,
T5,' 4 = RETURN TO THE PREVIOUS MENU')
                                                                                              00045700
                                                                                              00045800
                                                                                              00045900
       READ(LUR, *)MENU3
                                                                                              00046000
       GO TO (300, 230, 250, 20), MENU3
                                                                                              00046100
                                                                                              00046200
  230 WRITE(LUW,231)
231 FORMAT(' FOR VARIABLES: HF, ISTEPS, IK, KF, IN, AND NF,',/,/,

* ENTER INITIAL STEP SIZES:')
                                                                                              00046300
                                                                                              00046400
                                                                                              00046500
       READ(LUR,*)(DEL(I),I=1,6)
                                                                                              00046600
       WRITE(LUW, 233)
                                                                                              00046700
  233 FORMAT(' THE MIN. LIMIT ON STEP SIZES:')
                                                                                              00046800
       READ(LUR,*) (DELMN(I), I=1,6)
                                                                                              00046900
       WRITE(LUW, 235)
                                                                                              00047000
  235 FORMAT(' THE MAX. LIMIT ON STEP SIZES:')
                                                                                              00047100
       READ(LUR,*) (DELMX(I), I=1,6)
                                                                                              00047200
       WRITE(LUW, 237)
                                                                                              00047300
  237 FORMAT(' THE REQUIRED PRECISION FOR EACH VARIABLES:')
                                                                                              00047400
       READ(LUR,*) (XQLIM(I), I=1,6)
                                                                                              00047500
                                                                                              00047600
C**PARTIALLY CHECK THESE VALUES
                                                                                              00047700
C**
                                                                                              00047800
       ICHK=0
                                                                                              00047900
       DO 239 I=1,6
                                                                                              00048000
           IF( DEL(I).LT.DELMN(I) ) WRITE(LUW,240)I
                                                                                              00048100
           FORMAT(' !?! ERROR -- THE ',Il,'TH INITIAL STEP SIZE IS ',
'LESS THAN ITS MIN. !',/,' DO IT OVER !')
  240
                                                                                              00048200
                                                                                              00048300
           IF( DEL(I).LT.DELMN(I) ) ICHK=1

IF( DEL(I).GT.DELMX(I) ) WRITE(LUW,242)I

FORMAT(' !?! ERROR -- THE ',II,'TH INITIAL STEP SIZE IS ',

'MORE THAN ITS MAX. !',/,' DO IT OVER !')
                                                                                              00048400
                                                                                              00048500
  242
                                                                                              00048600
                                                                                              00048700
           IF( DEL(I).GT.DELMX(I) ) ICHK=1

IF( DELMN(I).GT.DELMX(I) ) WRITE(LUW,244)I

FORMAT(' !?! ERROR -- THE ',II,'TH MIN. LIMIT IS ',

'MORE THAN THE ',II,'TH MAX. !',/,' DO IT OVER !')

IF( DELMN(I).GT.DELMX(I) ) ICHK=1
                                                                                              00048800
                                                                                              00048900
  244
                                                                                              00049000
                                                                                              00049100
                                                                                              00049200
  239 CONTINUE
                                                                                              00049300
C*.
                                                                                              00049400
C*.IF A VALUE IS ENTERED INCORRECTLY , RETURN
                                                                                              00049500
                                                                                              00049600
       IF(ICHK.EO.1) GO TO 230
                                                                                              00049700
                                                                                              00049800
C*.IF EVERYTHING IS CORRECT, THEN PRINT THEM OUT
                                                                                              00049900
                                                                                              00050000
       IF(ICHK.EQ.0) GO TO 205
                                                                                              00050100
C**
                                                                                              00050200
C**THE USER SPECIFIES MAXIMUM NUMBER OF ITERATIONS
                                                                                              00050300
C**
                                                                                              00050400
  250 WRITE(LUW, 251)
                                                                                              00050500
  251 FORMAT(T5,' ENTER VALUES: ITRMX1, ITRMX2, ITRMX3')
                                                                                              00050600
       READ(LUR, *) ITRMX1, ITRMX2, ITRMX3
                                                                                              00050700
C*.
                                                                                              00050800
C*.PARTIALLY CHECK THESE VALUES
                                                                                              00050900
C*.
                                                                                              00051000
       IF(ITRMX1.LT.1) GO TO 256
                                                                                              00051100
       IF(ITRMX2.LT.1) GO TO 256
IF(ITRMX3.LT.1) GO TO 256
                                                                                              00051200
                                                                                              00051300
C*.
                                                                                              00051400
C*.IF EVERYTHING IS CORRECT, THEN PRINT THEM OUT
                                                                                              00051500
C*.
                                                                                              00051600
       GO TO 205
                                                                                              00051700
C*.
                                                                                              00051800
C*.IF A VALUE IS ENTERED INCORRECTLY, THEN RETURN
                                                                                              00051900
                                                                                              00052000
  256 WRITE(LUW.257)
                                                                                              00052100
```

```
257 FORMAT(' !?! ERROR -- THE MAX. NUMBER OF ITERATIONS SHOULD',
                                                                                                                   00052200
       * ' BE AT LEAST 1.',/,' DO IT OVER !')
                                                                                                                   00052300
         GO TO 250
                                                                                                                   00052400
C*
                                                                                                                   00052500
                                                                                                                   00052600
C*
                                                                                                                   00052700
   300 WRITE(LUW, 303)
                                                                                                                   00052800
   303 FORMAT(/,1x,71('*'),/,1x,14('*'),2x,'ECON. DESIGN OF THE DYNAMIC',00052900
* 'X-BAR CHART',2x,14('*'),/)
00053000
   WRITE(LUW, 305) THETA, ETA, WBMEAN 00053100
305 FORMAT(T5,' DISTRIBUTION INFORMATION;',/, 00053200
*T7,' WEIBULL W/ THETA=',F10.6,' AND ETA=',F10.6,' => MEAN=',F10.4)00053300
                                                                                                                   00053400
   WRITE(LUW,307)DELTA,B,C,DD,E,VZMV1,T,W

307 FORMAT(T5,' COST AND OTHER INFORMATION:',/,
*T7,' DELTA=',F10.4,' B=',F10.4,' C=',F10.4,' D=',F10.4,/,
*T7,' E=',F10.4,' M=',F10.4,' T=',F10.4,' W=',F10.4)
                                                                                                                   00053500
                                                                                                                   00053600
                                                                                                                   00053800
                                                                                                                   00053900
         WRITE(LUW, 309) IN, IH, IK, NF, HF, KF
                                                                                                                   00054000
  WRITE(LUW, 309)IN, IH, IK, NF, HF, KF

309 FORMAT(T5,' STARTING POINT FOR OPTIMIZATION IS:',/, 00054100

* T6,' IN=',I4,5x,' IH=',F10.4,' IK=',F10.4,/, 00054200

* T6,' NF=',F9.7,' HF=',F9.7,lx,' KF=',F9.7,/) 00054300

WRITE(LUW, 311)PROBPT 00054400

311 FORMAT(T5,' QUANTILE VALUE IS: ',F11.9,/) 00054500

WRITE(LUW, 313)ITRMX1, ITRMX2,ITRMX3

313 FORMAT(T5,' MAX. NUMBER OF ITERATIONS (LOSS-COST EVALUATIONS):',/,00054700

* T7,' ITRMX1=',I4,' ITRMX2=',I4,' ITRMX3=',I4./) 00054800

315 WRITE(LUW, 316)
   315 WRITE(LUW, 316)
316 FORMAT( ' CHECK THE ABOVE INFORMATION.',/,
* ' EVERYTHING IS CORRECT ? 1=YES 2=NO')
                                                                                                                   00054900
                                                                                                                   00055000
                                                                                                                   00055100
         READ(LUR, *) IYN7
                                                                                                                   00055200
         GO TO (330,321), IYN7
                                                                                                                   00055300
         WRITE(LUW, 318)
                                                                                                                   00055400
   318 FORMAT(' !?! ERROR -- DO IT OVER !')
                                                                                                                   00055500
         GO TO 315
                                                                                                                   00055600
   321 WRITE(LUW, 322)
                                                                                                                   00055700
   322 FORMAT(1x,71('*'),/)
                                                                                                                   00055800
C**
                                                                                                                    00055900
C**GO BACK TO THE MENU
                                                                                                                    00056000
C**
                                                                                                                    00056100
         GO TO 20
                                                                                                                    00056200
                                                                                                                    00056300
C**
                                                                                                                   00056400
   330 CALL DYNOPT
                                                                                                                    00056500
                                                                                                                   00056600
                                                                                                                    00056700
         WRITE(LUW, 340)
                                                                                                                   00056800
   340 FORMAT(/,T20,'** THE OPTIMAL DYNAMIC DESIGN IS: **')
                                                                                                                   00056900
   WRITE(LUW, 342)NWOPT, HWOPT, RKWOPT, FNWOPT, FKWOPT

342 FORMAT( T6,' IN=', I4,5X,' IH=',F10.4,' IK=',F10.4,/,

* T6,' NF=',F9.7,' HF=',F9.7,1X,' KF=',F9.7,/)

WRITE(LUW, 344)YFWOPT
                                                                                                                   00057000
                                                                                                                   00057100
                                                                                                                   00057200
                                                                                                                    00057300
   344 FORMAT(1x,10('*'),' LOSS-COST PER 100 HOURS = $',F11.3)
                                                                                                                    00057400
   WRITE(LUW,346)
346 FORMAT(1X,71('*'),/,1X,71('*'))
                                                                                                                    00057500
                                                                                                                    00057600
                                                                                                                    00057700
                                                                                                                    00057800
   348 FORMAT(' DO YOU WANT TO EMPLOY ANOTHER PASS OF OPTIMIZATION,',
                                                                                                                    00057900
        * /,' STARTING WITH THE BEST SOLUTION FOUND SO FAR?',/
* '1=YES 2=NO,RETURN TO THE PREVIOUS MENU')
                                                                                                                    00058000
                                                                                                                    00058100
         READ(LUR, *)IYN8
                                                                                                                    00058200
         GO TO(350,20),1YN8
                                                                                                                    00058300
С
                                                                                                                    00058400
C**
                                                                                                                   00058500
C**FOR THE SECOND PASS OF OPTIMIZATION, SET THE STARTING POINT TO
                                                                                                                    00058600
C**THE BEST PINT FOUND SO FAR.
                                                                                                                   00058700
```

```
C**
                                                                                    00058800
  350 IN=NWOPT
                                                                                    00058900
                                                                                    00059000
      IH=HWOPT
       IK=RKWOPT
                                                                                    00059100
                                                                                    00059200
      NF=FNWOPT
      HF=FHWOPT
                                                                                    00059300
      KF=FKWOPT
                                                                                    00059400
C*.REPEAT THE OPTIMIZATION STARTING FROM THIS NEW POINT
                                                                                    00059500
      GO TO 300
                                                                                    00059600
С
                                                                                    00059700
C*
                                                                                    00059800
             ECON. EVALUATION IN THE WEIBULL ENVIRONMENT
                                                                                    00059900
                                                                                    00060000
  390 WRITE(LUW, 391)
                                                                                    00060100
  391 FORMAT(' *** FOR ECON. EVALUATION IN THE WEIBULL ENVIRONMENT,',/ 00060200
      * T5,' ENTER:',/,5X,'IN, IH, IK, NF, HF, KF')
                                                                                    00060300
      READ(LUR, *) IN, IH, IK, NF, HF, KF
                                                                                    00060400
                                                                                    00060500
C**CHECK TO SEE IF THESE ARE IN THE ACCEPTABLE RANGE
                                                                                    00060600
                                                                                    00060700
       IF(IN.LT.1000 .AND.IN.GE.2 ) GO TO 394
    WRITE(LUW,393)
                                                                                    00060800
                                                                                    00060900
           FORMAT(' !?! ERROR -- IN SHOULD BE BETWEEN 2 AND 1000'
  393
                                                                                    00061000
                    ,/,' DO IT OVER !')
                                                                                    00061100
           GO TO 390
                                                                                    00061200
  394 IF(IH.GT.0.0 .AND.IH.LT.100. ) GO TO 398
                                                                                    00061300
           WRITE(LUW, 396)
                                                                                    00061400
           FORMAT(' !?! ERROR -- IH SHOULD BE BETWEEN 0.0 AND 100.'
  396
                                                                                    00061500
                    ,/,' DO IT OVER !')
                                                                                    00061600
           GO TO 390
                                                                                    00061700
  398 IF( IK.GT.0.0 .AND. IK.LT.12. ) GO TO 402
                                                                                    00061800
           WRITE(LUW, 399)
                                                                                    00061900
           FORMAT(' !?! ERROR -- IK SHOULD BE BETWEEN 0.0 AND 12.'
,/,' DO IT OVER !')
                                                                                    00062000
  399
                                                                                    00062100
  402 IF( NF.GT.2.D0 .OR. NF.LT.0.D0 ) GO TO 432 IF( HF.GT.2.D0 .OR. HF.LT.0.D0 ) GO TO 432 IF( KF.GT.2.D0 .OR. KF.LT.0.D0 ) GO TO 432
                                                                                    00062200
                                                                                    00062300
                                                                                    00062400
       GO TO 455
                                                                                    00062500
                                                                                    00062600
  432
           WRITE(LUW, 433)
                                                                                    00062700
           FORMAT(' !?! ERROR -- NF, NH, AND NK SHOULD BE BETWEEN ','0.0 AND 2.',/,' DO IT OVER !')
                                                                                    00062800
                                                                                    00062900
            GO TO 390
                                                                                    00063000
C**
                                                                                    00063100
C**ECHOPRINT THE VALUES FOR CHECK
                                                                                    00063200
C**
                                                                                    00063300
  455 WRITE(LUW, 458)IN, IH, IK, NF, HF, KF
458 FORMAT(' VALUES ENTERED: IN=', I4,5X,' IH=',F9.4,4X,' IK='
* ,F9.4,/,T17,' NF=',F9.7,' HF=',F9.7,4X,' KF=',F9.7)
                                                                                    00063400
                                                                                    00063500
                                                                                    00063600
  459 WRITE(LUW, 460)
                                                                                    00063700
  460 FORMAT(' CORRECT ?
                               1=YES 2=NO 3=RETURN TO THE PREVIOUS MENU') 00063800
       READ(LUR, *)IYN3
                                                                                    00063900
       GO TO (480,390,20),IYN3
                                                                                    00064000
       WRITE(LUW, 461)
                                                                                    00064100
  461 FORMAT(' !?! ERROR -- DO IT OVER !')
                                                                                    00064200
       GO TO 459
                                                                                    00064300
C*
                                                                                    00064400
                                                                                    00064500
C....WHEN THE DESIGN IS ACCEPTED...
                                                                                    00064600
                                                                                    00064700
C....THEN SUGGEST THE QUANTILE.....
                                                                                    00064800
                                                                                    00064900
  480 PROBPT=.99D0
                                                                                    00065000
   482 WRITE(LUW, 484)PROBPT
                                                                                    00065100
  484 FORMAT(' QUANTILE VALUE OF ',F11.9,' IS USED.',

* /,' YOU ACCEPT THIS.')
                                                                                    00065200
                                                                                    00065300
```

```
485 WRITE(LUW, 486)
  486 FORMAT(' CORRECT ? 1=YES 2=NO 3=RETURN TO THE PREVIOUS MENU') 00065500
       READ(LUR, *)IYN5
                                                                                             00065600
       GO TO (500,490,20),IYN5
                                                                                             00065700
       WRITE(LUW, 488)
                                                                                             00065800
  488 FORMAT(' !?! ERROR -- DO IT OVER !')
                                                                                             00065900
       GO TO 485
                                                                                             00066000
C_490 WRITE(LUW, 493)
        .....IF QUANTILE VALUE IS NOT ACCEPTED .....
                                                                                             00066100
                                                                                             00066200
  493 FORMAT(' ENTER YOUR DESIRED QUANTILE:')
                                                                                             00066300
       READ(LUR, *) PROBPT
                                                                                             00066400
       GO TO 482
                                                                                             00066500
00066600
                                                                                             00066700
C**ECONOMICALLY EVALUATE THIS DESIGN
                                                                                             00066800
                                                                                             00066900
  500 WRITE(LUW,521)
                                                                                             00067000
  500 WRITE(LUW,521)
521 FORMAT(1X,71('*'),/,1X,14('*'),2X,'ECON. EVALUATION IN ', 00067100

* 'WEIBULL ENVIRONMENT',2X,14('*'),/) 00067200

WRITE(LUW,523)THETA,ETA,WBMEAN 00067300

523 FORMAT(T5,' DISTRIBUTION INFORMATION;',/, 00067400

*T7,' WEIBULL W/ THETA=',F10.6,' AND ETA=',F10.6,' => MEAN=',F10.4)00067500
                                                                                             00067600
       WRITE(LUW, 525) DELTA, B, C, DD, E, VZMV1, T, W
                                                                                             00067700
  525 FORMAT(T5,' COST AND OTHER INFORMATION:',/,
*T7,' DELTA=',F10.4,' B=',F10.4,' C=',F10.4,' D=',F10.4,/,
*T7,' E=',F10.4,' M=',F10.4,' T=',F10.4,' W=',F10.4)
                                                                                             00067800
                                                                                             00067900
                                                                                             00068000
C
                                                                                             00068100
  WRITE(LUW,526)IN,IH,IK,NF,HF,KF
526 FORMAT(T5,' *** THE DESIGN TO BE EVALUATED IS: ***',/,
                                                                                             00068200
                                                                                             00068300
      * T6,' IN=',I4,5X,' IH=',F10.4,' IK=',F10.4,/,

* T6,' NF=',F9.7,' HF=',F9.7,1X,' KF=',F9.7,/)
                                                                                             00068400
                                                                                             00068500
  WRITE(LUW,527)PROBPT

527 FORMAT(T5,' QUANTILE VALUE IS:',F11.9,/)

530 WRITE(LUW,536)

536 FORMAT(' CHECK THE ABOVE INFORMATION.',/,

* ' EVERYTHING IS CORRECT? 1=YES 2=NO')
                                                                                             00068600
                                                                                             00068700
                                                                                             00068800
                                                                                             00068900
                                                                                             00069000
       READ(LUR.*)IYN9
                                                                                             00069100
        GO TO (580,541), IYN9
                                                                                             00069200
       WRITE(LUW, 538)
                                                                                             00069300
  538 FORMAT(' !?! ERROR -- DO IT OVER !')
                                                                                             00069400
       GO TO 315
                                                                                             00069500
  541 WRITE(LUW,542)
542 FORMAT(1X,71('*'),/)
                                                                                             00069600
                                                                                             00069700
C**
                                                                                             00069800
C**GO BACK TO THE MENU
                                                                                             00069900
C**
                                                                                             00070000
       GO TO 20
                                                                                             00070100
С
                                                                                             00070200
C**
                                                                                             00070300
C**CALCULATE ISTEPS FOR THE DESIGN TO BE EVALUATED.
                                                                                             00070400
                                                                                             00070500
  580 ISTEPS=CSTPW(HF.IH)
                                                                                             00070600
      ISTPP=ISTEPS
                                                                                             00070700
                                                                                             00070800
C*.SINCE WE CANNOT EVALUATE EXACTLY THE SAME DESIGN, FOR
                                                                                             00070900
C*.THE GIVEN QUANTILE VALUE, GIVE A RANGE
                                                                                             00071000
C*.OF LOSS-COST VALUES WHICH INCLUEDS THE LOSS-COST OF THE DESIRED
                                                                                             00071100
                                                                                             00071200
Č*.
                                                                                             00071300
C..CALCULATE THE IH DETERMINED BY THE ABOVE ISTEPS AND QUANTILE
                                                                                             00071400
с..
                                                                                             00071500
       TEMPIH=IH
                                                                                             00071600
       IH=SINTW(HF, ISTEPS)
                                                                                             00071700
С
                                                                                             00071800
       DO 581 I=1.2
                                                                                             00071900
```

```
C**
                                                                      00072000
         CALL DYMEVA
                                                                      00072100
                                                                      00072200
        WRITE(LUW,584)PROBPT,IN,IH,IK,NF,HF,KF

FORMAT(T5,'*** FOR THE FOLLOWING DESIGN QUANTILE IS FIXED AT '00072400

F10.8 ,/,T6,' IN=',I4,5X,' IH=',F10.4,' IK=',F10.4,/,

T6,' NF=',F9.7,' HF=',F9.7,1X,' KF=',F9.7)

O0072500

O0072500
  584
         WRITE(LUW,587)DYMLCS
FORMAT(T5,' LOSS-COST PER 100 HOURS = $',F11.3,/)
                                                                      00072700
  587
                                                                      00072800
         ISTEPS=ISTEPS+1
                                                                      00072900
         ISTPP=ISTEPS
                                                                      00073000
         IH=SINTW(HF, ISTEPS)
                                                                      00073100
  581 CONTINUE
                                                                      00073200
C*.
                                                                      00073300
C*.NOW EVALUATE EXACTLY THE SAME DESIGN AS THE USER WANTS WHILE
                                                                      00073400
C*.ACHIEVING A SLIGHTLY DIFFERENT QUANTILE.
                                                                      00073500
                                                                      00073600
      IH=TEMPIH
                                                                      00073700
C**
                                                                      00073800
                                                                      00073900
     CALL DYMEVA
C**
                                                                      00074000
     WRITE(LUW, 588) CUPROX, IN, IH, IK, NF, HF, KF
                                                                      00074100
  588 FORMAT(T5,'*** FOR THE FOLLOWING DESIGN, THE ACTUAL QUANTILE'
                                                                      00074200
        IS ', F10.8 ,/,T6,' IN=',I4,5X,' IH=',F10.4,' IK=',F10.4,/, 00074300 T6,' NF=',F9.7,' HF=',F9.7,1X,' KF=',F9.7) 00074400
     WRITE (LUW, 587) DYMLCS
                                                                      00074500
C**
                                                                      00074600
      WRITE(LUW, 589)
                                                                      00074700
  589 FORMAT(1x,71('*'),/,1x,71('*'))
                                                                      00074800
C**
                                                                      00074900
C**RETURN TO THE MAIN MENU
                                                                      00075000
C**
                                                                      00075100
     GO TO 20
                                                                      00075200
С
                                                                      00075300
C**
                                                                      00075400
C**EXIT; RETURN TO THE MAIN MENU
                                                                      00075500
C**
                                                                      00075600
  600 RETURN
                                                                      00075700
                                                                      00075800
C
                                                                      00075900
С
                                                                      00076000
Č
                                                                      00076100
                                                                      00076200
                                                                      00076300
SUBROUTINE DUNC
                                                                      00076700
C**
                                                                    * 00076900
C** THIS ROUTINE PROMPTS THE USER FOR THE NECESSARY INFORMATION
                                                                    * 00077000
C** NEEDED FOR DUNCAN'S X-BAR CHART DESIGN OR EVALUATION.
                                                                    * 00077100
C**
                                                                    * 00077200
C** THIS ROUTINE CALLS THE FOLLOWING SUBROUTINES:
                                                                    * 00077300
C**
        DUNOPT-- TO OPTIMZE DUNCAN'S MODEL.
                                                                    * 00077400
        DUNEVA -- EVALUATE DUNCAN'S COST MODEL FOR A GIVEN DESIGN.
C**
                                                                    * 00077500
C**
                                                                    * 00077600
C**
                                                                      00077700
C**
                                                                    * 00077800
C**
                                                                    * 00077900
     ****************** 00078000
C***
C**
                                                                      00078100
C*
                                                                      00078200
      IMPLICIT REAL*8(A-H,O-Z)
                                                                      00078300
      REAL*8 LAMBDA
                                                                      00078400
      COMMON / MAIN1 /LUR, LUW
                                                                      00078500
```

```
COMMON / DUNC1 / LAMBDA
                                                                                                          00078600
        COMMON / DUNC4 / N,H,RK
                                                                                                          00078700
        COMMON / DUNC5 / NOCOPT, HDCOPT, RKDCOP, FDCOPT
COMMON / DUNC6 / NTPRNT(20), HTPRNT(20), RKPRNT(20), FTPRNT(20)
COMMON / DUNC7 / DCLCST
COMMON / DCDY1 / DELTA, B,C,DD,E,VZMV1,T,W
                                                                                                          00078800
                                                                                                          00078900
                                                                                                          00079000
                                                                                                          00079100
C*
                                                                                                          00079200
C*
                                                                                                          00079300
              ENTER DISTRIBUTION, COST, AND OTHER PARAMETERS
                                                                                                          00079400
                                                                                                          00079500
    10 WRITE(LUW, 11)
                                                                                                          00079600
    11 FORMAT(/,T5,' >> EXPONENTIAL ENVIRONMENT <<',/,
* 55H *** FOR DUNCAN'S ECONOMIC X-BAR CHART, ENTER VALUES:
                                                                                                          00079700
                                                                                                          00079800
        *,/,T5,' LAMBDA, DELTA, B, C, D, E, M, T, W')
READ(LUR,*)LAMBDA, DELTA, B,C,DD,E,VZMV1, T, W
                                                                                                          00079900
                                                                                                          00080000
C**
                                                                                                          00080100
C**CALCULATE MEAN AND VARIANCE OF THE EXPONENTIAL
                                                                                                          00080200
C**
                                                                                                          00080300
        XPMEAN=1.D0/LAMBDA
                                                                                                          00080400
        XPVAR=XPMEAN/LAMBDA
                                                                                                          00080500
C**
                                                                                                          00080600
        WRITE(LUW, 14) LAMBDA, XPMEAN, DELTA, B, C, DD, E, VZMV1, T, W
                                                                                                          00080700
    14 FORMAT(' VALUES ENTERED ARE:',/,

* T5,' DISTRIBUTION INFORMATION;',/,

*T7,' EXPONENTIAL W/ LAMBDA= ',F10.4,' => MEAN=',F10.4,/,
                                                                                                          00808000
                                                                                                          00080900
                                                                                                          00081000
       * T5,' COST AND OTHER INFORMATION:',/,
*T7,' DELTA=',F10.4,' B=',F10.4,' C=',F10.4,' D=',F10.4,/,
*T7,' E=',F10.4,' M=',F10.4,' T=',F10.4,' W=',F10.4)
                                                                                                          00081100
                                                                                                          00081200
                                                                                                          00081300
C*
                                                                                                          00081400
            SELECTION FOR DESIGN, EVALUATION, ETC.
                                                                                                          00081500
                                                                                                          00081600
    18 WRITE(LUW,19)

19 FORMAT(/,' *** ENTER OPTION NUMBER',/,

* T5, ' 1 = ECON. DESIGN OF DUNCAN,S X-BAR CHART (OPTIMIZATION)' 00081900

* ,/,T5,' 2 = ECON. EVALUATION IN THE EXPONENTIAL ENVIRONMENT',/, 00082000

* T5,' 3 = RETURN TO REVISE COST AND DISTRIBUTION PARAMETERS',/, 00082100

* T5,' 4 = RETURN TO THE MAIN MENU')

00082200
         READ(LUR, *)MENU2
                                                                                                          00082300
         GO TO (100,200,10,400), MENU2
                                                                                                          00082400
         WRITE(LUW, 20)
                                                                                                          00082500
    20 FORMAT(' !?! ERROR -- DO IT OVER !')
                                                                                                          00082600
                                                                                                          00082700
         GO TO 18
C*
                                                                                                          00082800
              ECON. DESIGN (OPTIMIZATION) OF DUNCAN'S X-BAR CHART
                                                                                                          00082900
C<del>≭</del>
                                                                                                          00083000
C**
                                                                                                          00083100
C**INITIALIZATION OF STARTING POINT FOR OPTIMIZATION
                                                                                                          00083200
C**
                                                                                                          00083300
   100 N=5
                                                                                                          00083400
        H=1.D0
                                                                                                          00083500
         RK=3.D0
                                                                                                          00083600
         WRITE(LUW, 102)N, H, RK
                                                                                                          00083700
   102 FORMAT( 56H *** FOR ECON. OPTIMIZATION OF DUNCAN'S X-BAR CHART, 
* ,/,T5,' THE FOLLOWING STARTING POINT IS SUGGESTED:',/,

* T5,' N=',I4,' H=',F10.4,' K=',F10.4,/,

* T5,' YOU ACCEPT THIS POINT.')
                                                                                                          00083800
                                                                                                          00083900
                                                                                                          00084000
                                                                                                          00084100
   103 WRITE(LUW, 104)
                                                                                                          00084200
   104 FORMAT(' CORRECT ?
                                       1=YES 2=NO 3=RETURN TO THE PREVIOUS MENU') 00084300
         READ(LUR, *) I YN1
                                                                                                          00084400
         GO TO (110,150,18),1YN1
                                                                                                          00084500
         WRITE(LUW, 106)
                                                                                                          00084600
   106 FORMAT(' !?! ERROR -- DO IT OVER !')
                                                                                                          00084700
         GO TO 103
                                                                                                          00084800
                                                                                                          00084900
C....IF THE SUGGESTED STARTING POINT IS ACCEPTED...
                                                                                                          00085000
                                                                                                          00085100
```

```
110 WRITE(LUW,111)
                                                                                    00085200
  111 FORMAT(1x,66('*'),/,1x,13('*'),2x,24HECON. DESIGN OF DUNCAN'S
                                                                                     00085300
      * , ' X-BAR CHART', 2X, 13('*'), /)
                                                                                     00085400
                                                                                     00085500
       WRITE(LUW, 112)LAMBDA, XPMEAN, DELTA, B, C, DD, E, VZMV1, T, W
                                                                                     00085600
  112 FORMAT(' VALUES ENTERED ARE:',/,
                                                                                     00085700
            T5,' DISTRIBUTION INFORMATION;',/,
EXPONENTIAL W/ LAMBDA= ',F10.4,' => MEAN=',F10.4,/,
                                                                                     00085800
                                                                                     00085900
              T5,' COST AND OTHER INFORMATION:',/,
                                                                                     00086000
          ,' DELTA=',F10.4,' B=',F10.4,' C=',F10.4,' D=',F10.4,/,
,' E=',F10.4,' M=',F10.4,' T=',F10.4,' W=',F10.4,/)
                                                                                     00086100
                                                                                     00086200
  115 WRITE(LUW,116)N,H,RK 00086300
116 FORMAT(T5,' STARTING POINT FOR OPTIMIZATION IS:',/,lx,T6,' N=',I4,00086400

* ' H=', F8.4, ' K=',F8.4,/,' CHECK THE ABOVE INFORMATION.',/, 00086500

* ' EVERYTHING IS CORRECT ? l=YES 2=NO') 00086700
      READ(LUR, *)IYN2
                                                                                     00086700
       GO TO (130,120), IYN2
                                                                                     00086800
       WRITE(LUW, 118)
                                                                                     00086900
  118 FORMAT(' !?! ERROR -- DO IT OVER !')
                                                                                     00087000
       GO TO 115
                                                                                     00087100
  120 WRITE(LUW, 121)
                                                                                     00087200
  121 FORMAT(1X,66('*'),/)
                                                                                     00087300
       GO TO 18
                                                                                     00087400
C
                                                                                     00087500
C**
                                                                                     00087600
  130 CALL DUNOPT
                                                                                     00087700
C**
                                                                                     00087800
                                                                                    .00087900
       WRITE(LUW, 132)
                                                                                    00088000
  132 FORMAT(T10,'N',T16,'H',T26,'K',T34,'LOSS-COST',/)
                                                                                     00088100
C**
                                                                                     00088200
C**PRINT OPTIMIZATION ITERATIONS
                                                                                     00088300
C**
                                                                                     00088400
       DO 133 I=1,20
                                                                                     00088500
                                                                                     00088600
C*.IF NO MORE INFORMATION IS AVAILABLE IN THE ARRAYS, THEN QUIT THE LOOP00088700
C*.
                                                                                     00088800
          IF(FTPRNT(I).LT.1.E-10) GO TO 142
                                                                                     00088900
               WRITE(LUW, 135)NTPRNT(I), HTPRNT(I), RKPRNT(I), FTPRNT(I)
                                                                                     00089000
  135
               FORMAT(T8, I3, T13, F7.4, T23, F7.4, T33, F11.3)
                                                                                     00089100
  133 CONTINUE
                                                                                     00089200
                                                                                     00089300
  142 WRITE(LUW, 143)
                                                                                     00089400
  143 FORMAT(/,T10,38H ** THE OPTIMAL DUNCAN'S DESIGN IS: **
                                                                                     00089500
       WRITE(LUW, 145)NDCOPT, HDCOPT, RKDCOP
                                                                                     00089600
                                 H=',F8.4,' K=',F8.4,/)
  145 FORMAT(T10,' N=',14,'
                                                                                     00089700
       WRITE(LUW, 147) FDCOPT
                                                                                     00089800
  147 FORMAT(1x,7('*'),' THE MIN. LOSS-COST PER 100 HOURS = $',F11.3)
                                                                                     00089900
       WRITE(LUW, 149)
                                                                                     00090000
  149 FORMAT(1x,66('*'),/1x,66('*'),/)
                                                                                     00090100
C**
                                                                                     00090200
C**GO BACK TO MENU
                                                                                     00090300
C**
                                                                                     00090400
       GO TO 18
                                                                                     00090500
                                                                                     00090600
C. ....IF THE SUGGESTED STARTING POINT IS NOT ACCEPTED...
                                                                                     00090700
                                                                                     00090800
  150 WRITE(LUW, 151)
                                                                                     00090900
  151 FORMAT(T5,' FOR YOUR DESIRED STARTING POINT FOR OPTIMIZATION, ', 'ENTER:',/,5X,' N, H, K')
                                                                                    00091000
                                                                                     00091100
      READ(LUR, *)N, H, RK
                                                                                     00091200
C**
                                                                                     00091300
C**CHECK TO SEE IF THESE ARE IN THE ACCEPTABLE RANGE
                                                                                     00091400
C**
                                                                                    00091500
       IF( N.LT.1000 .AND. N.GE.2 ) GO TO 153
                                                                                    00091600
           WRITE(LUW, 152)N
                                                                                    00091700
```

```
FORMAT(' !?! ERROR -- N SHOULD BE BETWEEN 2 AND 1000'
  152
                                                                                    00091800
                    ,/,' DO IT OVER !')
                                                                                    00091900
           GO TO 150
                                                                                    00092000
  153 IF( H.GT.0.0 .AND. H.LT.100. ) GO TO 155
                                                                                    00092100
           WRITE(LUW, 154)H
                                                                                    00092200
           FORMAT(' !?! ERROR -- H SHOULD BE BETWEEN 0.0 AND 100.'
  154
                                                                                    00092300
                    ,/,' DO IT OVER !')
                                                                                    00092400
           GO TO 150
                                                                                    00092500
  155 IF( RK.GT.0.0 .AND. RK.LT.12. ) GO TO 157
                                                                                    00092600
           WRITE(LUW, 156)RK
                                                                                    00092700
           FORMAT(' !?! ERROR -- K SHOULD BE BETWEEN 0.0 AND 12.'
  156
                                                                                    00092800
                   ,/,' DO IT OVER !')
                                                                                    00092900
           GO TO 150
                                                                                    00093000
C**
                                                                                    00093100
C**ECHOPRINT THE VALUES FOR CHECK
                                                                                    00093200
C**
                                                                                    00093300
  157 WRITE(LUW, 158)N, H, RK
                                                                                    00093400
  158 FORMAT(T5,' VALUES ENTERED: N=',14,4X,' H=',F8.4,4X,' K=',F8.4)00093500
159 WRITE(LUW,160) 00093600
  160 FORMAT(' CORRECT ? READ(LUR,*)IYN3
                               1=YES 2=NO 3=RETURN TO THE PREVIOUS MENU') 00093700
                                                                                    00093800
      GO TO (110,150,18), IYN3
                                                                                    00093900
      WRITE(LUW, 161)
                                                                                    00094000
  161 FORMAT(' !?! ERROR -- DO IT OVER !')
                                                                                    00094100
      GO TO 159
                                                                                    00094200
C*
                                                                                    00094300
C*
                                                                                    00094400
             ECON. EVALUATION IN THE EXPONENTIAL ENVIRONMENT
                                                                                    00094500
                                                                                    00094600
  200 WRITE(LUW, 201)
                                                                                    00094700
  201 FORMAT(' *** FOR ECON. EVALUATION IN THE EXPONENTIAL',

* 'ENVIRONMENT,',/,T5,' ENTER VALUES: N, H, K')
READ(LUR,*)N,H,RK
                                                                                    00094800
                                                                                    00094900
                                                                                    00095000
  WRITE(LUW,205)N,H,RK 00095100
205 FORMAT(T5,' VALUES ENTERED: N=',14,4X,' H=',F8.4,4X,' K=',F8.4)00095200
206 WRITE(LUW,208) 00095300
  208 FORMAT(' CORRECT ?
                               1=YES 2=NO 3=RETURN TO THE PREVIOUS MENU') 00095400
      READ(LUR, *)IYN4
                                                                                    00095500
      GO TO (220,200,18),IYN4
                                                                                    00095600
      WRITE(LUW, 209)
                                                                                    00095700
  209 FORMAT(' !?! ERROR -- DO IT OVER !')
                                                                                    00095800
      GO TO 206
                                                                                    00095900
C**
                                                                                    00096000
C**ECONOMICALLY EVALUATE THIS DESIGN
                                                                                    00096100
C**
                                                                                    00096200
  220 WRITE(LUW, 221)
                                                                                    00096300
  221 FORMAT(1X,66('*'),/,1X,9('*'),2X,'ECON. EVALUATION IN ',
                                                                                    00096400
          'EXPONENTIAL ENVIRONMENT', 2X, 9('*'),/)
                                                                                    00096500
                                                                                    00096600
  WRITE(LUW,224)LAMBDA,XPMEAN,DELTA,B,C,DD,E,VZMV1,T,W
224 FORMAT(T5,' DISTRIBUTION INFORMATION;',/,
*T7,' EXPONENTIAL W/ LAMBDA= ',F10.4,' => MEAN=',F10.4,/,
                                                                                    00096700
                                                                                    00096800
                                                                                    00096900
     * T5,' COST AND OTHER INFORMATION:',/,

*T7,' DELTA=',F10.4,' B=',F10.4,' C=',F10.4,' D=',F10.4,/,

*T7,' E=',F10.4,' M=',F10.4,' T=',F10.4,' W=',F10.4)
                                                                                    00097000
                                                                                    00097100
                                                                                    00097200
      WRITE(LUW, 226)N,H,RK
                                                                                    00097300
  00097400
                                                                                    00097500
                                                                                    00097600
  228 FORMAT(' EVERYTHING IS CORRECT ? 1=YES 2=NO')
                                                                                    00097700
      READ(LUR, *) IYN5
                                                                                    00097800
      GO TO (250,240),1YN5
                                                                                    00097900
      WRITE(LUW, 229)
                                                                                    00098000
  229 FORMAT(' !?! ERROR -- DO IT OVER !')
                                                                                    00098100
      GO TO 227
                                                                                    00098200
  240 WRITE(LUW.241)
                                                                                    00098300
```

```
241 FORMAT(1X,66('*'),/)
                                                                     U0098400
     GO TO 18
                                                                     00098500
                                                                     00098600
C**
                                                                     00098700
 250 CALL DUNEVA
                                                                     00098800
                                                                     00098900
                                                                     00099000
     WRITE(LUW, 253)DCLCST
                                                                     00099100
  253 FORMAT(1X,7('*'),' LOSS-CAST PER 100 HOURS= $',F11.3)
                                                                     00099200
     WRITE(LUW, 255)
                                                                     00099300
 255 FORMAT(1X,66('*'),/,1X,66('*'),/)
                                                                     00099400
C**
                                                                     00099500
C**GO BACK TO THE MENU
                                                                     00099600
C**
                                                                     00099700
     GO TO 18
                                                                     00099800
                                                                      00099900
C**RETURN TO THE MAIN MENU
                                                                      00100000
C**
                                                                      00100100
                                                                      00100200
  400 RETURN
     END
                                                                     00100300
C*
                                                                     00100400
C*
                                                                     00100500
C*
                                                                     00100600
C*
                                                                      00100700
SUBROUTINE DUNOPT
                                                                      00101100
C**
                                                                    * 00101300
C** THIS SUBROUTINE OPTIMIZES DUNCAN'S COST MODEL USING ZXMIN ROUTINE * 00101400
C** PROVIDED IN INTERNATIONAL MATHEMATICAL SCIENTIFIC LIBRARY (IMSL). * 00101500
C**
                                                                   * 00101600
                                                                    * 00101700
C** OPTIMIZATION IS PERFORMED IN TWO STAGES. IN THE FIRST STAGE
C** THE SAMPLE SIZE IS TREATED AS A REAL-VALUED VARIABLE AND THE
                                                                   * 00101800
C** LOSS-FUNCTION IS OPTIMIZED OVER ALL THREE VARIABLES. IN THE SECOND* 00101900
C** STAGE THE SAMPLE SIZE IS TREATED AS AN INTERGER-VALUED VARIABLE. * 00102000 C** THUS, SAMPLE SIZE IS SET TO A TENTATIVE VALUE, AND THE LOSS-COST * 00102100 C** FUNCTION IS THEN OPTIMIZED OVER THE OTHER TWO REMAINING VARIABLES.* 00102200
C**
                                                                    * 00102300
                                                                    * 00102400
C** THE FOLLOWING SUBROUTINES ARE CALLED BY THIS SUBROUTINE:
C**
        (1) SUBROUTINE ZXMIN
                                                                     00102500
        (2) SUBROUTINE FUNCT (THROUGH ZXMIN)
C**
                                                                    * 00102600
C**
                                                                    * 00102700
        (3) SUBROUTINE FUNCT2 (THROUGH ZXMIN)
C**
                                                                    * 00102800
C*****************************
                                                                      00102900
C**
                                                                      00103000
C*
                                                                      00103100
       IMPLICIT REAL*8(A-H,O-Z)
                                                                      00103200
        EXTERNAL FUNCT
                                                                      00103300
        EXTERNAL FUNCT2
                                                                      00103400
       COMMON / MAIN1 /LUR, LUW
                                                                      00103500
       COMMON / DUNC4 / INN,RH,RK
COMMON / DUNC5 / NDCOPT,HDCOPT,RKDCOP,FDCOPT
                                                                      00103600
                                                                      00103700
      COMMON / SAMPLS /IX
COMMON / DUNC6 / NTPRNT(20), HTPRNT(20), RKPRNT(20), FTPRNT(20)
                                                                      00103800
                                                                      00103900
       INTEGER N, NSIG, MAXFN, IOPT
                                                                      00104000
       REAL*8 FMIN(5)
                                                                      00104100
       REAL*4 X(3), H(6), G(3), W(9)
                                                                      00104200
                                                                      00104300
       DATA HTPRNT, RKPRNT, NTPRNT, FTPRNT/20*0.D0, 20*0.D0, 20*0, 20*0.D0/
                                                                      00104400
C*
                                                                      00104500
       NFUEVA=0
                                                                      00104600
C*
                                                                      00104700
C**N IS DIMENSIONALITY OF SEARCH
                                                                      00104800
C**
                                                                      00104900
```

```
C
                                                                           00105000
C**FIRST OPTIMIZE OVER ALL THE VARIABLES; CONSIDERING SAMPLE SIZE
                                                                           00105100
C**AS A REAL VARIABLE RATHER THAN INTEGER.
                                                                           00105200
                                                                           00105300
       N=3
                                                                           00105400
       NSIG=3
                                                                           00105500
       MAXFN=10000
                                                                           00105600
       IOPT=0
                                                                           00105700
C**
                                                                           00105800
C**INITIALIZE THE VARIABLES
                                                                           00105900
C**
                                                                           00106000
         X(1)=RH
                                                                           00106100
         X(2)=RK
                                                                           00106200
         X(3)=DFLOAT(INN)
                                                                           00106300
C*
                                                                           00106400
         CALL ZXMIN(FUNCT, N, NSIG, MAXFN, IOPT, X, H, G, F, W, IER)
                                                                           00106500
C*
                                                                           00106600
                                                                         - 00106700
C-
          -----TWO DIMENSIONAL SEARCH-----
                                                                           00106800
C
C**NOW OPTIMIZE OVER H AND K FOR KNOWN BUT DIFFERENT VALUES OF SAMPLE
                                                                           00106900
C**SIZE.
                                                                           00107000
С
                                                                           00107100
                                                                           00107200
C**
                                                                           00107300
C**SET SAMPLE SIZE TO THE INTEGER EQUIVALENT OF OPTIMUM FOUND IN
                                                                           00107400
C**THE PREVIOUS SEARCH MINUS TWO.
                                                                           00107500
C**
                                                                           00107600
      IX=X(3)-2.
                                                                           00107700
C*.
                                                                           00107800
C*.MINIMUM SAMPLE SIZE POSSIBLE IS 2.
                                                                           00107900
C*.
                                                                           00108000
      IF(IX.LT.2)IX=2
                                                                           00108100
C**
                                                                           00108200
C**FIND THE OPTIMAL ECONOMIC DESIGN FOR THIS INTEGER SAMPLE SIZE
                                                                           00108300
C**
                                                                           00108400
C*.INITIALIZE THE VARIABLES
                                                                           00108500
C
                                                                           00108600
         X(1)=RH
                                                                           00108700
         X(2) = RK
                                                                           00108800
         IOPT=0
                                                                           00108900
         NSIG=3
                                                                           00109000
C
                                                                           00109100
      CALL ZXMIN(FUNCT2, N, NSIG, MAXFN, IOPT, X, H, G, F, W, IER)
                                                                           00109200
C**
                                                                           00109300
C**KEEP THIS RESULTS IN A TEMPORARY ARRAY TO BE PRINTED ONLY IN THE
                                                                           00109400
C**CASE THE USER IS INTERESTED IN DUNCAN'S OPTIMUM DESIGN.
                                                                           00109500
C**
                                                                           00109600
      I=1
                                                                           00109700
      HTPRNT(I)=X(1)
                                                                           00109800
      RKPRNT(I)=X(2)
                                                                           00109900
      NTPRNT(I)=IX
                                                                           00110000
      FTPRNT(I)=F
                                                                           00110100
C**
                                                                           00110200
C**INCR SHOWS THE DIRECTION OF SEARCH ALONG THE SAMPLE SIZE DIRECTION
                                                                           00110300
C**
                                                                           00110400
      INCR=1
                                                                           00110500
      ITIME=0
                                                                           00110600
C**
                                                                           00110700
C**KEEP THE POINT AS THE BEST OPTIMUM SO FAR
                                                                           00110800
C**
                                                                           00110900
 7
      FMIN(5)=F
                                                                           00111000
      DO 8 I=1,2
                                                                           00111100
          FMIN(I)=X(I)
                                                                           00111200
 8
      CONTINUE
                                                                           00111300
C**
                                                                           00111400
C**INCREMENT OR DECREMENT SAMPLE SIZE BASED ON DIRECTION OF INCR
                                                                           00111500
```

```
C**
                                                                                 00111600
 9
      IX=IX+INCR
                                                                                 00111700
C**
                                                                                 00111800
C**FIND THE OPTIMAL DESIGN FOR THIS VALUE OF SAMPLE SIZE
                                                                                 00111900
C**
                                                                                 00112000
C*.INITIALIZE THE VARIABLES
                                                                                 00112100
C
                                                                                 00112200
10
          X(1)=RH
                                                                                 00112300
          X(2)=RK
                                                                                 00112400
          I.OPT=0
                                                                                 00112500
          NSIG=3
                                                                                 00112600
          G(1)=0.
                                                                                 00112700
          G(2)=0.
                                                                                 00112800
          F=0
                                                                                 00112900
       CALL ZXMIN(FUNCT2,N,NSIG,MAXFN,IOPT,X,H,G,F,W,IER)
                                                                                 00113000
C**
                                                                                 00113100
C**KEEP THE RESULTS IN A TEMPORARY ARRAY TO BE PRINTED ONLY IN THE
                                                                                 00113200
C**CASE THE USER IS INTERESTED IN DUNCAN'S OPTIMUM DESIGN.
                                                                                 00113300
C**
                                                                                 00113400
                                                                                 00113500
      I = I + 1
      HTPRNT(I)=X(1)
                                                                                 00113600
      RKPRNT(I)=X(2)
                                                                                 00113700
      NTPRNT(I)=IX
                                                                                 00113800
      FTPRNT(I)=F
                                                                                 00113900
C**
                                                                                 00114000
C**IF THIS IS THE FIRST INCREMENT IN SAMPLE SIZE...
                                                                                 00114100
C**
                                                                                 00114200
      IF(ITIME.EO.1) GO TO 23
                                                                                 00114300
                                                                                 00114400
C*, AND THERE IS AN IMPROVEMENT IN OBJECTIVE FUNCTION
                                                                                 00114500
Č*.
                                                                                 00114600
IF(F .GT.FMIN(5)) GO TO 13
C..THEN UPDATE THE BEST OPTIMUM
                                                                                 00114700
                                                                                 00114800
              ITIME=1
                                                                                 00114900
              FMIN(5)=F
                                                                                 00115000
              DO 11 I=1,2
                                                                                 00115100
                  FMIN(I)=X(I)
                                                                                 00115200
              CONTINUE
 11
                                                                                 00115300
C.. INCREMENT SAMPLE SIZE AND REPEAT
                                                                                 00115400
              IX=IX+1
                                                                                 00115500
              GO TO 10
                                                                                 00115600
C*.
                                                                                 00115700
C*.IF THE OBJECTIVE FUNCTION GETS WORSE , THEN SWITCH DIRECTION .
                                                                                 00115800
C*.ALSO FOR THIS FIRST SAMPLE SIZE DECREMENT IT BY 2.
                                                                                 00115900
C*.
                                                                                 00116000
 13
          INCR=-INCR
                                                                                 00116100
          IX=IX-2
                                                                                 00116200
          GO TO 10
                                                                                 00116300
C**
                                                                                 00116400
C**IF THERE IS AN IMPROVEMENT IN OBJ. FUN. AND THIS IS NOT THE FIRST
                                                                                 00116500
C**INCREMENT ON SAMPLE SIZE
                                                                                 00116600
C**THEN UPDATE FMIN AND KEEP GOING IN THIS DIRECTION
                                                                                 00116700
C**
                                                                                 00116800
 23
      IF(F .LE.FMIN(5)) GO TO 7
                                                                                 00116900
                                                                                 00117000
C**IF THE NEW OBJ. FUN. IS WORSE AND THIS IS NOT THE FIRST STEP TO C**INCREMENT THE SAMPLE SIZE , THEN FMIN ASSOCIATED WITH THE C**PREVIOUS SAMPLE SIZE TRIED IS THE GLOBAL MINIMUM.
                                                                                 00117100
                                                                                 00117200
                                                                                 00117300
C**
                                                                                 00117400
       IXMIN=IX-INCR
                                                                                 00117500
C*.
                                                                                 00117600
C*.STORE DUNCAN'S MODEL OPTIMAL DESIGN IN THE FOLLOWING:
                                                                                 00117700
C*.
                                                                                 00117800
         NDCOPT=IXMIN
                                                                                 00117900
         HDCOPT=FMIN(1)
                                                                                 00118000
         RKDCOP=FMIN(2)
                                                                                 00118100
```

```
00118200
C*.FDCOPT IS LOSS-COST PER 100 HOURS
                                                                     00118300
C*.
                                                                     00118400
       FDCOPT=FMIN(5)*100.D0
                                                                     00118500
       RETURN
                                                                     00118600
                                                                     00118700
                                                                     00118800
                                                                     00118900
C**********************************
                                                                     00119000
C***********************
                                                                     00119100
       SUBROUTINE FUNCT(N,X,F)
                                                                     00119200
C**
                                                                   * 00119400
C** THIS SUBROUTINE CALLS DIFFERENT SUBROUTINES TO CALCULATE DUNAN'S * 00119500
C** LOSS-COST FUNCTION NEEDE FOR 3-DIMENSIOAL SEARCH OVER H, K, AND N * 00119600
C**
                                                                   * 00119700
                                                                   * 00119800
C** THE FOLLOWING SUBROUTINES ARE CALLED BY THIS ROUTINE:
     (1) SUBROUTINE PD TO CALCULATE PROBABILITY OF DETECTEING.
                                                                   * 00119900
C**
C**
                                                                   * 00120000
      (2) SUBROULINE ALFA TO CALCULATE PROB. OF FALSE ALARM.
C**
                                                                   * 00120100
C**
                                                                     00120300
C*
                                                                     00120400
      IMPLICIT REAL*8(A-H,O-Z)
                                                                     00120500
     REAL*8 LAMBDA
                                                                     00120600
     COMMON / MAIN1 /LUR,LUW
COMMON / DUNC1 / LAMBDA
COMMON / DCDY1 / DELTA, B,C,DD,E,VZMV1,T,W
                                                                     00120700
                                                                     00120800
                                                                     00120900
        INTEGER N
                                                                     00121000
        REAL*4 X(N)
                                                                     00121100
        REAL*4 F
                                                                     00121200
                                                                     00121300
 123
      CONTINUE
                                                                     00121400
C**FIRST MAKE THE SINGLE PRECISION VALUES OF X(1), X(2), AND X(3)
                                                                     00121530
C**DOUBLE PRECISION (FOR MORE ACCURATE CALCULATION) BY ASSIGNING C**THEM TO DX1, DX2, AND DX3. THIS IS DONE BECAUSE THE IMSL ROUTINE
                                                                     00121600
                                                                     00121700
C**IS SINGLE PRECISION
                                                                     00121800
                                                                     00121900
       DX1=X(1)
       DX2=X(2)
                                                                     00122000
       DX3=X(3)
                                                                     00122100
                                                                     00122200
C++CHECK IF ANY OF THE DECISION VARIABLES ARE OUT OF RANGE
                                                                     00122300
C++THEN RETURN WITH A BIG VALUE FOR F
                                                                     00122400
C++
                                                                     00122500
        IF(DX1.GT.70..OR.DX2.GT.12.)F=100000000.
                                                                     00122600
        IF(DX1.GT.70..OR.DX2.GT.12.)RETURN
                                                                     00122700
        IF(DX1.LT.0..OR.DX2.LT.0.)F=100000000.
                                                                     00122800
        IF(DX1.LT.0..OR.DX2.LT.0.)RETURN
                                                                     00122900
        IF(DX3.LT.1.)F=100000000.
                                                                     00123000
        IF(DX3.LT.1.)RETURN
                                                                     00123100
C**
                                                                     00123200
C**CALCULATE THE AVG. TIME OF OCCURANCE W/IN THE NH TO (N+1)H TIME INTER00123300
C** INTERVAL.
                                                                     00123400
C**
                                                                     00123500
           DXPH=DEXP(-LAMBDA*DX1)
                                                                     00123600
      ATWIN=(1.D0-(1.D0+LAMBDA*DX1)*DXPH)/(LAMBDA*(1.D0-DXPH))
                                                                     00123700
C**
                                                                     00123800
C**CALCULATE PD; PROBABILITY OF DETECTING THE SHIFT.
                                                                     00123900
C**
                                                                     00124000
      CALL PROBD(DX2,DX3,DELTA,PD)
                                                                     00124100
                                                                     00124200
C**CALCULATE APP. APP IS AVG. TIME OUT OF CONTROL BEFORE A SAMPLE
                                                                     00124300
C**FALLS OUTSIDE THE CONTROL LIMIT ( EXCLUDING ATWIN.)
                                                                     00124400
C**
                                                                     00124500
      APP=DX1/PD-ATWIN+E*DX3
                                                                     00124600
C**
                                                                     00124700
```

```
C**CALCULATE CYCLE LENGTH
                                                                00124800
C**
                                                                00124900
     CYCLE=1.D0/LAMBDA+APP+DD
                                                                00125000
C**
                                                                00125100
C**CALCULATE ALPHA ; PROB. OF FALSE ALARM
                                                                00125200
C**
                                                                00125300
     CALL PROFA(DX2, ALPHA)
                                                                00125400
C**
                                                                00125500
C**CALCULATE POOC ; PROPORTION OF TIME OUT OF CONTROL
                                                                00125600
                                                                00125700
      PINC=1.D0/(LAMBDA*CYCLE)
                                                                00125800
      POOC=1.D0-PINC
                                                                00125900
C**
                                                                00126000
C**CALCULATE ENFALSE; EXPECTED NUMBER OF FALSE ALARMS
                                                                00126100
C**
                                                                00126200
      ENFALS=ALPHA*DXPH/(1.D0-DXPH)
                                                                00126300
C**
                                                                00126400
C**CALCULATE THE LOSS-COST PER HOUR OF OPERATION.
                                                                00126500
                                                                00126600
      BAHMC=(B+C*DX3)/DX1
                                                                00126700
      RLOSSC=POOC*VZMV1+T*ENFALS/CYCLE+W/CYCLE+BAHMC
                                                                00126800
      F=RLOSSC
                                                                00126900
C
                                                                00127000
      RETURN
                                                                00127100
      END
                                                                00127200
С
                                                                00127300
                                                                00127400
SUBROUTINE FUNCT2(N,X,F)
                                                                00127800
C**
                                                              * 00128000
C** THIS SUBROUTINE IS EQUIVALENT OF FUNCT AS IS NEEDE FOR
                                                              * 00128100
                                                              * 00128200
C** TWO-DIMENSIONAL SEARCH.
                                                              * 00128300
C**
                                                              * 00128400
C** THE FOLLOWING SUBROUTINES ARE CALLED BY THIS SUBROUTINE:
C**
                                                              * 00128500
    (1) SUBROUTINE PD TO CALCULATE PROBABILITY OF DETECTEING.
C**
       (2) SUBROUTINE ALFA TO CALCULATE PROB. OF FALSE ALARM.
                                                              * 00128600
C**
                                                              * 00128700
C**
                                                                00128900
C*
                                                                00129000
     IMPLICIT REAL*8(A-H,O-Z)
                                                                00129100
     REAL*8 LAMBDA
                                                                00129200
     COMMON / MAIN1 /LUR,LUW
COMMON / DUNC1 / LAMBDA
COMMON / DCDY1 / DELTA, B,C,DD,E,VZMV1,T,W
COMMON / SAMPLS /IX
                                                                00129300
                                                                00129400
                                                                00129500
                                                                00129600
        INTEGER N
                                                                00129700
        REAL*4 X(N)
                                                                00129800
        REAL*4 F
                                                                00129900
C**
                                                                00130000
C**FIRST MAKE THE SINGLE PRECISION VALUES OF X(1), X(2), AND X(3)
                                                                00130100
C**DOUBLE PRECISION (FOR MORE ACCURATE CALCULATION) BY ASSIGNING C**THEM TO DX1, DX2, AND DX3. THIS IS DONE BECAUSE THE IMSL ROUTINE
                                                                00130200
                                                                00130300
C**IS SINGLE PRECISION
                                                                00130400
C**
                                                                00130500
       DX1=X(1)
                                                                00130600
       DX2=X(2)
                                                                00130700
       DX3=IX
                                                                00130800
                                                                00130900
C++CHECK IF ANY OF THE DECISION VARIABLES ARE OUT OF RANGE
                                                                00131000
C++THEN RETURN WITH A BIG VALUE FOR F
                                                                00131100
                                                                00131200
       IF(DX1.GT.70..OR.DX2.GT.12.)F=100000000.
                                                                00131300
```

```
IF(DX1.GT.70..OR.DX2.GT.12.)RETURN
                                                              00131400
      IF(DX1.LT.0..OR.DX2.LT.0.)F=100000000.
                                                               00131500
                                                               00131600
      IF(DX1.LT.0..OR.DX2.LT.0.)RETURN
      IF(DX3.LT.1.)F=100000000.
                                                               00131700
      IF(DX3.LT.1.)RETURN
                                                               00131800
C**
                                                               00131900
C**CALCULATE THE AVG. TIME OF OCCURANCE W/IN THE NH TO (N+1)H TIME INTER00132000
C** INTERVAL.
                                                               00132100
C**
                                                               00132200
         DXPH=DEXP(-LAMBDA*DX1)
                                                               00132300
     ATWIN=(1.D0-(1.D0+LAMBDA*DX1)*DXPH)/(LAMBDA*(1.D0-DXPH))
                                                               00132400
C**
                                                               00132500
C**CALCULATE PD; PROBABILITY OF DETECTING THE SHIFT
                                                               00132600
C**
                                                               00132700
                                                               00132800
     CALL PROBD(DX2,DX3,DELTA,PD)
C**
                                                               00132900
C**CALCULATE APP. APP IS AVG. TIME OUT OF CONTROL BEFORE A SAMPLE C**FALLS OUTSIDE THE CONTROL LIMIT ( EXCLUDING ATWIN.)
                                                               00133000
                                                               00133100
                                                               00133200
     APP=DX1/PD-ATWIN+E*DX3
                                                               00133300
C**
                                                               00133400
C**CALCULATE CYCLE LENGTH
                                                               00133500
C**
                                                               00133600
                                                               00133700
     CYCLE=1.D0/LAMBDA+APP+DD
                                                               00133800
C**CALCULATE ALPHA ; PROB. OF FALSE ALARM
                                                               00133900
C**
                                                               00134000
     CALL PROFA(DX2, ALPHA)
                                                               00134100
C**
                                                               00134200
C**CALCULATE POOC ; PROPORTION OF TIME OUT OF CONTROL
                                                               00134300
C**
                                                               00134400
      PINC=1.D0/(LAMBDA*CYCLE)
                                                               00134500
      POOC=1.DO-PINC
                                                               00134600
C**
                                                               00134700
C**CALCULATE ENFALSE ; EXPECTED NUMBER OF FALSE ALARMS
                                                               00134800
C**
                                                               00134900
      ENFALS=ALPHA*DXPH/(1.D0-DXPH)
                                                               00135000
C**
                                                               00135100
C**CALCULATE THE LOSS-COST PER HOUR OF OPERATION.
                                                               00135200
C**
                                                               00135300
      BAHMC = (B+C*DX3)/DX1
                                                               00135400
      RLOSSC=POOC*VZMV1+T*ENFALS/CYCLE+W/CYCLE+BAHMC
                                                               00135500
      F=RLOSSC
                                                               00135600
С
                                                               00135700
      RETURN
                                                               00135800
      END
                                                               00135900
C
                                                               00136000
                                                               00136100
C
С
                                                               00136200
C
                                                               00136300
SUBROUTINE DUNEVA
                                                               00136700
C**
                                                             * 00136900
C** THIS SUBROUTINE IS USED TO EVALUATE DUNCAN'S COST FUNCTION FOR THE* 00137000
C** GIVEN VALUES OF N, H, AND K.
                                                             * 00137100
C**
                                                             * 00137200
                                                             * 00137300
C** THE FOLLOWING SUBROUTINES ARE CALLED BY THIS SUBROUTINE:
C**
                                                             * 00137400
      (1) SUBROUTINE PD TO CALCULATE PROBABILITY OF DETECTEING.
C**
      (2) SUBROUTINE ALFA TO CALCULATE PROB. OF FALSE ALARM.
                                                             * 00137500
C**
                                                             * 00137600
C**
                                                               00137800
C*
                                                               00137900
```

```
IMPLICIT REAL*8(A-H,O-Z)
                                                                 00138000
     REAL*8 LAMBDA
                                                                 00138100
     COMMON / MAIN1 /LUR, LUW
                                                                 00138200
     COMMON / DUNC1 / LAMBDA
COMMON / DUNC4 / N,H,RK
COMMON / DUNC7 / DCLCST
                                                                 00138300
                                                                 00138400
                                                                 00138500
     COMMON / DCDY1 / DELTA, B,C,DD,E,VZMV1,T,W
                                                                 00138600
C
                                                                 00138700
C**
                                                                 00138800
C**DX1, DX2, DX3 CORRESPOND TO H, K, AND N, RESPECTIVELY.
                                                                 00138900
C**
                                                                 00139000
       DX1=H
                                                                 00139100
       DX2=RK
                                                                 00139200
                                                                 00139300
       DX3=N
C**
                                                                 00139400
C**CALCULATE THE AVG. TIME OF OCCURANCE W/IN THE NH TO (N+1)H TIME INTER00139500
C** INTERVAL.
                                                                 00139600
C**
                                                                 00139700
          DXPH=DEXP(-LAMBDA*DX1)
                                                                 00139800
     ATWIN=(1.D0-(1.D0+LAMBDA*DX1)*DXPH)/(LAMBDA*(1.D0-DXPH))
                                                                 00139900
C**
                                                                 00140000
C**CALCULATE PD; PROBABILITY OF DETECTING THE SHIFT
                                                                 00140100
Č**
                                                                 00140200
     CALL PROBD(DX2,DX3,DELTA,PD)
                                                                 00140300
                                                                 00140400
C**CALCULATE APP. APP IS AVG. TIME OUT OF CONTROL BEFORE A SAMPLE
                                                                 00140500
C**FALLS OUTSIDE THE CONTROL LIMIT ( EXCLUDING ATWIN.)
                                                                 00140600
C**
                                                                 00140700
     APP=DX1/PD-ATWIN+E*DX3
                                                                 00140800
C**
                                                                 00140900
C**CALCULATE CYCLE LENGTH
                                                                 00141000
C**
                                                                 00141100
                                                                 00141200
     CYCLE=1.D0/LAMBDA+APP+DD
C**
                                                                 00141300
C**CALCULATE ALPHA ; PROB. OF FALSE ALARM
                                                                 00141400
C**
                                                                 00141500
     CALL PROFA (DX2, ALPHA)
                                                                 00141600
C**
                                                                 00141700
C**CALCULATE POOC ; PROPORTION OF TIME OUT OF CONTROL
                                                                 00141800
C**
                                                                 00141900
      PINC=1.D0/(LAMBDA*CYCLE)
                                                                 00142000
      POOC=1.D0-PINC
                                                                 00142100
C**
                                                                 00142200
C**CALCULATE ENFALSE ; EXPECTED NUMBER OF FALSE ALARMS
                                                                 00142300
C**
                                                                 00142400
      ENFALS=ALPHA*DXPH/(1.D0-DXPH)
                                                                 00142500
C**
                                                                 00142600
C**CALCULATE THE LOSS-COST PER HOUR OF OPERATION.
                                                                 00142700
C**
                                                                 00142800
      BAHMC = (B+C*DX3)/DX1
                                                                 00142900
      RLOSSC=POOC*VZMV1+T*ENFALS/CYCLE+W/CYCLE+BAHMC
                                                                 00143000
      DCLCST=100.D0*RLOSSC
                                                                 00143100
С
                                                                 00143200
      RETURN
                                                                 00143300
      END
                                                                 00143400
C
                                                                 00143500
SUBROUTINE PROBD(RK,RN,DELTA,PD)
                                                                 00143800
C*
                                                               * 00144000
C** THIS SUBROUTINE CALCULATES PD ; PROB. OF DETECTING THE SHIFT.
                                                               * 00144100
C*
                                                               * 00144200
C
                                                                 00144400
     IMPLICIT REAL*8(A-H,O-Z)
                                                                 00144500
```

```
00144600
C
     Y=RK-DELTA*DSQRT(RN)
                                                              00144700
     CALL MDNORD(Y,P)
                                                              00144800
C*
                                                              00144900
C*PD IS PROB. OF DETECTING THE SHIFT. ( IT IS ASSUMED THAT THE SHIFT
                                                              00145000
C*
   IS POSITIVE.)
                                                              00145100
C*
                                                              00145200
     PD=1.D0-P
                                                              00145300
C
                                                              00145400
     RETURN
                                                              00145500
                                                              00145600
     END
С
                                                              00145700
                                                              00145800
C
C****
                                                              00145900
C*****************
                                                              00146000
     SUBROUTINE PROFA(RK, ALPHA)
                                                              00146100
C********************
                                                              00146200
C*
                                                              00146300
C** THIS SUBROUTINE CALACULATES ALPHA; PROB. OF FALSE ALARM.
                                                              00146400
C:#
                                                              00146500
<u>C</u>*********************
                                                              00146600
C
                                                              00146700
     IMPLICIT REAL*8(A-H,O-Z)
                                                              00146800
C
                                                              00146900
     CALL MDNORD(RK,P)
                                                              00147000
C**
                                                              00147100
C**ALPHA IS PROB. OF FALSE ALARM.
                                                              00147200
.C**
                                                              00147300
     ALPHA=2.D0*(1.D0-P)
                                                              00147400
C
                                                              00147500
     RETURN
                                                              00147600
     END
                                                              00147700
C
                                                              00147800
C
                                                              00147900
                                                              00148000
SUBROUTINE DYNOPT
                                                              00148400
C**
                                                            * 00148600
C** THIS SUBROUTINE OPTIMIZES THE DYNAMIC LOSS-COST MODEL.
                                                            * 00148700
C**
                                                            * 00148800
C** THE FOLLOWING SUBROUTINES ARE CALLED BY THIS SUBROUTINE:
                                                            * 00148900
C**
      (1) SUBROUTINE TWOSCH TO PERFORM A 2-DIMENSIONAL SEARCH.
                                                            * 00149000
C**
                                                            * 00149100
      (2) SUBROUTINE SINTW TO CALCULATE ISTEPS.
C**
                                                            * 00149200
C**
                                                              00149400
C*
                                                              00149500
C**
                                                              00149600
     IMPLICIT REAL*8(A-H,O-Z)
                                                              00149700
     REAL*8 NF, IH, HF, IK, KF
                                                              00149800
     COMMON / MAIN1 /LUR, LUW
                                                              00149900
     COMMON / DYNM2 / ISTEPS
COMMON / DYNM3 / IN, NF,IH,HF,IK,KF
COMMON / DYNM4 / PROBPT
                                                              00150000
                                                              00150100
                                                              00150200
     COMMON / DYNM5 / ITRMX1,ITRMX2,ITRMX3

COMMON / DYNM6 / DEL(6),DELMN(6),DELMX(6),XQLIM(6)

COMMON / DYOPT1 / XXX(6),YYYF

COMMON / DYOPT2 / ISIDEL(6), NFUEVA,NFTERM
                                                              00150300
                                                              00150400
                                                              00150500
                                                              00150600
     COMMON / DYOPT3 / TMPMIN(10)
COMMON / DYOPT4 / NWOPT, HWOPT, RKWOPT, FHWOPT, FKWOPT, YFWOPT
                                                              00150700
                                                              00150800
     DATA TMPMIN/10*999999.D0/
                                                              00150900
C**
                                                              00151000
C**
                                                              00151100
```

```
C**NOTE THAT VARIABLES: XXX(1), XXX(2), XXX(3), XXX(4), XXX(5), XXX(6) 00151200
C**CORRESPOND TO:
                                                                           00151300
                             , ISTEPS, IK , KF , IN , NF ,
                         HF
C**RESPECTIVELY.
                                                                           00151400
C**
                                                                           00151500
C
                                                                           00151600
C**
                                                                            00151700
C**INITIALIZE ALL THE VARIABLES TO THEIR STARTING POINT SET IN ROUTINE
                                                                           00151800
C**DYNM.
                                                                           00151900
C**
                                                                            00152000
      XXX(1)=HF
                                                                            00152100
      XXX(2)=ISTEPS
                                                                            00152200
                                                                           00152300
      XXX(3)=IK
                                                                            00152400
      XXX(4) = KF
                                                                           00152500
      XXX(5) = IN
                                                                            00152600
      XXX(6)=NF
C**
                                                                           00152700
C**INITIALIZE THE SIGN (DIRECTION) OF SEARCH FOR EACH VARIABLE
                                                                            00152800
C**
                                                                           00152900
                                                                           00153000
      DO 25 I=1,6
         ISIDEL(I)=+1
                                                                           00153100
   25 CONTINUE
                                                                           00153200
C**
                                                                           00153300
C**OPTIMIZE OVER THE FIRST TWO VARIABLES. THAT IS, HF AND ISTEPS.
                                                                           00153400
C**
                                                                           00153500
      IVARL=2
                                                                            00153600
      CALL TWOSCH(IVARL)
                                                                            00153700
C*.
                                                                            00153800
C*.SET VARIABLES TO THEIR BEST OPTIMUMFOUND SO FAR. (THIS IS HELPFUL
                                                                            00153900
C*.ESPECIALLY IF ITERMAX1 IS REACHED.)
                                                                            00154000
                                                                            00154100
                                                                            00154200
      DO 35 I=1,6
         XXX(I) = TMPMIN(I)
                                                                            00154300
   35 CONTINUE
                                                                            00154400
      YYYF=TMPMIN(10)
                                                                            00154500
C**
                                                                            00154600
C**OPTIMIZE OVER THE VARIABLES KF AND IK
                                                                            00154700
C**
                                                                            00154800
      IVARL=4
                                                                            00154900
      CALL TWOSCH(IVARL)
                                                                            00155000
C*.
                                                                            00155100
C*.SET VARIABLES TO THEIR BEST OPTIMUMFOUND SO FAR. (THIS IS HELPFUL C*.ESPECIALLY IF ITERMAX2 IS REACHED.)
                                                                            00155200
                                                                            00155300
                                                                            00155400
      DO 40 I=1,6
                                                                            00155500
         XXX(I)=TMPMIN(I)
                                                                            00155600
   40 CONTINUE
                                                                            00155700
      YYYF=TMPMIN(10)
                                                                            00155800
C**
                                                                            00155900
C**OPTIMIZE OVER THE VARIABLES NF AND IN
                                                                            00156000
C**
                                                                            00156100
      IVARL=6
                                                                            00156200
      CALL TWOSCH(IVARL)
                                                                            00156300
C*.
                                                                            00156400
C*.SET VARIABLES TO THEIR BEST OPTIMUMFOUND SO FAR. (THIS IS HELPFUL
                                                                            00156500
C*.ESPECIALLY IF ITERMAX3 IS REACHED.)
                                                                            00156600
                                                                            00156700
      DO 45 I=1,6
                                                                            00156800
         XXX(I)=TMPMIN(I)
                                                                            00156900
   45 CONTINUE
                                                                            00157000
      YYYF=TMPMIN(10)
                                                                            00157100
                                                                            00157200
C**NOTE THAT WHEN RETURNING FROM THIS ROUTINE XXX,S AND YYYF AS WELL
                                                                            00157300
C**AS THE ARRAY TMPMIN CONTAIN THE OPTIMAL DYNAMIC DESIGN. ALSO,
                                                                            00157400
C**KEEP THE OPTIMUM DESIGN IN THE FOLLOWING:
                                                                            00157500
C**
                    NWOPT, HWOPT, RKWOPT, FNWOPT, FHWOPT, FKWOPT, YFWOPT
                                                                            00157600
C** CORRESPONDING TO: IN
                             ΙH
                                   ΙK
                                         NF
                                                HF
                                                        KF
                                                                            00157700
```

```
C**
                                                             00157800
                                                             00157900
     FHWOPT=TMPMIN(1)
    HWOPT=SINTW(FHWOPT, ISTEPS)
                                                             00158000
     RKWOPT=TMPMIN(3)
                                                             00158100
     FKWOPT=TMPMIN(4)
                                                             00158200
     NWOPT=TMPMIN(5)
                                                             00158300
     FNWOPT=TMPMIN(6)
                                                             00158400
     YFWOPT=100.D0*TMPMIN(10)
                                                             00158500
C
                                                             00158600
     RETURN
                                                             00158700
     END 1
                                                             00158800
C
                                                             00158900
                                                             00159000
C
C
                                                             00159100
                                                             00159200
C
                                                             00159300
00159700
C
                                                             00159800
                                                             00159900
     SUBROUTINE TWOSCH(IVARL)
C**
                                                           * 00160100
                                                           * 00160200
C** THIS SUBROUTINE OPTIMIZES PERFORMS A TWO-DIMENSIONAL SEARCH.
C**
                                                           * 00160300
                                                           * 00160400
C** THE FOLLOWING SUBROUTINES ARE CALLED BY THIS SUBROUTINE:
C**
      (1) SUBROUTINE COGGIN TO PERFORMA PRECISE LINE SEARCH USING
                                                           * 00160500
C**
                                                           * 00160600
         COGGINS' METHOD.
C**
      (2) SUBROUTINE OMYSCH WHICH IS USED IN CONJUNCTON WITH THIS
                                                           * 00160700
C**
         ROUTINE TO PERFORM A TWO-AT-A-TIME SEARCH.
                                                           * 00160800
C**
      (3) SUBROUTINE RUNC WHICH CALCULATES THE LOSS-COST OF THE
                                                           * 00160900
C**
                                                           * 00161000
         DYNAMIC MODEL.
C**
                                                           * 00161100
C**
                                                           * 00161200
C**
                                                             00161400
С
                                                             00161500
                                                             00161600
                                                          ** 00161700
C**NOTE THE FOLLOWING VARIABLES DEFINITION USED IN THIS ROUTINE.
                                                             00161800
                                                             00161900
č
   XQLIM(I): QUITTING LIMIT OR DESIRED ACCURRACY FOR
                                                             00162000
             VARIABLE I.
                                                             00162100
CC
                                                             00162200
   DELMN(I): MIN. LIMIT ON STEP SIZE, FOR VARIABLE I.
                                                             00162300
C
                                                             00162400
   DELMX(I): MAX. LIMIT ON STEP SIZE , FOR VARIABLE I.
                                                             00162500
                                                             00162600
Č
   DEL(I) : CURRENT STEP SIZE , FOR VARIABLE I.
                                                             00162700
                                                             00162800
č
   ISIDEL(I): SIGN OF DEL FOR THE LAST SUCCESSFUL MOVE ALONG
                                                             00162900
CCC
             VARIABLE I.
                                                             00163000
                                                             00163100
   XXX'S
           : DECISION VARIABLES; HF, ISTEPS, IK, KF, IN, NF
                                                             00163200
Č
             ITS FINAL VALUE CONTAINS THE LAST POINT TRIED. THUS,
                                                             00163300
C
             IT MIGHT NOT BE THE OPTIMUM POINT.
                                                             00163400
Č
                                                             00163500
C
           : INITIAL STEP SIZE . THIS IS CALCULATE IN ROUTINE RUNC
                                                             00163600
Ċ
            FOR ANY GIVEN VALUES OF HF AND ISTEPS.
                                                             00163700
С
                                                             00163800
C
   NFTERM : THE MAX. NUMBER OF OBJ. FUNCTION EVALUATIONS ALLOWED.
                                                             00163900
Č
                                                             00164000
C
                                                             00164100
C**
                                                             00164200
                                                             00164300
```

```
C
                                                                             00164400
      IMPLICIT REAL*8(A-H,O-Z)
COMMON / MAIN1 /LUR,LUW
COMMON / DYNM5 / ITRMX1,ITRMX2,ITRMX3
                                                                             00164500
                                                                             00164600
                                                                             00164700
      COMMON / DYNM6 / DEL(6), DELMN(6), DELMX(6), XQLIM(6)
                                                                             00164800
      COMMON / DYOPT1 / XXX(6), YYYF
COMMON / DYOPT2 / ISIDEL(6), NFUEVA, NFTERM
                                                                             00164900
                                                                             00165000
      DIMENSION TEMP(10)
                                                                             00165100
      DIMENSION ITRMAX(3)
                                                                             00165200
C**
                                                                             00165300
      IFLAG=1
                                                                             00165400
      IDIRC=IVARL
                                                                             00165500
C**
                                                                             00165600
C**NFTERM IS MAX. NUMBER OF OBJ. FUN. EVALUATIONS ALLOWED.
                                                                             00165700
C**
                                                                             00165800
                                                                             00165900
C**NFUEVA: NUMBER OF OBJ. FUN. EVALUATED SO FAR IS 0.
C**
                                                                             00166000
      ITRMAX(1)=ITRMX1
                                                                             00166100
      ITRMAX(2)=ITRMX2
                                                                             00166200
      ITRMAX(3)=ITRMX3
                                                                             00166300
      IFTR=IVARL/2
                                                                             00166400
      NFTERM=ITRMAX(IFTR)
                                                                             00166500
      NFUEVA=0
                                                                             00166600
C**
                                                                             00166700
      IF(IVARL.NE.2)GO TO 88
                                                                             00166800
C**
                                                                             00166900
C**FOR THE FIRST VARIABLE ( HF) USE COGGIN TO DO A LINE SEARCH
                                                                             00167000
C**
                                                                             00167100
                                                                             00167200
      IDIRC=1
      CALL COGGIN(IDIRC)
                                                                             00167300
                                                                             00167400
C*. CHECK DEL AFTER COGGIN, S EXECUTION
                                                                             00167500
                                                                             00167600
C*.IF THE LAST DEL USED IN COGGIN IS LESS THAN DELMN ,THEN USE DELMN BUT00167700
C*.RESERVE THE SIGN OF DEL ( THE LAST SUCCESFUL STEP TAKEN IN COGGIN)
                                                                             00167800
C*.FOR THE NEXT STEP TAKEN ALONG THAT VARIABLE
                                                                             00167900
                                                                             00168000
      IF(DABS(DEL(IDIRC)).LT.DELMN(IDIRC))DEL(IDIRC)=DSIGN(DELMN(IDIRC),00168100
                                                               DEL(IDIRC))
                                                                             00168200
C*.
                                                                             00168300
                                                                             00168400
      IDIRC=2
C**
                                                                             00158500
C**NOTE THAT FOR IVARL=4 AND 6, THE PREVIOUS OPTIMUM POINT IS USED AS
                                                                             00168600
C**THE STARTING POINT. VARIABLES XXX,S AND YYYF CONTAIN THIS INFORMATION00168700
C**AND ARE SET IN SUBROUTINE DYNOPT.
                                                                             00168800
C**
                                                                             00168900
С
                                                                             00169000
С
                                                                             00169100
C**
                                                                             00169200
C**CALL SUBROUTINE OMYSCH TO MOVE ALONG THE VARIABLE IDIRC.
                                                                             00169300
C**
                                                                             00169400
 88
      CALL OMYSCH(IDIRC, IFLAG)
                                                                             00169500
С
                                                                             00169600
C
                                                                             00169700
C**
                                                                             00169800
C**NOW CALL SUBROUTINE OMYSCH TO MOVE ALONG THE OTHER VARIABLE, IF
                                                                             00169900
C** IFLAG IS NOT 3 AND NFTERM IS NOT REACHED.
                                                                             00170000
                                                                             00170100
       IDIRC=IDIRC+1
                                                                             00170200
      IF(IDIRC.GT.IVARL)IDIRC=IVARL-1
                                                                             00170300
      IF(IFLAG.LT.3 .AND. NFUEVA.LT.NFTERM) GO TO 88
                                                                             00170400
C**
                                                                             00170500
C**IF IFLAG=3, THEN SEARCH HAS FAILED ALONG BOTH DIRECTIONS .
                                                                             00170600
C**TRY SOME POINTS FURTHER AHEAD TO SEE IF ANY IMPROVEMENTS IS POSSIBLE 00170700
C**IF MAX. OF OBJ. FUN. EVALUATIONS IS REACHED THEN QUIT.
                                                                             00170800
C**
                                                                             00170900
```

```
IF(NFUEVA.GE.NFTERM) GO TO 313
                                                                          00171000
C**
                                                                          00171100
      WRITE(6,249)
                                                                          00171200
249 FORMAT(1X,' IFLAG=3; SEARCH FAILED AT BOTH DIRECTIONS.<<')
                                                                          00171300
C**
                                                                          00171400
C**KEEP THE CURRENT POINT IN TEMP ( NOTE THAT THIS MIGHT NOT BE THE
                                                                          00171500
C**BEST POINT FOUND SO FAR.)
                                                                          00171600
C**
                                                                          00171700
      DO 260 I=1,6
                                                                          00171800
         TEMP(I)=XXX(I)
                                                                          00171900
260
     CONTINUE
                                                                          00172000
      TEMP(10)=YYYF
                                                                          00172100
                                                                          00172200
C**MOVE TO A NEW POINT. THIS STRATEGY TO SOME DEGREE GUARDS AGAINST
                                                                          00172300
C**THE OCCURANCE OF SMALL BUMPS IN THE OBJECTIVE FUNCTION, S LANDSCAPE.
                                                                          00172400
                                                                          00172500
      ISTRT=IVARL-1
                                                                          00172600
      DO 263 I=ISTRT, IVARL
                                                                          00172700
         XXX(I) = XXX(I) + ISIDEL(I) * DELMN(I)
                                                                          00172800
263 CONTINUE
                                                                          00172900
C*.
                                                                          00173000
C*. EVALUATE OBJ. FUNCTION FOR THIS NEW POINT
                                                                          00173100
C*.
                                                                          00173200
      CALL RUNC
                                                                          00173300
C*.
                                                                          00173400
C*.IF THE NEW POINT IS BETTER (SUCCESS), THEN SET IFLAG=1, IDIRC=IVARL-100173500
C*.AND RETURN TO SEARCH.
                                                                          00173600
C*.
                                                                          00173700
      IF(YYYF.GT.TEMP(10)) GO TO 270
                                                                          00173800
      IFLAG=1
                                                                          00173900
      IDIRC=IVARL-1
                                                                          00174000
      GO TO 88
                                                                          00174100
C*..
                                                                          00174200
C*..IF THE NEW POINT IS WORSE , STILL TRY ANOTHER POINT
                                                                          00174300
Č*..
                                                                          00174400
 270 XXX(IVARL)=XXX(IVARL)+ISIDEL(IVARL)*DELMN(IVARL)
                                                                          00174500
      CALL RUNC
                                                                          00174600
C*..
                                                                          00174700
C*..IF THIS SECOND POINT IS BETTER (SUCCESS) , THEN SET IFLAG=1 ,
                                                                          00174800
C*..IDIRC=IVARL-1, AND RETURN TO SEARCH.
                                                                          00174900
                                                                          00175000
      IF(YYYF.GT.TEMP(10)) GO TO 300
                                                                          00175100
      IFLAG=1
                                                                          00175200
      IDIRC=IVARL-1
                                                                          00175300
      GO TO 88
                                                                          00175400
C*..IF THE SECOND POINT IS WORSE, THEN QUIT THE SEARCH
                                                                          00175500
C*..
                                                                          00175600
                                                                          00175700
C**COPY BACK THE MIN. POINT INFORMATION FROM TEMP INTO XXX
                                                                          00175800
C**NOTE THAT THE REAL MIN. IS IN ARRAY TMPMIN IN ROUTINE RUNC.
                                                                          00175900
C**
                                                                          00176000
 300 DO 305 I=1,6
                                                                          00176100
         XXX(I) = TEMP(I)
                                                                          00176200
 305 CONTINUE
                                                                          00176300
      YYYF=TEMP(10)
                                                                          00176400
C**
                                                                          00176500
C**PRINT THE INFORMATION AND STOP
                                                                          00176600
C**
                                                                          00176700
313
      IF (NFUEVA.EQ.NFTERM) WRITE (LUW, 314) NFTERM
                                                                          00176800
      FORMAT(1X, 'MAX. NUMBER OF FUNCTION EVALUATIONS REACHED;',15)
 314
                                                                          00176900
C
                                                                          00177000
      RETURN
                                                                          00177100
      END
                                                                          00177200
                                                                          00177300
C
                                                                          00177400
                                                                          00177500
```

```
С
                                                                  00177600
C
                                                                  00177700
SUBROUTINE COGGIN(IDIRC)
                                                                  00178100
C**
                                                                * 00178300
C** THIS ROUTINE USES COGGINS' TECHNIQUE TO DO A PRECISE LINE SEARCH * 00178400
                                                                * 00178500
C** ALONG THE VARIABLE SPECIFIED BY IDIRC.
C**
                                                                * 00178600
C** THIS ROUTIE CALLS THE FOLLOWING SUBROUTINES:
C** (1) SUBROUTINE RUNC TO CALCULATE THE LOSS-COST FUNCTION OF
                                                                * 00178700
                                                                 * 00178800
C**
          THE DYNAMIC MODEL.
                                                                * 00178900
C**
                                                                 * 00179000
C**
                                                                 * 00179100
C**
                                                                  00179300
C*
                                                                  00179400
     IMPLICIT REAL*8(A-H,O-Z)
                                                                  00179500
     COMMON / MAIN1 /LUR, LUW
COMMON / DYNM6 / DEL(6), DELMN(6), DELMX(6), XQLIM(6)
COMMON / DYOPT1 / XXX(6), YYYF
COMMON / DYOPT2 / ISIDEL(6), NFUEVA, NFTERM
COMMON / DYOPT3 / TMPMIN(10)
                                                                  00179600
                                                                  00179700
                                                                  00179800
                                                                   00179900
                                                                   00180000
     DIMENSION ZZ(3), WW(3), YF(3)
                                                                   00180100
                                                                   00180200
                                                                  00180300
  SET VALUES OF LIMIT, STEP SIZE, AND INITIAL GUESS FOR X
                                                                   00180400
                                                                  00180500
                                                                   00180600
      XLIM=XOLIM(IDIRC)
                                                                   00180700
      DELX=DEL(IDIRC)
                                                                   00180800
      X1=XXX(IDIRC)
                                                                   00180900
С
                                                                   00181000
С
                                                                   00181100
    DAVIS , SWANN, AND CAMPEY ALGORITHM
                                                                   00181200
С
                                                                   00181300
    EVALUATE THE FUNCTION FOR THE INITIAL VALUE OF THE INDEPENDENT
C
                                                                   00181400
C
    VARIABLE.
                                                                   00181500
                                                                   00181600
     XXX(IDIRC)=X1
                                                                   00181700
     CALL RUNC
                                                                   00181800
     Y1=YYYF
                                                                   00181900
CC
                                                                   00182000
     WRITE(LUW, 210)
                                                                   00182100
 210 FORMAT(1X,//,15X,'D.S.C ALGORITHM')
                                                                  00182200
     WRITE(LUW, 204)
                                                                  00182300
 204 FORMAT(1X,//,1X,21X,'XX',14X,'YY',8X,'STEP SIZE')
WRITE(LUW,205)X1,Y1,DELX
                                                                  00182400
                                                                  00182500
 205 FORMAT(1X,//,1X,15X,E14.7,2X,E14.7,2X,E14.7)
                                                                  00182600
С
                                                                   00182700
С
   INCREMENT THE INDEPENDENT VARIABLE AND EVALUATE THE FUNCTION
                                                                   00182800
С
                                                                   00182900
      X2=X1+DELX
                                                                   00183000
      II = 0
                                                                   00183100
C
                                                                   00183200
     XXX(IDIRC)=X2
                                                                   00183300
      CALL RUNC
                                                                   00183400
      Y2=YYYF
                                                                   00183500
      WRITE(LUW, 205) X2, Y2, DELX
                                                                   00183600
C
                                                                   00183700
                                                                   00183800
C+ SEE WHICH OF THE FUNCTION EVALUATIONS IS THE SMALLEST
                                                                   00183900
C
 9
      IF(Y1-Y2)10,12,12
                                                                   00184000
C+ STMT 10 IS WHEN FAILED
                                                                  00184100
```

```
10
      IF(II-1)14,14,16
                                                                           00184200
14
      II=II+1
                                                                           00184300
C
                                                                           00184400
C+ IF THE FUNCTION EVALUATION IS MORE AFTER THE INDEPENDENT
                                                                           00184500
  VARIABLE HAS BEEN INCREMENTED, SWITCH DIRECTION
                                                                           00184600
C
                                                                           00184700
      DELX=-DELX
                                                                           00184800
      X2=X1+DELX
                                                                           00184900
      GO TO 8
                                                                           00185000
C
                                                                           00185100
  IF THE MAX. IS BRACKETED, MOVE TO THE POWEL ALGOITHM.
                                                                           00185200
C
                                                                           00185300
16
      GO TO 80
                                                                           00185400
12
      CONTINUE
                                                                           00185500
       X3=X2+DELX
                                                                           00185600
C
                                                                           00185700
       XXX(IDIRC)=X3
                                                                           00185800
       CALL RUNC
                                                                           00185900
       Y3=YYYF
                                                                           00186000
       WRITE(LUW, 205) X3, Y3, DELX
                                                                           00186100
С
                                                                           00186200
       GO TO 171
                                                                           00186300
C+>+>+>+>+>+
                                                                           00186400
17
      X3=X4
                                                                           00186500
       Y3=Y4
                                                                           00186600
C+>+>+>+>+>+
                                                                           00186700
                                                                           00186800
C+ IF THE FUNCTION EVALUATION IS SMALLER THAN THE PREVIOUS VALUE
                                                                           00186900
  THEN DOUBLE THE STEP SIZE.
С
                                                                           00187000
C
                                                                           00187100
171 DELX=2.*DELX
                                                                           00187200
                                                                           00187300
      X4=X3+DELX
                                                                           00187400
      XXX(IDIRC)=X4
                                                                           00187500
      CALL RUNC
                                                                           00187600
      Y4 = YYYF
                                                                           00187700
C+
                                                                           00187800
      IF(Y3-Y4) 20,22,22
                                                                           00187900
C+ FAILURE
                                                                           00188000
20 GO TO 90
                                                                           00188100
C+ SUCCESS
                                                                           00188200
22 GO TO 17
                                                                           00188300
С
                                                                           00188400
С
 WHEN THE OPTIMUM IS STRADDLED, EVALUATE THREE POINTS ABOUT THE MAX
                                                                           00188500
                                                                           00188600
C
                                                                           00188700
С
  POWELL ALGORITHM
                                                                           00188800
C
                                                                           00188900
80
     ZZ(1)=X1
                                                                           00189000
C+>+>+>+>+>+
                                                                           00189100
     YF(1)=Y1
                                                                           00189200
C+>+>+>+>+>+
                                                                           00189300
     ZZ(2)=X1+DELX/2.
                                                                           00189400
C+>NEED TO EVALUATE YF(2)
                                                                           00189500
     ZZ(3)=X2
                                                                           00189600
C+>+>+>+>+>+
                                                                           00189700
     YF(3)=Y2
                                                                           00189800
C+>+>+>+>+>+
                                                                           00189900
     GO TO 99
                                                                           00190000
C
                                                                           00190100
90
     ZZ(1)=X3
                                                                           00190200
C+>+>+>+>+>+
                                                                           00190300
     YF(1)=Y3
                                                                           00190400
C+>+>+>+>+>+
                                                                           00190500
      ZZ(2)=X3+DELX/2
                                                                           00190600
C+>NEED TO EVALUATE YF(2)
                                                                           00190700
```

```
ZZ(3)=X4
                                                                           00190800
C+>+>+>+>+>+
                                                                           00190900
      YF(3)=Y4
                                                                           00191000
C+>+>+>+>+>+
                                                                           00191100
      GO TO 99
                                                                           00191200
                                                                           00191300
C EVALUATE THE FUNCTION AT THESE THREE POINTS
                                                                           00191400
C+>INFACT ONLY AT THE MIDDLE POINT SINCE THE FUNC. VALUES AT THE OTHER
                                                                           00191500
C+> POINTS ARE KNOWN.
                                                                           00191600
                                                                           00191700
C+>+>+>+>+'>+
                                                                           00191800
99 XXX(IDIRC)=ZZ(2)
                                                                           00191900
      CALL RUNC
                                                                           00192000
      YF(2)=YYYF
                                                                           00192100
C
                                                                           00192200
C FIT A QUADRATIC TO THESE THREE POINTS
                                                                           00192300
С
                                                                           00192400
С
                                                                           00192500
C
                                                                           00192600
    NFITS COUNTS THE NUMBER OF FITS EMPLOYED
C
                                                                           00192700
C
                                                                           00192800
      NFITS=1
                                                                           00192900
   NQTFT2 IS THE MAX. NUMBER OF FITS ALLOWED
                                                                           00193000
      NQTFT2=10
                                                                           00193100
      WRITE(LUW, 211)
                                                                           00193200
 211 FORMAT(1x,//,20x,'xMAX',12x,'yMAX')
                                                                           00193300
      CONTINUE
                                                                           00193400
      A=ZZ(2)-ZZ(3)
                                                                           00193500
      B=ZZ(3)-ZZ(1)
                                                                           00193600
      C=ZZ(1)-ZZ(2)
                                                                           00193700
      D=ZZ(2)**2-ZZ(3)**2
                                                                           00193800
      E=ZZ(3)**2-ZZ(1)**2
                                                                           00193900
      F=2Z(1)**2-ZZ(2)**2
                                                                           00194000
                                                                           00194100
C+ ANALYTICALLY EVALUATE THE MIN. OF THE FITTED CURVE
                                                                           00194200
                                                                           00194300
      ZZT=.5*(D*YF(1)+E*YF(2)+F*YF(3))
                                                                           00194400
      ZZB=A*YF(1)+B*YF(2)+C*YF(3)
                                                                           00194500
      ZZM = ZZT/ZZB
                                                                           00194600
                                                                           00194700
  EVALUATE THE FUCTION VALUE AT THIS POINT
                                                                           00194800
                                                                           00194900
      XXX(IDIRC)=ZZM
                                                                           00195000
      CALL RUNC
                                                                           00195100
      YFM=YYYF
                                                                           00195200
      WRITE(LUW, 205)ZZM, YFM
                                                                           00195300
C
                                                                           00195400
C
    CHECK TO SEE IF ANY OF THE POINTS ARE WITHIN THE DESIRED ACCURACY
                                                                           00195500
С
                                                                           00195600
                                                                           00195700
      DO 100 J=1,3
                                                                           00195800
         WW(J) = DABS(ZZ(J) - ZZM)
                                                                           00195900
         IF(WW(J)-XLIM) 105,105,106
                                                                           00196000
C+>+>+>+>+>
                                                                           00196100
C+>+COULD SET XMIN TO ZZM AND YMIX TO YFM : THE MIN OF QUADRATIC
                                                                           00196200
105
         XMIN = ZZM
                                                                           00196300
         YMIN=YFM
                                                                           00196400
C
                                                                           00196500
         XXX(IDIRC)=XMIN
                                                                           00196600
         YYYF=YMIN
                                                                           00196700
                                                                           00196800
C+>+>+>+>+>
                                                                           00196900
         GO TO 200
                                                                           00197000
106
         CONTINUE
                                                                           00197100
C IF MAX. NUMBER OF OBJ. FUN. EVALUATIONS OR MAX. NUMBER OF FITS
                                                                           00197200
C ALLOWED IS REACHED, THEN STOP
                                                                           00197300
```

```
С
                                                                  00197400
          IF(NFUEVA.EQ.NFTERM .OR. NFITS.EQ.NQTFT2) GO TO 200
                                                                  00197500
100 CONTINUE
                                                                  00197600
                                                                  00197700
C
C+ SEE WHICH FUNCTION VALUE IS THE LARGEST AND REPLACE IT WITH
                                                                  00197800
                                                                  00197900
C+ THE INTERPOLATED MAX. POINT
Ċ+
                                                                  00198000
                                                                  00198100
     JK=1
     IF(YF(JK),LE,YF(2))JK=2
                                                                  00198200
     IF(YF(JK).LE.YF(3))JK=3
                                                                  00198300
117 ZZ(JK) = ZZM
                                                                  00198400
     YF(JK)=YFM
                                                                  00198500
                                                                  00198600
                                                                  00198700
  FIT A QUADRATIC TO THE NEW POINTS
C
                                                                  00198800
                                                                  00198900
     NFITS=NFITS+1
                                                                  000000
                                                                  00199100
     GO TO 98
                                                                  00199200
C+>THE MIN FUNC. VALUE IS ALREADY EVALUATED
                                                                  00199300
                                                                  00199400
                                                                  00199500
  XXX(IDIRC) AND YYYF ARE ALREADY SET EQUAL TO XMIN AND YMIN, WHICH
                                                                  00199600
  ARE THE MIN. OF XXX AND ITS OBJ. FUNCTION.
                                                                  00199700
C
                                                                  00199800
                                                                  00199900
200 IF(NFUEVA.EQ.NFTERM)WRITE(LUW,401)NFUEVA
401 FORMAT(1X,' MAX. NUMBER OF FUNCTION EVALUATONS REACHED.')
                                                                  00200000
                                                                  00200100
     IF(NFITS.EQ.NQTFT2)WRITE(LUW,402)
                                                                  00200200
 402 FORMAT(1X, MAX. NUMBER OF QUAD. FITS ALLOWED REACHED.')
                                                                  00200300
                                                                  00200400
  IF MAX. NUMBER OF FUNCTION EVALUATIONS OR QUAD. FITS ALLOWED
                                                                  00200500
  IS REACHED, THEN COPY THE BEST OPTIMUM OBTAINED SO FAR INTO XXX(1) AND YYYF.
                                                                  00200600
                                                                  00200700
                                                                  00200800
     IF (NFUEVA.EQ.NFTERM.OR.NFITS.EQ.NQTFT2)XXX(IDIRC) = TMPMIN(IDIRC)
                                                                  00200900
     IF (NFUEVA.EQ.NFTERM.OR.NFITS.EQ.NQTFT2) YYYF=TMPMIN(10)
                                                                  00201000
                                                                  00201100
     RETURN
     END
                                                                  00201200
С
                                                                  00201300
С
                                                                  00201400
                                                                  00201500
С
                                                                  00201600
00201700
C***********************
                                                                  00201800
00201900
     SUBROUTINE OMYSCH(IDIRC, IFLAG)
                                                                  00202000
00202100
C** THIS SUBROUTINE IS USED IN CONJUNCTION WITH SUBROUTINE TWOSCH
                                                                  00202200
C** FOR A TWO-DIMENSIONAL OPTIMIZATION.
                                                               *
                                                                  00202300
C**
                                                                  00202400
C** THIS SUBROUTINE IS CALLED FOR SEARCH ALONG ONE AXIS BUT
                                                                  00202500
C** USES THE INFORMATION OBTAINED FROM SEARCHING ALONG THE OTHER
                                                                  00202600
C** DIRECTION.
                                                                  00202700
C**
                                                                  00202800
C** THIS ROUTINE CALLS THE FOLLOWING SUBROUTINES:
                                                                  00202900
C**
       (1) SUBROUTINE RUNC TO CALCULATE THE LOSS-COST FUNCTION OF
                                                               *
                                                                  00203000
C**
           THE DYNAMIC MODEL.
                                                                  00203100
C**
                                                                  00203200
C**********************
                                                                  00203300
C**
                                                                  00203400
С
                                                                  00203500
С
                                                                  00203600
                                                                  00203700
C**NOTE THE FOLLOWING VARIABLES DEFINITION USED IN THIS ROUTINE.
                                                               ** 00203800
                                                                  00203900
```

```
IDIRC : SHOWS THE DIRECTION OF SEARCH ; VARIABLE NUMBER. E.G.:
                                                                            00204000
C
            IDIREC=1 -- SEARCH ALONG VARIABLE NUMBER ONE ; H
                                                                            00204100
C
            IDIREC=2 -- SEARCH ALONG VARIABLE NUMBER TWO ; ISTEPS
                                                                            00204200
Ċ
                                                                            00204300
    IFLAG: CONTAINS THE INFORMATION ABOUT THE SEARCH ALONG THE
                                                                            00204400
C
            DIRECTIONS;
                                                                            00204500
C
            IFLAG=1 -- SEARCH ALONG THE OTHER DIRECTION WAS SUCCESSFULL 00204600
Ċ
            IFLAG=2 -- SEARCH ALONG THE OTHER DIRECTION FAILED.
                                                                            00204700
C
            IFLAG=3 -- SEARCH ALONG BOTH DIRECTIONS FAILED.
                                                                            00204800
Ċ
                                                                            00204900
          : SHOWS THE NUMBER OF FAILURES ALONG THE CURRENT DIRECTION.
                                                                            00205000
C
                                                                            00205100
    NQFIT: THE NUMBER OF QUADRATIC FITS TO BE EMPLOYED . IT IS SET
                                                                            00205200
            EQUOL TO IFLAG FOR THIS SUBROUTINE.
                                                                            00205300
                                                                            00205400
С
    ISIDEL: THE SIGN OF THE LAST SUCCESSFUL STEP FOR EACH DIRECTION.
                                                                            00205500
С
                                                                            00205600
C**
                                                                            00205700
C*
                                                                            00205800
      IMPLICIT REAL*8(A-H,O-Z)
                                                                            00205900
      COMMON / MAIN1 /LUR, LUW
COMMON / DYNM6 / DEL(6), DELMN(6), DELMX(6), XQLIM(6)
                                                                            00206000
                                                                            00206100
      COMMON / DYOPT1 / XXX(6), YYYF
COMMON / DYOPT2 / ISIDEL(6), NFUEVA, NFTERM
DIMENSION ZZ(3), WW(3), YF(3)
                                                                            00206200
                                                                            00206300
                                                                            00206400
      DIMENSION IQACU(6)
                                                                            00206500
C**
                                                                            00206600
C**THE STARTING POINT AND ITS CORRESPONDING OBJ. FUN. VALUE
                                                                            C0206700
C**
                                                                            00206800
      X1=XXX(IDIRC)
                                                                            00206900
      Yl = YYYF
                                                                            00207000
C**
                                                                            00207100
C**INCREMENT THE VARIABLE AND EVALUATE THE FUNCTION
                                                                            00207200
C**
                                                                            00207300
 7
      X2=X1+DEL(IDIRC)
                                                                            00207400
      II=0
                                                                            00207500
С
                                                                            00207600
      XXX(IDIRC)=X2
                                                                            00207700
      CALL RUNC
                                                                            00207800
      Y2=YYYF
                                                                            00207900
C**
                                                                            00208000
C**IF THE NEW POINT IS BETTER (SMALLER OR THE SAME); SUCCESSS
                                                                            00208100
C**
                                                                            00208200
      IF(Y2.GT.Y1) GO TO 10
                                                                            00208300
C.. THEN DOUBLE THE STEP SIZE FOR THE NEXT SEARCH ALONG THIS DIRECTION.
                                                                            00208400
C..AND RETURN TO THE MAIN PROGRAM TO MOVE IN THE OTHER DIRECTION.
                                                                            00208500
      IF(DABS(DEL(IDIRC)).LT.DELMX(IDIRC))DEL(IDIRC)=2.D0*DEL(IDIRC)
                                                                            00208600
      XXX(IDIRC)=X2
                                                                            00208700
      YYYF=Y2
                                                                            00208800
C...SET FLAG TO SHOW THE SUCCESS.
                                                                            00208900
                                                                            00209000
C...AND KEEP THE SIGN OF THIS SUCCESSFUL MOVE
                                                                            00209100
                                                                            00209200
      IFLAG=1
                                                                            00209300
     ISIDEL(IDIRC)=DSIGN(1.D0,DEL(IDIRC))
                                                                            00209400
C>>>>
                                                                            00209500
      RETURN
                                                                            00209600
C......
                                                                            00209700
C**
                                                                            00209800
C**THE NEW POINT IS WORSE (BIGGER); FAILURE
C*.IF IT IS THE SECOND CONSEQUTIVE FAILURE IN THIS VARIABLE DIRECTION
                                                                            00209900
                                                                            00210000
C*.GIVEN THAT | DEL | IS EQUAL TO DELMN , THEN
                                                                            00210100
C*.THE MIN IS BRACKETED; USE POWEL, S METHOD TO FIT QUADRATICS.
                                                                            00210200
                                                                            00210300
 10
      IF(II.GE.1) GO TO 80
                                                                            00210400
                                                                            00210500
```

```
C*.IF IT IS THE FIRST FAILURE AFTER A SUCCESS , AND
                                                                           00210600
C...IF | DEL | IS GREATER THAN DELMN, THEN SET DEL=DELMN WHILE
                                                                           00210700
C...RESERVING THE SIGN OF DEL AND MOVE TO A NEW POINT.
                                                                           00210800
                                                                           00210900
C... ( DSIGN(A,B) IS A FORTRAN FUNCTION WHICH TRANSFERS THE SIGN OF
                                                                           00211000
C... B TO A.)
                                                                           00211100
                                                                           00211200
      IF(DABS(DEL(IDIRC)).GT.DELMN(IDIRC))DEL(IDIRC)=DSIGN(
                                                                           00211300
                                           DELMN(IDIRC),DEL(IDIRC))
                                                                           00211400
      IF (DABS (DEL (IDIRC)).GT.DELMN (IDIRC))GO TO 7
                                                                           00211500
C...OTHERWISE, REVERSE THE SEARCH DIRECTION
                                                                           00211600
                                                                           00211700
      DEL(IDIRC) = -DSIGN(DELMN(IDIRC), DEL(IDIRC))
                                                                           00211800
                                                                           00211900
      II=II+l
C...SAVE THE CURRENT POINTS FOR POWELL, S METHOD , IN THE CASE THAT
                                                                           00212000
C...BECOMES NECCESSARY TO USE THEM.
                                                                           00212100
      ZZ(1)=X2
                                                                           00212200
      YF(1)=Y2
                                                                           00212300
      ZZ(2)=X1
                                                                           00212400
      YF(2)=Y1
                                                                           00212500
C...AND MOVE TO A NEW POINT.
                                                                           00212600
                                                                           00212700
      X2=X1+DEL(IDIRC)
                                                                           00212800
      GO TO 8
                                                                           00212900
00213000
C**
                                                                           00213100
C**MODIFIED POWELL, S METHOD ; FIT QUADRATIC TO ESTIMATE MIN.
                                                                           00213200
                                                                           00213300
C**-----
C**
                                                                           00213400
80 WRITE(LUW,216)
216 FORMAT(lX,//,lX,' POWELL,S METHOD')
C*..INCREMENT IFLAG , SINCE WE HAD TWO CONSEQUTIVE FAILURES IN THIS
                                                                           00213500
                                                                           00213600
                                                                           00213700
C*..DIRECTION. SET NUMBER OF QUAD. FITS EQUAL TO IFLAG.
                                                                           00213800
                                                                           00213900
      IFLAG=IFLAG+1
                                                                           00214000
      NQFIT=IFLAG
                                                                           00214100
      WRITE(LUW, 217)NQFIT
                                                                           00214200
 217 FORMAT(1x,'NUMBER OF QUADRATIC FITS EMPLOYED=',13)
                                                                           00214300
                                                                           00214400
C*.. THREE POINTS ARE REQUIRED FOR A QUADRATIC FIT . TWO POINTS
                                                                           00214500
C*..HAVE ALREADY BEEN STORED IN ZZ(1) AND ZZ(2)
                                                                           00214600
                                                                           00214700
      ZZ(3)=X2
                                                                           00214800
      YF(3)=Y2
                                                                           00214900
                                                                           00215000
C*+
C*+IF REQUIRED PRECISION , XQLIM, IS EQUAL TO THE MIN. LIMIT ON
                                                                           00215100
C*+STEP SIZE, THEN SKIP THE POWELL, SMETHOD. THIS STRATEGY C*+IS ESPECIALLY USEFUL IN GUARDING AGAINST INTERPOLATIONS OF
                                                                           00215200
                                                                           00215300
C*+INTEGER-VALUED VARIABLES.
                                                                           00215400
C*+
                                                                           00215500
      IF(XQLIM(IDIRC).GT.DELMN(IDIRC)-1.D-14 )GO TO 350
                                                                           00215600
C*+
                                                                           00215700
                                                                           00215800
C*.. ANALYTICALL EVALUATE THE MIN. OF THE FITTED CURVE ; ZZM
                                                                           00215900
                                                                           00216000
      AAAA=ZZ(2)-ZZ(3)
                                                                           00216100
      BBBB=ZZ(3)-ZZ(1)
                                                                           00216200
       CCCC=ZZ(1)-ZZ(2)
                                                                           00216300
      DDDD=ZZ(2)**2-ZZ(3)**2
                                                                           00216400
      EEEE=ZZ(3)**2-ZZ(1)**2
                                                                           00216500
      FFFF=ZZ(1)**2-ZZ(2)**2
                                                                           00216600
       ZZT=.5*(DDDD*YF(1)+EEEE*YF(2)+FFFF*YF(3))
                                                                           00216700
       ZZB=AAAA*YF(1)+BBBB*YF(2)+CCCC*YF(3)
                                                                           00216800
       ZZM=ZZT/ZZB
                                                                           00216900
                                                                           00217000
C*..EVALUATE THE FUNCTION VALUE AT THIS POINT
```

```
С
                                                                  00217200
     XXX(IDIRC) = ZZM
                                                                  00217300
     CALL RUNC
                                                                  00217400
     YFM=YYYF
                                                                  00217500
     WRITE(LUW, 218)
                                                                  00217600
218 FORMAT(1X,' THE ABOVE IS THE MIN OF QUADRATIC. << ')
                                                                  00217700
                                                                  00217800
C*..IF ANY OF THE POINTS ARE WITHIN THE DESIRED (QUITTING) ACCURRACY
                                                                  00217900
C*..THEN QUIT FITTING QUADRATIC. SET IQACU=1 TO SHOW THIS SITUATION
                                                                  00218000
                                                                  00218100
     IQACU(IDIRC)=0
                                                                  00218200
     DO 300 J=1,3
                                                                  00218300
         WW(J) = DABS(ZZ(J) - ZZM)
                                                                  00218400
         IF(WW(J).GT.XQLIM(IDIRC))GO TO 300
                                                                  00218500
             XMIN=ZZ(J)
                                                                  00218600
             YMIN=YF(J)
                                                                  00218700
C
                                                                  00218800
             XXX(IDIRC)=XMIN
                                                                  00218900
             YYYF=YMIN
                                                                  00219000
C
                                                                  00219100
             IQACU(IDIRC)=1
                                                                  00219200
             GO TO 400
                                                                  00219300
300 CONTINUE
                                                                  00219400
                                                                  00219500
C*..SEE WHICH FUNCTION VALUE IS THE LARGGEST (WORST) AND REPLACE IT
                                                                  00219600
C*..WITH THE QUADRATIC, S MIN. NOTE YF(JK) CONTAINS THE LARRGEST VALUE.
                                                                  00219700
                                                                  00219800
                                                                  00219900
     IF(YF(JK).LE.YF(2))JK=2
                                                                  00220000
     IF(YF(JK).LE.YF(3))JK=3
                                                                  00220100
     ZZ(JK) = ZZM
                                                                  00220200
     YF(JK)=YFM
                                                                  00220300
                                                                  00220400
C*..SEE IF MORE FITS ARE REQUIRED, FIT ONE TO THE NEW POINTS.
                                                                  00220500
                                                                  00220600
     NQFIT=NQFIT-1
                                                                  00220700
     IF(NOFIT.GT.0) GO TO 90
                                                                  00220800
                                                                  00220900
C^{\star}..ENOUGH QUADRATIC FITTED , SET THE MINIMUM EQUAL TO THE SMALLEST C^{\star}..OF THE LAST THREE POINTS IN ORDER TO RETURN FROM THIS ROUTINE.
                                                                  00221000
                                                                  00221100
                                                                  00221200
 350 JK=1
                                                                  00221300
     IF(YF(JK).GE.YF(2))JK=2
                                                                  00221400
     IF(YF(JK).GE.YF(3))JK=3
                                                                  00221500
                                                                  00221600
C*.. THE MIN. IS IN ZZ(JK)
                                                                  00221700
                                                                  00221800
     XXX(IDIRC) = ZZ(JK)
                                                                  00221900
     YYYF=YF(JK)
                                                                  00222000
C
                                                                  00222100
С..
                                                                  00222200
C
                                                                  00222300
 400
     RETURN
                                                                  00222400
     END
                                                                  00222500
C
                                                                  00222600
C
                                                                  00222700
C
                                                                  00222800
                                                                  00222900
     C****
SUBROUTINE RUNC
                                                                  00223300
С
                                                                * 00223500
C** THIS ROUTINE CALCULATES THE LOSS-COST OF THE DUNAMIC MODEL.
                                                                * 00223600
C**
                                                                * 00223700
```

```
C** THIS ROUTINE CALLS THE FOLLOWING SUBROUTINE:
                                                                         * 00223800
C** (1) SUBROUTINE APP TO CALCULATE THE AVERAGE TIME OUT OF CONTROL* 00223900
C**
           BEFORE A POINT FALLS AND IS CHARTED OUTSIDE THE CONTROL * 00224000
                                                                        * 00224100
C**
           LIMITS.
C**
                                                                        * 00224200
    (2) SUBROUTINE CMAINT TO CALCUALTE AVERAGE HOURLY COST OF
                                                                        * 00224300
          MAINTAINING THE CONTROL CHART.
C**.
C**
     (3) SUBROUTINE FALSA TO CALCULATE THE EXPECTED NUMBER OF FALSE * 00224400
                                                                         * 00224500
C**
           ALARMS.
C**
                                                                         * 00224600
                                                                         * 00224700
C** THE RESTRICTION ON M, IM, D, AND ID IS INCORPORATED AS:
C**
                                                                         * 00224800
C** (1) PENALTY FUNCTION
C** (2) BARRIER FUNCTION
                                                                         * 00224900
                                                                        * 00225000
C** A FLAG CALLED IPENAL IS USED TO SELECT EITHER OF (1) OR (2).
C**
                                                                        * 00225100
C**
                                                                           00225300
C*
                                                                           00225400
      IMPLICIT REAL*8(A-H,O-Z)
                                                                           00225500
      REAL*8 LAMBDA,L,IL,M,IM,ID
REAL*8 NF,IH,HF,IK,KF
                                                                           00225600
                                                                           00225700
      COMMON / MAIN1 /LUR, LUW
                                                                           00225800
     COMMON / MAINI /LUK,LUW

COMMON / DCDY1 / DELTA, B,C,DD,E,VZMV1,T,W

COMMON / DYNM1 / THETA,ETA

COMMON / DYNM4 / PROBPT

COMMON / DYOPT1 / XXX(6),YYYF

COMMON / DYOPT2 / ISIDEL(6), NFUEVA,NFTERM

COMMON / DYOPT3 / TMPMIN(10)

COMMON/Blll1/L,IL,M,IM,D,ID

COMMON/Blll1/LSTEPS
                                                                           00225900
                                                                           00226000
                                                                           00226100
                                                                           00226200
                                                                           00226300
                                                                           00226400
                                                                           00226500
         COMMON/B1111/ISTEPS
                                                                           00226600
        COMMON/BE1/ITR
                                                                           00226700
С
                                                                           00226800
C.....
                                                                           00226900
С
                                                                           00227000
C**
                                                                           00227100
C**
                                                                           00227200
C*+SET IPENAL TO 1 IF PENALTY METHOD IS DESIRED
                                                                           00227300
C*+SET IPENAL TO 2 IF BARRIER METHOD IS DESIRED
                                                                           00227400
                                                                           00227500
IPENAL=1
                                                                           00227600
C**
                                                                           00227700
C**INCREMENT THE NUMBER OF OBJ. FUNCTION EVALUATIONS.
                                                                           00227800
C**
                                                                           00227900
      NFUEVA=NFUEVA+1
                                                                           00228000
      WRITE(LUW,1)NFUEVA
                                                                           00228100
      FORMAT(1X,' BELOW IS THE ', 14, 'TH FUNCTION EVALUATED.')
1
                                                                           00228200
C**
                                                                           00228300
C**
                                                                           00228400
C**IF THE LIMIT OF OBJ. FUN. EVALUATIONS IS REACHED THEN SET YYYF TO A 00228500
C** VERY BIG VALUE AND RETURN.
                                                                           00228600
C**
                                                                           00228700
      IF(NFUEVA.GT.NFTERM)YYYF=999888777.
                                                                           00228800
      IF (NFUEVA.GT.NFTERM) RETURN
                                                                           00228900
C**
                                                                           00229000
C**IF THIS IS THE FIRST TIME THIS SUBROUTINE IS CALLED THEN INITIALIZE 00229100
C**RQQR.
                                                                           00229200
                                                                           00229300
      IF(NFUEVA.GT.1) GO TO 95
                                                                           00229400
                                                                           00229500
C+.INITIALIZE ROOR , PARAMETER USED IN PENALTY AND BARRIER METHODS.
                                                                           00229600
C+.
                                                                           00229700
       RQQR=1.D0
                                                                           00229800
C*.
                                                                           00229900
C ..........
                                                                           00230000
C*.
                                                                           00230100
C*.FOR THIS AND THE SUBROUTINES CALLED FROM THIS ROUTINE, THE
                                                                           00230200
C*.DECISION VARIABLES ARE RENAMED AS FOLLOWS.
                                                                           00230300
```

```
C*. ISTEPS IS THE NUMBER OF STEPS, SAMPLES, TAKEN TO GET TO
                                                                         00230400
C*. A DESIRED QUANTILE VALUE, PROBPT.
                                                                          00230500
C*.
                                                                          00230600
C*. L IS INTERVAL SIZE FACTOR,
                                   IL IS INITIAL LENGTH OF INTERVAL
                                                                          00230700
C*. M IS CNT. LMT. WIDTH FACTOR, IM IS INITIAL WIDTH OF CNT. LMT.
                                                                          00230800
C*. D IS SAMPLE SIZE FACTOR,
                                   ID IS INITIAL SAMPLE SIZE
                                                                          00230900
C*.
                                                                          00231000
C*.THEREFORE, THE VARIABLES IN EACH OF THE FOLLOWING COLUMNS ARE EQUAL. 00231100
Č*.
                                                                          00231200
Č*.
      XXX(l)
               XXX(2)
                        XXX(3)
                                 XXX(4)
                                          XXX(5)
                                                    XXX(6)
                                                                          00231300
C*.
      HF 🕯
                                                                          00231400
               ISTEPS
                         ΙK
                                  KF
                                          IN
                                                      NF
C*.
       L
               ISTEPS
                         IM
                                   М
                                           ID
                                                       D
                                                                          00231500
C*.
                                                                          00231600
с..
                                                                          00231700
C*
                                                                          00231800
C**
                                                                          00231900
C**
                                                                          00232300
C**SET ALL THE VARIABLES EQUAL TO THEIR CORRESPONDING DECISION VARIABLES00232400
C**
                                                                          00232500
  95
        L=XXX(1)
                                                                          00232600
        ISTEPS=XXX(2)
                                                                          00232700
                                                                          00232800
        IM=XXX(3)
        M=XXX(4)
                                                                          00232900
        ID=XXX(5)
                                                                          00233000
        D=XXX(6)
                                                                          00233100
C**
                                                                          00233200
C**FIRST CALCULATE IL (IH); THE INITIAL SAMPLING INTERVAL TO ACHIEVE
                                                                          00233300
C**-----
                                                                          00233400
C**A DESIRED QUANTILE (PROBPT) IN ISTEPS SAMPLS.
                                                                          00233500
                                                                          00233600
C*.FIND THE VALUE THAT THE SUM OF ISTEPS INTERVALS SHOULD BE EQUAL TO.
                                                                          00233700
C*.
                                                                          00233800
         SUMATN=DEXP(DLOG(-DLOG(1.D0-PROBPT))/ETA-DLOG(THETA))
                                                                          00233900
                                                                          00234000
C*.CALCULATION OF IL (OR IH) IF L IS 1.
                                                                          00234100
C*.
                                                                          00234200
         IF(L.EQ.1)IL=SUMATN/ISTEPS
                                                                          00234300
         IF(L.EQ.1)GO TO 97
                                                                          00234400
                                                                          00234500
C*.CALCULATION OF IL (OR IH) IF L IS NOT EQUAL TO 1.
                                                                          00234600
                                                                          00234700
         BX=ISTEPS*DLOG(L)
                                                                          00234800
         IL=SUMATN*(1.D0-L)/(1.D0-DEXP(BX))
                                                                          00234900
C**
                                                                          00235000
C**ITR IS USED TO CALCULATE NUMBER OF ITERATIONS FOR DO LOOP
                                                                          00235100
C**
                                                                          00235200
 97
         ITR=ISTEPS+13
                                                                          00235300
C
                                                                          00235400
           THE RESTRICTION ON M , IM , D , AND ID
                                                                          00235500
c-
                                                                          00235600
C**
                                                                          00235700
C** CALCULATE RIDE ; ENDING SAMPLE SIZE
                                                                          00235800
C**
                                                                          00235900
      RIDE=ID*(D**(ISTEPS-1))
                                                                          00236000
C**
                                                                          00236100
C**CALCULATE RIME ; ENDING CONTROL LIMIT WIDTH
                                                                          00236200
C**
                                                                          00236300
      RIME=IM*(M**(ISTEPS-1))
                                                                          00236400
C**
                                                                          00236500
C**CALCULATE UPPER LIMIT ON IM , AND RIME
                                                                          00236600
C**
                                                                          00236700
      UPIM=DELTA*DSQRT(ID)
                                                                          00236800
      UPRIME=DELTA*DSQRT(RIDE)
                                                                          00236900
C**
                                                                          00237000
      IF (IM.LT.UPIM.AND.RIME.LT.UPRIME)GO TO 105
                                                                          00237100
C**
                                                                          00237200
```

```
C*+++++++++++++++
                                                                           00237300
C*+PENALTY METHOD +
                                                                           00237400
C*++++++++++++++
                                                                           00237500
105
     IF(IPENAL.EQ.2) GO TO 106
                                                                           00237600
      YYQQ=DMIN1((UPIM-IM),0.0D0)**2+DMIN1((UPRIME-RIME),0.0D0)**2
                                                                           00237700
      ROOR=ROOR*.65D0
                                                                           00237800
      IF(RQQR.LT.1.E-45)RQQR=1.E-45
                                                                           00237900
      YYYQ=YYQQ/RQQR
                                                                           00238000
      GO TO 107
                                                                           00238100
C*+++++++++++++++
                                                                           00238200
C*+BARRIER METHOD +
                                                                           00238300
C*+++++++++++++++
                                                                           00238400
       YYQQ=(1.D0/(UPIM-IM)+1.D0/(UPRIME-RIME))
                                                                           00238500
106
       RQQR=RQQR*.6D0
                                                                           00238600
       YYYQ=YYQQ*RQQR
                                                                           00238700
С
                                                                           00238800
c
c
                                                                           00238900
                                                                           00239000
C**
                                                                           00239100
C**CALL APP TO CALCULATE A DOUBLE PRIME; ADBP.
                                                                           00239200
C**
                                                                           00239300
107
        CALL APP(ADBP)
                                                                           00239400
C**
                                                                           00239500
C**CYCLE IS THE AVERAGE CYCLE LENGTH.
                                                                           00239600
C**
                                                                           00239700
C*.RMEAN IS MEAN OF WEIBALL DISTRIBUTION
                                                                           00239800
         XGA=1.D0+1.D0/ETA
                                                                           00239900
C*.DGAMMA IS AN IMSL ROUTINE TO CALCULATE GAMMA FUNCTION
                                                                           00240000
         RMEAN=1.D0/THETA*DGAMMA(XGA)
                                                                           00240100
C*.
                                                                           00240200
        CYCLE=RMEAN+ADBP+DD
                                                                           00240300
C**
                                                                           00240400
C**GAMMA IS THE PROPORTION OF TIME A PROCESS WILL BE OUT OF CONTROL.
                                                                           00240500
C**
                                                                           00240600
        GAMMA=(ADBP+DD)/CYCLE
                                                                           00240700
C**
                                                                           00240800
C**CALL FALSA TO CALCULATE ENFALS; EXPECTED NUMBER OF FALSE ALARMS
                                                                           00240900
C**
                                                                           00241000
         CALL FALSA (ENFALS)
                                                                           00241100
C**
                                                                           00241200
C**CALL CMAINT TO CALCULATE BAHCM
                                                                           00241300
C**
                                                                           00241400
         CALL CMAINT (BAHCM)
                                                                           00241500
C**
                                                                           00241600
C**
                                                                           00241700
C**YYYF IS THE LOSS COST PER UNIT TIME; OBJECTIVE FUNCTION
                                                                           00241800
C**
                                                                           00241900
        YYYP=GAMMA*VZMV1+T*ENFALS/CYCLE+W/CYCLE+BAHCM
                                                                           00242000
                                                                           00242100
C*+ADD THE PENALTY TO OBJ. FUN.
                                                                           00242200
C*+
                                                                           00242300
        YYYF=YYYP+YYYO
                                                                           00242400
C
                                                                           00242500
C**
                                                                           00242600
C**KEEP THE MIN. SO FAR IN AN ARRAY CALLED TMPMIN
                                                                           00242700
                                                                           00242800
       IF(YYYF.GE.TMPMIN(10))RETURN
                                                                           00242900
C*.
                                                                           00243000
C*.IF NEWLY CALCULATED YYYF IS BETTER , THEN UPDDATE TMPMIN
                                                                           00243100
C*.
                                                                           00243200
       DO 19 I=1,6
                                                                           00243300
         TMPMIN(I)=XXX(I)
                                                                           00243400
 19
       CONTINUE
                                                                           00243500
       TMPMIN(10)=YYYF
                                                                           00243600
C
                                                                           00243700
       RETURN
                                                                           00243800
```

```
END
                                                                     00243900
C
                                                                      00244000
                                                                      00244100
C
                                                                      00244200
                                                                      00244300
SUBROUTINE DYMEVA
                                                                      00244700
C**
                                                                   * 00244900
                                                                   * 00245000
C** THIS SUBROUTINE CALCULATES THE LOSS-COST FOR THE DYNAMIC MODEL
C** FOR THE GIVEN DESIGN TO BE EVALUATED.
                                                                   * 00245100
C**
                                                                   * 00245200
                                                                   * 00245300
C** THIS ROUTINE CALLS THE FOLLOWING SUBROUTINE:
      (1) SUBROUTINE APP TO CALCULATE THE AVERAGE TIME OUT OF CONTROL* 00245400 BEFORE A POINT FALLS AND IS CHARTED OUTSIDE THE CONTROL * 00245500
C**
C**
C**
                                                                   * 00245600
     (2) SUBROUTINE CMAINT TO CALCUALTE AVERAGE HOURLY COST OF
C**
                                                                   * 00245700
C**
                                                                   * 00245800
          MAINTAINING THE CONTROL CHART.
C**
      (3) SUBROUTINE FALSA TO CALCULATE THE EXPECTED NUMBER OF FALSE * 00245900
C**
                                                                   * 00246000
         ALARMS.
C**
                                                                   * 00246100
C**
                                                                    * 00246200
C**
                                                                      00246400
C*
                                                                      00246500
     IMPLICIT REAL*8(A-H,O-Z)
                                                                      00246600
        REAL*8 LAMBDA, L, IL, M, IM, ID
                                                                      00246700
     REAL*8 NF, IH, HF, IK, KF
                                                                      00246800
     REAL*8 NF,IH,HF,IK,KF
COMMON / MAIN1 /LUR,LUW
COMMON / DCDY1 / DELTA, B,C,DD,E,VZMV1,T,W
COMMON / DYNM1 / THETA,ETA
COMMON / DYNM3 / IN, NF,IH,HF,IK,KF
COMMON / DYNM4 / PROBPT
COMMON / DYNM7 / DYMLCS
COMMON / DYNM8 / ISTPP
COMMON / DYOPT3 / TMPMIN(10)
COMMON / CMN1 / CUPROX
COMMON/A1111/L,IL,M,IM,D,ID
COMMON/B1111/ISTEPS
                                                                      00246900
                                                                      00247000
                                                                      00247100
                                                                      00247200
                                                                      00247300
                                                                      00247400
                                                                      00247500
                                                                      00247600
                                                                      00247700
                                                                      00247800
        COMMON/B1111/ISTEPS
                                                                      00247900
        COMMON/BE1/ITR
                                                                      00248000
С
                                                                      00248100
                                                                      00248200
                                                                      00248300
                                                                      00248400
C*.FOR THIS AND THE SUBROUTINES CALLED FROM THIS ROUTINE, THE
                                                                      00248500
C*.DECISION VARIABLES ARE RENAMED AS FOLLOWS.
                                                                      00248600
C^{\star}. ISTEPS IS THE NUMBER OF STEPS, SAMPLES, TAKEN TO GET TO
                                                                      00248700
C*. A DESIRED QUANTILE VALUE, PROBPT.
                                                                     .00248800
C*.
                                                                      00248900
C*. L IS INTERVAL SIZE FACTOR,
                                IL IS INITIAL LENGTH OF INTERVAL
                                                                      00249000
C*. M IS CNT. LMT. WIDTH FACTOR, IM IS INITIAL WIDTH OF CNT. LMT.
C*. D IS SAMPLE SIZE FACTOR, ID IS INITIAL SAMPLE SIZE
                                                                      00249100
                                                                      00249200
                                                                      00249300
C*.THEREFORE, THE VARIABLES IN EACH OF THE FOLLOWING COLUMNS ARE EQUAL. 00249400
C*.
                                                                      00249500
                               xxx(4) xxx(5) xxx(6)
Č*.
     XXX(1)
                       XXX(3)
             XXX(2)
                                                                      00249600
           ISTEPS
                     IK
C*.
                               KF IN NF
     HF
                                                                      00249700
      L'
C*.
              ISTEPS
                                        ID
                        ΙM
                                 M
                                                    D
                                                                      00249800
C*.
                                                                      00249900
                                                                      00250000
C......
                                                                      00250100
C*
                                                                      00250200
C**
                                                                      00250300
C**SET ALL THE VARIABLES EQUAL TO THEIR CORRESPONDING DECISION VARIABLES00250400
```

```
C**
                                                           00250500
                                                           00250600
      L=HF
                                                           00250700
      IL=IH
      ISTEPS=ISTPP
                                                           00250800
                                                           00250900
      IM=IK
                                                           00251000
      M=KF
      ID=IN
                                                           00251100
      D=NF
                                                           00251200
C**
                                                           00251300
C**ITR IS USED TO CALCULATE NUMBER OF ITERATIONS FOR DO LOOP
                                                           00251400
C**
                                                           00251500
00251600
 97
       ITR=ISTEPS+13
C**
                                                           00251700
C**CALL APP TO CALCULATE A DOUBLE PRIME: ADBP.
                                                           00251800
C**
                                                           00251900
107
      CALL APP(ADBP)
                                                           00252000
                                                           00252100
C**
C**CYCLE IS THE AVERAGE CYCLE LENGTH.
                                                           00252200
C**
                                                           00252300
C*.RMEAN IS MEAN OF WEIBALL DISTRIBUTION
                                                           00252400
       XGA=1.D0+1.D0/ETA
                                                           00252500
C*.DGAMMA IS AN IMSL ROUTINE TO CALCULATE GAMMA FUNCTION
                                                           00252600
       RMEAN=1.D0/THETA*DGAMMA(XGA)
                                                           00252700
                                                           00252800
C*.
      CYCLE=RMEAN+ADBP+DD
                                                           00252900
C**
                                                           00253000
C**GAMMA IS THE PROPORTION OF TIME A PROCESS WILL BE OUT OF CONTROL.
                                                           00253100
C**
                                                           00253200
      GAMMA=(ADBP+DD)/CYCLE
                                                           00253300
                                                           00253400
C**CALL FALSA TO CALCULATE ENFALS; EXPECTED NUMBER OF FALSE ALARMS
                                                           00253500
C**
                                                           00253600
       CALL FALSA (ENFALS)
                                                           00253700
C**
                                                           00253800
C**CALL CMAINT TO CALCULATE BAHCM
                                                           00253900
C**
                                                           00254000
       CALL CMAINT (BAHCM)
                                                           00254100
C**
                                                           00254200
C**
                                                           00254300
C**YYYF IS THE LOSS COST PER UNIT TIME: OBJECTIVE FUNCTION
                                                           00254400
C**
                                                           00254500
      YYYP=GAMMA*VZMV1+T*ENFALS/CYCLE+W/CYCLE+BAHCM
                                                           00254600
C
                                                           00254700
      DYMLCS=100.D0*YYYP
                                                           00254800
     RETURN
                                                           00254900
     END
                                                           00255000
С
                                                           00255100
C
                                                           00255200
C
                                                           00255300
                                                           00255400
C
                                                           00255500
                                                           00255600
SUBROUTINE APP(ADBP)
                                                           00256000
C**
                                                         * 00256200
C** THIS ROUTINE CALCULATES THE AVERAGE TIME OUT OF CONTROL BEFORE THE* 00256300
C** DETECTING SAMPLE IS PLOTED ON THE CHART.
                                                         * 00256400
C**
                                                          * 00256500
C**
                                                           00256700
C*
                                                           00256800
     IMPLICIT REAL*8(A-H,O-Z)
                                                           00256900
     REAL*8 NUM, INTERM, LAMBDA, M, L, ITINTR, IM, ID, IL
                                                           00257000
```

```
REAL*8 IJTINT, ICUINT
                                                                            00257100
      REAL*8 ICNTLT
                                                                            00257200
      COMMON / DCDY1 / DELTA, B,C,DD,E,VZMV1,T,W
COMMON / DYNM1 / THETA,ETA
                                                                            00257300
                                                                            00257400
                                                                            00257500
C**THETA CORSPONDS TO LAMBDA ( ALPHA)
                                          ETA CORRESPONDS TO BETA
                                                                            00257600
                                                                            00257700
         COMMON/Allll/L,IL,M,IM,D,ID
                                                                            00257800
         COMMON/B1111/ISTEPS
                                                                            00257900
         COMMON/BE1/ITR
                                                                            00258000
                                                                            00258100
C
C**
                                                                            00258200
C**
     L IS INTERVAL SIZE FACTOR
                                                                            00258300
C**
   M IS SPEC. LIMIT FACTOR
                                                                            00258400
C**
      D IS SAMPLE SIZE FACTOR
                                                                            00258500
C**
                                                                            00258600
С
                                                                            00258700
      ADBP=0.
                                                                            00258800
      OCUINT=0.
                                                                            00258900
      ITINTR=IL/L
                                                                            00259000
C**
                                                                            00259100
C**OCNTLT IS OUTER LOOP CONTROL LIMIT.
                                                                            00259200
C**
                                                                            00259300
      OCNTLT=IM/M
                                                                            00259400
C
                                                                            00259500
      ILT=ITR-13
                                                                            00259600
С
                                                                            00259700
      DO 200 I=1,ILT
                                                                            00259800
                                                                            00259900
C**ITINTR IS THE LENGTH OF THE (I-1)TH INTERVAL: H SUB (I-1)
                                                                            00260000
                                                                            00260100
C** IJTINT IS THE LENGTH OF THE (I+J-1)TH INTERVAL; H SUB (I+J-1)
                                                                            00260200
                                                                            00260300
      ITINTR=ITINTR*L
                                                                            00260400
      IJTINT=ITINTR/L
                                                                            00260500
                                                                            00260600
      OCNTLT=OCNTLT*M
                                                                            00260700
                                                                            00260800
C+=+CONTROL LIMIT SPREAD RESTRICTED BETWEEN .5 AND 5.5.
                                                                            00260900
                                                                            00261000
      IF(OCNTLT.LT.0.5)OCNTLT=0.5
                                                                            00261100
      IF(OCNTLT.GT.5.5)OCNTLT=5.5
                                                                            00261200
                                                                            00261300
      ICNTLT=OCNTLT/M
                                                                            00261400
                                                                            00261500
C**SAMPLI IS THE SAMPLE SIZE AT THE (I-1)TH INTERVAL
                                                                            00261600
С
                                                                            00261700
      SAMPLI = ID*D**(I-2)
                                                                            00261800
C
                                                                            00261900
C
                                                                            00262000
      ICUINT=0.
                                                                            00262100
      PRODPL=0.0
                                                                            00262200
      INTERM=0.
                                                                            00262300
С
                                                                            00262400
      J=0
                                                                            00262500
300
      J=J+1
                                                                            00262600
      IJTINT=IJTINT*L
                                                                            00262700
                                                                            00262800
      ICNTLT=ICNTLT*M
                                                                            00262900
                                                                            00263000
C+=+CONTROL LIMIT SPREAD RESTRICTED BETWEEN .5 AND 5.5
                                                                            00263100
                                                                            00263200
       IF(ICNTLT.LT.0.5)ICNTLT=0.5
                                                                            00263300
       IF(ICNTLT.GT.5.5)ICNTLT=5.5
                                                                            00263400
C+=+=+
                                                                            00263500
C
                                                                            00263600
```

```
C**ICUINT IS THE CUMULATIVE INTERVAL STARTING FROM THE ITH INTERVAL TO 00263700
                                                                           00263800
C**, AND EXCLUDING, THE (I+J)TH INTERVAL.
                                                                           00263900
C**PNDTCB IS PROB. OF NOT DETECTING BEFORE, AND EXCLUDING, THE JTH SAMPL00264000
C** TAKEN AFTER THE PROCESS WENT OUT OF CONTROL IN THE ITH INTRVAL.
                                                                           00264100
C^{**} THAT IS PNDTCB=(1-PI)(1-PI+1)...(1-PI+J-1)
                                                                           00264200
                                                                           00264300
                                                                           00264400
         ICUINT=ICUINT+IJTINT
                                                                           00264500
C**SAMPLI IS THE SAMPLE SIZE AT THE (I+J-1)TH INTERVAL
                                                                           00264600
C++ALSO, MAKE SURE THAT THE SAMPLE SIZE IS INTEGER AND >=2.
                                                                           00264700
                                                                           00264800
C++
                                                                           00264900
         SAMPLI = SAMPLI *D
         ISAMPL=SAMPLI+.499999D0
                                                                           00265000
         IF(ISAMPL.LT.2)ISAMPL=2
                                                                           00265100
C+=+=+=+
                                                                           00265200
C+=+UPPER LIMIT FOR SAMPLE SIZE IS 1000
                                                                           00265300
                                                                           00265400
C+=+=+=+
         IF(ISAMPL.GT.1000)ISAMPL=1000
                                                                           00265500
                                                                           00265600
C+=+=+=+
         SAMPLS=ISAMPL
                                                                           00265700
                                                                           00265800
C**Y IS ONE OF THE LIMITS OF THE STD. NORMAL INTEGRATION.
                                                                           00265900
C** THE OTHER LIMIT IS INFINITY.
                                                                           00266000
C**NOTE THAT P IS THE INTEGRAL FROM -INFINITY TO Y, AND PDCTJ IS THE INTO0266100
C**INTEGRAL FROM Y TO INFINITY OF THE STD. NORMAL DISTRIBUTION.
                                                                           00266200
                                                                           00266300
         Y=ICNTLT-DELTA*DSORT(SAMPLS)
                                                                           00266400
C
                                                                           00266500
C** MDNOR IS AN IMSL ROUTINE WHICH CALCULATES NORMAL DENSITY INTEGRAL
                                                                           00266600
                                                                           00266700
         CALL MDNORD(Y,P)
                                                                           00266800
                                                                           00266900
C
C**PDTC IS THE PROB OF DETECTING THE SHIFT ON THE (I+J-1)TH SAMPLE
                                                                           00267000
C
                                                                           00267100
         PDTC=1.-P
                                                                           00267200
          IF(I.LT.5)WRITE(6,912)Y,PDTC
                                                                           00267300
C
 912
          FORMAT(1X,'Y & PDTC ',2E15.8)
                                                                           00267400
С
                                                                           00267500
                                                                           00267600
C++USING LN
                                                                           00267700
                                                                           00267800
C++
C++PLN IS LN OF PDTC
                                                                           00267900
C++PRODPL IS LN OF PROB. OF NOT DETECTING; PNDTCB C++TLN IS LN OF (ICUINT+SAMPLI*E)
                                                                           00268000
                                                                           00268100
                                                                           00268200
      PLN=DLOG(PDTC)
                                                                           00268300
      TERM=ICUINT+SAMPLS*E
                                                                           00268400
      TLN=DLOG (TERM)
                                                                           00268500
      PAINTL=PLN+PRODPL+TLN
                                                                           00268600
C++
                                                                           00268700
C++
                                                                           00268800
C**INTERM IS THE 'UNADJUSTED' TERM WHICH IS GOING TO BE MULTIPLIED BY 00268900
C** PROB. OF GOING OUT OF CONTROL IN THE ITH INTERVAL.
                                                                           00269000
                                                                           00269100
                                                                           00269200
C++NOTE THAT WHEN PAINTL IS LESS THAN -20.(OR-15) THEN INTERM=INTERM+0.000269300
C++OR NOTHING IS ADDED TO INTER. SO, TERMINATE THE LOOP;
                                                                           00269400
C++ THE NUMBER OF ITERATIONS FOR J IS ENOUGH.
                                                                           00269500
C++
                                                                           00269600
        IF(PAINTL.LT.-20.) GO TO 41
                                                                           00269700
        IF (PAINTL.GT.-20.) INTERM=INTERM+DEXP(PAINTL)
                                                                           00269800
C**
                                                                           00269900
C**PNDTCB IS PROB OF NOT DETECTING BEFORE, AND EXCLUDING, THE JTH SAMPLE 00270000
C** TAKEN AFTER THE PROCESS WENT OUT OF CONTROL IN THE ITH INTERVAL.
                                                                           00270100
C^{**} THAT IS PNDTCB = (1-PI)(1-PI+1)...(1-PI+J-1)
                                                                           00270200
```

```
00270300
C**
                                                                         00270400
C
                                                                         00270500
C++PRODPL IS LN OF PNDTCB
                                                                         00270600
C++
                                                                         00270700
C++UPDATE PRODPL
C++.NOTE THAT WHEN PDTC IS ALMOST 1. , THEN THE CORRESPONDING Q IS ZERO.00270800
C++.THAT IS, ALL THE SUBSEQUENT PNDTCB,S (PROB. OF NOT DETECT. BEFORE) 00270900
C++.ARE ZERO. SO, WE SHOULD TERMINATE THE LOOP.
                                                                         00271000
                                                                         00271100
C++
                                                                         00271200
C++WHEN P>Q. , NEED TO UPDATE PNDTC.
C++
                                                                         00271300
                                                                         00271400
C++
         IF(PDTC.LT.0.99999999)PRODPL=PRODPL+DLOG(1.-PDTC)
                                                                         00271500
        IF(PDTC.GT.0.99999999)GO TO 41
                                                                         00271600
C
                                                                         00271700
       GO TO 300
                                                                         00271800
С
                                                                         00271900
                        SPECIFIC FOR WEIBULL
                                                                         00272000
                                                                         00272100
C
C**********
                                                                         00272200
C**CALCULATE AVG. TIME OF OCCURANCE OF THE SHIFT GIVEN THE OCCURANCE
                                                                         00272300
C** IS IN THE ITH INTERVAL ; AT.
                                                                         00272400
C***********
                                                                         00272500
                                                                         00272600
C**GAMPRM IS PARAMETER OF GAMMA DISTRIBUTION
                                                                         00272700
41
                                                                         00272800
       GAMPRM=1.+1./ETA
                                                                         00272900
C**
C**RIA IS THE INTEGRAL OF GAMMA DENSITY FROM 0 TO ALOCU
                                                                         00273000
C**RIB IS THE INTEGRAL OF GAMMA DENSITY FROM 0 TO BLTI
                                                                         00273100
                                                                         00273200
    OCUINT IS T SUB I-1
                                                                         00273300
   TI IS T SUB I
C*
                                                                         00273400
                                                                         00273500
        TI = OCUINT+ITINTR
                                                                         00273600
        ALOCU=(THETA*OCUINT)**ETA
                                                                         00273700
        BLTI = (THETA*TI) **ETA
                                                                         00273800
                                                                         00273900
  MYGAMA IS MY GAMMA SUBROUTINE WHICH CALLS IMSL ROUTINE MDGAMA
                                                                         00274000
   THIS WAS DONE SINCE MDGAM IS NOT A DOUBLE PRECISION ROUTINE.
                                                                         00274100
                                                                         00274200
        CALL MYGAMA (ALOCU, GAMPRM, RIA, IER1)
                                                                         00274300
        CALL MYGAMA (BLTI.GAMPRM.RIB.IER2)
                                                                         00274400
                                                                         00274500
C**RNAT IS THE INCOMPLETE GAMMA INTEGRAL FROM ALOCU TO BLTI
                                                                         00274600
                                                                         00274700
C* DGAMMA IS AN IMSL ROUTINE THAT GIVES THE GAMMA FUNCTION OF A NUMBER
                                                                         00274800
                                                                         00274900
       RNAT=1./THETA*(RIB-RIA)*DGAMMA(GAMPRM)
                                                                         00275000
                                                                         00275100
C**DAT IS EQUAL TO THE AREA UNDER THE WEIBULL DISTRIBUTION FROM
                                                                         00275200
C** T=OCUINT TO T=TI
                                                                         00275300
C**
                                                                         00275400
С
                                                                         00275500
         RDAT=PROOCW(OCUINT, ITINTR, THETA, ETA)
                                                                         00275600
C**
                                                                         00275700
C**NOW CALCULATE AT
                                                                         00275800
C**
                                                                         00275900
          AT=RNAT/RDAT-OCUINT
                                                                         00276000
                                                                         00276100
C
                                                                         00276200
                                                                         00276300
C**CINTRM IS THE 'ADJUSTED' INTERM, THAT IS , THE TIME IN CONTROL IS
                                                                         00276400
C**DEDUCTED FROM INTREM
                                                                         00276500
С
                                                                         00276600
      CINTRM=INTERM-AT
                                                                         00276700
C
                                                                         00276800
```

```
C**OCUINT IS THE OUTER CUMULATIVE INTERVAL , THAT IS , T SUB (I-1)
                                                    00276900
C**PROOCX IS THE PROB. OF GOING OUT OF CONTROL IN THE INTERVAL I.
                                                    00277000
                                                    00277100
    ADBP=ADBP+CINTRM*PROOCW(OCUINT, ITINTR, THETA, ETA)
                                                    00277200
                                                    00277300
C**UPDATE OCUINT FOR THE NEXT ITERATION
                                                    00277400
                                                    00277500
    OCUINT=OCUINT+ITINTR
                                                    00277600
                                                    00277700
200
    CONTINUE
                                                    00277800
                                                    00277900
    RETURN
                                                    00278000
    END
                                                    00278100
C
                                                    00278200
č
                                                    00278300
C
                                                    00278400
Č
                                                    00278500
                                                    00278600
                                           ******* 00278700
SUBROUTINE MYGAMA (A, GPRM, RIA, IER)
                                                    00279000
C**
                                                  * 00279200
C** THIS ROUTINE CALCULATES THE INCOMPLETE GAMMA INTEGRAL.
                                                  * 00279300
C**
                                                  * 00279400
C**
                                                    00279600
С
                                                    00279700
   THIS SUBROUTINE CALLS IMSL ROUTINE MDGAM TO GET THE SINGLE VALUE
С
                                                    00279800
   VARIABLE FOR RIA WHICH IS THEN CONVERTED TO DOUBLE PRECISION
                                                    00279900
                                                    00280000
      DOUBLE PRECISION A, GPRM, RIA
                                                    00280100
                                                    00280200
      SA≠A
      SGPRM=GPRM
                                                    00280300
      CALL MDGAM(SA, SGPRM, SRIA, IER)
                                                    00280400
      RIA=SRIA*1.D0
                                                    00280500
      RETURN
                                                    00280600
      END
                                                    00280700
                                                    00280800
C
                                                    00280900
C
                                                    00281000
C
                                                    00281100
SUBROUTINE CMAINT (BAHCM)
                                                    00281500
C**
                                                   * 00281700
C** THIS ROUTINE CALCULATES THE AVERAGE HOURLY COST OF MAINTAINING
                                                  * 00281800
C** THE CONTROL CHART.
                                                   * 00281900
C**
                                                   * 00282000
C**
                                                    00282200
C*
                                                    00282300
    IMPLICIT REAL*8(A-H,O-Z)
                                                    00282400
    REAL*8 NUM, INTERM, LAMBDA, M, L, ITINTR, IM, ID, IL
                                                    00282500
    COMMON / MAIN1 /LUR,LUW
COMMON / DCDY1 / DELTA, B,C,DD,E,VZMV1,T,W
COMMON / DYNM1 / THETA,ETA
COMMON / CMN1 / CUPROX
                                                    00282600
                                                    00282700
                                                    00282800
                                                    00282900
C**
                                                    00283000
C**TETHA CORRESPONDS TO LAMBDA ( ALPHA) ETA CORRESPONDS TO BETA
                                                    00283100
C**
                                                    00283200
      COMMON/Allll/L,IL,M,IM,D,ID
                                                    00283300
      COMMON/B1111/ISTEPS
                                                    00283400
```

```
COMMON/BE1/ITR
                                                                           00283500
      DIMENSION A(9999), P(9999), AINTR(9999), ASAMP(9999), ACNTL(9999)
                                                                           00283600
      DIMENSION S(260), AS(260)
                                                                           00283700
                                                                           00283800
C INITIALIZE THE ARRAYS
                                                                           00283900
                                                                           00284000
      DATA A/9999*0.0D0/
                                                                           00284100
      DATA P/9999*0.0D0/
                                                                           00284200
      DATA AINTR/9999*0.0D0/
                                                                           00284300
      DATA ASAMP/9999*0.0D0/
                                                                           00284400
      DATA ACNTL/9999*0.0D0/
                                                                           00284500
CC
                                                                           00284600
C**
                                                                           00284700
C**IN ORDER TO MAKE CALCULATIONS EFFICIENT AND FAST THE
                                                                           00284800
C**COST VALUES ARE FIRST CALCULATED AND STORED IN ARRAY A AND
                                                                           00284900
C**THEN UTILIZED.
                                                                           00285000
C**SIMILARLY PROB OF CATCHING THE SHIFT ON ANY OF THE SAMPLES
                                                                           00285100
C**ARE FIRST CALCULATED AND STORED IN ARRAY P AND THEN UTILIZED.
                                                                           00285200
C**
                                                                           00285300
                                                                           00285400
                                                                           00285500
C**IFLAG IS USED TO SPECIFY IF SAMPLING SHOULD BE DONE DURING
                                                                           00285600
C** THE SEARCH FOR THE ASSIGNABLE CAUSE OR NOT.
                                                                           00285700
C** IFLAG=1 ; TAKE SAMPLES; LIKE DUNCAN'S MODEL
                                                                           00285800
C** IFLAG=0 ; DONOT TAKE SAMPLES DURING SEARCH ....
                                                                           00285900
C**
                                                                           00286000
                                                                           00286100
                                                                           00286200
C**DELTA IS AMOUNT OF SHIFT IN THE PROCESS ( DELTA IS ASSUMED TO BE
                                                                           00286300
C**POSITIVE. NOTE THAT PROB. CALCULATIONS NEED SOME MODIFICATIONS
                                                                           00286400
C** IF DELTA IS NEGATIVE)
                                                                           00286500
C**
                                                                           00286600
C**
                                                                           00286700
       ITINTR=IL/L
                                                                           00286800
       CNTLMT=IM/M
                                                                           00286900
       SAMPLI=ID/D
                                                                           00287000
C**
                                                                           00287100
C**FIRST CALCULATE ALL ITINTR,S; H SUB I,S
                                                                           00287200
C**
                         CNTLMT,S; K SUB I,S
SAMPLI,S; N SUB I,S, AND STORE THEM IN
                                                                           00287300
C**
                                                                           00287400
C** ARRAYS AINTR, ACNTL, AND ASAMP.
                                                                           00287500
C**
                                                                           00287600
       ILT=ITR+200
                                                                           00287700
C
                                                                           00287800
       DO 11 I=1,ILT
                                                                           00287900
C**
                                                                           00288000
C**ITINTR IS THE LENGTH OF THE ITH INTERVAL; H SUB I
                                                                           00288100
C**CNTLMT IS THE WIDTH OF THE ITH CONTROL LIMITS ; K SUB I
                                                                           00288200
C**SAMPLI IS THE SIZE OF ITH SAMPLE ;N SUB I
                                                                           00288300
                                                                           00288400
             ITINTR=ITINTR*L
                                                                           00288500
             AINTR(!)=ITINTR
                                                                           00288600
             CNTLMT=CNTLMT*M
                                                                           00288700
C+=+=+
                                                                           00288800
C+=+CONTROL LIMIT SPREAD STRICTED BETWEEN .5 AND 5.5
                                                                           00288900
C+=+=+
                                                                           00289000
             IF(CNTLMT.LT.0.5)CNTLMT=0.5
                                                                           00289100
             IF(CNTLMT.GT.5.5)CNTLMT=5.5
                                                                           00289200
C+=+=+
                                                                           00289300
            ACNTL(I)=CNTLMT
                                                                           00289400
C++
                                                                           00289500
C++SAMPLE SIZE SHOULD BE INTEGER. E.G. IF SAMPLE SIZE IS BETWEEN 5.5
                                                                           00289600
C++TO 6., THEN IT IS SET TO 6.
                                                                           00289700
                                                                           00289800
             SAMPLI = SAMPLI *D
                                                                           00289900
             ISAMPL=SAMPLI+.4999999D0
                                                                           00290000
```

```
IF(ISAMPL.LT.2)ISAMPL=2
                                                                                       00290100
C+=+=+=+
                                                                                       00290200
C+=UPPER LIMIT FOR SAMPLE SIZE IS 1000.
                                                                                       00290300
                                                                                       00290400
               IF(ISAMPL.GT.1000)ISAMPL=1000
                                                                                       00290500
C+=+=+=
                                                                                       00290600
               ASAMP(I)=ISAMPL
                                                                                       00290700
C++
                                                                                       00290800
C++
                                                                                       00290900
            Y=CNTLMT-DELTA*DSQRT(ASAMP(I))
                                                                                       00291000
             CALL MDNORD(Y,XX)
                                                                                       00291100
C**
                                                                                       00291200
C**P(I) IS THE PROB. OF DETECTING THE SHIFT ON THE ITH SAMPLE
                                                                                       00291300
C**NOTE THAT IN THE CALCULATION OF THIS PROBABILITY IT IS ASSUMED THAT
                                                                                       00291400
C***DELTA IS POSITIVE.
                                                                                       00291500
C**
                                                                                       00291600
              P(I)=1.-XX
                                                                                       00291700
                                                                                       00291800
С
 11
        CONTINUE
                                                                                       00291900
C
                                                                                       00292000
         PANUM=0.
                                                                                       00292100
         NUM=0.
                                                                                       00292200
         PADENM=0.
                                                                                       00292300
         DENM=0.
                                                                                       00292400
C**NOTE THAT ILT2=ISTEPS+113
                                                                                       00292500
          ILT2=ITR+100
                                                                                       00292600
                                                                                       00292700
                                                                                       00292800
         DO 12 I=1,ILT2
C**PANUM IS THE VALUE OF NUMERATOR IF IFLAG=0. IT IS ALSO NEEDED IN
                                                                                       00292900
C** CALCULATION OF NUM FOR THE CASE OF IFLAG=1.
C**NUM IS THE VALUE OF NUMERATOR OF THE COST AT THE ITH ITERATION
C**DENUM IS THE VALUE OF DENUMINATOR OF THE COST AT THE ITH ITERATION
                                                                                       00293000
                                                                                       00293100
                                                                                       00293200
C**
                                                                                       00293300
          PANUM=PANUM+B+C*ASAMP(I)
                                                                                       00293400
           IF(IFLAG.NE.1) GO TO 124
                                                                                       00293500
C**
                                                                                       00293600
C**
                                                                                       00293700
C**COST OF SAMPLING DURING THE SEARCH FOR THE ASSIGNABLE CAUSE
                                                                                       00293800
C** SININT IS THE SUM OF INTERNAL INTERVALS(USED IN THE UPPER LIMIT C** OF SUMMATION USED IN CALCULATION OF NUMERATOR)
                                                                                       00293900
                                                                                       00294000
C**
                                                                                       00294100
C**FIRST SET NUMERATOR EQUAL TO PANUM
                                                                                       00294200
          NUM=PANUM
                                                                                       00294300
           SININT=AINTR(I+1)
                                                                                       00294400
           TPFAC=E*ASAMP(I)+DD
                                                                                       00294500
              J = I
                                                                                       00294600
           IF(SININT.GT.TPFAC)GO TO 123
                                                                                       00294700
  122
               J=I+1
                                                                                       00294800
               SININT=SININT+AINTR(J+1)
                                                                                       00294900
С
                                                                                       00295000
C**
                                                                                       00295100
C***NOTE THAT WHEN IFLAG=1 THEN IF INTERVAL SIZE, AINTR, GETS SMALLER
                                                                                       00295200
C** THAN .05*IL WE FIX THE INTERVAL SIZE AT THAT VALUE ;.05*IL, IN 00295300 C** ORDER TO CALCULATE THE REMAINING NUMBER OF SAMPLES TAKEN DURING THE 00295400 C** SEARCH FOR THE ASSIGNABLE CAUSE; THAT IS DURING E*ASAMP(I)+DD OR 00295500
C** E*ASAMP(I)+DD SBINT.
                                                                                       00295600
C**
                                                                                       00295700
C***ALSO, IN ORDER TO CALCULATE THE DENOMINATOR (WHETHER IFLAG=0 OR 1
                                                                                       00295800
C** IF AINTR GETS SMALLER THAN .05*IL THEN THE INTERVAL SIZE IS FIXED
                                                                                       00295900
C** AT THAT VALUE; .05*IL.
                                                                                       00296000
C**
                                                                                       00296100
                        IF(AINTR(J+1).LT. .05*IL)GO TO 1221
                                                                                       00296200
               NUM=NUM+B+C*ASAMP(J)
                                                                                       00296300
           IF(SININT.LE.TPFAC)GO TO 122
                                                                                       00296400
           GO TO 123
                                                                                       00296500
C**
                                                                                       00296600
```

```
C**FIND THE INTEGER NUMBER OF THE REMAINING INTERVALS
                                                                               00296700
C**
                                                                               00296800
1221
             SBINT=SININT-AINTR(J+1)
                                                                               00296900
             IREMN=(E*ASAMP(I)+DD-SBINT)/(.05*IL)
                                                                               00297000
                                                                               00297100
C*SO WE GET TO SININT=SININT+IREMN*(.05)*IL
                                                                               00297200
              NUM=NUM+IREMN*(B+C*ASAMP(J))
                                                                               00297300
C**DONOT INTERPOLATE FOR THIS CASE.
                                                                               00297400
             GO TO 124
                                                                               00297500
                                                                               00297600
C**DENM AT THIS POINT IS TI
                                                                               00297700
C** .
                                                                               00297800
C**INTERPOLATE FOR THE COST
                                                                               00297900
C**
                                                                               00298000
 123
           SBINT=SININT-AINTR(J+1)
                                                                               00298100
          NUM=NUM+(TPFAC-SBINT)/AINTR(J+1)*(B+C*ASAMP(J+1))
                                                                               00298200
                                                                               00298300
C** SET THE INTERVALS EQUAL TO .05*IL IF THEY ARE < OR = TO .05*IL
                                                                               00298400
C**
                                                                               00298500
124
          IF(AINTR(I).LE..O5*IL)AINTR(I)=.O5*IL
                                                                               00298600
      PADENM=PADENM+AINTR(I)
                                                                               00298700
          IF(IFLAG.EQ.0)NUM=PANUM
                                                                               00298800
         DENM=PADENM+E*ASAMP(I)+DD
                                                                               00298900
C
                                                                               00299000
          A(I)=NUM/DENM
                                                                               00299100
С
                                                                               00299200
 12
          CONTINUE
                                                                               00299300
C**
                                                                               00299400
C**CALCULATION OF BAHCM
                                                                               00299500
                                                                               00299600
      BAHCM=0.
                                                                               00299700
      FRSTIN=0.
                                                                               00299800
      OCUINT=0.
                                                                               00299900
      ITINTR=IL/L
                                                                               00300000
      ILT3=ILT2-113
                                                                               00300100
C**NOTE THAT ILT3=ISTEPS
                                                                               00300200
      I = 0
                                                                               00300300
      CUPROX=0.
                                                                               00300400
С
                                                                               00300500
 35
      I = I + 1
                                                                               00300600
С
                                                                               00300700
      PRODP=1.
                                                                               00300800
      PRODPL=0.
                                                                               00300900
      INTERM=0.
                                                                               00301000
C**
                                                                               00301100
C**ITINTR IS THE LENGTH OF THE ITH INTERVAL ; H SUB I
                                                                               00301200
C**
                                                                               00301300
       ITINTR=ITINTR*L
                                                                               00301400
       J=0
                                                                               00301500
C----
                                                                               00301600
40
         J=J+1
                                                                               00301700
C**
                                                                               00301800
C**PRODP IS PRODUCT OF PROBABILITIES AND IS EQUAL TO:

C** Q SUB I*Q SUB (I+1)*...*Q SUB(J-1), WHERE Q SUB I IS (1-P SUB I)

C**INTERM IS THE INTERMEDIATE TERM WHICH IS TO BE MULTIPLIED BY

00302100
C**PROB. OF OUT OF CONTROL. IN OTHER WORDS INTERM IS THE COST OF
                                                                               00302200
C**MAINTAINING THE CHART GIVEN THAT THE PROCESS GOES OUT OF CONTROL IN 00302300
C**THE ITH INTERVAL.
                                                                               00302400
C**
                                                                               00302500
                                                                               00302600
C**PRODPL IS LN OF PRODP
                                                                               00302700
C**ALN IS LN OF A(I+J-1)
                                                                               00302800
C**PLN IS LN OF P(I+J-1)
                                                                               00302900
                                                                               00303000
C+
                                                                               00303100
               IF(I+J.GE.ITR+99)GO TO 401
                                                                               00303200
```

```
00303300
C+
              ALN=DLOG(A(I+J-1))
                                                                      00303400
              PLN=DLOG(P(I+J-1))
                                                                      00303500
              PAINTL=PRODPL+PLN+ALN
                                                                      00303600
C**
                                                                      00303700
C**NOTE THAT WHEN PAINTL IS LESS THAN -20. THEN INTERM=INTERM+0.
                                                                      00303800
C++
      OR NOTHING IS ADDED TO INTERM. SO, TERMINATE THE LOOP; THE
                                                                      00303900
      NUMBER OF ITERATIONS FOR J IS ENOUGH.
C++
                                                                      00304000
C++
                                                                      00304100
             IF(PAINTL.LT.-20.) GO TO 41
                                                                      00304200
             IF(PAINTL.GT.-20.)INTERM=INTERM+DEXP(PAINTL)
                                                                      00304300
C**
                                                                      00304400
 C**UPDATE PRODPL FOR THE NEXT ITERATION
                                                                      00304500
C**
                                                                      00304600
C++NOTE THAT WHEN P(I+J-1) IS ALMOST =1. THEN Q(I+J-1) IS ZERO AND C++ALL SUBSEQUENT PRODP( PRODUCT OF PROB.S OR PROB. OF NOT DET.
                                                                      00304700
                                                                      00304800
 C++BEFORE) ARE ZERO. SO, NOTHING WILL BE ADDED RO INTERM. THERFORE,
                                                                      00304900
C++TERMINATE THE LOOP.
                                                                      00305000
C++
                                                                      00305100
             IF(P(I+J-1).LT..999999) PRODPL=PRODPL+DLOG(1.-P(I+J-1))
IF(P(I+J-1).GT..999999) GO TO 41
                                                                      00305200
                                                                      00305300
C
                                                                      00305400
      GO TO 40
                                                                      00305500
 C--
                                                                      00305600
С
                                                                      00305700
 C++
                                                                      00305800
 401
              WRITE(6,4031)I+J-1
                                                                      00305900
              FORMAT(1x,'VALUE OF A(',14,') IS NOT DEFINED . ITERATIONS'00306000
        , 'TERMINATED')
                                                                      00306100
 C++
                                                                      00306200
C**
                                                                      00306300
C**FRSTIN IS THE COST OF TAKING THE FIRST (I-1) SAMPLES (, GIVEN C**THAT THE PROCESS GOES OUT OF CONTROL IN THE ITH INTERVAL.)
                                                                      00306400
                                                                      00306500
 C**
                                                                      00306600
             PROX=PROOCW(OCUINT, ITINTR, THETA, ETA)
 41
                                                                      00306700
           BAHCM=BAHCM+INTERM*PROX
                                                                      00306800
 C**CUPROX IS CUMULATIVE SUM OF PROBABILITIES OF OUT OF
                                                                      00306900
 C**CONTROL USED FOR TERMINATING THE
                                                                      00307000
         CUPROX=CUPROX+PROX
                                                                      00307100
 C**
                                                                      00307200
 C**UPDATE OCUINT FOR THE NEXT ITERATION. OCUINT IS THE CUMULATIVE
                                                                      00307300
 C**SUM OF INTERVALS; OCUINT USED IN PROOCX, ABOVE, IS T SUB (I-1).
                                                                      00307400
C**
                                                                      00307500
 667
            OCUINT=OCUINT+ITINTR
                                                                      00307600
            IF(I.LT.ILT3) GO TO 35
                                                                      00307700
                                                                      00307800
  669
     RETURN
                                                                      00307900
      END
                                                                      00308000
 C
                                                                      00308100
 C
                                                                      00308200
 C
                                                                      00308300
 00308400
 C***********************************
                                                                      00308500
 00308600
      DOUBLE PRECISION FUNCTION PROOCW(OCUINT, ITINTR, LAMBDA, BETA)
                                                                      00308700
 00308800
: C**
                                                                      00308900
 C** THIS ROUTINE CALCULATE THE AREA UNDER A WEIBULL DENSITY
                                                                  *
                                                                      00309000
 C** BETWEEN OCUINT AND ITINTR.
                                                                      00309100
 C**
                                                                  *
                                                                      00309200
 C**
                                                                      00309300
 00309400
 C**
                                                                      00309500
 C*
                                                                      00309600
      IMPLICIT REAL*8(A-H.O-Z)
                                                                      00309700
       REAL*8 LAMBDA, ITINTR
                                                                      00309800
```

```
00309900
C**LAMBDA CORRESPONDS TO THETA BETA CORRESPONDS TO ETA.
                                                                      00310000
                                                                      00310100
C**OCUINT IS T SUB I-1
                                                                      00310200
                                                                      00310300
C** LET TI BE T SUB I
                                                                      00310400
                                                                      00310500
                                                                      00310600
     TI=OCUINT+ITINTR
                                                                      00310700
C**FIRST CHECK FOR UNDERFLOW
                                                                      00310800
                                                                      00310900
     DUM1 = - (LAMBDA * OCUINT) * * BETA
                                                                      00311000
     DUM2=~(LAMBDA*TI)**BETA
                                                                      00311100
                                                                      00311200
C**NOTE THAT DUM1=-LAMBDA*T SUB (I-1) AND
                                                                      00311300
C** DUM2=-LAMBDA*T SUB I , SO |DUM2| .GT. |DUM1|
C** THAT IS IF DUM1 IS .LT. -70 , THEN DUM2 IS .LT. -70
                                                                      00311400
                                                                      00311500
                                                                      00311600
      IF (DUM2.LT.-70.)PROOCW=0.
                                                                      00311700
      IF(DUM2.LT.-70.)RETURN
                                                                      00311800
C
                                                                      00311900
      PROOCW=DEXP(DUM1)-DEXP(DUM2)
                                                                      00312100
C
                                                                      00312200
      RETURN
                                                                      00312700
      END
                                                                       00312800
C
                                                                       00312900
С
                                                                       00313000
С
                                                                       00313700
                                                                       00313800
C************************
                                                                       00313810
C**********************
                                                                       00313820
C************************
                                                                       00313830
    SUBROUTINE FALSA (ENFALS)
                                                                       00313841
C******************
                                                                       00313850
C**
                                                                       00313860
C** THIS ROUTINE CALCULATE THE EXPECTED NUMBER OF FALSE ALARMS.
                                                                       00313870
C**
                                                                       00313890
C**
                                                                       00313891
C********************
                                                                       00313892
C**
                                                                       00313893
C*
                                                                       00313894
                                                                       00313900
      IMPLICIT REAL*8(A-H,O-Z)
                                                                       00314400
      REAL*8 NUM, INTERM, LAMBDA, M, L, ITINTR, IM, ID, IL
                                                                       00314500
COMMON / MAIN1 /LUR, LUW
COMMON / DCDY1 / DELTA, B,C,DD,E,VZMV1,T,W
COMMON / DYNM1 / THETA,ETA

C**THETA CORRESPONDS TO LAMBDA (ALPHA) ETA CORRESPONDS TO BETA
                                                                       00314600
                                                                       00314700
                                                                       00314800
                                                                       00314900
         COMMON/Allll/L,IL,M,IM,D,ID
                                                                       00315000
         COMMON/B1111/ISTEPS
                                                                       00315100
         COMMON/BEL/ITR
                                                                       00315200
C**
                                                                       00315300
      ENFALS=0.
                                                                       00316200
      OCUINT=0.
                                                                       00316300
      CNTLMT=IM/M
                                                                       00316400
      CUALPH=0.
                                                                       00316500
      ITINTR=IL
                                                                       00316600
C**
                                                                       00316700
C**NOTE THAT LT IS ISTEPS
                                                                       00316800
C**
                                                                       00316810
      ILT=ITR-13
                                                                       00316900
С
                                                                       00317000
      DO 20 I=1,ILT
                                                                       00317100
C**
                                                                       00317200
C**MDNOR IS AN IMSL ROUTINE THAT CALCULATES NORMAL DENSITY INTEGRAL.
                                                                       00317300
                                                                       00317400
```

```
C**
                                                                     00317500
C**CNTLMT IS THE WIDTH OF ITH CONTROL LIMIT
                                                                     00317600
C**
                                                                     00317700
       CNTLMT=CNTLMT*M
                                                                     00317800
                                                                     00317900
C+=+CONTROL LIMIT SPREAD RESTRICTED BETWEEN .5 AND 5.5
                                                                     00318000
                                                                     00318100
       IF (CNTLMT.LT.0.5) CNTLMT=0.5
                                                                     00318200
       IF(CNTLMT.GT.5.5)CNTLMT=5.5
                                                                     00318300
                                                                     00318400
C+=+=+
       CALL MDNORD (CNTLMT, XX)
                                                                     00318500
C**
                                                                     00318600
C**ALPHA IS THE PROB. OF A FALSE ALARM IN THE ITH INTERVAL
                                                                     00318700
C**
                                                                     00318800
       ALPHAI = 2.*(1.-XX)
                                                                     00318900
                                                                     00319000
C**CUALPH IS CUMULATIVE SUM OF ALPHAS; ALPHA1+ ALPHA2+ ...+ ALPHA(I-1)
                                                                     00319100
C**
                                                                     00319200
                                                                     00319300
       CUALPH=CUALPH+ALPHAI
C**
                                                                     00319400
C**ITINTR IS THE LENGTH OF ITH INTERVAL ; H SUB I
                                                                     00319500
C**
                                                                     00319600
C**OCUINT IS CUMULATIVE SUM OF INTERVALS; T SUB I
                                                                     00319700
C** OR IT IS T SUB (I-1) FOR FOR PROB. OF GOING OUT OF CONTROL IN
                                                                     00319800
C** (I+1)TH INTERVAL.
                                                                     00319900
C**
                                                                     00320000
         OCUINT=OCUINT+ITINTR
                                                                     00320100
         ENFALS=ENFALS+CUALPH*PROOCW(OCUINT, ITINTR, THETA, ETA)
                                                                     00320200
C**
                                                                     00320300
C**UPDATE ITINTR FOR THE NEXT ITERATION
                                                                     00320400
                                                                     00320500
        ITINTR=ITINTR*L
                                                                     00320600
C++
                                                                     00320700
                                                                     00320800
        IF(ITINTR.LT..05*IL)ITINTR=.05*IL
                                                                     00320900
                                                                     00321500
  20
      CONTINUE
C**
                                                                     00321600
      RETURN
                                                                     00321900
       END
                                                                     00322000
С
                                                                     00322100
C
                                                                     00322200
C
                                                                     00322300
                                                                     00322400
                                                                     00322500
                                                                     00322700
C***********************
                                                                     00322800
                                                                     00322900
      DOUBLE PRECISION FUNCTION SINTW(HF, ISTEPS)
                                                                     00323100
00323200
C**
                                                                     00323300
C**THIS ROUTINE CALCULATES THE VALUE OF IH=SINTW (INITIAL SAMPLING *
                                                                     00323400
C**INTERVAL) FOR ANY GIVEN VALUES OF HF, ISTEPS, PROBPT, AND WEIBULL*
                                                                     00323500
C**DISTRIBUTION PARAMETERS.
                                                                     00323600
C**
                                                                     00323700
C***********************************
                                                                     00323800
С
                                                                     00323900
       IMPLICIT REAL*8(A-H,O-Z)
                                                                     00324000
       REAL*8 IH, HF
                                                                     00324100
      COMMON / DYNM4 / PROBPT
                                                                     00324200
      COMMON / DYNM1 / THETA, ETA
                                                                     00324300
C**
                                                                     00324400
C**CALCULATE IL ; INITIAL INTERVAL SIZE TO ACHIEVE THE QUANTILE
                                                                     00324500
C**OF PROBPT, SPECIFIED BY USER, IN ISTEPS ITERATIONS
                                                                     00324600
C**
                                                                     00324700
                                                                     00324800
C**
                                                                     00324900
```

```
C**FIND THE VALUE THAT THE SUM OF ISTEPS INTERVALS SHOULD BE EQUAL TO.
                                                                 00325000
C**
                                                                 00325100
      SUMATN=DEXP(DLOG(-DLOG(1.D0-PROBPT))/ETA-DLOG(THETA))
                                                                 00325200
C**
                                                                 00325300
C**CALCULATION OF IH=SINTW, IF HF IS 1.
                                                                 00325400
C**
                                                                 00325500
      IF(HF.EQ.1)SINTW=SUMATN/DFLOAT(ISTEPS)
                                                                 00325600
      IF(HF.EO.1)RETURN
                                                                 00325700
C**
                                                                 00325800
C**CALCULATION OF IH=SINTW, IF HF IS NOT EQUAL TO 1.
                                                                 00325900
C**
                                                                 00326000
      BX=ISTEPS*DLOG(HF)
                                                                 00326100
      IH=SUMATN*(1.D0-HF)/(1.D0-DEXP(BX))
                                                                 00326200
      SINTW=IH
                                                                 00326300
                                                                 00326400
      RETURN
      END
                                                                 00326500
C
                                                                 00326600
C
                                                                 00326700
C
                                                                 00326800
C*******************
                                                                 00326900
C**********************************
                                                                 00327000
      DOUBLE PRECISION FUNCTION SINTX(HF, ISTEPS)
                                                                 00327100
00327200
C**
                                                                 00327300
C**THIS ROUTINE CALCULATES THE VALUE OF IH=SINTW (INITIAL SAMPLING
                                                                 00327400
C**INTERVAL) FOR ANY GIVEN VALUES OF HF, ISTEPS, PROBPT, AND
                                                                 00327500
C**EXPONENTIAL DISTRIBUTION PARAMETER.
                                                             *
                                                                 00327600
C**
                                                                 00327700
<u>C</u>**********************
                                                                 00327800
С
                                                                 00327900
      IMPLICIT REAL*8(A-H,O-Z)
                                                                 00328000
      REAL*8 IH, HF
                                                                 00328100
      REAL*8 LAMBDA
                                                                 00328200
     COMMON / DUNC1 / LAMBDA
COMMON / DYNM4 / PROBPT
                                                                 00328300
                                                                 00328400
C**
                                                                 00328500
C**CALCULATE IL ; INITIAL INTERVAL SIZE TO ACHIEVE THE QUANTILE
                                                                 00328600
C**OF PROBPT, SPECIFIED BY USER, IN ISTEPS ITERATIONS
                                                                 00328700
C**
                                                                 00328800
С
                                                                 00328900
C**
                                                                 00329000
C**FIND THE VALUE THAT THE SUM OF ISTEPS INTERVALS SHOULD BE EQUAL TO.
                                                                 00329100
C**
                                                                 00329200
      SUMATN=1.D0/LAMBDA*(-DLOG(1.D0-PROBPT))
                                                                 00329300
C**
                                                                 00329400
C**CALCULATION OF IH=SINTX, IF HF IS 1.
                                                                 00329500
C**
                                                                 00329600
      IF(HF.EQ.1)SINTX=SUMATN/DFLOAT(ISTEPS)
                                                                 00329700
      IF (HF.EQ.1) RETURN
                                                                 00329800
C**
                                                                 00329900
C**CALCULATION OF IH=SINTX, IF HF IS NOT EQUAL TO 1.
                                                                 00330000
C**
                                                                 00330100
      BX=ISTEPS*DLOG(HF)
                                                                 00330200
      IH=SUMATN*(1.D0-HF)/(1.D0-DEXP(BX))
                                                                 00330300
      SINTX=IH
                                                                 00330400
      RETURN
                                                                 00330500
      END
                                                                 00330600
C
                                                                 00330700
CCC
                                                                 00330800
                                                                 00330900
                                                                 00331000
С
                                                                 00331200
C
                                                                 00331300
                                                                 00331400
00331500
C**********************************
                                                                 00331600
```

```
00331700
      DOUBLE PRECISION FUNCTION CSTPW(HF, IH)
                                                            00331800
00331900
C**
                                                            00332000
C**THIS ROUTINE CALCULATES THE VALUE OF ISTEPS=CSTPW (NUMBER OF STEPS*
                                                            00332100
C**TO GET TO A GIVEN QUANTILE POINT, PROBPT) FOR GIVEN VALUES OF
                                                            00332200
                                                          •
C**HF, IH, PROBPT, AND WEIBULL DISTRIBUTION PARAMETERS.
                                                             00332300
C**
                                                             00332400
C**********************
                                                             00332500
С
                                                             00332600
      IMPLICIT REAL*8(A-H.O-Z)
                                                             00332700
      REAL*8 IH, HF
                                                             00332800
     COMMON / DYNM4 / PROBPT
                                                             00332900
     COMMON / DYNM1 / THETA, ETA
                                                             00333000
C**
                                                             00333100
C**CALCULATE IL ; INITIAL INTERVAL SIZE TO ACHIEVE THE QUANTILE
                                                             00333200
C**OF PROBPT, SPECIFIED BY USER, IN ISTEPS ITERATIONS
                                                             00333300
C**
                                                             00333400
                                                             00333500
С
C**
                                                             00333600
'C**FIND THE VALUE THAT THE SUM OF ISTEPS INTERVALS SHOULD BE EQUAL TO.
                                                             00333700
                                                             00333800
      SUMATN=DEXP(DLOG(-DLOG(1.D0-PROBPT))/ETA-DLOG(THETA))
                                                             00333900
C**
                                                             00334000
C**CALCULATION OF IH=SINTW, IF HF IS 1.
                                                             00334100
C**
                                                             00334200
      IF (HF.EQ.1) ISTEPS = SUMATN/IH
                                                             00334900
      IF(HF.EQ.1)CSTPW=ISTEPS
                                                             00335000
      IF (HF.EQ.1) RETURN
                                                             00335100
C**
                                                             00335200
C**CALCULATION OF IH=SINTW, IF HF IS NOT EQUAL TO 1.
                                                             00335300
C**
                                                             00335400
      AX=SUMATN*(1.D0-HF)/IH
                                                             00335500
      ISTEPS=DLOG(1.D0-AX)/DLOG(HF)
                                                             00335600
                                                             00335700
      CSTPW=ISTEPS
      RETURN
                                                             00335800
      END
                                                             00335900
С
                                                             00336000
С
                                                             00336100
C
                                                             00336110
                                                             00336120
C***
                                                             00336130
C********************
                                                             00336140
00336150
     DOUBLE PRECISION FUNCTION FCTR(DORM, DORME)
                                                             00336200
00336300
C**
                                                             00336400
C**THIS ROUTINE CALCULATES THE VALUE OF D OR M FOR ANY GIVEN VALUES *
                                                             00336500
C** OF DORM ( ID OR IM ) AND DORME (IDENDING OR IM ENDING )
                                                             00336600
C**
                                                             00336700
C**
                                                             00336800
00336900
С
                                                             00337000
      IMPLICIT REAL*8(A-H,O-Z)
                                                             00337100
     COMMON / DYNM2 / ISTEPS
                                                             00337200
                                                             00337300
٠č
                                                             00337400
      QDORM=DORME/DORM
                                                             00337500
С
                                                             00337600
      QISTM1=ISTEPS-1
                                                             00337700
C
                                                             00337800
      FCTR=DEXP(DLOG(QDORM)/OISTM1)
                                                             00337900
C
                                                             00338000
      RETURN
                                                             00338100
      END
                                                             00338200
C
                                                             00338300
```

C	00338400
C	00338500
c	00338600
C	00338800
C***********************	00338900
C*************************************	00339000
C*************************************	00339100
DOUBLE PRECISION FUNCTION SETDEL(DELIME)	00339400
C*********************	00339500
C**	00339600
C**THIS ROUTINE CALCULATES THE VALUE OF DEL FOR KF (NF AND HF) GIVEN*	00339700
C**THE DESIRED INCREMENT IN IM ENDING OR ID ENDING; DELIME. *	00339800
C**	00339900
C**INITIAL VALUE OF KF IS ASSUMED TO BE 1. HOWEVER , THE RESULTS *	00340000
C** SHOULD BE QUITE GOOD EVEN WHEN KF IS NOT 1 BUT IN THE RANGE OF *	00340100
C** .99 TO 1.1 . *	00340200
C**	00340300
C*****************	00340400
C	00340500
IMPLICIT REAL*8(A-H,O-Z)	00340600
COMMON / DYNM2 / ISTEPS	00340700
C	00340800
C	00340900
DELIME=DABS(DELIME)	00341000
C**	00341100
C**FCTIME IS THE MULTIPLE OF IM ENDING	00341200
C**	00341300
FCTIME=1.D0+DELIME	00341400
C**	00341500 00341600
QISTM1=ISTEPS-1 C**	00341800
C**RNEWM IS THE NEW VALUE OF M	00341700
C**	00341800
RNEWM=DEXP(DLOG(FCTIME)/OISTMl)	00341900
C**	00342000
C**SETDEL IS THE INCREMENT IN KF OR NF.	00342100
C**	00342200
SETDEL=RNEWM-1.DO	00342400
C SEIDED-RNEWWII.DO	00342500
RETURN	00342600
END	00342700
C	00342700
Č	00342900
~	. 55542500

ATIV

Behrooz Parkhideh

Candidate for the Degree of

Doctor of Philosophy

Thesis: THE ECONOMIC DESIGN OF DYNAMIC \bar{X} -CONTROL CHARTS

Major Fields: Industrial Engineering and Management

Biographical:

Personal Data: Born in Tehran, Iran, January 4, 1957, the son of Tirandaz and Delbar Parkhideh.

Education: Graduated from Firooz-Bahraam High School, Tehran, in June 1973; received the Bachelor of Science degree in Industrial Engineering from Iran College of Science and Technology, Tehran, Iran, in June 1977; received the Master of Science degree in Industrial Engineering and Management from Oklahoma State University in December 1980; completed requirements for the Doctor of Philosophy degree at Oklahoma State University in May, 1984.

Professional Experience: Mathematics teacher, Rahnama Public High School, Tehran, 1976-1977; Transportation System Analyst, Butane Gas Distribution Co., Tehran, Summer 1977; Research Associate, OSU Institute for Energy Analysis, 1981-1982; Teaching and Research Assistant, Oklahoma State University, 1979-1983.