THE ECONOMIC DESIGN OF DYNAMIC

$\bar{X}$-CONTROL CHARTS

By

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
1984D
P246e
cop.2

OKLAHOMA STATE UNIVERSITY LIBRARY

THE ECONOMIC DESIGN OF DYNAMIC

$\bar{X}$-CONTROL CHARTS

Thesis Approved:

Kenneth E Case

Thesis Adviser

C. Patrick Koelling

M. Palmer Terrell

Don Holbert

Norman N Durham

Dean of the Graduate College

1200623

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 orignate 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

LIST OF TABLES

LIST OF FIGURES

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,
3. 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.

$\bar{X}$-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.

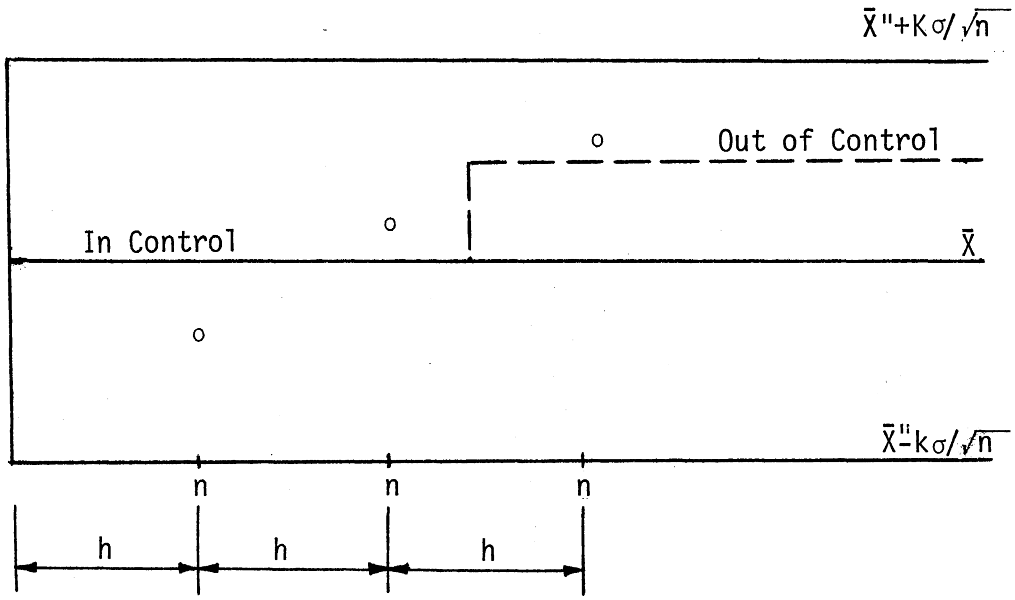
Among many different control charts and procedures developed for monitoring of a process, the $\bar{X}$-control chart for averages is the most widely used technique [35]. A scientific survey of many firms in the United States in 1976 shows that the use of $\bar{X}$-control charts dominates the use of 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 $\bar{X}$-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 $k\sigma_{\bar{X}}$ 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 ($\pm 3\sigma_{\bar{X}}$). 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 $\bar{X}$-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 in-control 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.

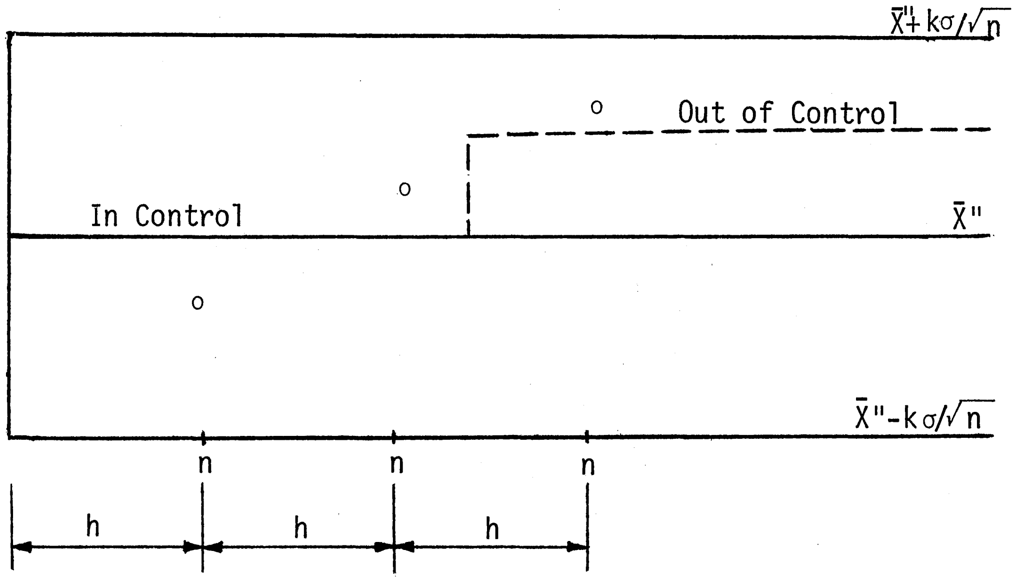
Dynamic** $\bar{X}$-Control Chart

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.

**"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.



n, h, and 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.

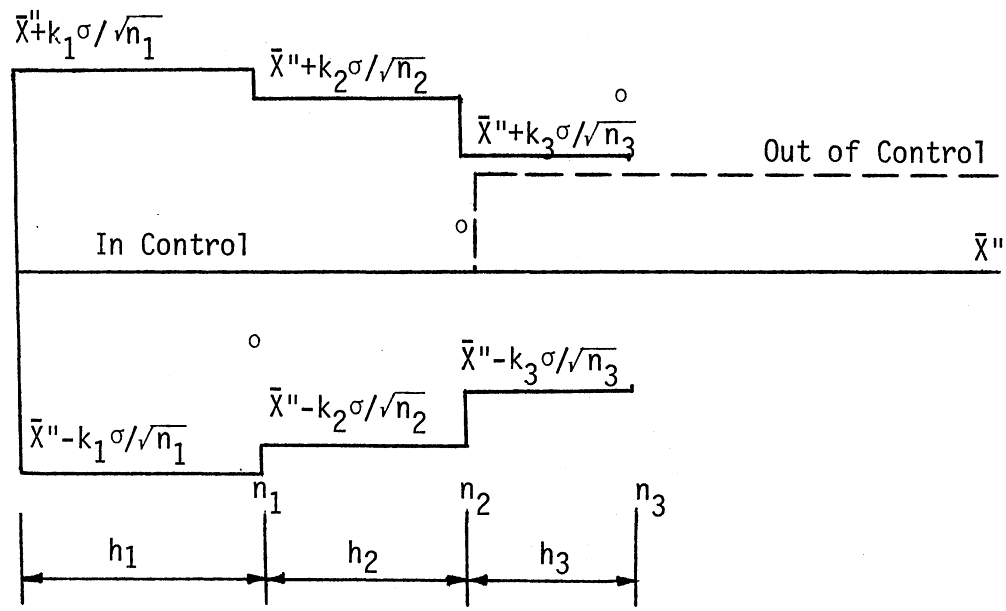
A new control chart methodology in which the control chart parameters n_i, h_i, and k_i are dynamic might be needed in the optimal design of control chart models which better reflect reality. That is, if some of the simplistic assumptions used in the classical economic design of $\bar{X}$-control chart 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.



n_i, h_i, and 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 $\bar{X}$-Control Chart

Introduction and Importance

Fixed sample sizes and intervals between samples are used in the optimal economic designs of the $\bar{X}$-control chart. In Duncan's economic design of the $\bar{X}$-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 $\bar{X}$-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:

Objective:

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.

Subobjectives:

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--$\bar{X}$; 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 T^2; 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 $\bar{X}$-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 $\bar{X}$-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 in-control 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 i^{th} sample; that is; the i^{th} sample size.

h_i - length of the i^{th} interval; the interval between the (i-1)th and i^{th} 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; $t_i = \sum_{j=1}^{i} h_j$.

θ - the scale parameter of a Weibull distribution. See also the definition for η.

η - the shape parameter of a Weibull distribution; density function f(t) is Weibull if:

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

λ - the rate of occurrence per hour of assignable causes when the process failure mechanism is governed by the exponential distribution; that is, $\theta = \lambda$ when $\eta = 1$.

$\bar{X}''$ - standard or desired process mean.

σ - standard or true process standard deviation.

δ - magnitude of the out of control shift in the process mean in multiples of σ. 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_i - probability of detecting a shift on the i^{th} 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 i^{th} sample, when there is an assignable cause; $Q_i = 1 - P_i$.

α_i - probability of a false alarm on the i^{th} sample when there is no assignable cause; $\alpha_i = 2\,\Phi\,(-k_i)$.

OOC_i - an abbreviation for out-of-control in the i^{th} interval.

$P(OOC_i)$ - 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; $\int_0^x Z^{a-1}\, e^{-Z} dZ$.

e - 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.

V_1 - 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.

b - the cost per sample of sampling, testing, and charting that is fixed and independent of the sample size.

c - 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 i^{th} interval, given that the shift has occurred in the i^{th} interval.

TOOC - the time that the process is operating in the out-of-control condition before the detecting sample is plotted on the chart.

AOOC - 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].

β - 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.

C_1 - the average hourly cost of looking for false alarms.

C_2 - the average hourly cost of finding the assignable cause.

C_3 - the average hourly cost of sampling, testing, and charting; AHCS.

L - 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 $\bar{X}$-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 n_1 is taken after the process has been operationg for h_1 hours, while the control limits are set at $\bar{X}'' \pm k_1 \frac{\sigma}{\sqrt{n_1}}$. This is followed by a second sample of size n_2 taken at time $h_1 + h_2$, while the control

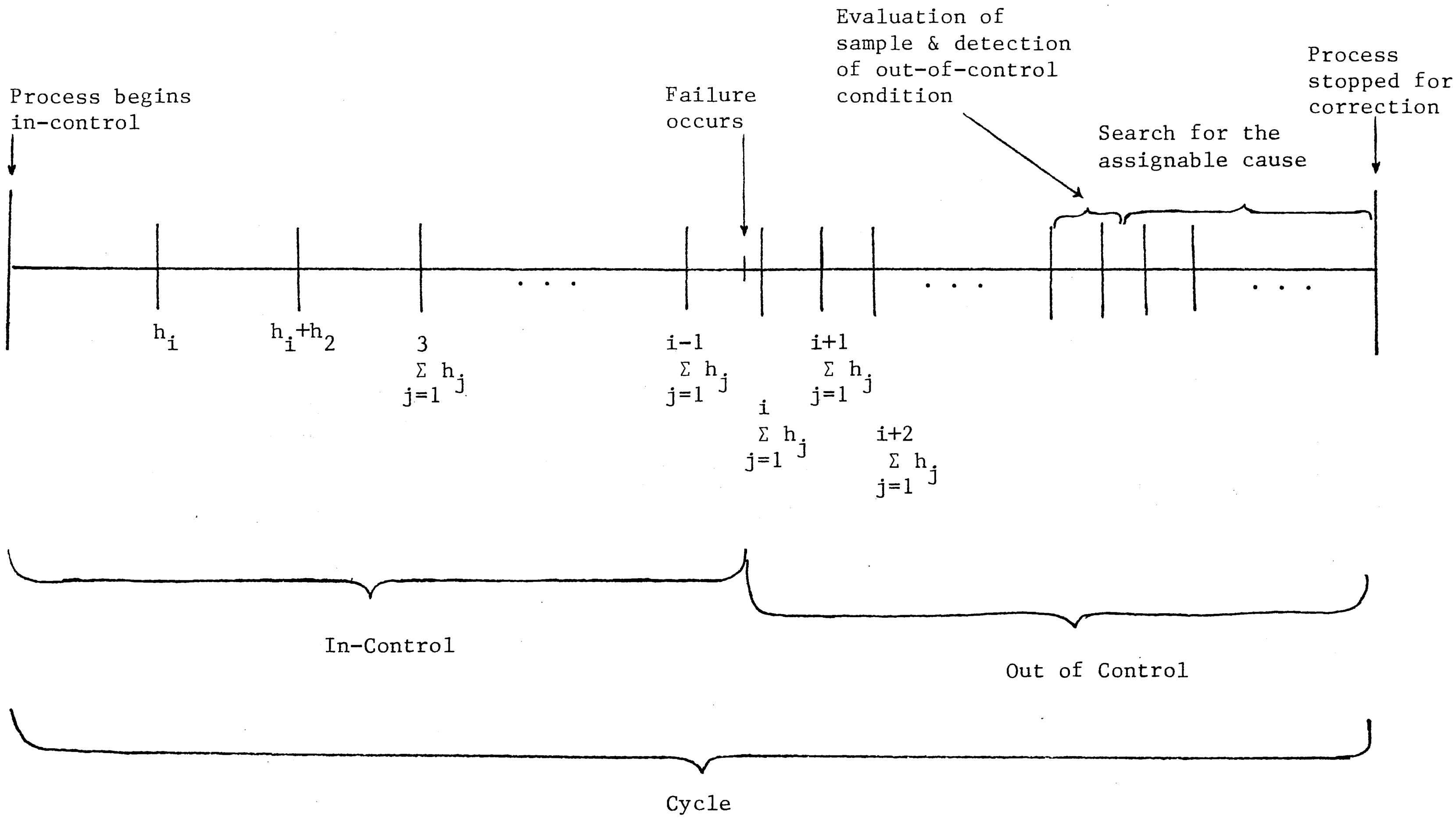


Figure 3.1. Cycle Time

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 i^{th} sample of size n_i is taken at time

$t_i = \sum_{j=1}^{i} h_j$ and is plotted on a chart with control limits at

$\bar{X}'' \pm k_i \frac{\sigma}{\sqrt{n_i}}$.

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 i^{th} sample point falls outside the control limit, when the process mean has shifted from $\bar{X}''$ to $\bar{X}'' + \delta\sigma$. Therefore,

$$P_i = \int_{-\infty}^{-k_i-\delta\sqrt{n_i}} \frac{e^{-\frac{z^2}{2}}}{\sqrt{2\pi}}\, dZ + \int_{k_i-\delta\sqrt{n_i}}^{\infty} \frac{e^{-\frac{z^2}{2}}}{\sqrt{2\pi}}\, dZ \qquad (3.1)$$

$$= \Phi(-k_i - \delta\sqrt{n_i}) + 1 - \Phi(k_i - \delta\sqrt{n_i}) \qquad (3.2)$$

Q_i - Probability of not detecting a shift on the i^{th} sample, when there is an assignable cause. Therefore,

$$Q_i = 1 - P_i \qquad (3.3)$$

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

$$\alpha_i = 2 \int_{k_i}^{\infty} \frac{e^{-\frac{z^2}{2}}}{\sqrt{2\pi}}\, dZ \qquad (3.4)$$

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

$P(OOC_i)$ - 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,

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

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:

$$\begin{matrix}\text{Average} \\ \text{cycle} \\ \text{time}\end{matrix} = \begin{pmatrix}\text{Average} \\ \text{in-control} \\ \text{time}\end{pmatrix} + \begin{pmatrix}\text{Average time the} \\ \text{process is out-of-} \\ \text{control before the} \\ \text{detecting sample is} \\ \text{plotted on the chart}\end{pmatrix} + \begin{pmatrix}\text{Average time} \\ \text{to find the} \\ \text{assignable cause} \\ \text{during which} \\ \text{the process is} \\ \text{out of control}\end{pmatrix}$$

or,

$$ACT = AIC + AOOC + D \tag{3.7}$$

where:

ACT - the average cycle time.

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

AOOC - 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 $\frac{1}{\theta}\Gamma\ (1 + \frac{1}{\eta})$.

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. $P(OOC_i)$ denotes this probability of a shift to the out-of-control condition in the i^{th} interval.

First, assume that the process has shifted to the out-of-control state in an arbitrary interval, e.g., the i^{th} 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):

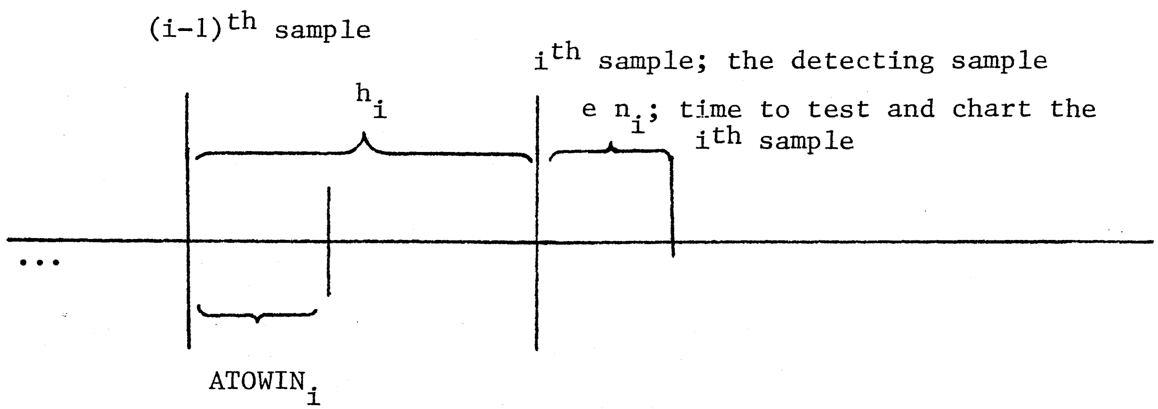


Figure 3.2 Out-of-Control Time Before the Detecting Sample is Plotted for Case (1)

$$h_i + e\, n_i - ATOWIN_i \tag{3.8}$$

where:

h_i - the length of the i^{th} interval.

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

$ATOWIN_i$ - the average time of the occurrence of the shift within the i^{th} interval, given that the shift occurs in the i^{th} interval. Development of the expression for $ATOWIN_i$ 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):

$$h_i + h_{i+1} + e\, n_{i+1} - ATOWIN_i \tag{3.9}$$

(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):

$$h_i + h_{i+1} + h_{i+2} + e\, n_{i+2} - ATOWIN_i \tag{3.10}$$

.
.
.

(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

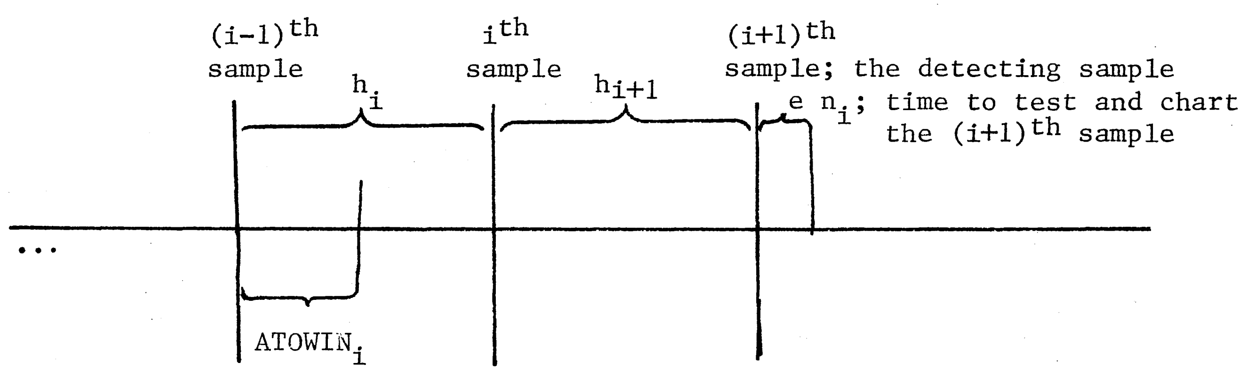


Figure 3.3. Out-of-Control Time Before the Detecting Sample is Plotted for Case (2).

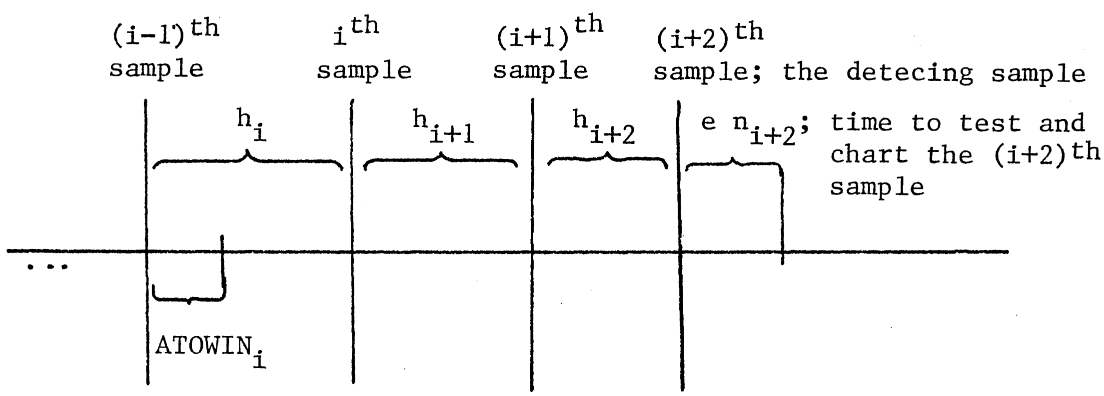


Figure 3.4. Out-of-Control Time Before the Detecting Sample is Plotted for Case (3)

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:

$$
\begin{aligned}
E\,[TOOC|OOC_i] = {} & P_i[h_i + e\ n_i - ATOWIN_i] \\
& + Q_i\ P_{i+1}\ [h_i + h_{i+1} + e\ n_{i+1} - ATOWIN_i] \\
& + Q_iQ_{i+1}P_{i+2}[h_i + h_{i+1} + h_{i+2} + e\ n_{i+2} - ATOWIN_i] \\
& + \ldots
\end{aligned}
\tag{3.11}
$$

$$
\begin{aligned}
= {} & P_i\ [h_i + e\ n_i - ATOWIN_i] \\
& + \sum_{j=i+1}^{\infty} \left[\prod_{k=1}^{j-1} Q_k \right] P_j \left[\sum_{k=1}^{j} h_k + e\ n_j - ATOWIN_i \right]
\end{aligned}
\tag{3.12}
$$

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:

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

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 in-control 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:

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

<u>$ATOWIN_i$ Expression</u>

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

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

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

This can be simplified as follows:

$$ATOWIN_i = \frac{\int_{t_{i-1}}^{t_i} t\, f(t)\, dt - \int_{t_{i-1}}^{t_i} t_{i-1}\, f(t)\, dt}{\int_{t_{i-1}}^{t_i} f(t)\, dt} \tag{3.17}$$

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

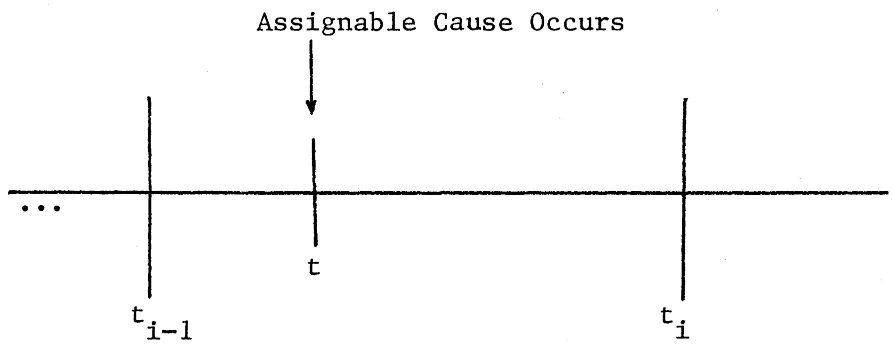


Figure 3.5. Average Time of Occurrence of Assignable Cause Within the i^{th} Interval

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

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

$ATOWIN_i$ is as follows:

$$ATOWIN_i = \frac{\int_{t_{i-1}}^{t_i} t \ \theta\eta (\theta t)^{\eta-1} \ e^{-(\theta t)^{\eta}} \ dt}{\int_{t_{i-1}}^{t_i} \theta\eta (\theta t)^{\eta-1} \ e^{-(\theta t)^{\eta}} \ dt} - \sum_{j=1}^{i-1} h_j \tag{3.20}$$

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

$$ATOWIN_i = \frac{\int_{(\theta t_{i-1})^{\eta}}^{(\theta t_i)^{\eta}} \frac{1}{\theta} u^{\frac{1}{\eta}} \ e^{-u} \ du}{\int_{(\theta t \quad)^{\eta}}^{(\theta t_i)^{\eta}} e^{-u} \ d \ u} - \sum_{j=1}^{i-1} h_j \tag{3.21}$$

$$= \frac{\frac{1}{\theta} [\gamma((\theta t_i)^{\eta}, \frac{1}{\eta} + 1) - \gamma((\theta t_{i-1})^{\eta}, \frac{1}{\eta} + 1)]}{\theta [e^{-(\theta t_i)^{\eta}} - e^{-(\theta t_{i-1})^{\eta}}]} - \sum_{j=1}^{i-1} h_j \tag{3.22}$$

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 i^{th} interval. Given this assumption, the expected number of false alarms is:

$$E\left[\begin{matrix}\text{number of}\\ \text{false alarms}\end{matrix} \mid OOC_i\right] = \alpha_1 + \alpha_2 + \dots + \alpha_{i-1} \qquad (3.23)$$

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,

$$ENFALS = \sum_{i=1}^{\infty} P(OOC_i) \left[\sum_{j=1}^{i-1} \alpha_j\right] \qquad (3.24)$$

The expected number of false alarms per hour of operation is equal to $\frac{ENFALS}{ACT}$.

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:

$$C_1 = T \frac{ENFALS}{ACT} \tag{3.25}$$

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^{th} interval. Given this assumption, consider the following cases:

(1) The shift which occurred in the i^{th} 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

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

where:

$$m1 = \min \ell 1 \ni \sum_{j=i+1}^{\ell 1} h_j > e\, n_i + D$$

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

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

So, the average hourly cost of sampling and charting for this case is equal to $\frac{CI1}{LI1}$.

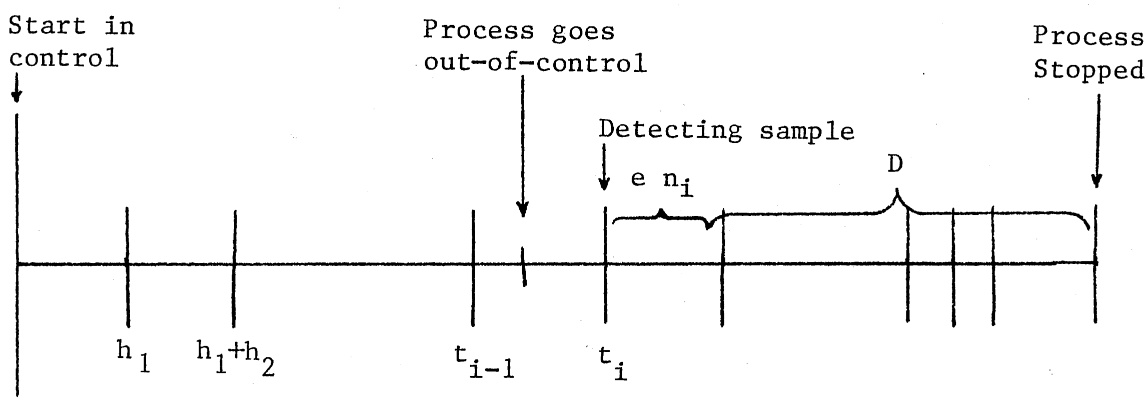


Figure 3.6. Cycle Length for Case (1)

(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:

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

where

$$m2 = \min \ell 2 \ni \sum_{j=i+2}^{\ell 2} h_j \geq e\, n_{i+1} + D$$

The cycle length in this case is:

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

So, the average hourly cost of sampling and charting for this case is equal to $\frac{CI2}{LI2}$.

(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:

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

where:

$$m3 = \min \ell 3 \ni \sum_{j=i+3}^{\ell 3} h_j \geq e\, n_{i+2} + D$$

The cycle length in this case is:

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

So, the average hourly cost of sampling and charting for this case is equal to $\frac{CI3}{LI3}$.

.
.
.

(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:

$$E \begin{bmatrix} \text{hourly cost of} \\ \text{sampling and} \\ \text{charting} \end{bmatrix} \begin{matrix} \\ \mid OOC_i \\ \\ \end{matrix} = P_i \frac{CI1}{LI1}$$

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

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

$$C_3 = AHCS = \sum_{i=1}^{\infty} E \left[\begin{matrix} \text{hourly cost of} \\ \text{sampling and} \\ \text{charting} \end{matrix} \ \middle|\ OOC_i \right] P(OOC_i) \qquad (3.34)$$

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 $\bar{X}$-control chart is developed as follows.

$$
\begin{aligned}
\begin{matrix}\text{Process average}\\ \text{hourly net}\\ \text{income}\end{matrix} &= \begin{pmatrix}\text{Average hourly}\\ \text{income from}\\ \text{in-control operation}\end{pmatrix} \\
&+ \begin{pmatrix}\text{Average hourly}\\ \text{income from}\\ \text{out-of-control}\\ \text{operation}\end{pmatrix} - \begin{pmatrix}\text{Average hourly}\\ \text{cost of}\\ \text{false alarms}\end{pmatrix} \\
&- \begin{pmatrix}\text{Average hourly}\\ \text{cost of finding}\\ \text{assignable causes}\end{pmatrix} - \begin{pmatrix}\text{Average hourly}\\ \text{cost of sampling,}\\ \text{testing, and}\\ \text{charting}\end{pmatrix}
\end{aligned}
\tag{3.35}
$$

where

$$
\begin{aligned}
\begin{matrix}\text{Average hourly}\\ \text{income from}\\ \text{in-control}\\ \text{operation}\end{matrix} &= \begin{pmatrix}\text{Hourly income}\\ \text{from in-control}\\ \text{operation}\end{pmatrix} \\
&\times \begin{pmatrix}\text{Proportion of the}\\ \text{time the process}\\ \text{is in-control}\end{pmatrix} = V_0\,\beta
\end{aligned}
\tag{3.36}
$$

$$\text{Average hourly income from out-of-control operation} = \left(\text{Hourly income from out-of-control operation}\right)$$

$$\times\left(\text{Proportion of the time the process is out-of-control}\right) = V_1 \ (1-\beta) \tag{3.37}$$

Average hourly cost of false alarms is:

$$C_1 = T \frac{\text{ENFALS}}{\text{ACT}} \tag{3.38}$$

Average hourly cost of finding the assignable cause is:

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

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

$$C_3 = \text{AHCS} \tag{3.40}$$

Therefore,

$$\begin{aligned} \text{Process average hourly net income} &= V_0 \beta + V_1(1-\beta) + C_1 + C_2 + C_3 \\ &= V_0 \beta + V_1(1-\beta) - T \frac{\text{ENFALS}}{\text{ACT}} \\ &\quad - \frac{W}{\text{ACT}} - \text{AHCS} \end{aligned} \tag{3.41}$$

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

where:

$$M = V_0 - V_1 \tag{3.43}$$

$$= V_0 - L \tag{3.44}$$

where:

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

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_i - time interval elapsed between taking the $(i-1)^{th}$ and the i^{th} samples.

Ih - time interval elapsed from the start of the process until the first sample is taken.

ISTEPS - Integer number of steps taken to achieve a desired quantile value for a process failure distribution. That is, $\sum_{i=1}^{ISTEPS} h_i$ is equal to the desired quantile. Note that the specification of any two of h_i, Ih, and ISTEPS determines the third.

t_i - time elapsed from startup of the process in-control until the i^{th} sample is taken; $t_i = \sum_{j=1}^{i} h_j$.

hf - 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 i^{th} sample average standard deviations separating each control limit and the original process mean. That is, the i^{th} 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_i 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 $\theta = \lambda$ when $\eta = 1$.

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.

e - 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.

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

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

n', n'' - temporary trial sample sizes used in Duncan's model optimization.

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

A Dynamic $\bar{X}$-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.

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

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

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

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 $t_i = \sum_{j=1}^{i} h_j$.

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

nf - 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 $t_i = \sum_{j=1}^{i} h_j$ 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_i 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 n_{i-1} and n_i, h_{i-1} and h_i, and k_{i-1} and k_i is desirable. Rather, an optimization procedure should determine n_i, h_i, and 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 i^{th} term of the series involved $P(OOC_i)$, the probability that the process shifts to the out-of-control condition in the i^{th} 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 j^{th} element involves $\prod_{k=i}^{j-1} Q_k$, 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 $P(OOC_i)$, first note that $P(OOC_i)$, 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 $P(OOC_i)$ 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, $P(OOC_i)$ approaches zero. Therefore, there is a quantile point for the cumulative distribution function after which $P(OOC_i)$ 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 $P(OOC_i)$. 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

Ex. No.	Selected Design n	h	k	λ	.99 Quantile Value	.99 Quantile Loss-Cost†	.9999 Quantile Value	.9999 Quantile Loss-Cost	.999999 Quantile Value	.999999 Quantile Loss-Cost	Duncan's Exact* Model Loss-Cost
1	5	1.41	3.08	.01	460.52	3.99170	921.03	4.01254	1381.55	4.01278	4.01278
3	4	0.78	2.94	.03	153.51	9.55198	307.01	9.59190	460.51	9.59238	9.59239
10	6	1.4	3.7	.01	460.52	6.34565	921.03	6.36819	1381.55	6.36845	6.36845
20	8	12.	1.9	.01	460.52	2.39604	921.03	2.42093	1381.55	2.42128	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 $\prod_{k=i}^{j-1} Q_k$, 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 value of $\prod_{k=i}^{j-1} Q_k$ 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

Ex. No.*	Desc.†	Optimal Design n	h	k	$k-\delta\sqrt{n}$	P
1	δ=2,G	5	1.41	3.08	-1.3921	.9181
2	δ=2,G	5	1.02	3.08	-1.3921	.9181
3	δ=2,G	4	0.78	2.94	-1.0600	.8554
4	δ=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	δ=2,G	5	1.62	3.05	-1.4221	.9225
9	δ=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	δ=2,D	3	2.6	2.4	-1.0641	.8564
14	δ=2,G	1	4.66	1.46	-.5400	.7054
15	δ=2,D	3	.8	2.4	-1.0610	.8564
16	δ=1,G	14	5.47	2.68	-1.0617	.8558
17	δ=1,D	12	1.6	2.6	-.8641	.8062
18	δ=1,G	21	7.23	3.39	-1.1926	.8835
19	δ=1,G	18	11.02	2.56	-1.6826	.9538
20	δ=1,D	8	12.	1.9	-.9284	.8234
21	δ=.5,G	38	23.45	2.21	-.8722	.8085
22	δ=.5,G	21	1.3	2.11	-.1813	.5719
23	δ=.5,D	55	30.	2.3	-1.4081	.9204
24	δ=.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.
†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

$$P_i > .5 \qquad (4.4)$$

Substitute in equation (3.2) for P_i results in:

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

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,

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

or

$$\delta \sqrt{n_i} - k_i > 0 \qquad (4.7)$$

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, multidimensional 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:

1. 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.
   (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:

1. 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.
2. 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:

1. Start with the optimal solution obtained in the conclusion of the search over the kf and Ik variables or the best solution found thusfar.
2. 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].

$$F_1 = -3803.84 - 138.08\ X_1 - 232.92\ X_2 + 123.08\ X_1^2 + 203.64\ X_2^2 + 182.25\ X_1\ X_2 \tag{4.8}$$

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:

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

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 $.284 \times 10^{-7}$ 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:

$$F_3 = (1.5 - X_1(1 - X_2))^2 + (2.25 - X_1(1 - X_2^2))^2 + (2.625 - X_1(1 - X_2^3))^2 \qquad (4.10)$$

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 $.7 \times 10^{-7}$ 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.

<u>Duncan's Model Optimization</u>

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 $L_{n'}^*$.

4. Fix the integer sample size at $n'' = n' + \text{IDIRC}$, 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 $L^*_{n''}$.

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	e	D	T	W	b	c	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					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; η is 6
3b		.03								6	.02783158	Same as 3 but η is 6
16b	1	.01	12.87							6	.00927719	Same as 16 but η is 6
22b	.5		225							6	.00927719	Same as 22 but η 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 η parameter of the Weibull distribution used in the dynamic $\bar{X}$-chart model has a value of 3. For group 4 examples, the η parameter is set at 6. Note that in any case the θ 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.

1. 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

1. 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 $\lambda = .01$. If the η 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; $\mu = 1/\lambda = 100$. Therefore,

TABLE 4.4

OPTIMAL ECONOMIC DESIGNS OF DUNCAN'S AND THE DYNAMIC $\bar{X}$-CHART AND THEIR COMPARISONS

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

Table 4.4 (Continued)

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

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

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

$$\theta = .00892975 \tag{4.13}$$

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 $\bar{X}$-chart design becomes more significant.

CHAPTER V

USING THE INTERACTIVE COMPUTER PROGRAM

Introduction

Overview

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 $\bar{X}$-control chart and compare it with the corresponding Duncan's $\bar{X}$-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.

```
     ==> MAIN MENU <==

*** ENTER OPTION NUMBER                                              (M.1)
     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=    99.9999
CORRECT ?   1=YES  2=NO  3=RETURN TO THE MAIN MENU
?
1
    ENTER VALUES: DELTA, B, C, D, E, M, T, W
?
1  .5  .1  2.  .05  12.87  50.  25.

    COST AND OTHER INFORMATION:                                     (M.2)
      DELTA=     1.0000  B=     0.5000  C=     0.1000  D=     2.0000
          E=     0.0500  M=    12.8700  T=    50.0000  W=    25.0000
CORRECT ?   1=YES  2=NO  3=RETURN TO THE MAIN MENU
?
1

*** 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
?
1
```

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:
      IN=  14        IH=    5.4828  IK=    2.6734
      NF=1.0000000  HF=1.0000000   KF=1.0000000
    YOU ACCEPT THIS POINT.
CORRECT ?   1=YES  2=NO  3=RETURN TO THE PREVIOUS MENU
?
2
FOR YOUR DESIRED STARTING POINT FOR OPTIMIZATION, ENTER:
    IN, IH, IK,  NF, HF, KF
?
14  5.5  -2.7  1.  10.  1.
!?!  ERROR -- IK SHOULD BE BETWEEN 0.0 AND 12.
!?!  ERROR -- NF, NH, AND NK SHOULD BE BETWEEN 0.0 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      IK=    2.7000
                NF=1.0000000  HF=1.0000000       KF=1.0000000
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    IK         KF         IN        NF

STEP SIZE:      0.001000   2.     0.1000   0.0069976   1.    0.0069976
MIN STEP SIZE: 0.000100   1.     0.1000   0.0017494   1.    0.0017494
MAX STEP SIZE: 0.004000  32.     0.2000   0.0069976   2.    0.0069976
REQ PRECISION: 0.000050   1.     0.0500   0.0011663   1.    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
?
3
```

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
?
95   25   25
    OPTIMIZATION PARAMETERS:
                   HF      ISTEPS    IK         KF         IN        NF

STEP SIZE:      0.001000   2.     0.1000   0.0069976   1.    0.0069976
MIN STEP SIZE: 0.000100   1.     0.1000   0.0017494   1.    0.0017494
MAX STEP SIZE: 0.004000  32.     0.2000   0.0069976   2.    0.0069976
REQ PRECISION: 0.000050   1.     0.0500   0.0011663   1.    0.0011663

MAX. NUMBER OF ITERATIONS (LOSS-COST EVALUATIONS):
    ITRMX1=  95   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
?
1
```

```
************************************************************************
**************  ECON. DESIGN OF THE DYNAMIC X-BAR CHART  **************

    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
          E=    0.0500  M=   12.8700  T=   50.0000  W=   25.0000
    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: 0.990000000

    MAX. NUMBER OF ITERATIONS (LOSS-COST EVALUATIONS):
      ITRMX1=  95   ITRMX2=  25   ITRMX3=  25

CHECK THE ABOVE INFORMATION.
EVERYTHING IS CORRECT ?   1=YES   2=NO
?
1

                 ** 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
?
2
*** 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        IH=   5.5000       IK=   2.7000
                NF=1.0000000  HF=1.0000000      KF=1.0000000
CORRECT ?   1=YES  2=NO  3=RETURN TO THE PREVIOUS MENU
?
10
!?! ERROR -- DO IT OVER !
CORRECT ?   1=YES  2=NO  3=RETURN TO THE PREVIOUS MENU
?
1
QUANTILE VALUE OF 0.99000000 IS USED.
YOU ACCEPT THIS.
CORRECT ?   1=YES  2=NO  3=RETURN TO THE PREVIOUS MENU
?
2
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
?
1

************************************************************************
**************  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
          E=    0.0500  M=   12.8700  T=   50.0000  W=   25.0000
    *** THE DESIGN TO BE EVALUATED IS: ***
      IN=  14        IH=    5.5000  IK=    2.7000
      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
?
1
```

```
*** 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
    IN=  14        IH=     5.4686  IK=     2.7000
    NF=1.0000000  HF=1.0000000   KF=1.0000000
  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
?
3
    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
?
1

*** 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
?
1
*** FOR ECON. OPTIMIZATION OF THE DYNAMIC X-BAR CHART,
    THE FOLLOWING STARTING POINT IS SUGGESTED:
      IN=  14        IH=     5.4817  IK=     2.6736
      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    IK          KF       IN        NF

STEP SIZE:      0.001000   2.     0.1000   0.0093410   1.    0.0093410
MIN STEP SIZE: 0.000100   1.     0.1000   0.0023353   1.    0.0023353
MAX STEP SIZE: 0.004000  32.     0.2000   0.0093410   2.    0.0093410
REQ PRECISION: 0.000050   1.     0.0500   0.0015568   1.    0.0015568

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
?
1

*********************************************************************
**************  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=     0.1000  D=     2.0000
          E=     0.0500  M=    12.8700  T=    50.0000  W=    25.0000
    STARTING POINT FOR OPTIMIZATION IS:
      IN=  14        IH=     5.4817  IK=     2.6736
      NF=1.0000000  HF=1.0000000   KF=1.0000000

    QUANTILE VALUE IS: 0.990000000

    MAX. NUMBER OF ITERATIONS (LOSS-COST EVALUATIONS):
      ITRMX1=  45   ITRMX2=  25   ITRMX3=  25

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
?
1

*********************************************************************
**************  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=     0.1000  D=     2.0000
          E=     0.0500  M=    12.8700  T=    50.0000  W=    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: 0.990000000

    MAX. NUMBER OF ITERATIONS (LOSS-COST EVALUATIONS):
      ITRMX1=  45   ITRMX2=  25   ITRMX3=  25

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
?
2

*** 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
?
4
    ENTER VALUES: DELTA, B, C, D, E, M, T, W
?
1.  .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  W=   25.0000
CORRECT ?   1=YES  2=NO  3=RETURN TO THE MAIN MENU
?
1

*** 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
?
2
```

```
*** 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        IH=   5.5000      IK=   2.7000
                 NF=1.0000000  HF=1.0000000      KF=1.0000000
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

************************************************************************
**************  ECON. EVALUATION IN WEIBULL ENVIRONMENT  **************

    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=     2.0000
          E=     0.0500  M=    12.8700  T=    50.0000  W=    25.0000
    *** THE DESIGN TO BE EVALUATED IS: ***
      IN=  14        IH=     5.5000  IK=     2.7000
      NF=1.0000000  HF=1.0000000   KF=1.0000000

    QUANTILE VALUE IS:0.990000000

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
    LOSS-COST PER 100 HOURS = $    364.735

   *** 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 $\bar{X}$-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
    ?
    2

         >> EXPONENTIAL ENVIRONMENT <<
    *** FOR DUNCAN'S ECONOMIC X-BAR CHART, ENTER VALUES:
        LAMBDA, DELTA, B, C, D, E, M, T, W
    VALUES ENTERED ARE:
    ?
    .01  1.  .5  .1  2.  .05  12.87  50.  25.
        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

    *** 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
    ?
    1
```

(M.3)

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=     3.0000
    YOU ACCEPT THIS POINT.
CORRECT ?   1=YES  2=NO  3=RETURN TO THE PREVIOUS MENU
?
9
!?! ERROR -- DO IT OVER !
CORRECT ?   1=YES  2=NO  3=RETURN TO THE PREVIOUS MENU
?
1
    FOR YOUR DESIRED STARTING POINT FOR OPTIMIZATION, ENTER:
     N, H, K
?
4  1.  3.
    VALUES ENTERED:  N=   4      H=  1.0000      K=  3.0000
CORRECT ?   1=YES  2=NO  3=RETURN TO THE PREVIOUS MENU
?
1

*******************************************************************
*************  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
          E=     0.0500  M=    12.8700  T=    50.0000  W=    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
?
1
        ** 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
?
2
*** FOR ECON. EVALUATION IN THE EXPONENTIALENVIRONMENT,
    ENTER VALUES:  N, H, K
?
4  10.  3.
    VALUES ENTERED:  N=    4       H= 10.0000       K=  3.0000
CORRECT ?   1=YES  2=NO  3=RETURN TO THE PREVIOUS MENU
?
1
*** FOR ECON. EVALUATION IN THE EXPONENTIALENVIRONMENT,
    ENTER VALUES:  N, H, K
?
4  1.  3.
    VALUES ENTERED:  N=    4       H=  1.0000       K=  3.0000
CORRECT ?   1=YES  2=NO  3=RETURN TO THE PREVIOUS MENU
?
1

****************************************************************
*********  ECON. EVALUATION IN EXPONENTIAL ENVIRONMENT  *********

    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
?
1
******* 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
?
3
```

```
     >> 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;
      EXPONENTIAL W/ LAMBDA=     0.0100 => MEAN=  100.0000
    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  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
?
1
*** 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
?
1

******************************************************************
*************  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=     1.0000  D=     2.0000
          E=     0.0500  M=    12.8700  T=    50.0000  W=    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
?
1
         ** 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
?
4


    ==> MAIN MENU <==

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


?
3
```

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

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 $\bar{X}$-Chart." AIIE Transactions, 3, 4 (Dec., 1971), 257-263.

[3] Barish, N. N. and N. Hauser. "Economic Design for Control Decisions." Journal of Industrial Engineering, 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." International Journal of Production Research, 15, 2 (1977), 117-128.

[10] Case, K. E. and L. L. Jones. Profit Through Quality--A Quality Assurance Program for Manufacturers. AIIE Monograph Series. Norcross, Georgia: American Institute of Industrial Engineers, 1978.

[11] Chiu, W. K. "Comments on the Economic Design of $\bar{X}$-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 $\bar{X}$-Charts." Journal of Quality Technology, 6, 2 (April, 1974), 63-69.

[13] Chiu, W. K. and G. B. Wetherill. "Quality Control Practices." International Journal of Production Research, 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 $\bar{X}$-Charts." Metrika, 23, 3 (Sept., 1976), 135-147.

[16] Chiu, W. K. and C. H. Cheung. "An Economic Study of $\bar{X}$-Charts with Warning Limits." Journal of Quality Technology, 9, 4 (Oct., 1977), 166-171.

[17] Cowden, D. J. Statistical Methods in Quality Control. Englewood Cliffs: Prentice Hall, Inc., 1957.

[18] Craig, C. C. "The $\bar{X}$- and R-Charts and Its Competitors." Journal of Quality Technology, 1, 2 (April, 1969), 102-104.

[19] Drury, C. G. and J. G. Fox. Human Reliability in Quality Control. London: Taylor and Francis Ltd., 1975.

[20] Duncan, A. J. "The Economic Design of $\bar{X}$-Charts Used to Maintain Current Control of a Process." Journal of the American Statistical Association, 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." Journal of the American Statistical Association, 66, 333 (March, 1971), 107-121.

[22] Duncan, A. J. Quality Control and Industrial Statistics. 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. Practical Methods of Optimization. Vol. 1: Unconstrained Optimization, John Wiley & Sons, Ltd., 1980.

[25] Fruend, R. A. "Acceptance Control Charts." Industrial Quality Control, 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." Industrial Quality Control, 18, 7 (Jan., 1962), 15-22.

[28] Ghare, P. M. and P. E. Torgersen. "The Multicharacteristic Control Chart." Journal of Industrial Engineering, 19, 6 (June, 1968), 269-272.

[29] Gibra, I. N. "Optimal Control of Processes Subject to Linear Trends." Journal of Industrial Engineering, 18, 1 (1967), 35-41.

[30] Gibra, I. N. "Economically Optimal Determination of the Parameters of $\bar{X}$-Control Chart." Management Science, 17, 9 (May, 1971), 635-646.

[31] Gibra, I. N. "Recent Developments in Control Chart Techniques." Journal of Quality Technology, 7, 4 (Oct., 1975), 183-192.

[32] Girshick, M. A. and H. Rubin. "A Bayes' Approach to a Quality Control Model." Annals of Mathematical Statistics, 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 $\bar{X}$-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. Statistical Quality Control. 5th Ed. New York: McGraw-Hill Book Company, 1980.

[37] Hayes, G. E. and H. G. Romig. Modern Quality Control. Encino, Cal.: Bruce, 1977.

[38] Heikes, R. G., D. C. Montgomery and J. Y. H. Yeung. "Alternative Process Models in the Economic Design of T^2 Control Charts." AIIE Transactions, 6, 1 (1974), 55-61.

[39] Himmelblau, D. M. Applied Nonlinear Programming. 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." AIIE Transactions, 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 T^2 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." AIIE Transactions, 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." Journal of Quality Technology, 12, 2 (April, 1980), 75-87.

[51] Montgomery, D. C. "Economic Design of an $\bar{X}$-Control Chart." Computer Programs, Journal of Quality Technology, 14, 1, (Jan. 1982), 40-43.

[52] Moore, P. G. "Some Properties of Runs in Quality Control Procedures." Biometrika, 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." Journal of Quality Technology, 6, 1, (Jan. 1974), 53-57.

[55] Page, E. S. "Cumulative Sum Charts." Technometrics, 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." Industrial Quality Control, 21, 5 (Nov., 1964), 245-249.

[58] Parzen, E. Stochastic Processes. San Francisco: Holden-Day, Inc., 1962.

[59] Roberts, S. W. "Control Charts Based on Geometric Moving Averages." Technometrics, 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." Computer Journal, 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] Scheffé, H. "Operating Characteristics of Average and Range Charts." Industrial Quality Control, 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." Technometrics, 9, 1 (1967), 29-41.

[73] Weiler, H. "On the Most Economical Sample Size for Controlling the Mean of a Population." Annals of Mathematical Statistics, 23 (1952), 247-254.

[74] Weiler, H. "The Use of Runs to Control the Mean in Quality Control." Journal of the American Statistical Association, 48, 264 (1953), 816-825.

[75] Weindling, J. L., S. B. Littauer and J. Tiago de Oliveria. "Mean Action Time of the $\bar{X}$-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." Management Science, 23, 8 (1974), 843-852.

[77] "ZXMIN," International Mathematical and Satistical Library, (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.

$$n_i = n, \ \forall i \qquad \text{(A.1)}$$

$$h_i = h, \ \forall i \qquad \text{(A.2)}$$

$$k_i = k, \forall i \tag{A.3}$$

$$t_i = \sum_{j=1}^{i} h_j = (i)\,h \tag{A.4}$$

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

$$Q_i = Q = 1 - P \tag{A.6}$$

$$\alpha_i = \alpha = 2\,\Phi\,(-k) \tag{A.7}$$

Also, equation (3.6) is simplified to:

$$P(OOC_i) = P(OOC) = \int_{t_{i-1}}^{t_i} \lambda\, e^{-\lambda t}\, dt = e^{-\lambda(i-1)h} - e^{-\lambda i h} \tag{A.8}$$

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:

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

AOOC--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:

$$AOOC = \sum_{i=1}^{\infty} P(OOC_i) \left\{ \left[P_i(h_i + en_i) \right.\right.$$

$$+ Q_i P_{i+1}(h_i + h_{i+1} + en_{i+1}) + \prod_{j=i}^{i+1} Q_j P_{i+2} \left(\sum_{j=i}^{i+2} h_j + en_{i+2} \right)$$

$$\left.\left. + \ldots + \prod_{j=1}^{r-1} Q_j P_r \left(\sum_{j=i}^{r} h_j + en_r \right) + \ldots \right] - ATOWIN_i \right\} \qquad (A.10)$$

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

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

$$\left.\left. + (1 - P)^2 P(3h + en) + \ldots \right] - ATOWIN \right\}$$

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

$$\left.\left. + (1 - P)^2 P(3h + en) + \ldots \right] - ATOWIN \right\} + \ldots \qquad (A.11)$$

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

$$AOOC = \left[P(h + en) + (1 - P)P(2h + en) \right.$$

$$\left. + (1 - P)^2 P(3h + en) + \ldots \right] - ATOWIN \qquad (A.12)$$

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

$$\text{AOOC} = \text{Pen}\left[1 + (1 - P) + (1 - P)^2 + \ldots\right]$$

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

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

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

and

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

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

$$\text{AOOC} = \text{Ph}\frac{1}{P^2} + P\ \text{en}\ \frac{1}{P} - \text{ATOWIN}$$

$$= h/P + \text{en} - \text{ATOWIN}$$

$$= h/P + \text{en} - h/2 + \lambda h^2/2 \qquad (A.14)$$

AIC--Average Cycle Length

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

$$\text{ACT} = \text{AIC} + \text{AOOC} + D \qquad (A.15)$$

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:

$$ACT = 1/\lambda + h/P + en - h/2 + \lambda h^2/2 \tag{A.16}$$

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:

$$ENFALS = \sum_{i=1}^{\infty} P(OOC_i) \left[\sum_{j=1}^{i-1} \alpha_j \right] \tag{A.17}$$

Substituting for $P(OOC_i)$ from equation (A.8) results in the following:

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

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

After simplification, the above equation can be written as:

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

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).

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

$$\left. + \; Q_i \; Q_{i+1} \; P_{i+2} \frac{CI3}{LI3} + \ldots \right) P(OOC_i) \qquad \text{(A.20)}$$

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

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

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

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

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

$$\left. + \; (1 - P)^2 P\left(\frac{b + cn}{h}\right) + \ldots \right] + \; (e^{-\lambda h} - e^{-2\lambda h}) \left[P\left(\frac{b + cn}{h}\right) \right.$$

$$\left. + \; (1 - P)\ P\left(\frac{b + cn}{h}\right) + (1 - P)^2 P\left(\frac{b + cn}{h}\right) + \ldots \right] + \ldots$$

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

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

$$= \frac{b + cn}{h} \qquad \text{(A.22)}$$

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:

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

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 $\frac{\alpha\ e^{-\lambda h}}{1 - e^{-\lambda h}}$ 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*************************************************************************
C*
C*  THIS INTRACTIVE PROGRAM PERFORMS
C*     (1) ECONOMIC DESIGN AND EVALUATION OF THE DYNAMIC X-BAR CHART
C*     (2) ECONOMIC DESIGN AND EVALUATION OF DUNCAN'S X-BAR CHART
C*
C*  BY BEHROOZ PARKHIDEH,  SCHOOL OF INDUSTRIAL ENGINEERING AND
C*                           MANAGEMENT
C*                         OKLAHOMA STATE UNIVERSITY
C*  DISSERTATION ADVISOR:  DR. KENNETH E. CASE
C*
C*
C*************************************************************************
C*
C*  GENERAL STRUCTURE AND INPUT REQUIREMENTS
C*
C*
C*    SUBROUTINE                  FUNCTION
C*   ------------          -----------------------
C*     DYNM          PROMPTS THE USER FOR INFORMATION NEEDE FOR
C*                   ECON. DESIGN AND/OR ECON. EVALUATION OF THE
C*                   DYNAMIC X-BAR CHART.
C*
C*     DUNC          PROMPTS THE USER FOR INFORMATION NEEDE FOR
C*                   ECON. DESIGN AND/OR ECON. EVALUATION OF
C*                   DUNCAN'S X-BAR CHART.
C*
C*     DUNOPT        OPTIMIZES DUNCAN'S MODEL.
C*
C*     FUNCT         DUNCAN'S COST MODEL USED FOR 3-DIMENSIONAL
C*                   OPTIMIZATION.
C*
C*     FUNCT2        DUNCAN'S COST MODEL USED FOR 2-DIMENSIONAL
C*                   OPTIMIZATION OVER H AND K.
C*
C*     PROBD         CALCUATES PROBABILITY OF DETECTING THE SHIFT
C*                   FOR DUNCAN'S MODEL.
C*
C*     PROFA         CALCUATES PROBABILITY OF FALSE ALARMS FOR
C*                   DUNCAN'S MODEL.
C*
C*     DYNOPT        OPTIMIZES THE DYNAMIC MODEL.
C*
C*     TWOSCH        TWO-AT-A-TIME SEARCH ROUTINE.
C*
C*     OMYSCH        USED IN CONJUNCTION WITH TWOSCH TO PERFORM A
C*                   TWO-DIMENSIONAL SEARCH.
C*
C*     COGGIN        PERFORMS A PRECISE LINE SEARCH USING THE
C*                   METHOD OF COGGINS.
C*
C*     CSTPW         CALCULATES THE VALUE OF ISTEPS FOR GIVEN VALUES
C*                   OF HF, IH, AND WEIBULL DISTRIBUTION PARAMETERS.
C*
C*     FTCR          CALCULATES THE VALUE OF  NF ( OR KF) FOR GIVEN
C*                   VALUES OF IN ( OR IK) AND IN ENDING ( OR IK ENDING)
C*
C*     SETDEL        CALAULATES THE VALUE OF INITIAL STEP SIZE FOR
C*                   KF ( OR NF) FOR THE SEARCH ROUTINE.
C*
C*     APP           CALCULATES THE AVERGE TIME THE PROCESS IS
C*                   IN OUT-OF-CONTROL CONDITION BEFORE THE DETECTING
C*                   SAMPLE IS PLOTTED ON THE CHART.
```

```
C*
C*      CMAINT          CALCULATES THE AVERAGE HOURLY COST OF MAINTAINING
C*                      THE CONTROL CHART ( FOR THE DYNAMIC MODEL.)
C*
C*      FALSA           CALCULATES THE AVERAGE NUMBER OF FALSE ALARMS PER
C*                      CYCLE ( FOR THE DYNAMIC MODEL.)
C*
C*      PROOCW          CALCULATES THE AREA UNDER WEIBULL DENSITY BETWEEN
C*                      A TO B ( FOR THE DYNAMIC MODEL.)
C*
C*
C***********************************************************************
C*
C*  EXTERNAL FUNCTIONS REQUIRED:
C*    (1) REGULAR SYSTEM SUPPLIED FORTRAN FUNCTIONS
C*    (2) FOUR IMSL SUBROUTINES
C*          MDNORD-- CALCULATES NORMAL DENSITY INTEGRAL.
C*          MDGAM -- CALCULATES THE INCOMPLETE GAMA INTEGRAL.
C*        MDGAMMA-- CALCULATES THE GAMMA FUNCTION.
C*          ZXMIN -- PERFOMS FUNCTION MINIMIZATION USING A QUASI-NEWTON
C*                   METHOD.
C*
C*
C*
C***********************************************************************
C*
C*
C*
C*
C*
C***********************************************************************
C** MAIN PROGRAM                                                        *
C**                                                                    *
C** THIS PROGRAM CALLS SUBROUTINES DYNM AND DUNC TO PERFORM THE         *
C** FOLLOWING TASKS:                                                   *
C**     (1) ECONOMIC DESIGN AND EVALUATION OF THE DYNAMIC X-BAR CHART  *
C**     (2) ECONOMIC DESIGN AND EVALUATION OF DUNCAN'S X-BAR CHART     *
C**                                                                    *
C***********************************************************************
C*
C*
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON / MAIN1 / LUR,LUW
C*
C
C
C**
C**LUR IS THE LOGICAL UNIT NUMBER OF THE READER
C**LUW IS THE LOGICAL UNIT NUMBER OF THE PRINTER
C**
      LUR=5
      LUW=6
C**
C**PROMPT THE USER WITH THE MAIN MENU
C**
   10 WRITE(LUW,11)
   11 FORMAT(/,/,T5,' ==> MAIN MENU <==',/,/,
     *        ' *** ENTER OPTION NUMBER',/,
     *        T5,' 1 = THE DYNAMIC X-BAR CHART(WEIBULL ENVIRONMENT)',/,
     *   T5,55H 2 = DUNCAN'S X-BAR CHART(EXPONENTIAL ENVIRONMENT)
     *     ,/,T5,' 3 = EXIT SYSTEM')
      READ(LUR,*)MENU1
      GO TO (100,200,300),MENU1
      WRITE(LUW,20)
   20 FORMAT(' !?! ERROR -- DO IT OVER !')
      GO TO 10
```

```
  100 CALL DYNM
      GO TO 10
  200 CALL DUNC
      GO TO 10
C**
C**EXIT SYSTEM
C**
  300 STOP
      END
C*
C*
C*
C*
C*********************************************************************
C*********************************************************************
C*********************************************************************
      SUBROUTINE DYNM
C*********************************************************************
C**                                                                 *
C** THIS ROUTINE PROMPTS THE USER FOR THE NECESSARY INFORMATION      *
C** NEEDED FOR THE DYNAMIC X-BAR CHART DESIGN OR EVALUATION.         *
C**                                                                 *
C** THIS ROUTINE CALLS THE FOLLOWING SUBROUTINES:                   *
C**      DUNOPT-- TO OPTIMZE DUNCAN'S MODEL EQUIVALENT TO THE DYNAMIC *
C**               MODEL IN ORDER TO GET A GOOD STARTING POINT FOR    *
C**               DYNOPT ROUTINE.                                   *
C**      DYNOPT-- OPTIMIZE DYNAMIC COST MODEL                       *
C**      DYMEVA-- EVALUATE DYNAMIC COST MODEL FOR A GIVEN DESIGN     *
C**      DUNOPT-- TO OPTIMZE DUNCAN'S MODEL EQUIVALENT TO THE DYNAMIC *
C**               MODEL IN ORDER TO GET A GOOD STARTING POINT FOR    *
C**               DYNOPT ROUTINE.                                   *
C**                                                                 *
C*********************************************************************
C**
C*
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*8 LAMBDA
      REAL*8 NF,IH,HF,IK,KF
      COMMON / MAIN1 /LUR,LUW
      COMMON / DCDY1 / DELTA, B,C,DD,E,VZMV1,T,W
      COMMON / DUNC1 / LAMBDA
      COMMON / DUNC4 / N,H,RK
      COMMON / DUNC5 / NDCOPT,HDCOPT,RKDCOP,FDCOPT
      COMMON / DYNM1 / THETA,ETA
      COMMON / DYNM2 / ISTEPS
      COMMON / DYNM3 / IN, NF,IH,HF,IK,KF
      COMMON / DYNM4 / PROBPT
      COMMON / DYNM5 / ITRMX1,ITRMX2,ITRMX3
      COMMON / DYNM6 / DEL(6),DELMN(6),DELMX(6),XQLIM(6)
      COMMON / DYNM7 / DYMLCS
      COMMON / DYNM8 / ISTPP
      COMMON / DYOPT4 / NWOPT,HWOPT,RKWOPT,FNWOPT,FHWOPT,FKWOPT,YFWOPT
      COMMON / CMN1 / CUPROX
C*
C_________ENTER DISTRIBUTION, COST, AND OTHER PARAMETERS_______________
C*
    5 WRITE(LUW,6)
    6 FORMAT(/,T5,'  >> WEIBULL ENVIRONMENT <<',/,
     *       ' *** FOR THE DYNAMIC ECONOMIC X-BAR CHART,')
C**
C**ENTER DISTRIBUTION INFROMATION
C**
C*.IREV=1 INDICATES ORDINARY INITIALIZATION OF PARAMETERS
C*.IREV=2 INDICATES THE PASS AFTER 3 ( DIST. PARMS. REVISION) IS
C*.SELECTED FROM THE FOLLOWING MENU.
C*.
```

```
      IREV=0
 6001 IREV=IREV+1
    7 WRITE(LUW,8)
    8 FORMAT(T5,' ENTER VALUES: THETA, ETA')
      READ(LUR,*)THETA,ETA
C*.
C*.CALCULATE MEAN OF THE WEIBULL DISTRIBUTION
C*.
       USEWMN=1.D0+1.D0/ETA
       WBMEAN=1.D0/THETA*DGAMMA(USEWMN)
      WRITE(LUW,12)THETA,ETA,WBMEAN
   12 FORMAT(' VALUES ENTERED ARE:',/,
     *       T5,' DISTRIBUTION INFORMATION;',/,
     *T7,' WEIBULL W/ THETA=',F10.6,' AND ETA=',F10.6,' => MEAN=',F10.4)
   13 WRITE(LUW,14)
   14 FORMAT(' CORRECT ?   1=YES  2=NO  3=RETURN TO THE MAIN MENU')
      READ(LUR,*)IYN1
      GO TO ( 15, 7, 600),IYN1
      WRITE(LUW,22)
      GO TO 13
C**
C**ENTER COST AND SHIFT PARAMETERS
C**
   15 IF(IREV.GT.1) GO TO 20
      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
   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)
   18 WRITE(LUW,19)
   19 FORMAT(' CORRECT ?   1=YES  2=NO  3=RETURN TO THE MAIN MENU')
      READ(LUR,*)IYN2
      GO TO ( 20, 15, 600),IYN2
      WRITE(LUW,22)
      GO TO 18
C*
C_________SELECTION FOR DESIGN, EVALUATION , ETC.______________________
C*
   20 IREV=1
      WRITE(LUW,21)
   21 FORMAT(/,' *** ENTER OPTION NUMBER',/,
     *   T5,' 1 = ECON. DESIGN OF THE DYNAMIC X-BAR CHART (OPTIMIZATION'
     *          ,')',/,
     *   T5,' 2 = ECON. EVALUATION IN THE WEIBULL ENVIRONMENT',/,
     *   T5,' 3 = RETURN TO REVISE WEIBULL PARAMETERS',/,
     *   T5,' 4 = RETURN TO REVISE COST PARAMETERS',/,
     *   T5,' 5 = RETURN TO THE MAIN MENU')
      READ(LUR,*)MENU2
      GO TO ( 100, 390 , 6001, 15, 600),MENU2
      WRITE(LUW,22)
   22 FORMAT(' !?! ERROR -- DO IT OVER !')
      GO TO 18
C*
C_________ECON. DESIGN (OPTIMIZATION) OF THE DYNAMIC X-BAR CHART_______
C---------------------------------------------------------------------
C*
C**
C**INITIALIZATION OF STARTING POINT FOR OPTIMIZATION
C**
C*.THE FOLLOWING N , H , AND K ARE USED AS STARTING POINT FOR OPTIMIZING
C*.DUNCAN'S MODEL WHICH PROVIDES THE STARTING POINT FOR OPTIMIZATION OF
C*.THE DYNAMIC MODEL.
C*.
  100 N=5
```

```
      H=1.D0
      RK=3.D0
C**
C**CALCULATE THE CORRESPONDING EXPONENTIAL PARAMETER USED IN DUNCAN'S
C**COST MODEL.
C**
      LAMBDA=1.D0/WBMEAN
C
C*.
      CALL DUNOPT
C*.
C
      IN=NDCOPT
      NF=1.D0
      IH=HDCOPT
      HF=1.0
      IK=RKDCOP
      KF=1.0
C*.
  101 WRITE(LUW,102)IN,IH,IK,NF,HF,KF
  102 FORMAT(' *** FOR ECON. OPTIMIZATION OF THE DYNAMIC X-BAR CHART,'
     * ,/,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.')
  103 WRITE(LUW,104)
  104 FORMAT(' CORRECT ?   1=YES  2=NO  3=RETURN TO THE PREVIOUS MENU')
      READ(LUR,*)IYN1
      GO TO (180,110,20),IYN1
      WRITE(LUW,106)
  106 FORMAT(' !?! ERROR -- DO IT OVER !')
      GO TO 103
C*
C_._.__._._IF THE SUGGESTED STARTING POINT IS NOT ACCEPTED_._.
C*
  110 WRITE(LUW,111)
  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
C**
C**CHECK TO SEE IF THESE ARE IN THE ACCEPTABLE RANGE
C**
      IF(IN.LT.1000 .AND.IN.GE.2 ) GO TO 118
          WRITE(LUW,115)
  115     FORMAT(' !?!  ERROR -- IN SHOULD BE BETWEEN 2 AND 1000')
          GO TO 135
  118 IF(IH.GT.0.0 .AND.IH.LT.100. ) GO TO 125
          WRITE(LUW,120)
  120     FORMAT(' !?!  ERROR -- IH SHOULD BE BETWEEN 0.0 AND 100.')
          GO TO 135
  125 IF( IK.GT.0.0 .AND. IK.LT.12. ) GO TO 129
          WRITE(LUW,127)
  127     FORMAT(' !?!  ERROR -- IK SHOULD BE BETWEEN 0.0 AND 12.')
  129 IF( NF.GT.2.D0 .OR. NF.LT.0.D0 ) GO TO 132
      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
      GO TO 155
C
  132     WRITE(LUW,133)
  133     FORMAT(' !?!  ERROR -- NF, NH, AND NK SHOULD BE BETWEEN '
     *    ,'0.0 AND 2.',/)
C
  135     WRITE(LUW,136)
  136     FORMAT(' DO IT OVER !')
          GO TO 110
C**
```

```
C**ECHOPRINT THE VALUES FOR CHECK
C**
  155 WRITE(LUW,158)IN,IH,IK,NF,HF,KF
  158 FORMAT(' VALUES ENTERED:  IN=',I4,5X,'  IH=',F9.4,4X,'  IK='
     *        ,F9.4,/,T17,'  NF=',F9.7, '  HF=',F9.7,4X,'  KF=',F9.7)
  159 WRITE(LUW,160)
  160 FORMAT(' CORRECT ?   1=YES  2=NO  3=RETURN TO THE PREVIOUS MENU')
      READ(LUR,*)IYN3
      GO TO (180,110,20),IYN3
      WRITE(LUW,161)
  161 FORMAT(' !?! ERROR -- DO IT OVER !')
      GO TO 159
C*
C*
C_._._._WHEN THE STARTING POINT IS ACCEPTED_._._
C*
C_._._._THEN SUGGEST THE QUANTILE_._._._._._._._
C*
  180 PROBPT=0.99D0
      WRITE(LUW,184)PROBPT
  184 FORMAT(T5, ' QUANTILE VALUE OF ',F11.9,' IS USED.',
     *       /,T5,' YOU ACCEPT THIS.')
  185 WRITE(LUW,186)
  186 FORMAT(' CORRECT ?   1=YES  2=NO  3=RETURN TO THE PREVIOUS MENU')
      READ(LUR,*)IYN5
      GO TO (200,190,18),IYN5
      WRITE(LUW,188)
  188 FORMAT(' !?! ERROR -- DO IT OVER !')
      GO TO 185
C_._._._._._._IF QUANTILE VALUE IS NOT ACCEPTED_._._._.
C*
C*
  190 WRITE(LUW,191)
  191 FORMAT(' ENTER YOUR DESIRED VALUE FOR QUANTILE:')
      READ(LUR,*)PROBPT
C**
C**CHECK TO SEE IF IT IS IN THE ACCEPTABLE RANGE
C**
      IF( PROBPT.LE.1.0D0 .AND. PROBPT.GT.0.D0 ) GO TO 200
      WRITE(LUW,195)
  195 FORMAT(' !?! ERROR -- QUANTILE SHOULD BE BETWEEN 0.0 AND 1.0',
     *       /,' DO IT OVER !')
      GO TO 190
C*
C_._._._._._IF QUANTILE VALUE IS ACCEPTED_._._._._._._._._._.
C_._._._._._THEN SUGGEST OPTIMIZATION PARAMETERS, ETC._._._.
C*
C**NOTE THAT VARIABLES:   1      2      3      4      5      6
C**CORRESPOND TO          HF  ISTEPS  IK     KF     IN     NF ,RESPECTIVELY
C*.
C*.CALCULATE ISTEPS FOR THE GIVEN VALUES OF HF, IH, PROBPT, AND
C*.DISTRIBUTIONA INFORMATION
C*.
  200 ISTEPS=CSTPW(HF,IH)
C*.
C*.ARRAY DEL CONTAINS THE INITIAL STEP SIZES
C*.
      DEL(1)=.001
      DEL(2)=2.
      DEL(3)=.1
C..
C..SET THE LIMIT ON KF AUTOMATICALLY BY SPECIFYING AN INCREMENT ON THE
C..IK-ENDING, DELIME.
C..
      DELIKE=.25D0
      DEL(4)=SETDEL(DELIKE)
```

```
      DEL(5)=1.
C..
C..SET THE LIMIT ON NF AUTOMATICALLY BY SPECIFYING AN INCREMENT ON THE
C..IN-ENDING, DELIME.
C..
      DELINE=.25D0
      DEL(6)=SETDEL(DELINE)
C*.
C*.INITIALIZE MINIMUNM LIMITS ON STEP SIZES
C*.
      DELMN(1)=.0001
      DELMN(2)=1.D0
      DELMN(3)=.1D0
      DELMN(4)=DEL(4)/4.D0
      DELMN(5)=1.D0
      DELMN(6)=DEL(6)/4.D0
C*.
C*.INITIALIZE MAXIMUM LIMITS ON STEP SIZES
C*.
      DELMX(1)=.004D0
      DELMX(2)=32.D0
      DELMX(3)=.2D0
      DELMX(4)=DEL(4)
      DELMX(5)=2.D0
      DELMX(6)=DEL(6)
C*.
C*.INITIALIZE THE REQUIRED PRECISION
C*.
      XQLIM(1)=.00005
      XQLIM(2)=1.D0
      XQLIM(3)=.05D0
      XQLIM(4)=DEL(4)/6.D0
      XQLIM(5)=1.D0
      XQLIM(6)=DEL(6)/6.D0
C*.
C*.INITIALIZE THE LIMITS ON THE MAXIMUM NUMBER OF LOSS-COST EVALUATIONS
C*.FOR THE THREE STAGS OF THE SEARCH.
C*.
      ITRMX1=45
      ITRMX2=25
      ITRMX3=25
C*.
C*.SUGGEST THESE TO THE USER
C*.
C
  205 WRITE(LUW,210)
  210 FORMAT(T5,' OPTIMIZATION PARAMETERS:',/,
     * T18,'  HF    ISTEPS',T36,'IK          KF',T54,'IN        NF',/)
      WRITE(LUW,213) (DEL(I),I=1,6)
  213 FORMAT(' STEP SIZE:',T17,F8.6,T27,F3.0,T33,F7.4,T43,F9.7,T54,
     *     F3.0,T60,F9.7)
      WRITE(LUW,215) (DELMN(I),I=1,6)
  215 FORMAT(' MIN STEP SIZE:',T17,F8.6,T27,F3.0,T33,F7.4,T43,F9.7,T54,
     *     F3.0,T60,F9.7)
      WRITE(LUW,217) (DELMX(I),I=1,6)
  217 FORMAT(' MAX STEP SIZE:',T17,F8.6,T27,F3.0,T33,F7.4,T43,F9.7,T54,
     *     F3.0,T60,F9.7)
      WRITE(LUW,219) (XQLIM(I),I=1,6)
  219 FORMAT(' REQ PRECISION:',T17,F8.6,T27,F3.0,T33,F7.4,T43,F9.7,T54,
     *     F3.0,T60,F9.7,/)
C
      WRITE(LUW,222)ITRMX1,ITRMX2,ITRMX3
  222 FORMAT(' MAX. NUMBER OF ITERATIONS (LOSS-COST EVALUATIONS):',/,
     *   T5,' ITRMX1=',I4,'    ITRMX2=',I4,'    ITRMX3=',I4,/)
  225 WRITE(LUW,226)
  226 FORMAT(' ***ENTER OPTION NUMBER:',/,
```

```
     *     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')
      READ(LUR,*)MENU3
      GO TO (300, 230, 250, 20),MENU3
C**
  230 WRITE(LUW,231)
  231 FORMAT(' FOR VARIABLES: HF, ISTEPS, IK, KF, IN, AND NF,',/,/,
     *         ' ENTER INITIAL STEP SIZES:')
      READ(LUR,*)(DEL(I),I=1,6)
      WRITE(LUW,233)
  233 FORMAT(' THE MIN. LIMIT ON STEP SIZES:')
      READ(LUR,*) (DELMN(I),I=1,6)
      WRITE(LUW,235)
  235 FORMAT(' THE MAX. LIMIT ON STEP SIZES:')
      READ(LUR,*) (DELMX(I),I=1,6)
      WRITE(LUW,237)
  237 FORMAT(' THE REQUIRED PRECISION FOR EACH VARIABLES:')
      READ(LUR,*) (XQLIM(I),I=1,6)
C**
C**PARTIALLY CHECK THESE VALUES
C**
      ICHK=0
      DO 239 I=1,6
         IF( DEL(I).LT.DELMN(I) ) WRITE(LUW,240)I
  240    FORMAT(' !?! ERROR -- THE ',I1,'TH INITIAL STEP SIZE IS ',
     *          'LESS THAN ITS MIN. !',/,' DO IT OVER !')
         IF( DEL(I).LT.DELMN(I) ) ICHK=1
         IF( DEL(I).GT.DELMX(I) ) WRITE(LUW,242)I
  242    FORMAT(' !?! ERROR -- THE ',I1,'TH INITIAL STEP SIZE IS ',
     *          'MORE THAN ITS MAX. !',/,' DO IT OVER !')
         IF( DEL(I).GT.DELMX(I) ) ICHK=1
         IF( DELMN(I).GT.DELMX(I) ) WRITE(LUW,244)I
  244    FORMAT(' !?! ERROR -- THE ',I1,'TH MIN. LIMIT IS ',
     *          'MORE THAN THE ',I1,'TH MAX. !',/,' DO IT OVER !')
         IF( DELMN(I).GT.DELMX(I) ) ICHK=1
  239 CONTINUE
C*.
C*.IF A VALUE IS ENTERED INCORRECTLY , RETURN
C*.
      IF(ICHK.EQ.1) GO TO 230
C*.
C*.IF EVERYTHING IS CORRECT, THEN PRINT THEM OUT
C*.
      IF(ICHK.EQ.0) GO TO 205
C**
C**THE USER SPECIFIES MAXIMUM NUMBER OF ITERATIONS
C**
  250 WRITE(LUW,251)
  251 FORMAT(T5,' ENTER VALUES: ITRMX1, ITRMX2, ITRMX3')
      READ(LUR,*)ITRMX1,ITRMX2,ITRMX3
C*.
C*.PARTIALLY CHECK THESE VALUES
C*.
      IF(ITRMX1.LT.1) GO TO 256
      IF(ITRMX2.LT.1) GO TO 256
      IF(ITRMX3.LT.1) GO TO 256
C*.
C*.IF EVERYTHING IS CORRECT, THEN PRINT THEM OUT
C*.
      GO TO 205
C*.
C*.IF A VALUE IS ENTERED INCORRECTLY, THEN RETURN
C*.
  256 WRITE(LUW,257)
```

```
  257 FORMAT(' !?! ERROR -- THE MAX. NUMBER OF ITERATIONS SHOULD',
     *  ' BE AT LEAST 1.',/,' DO IT OVER !')
      GO TO 250
C*
C----------------------------------------
C*
  300 WRITE(LUW,303)
  303 FORMAT(/,1X,71('*'),/,1X,14('*'),2X,'ECON. DESIGN OF THE DYNAMIC',
     *    ' X-BAR CHART',2X,14('*'),/)
      WRITE(LUW,305)THETA,ETA,WBMEAN
  305 FORMAT(T5,' DISTRIBUTION INFORMATION;',/,
     *T7,' WEIBULL W/ THETA=',F10.6,' AND ETA=',F10.6,' => MEAN=',F10.4)
C
      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)
C
      WRITE(LUW,309)IN,IH,IK,NF,HF,KF
  309 FORMAT(T5,' STARTING POINT FOR OPTIMIZATION IS:',/,
     *    T6,'  IN=',I4,5X,'  IH=',F10.4,'  IK=',F10.4,/,
     *    T6,'  NF=',F9.7,'  HF=',F9.7,1X,'  KF=',F9.7,/)
      WRITE(LUW,311)PROBPT
  311 FORMAT(T5,' QUANTILE VALUE IS: ',F11.9,/)
      WRITE(LUW,313)ITRMX1,ITRMX2,ITRMX3
  313 FORMAT(T5,' MAX. NUMBER OF ITERATIONS (LOSS-COST EVALUATIONS):',/,
     *   T7,' ITRMX1=',I4,'   ITRMX2=',I4,'   ITRMX3=',I4./)
  315 WRITE(LUW,316)
  316 FORMAT( ' CHECK THE ABOVE INFORMATION.',/,
     * ' EVERYTHING IS CORRECT ?   1=YES  2=NO')
      READ(LUR,*)IYN7
      GO TO (330,321),IYN7
      WRITE(LUW,318)
  318 FORMAT(' !?! ERROR -- DO IT OVER !')
      GO TO 315
  321 WRITE(LUW,322)
  322 FORMAT(1X,71('*'),/)
C**
C**GO BACK TO THE MENU
C**
      GO TO 20
C
C**
  330 CALL DYNOPT
C**
C
      WRITE(LUW,340)
  340 FORMAT(/,T20,'** THE OPTIMAL DYNAMIC DESIGN IS: **')
      WRITE(LUW,342)NWOPT,HWOPT,RKWOPT,FNWOPT,FHWOPT,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
  344 FORMAT(1X,10('*'),' LOSS-COST PER 100 HOURS = $',F11.3)
      WRITE(LUW,346)
  346 FORMAT(1X,71('*'),/,1X,71('*'))
C
      WRITE(LUW,348)
  348 FORMAT(' 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')
      READ(LUR,*)IYN8
      GO TO(350,20),IYN8
C
C**
C**FOR THE SECOND PASS OF OPTIMIZATION, SET THE STARTING POINT TO
C**THE BEST PINT FOUND SO FAR.
```

```
C**
  350 IN=NWOPT
      IH=HWOPT
      IK=RKWOPT
      NF=FNWOPT
      HF=FHWOPT
      KF=FKWOPT
C*.REPEAT THE OPTIMIZATION STARTING FROM THIS NEW POINT
      GO TO 300
C
C*
C___________ECON. EVALUATION IN THE WEIBULL ENVIRONMENT________________
C*
  390 WRITE(LUW,391)
  391 FORMAT(' *** FOR ECON. EVALUATION IN THE WEIBULL ENVIRONMENT,',/
     *  T5,' ENTER:',/,5X,'IN, IH, IK,  NF, HF, KF')
      READ(LUR,*)IN,IH,IK,NF,HF,KF
C**
C**CHECK TO SEE IF THESE ARE IN THE ACCEPTABLE RANGE
C**
      IF(IN.LT.1000 .AND.IN.GE.2 ) GO TO 394
          WRITE(LUW,393)
  393     FORMAT(' !?!  ERROR -- IN SHOULD BE BETWEEN 2 AND 1000'
     *           ,/,' DO IT OVER !')
          GO TO 390
  394 IF(IH.GT.0.0 .AND.IH.LT.100. ) GO TO 398
          WRITE(LUW,396)
  396     FORMAT(' !?!  ERROR -- IH SHOULD BE BETWEEN 0.0 AND 100.'
     *           ,/,' DO IT OVER !')
          GO TO 390
  398 IF( IK.GT.0.0 .AND. IK.LT.12. ) GO TO 402
          WRITE(LUW,399)
  399     FORMAT(' !?!  ERROR -- IK SHOULD BE BETWEEN 0.0 AND 12.'
     *           ,/,' DO IT OVER !')
  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
      GO TO 455
C
  432     WRITE(LUW,433)
  433     FORMAT(' !?!  ERROR -- NF, NH, AND NK SHOULD BE BETWEEN '
     *    ,'0.0 AND 2.',/,' DO IT OVER !')
          GO TO 390
C**
C**ECHOPRINT THE VALUES FOR CHECK
C**
  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)
  459 WRITE(LUW,460)
  460 FORMAT(' CORRECT ?   1=YES  2=NO  3=RETURN TO THE PREVIOUS MENU')
      READ(LUR,*)IYN3
      GO TO (480,390,20),IYN3
      WRITE(LUW,461)
  461 FORMAT(' !?! ERROR -- DO IT OVER !')
      GO TO 459
C*
C*
C_._._._WHEN THE DESIGN IS ACCEPTED_._._
C*
C_._._._THEN SUGGEST THE QUANTILE_._._._._._._._
C*
  480 PROBPT=.99D0
  482 WRITE(LUW,484)PROBPT
  484 FORMAT(' QUANTILE VALUE OF ',F11.9,' IS USED.',
     *        /,' YOU ACCEPT THIS.')
```

```
  485 WRITE(LUW,486)
  486 FORMAT(' CORRECT ?   1=YES  2=NO  3=RETURN TO THE PREVIOUS MENU')
      READ(LUR,*)IYN5
      GO TO (500,490,20),IYN5
      WRITE(LUW,488)
  488 FORMAT(' !?! ERROR -- DO IT OVER !')
      GO TO 485
C_._._._._._._IF QUANTILE VALUE IS NOT ACCEPTED_._._._.
  490 WRITE(LUW,493)
  493 FORMAT(' ENTER YOUR DESIRED QUANTILE:')
      READ(LUR,*)PROBPT
      GO TO 482
C_._._._._._._._._._._._._._._._._._._._._._._._._._._._.
C**
C**ECONOMICALLY EVALUATE THIS DESIGN
C**
  500 WRITE(LUW,521)
  521 FORMAT(1X,71('*'),/,1X,14('*'),2X,'ECON. EVALUATION IN ',
     *   'WEIBULL ENVIRONMENT',2X,14('*'),/)
      WRITE(LUW,523)THETA,ETA,WBMEAN
  523 FORMAT(T5,' DISTRIBUTION INFORMATION;',/,
     *T7,' WEIBULL W/ THETA=',F10.6,' AND ETA=',F10.6,' => MEAN=',F10.4)
C
      WRITE(LUW,525)DELTA,B,C,DD,E,VZMV1,T,W
  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)
C
      WRITE(LUW,526)IN,IH,IK,NF,HF,KF
  526 FORMAT(T5,' *** THE DESIGN TO BE EVALUATED IS: ***',/,
     *    T6,'  IN=',I4,5X,'  IH=',F10.4,'  IK=',F10.4,/,
     *    T6,'  NF=',F9.7,'  HF=',F9.7,1X,'  KF=',F9.7,/)
      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')
      READ(LUR,*)IYN9
      GO TO (580,541),IYN9
      WRITE(LUW,538)
  538 FORMAT(' !?! ERROR -- DO IT OVER !')
      GO TO 315
  541 WRITE(LUW,542)
  542 FORMAT(1X,71('*'),/)
C**
C**GO BACK TO THE MENU
C**
      GO TO 20
C
C**
C**CALCULATE ISTEPS FOR THE DESIGN TO BE EVALUATED.
C**
  580 ISTEPS=CSTPW(HF,IH)
      ISTPP=ISTEPS
C*.
C*.SINCE WE CANNOT EVALUATE EXACTLY THE SAME DESIGN, FOR
C*.THE GIVEN QUANTILE VALUE, GIVE A RANGE
C*.OF LOSS-COST VALUES WHICH INCLUEDS THE LOSS-COST OF THE DESIRED
C*.DESIGN.
C*.
C..CALCULATE THE IH DETERMINED BY THE ABOVE ISTEPS AND QUANTILE
C..
      TEMPIH=IH
      IH=SINTW(HF,ISTEPS)
C
      DO 581 I=1,2
```

```
C**
          CALL DYMEVA
C**
          WRITE(LUW,584)PROBPT,IN,IH,IK,NF,HF,KF
  584     FORMAT(T5,'*** FOR THE FOLLOWING DESIGN QUANTILE IS FIXED AT '
     *    , F10.8 ,/,T6,'  IN=',I4,5X,'  IH=',F10.4,'  IK=',F10.4,/,
     *       T6,'  NF=',F9.7,'  HF=',F9.7,1X,'  KF=',F9.7)
          WRITE(LUW,587)DYMLCS
  587     FORMAT(T5,' LOSS-COST PER 100 HOURS = $',F11.3,/)
          ISTEPS=ISTEPS+1
          ISTPP=ISTEPS
          IH=SINTW(HF,ISTEPS)
  581 CONTINUE
C*.
C*.NOW EVALUATE EXACTLY THE SAME DESIGN AS THE USER WANTS WHILE
C*.ACHIEVING A SLIGHTLY DIFFERENT QUANTILE.
C*.
      IH=TEMPIH
C**
      CALL DYMEVA
C**
      WRITE(LUW,588)CUPROX,IN,IH,IK,NF,HF,KF
  588 FORMAT(T5,'*** FOR THE FOLLOWING DESIGN, THE ACTUAL QUANTILE'
     *,' IS ', F10.8 ,/,T6,'  IN=',I4,5X,'  IH=',F10.4,'  IK=',F10.4,/,
     *       T6,'  NF=',F9.7,'  HF=',F9.7,1X,'  KF=',F9.7)
      WRITE(LUW,587)DYMLCS
C**
      WRITE(LUW,589)
  589 FORMAT(1X,71('*'),/,1X,71('*'))
C**
C**RETURN TO THE MAIN MENU
C**
      GO TO 20
C
C**
C**EXIT; RETURN TO THE MAIN MENU
C**
  600 RETURN
      END
C
C
C
C
C
C*********************************************************************
C*********************************************************************
C*********************************************************************
      SUBROUTINE DUNC
C*********************************************************************
C**                                                                  *
C** THIS ROUTINE PROMPTS THE USER FOR THE NECESSARY INFORMATION      *
C** NEEDED FOR DUNCAN'S X-BAR CHART DESIGN OR EVALUATION.            *
C**                                                                  *
C** THIS ROUTINE CALLS THE FOLLOWING SUBROUTINES:                    *
C**       DUNOPT-- TO OPTIMZE DUNCAN'S MODEL.                        *
C**       DUNEVA-- EVALUATE DUNCAN'S COST MODEL FOR A GIVEN DESIGN.  *
C**                                                                  *
C**                                                                  *
C**                                                                  *
C**                                                                  *
C*********************************************************************
C**
C*
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*8 LAMBDA
      COMMON / MAIN1 /LUR,LUW
```

```
      COMMON / DUNC1 / LAMBDA
      COMMON / DUNC4 / N,H,RK
      COMMON / DUNC5 / NDCOPT,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
C*
C*
C_________ENTER DISTRIBUTION, COST, AND OTHER PARAMETERS_______________
C*
   10 WRITE(LUW,11)
   11 FORMAT(/,T5,'  >> EXPONENTIAL ENVIRONMENT <<',/,
     *      55H *** FOR DUNCAN'S ECONOMIC X-BAR CHART, ENTER VALUES:
     * ,/,T5,' LAMBDA, DELTA, B, C, D, E, M, T, W')
      READ(LUR,*)LAMBDA, DELTA, B,C,DD,E,VZMV1, T, W
C**
C**CALCULATE MEAN AND VARIANCE OF THE EXPONENTIAL
C**
      XPMEAN=1.D0/LAMBDA
      XPVAR=XPMEAN/LAMBDA
C**
      WRITE(LUW,14)LAMBDA,XPMEAN,DELTA,B,C,DD,E,VZMV1,T,W
   14 FORMAT(' VALUES ENTERED ARE:',/,
     *         T5,' DISTRIBUTION INFORMATION;',/,
     *T7,' EXPONENTIAL W/ LAMBDA= ',F10.4,' => MEAN=',F10.4,/,
     *         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)
C*
C_________SELECTION FOR DESIGN, EVALUATION , ETC._____________________
C*
   18 WRITE(LUW,19)
   19 FORMAT(/,' *** ENTER OPTION NUMBER',/,
     *   T5, ' 1 = ECON. DESIGN OF DUNCAN,S X-BAR CHART (OPTIMIZATION)'
     * ,/,T5,' 2 = ECON. EVALUATION IN THE EXPONENTIAL ENVIRONMENT',/,
     *   T5,' 3 = RETURN TO REVISE COST AND DISTRIBUTION PARAMETERS',/,
     *   T5,' 4 = RETURN TO THE MAIN MENU')
      READ(LUR,*)MENU2
      GO TO (100,200,10,400),MENU2
      WRITE(LUW,20)
   20 FORMAT(' !?! ERROR -- DO IT OVER !')
      GO TO 18
C*
C_________ECON. DESIGN (OPTIMIZATION) OF DUNCAN'S X-BAR CHART__________
C*
C**
C**INITIALIZATION OF STARTING POINT FOR OPTIMIZATION
C**
  100 N=5
      H=1.D0
      RK=3.D0
      WRITE(LUW,102)N,H,RK
  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.')
  103 WRITE(LUW,104)
  104 FORMAT(' CORRECT ?   1=YES  2=NO  3=RETURN TO THE PREVIOUS MENU')
      READ(LUR,*)IYN1
      GO TO (110,150,18),IYN1
      WRITE(LUW,106)
  106 FORMAT(' !?! ERROR -- DO IT OVER !')
      GO TO 103
C*
C_._._._IF THE SUGGESTED STARTING POINT IS ACCEPTED_._._
C*
```

```
  110 WRITE(LUW,111)
  111 FORMAT(1X,66('*'),/,1X,13('*'),2X,24HECON. DESIGN OF DUNCAN'S
     * , ' X-BAR CHART',2X,13('*'),/)
C
      WRITE(LUW,112)LAMBDA,XPMEAN,DELTA,B,C,DD,E,VZMV1,T,W
  112 FORMAT(' VALUES ENTERED ARE:',/,
     *        T5,' DISTRIBUTION INFORMATION;',/,
     *T7,' EXPONENTIAL W/ LAMBDA= ',F10.4,' => MEAN=',F10.4,/,
     *        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,/)
  115 WRITE(LUW,116)N,H,RK
  116 FORMAT(T5,' STARTING POINT FOR OPTIMIZATION IS:',/,1X,T6,' N=',I4,
     *   '  H=', F8.4, '  K=',F8.4,/,' CHECK THE ABOVE INFORMATION.',/,
     * ' EVERYTHING IS CORRECT ?   1=YES  2=NO')
      READ(LUR,*)IYN2
      GO TO (130,120),IYN2
      WRITE(LUW,118)
  118 FORMAT(' !?! ERROR -- DO IT OVER !')
      GO TO 115
  120 WRITE(LUW,121)
  121 FORMAT(1X,66('*'),/)
      GO TO 18
C
C**
  130 CALL DUNOPT
C**
C
      WRITE(LUW,132)
  132 FORMAT(T10,'N',T16,'H',T26,'K',T34,'LOSS-COST',/)
C**
C**PRINT OPTIMIZATION ITERATIONS
C**
      DO 133 I=1,20
C*.
C*.IF NO MORE INFORMATION IS AVAILABLE IN THE ARRAYS, THEN QUIT THE LOOP
C*.
         IF(FTPRNT(I).LT.1.E-10) GO TO 142
             WRITE(LUW,135)NTPRNT(I),HTPRNT(I),RKPRNT(I),FTPRNT(I)
  135        FORMAT(T8,I3,T13,F7.4,T23,F7.4,T33,F11.3)
  133 CONTINUE
C*
  142 WRITE(LUW,143)
  143 FORMAT(/,T10,38H ** THE OPTIMAL DUNCAN'S DESIGN IS: **      )
      WRITE(LUW,145)NDCOPT,HDCOPT,RKDCOP
  145 FORMAT(T10,'  N=',I4,'  H=',F8.4,'  K=',F8.4,/)
      WRITE(LUW,147)FDCOPT
  147 FORMAT(1X,7('*'),' THE MIN. LOSS-COST PER 100 HOURS = $',F11.3)
      WRITE(LUW,149)
  149 FORMAT(1X,66('*'),/1X,66('*'),/)
C**
C**GO BACK TO MENU
C**
      GO TO 18
C*
C_._._._._._._IF THE SUGGESTED STARTING POINT IS NOT ACCEPTED_._.
C*
  150 WRITE(LUW,151)
  151 FORMAT(T5,' FOR YOUR DESIRED STARTING POINT FOR OPTIMIZATION, ',
     *    'ENTER:',/,5X,' N, H, K')
      READ(LUR,*)N,H,RK
C**
C**CHECK TO SEE IF THESE ARE IN THE ACCEPTABLE RANGE
C**
      IF( N.LT.1000 .AND. N.GE.2 ) GO TO 153
          WRITE(LUW,152)N
```

```
  152     FORMAT(' !?!  ERROR -- N SHOULD BE BETWEEN 2 AND 1000'
     *           ,/,' DO IT OVER !')
          GO TO 150
  153 IF( H.GT.0.0 .AND. H.LT.100. ) GO TO 155
          WRITE(LUW,154)H
  154     FORMAT(' !?!  ERROR -- H SHOULD BE BETWEEN 0.0 AND 100.'
     *           ,/,' DO IT OVER !')
          GO TO 150
  155  IF( RK.GT.0.0 .AND. RK.LT.12. ) GO TO 157
          WRITE(LUW,156)RK
  156     FORMAT(' !?!  ERROR -- K SHOULD BE BETWEEN 0.0 AND 12.'
     *           ,/,' DO IT OVER !')
          GO TO 150
C**
C**ECHOPRINT THE VALUES FOR CHECK
C**
  157 WRITE(LUW,158)N,H,RK
  158 FORMAT(T5,' VALUES ENTERED:  N=',I4,4X,'  H=',F8.4,4X,'  K=',F8.4)
  159 WRITE(LUW,160)
  160 FORMAT(' CORRECT ?   1=YES   2=NO   3=RETURN TO THE PREVIOUS MENU')
      READ(LUR,*)IYN3
      GO TO (110,150,18),IYN3
      WRITE(LUW,161)
  161 FORMAT(' !?! ERROR -- DO IT OVER !')
      GO TO 159
C*
C*
C___________ECON. EVALUATION IN THE EXPONENTIAL ENVIRONMENT___________
C*
  200 WRITE(LUW,201)
  201 FORMAT(' *** FOR ECON. EVALUATION IN THE EXPONENTIAL',
     *   'ENVIRONMENT,',/,T5,' ENTER VALUES:  N, H, K')
      READ(LUR,*)N,H,RK
      WRITE(LUW,205)N,H,RK
  205 FORMAT(T5,' VALUES ENTERED:  N=',I4,4X,'  H=',F8.4,4X,'  K=',F8.4)
  206 WRITE(LUW,208)
  208 FORMAT(' CORRECT ?   1=YES   2=NO   3=RETURN TO THE PREVIOUS MENU')
      READ(LUR,*)IYN4
      GO TO (220,200,18),IYN4
      WRITE(LUW,209)
  209 FORMAT(' !?! ERROR -- DO IT OVER !')
      GO TO 206
C**
C**ECONOMICALLY EVALUATE THIS DESIGN
C**
  220 WRITE(LUW,221)
  221 FORMAT(1X,66('*'),/,1X,9('*'),2X,'ECON. EVALUATION IN ',
     *   'EXPONENTIAL ENVIRONMENT',2X,9('*'),/)
C
      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,/,
     *        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)
      WRITE(LUW,226)N,H,RK
  226 FORMAT(T5,' *** THE DESIGN TO BE EVALUATED IS: ***',/,T6,'  N=',
     * I4,'  H=',F8.4, '  K=',F8.4,/,' CHECK THE ABOVE INFORMATION.')
  227 WRITE(LUW,228)
  228 FORMAT(' EVERYTHING IS CORRECT ?   1=YES  2=NO')
      READ(LUR,*)IYN5
      GO TO (250,240),IYN5
      WRITE(LUW,229)
  229 FORMAT(' !?! ERROR -- DO IT OVER !')
      GO TO 227
  240 WRITE(LUW,241)
```

```
  241 FORMAT(1X,66('*'),/)
      GO TO 18
C
C**
  250 CALL DUNEVA
C**
C
      WRITE(LUW,253)DCLCST
  253 FORMAT(1X,7('*'),' LOSS-CAST PER 100 HOURS= $',F11.3)
      WRITE(LUW,255)
  255 FORMAT(1X,66('*'),/,1X,66('*'),/)
C**
C**GO BACK TO THE MENU
C**
      GO TO 18
C**
C**RETURN TO THE MAIN MENU
C**
  400 RETURN
      END
C*
C*
C*
C*
C*********************************************************************
C*********************************************************************
C*********************************************************************
      SUBROUTINE DUNOPT
C*********************************************************************
C**                                                                 *
C** THIS SUBROUTINE OPTIMIZES DUNCAN'S COST MODEL USING ZXMIN ROUTINE *
C** PROVIDED IN INTERNATIONAL MATHEMATICAL SCIENTIFIC LIBRARY (IMSL). *
C**                                                                 *
C** OPTIMIZATION IS PERFORMED IN TWO STAGES. IN THE FIRST STAGE      *
C** THE SAMPLE SIZE IS TREATED AS A REAL-VALUED VARIABLE AND THE     *
C** LOSS-FUNCTION IS OPTIMIZED OVER ALL THREE VARIABLES. IN THE SECOND*
C** STAGE THE SAMPLE SIZE IS TREATED AS AN INTERGER-VALUED VARIABLE.  *
C** THUS, SAMPLE SIZE IS SET TO A TENTATIVE VALUE , AND THE LOSS-COST *
C** FUNCTION IS THEN OPTIMIZED OVER THE OTHER TWO REMAINING VARIABLES.*
C**                                                                 *
C** THE FOLLOWING SUBROUTINES ARE CALLED BY THIS SUBROUTINE:         *
C**      (1) SUBROUTINE ZXMIN                                        *
C**      (2) SUBROUTINE  FUNCT  (THROUGH ZXMIN)                      *
C**      (3) SUBROUTINE  FUNCT2 (THROUGH ZXMIN)                      *
C**                                                                 *
C********************************************************************
C**
C*
       IMPLICIT REAL*8(A-H,O-Z)
         EXTERNAL FUNCT
         EXTERNAL FUNCT2
       COMMON / MAIN1 /LUR,LUW
       COMMON / DUNC4 / INN,RH,RK
       COMMON / DUNC5 / NDCOPT,HDCOPT,RKDCOP,FDCOPT
       COMMON / SAMPLS /IX
      COMMON / DUNC6 / NTPRNT(20),HTPRNT(20),RKPRNT(20),FTPRNT(20)
       INTEGER N,NSIG,MAXFN,IOPT
       REAL*8 FMIN(5)
       REAL*4 X(3),H(6),G(3),W(9)
       REAL*4 F
       DATA HTPRNT,RKPRNT,NTPRNT,FTPRNT/20*0.D0,20*0.D0,20*0,20*0.D0/
C*
       NFUEVA=0
C*
C**N IS DIMENSIONALITY OF SEARCH
C**
```

```
C
C**FIRST OPTIMIZE OVER ALL THE VARIABLES ; CONSIDERING SAMPLE SIZE
C**AS A REAL VARIABLE RATHER THAN INTEGER.
C*
       N=3
       NSIG=3
       MAXFN=10000
       IOPT=0
C**
C**INITIALIZE THE VARIABLES
C**
         X(1)=RH
         X(2)=RK
         X(3)=DFLOAT(INN)
C*
         CALL ZXMIN(FUNCT,N,NSIG,MAXFN,IOPT,X,H,G,F,W,IER)
C*
C-----------------------------TWO DIMENSIONAL SEARCH-------------------
C
C**NOW OPTIMIZE OVER H AND K FOR KNOWN BUT DIFFERENT VALUES OF SAMPLE
C**SIZE.
C
      N=2
C**
C**SET SAMPLE SIZE TO THE INTEGER EQUIVALENT OF OPTIMUM FOUND IN
C**THE PREVIOUS SEARCH MINUS TWO.
C**
      IX=X(3)-2.
C*.
C*.MINIMUM SAMPLE SIZE POSSIBLE IS 2.
C*.
      IF(IX.LT.2)IX=2
C**
C**FIND THE OPTIMAL ECONOMIC DESIGN FOR THIS INTEGER SAMPLE SIZE
C**
C*.INITIALIZE THE VARIABLES
C
         X(1)=RH
         X(2)=RK
         IOPT=0
         NSIG=3
C
      CALL ZXMIN(FUNCT2,N,NSIG,MAXFN,IOPT,X,H,G,F,W,IER)
C**
C**KEEP THIS RESULTS IN A TEMPORARY ARRAY TO BE PRINTED ONLY IN THE
C**CASE THE USER IS INTERESTED IN DUNCAN'S OPTIMUM DESIGN.
C**
      I=1
      HTPRNT(I)=X(1)
      RKPRNT(I)=X(2)
      NTPRNT(I)=IX
      FTPRNT(I)=F
C**
C**INCR SHOWS THE DIRECTION OF SEARCH ALONG THE SAMPLE SIZE DIRECTION
C**
      INCR=1
      ITIME=0
C**
C**KEEP THE POINT AS THE BEST OPTIMUM SO FAR
C**
 7    FMIN(5)=F
      DO  8 I=1,2
          FMIN(I)=X(I)
 8    CONTINUE
C**
C**INCREMENT OR DECREMENT SAMPLE SIZE BASED ON DIRECTION OF INCR
```

```
C**
 9    IX=IX+INCR
C**
C**FIND THE OPTIMAL DESIGN FOR THIS VALUE OF SAMPLE SIZE
C**
C*.INITIALIZE THE VARIABLES
C
10         X(1)=RH
           X(2)=RK
           IOPT=0
           NSIG=3
           G(1)=0.
           G(2)=0.
           F=0
      CALL ZXMIN(FUNCT2,N,NSIG,MAXFN,IOPT,X,H,G,F,W,IER)
C**
C**KEEP THE RESULTS IN A TEMPORARY ARRAY TO BE PRINTED ONLY IN THE
C**CASE THE USER IS INTERESTED IN DUNCAN'S OPTIMUM DESIGN.
C**
      I=I+1
      HTPRNT(I)=X(1)
      RKPRNT(I)=X(2)
      NTPRNT(I)=IX
      FTPRNT(I)=F
C**
C**IF THIS IS THE FIRST INCREMENT IN SAMPLE SIZE...
C**
      IF(ITIME.EQ.1) GO TO 23
C*.
C*.AND THERE IS AN IMPROVEMENT IN OBJECTIVE FUNCTION
C*.
         IF(F .GT.FMIN(5)) GO TO 13
C..THEN UPDATE THE BEST OPTIMUM
             ITIME=1
             FMIN(5)=F
             DO 11 I=1,2
                 FMIN(I)=X(I)
 11          CONTINUE
C..INCREMENT SAMPLE SIZE AND REPEAT
             IX=IX+1
             GO TO 10
C*.
C*.IF THE OBJECTIVE FUNCTION GETS WORSE , THEN SWITCH DIRECTION .
C*.ALSO FOR THIS FIRST SAMPLE SIZE DECREMENT IT BY 2.
C*.
 13      INCR=-INCR
         IX=IX-2
         GO TO 10
C**
C**IF THERE IS AN IMPROVEMENT IN OBJ. FUN. AND THIS IS NOT THE FIRST
C**INCREMENT ON SAMPLE SIZE
C**THEN UPDATE FMIN AND KEEP GOING IN THIS DIRECTION
C**
 23   IF(F .LE.FMIN(5)) GO TO 7
C**
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.
C**
      IXMIN=IX-INCR
C*.
C*.STORE DUNCAN'S MODEL OPTIMAL DESIGN IN THE FOLLOWING:
C*.
        NDCOPT=IXMIN
        HDCOPT=FMIN(1)
        RKDCOP=FMIN(2)
```

```
C*.
C*.FDCOPT IS LOSS-COST PER 100 HOURS
C*.
       FDCOPT=FMIN(5)*100.D0
       RETURN
       END
C
C**********************************************************************
C**********************************************************************
C**********************************************************************
       SUBROUTINE FUNCT(N,X,F)
C***********************************************************************
C**                                                                   *
C** THIS SUBROUTINE CALLS DIFFERENT SUBROUTINES TO CALCULATE DUNAN'S  *
C** LOSS-COST FUNCTION NEEDE FOR 3-DIMENSIOAL SEARCH OVER H, K, AND N *
C**                                                                   *
C** THE FOLLOWING SUBROUTINES ARE CALLED BY THIS ROUTINE:             *
C**   (1) SUBROUTINE PD TO CALCULATE PROBABILITY OF DETECTEING.       *
C**   (2) SUBROUTINE ALFA TO CALCULATE PROB. OF FALSE ALARM.          *
C**                                                                   *
C***********************************************************************
C**
C*
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*8 LAMBDA
      COMMON / MAIN1 /LUR,LUW
      COMMON / DUNC1 / LAMBDA
      COMMON / DCDY1 / DELTA, B,C,DD,E,VZMV1,T,W
         INTEGER N
         REAL*4  X(N)
         REAL*4 F
C
 123   CONTINUE
C**FIRST MAKE THE SINGLE PRECISION VALUES OF X(1), X(2), AND X(3)
C**DOUBLE PRECISION (FOR MORE ACCURATE CALCULATION) BY ASSIGNING
C**THEM TO DX1, DX2, AND DX3. THIS IS DONE BECAUSE THE IMSL ROUTINE
C**IS SINGLE PRECISION
       DX1=X(1)
       DX2=X(2)
       DX3=X(3)
C++
C++CHECK IF ANY OF THE DECISION VARIABLES ARE OUT OF RANGE
C++THEN RETURN WITH A BIG VALUE FOR F
C++
       IF(DX1.GT.70..OR.DX2.GT.12.)F=100000000.
       IF(DX1.GT.70..OR.DX2.GT.12.)RETURN
       IF(DX1.LT.0..OR.DX2.LT.0.)F=100000000.
       IF(DX1.LT.0..OR.DX2.LT.0.)RETURN
       IF(DX3.LT.1.)F=100000000.
       IF(DX3.LT.1.)RETURN
C**
C**CALCULATE THE AVG. TIME OF OCCURANCE W/IN THE NH TO (N+1)H TIME INTER
C** INTERVAL.
C**
           DXPH=DEXP(-LAMBDA*DX1)
      ATWIN=(1.D0-(1.D0+LAMBDA*DX1)*DXPH)/(LAMBDA*(1.D0-DXPH))
C**
C**CALCULATE PD; PROBABILITY OF DETECTING THE SHIFT.
C**
      CALL PROBD(DX2,DX3,DELTA,PD)
C**
C**CALCULATE APP. APP IS AVG. TIME OUT OF CONTROL BEFORE A SAMPLE
C**FALLS OUTSIDE THE CONTROL LIMIT ( EXCLUDING ATWIN.)
C**
      APP=DX1/PD-ATWIN+E*DX3
C**
```

```
C**CALCULATE CYCLE LENGTH
C**
      CYCLE=1.D0/LAMBDA+APP+DD
C**
C**CALCULATE ALPHA ; PROB. OF FALSE ALARM
C**
      CALL PROFA(DX2,ALPHA)
C**
C**CALCULATE POOC ; PROPORTION OF TIME OUT OF CONTROL
C**
       PINC=1.D0/(LAMBDA*CYCLE)
       POOC=1.D0-PINC
C**
C**CALCULATE ENFALSE ; EXPECTED NUMBER OF FALSE ALARMS
C**
       ENFALS=ALPHA*DXPH/(1.D0-DXPH)
C**
C**CALCULATE THE LOSS-COST PER HOUR OF OPERATION.
C**
       BAHMC=(B+C*DX3)/DX1
       RLOSSC=POOC*VZMV1+T*ENFALS/CYCLE+W/CYCLE+BAHMC
       F=RLOSSC
C
       RETURN
       END
C
C
C**********************************************************************
C**********************************************************************
C**********************************************************************
        SUBROUTINE FUNCT2(N,X,F)
C**********************************************************************
C**                                                                   *
C** THIS SUBROUTINE IS EQUIVALENT OF FUNCT AS IS NEEDE FOR            *
C** TWO-DIMENSIONAL SEARCH.                                           *
C**                                                                   *
C** THE FOLLOWING SUBROUTINES ARE CALLED BY THIS SUBROUTINE:          *
C**     (1) SUBROUTINE PD TO CALCULATE PROBABILITY OF DETECTEING.     *
C**     (2) SUBROUTINE ALFA TO CALCULATE PROB. OF FALSE ALARM.        *
C**                                                                   *
C**********************************************************************
C**
C*
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*8 LAMBDA
      COMMON / MAIN1 /LUR,LUW
      COMMON / DUNC1 / LAMBDA
      COMMON / DCDY1 / DELTA, B,C,DD,E,VZMV1,T,W
       COMMON / SAMPLS /IX
         INTEGER N
         REAL*4  X(N)
         REAL*4 F
C**
C**FIRST MAKE THE SINGLE PRECISION VALUES OF X(1), X(2), AND X(3)
C**DOUBLE PRECISION (FOR MORE ACCURATE CALCULATION) BY ASSIGNING
C**THEM TO DX1, DX2, AND DX3. THIS IS DONE BECAUSE THE IMSL ROUTINE
C**IS SINGLE PRECISION
C**
        DX1=X(1)
        DX2=X(2)
        DX3=IX
C++
C++CHECK IF ANY OF THE DECISION VARIABLES ARE OUT OF RANGE
C++THEN RETURN WITH A BIG VALUE FOR F
C++
        IF(DX1.GT.70..OR.DX2.GT.12.)F=100000000.
```

```
      IF(DX1.GT.70..OR.DX2.GT.12.)RETURN
      IF(DX1.LT.0..OR.DX2.LT.0.)F=100000000.
      IF(DX1.LT.0..OR.DX2.LT.0.)RETURN
      IF(DX3.LT.1.)F=100000000.
      IF(DX3.LT.1.)RETURN
C**
C**CALCULATE THE AVG. TIME OF OCCURANCE W/IN THE NH TO (N+1)H TIME INTER
C** INTERVAL.
C**
          DXPH=DEXP(-LAMBDA*DX1)
     ATWIN=(1.D0-(1.D0+LAMBDA*DX1)*DXPH)/(LAMBDA*(1.D0-DXPH))
C**
C**CALCULATE PD; PROBABILITY OF DETECTING THE SHIFT
C**
     CALL PROBD(DX2,DX3,DELTA,PD)
C**
C**CALCULATE APP. APP IS AVG. TIME OUT OF CONTROL BEFORE A SAMPLE
C**FALLS OUTSIDE THE CONTROL LIMIT ( EXCLUDING ATWIN.)
C**
     APP=DX1/PD-ATWIN+E*DX3
C**
C**CALCULATE CYCLE LENGTH
C**
     CYCLE=1.D0/LAMBDA+APP+DD
C**
C**CALCULATE ALPHA ; PROB. OF FALSE ALARM
C**
     CALL PROFA(DX2,ALPHA)
C**
C**CALCULATE POOC ; PROPORTION OF TIME OUT OF CONTROL
C**
      PINC=1.D0/(LAMBDA*CYCLE)
      POOC=1.D0-PINC
C**
C**CALCULATE ENFALSE ; EXPECTED NUMBER OF FALSE ALARMS
C**
      ENFALS=ALPHA*DXPH/(1.D0-DXPH)
C**
C**CALCULATE THE LOSS-COST PER HOUR OF OPERATION.
C**
      BAHMC=(B+C*DX3)/DX1
      RLOSSC=POOC*VZMV1+T*ENFALS/CYCLE+W/CYCLE+BAHMC
      F=RLOSSC
C
      RETURN
      END
C
C
C
C
C***********************************************************************
C***********************************************************************
C***********************************************************************
       SUBROUTINE DUNEVA
C***********************************************************************
C**                                                                    *
C** THIS SUBROUTINE IS USED TO EVALUATE DUNCAN'S COST FUNCTION FOR THE*
C** GIVEN VALUES OF N, H, AND K.                                       *
C**                                                                    *
C** THE FOLLOWING SUBROUTINES ARE CALLED BY THIS SUBROUTINE:           *
C**     (1) SUBROUTINE PD TO CALCULATE PROBABILITY OF DETECTEING.      *
C**     (2) SUBROUTINE ALFA TO CALCULATE PROB. OF FALSE ALARM.         *
C**                                                                    *
C***********************************************************************
C**
C*
```

```
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*8 LAMBDA
      COMMON / MAIN1 /LUR,LUW
      COMMON / DUNC1 / LAMBDA
      COMMON / DUNC4 / N,H,RK
      COMMON / DUNC7 / DCLCST
      COMMON / DCDY1 / DELTA, B,C,DD,E,VZMV1,T,W
C
C**
C**DX1, DX2, DX3 CORRESPOND TO H, K, AND N, RESPECTIVELY.
C**
        DX1=H
        DX2=RK
        DX3=N
C**
C**CALCULATE THE AVG. TIME OF OCCURANCE W/IN THE NH TO (N+1)H TIME INTER
C** INTERVAL.
C**
            DXPH=DEXP(-LAMBDA*DX1)
      ATWIN=(1.D0-(1.D0+LAMBDA*DX1)*DXPH)/(LAMBDA*(1.D0-DXPH))
C**
C**CALCULATE PD; PROBABILITY OF DETECTING THE SHIFT
C**
      CALL PROBD(DX2,DX3,DELTA,PD)
C**
C**CALCULATE APP. APP IS AVG. TIME OUT OF CONTROL BEFORE A SAMPLE
C**FALLS OUTSIDE THE CONTROL LIMIT ( EXCLUDING ATWIN.)
C**
      APP=DX1/PD-ATWIN+E*DX3
C**
C**CALCULATE CYCLE LENGTH
C**
      CYCLE=1.D0/LAMBDA+APP+DD
C**
C**CALCULATE ALPHA ; PROB. OF FALSE ALARM
C**
      CALL PROFA(DX2,ALPHA)
C**
C**CALCULATE POOC ; PROPORTION OF TIME OUT OF CONTROL
C**
       PINC=1.D0/(LAMBDA*CYCLE)
       POOC=1.D0-PINC
C**
C**CALCULATE ENFALSE ; EXPECTED NUMBER OF FALSE ALARMS
C**
       ENFALS=ALPHA*DXPH/(1.D0-DXPH)
C**
C**CALCULATE THE LOSS-COST PER HOUR OF OPERATION.
C**
       BAHMC=(B+C*DX3)/DX1
       RLOSSC=POOC*VZMV1+T*ENFALS/CYCLE+W/CYCLE+BAHMC
       DCLCST=100.D0*RLOSSC
C
       RETURN
       END
C
C***********************************************************************
C***********************************************************************
      SUBROUTINE PROBD(RK,RN,DELTA,PD)
C***********************************************************************
C*                                                                    *
C** THIS SUBROUTINE CALCULATES PD ; PROB. OF DETECTING THE SHIFT.     *
C*                                                                    *
C***********************************************************************
C
      IMPLICIT REAL*8(A-H,O-Z)
```

```
C
      Y=RK-DELTA*DSQRT(RN)
      CALL MDNORD(Y,P)
C*
C*PD IS PROB. OF DETECTING THE SHIFT.( IT IS ASSUMED THAT THE SHIFT
C*  IS POSITIVE.)
C*
      PD=1.D0-P
C
      RETURN
      END
C
C
C**********************************************************************
C**********************************************************************
      SUBROUTINE PROFA(RK,ALPHA)
C**********************************************************************
C*                                                                    *
C** THIS SUBROUTINE CALACULATES ALPHA ; PROB. OF FALSE ALARM.         *
C*                                                                    *
C**********************************************************************
C
      IMPLICIT REAL*8(A-H,O-Z)
C
      CALL MDNORD(RK,P)
C**
C**ALPHA IS PROB. OF FALSE ALARM.
C**
      ALPHA=2.D0*(1.D0-P)
C
      RETURN
      END
C
C
C
C***********************************************************************
C***********************************************************************
C***********************************************************************
      SUBROUTINE DYNOPT
C***********************************************************************
C**                                                                    *
C** THIS SUBROUTINE OPTIMIZES THE DYNAMIC LOSS-COST MODEL.             *
C**                                                                    *
C** THE FOLLOWING SUBROUTINES ARE CALLED BY THIS SUBROUTINE:           *
C**     (1) SUBROUTINE TWOSCH TO PERFORM A 2-DIMENSIONAL SEARCH.       *
C**     (2) SUBROUTINE SINTW TO CALCULATE ISTEPS.                      *
C**                                                                    *
C***********************************************************************
C**
C*
C**
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*8 NF,IH,HF,IK,KF
      COMMON / MAIN1 /LUR,LUW
      COMMON / DYNM2 / ISTEPS
      COMMON / DYNM3 / IN, NF,IH,HF,IK,KF
      COMMON / DYNM4 / PROBPT
      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
      COMMON / DYOPT3 / TMPMIN(10)
      COMMON / DYOPT4 / NWOPT,HWOPT,RKWOPT,FNWOPT,FHWOPT,FKWOPT,YFWOPT
      DATA  TMPMIN/10*999999.D0/
C**
C**
```

```
C**NOTE THAT VARIABLES: XXX(1), XXX(2), XXX(3), XXX(4), XXX(5), XXX(6)
C**CORRESPOND TO:        HF    , ISTEPS,  IK   ,  KF   ,  IN   ,  NF ,
C**RESPECTIVELY.
C**
C
C**
C**INITIALIZE ALL THE VARIABLES TO THEIR STARTING POINT SET IN ROUTINE
C**DYNM.
C**
      XXX(1)=HF
      XXX(2)=ISTEPS
      XXX(3)=IK
      XXX(4)=KF
      XXX(5)=IN
      XXX(6)=NF
C**
C**INITIALIZE THE SIGN (DIRECTION) OF SEARCH FOR EACH VARIABLE
C**
      DO 25 I=1,6
         ISIDEL(I)=+1
   25 CONTINUE
C**
C**OPTIMIZE OVER THE FIRST TWO VARIABLES. THAT IS, HF AND ISTEPS.
C**
      IVARL=2
      CALL TWOSCH(IVARL)
C*.
C*.SET VARIABLES TO THEIR BEST OPTIMUMFOUND SO FAR. (THIS IS HELPFUL
C*.ESPECIALLY IF ITERMAX1 IS REACHED.)
C*.
      DO 35 I=1,6
         XXX(I)=TMPMIN(I)
   35 CONTINUE
      YYYF=TMPMIN(10)
C**
C**OPTIMIZE OVER THE VARIABLES KF AND IK
C**
      IVARL=4
      CALL TWOSCH(IVARL)
C*.
C*.SET VARIABLES TO THEIR BEST OPTIMUMFOUND SO FAR. (THIS IS HELPFUL
C*.ESPECIALLY IF ITERMAX2 IS REACHED.)
C*.
      DO 40 I=1,6
         XXX(I)=TMPMIN(I)
   40 CONTINUE
      YYYF=TMPMIN(10)
C**
C**OPTIMIZE OVER THE VARIABLES NF AND IN
C**
      IVARL=6
      CALL TWOSCH(IVARL)
C*.
C*.SET VARIABLES TO THEIR BEST OPTIMUMFOUND SO FAR. (THIS IS HELPFUL
C*.ESPECIALLY IF ITERMAX3 IS REACHED.)
C*.
      DO 45 I=1,6
         XXX(I)=TMPMIN(I)
   45 CONTINUE
      YYYF=TMPMIN(10)
C**
C**NOTE THAT WHEN RETURNING FROM THIS ROUTINE XXX,S AND YYYF AS WELL
C**AS THE ARRAY TMPMIN CONTAIN THE OPTIMAL DYNAMIC DESIGN. ALSO,
C**KEEP THE OPTIMUM DESIGN IN THE FOLLOWING:
C**                  NWOPT,HWOPT,RKWOPT,FNWOPT,FHWOPT,FKWOPT,  YFWOPT
C** CORRESPONDING TO: IN    IH    IK     NF     HF     KF       YYYF
```

```
C**
      FHWOPT=TMPMIN(1)
      HWOPT=SINTW(FHWOPT,ISTEPS)
      RKWOPT=TMPMIN(3)
      FKWOPT=TMPMIN(4)
      NWOPT=TMPMIN(5)
      FNWOPT=TMPMIN(6)
      YFWOPT=100.D0*TMPMIN(10)
C
      RETURN
      END
C
C
C
C
C
C*********************************************************************
C*********************************************************************
C*********************************************************************
C
C
      SUBROUTINE TWOSCH(IVARL)
C*********************************************************************
C**                                                                 *
C** THIS SUBROUTINE OPTIMIZES PERFORMS A TWO-DIMENSIONAL SEARCH.     *
C**                                                                 *
C** THE FOLLOWING SUBROUTINES ARE CALLED BY THIS SUBROUTINE:        *
C**     (1) SUBROUTINE COGGIN TO PERFORMA PRECISE LINE SEARCH USING *
C**         COGGINS' METHOD.                                        *
C**     (2) SUBROUTINE OMYSCH WHICH IS USED IN CONJUNCTON WITH THIS *
C**         ROUTINE TO PERFORM A TWO-AT-A-TIME SEARCH.              *
C**     (3) SUBROUTINE RUNC WHICH CALCULATES THE LOSS-COST OF THE   *
C**         DYNAMIC MODEL.                                          *
C**                                                                 *
C**                                                                 *
C*********************************************************************
C**
C
C
C**NOTE THE FOLLOWING VARIABLES DEFINITION USED IN THIS ROUTINE.    **
C
C
C   XQLIM(I) : QUITTING LIMIT OR DESIRED ACCURRACY FOR
C              VARIABLE I.
C
C   DELMN(I) : MIN. LIMIT ON STEP SIZE , FOR VARIABLE I.
C
C   DELMX(I) : MAX. LIMIT ON STEP SIZE , FOR VARIABLE I.
C
C   DEL(I)   : CURRENT STEP SIZE , FOR VARIABLE I.
C
C   ISIDEL(I): SIGN OF DEL FOR THE LAST SUCCESSFUL MOVE ALONG
C              VARIABLE I.
C
C   XXX'S    : DECISION VARIABLES; HF, ISTEPS,IK,KF,IN,NF
C              ITS FINAL VALUE CONTAINS THE LAST POINT TRIED. THUS,
C              IT MIGHT NOT BE THE OPTIMUM POINT.
C
C   IH       : INITIAL STEP SIZE . THIS IS CALCULATE IN ROUTINE RUNC
C              FOR ANY GIVEN VALUES OF HF AND ISTEPS.
C
C   NFTERM   : THE MAX. NUMBER OF OBJ. FUNCTION EVALUATIONS ALLOWED.
C
C
C**
C
```

```
C
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON / MAIN1 /LUR,LUW
      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
      DIMENSION TEMP(10)
      DIMENSION ITRMAX(3)
C**
      IFLAG=1
      IDIRC=IVARL
C**
C**NFTERM IS MAX. NUMBER OF OBJ. FUN. EVALUATIONS ALLOWED.
C**
C**NFUEVA ; NUMBER OF OBJ. FUN. EVALUATED SO FAR IS 0.
C**
      ITRMAX(1)=ITRMX1
      ITRMAX(2)=ITRMX2
      ITRMAX(3)=ITRMX3
      IFTR=IVARL/2
      NFTERM=ITRMAX(IFTR)
      NFUEVA=0
C**
      IF(IVARL.NE.2)GO TO 88
C**
C**FOR THE FIRST VARIABLE ( HF) USE COGGIN TO DO A LINE SEARCH
C**
      IDIRC=1
      CALL COGGIN(IDIRC)
C*.
C*.CHECK DEL AFTER COGGIN,S EXECUTION
C*.
C*.IF THE LAST DEL USED IN COGGIN IS LESS THAN DELMN ,THEN USE DELMN BUT
C*.RESERVE THE SIGN OF DEL ( THE LAST SUCCESFUL STEP TAKEN IN COGGIN)
C*.FOR THE NEXT STEP TAKEN ALONG THAT VARIABLE
C*.
      IF(DABS(DEL(IDIRC)).LT.DELMN(IDIRC))DEL(IDIRC)=DSIGN(DELMN(IDIRC),
     *                                                      DEL(IDIRC))
C*.
      IDIRC=2
C**
C**NOTE THAT FOR IVARL=4 AND 6, THE PREVIOUS OPTIMUM POINT IS USED AS
C**THE STARTING POINT. VARIABLES XXX,S AND YYYF CONTAIN THIS INFORMATION
C**AND ARE SET IN SUBROUTINE DYNOPT.
C**
C
C
C**
C**CALL SUBROUTINE OMYSCH TO MOVE ALONG THE VARIABLE IDIRC.
C**
 88   CALL OMYSCH(IDIRC,IFLAG)
C
C
C**
C**NOW CALL SUBROUTINE OMYSCH TO MOVE ALONG THE OTHER VARIABLE, IF
C** IFLAG IS NOT 3  AND NFTERM IS NOT REACHED.
C**
      IDIRC=IDIRC+1
      IF(IDIRC.GT.IVARL)IDIRC=IVARL-1
      IF(IFLAG.LT.3 .AND. NFUEVA.LT.NFTERM) GO TO 88
C**
C**IF IFLAG=3, THEN SEARCH HAS FAILED ALONG BOTH DIRECTIONS .
C**TRY SOME POINTS FURTHER AHEAD TO SEE IF ANY IMPROVEMENTS IS POSSIBLE
C**IF MAX. OF OBJ. FUN. EVALUATIONS IS REACHED THEN QUIT.
C**
```

```
      IF(NFUEVA.GE.NFTERM) GO TO 313
C**
      WRITE(6,249)
 249  FORMAT(1X,' IFLAG=3 ; SEARCH FAILED AT BOTH DIRECTIONS.<<')
C**
C**KEEP THE CURRENT POINT IN TEMP ( NOTE THAT THIS MIGHT NOT BE THE
C**BEST POINT FOUND SO FAR.)
C**
      DO 260 I=1,6
         TEMP(I)=XXX(I)
 260  CONTINUE
      TEMP(10)=YYYF
C**
C**MOVE TO A NEW POINT. THIS STRATEGY TO SOME DEGREE GUARDS AGAINST
C**THE OCCURANCE OF SMALL BUMPS IN THE OBJECTIVE FUNCTION,S LANDSCAPE.
C**
      ISTRT=IVARL-1
      DO 263 I=ISTRT,IVARL
         XXX(I)=XXX(I)+ISIDEL(I)*DELMN(I)
 263  CONTINUE
C*.
C*.EVALUATE OBJ. FUNCTION FOR THIS NEW POINT
C*.
      CALL RUNC
C*.
C*.IF THE NEW POINT IS BETTER (SUCCESS), THEN SET IFLAG=1, IDIRC=IVARL-1
C*.AND RETURN TO SEARCH.
C*.
      IF(YYYF.GT.TEMP(10)) GO TO 270
      IFLAG=1
      IDIRC=IVARL-1
      GO TO 88
C*..
C*..IF THE NEW POINT IS WORSE , STILL TRY ANOTHER POINT
C*..
 270  XXX(IVARL)=XXX(IVARL)+ISIDEL(IVARL)*DELMN(IVARL)
      CALL RUNC
C*..
C*..IF THIS SECOND POINT IS BETTER (SUCCESS) , THEN SET IFLAG=1 ,
C*..IDIRC=IVARL-1, AND RETURN TO SEARCH.
C*..
      IF(YYYF.GT.TEMP(10)) GO TO 300
      IFLAG=1
      IDIRC=IVARL-1
      GO TO 88
C*..IF THE SECOND POINT IS WORSE, THEN QUIT THE SEARCH
C*..
C**
C**COPY BACK THE MIN. POINT INFORMATION FROM TEMP INTO XXX
C**NOTE THAT THE REAL MIN. IS IN ARRAY TMPMIN IN ROUTINE RUNC.
C**                                   -------
 300  DO 305 I=1,6
         XXX(I)=TEMP(I)
 305  CONTINUE
      YYYF=TEMP(10)
C**
C**PRINT TNE INFORMATION AND STOP
C**
313   IF(NFUEVA.EQ.NFTERM)WRITE(LUW,314)NFTERM
 314  FORMAT(1X, ' MAX. NUMBER OF FUNCTION EVALUATIONS REACHED ;',I5 )
C
      RETURN
      END
C
C
C
```

```
C
C
C***********************************************************************
C***********************************************************************
C**********************************************************************
      SUBROUTINE COGGIN(IDIRC)
C***********************************************************************
C**                                                                   *
C** THIS ROUTINE USES COGGINS' TECHNIQUE TO DO A PRECISE LINE SEARCH  *
C** ALONG THE VARIABLE SPECIFIED BY IDIRC.                            *
C**                                                                   *
C** THIS ROUTIE CALLS THE FOLLOWING SUBROUTINES:                      *
C**    (1) SUBROUTINE RUNC TO CALCULATE THE LOSS-COST FUNCTION OF     *
C**         THE DYNAMIC MODEL.                                        *
C**                                                                   *
C**                                                                   *
C**********************************************************************
C**
C*
      IMPLICIT REAL*8(A-H,O-Z)
      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)
      DIMENSION ZZ(3),WW(3),YF(3)
C
C
C  SET VALUES OF LIMIT, STEP SIZE, AND INITIAL GUESS FOR X
C_________
C
       XLIM=XQLIM(IDIRC)
       DELX=DEL(IDIRC)
       X1=XXX(IDIRC)
C
C
C   DAVIS , SWANN, AND CAMPEY ALGORITHM
C
C   EVALUATE THE FUNCTION FOR  THE INITIAL VALUE OF THE INDEPENDENT
C   VARIABLE.
C
      XXX(IDIRC)=X1
      CALL RUNC
      Y1=YYYF
CC
      WRITE(LUW,210)
 210  FORMAT(1X,//,15X,'D.S.C ALGORITHM')
      WRITE(LUW,204)
 204  FORMAT(1X,//,1X,21X,'XX',14X,'YY',8X,'STEP SIZE')
      WRITE(LUW,205)X1,Y1,DELX
 205  FORMAT(1X,//,1X,15X,E14.7,2X,E14.7,2X,E14.7)
C
C  INCREMENT THE INDEPENDENT VARIABLE AND EVALUATE THE FUNCTION
C
      X2=X1+DELX
      II=0
C
 8    XXX(IDIRC)=X2
      CALL RUNC
      Y2=YYYF
      WRITE(LUW,205) X2,Y2,DELX
C
C+ SEE WHICH OF THE FUNCTION EVALUATIONS IS THE SMALLEST
C
 9    IF(Y1-Y2)10,12,12
C+ STMT 10 IS WHEN FAILED
```

```
 10   IF(II-1)14,14,16
 14   II=II+1
C
C+ IF THE FUNCTION EVALUATION IS MORE AFTER THE INDEPENDENT
C  VARIABLE HAS BEEN INCREMENTED, SWITCH DIRECTION
C
      DELX=-DELX
      X2=X1+DELX
      GO TO 8
C
C  IF THE MAX. IS BRACKETED, MOVE TO THE POWEL ALGOITHM.
C
 16   GO TO 80
 12   CONTINUE
       X3=X2+DELX
C
       XXX(IDIRC)=X3
       CALL RUNC
       Y3=YYYF
       WRITE(LUW,205)X3,Y3,DELX
C
       GO TO 171
C+>+>+>+>+>+
 17    X3=X4
       Y3=Y4
C+>+>+>+>+>+
C
C+ IF THE FUNCTION EVALUATION IS SMALLER THAN THE PREVIOUS VALUE
C  THEN DOUBLE THE STEP SIZE.
C
 171  DELX=2.*DELX
C
      X4=X3+DELX
      XXX(IDIRC)=X4
      CALL RUNC
      Y4=YYYF
C+
      IF(Y3-Y4) 20,22,22
C+ FAILURE
 20   GO TO 90
C+ SUCCESS
 22   GO TO 17
C
C  WHEN THE OPTIMUM IS STRADDLED, EVALUATE THREE POINTS ABOUT THE MAX
C
C
C  POWELL ALGORITHM
C
 80   ZZ(1)=X1
C+>+>+>+>+>+
      YF(1)=Y1
C+>+>+>+>+>+
      ZZ(2)=X1+DELX/2.
C+>NEED TO EVALUATE YF(2)
      ZZ(3)=X2
C+>+>+>+>+>+
      YF(3)=Y2
C+>+>+>+>+>+
      GO TO 99
C
 90   ZZ(1)=X3
C+>+>+>+>+>+
      YF(1)=Y3
C+>+>+>+>+>+
      ZZ(2)=X3+DELX/2.
C+>NEED TO EVALUATE YF(2)
```

```
      ZZ(3)=X4
C+>+>+>+>+>+
      YF(3)=Y4
C+>+>+>+>+>+
      GO TO 99
C
C  EVALUATE THE FUNCTION AT THESE THREE POINTS
C+>INFACT ONLY AT THE MIDDLE POINT SINCE THE FUNC. VALUES AT THE OTHER
C+> POINTS ARE KNOWN.
C
C+>+>+>+>+>+
99    XXX(IDIRC)=ZZ(2)
      CALL RUNC
      YF(2)=YYYF
C
C FIT A QUADRATIC TO THESE THREE POINTS
C
C
C
C   NFITS COUNTS THE NUMBER OF FITS EMPLOYED
C
      NFITS=1
C   NQTFT2 IS THE MAX. NUMBER OF FITS ALLOWED
      NQTFT2=10
      WRITE(LUW,211)
 211  FORMAT(1X,//,20X,'XMAX',12X,'YMAX')
 98   CONTINUE
      A=ZZ(2)-ZZ(3)
      B=ZZ(3)-ZZ(1)
      C=ZZ(1)-ZZ(2)
      D=ZZ(2)**2-ZZ(3)**2
      E=ZZ(3)**2-ZZ(1)**2
      F=ZZ(1)**2-ZZ(2)**2
C
C+ ANALYTICALLY EVALUATE THE MIN. OF THE FITTED CURVE
C
      ZZT=.5*(D*YF(1)+E*YF(2)+F*YF(3))
      ZZB=A*YF(1)+B*YF(2)+C*YF(3)
      ZZM=ZZT/ZZB
C
C  EVALUATE THE FUCTION VALUE AT THIS POINT
C
      XXX(IDIRC)=ZZM
      CALL RUNC
      YFM=YYYF
      WRITE(LUW,205)ZZM,YFM
C
C   CHECK TO SEE IF ANY OF THE POINTS ARE WITHIN THE DESIRED ACCURACY
C
C
      DO 100 J=1,3
         WW(J)=DABS(ZZ(J)-ZZM)
         IF(WW(J)-XLIM) 105,105,106
C+>+>+>+>+>
C+>+COULD SET XMIN TO ZZM AND YMIX TO YFM : THE MIN OF QUADRATIC
 105     XMIN=ZZM
         YMIN=YFM
C
         XXX(IDIRC)=XMIN
         YYYF=YMIN
C
C+>+>+>+>+>
         GO TO 200
 106     CONTINUE
C  IF MAX. NUMBER OF OBJ. FUN. EVALUATIONS OR MAX. NUMBER OF FITS
C  ALLOWED IS REACHED, THEN STOP
```

```
C
            IF(NFUEVA.EQ.NFTERM .OR. NFITS.EQ.NQTFT2) GO TO 200
 100  CONTINUE
C
C+ SEE WHICH FUNCTION VALUE IS THE LARGEST AND REPLACE IT WITH
C+ THE INTERPOLATED MAX. POINT
C+
      JK=1
      IF(YF(JK).LE.YF(2))JK=2
      IF(YF(JK).LE.YF(3))JK=3
 117  ZZ(JK)=ZZM
      YF(JK)=YFM
C
C
C  FIT A QUADRATIC TO THE NEW POINTS
C
      NFITS=NFITS+1
      GO TO 98
C
C+>THE MIN FUNC. VALUE IS ALREADY EVALUATED
C
C
C  XXX(IDIRC) AND YYYF ARE ALREADY SET EQUAL TO XMIN AND YMIN, WHICH
C  ARE THE MIN. OF XXX AND ITS OBJ. FUNCTION.
C
C
 200  IF(NFUEVA.EQ.NFTERM)WRITE(LUW,401)NFUEVA
 401  FORMAT(1X,' MAX. NUMBER OF FUNCTION EVALUATONS REACHED.')
      IF(NFITS.EQ.NQTFT2)WRITE(LUW,402)
 402  FORMAT(1X,' MAX. NUMBER OF QUAD. FITS ALLOWED REACHED.')
C
C  IF MAX. NUMBER OF FUNCTION EVALUATIONS OR QUAD. FITS ALLOWED
C  IS REACHED, THEN COPY THE BEST OPTIMUM OBTAINED SO FAR INTO
C  XXX(1) AND YYYF.
C
      IF(NFUEVA.EQ.NFTERM.OR.NFITS.EQ.NQTFT2)XXX(IDIRC)=TMPMIN(IDIRC)
      IF(NFUEVA.EQ.NFTERM.OR.NFITS.EQ.NQTFT2)YYYF=TMPMIN(10)
      RETURN
      END
C
C
C
C
C**********************************************************************
C**********************************************************************
C**********************************************************************
      SUBROUTINE OMYSCH(IDIRC,IFLAG)
C**********************************************************************
C** THIS SUBROUTINE IS USED IN CONJUNCTION WITH SUBROUTINE TWOSCH    *
C** FOR A TWO-DIMENSIONAL OPTIMIZATION.                              *
C**                                                                  *
C** THIS SUBROUTINE IS CALLED FOR SEARCH ALONG ONE AXIS BUT          *
C** USES THE INFORMATION OBTAINED FROM SEARCHING ALONG THE OTHER     *
C** DIRECTION.                                                       *
C**                                                                  *
C** THIS ROUTINE CALLS THE FOLLOWING SUBROUTINES:                    *
C**     (1) SUBROUTINE RUNC TO CALCULATE THE LOSS-COST FUNCTION OF   *
C**         THE DYNAMIC MODEL.                                       *
C**                                                                  *
C**********************************************************************
C**
C
C
C
C**NOTE THE FOLLOWING VARIABLES DEFINITION USED IN THIS ROUTINE.    **
C
```

```
C   IDIRC : SHOWS THE DIRECTION OF SEARCH ; VARIABLE NUMBER. E.G.:
C           IDIREC=1 -- SEARCH ALONG VARIABLE NUMBER ONE ; H
C           IDIREC=2 -- SEARCH ALONG VARIABLE NUMBER TWO ; ISTEPS
C
C   IFLAG : CONTAINS THE INFORMATION ABOUT THE SEARCH ALONG THE
C           DIRECTIONS;
C           IFLAG=1 -- SEARCH ALONG THE OTHER DIRECTION WAS SUCCESSFULL
C           IFLAG=2 -- SEARCH ALONG THE OTHER DIRECTION FAILED.
C           IFLAG=3 -- SEARCH ALONG BOTH DIRECTIONS FAILED.
C
C   II    : SHOWS THE NUMBER OF FAILURES ALONG THE CURRENT DIRECTION.
C
C   NQFIT : THE NUMBER OF QUADRATIC FITS TO BE EMPLOYED . IT IS SET
C           EQUQL TO IFLAG FOR THIS SUBROUTINE.
C
C   ISIDEL: THE SIGN OF THE LAST SUCCESSFUL STEP FOR EACH DIRECTION.
C
C**
C*
      IMPLICIT REAL*8(A-H,O-Z)
      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
      DIMENSION ZZ(3),WW(3),YF(3)
      DIMENSION IQACU(6)
C**
C**THE STARTING POINT AND ITS CORRESPONDING OBJ. FUN. VALUE
C**
      X1=XXX(IDIRC)
      Y1=YYYF
C**
C**INCREMENT THE VARIABLE AND EVALUATE THE FUNCTION
C**
 7    X2=X1+DEL(IDIRC)
      II=0
C
 8    XXX(IDIRC)=X2
      CALL RUNC
      Y2=YYYF
C**
C**IF THE NEW POINT IS BETTER (SMALLER OR THE SAME) ; SUCCESSS
C**
      IF(Y2.GT.Y1) GO TO 10
C..THEN DOUBLE THE STEP SIZE FOR THE NEXT SEARCH ALONG THIS DIRECTION.
C..AND RETURN TO THE MAIN PROGRAM TO MOVE IN THE OTHER DIRECTION.
      IF(DABS(DEL(IDIRC)).LT.DELMX(IDIRC))DEL(IDIRC)=2.D0*DEL(IDIRC)
      XXX(IDIRC)=X2
      YYYF=Y2
C...SET FLAG TO SHOW THE SUCCESS.
C
C...AND KEEP THE SIGN OF THIS SUCCESSFUL MOVE
C
      IFLAG=1
      ISIDEL(IDIRC)=DSIGN(1.D0,DEL(IDIRC))
C>>>>
      RETURN
C.........................................................
C**
C**THE NEW POINT IS WORSE (BIGGER) ; FAILURE
C*.IF IT IS THE SECOND CONSEQUTIVE FAILURE IN THIS VARIABLE DIRECTION
C*.GIVEN THAT |DEL| IS EQUAL TO DELMN , THEN
C*.THE MIN IS BRACKETED; USE POWEL,S METHOD TO FIT QUADRATICS.
C
 10   IF(II.GE.1) GO TO 80
C
```

```
C*.IF IT IS THE FIRST FAILURE AFTER A SUCCESS ,AND
C...IF |DEL| IS GREATER THAN DELMN, THEN SET DEL=DELMN WHILE
C...RESERVING THE SIGN OF DEL AND MOVE TO A NEW POINT.
C
C...( DSIGN(A,B) IS A FORTRAN FUNCTION WHICH TRANSFERS THE SIGN OF
C...  B TO A.)
C
      IF(DABS(DEL(IDIRC)).GT.DELMN(IDIRC))DEL(IDIRC)=DSIGN(
     *                                      DELMN(IDIRC),DEL(IDIRC))
      IF(DABS(DEL(IDIRC)).GT.DELMN(IDIRC))GO TO 7
C...OTHERWISE, REVERSE THE SEARCH DIRECTION
C
      DEL(IDIRC)=-DSIGN(DELMN(IDIRC),DEL(IDIRC))
      II=II+1
C...SAVE THE CURRENT POINTS FOR POWELL,S METHOD , IN THE CASE THAT
C...BECOMES NECCESSARY TO USE THEM.
      ZZ(1)=X2
      YF(1)=Y2
      ZZ(2)=X1
      YF(2)=Y1
C...AND MOVE TO A NEW POINT.
C
      X2=X1+DEL(IDIRC)
      GO TO 8
C.................................................................
C**
C**MODIFIED POWELL,S METHOD ; FIT QUADRATIC TO ESTIMATE MIN.
C**-------------------------
C**
80    WRITE(LUW,216)
 216  FORMAT(1X,//,1X,'  POWELL,S METHOD')
C*..INCREMENT IFLAG , SINCE WE HAD TWO CONSEQUTIVE FAILURES IN THIS
C*..DIRECTION. SET NUMBER OF QUAD. FITS EQUAL TO IFLAG.
C
      IFLAG=IFLAG+1
      NQFIT=IFLAG
      WRITE(LUW,217)NQFIT
 217  FORMAT(1X,'NUMBER OF QUADRATIC FITS EMPLOYED=',I3)
C
C*..THREE POINTS ARE REQUIRED FOR A QUADRATIC FIT . TWO POINTS
C*..HAVE ALREADY BEEN STORED IN ZZ(1) AND ZZ(2)
C
      ZZ(3)=X2
      YF(3)=Y2
C*+
C*+IF REQUIRED PRECISION , XQLIM, IS EQUAL TO THE MIN. LIMIT ON
C*+STEP SIZE, THEN SKIP THE POWELL,S METHOD. THIS STRATEGY
C*+IS ESPECIALLY USEFUL IN GUARDING AGAINST INTERPOLATIONS OF
C*+INTEGER-VALUED VARIABLES.
C*+
      IF(XQLIM(IDIRC).GT.DELMN(IDIRC)-1.D-14 )GO TO 350
C*+
C
C*..ANALYTICALL EVALUATE THE MIN. OF THE FITTED CURVE ; ZZM
C
 90   AAAA=ZZ(2)-ZZ(3)
      BBBB=ZZ(3)-ZZ(1)
      CCCC=ZZ(1)-ZZ(2)
      DDDD=ZZ(2)**2-ZZ(3)**2
      EEEE=ZZ(3)**2-ZZ(1)**2
      FFFF=ZZ(1)**2-ZZ(2)**2
      ZZT=.5*(DDDD*YF(1)+EEEE*YF(2)+FFFF*YF(3))
      ZZB=AAAA*YF(1)+BBBB*YF(2)+CCCC*YF(3)
      ZZM=ZZT/ZZB
C
C*..EVALUATE THE FUNCTION VALUE AT THIS POINT
```

```
C
      XXX(IDIRC)=ZZM
      CALL RUNC
      YFM=YYYF
      WRITE(LUW,218)
 218  FORMAT(1X,' THE ABOVE IS THE MIN OF QUADRATIC.<<< ')
C
C*..IF ANY OF THE POINTS ARE WITHIN THE DESIRED (QUITTING) ACCURRACY
C*..THEN QUIT FITTING QUADRATIC. SET IQACU=1 TO SHOW THIS SITUATION
C
      IQACU(IDIRC)=0
      DO 300 J=1,3
          WW(J)=DABS(ZZ(J)-ZZM)
          IF(WW(J).GT.XQLIM(IDIRC))GO TO 300
               XMIN=ZZ(J)
               YMIN=YF(J)
C
               XXX(IDIRC)=XMIN
               YYYF=YMIN
C
               IQACU(IDIRC)=1
               GO TO 400
 300  CONTINUE
C
C*..SEE WHICH FUNCTION VALUE IS THE LARGGEST (WORST) AND REPLACE IT
C*..WITH THE QUADRATIC,S MIN. NOTE YF(JK) CONTAINS THE LARRGEST VALUE.
C
      JK=1
      IF(YF(JK).LE.YF(2))JK=2
      IF(YF(JK).LE.YF(3))JK=3
      ZZ(JK)=ZZM
      YF(JK)=YFM
C
C*..SEE IF MORE FITS ARE REQUIRED, FIT ONE TO THE NEW POINTS.
C
      NQFIT=NQFIT-1
      IF(NQFIT.GT.0) GO TO 90
C
C*..ENOUGH QUADRATIC FITTED , SET THE MINIMUM EQUAL TO THE SMALLEST
C*..OF THE LAST THREE POINTS IN ORDER TO RETURN FROM THIS ROUTINE.
C*
  350 JK=1
      IF(YF(JK).GE.YF(2))JK=2
      IF(YF(JK).GE.YF(3))JK=3
C
C*..THE MIN. IS IN ZZ(JK)
C
      XXX(IDIRC)=ZZ(JK)
      YYYF=YF(JK)
C
C..............................................................
C
 400  RETURN
      END
C
C
C
C
C***********************************************************************
C***********************************************************************
C***********************************************************************
        SUBROUTINE RUNC
C***********************************************************************
C                                                                     *
C** THIS ROUTINE CALCULATES THE LOSS-COST OF THE DUNAMIC MODEL.       *
C**                                                                   *
```

```
C** THIS ROUTINE CALLS THE FOLLOWING SUBROUTINE:                        *
C**    (1) SUBROUTINE APP TO CALCULATE THE AVERAGE TIME OUT OF CONTROL*
C**        BEFORE A POINT FALLS AND IS CHARTED OUTSIDE THE CONTROL      *
C**        LIMITS.                                                      *
C**    (2) SUBROUTINE CMAINT TO CALCUALTE AVERAGE HOURLY COST OF        *
C**        MAINTAINING THE CONTROL CHART.                               *
C**    (3) SUBROUTINE FALSA TO CALCULATE THE EXPECTED NUMBER OF FALSE *
C**        ALARMS.                                                      *
C**                                                                     *
C** THE RESTRICTION ON M, IM, D, AND ID IS INCORPORATED AS:             *
C**    (1) PENALTY FUNCTION                                             *
C**    (2) BARRIER FUNCTION                                             *
C**  A FLAG CALLED IPENAL IS USED TO SELECT EITHER OF (1) OR (2).       *
C**                                                                     *
C***********************************************************************
C**
C*
      IMPLICIT REAL*8(A-H,O-Z)
         REAL*8  LAMBDA,L,IL,M,IM,ID
      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 / DYNM4 / PROBPT
      COMMON / DYOPT1 / XXX(6),YYYF
      COMMON / DYOPT2 / ISIDEL(6), NFUEVA,NFTERM
      COMMON / DYOPT3 / TMPMIN(10)
         COMMON/A1111/L,IL,M,IM,D,ID
         COMMON/B1111/ISTEPS
         COMMON/BE1/ITR
C
C...............................................
C
C**
C**
C*+SET IPENAL TO 1 IF PENALTY METHOD IS DESIRED
C*+SET IPENAL TO 2 IF BARRIER METHOD IS DESIRED
C*++++++++++++++++++++++++++++
      IPENAL=1
C**
C**INCREMENT THE NUMBER OF OBJ. FUNCTION EVALUATIONS.
C**
      NFUEVA=NFUEVA+1
      WRITE(LUW,1)NFUEVA
 1    FORMAT(1X,' BELOW IS THE ',I4,'TH FUNCTION EVALUATED.')
C**
C**
C**IF THE LIMIT OF OBJ. FUN. EVALUATIONS IS REACHED THEN SET YYYF TO A
C** VERY BIG VALUE AND RETURN.
C**
      IF(NFUEVA.GT.NFTERM)YYYF=999888777.
      IF(NFUEVA.GT.NFTERM)RETURN
C**
C**IF THIS IS THE FIRST TIME THIS SUBROUTINE IS CALLED THEN INITIALIZE
C**RQQR.
C**
      IF(NFUEVA.GT.1) GO TO 95
C+.
C+.INITIALIZE RQQR , PARAMETER USED IN PENALTY AND BARRIER METHODS.
C+.
       RQQR=1.D0
C*.
C ..................................
C*.
C*.FOR THIS AND THE SUBROUTINES CALLED FROM THIS ROUTINE, THE
C*.DECISION VARIABLES ARE RENAMED AS FOLLOWS.
```

```
C*. ISTEPS IS THE NUMBER OF STEPS, SAMPLES, TAKEN TO GET TO
C*. A DESIRED QUANTILE VALUE, PROBPT.
C*.
C*. L IS INTERVAL SIZE FACTOR,      IL IS INITIAL LENGTH OF INTERVAL
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
C*.
C*.THEREFORE, THE VARIABLES IN EACH OF THE FOLLOWING COLUMNS ARE EQUAL.
C*.
C*.    XXX(1)   XXX(2)   XXX(3)   XXX(4)  XXX(5)    XXX(6)
C*.     HF      ISTEPS   IK        KF      IN         NF
C*.     L       ISTEPS   IM        M       ID          D
C*.
C ...................................
C*
C**
C**
C**SET ALL THE VARIABLES EQUAL TO THEIR CORRESPONDING DECISION VARIABLES
C**
   95     L=XXX(1)
          ISTEPS=XXX(2)
          IM=XXX(3)
          M=XXX(4)
          ID=XXX(5)
          D=XXX(6)
C**
C**FIRST CALCULATE IL (IH);THE INITIAL SAMPLING INTERVAL TO ACHIEVE
C**-------------------
C**A DESIRED QUANTILE (PROBPT) IN ISTEPS SAMPLS.
C*.
C*.FIND THE VALUE THAT THE SUM OF ISTEPS INTERVALS SHOULD BE EQUAL TO.
C*.
           SUMATN=DEXP(DLOG(-DLOG(1.D0-PROBPT))/ETA-DLOG(THETA))
C*.
C*.CALCULATION OF IL (OR IH) IF L IS 1.
C*.
           IF(L.EQ.1)IL=SUMATN/ISTEPS
           IF(L.EQ.1)GO TO 97
C*.
C*.CALCULATION OF IL (OR IH) IF L IS NOT EQUAL TO 1.
C*.
           BX=ISTEPS*DLOG(L)
           IL=SUMATN*(1.D0-L)/(1.D0-DEXP(BX))
C**
C**ITR IS USED TO CALCULATE NUMBER OF ITERATIONS FOR DO LOOP
C**
  97       ITR=ISTEPS+13
C
C__________THE RESTRICTION ON M , IM , D , AND ID _____________________
C
C**
C** CALCULATE RIDE ; ENDING SAMPLE SIZE
C**
        RIDE=ID*(D**(ISTEPS-1))
C**
C**CALCULATE RIME ; ENDING CONTROL LIMIT WIDTH
C**
        RIME=IM*(M**(ISTEPS-1))
C**
C**CALCULATE UPPER LIMIT ON IM , AND RIME
C**
        UPIM=DELTA*DSQRT(ID)
        UPRIME=DELTA*DSQRT(RIDE)
C**
        IF(IM.LT.UPIM.AND.RIME.LT.UPRIME)GO TO 105
C**
```

```
C*+++++++++++++++++
C*+PENALTY METHOD +
C*+++++++++++++++++
105   IF(IPENAL.EQ.2) GO TO 106
      YYQQ=DMIN1((UPIM-IM),0.0D0)**2+DMIN1((UPRIME-RIME),0.0D0)**2
      RQQR=RQQR*.65D0
      IF(RQQR.LT.1.E-45)RQQR=1.E-45
      YYYQ=YYQQ/RQQR
      GO TO 107
C*+++++++++++++++++
C*+BARRIER METHOD +
C*+++++++++++++++++
106    YYQQ=(1.D0/(UPIM-IM)+1.D0/(UPRIME-RIME))
       RQQR=RQQR*.6D0
       YYYQ=YYQQ*RQQR
C
C______________________________________________________________________
C
C**
C**CALL APP TO CALCULATE A DOUBLE PRIME; ADBP.
C**
 107    CALL APP(ADBP)
C**
C**CYCLE IS THE AVERAGE CYCLE LENGTH.
C**
C*.RMEAN IS MEAN OF WEIBALL DISTRIBUTION
         XGA=1.D0+1.D0/ETA
C*.DGAMMA IS AN IMSL ROUTINE TO CALCULATE GAMMA FUNCTION
         RMEAN=1.D0/THETA*DGAMMA(XGA)
C*.
        CYCLE=RMEAN+ADBP+DD
C**
C**GAMMA IS THE PROPORTION OF TIME A PROCESS WILL BE OUT OF CONTROL.
C**
        GAMMA=(ADBP+DD)/CYCLE
C**
C**CALL FALSA TO CALCULATE ENFALS; EXPECTED NUMBER OF FALSE ALARMS
C**
         CALL FALSA(ENFALS)
C**
C**CALL CMAINT TO CALCULATE BAHCM
C**
         CALL CMAINT(BAHCM)
C**
C**
C**YYYF IS THE LOSS COST PER UNIT TIME; OBJECTIVE FUNCTION
C**
        YYYP=GAMMA*VZMV1+T*ENFALS/CYCLE+W/CYCLE+BAHCM
C
C*+ADD THE PENALTY TO OBJ. FUN.
C*+
        YYYF=YYYP+YYYQ
C
C**
C**KEEP THE MIN. SO FAR IN AN ARRAY CALLED TMPMIN
C**                                        -------
       IF(YYYF.GE.TMPMIN(10))RETURN
C*.
C*.IF NEWLY CALCULATED YYYF IS BETTER , THEN UPDDATE TMPMIN
C*.
       DO  19  I=1,6
         TMPMIN(I)=XXX(I)
 19    CONTINUE
       TMPMIN(10)=YYYF
C
       RETURN
```

```
      END
C
C
C
C
C*********************************************************************
C*********************************************************************
C*********************************************************************
      SUBROUTINE DYMEVA
C*********************************************************************
C**                                                                 *
C** THIS SUBROUTINE CALCULATES THE LOSS-COST FOR THE DYNAMIC MODEL   *
C** FOR THE GIVEN DESIGN TO BE EVALUATED.                            *
C**                                                                 *
C** THIS ROUTINE CALLS THE FOLLOWING SUBROUTINE:                     *
C**     (1) SUBROUTINE APP TO CALCULATE THE AVERAGE TIME OUT OF CONTROL*
C**         BEFORE A POINT FALLS AND IS CHARTED OUTSIDE THE CONTROL  *
C**         LIMITS.                                                 *
C**     (2) SUBROUTINE CMAINT TO CALCUALTE AVERAGE HOURLY COST OF    *
C**         MAINTAINING THE CONTROL CHART.                          *
C**     (3) SUBROUTINE FALSA TO CALCULATE THE EXPECTED NUMBER OF FALSE *
C**         ALARMS.                                                 *
C**                                                                 *
C**                                                                 *
C******************************************************************* *
C**
C*
      IMPLICIT REAL*8(A-H,O-Z)
         REAL*8  LAMBDA,L,IL,M,IM,ID
      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
         COMMON/BE1/ITR
C
C...............................................
C .................................
C*.
C*.FOR THIS AND THE SUBROUTINES CALLED FROM THIS ROUTINE, THE
C*.DECISION VARIABLES ARE RENAMED AS FOLLOWS.
C*. ISTEPS IS THE NUMBER OF STEPS, SAMPLES, TAKEN TO GET TO
C*. A DESIRED QUANTILE VALUE, PROBPT.
C*.
C*. L IS INTERVAL SIZE FACTOR,      IL IS INITIAL LENGTH OF INTERVAL
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
C*.
C*.THEREFORE, THE VARIABLES IN EACH OF THE FOLLOWING COLUMNS ARE EQUAL.
C*.
C*.   XXX(1)   XXX(2)   XXX(3)   XXX(4)  XXX(5)    XXX(6)
C*.    HF      ISTEPS    IK        KF      IN         NF
C*.     L      ISTEPS    IM        M       ID          D
C*.
C ..................................
C...............................................
C*
C**
C**SET ALL THE VARIABLES EQUAL TO THEIR CORRESPONDING DECISION VARIABLES
```

```
C**
        L=HF
        IL=IH
        ISTEPS=ISTPP
        IM=IK
        M=KF
        ID=IN
        D=NF
C**
C**ITR IS USED TO CALCULATE NUMBER OF ITERATIONS FOR DO LOOP
C**
  97     ITR=ISTEPS+13
C**
C**CALL APP TO CALCULATE A DOUBLE PRIME; ADBP.
C**
 107    CALL APP(ADBP)
C**
C**CYCLE IS THE AVERAGE CYCLE LENGTH.
C**
C*.RMEAN IS MEAN OF WEIBALL DISTRIBUTION
         XGA=1.D0+1.D0/ETA
C*.DGAMMA IS AN IMSL ROUTINE TO CALCULATE GAMMA FUNCTION
         RMEAN=1.D0/THETA*DGAMMA(XGA)
C*.
        CYCLE=RMEAN+ADBP+DD
C**
C**GAMMA IS THE PROPORTION OF TIME A PROCESS WILL BE OUT OF CONTROL.
C**
        GAMMA=(ADBP+DD)/CYCLE
C**
C**CALL FALSA TO CALCULATE ENFALS; EXPECTED NUMBER OF FALSE ALARMS
C**
         CALL FALSA(ENFALS)
C**
C**CALL CMAINT TO CALCULATE BAHCM
C**
         CALL CMAINT(BAHCM)
C**
C**
C**YYYF IS THE LOSS COST PER UNIT TIME; OBJECTIVE FUNCTION
C**
        YYYP=GAMMA*VZMV1+T*ENFALS/CYCLE+W/CYCLE+BAHCM
C
        DYMLCS=100.D0*YYYP
       RETURN
       END
C
C
C
C
C
C
C***********************************************************************
C***********************************************************************
C***********************************************************************
       SUBROUTINE APP(ADBP)
C***********************************************************************
C**                                                                   *
C** THIS ROUTINE CALCULATES THE AVERAGE TIME OUT OF CONTROL BEFORE THE*
C** DETECTING SAMPLE IS PLOTED ON THE CHART.                          *
C**                                                                   *
C***********************************************************************
C**
C*
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*8  NUM,INTERM,LAMBDA,M,L,ITINTR,IM,ID,IL
```

```
      REAL*8  IJTINT,ICUINT
      REAL*8  ICNTLT
      COMMON / DCDY1 / DELTA, B,C,DD,E,VZMV1,T,W
      COMMON / DYNM1 / THETA,ETA
C
C**THETA CORSPONDS TO LAMBDA ( ALPHA)     ETA CORRESPONDS TO BETA
C
         COMMON/A1111/L,IL,M,IM,D,ID
         COMMON/B1111/ISTEPS
         COMMON/BE1/ITR
C
C**
C**   L IS INTERVAL SIZE FACTOR
C**  M IS SPEC. LIMIT FACTOR
C**   D IS  SAMPLE SIZE FACTOR
C**
C
      ADBP=0.
      OCUINT=0.
      ITINTR=IL/L
C**
C**OCNTLT IS OUTER LOOP CONTROL LIMIT.
C**
      OCNTLT=IM/M
C
      ILT=ITR-13
C
      DO 200  I=1,ILT
C
C**ITINTR IS THE LENGTH OF THE (I-1)TH INTERVAL; H SUB (I-1)
C
C** IJTINT  IS THE LENGTH OF THE (I+J-1)TH INTERVAL; H SUB (I+J-1)
C
      ITINTR=ITINTR*L
      IJTINT=ITINTR/L
C++
      OCNTLT=OCNTLT*M
C+=+=+
C+=+CONTROL LIMIT SPREAD RESTRICTED BETWEEN .5 AND 5.5.
C+=+=+
      IF(OCNTLT.LT.0.5)OCNTLT=0.5
      IF(OCNTLT.GT.5.5)OCNTLT=5.5
C+=+=+
      ICNTLT=OCNTLT/M
C
C**SAMPLI IS THE SAMPLE SIZE AT THE (I-1)TH INTERVAL
C
      SAMPLI=ID*D**(I-2)
C
C
      ICUINT=0.
      PRODPL=0.0
      INTERM=0.
C
      J=0
300   J=J+1
      IJTINT=IJTINT*L
C++
      ICNTLT=ICNTLT*M
C+=+=+
C+=+CONTROL LIMIT SPREAD RESTRICTED BETWEEN .5 AND 5.5
C+=+=+
       IF(ICNTLT.LT.0.5)ICNTLT=0.5
       IF(ICNTLT.GT.5.5)ICNTLT=5.5
C+=+=+
C
```

```
C**ICUINT IS THE CUMULATIVE INTERVAL STARTING FROM THE ITH INTERVAL TO
C**,AND EXCLUDING, THE (I+J)TH INTERVAL.
C
C**PNDTCB IS PROB. OF NOT DETECTING BEFORE, AND EXCLUDING, THE JTH SAMPL
C** TAKEN  AFTER THE PROCESS WENT OUT OF CONTROL IN THE ITH INTRVAL.
C** THAT IS  PNDTCB=(1-PI)(1-PI+1)...(1-PI+J-1)
C
        ICUINT=ICUINT+IJTINT
C
C**SAMPLI IS THE SAMPLE SIZE AT THE (I+J-1)TH INTERVAL
C++ALSO, MAKE SURE THAT THE SAMPLE SIZE IS INTEGER AND >=2.
C++
        SAMPLI=SAMPLI*D
        ISAMPL=SAMPLI+.4999999D0
        IF(ISAMPL.LT.2)ISAMPL=2
C+=+=+=+
C+=+UPPER LIMIT FOR SAMPLE SIZE IS 1000
C+=+=+=+
        IF(ISAMPL.GT.1000)ISAMPL=1000
C+=+=+=+
        SAMPLS=ISAMPL
C
C**Y IS ONE OF THE LIMITS OF THE STD. NORMAL INTEGRATION.
C** THE OTHER LIMIT IS INFINITY.
C**NOTE THAT P IS THE INTEGRAL FROM -INFINITY TO Y, AND PDCTJ IS THE INT
C**INTEGRAL FROM Y TO INFINITY OF THE STD. NORMAL DISTRIBUTION.
C
        Y=ICNTLT-DELTA*DSQRT(SAMPLS)
C
C** MDNOR IS AN IMSL ROUTINE WHICH CALCULATES NORMAL DENSITY INTEGRAL
C
        CALL MDNORD(Y,P)
C
C**PDTC IS THE PROB OF DETECTING THE SHIFT ON THE (I+J-1)TH SAMPLE
C
        PDTC=1.-P
C        IF(I.LT.5)WRITE(6,912)Y,PDTC
 912     FORMAT(1X,'Y &  PDTC ',2E15.8)
C
C
C++USING LN
C++
C++PLN IS LN OF PDTC
C++PRODPL IS LN OF PROB. OF NOT DETECTING ; PNDTCB
C++TLN IS LN OF (ICUINT+SAMPLI*E)
C++
      PLN=DLOG(PDTC)
      TERM=ICUINT+SAMPLS*E
      TLN=DLOG(TERM)
      PAINTL=PLN+PRODPL+TLN
C++
C++
C**INTERM IS THE  'UNADJUSTED ' TERM WHICH IS GOING TO BE MULTIPLIED BY
C** PROB. OF GOING OUT OF CONTROL IN THE ITH INTERVAL.
C
C
C++NOTE THAT WHEN PAINTL IS LESS THAN -20.(OR-15) THEN INTERM=INTERM+0.0
C++OR NOTHING IS ADDED TO INTER. SO, TERMINATE THE LOOP;
C++ THE NUMBER OF ITERATIONS FOR J IS ENOUGH.
C++
       IF(PAINTL.LT.-20.) GO TO 41
       IF(PAINTL.GT.-20.)INTERM=INTERM+DEXP(PAINTL)
C**
C**PNDTCB IS PROB OF NOT DETECTING BEFORE, AND EXCLUDING,THE JTH SAMPLE
C** TAKEN AFTER THE PROCESS WENT OUT OF CONTROL IN THE ITH INTERVAL.
C** THAT IS PNDTCB =(1-PI)(1-PI+1)...(1-PI+J-1)
```

```
C**
C
C++PRODPL IS LN OF PNDTCB
C++
C++UPDATE PRODPL
C++.NOTE THAT WHEN PDTC IS ALMOST 1. , THEN THE CORRESPONDING Q IS ZERO.
C++.THAT IS, ALL THE SUBSEQUENT PNDTCB,S (PROB. OF NOT DETECT. BEFORE)
C++.ARE ZERO. SO, WE SHOULD TERMINATE THE LOOP.
C++
C++WHEN P>0. , NEED TO UPDATE PNDTC.
C++
C++
         IF(PDTC.LT.0.99999999)PRODPL=PRODPL+DLOG(1.-PDTC)
         IF(PDTC.GT.0.99999999)GO TO 41
C
       GO TO 300
C
C_______________________SPECIFIC FOR WEIBULL___________________________
C
C*********************************
C**CALCULATE AVG. TIME OF OCCURANCE OF THE SHIFT GIVEN THE OCCURANCE
C** IS IN THE ITH INTERVAL ;AT.
C**********************************
C
C**GAMPRM IS PARAMETER OF GAMMA DISTRIBUTION
 41    GAMPRM=1.+1./ETA
C**
C**RIA IS THE INTEGRAL OF GAMMA DENSITY FROM 0 TO ALOCU
C**RIB IS THE INTEGRAL OF GAMMA DENSITY FROM 0 TO BLTI
C
C*  OCUINT IS T SUB I-1
C*  TI IS T SUB I
C
        TI=OCUINT+ITINTR
        ALOCU=(THETA*OCUINT)**ETA
        BLTI=(THETA*TI)**ETA
C
C__MYGAMA IS MY GAMMA SUBROUTINE WHICH CALLS IMSL ROUTINE MDGAMA
C__THIS WAS DONE SINCE MDGAM IS NOT A DOUBLE PRECISION ROUTINE.
C
        CALL MYGAMA(ALOCU,GAMPRM,RIA,IER1)
        CALL MYGAMA(BLTI,GAMPRM,RIB,IER2)
C
C**RNAT IS THE INCOMPLETE GAMMA INTEGRAL FROM ALOCU TO BLTI
C
C* DGAMMA IS AN IMSL ROUTINE THAT GIVES THE GAMMA FUNCTION OF A NUMBER
C
       RNAT=1./THETA*(RIB-RIA)*DGAMMA(GAMPRM)
C
C**DAT IS EQUAL TO THE AREA UNDER THE WEIBULL DISTRIBUTION FROM
C** T=OCUINT TO T=TI
C** ___      ____
C
         RDAT=PROOCW(OCUINT,ITINTR,THETA,ETA)
C**
C**NOW CALCULATE AT
C**
          AT=RNAT/RDAT-OCUINT
C
C______________________________________________________________________
C
C**CINTRM IS THE 'ADJUSTED' INTERM, THAT IS , THE TIME IN CONTROL IS
C**DEDUCTED  FROM INTREM
C
      CINTRM=INTERM-AT
C
```

```
C**OCUINT IS THE OUTER CUMULATIVE INTERVAL , THAT IS , T SUB (I-1)
C**PROOCX IS THE PROB. OF GOING OUT OF CONTROL IN THE INTERVAL I.
C
      ADBP=ADBP+CINTRM*PROOCW(OCUINT,ITINTR,THETA,ETA)
C
C**UPDATE OCUINT FOR THE NEXT ITERATION
C
      OCUINT=OCUINT+ITINTR
C
200   CONTINUE
C
      RETURN
      END
C
C
C
C
C
C*********************************************************************
C*********************************************************************
C*********************************************************************
        SUBROUTINE  MYGAMA(A,GPRM,RIA,IER)
C*********************************************************************
C**                                                                 *
C** THIS ROUTINE CALCULATES THE INCOMPLETE GAMMA INTEGRAL.          *
C**                                                                 *
C*********************************************************************
C**
C
C   THIS SUBROUTINE CALLS IMSL ROUTINE MDGAM TO GET THE SINGLE VALUE
C   VARIABLE FOR RIA WHICH IS THEN CONVERTED TO DOUBLE PRECISION
C*
         DOUBLE PRECISION  A,GPRM,RIA
         SA=A
         SGPRM=GPRM
         CALL MDGAM(SA,SGPRM,SRIA,IER)
         RIA=SRIA*1.D0
         RETURN
         END
C
C
C
C
C*********************************************************************
C*********************************************************************
C*********************************************************************
      SUBROUTINE  CMAINT(BAHCM)
C*********************************************************************
C**                                                                 *
C** THIS ROUTINE CALCULATES THE AVERAGE HOURLY COST OF MAINTAINING   *
C** THE CONTROL CHART.                                              *
C**                                                                 *
C*********************************************************************
C**
C*
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*8  NUM,INTERM,LAMBDA,M,L,ITINTR,IM,ID,IL
      COMMON / MAIN1 /LUR,LUW
      COMMON / DCDY1 / DELTA, B,C,DD,E,VZMV1,T,W
      COMMON / DYNM1 / THETA,ETA
      COMMON / CMN1 / CUPROX
C**
C**TETHA CORRESPONDS TO LAMBDA ( ALPHA)     ETA CORRESPONDS TO BETA
C**
         COMMON/A1111/L,IL,M,IM,D,ID
         COMMON/B1111/ISTEPS
```

```
      COMMON/BE1/ITR
      DIMENSION A(9999),P(9999),AINTR(9999),ASAMP(9999),ACNTL(9999)
      DIMENSION S(260),AS(260)
C
C INITIALIZE THE ARRAYS
C
      DATA A/9999*0.0D0/
      DATA P/9999*0.0D0/
      DATA AINTR/9999*0.0D0/
      DATA ASAMP/9999*0.0D0/
      DATA ACNTL/9999*0.0D0/
CC
C**
C**IN ORDER TO MAKE CALCULATIONS EFFICIENT AND FAST THE
C**COST VALUES ARE FIRST CALCULATED AND STORED IN ARRAY A AND
C**THEN UTILIZED.
C**SIMILARLY PROB OF CATCHING THE SHIFT ON  ANY OF THE SAMPLES
C**ARE FIRST CALCULATED AND STORED IN ARRAY P AND THEN UTILIZED.
C**
       TI=0.
C**
C**IFLAG IS USED TO SPECIFY IF SAMPLING SHOULD BE DONE DURING
C** THE SEARCH FOR THE ASSIGNABLE CAUSE OR NOT.
C** IFLAG=1 ; TAKE SAMPLES; LIKE DUNCAN'S MODEL
C** IFLAG=0 ; DONOT TAKE SAMPLES DURING SEARCH ....
C**
        IFLAG=1
C**
C**DELTA IS AMOUNT OF SHIFT IN THE PROCESS ( DELTA IS ASSUMED TO BE
C**POSITIVE. NOTE THAT PROB. CALCULATIONS NEED SOME MODIFICATIONS
C** IF  DELTA IS NEGATIVE)
C**
C**
       ITINTR=IL/L
       CNTLMT=IM/M
       SAMPLI=ID/D
C**
C**FIRST CALCULATE ALL  ITINTR,S ; H SUB I,S
C**                     CNTLMT,S ; K SUB I,S
C**                     SAMPLI,S ; N SUB I,S , AND STORE THEM IN
C** ARRAYS AINTR, ACNTL, AND ASAMP.
C**
       ILT=ITR+200
C
       DO 11 I=1,ILT
C**
C**ITINTR IS THE LENGTH OF THE ITH INTERVAL; H SUB I
C**CNTLMT IS THE WIDTH OF THE ITH CONTROL LIMITS ; K SUB I
C**SAMPLI IS THE SIZE OF ITH SAMPLE ;N SUB I
C**
             ITINTR=ITINTR*L
             AINTR(I)=ITINTR
             CNTLMT=CNTLMT*M
C+=+=+
C+=+CONTROL LIMIT SPREAD STRICTED BETWEEN .5 AND 5.5
C+=+=+
             IF(CNTLMT.LT.0.5)CNTLMT=0.5
             IF(CNTLMT.GT.5.5)CNTLMT=5.5
C+=+=+
             ACNTL(I)=CNTLMT
C++
C++SAMPLE SIZE SHOULD BE INTEGER. E.G. IF SAMPLE SIZE IS BETWEEN 5.5
C++TO 6., THEN IT IS SET TO 6.
C++
             SAMPLI=SAMPLI*D
             ISAMPL=SAMPLI+.4999999D0
```

```
            IF(ISAMPL.LT.2)ISAMPL=2
C+=+=+=+
C+=UPPER LIMIT FOR SAMPLE SIZE IS 1000.
C+=+=+=+
            IF(ISAMPL.GT.1000)ISAMPL=1000
C+=+=+=
            ASAMP(I)=ISAMPL
C++
C++
          Y=CNTLMT-DELTA*DSQRT(ASAMP(I))
          CALL MDNORD(Y,XX)
C**
C**P(I) IS THE PROB. OF DETECTING THE SHIFT ON THE ITH SAMPLE
C**NOTE THAT IN THE CALCULATION OF THIS PROBABILITY IT IS ASSUMED THAT
C***DELTA IS POSITIVE.
C**
           P(I)=1.-XX
C
 11    CONTINUE
C
       PANUM=0.
       NUM=0.
       PADENM=0.
       DENM=0.
C**NOTE THAT ILT2=ISTEPS+113
        ILT2=ITR+100
C
       DO 12 I=1,ILT2
C**PANUM IS THE VALUE OF NUMERATOR IF IFLAG=0. IT IS ALSO NEEDED IN
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
C**
        PANUM=PANUM+B+C*ASAMP(I)
        IF(IFLAG.NE.1) GO TO 124
C**
C**
C**COST OF SAMPLING DURING THE SEARCH FOR THE ASSIGNABLE CAUSE
C** SININT IS THE SUM OF INTERNAL INTERVALS(USED IN THE UPPER LIMIT
C** OF SUMMATION USED IN CALCULATION OF NUMERATOR)
C**
C**FIRST SET NUMERATOR EQUAL TO PANUM
        NUM=PANUM
        SININT=AINTR(I+1)
        TPFAC=E*ASAMP(I)+DD
           J=I
        IF(SININT.GT.TPFAC)GO TO 123
  122       J=I+1
            SININT=SININT+AINTR(J+1)
C
C**
C***NOTE THAT WHEN IFLAG=1 THEN IF INTERVAL SIZE,AINTR, GETS SMALLER
C** THAN .05*IL WE FIX THE INTERVAL SIZE AT THAT VALUE ;.05*IL, IN
C** ORDER TO CALCULATE THE REMAINING NUMBER OF SAMPLES TAKEN DURING THE
C** SEARCH FOR THE ASSIGNABLE CAUSE; THAT IS DURING E*ASAMP(I)+DD OR
C** E*ASAMP(I)+DD SBINT.
C**
C***ALSO, IN ORDER TO CALCULATE THE DENOMINATOR (WHETHER IFLAG=0 OR 1
C** IF AINTR GETS SMALLER THAN .05*IL THEN THE INTERVAL SIZE IS FIXED
C** AT THAT VALUE; .05*IL.
C**
                   IF(AINTR(J+1).LT. .05*IL)GO TO 1221
            NUM=NUM+B+C*ASAMP(J)
        IF(SININT.LE.TPFAC)GO TO 122
        GO TO 123
C**
```

```
C**FIND THE INTEGER NUMBER OF THE REMAINING INTERVALS
C**
1221        SBINT=SININT-AINTR(J+1)
            IREMN=(E*ASAMP(I)+DD-SBINT)/(.05*IL)
C**
C*SO WE GET TO SININT=SININT+IREMN*(.05)*IL
             NUM=NUM+IREMN*(B+C*ASAMP(J))
C**DONOT INTERPOLATE FOR THIS CASE.
            GO TO 124
C**
C**DENM AT THIS POINT IS TI
C**
C**INTERPOLATE FOR THE COST
C**
 123      SBINT=SININT-AINTR(J+1)
          NUM=NUM+(TPFAC-SBINT)/AINTR(J+1)*(B+C*ASAMP(J+1))
C**
C** SET THE INTERVALS EQUAL TO .05*IL IF THEY ARE < OR = TO .05*IL
C**
124       IF(AINTR(I).LE..05*IL)AINTR(I)=.05*IL
      PADENM=PADENM+AINTR(I)
         IF(IFLAG.EQ.0)NUM=PANUM
         DENM=PADENM+E*ASAMP(I)+DD
C
          A(I)=NUM/DENM
C
 12       CONTINUE
C**
C**CALCULATION OF BAHCM
C**
      BAHCM=0.
      FRSTIN=0.
      OCUINT=0.
      ITINTR=IL/L
      ILT3=ILT2-113
C**NOTE THAT ILT3=ISTEPS
      I=0
      CUPROX=0.
C
 35   I=I+1
C
      PRODP=1.
      PRODPL=0.
      INTERM=0.
C**
C**ITINTR IS THE LENGTH OF THE ITH INTERVAL ; H SUB I
C**
       ITINTR=ITINTR*L
       J=0
C--------------------------------------------
40       J=J+1
C**
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
C**PROB. OF OUT OF CONTROL. IN OTHER WORDS INTERM IS THE COST OF
C**MAINTAINING THE CHART GIVEN THAT THE PROCESS GOES OUT OF CONTROL IN
C**THE ITH INTERVAL.
C**
C
C**PRODPL IS LN OF PRODP
C**ALN IS LN OF A(I+J-1)
C**PLN IS LN OF P(I+J-1)
C
C+
              IF(I+J.GE.ITR+99)GO TO 401
```

```
C+
              ALN=DLOG(A(I+J-1))
              PLN=DLOG(P(I+J-1))
              PAINTL=PRODPL+PLN+ALN
C**
C**NOTE THAT WHEN PAINTL IS LESS THAN -20. THEN INTERM=INTERM+0.
C++   OR NOTHING IS ADDED TO INTERM. SO, TERMINATE THE LOOP; THE
C++   NUMBER OF ITERATIONS FOR J IS ENOUGH.
C++
             IF(PAINTL.LT.-20.) GO TO 41
             IF(PAINTL.GT.-20.)INTERM=INTERM+DEXP(PAINTL)
C**
C**UPDATE PRODPL FOR THE NEXT ITERATION
C**
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.
C++BEFORE) ARE ZERO. SO, NOTHING WILL BE ADDED RO INTERM. THERFORE,
C++TERMINATE THE LOOP.
C++
             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
C
      GO TO 40
C--------------------------------------------
C
C++
401           WRITE(6,4031)I+J-1
4031          FORMAT(1X,'VALUE OF A(',I4,') IS NOT DEFINED . ITERATIONS'
     *   , 'TERMINATED')
C++
C**
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.)
C**
 41          PROX=PROOCW(OCUINT,ITINTR,THETA,ETA)
           BAHCM=BAHCM+INTERM*PROX
C**CUPROX IS CUMULATIVE SUM OF PROBABILITIES OF OUT OF
C**CONTROL USED FOR TERMINATING THE
          CUPROX=CUPROX+PROX
C**
C**UPDATE OCUINT FOR THE NEXT ITERATION. OCUINT IS THE CUMULATIVE
C**SUM OF INTERVALS; OCUINT USED IN PROOCX, ABOVE, IS T SUB (I-1).
C**
 667        OCUINT=OCUINT+ITINTR
            IF(I.LT.ILT3) GO TO 35
C
 669  RETURN
      END
C
C
C
C*****************************************************************
C*****************************************************************
C*****************************************************************
      DOUBLE PRECISION FUNCTION PROOCW(OCUINT,ITINTR,LAMBDA,BETA)
C*****************************************************************
C**                                                             *
C** THIS ROUTINE CALCULATE THE AREA UNDER A WEIBULL DENSITY     *
C** BETWEEN OCUINT AND ITINTR.                                  *
C**                                                             *
C**                                                             *
C*****************************************************************
C**
C*
      IMPLICIT REAL*8(A-H,O-Z)
       REAL*8 LAMBDA,ITINTR
```

```
C
C**LAMBDA CORRESPONDS TO THETA   BETA CORRESPONDS TO ETA.
C
C**OCUINT IS T SUB I-1
C
C** LET TI BE T SUB I
C
      TI=OCUINT+ITINTR
C
C**FIRST CHECK FOR UNDERFLOW
C
      DUM1=-(LAMBDA*OCUINT)**BETA
      DUM2=-(LAMBDA*TI)**BETA
C
C**NOTE THAT  DUM1=-LAMBDA*T SUB (I-1) AND
C** DUM2=-LAMBDA*T SUB I  , SO  |DUM2| .GT. |DUM1|
C** THAT IS IF DUM1 IS .LT.  -70 , THEN DUM2 IS .LT. -70
C
      IF(DUM2.LT.-70.)PROOCW=0.
      IF(DUM2.LT.-70.)RETURN
C
      PROOCW=DEXP(DUM1)-DEXP(DUM2)
C
      RETURN
      END
C
C
C
C
C*******************************************************************
C*******************************************************************
C*******************************************************************
      SUBROUTINE FALSA(ENFALS)
C*******************************************************************
C**                                                                *
C** THIS ROUTINE CALCULATE THE EXPECTED NUMBER OF FALSE ALARMS.     *
C**                                                                *
C**                                                                *
C*******************************************************************
C**
C*
C
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*8  NUM,INTERM,LAMBDA,M,L,ITINTR,IM,ID,IL
      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
          COMMON/A1111/L,IL,M,IM,D,ID
          COMMON/B1111/ISTEPS
          COMMON/BE1/ITR
C**
      ENFALS=0.
      OCUINT=0.
      CNTLMT=IM/M
      CUALPH=0.
      ITINTR=IL
C**
C**NOTE THAT LT IS ISTEPS
C**
      ILT=ITR-13
C
      DO 20 I=1,ILT
C**
C**MDNOR IS AN IMSL ROUTINE THAT CALCULATES NORMAL DENSITY INTEGRAL.
C**
```

```
C**
C**CNTLMT IS THE WIDTH OF ITH CONTROL LIMIT
C**
      CNTLMT=CNTLMT*M
C+=+=+
C+=+CONTROL LIMIT SPREAD RESTRICTED BETWEEN .5 AND 5.5
C+=+=+
      IF(CNTLMT.LT.0.5)CNTLMT=0.5
      IF(CNTLMT.GT.5.5)CNTLMT=5.5
C+=+=+
      CALL MDNORD(CNTLMT,XX)
C**
C**ALPHA IS THE PROB. OF A FALSE ALARM IN THE ITH INTERVAL
C**
      ALPHAI=2.*(1.-XX)
C**
C**CUALPH IS CUMULATIVE SUM OF ALPHAS; ALPHA1+ ALPHA2+ ...+ ALPHA(I-1)
C**
      CUALPH=CUALPH+ALPHAI
C**
C**ITINTR IS THE LENGTH OF ITH INTERVAL ; H SUB I
C**
C**OCUINT IS CUMULATIVE SUM OF INTERVALS; T SUB I
C** OR IT IS  T SUB (I-1) FOR FOR PROB. OF GOING OUT OF CONTROL IN
C** (I+1)TH INTERVAL.
C**
        OCUINT=OCUINT+ITINTR
        ENFALS=ENFALS+CUALPH*PROOCW(OCUINT,ITINTR,THETA,ETA)
C**
C**UPDATE ITINTR FOR THE NEXT ITERATION
C**
       ITINTR=ITINTR*L
C++
       IF(ITINTR.LT..05*IL)ITINTR=.05*IL
C
  20  CONTINUE
C**
      RETURN
      END
C
C
C
C
C
C*********************************************************************
C*********************************************************************
C*********************************************************************
      DOUBLE PRECISION FUNCTION SINTW(HF,ISTEPS)
C*********************************************************************
C**                                                                 *
C**THIS ROUTINE CALCULATES THE VALUE OF IH=SINTW (INITIAL SAMPLING  *
C**INTERVAL) FOR ANY GIVEN VALUES OF HF, ISTEPS, PROBPT, AND WEIBULL*
C**DISTRIBUTION PARAMETERS.                                         *
C**                                                                 *
C*********************************************************************
C
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*8 IH,HF
     COMMON / DYNM4 / PROBPT
     COMMON / DYNM1 / THETA,ETA
C**
C**CALCULATE IL ; INITIAL INTERVAL SIZE TO ACHIEVE THE QUANTILE
C**OF PROBPT,SPECIFIED BY USER, IN ISTEPS ITERATIONS
C**
C
C**
```

```
C**FIND THE VALUE THAT THE SUM OF ISTEPS INTERVALS SHOULD BE EQUAL TO.
C**
      SUMATN=DEXP(DLOG(-DLOG(1.D0-PROBPT))/ETA-DLOG(THETA))
C**
C**CALCULATION OF IH=SINTW, IF HF IS 1.
C**
      IF(HF.EQ.1)SINTW=SUMATN/DFLOAT(ISTEPS)
      IF(HF.EQ.1)RETURN
C**
C**CALCULATION OF IH=SINTW, IF HF IS NOT EQUAL TO 1.
C**
      BX=ISTEPS*DLOG(HF)
      IH=SUMATN*(1.D0-HF)/(1.D0-DEXP(BX))
      SINTW=IH
      RETURN
      END
C
C
C
C*******************************************************************
C*******************************************************************
      DOUBLE PRECISION FUNCTION SINTX(HF,ISTEPS)
C*******************************************************************
C**                                                               *
C**THIS ROUTINE CALCULATES THE VALUE OF IH=SINTW (INITIAL SAMPLING *
C**INTERVAL) FOR ANY GIVEN VALUES OF HF, ISTEPS, PROBPT, AND      *
C**EXPONENTIAL DISTRIBUTION PARAMETER.                            *
C**                                                               *
C*******************************************************************
C
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*8 IH,HF
      REAL*8 LAMBDA
     COMMON / DUNC1 / LAMBDA
     COMMON / DYNM4 / PROBPT
C**
C**CALCULATE IL ; INITIAL INTERVAL SIZE TO ACHIEVE THE QUANTILE
C**OF PROBPT,SPECIFIED BY USER, IN ISTEPS ITERATIONS
C**
C
C**
C**FIND THE VALUE THAT THE SUM OF ISTEPS INTERVALS SHOULD BE EQUAL TO.
C**
      SUMATN=1.D0/LAMBDA*(-DLOG(1.D0-PROBPT))
C**
C**CALCULATION OF IH=SINTX, IF HF IS 1.
C**
      IF(HF.EQ.1)SINTX=SUMATN/DFLOAT(ISTEPS)
      IF(HF.EQ.1)RETURN
C**
C**CALCULATION OF IH=SINTX, IF HF IS NOT EQUAL TO 1.
C**
      BX=ISTEPS*DLOG(HF)
      IH=SUMATN*(1.D0-HF)/(1.D0-DEXP(BX))
      SINTX=IH
      RETURN
      END
C
C
C
C
C
C
C
C*******************************************************************
C*******************************************************************
```

```
C*********************************************************************
      DOUBLE PRECISION FUNCTION CSTPW(HF,IH)
C*********************************************************************
C**                                                                  *
C**THIS ROUTINE CALCULATES THE VALUE OF ISTEPS=CSTPW (NUMBER OF STEPS*
C**TO GET TO A GIVEN QUANTILE POINT, PROBPT) FOR GIVEN VALUES OF     *
C**HF, IH, PROBPT, AND WEIBULL DISTRIBUTION PARAMETERS.              *
C**                                                                  *
C*********************************************************************
C
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*8 IH,HF
     COMMON / DYNM4 / PROBPT
     COMMON / DYNM1 / THETA,ETA
C**
C**CALCULATE IL ; INITIAL INTERVAL SIZE TO ACHIEVE THE QUANTILE
C**OF PROBPT,SPECIFIED BY USER, IN ISTEPS ITERATIONS
C**
C
C**
C**FIND THE VALUE THAT THE SUM OF ISTEPS INTERVALS SHOULD BE EQUAL TO.
C**
      SUMATN=DEXP(DLOG(-DLOG(1.D0-PROBPT))/ETA-DLOG(THETA))
C**
C**CALCULATION OF IH=SINTW, IF HF IS 1.
C**
      IF(HF.EQ.1)ISTEPS=SUMATN/IH
      IF(HF.EQ.1)CSTPW=ISTEPS
      IF(HF.EQ.1)RETURN
C**
C**CALCULATION OF IH=SINTW, IF HF IS NOT EQUAL TO 1.
C**
      AX=SUMATN*(1.D0-HF)/IH
      ISTEPS=DLOG(1.D0-AX)/DLOG(HF)
      CSTPW=ISTEPS
      RETURN
      END
C
C
C
C
C*********************************************************************
C*********************************************************************
C*********************************************************************
      DOUBLE PRECISION FUNCTION FCTR(DORM,DORME)
C*********************************************************************
C**                                                                  *
C**THIS ROUTINE CALCULATES THE VALUE OF D OR M FOR ANY GIVEN VALUES *
C** OF DORM ( ID OR IM ) AND DORME (IDENDING OR IM ENDING )         *
C**                                                                  *
C**                                                                  *
C*********************************************************************
C
      IMPLICIT REAL*8(A-H,O-Z)
     COMMON / DYNM2 / ISTEPS
C
C
      QDORM=DORME/DORM
C
      QISTM1=ISTEPS-1
C
      FCTR=DEXP(DLOG(QDORM)/QISTM1)
C
      RETURN
      END
C
```

```
C
C
C
C
C********************************************************************
C********************************************************************
C********************************************************************
      DOUBLE PRECISION FUNCTION SETDEL(DELIME)
C********************************************************************
C**                                                                 *
C**THIS ROUTINE CALCULATES THE VALUE OF DEL FOR KF (NF AND HF) GIVEN*
C**THE DESIRED INCREMENT IN IM ENDING OR ID ENDING ; DELIME.         *
C**                                                                 *
C**INITIAL VALUE OF KF IS ASSUMED TO BE 1. HOWEVER , THE RESULTS    *
C** SHOULD BE QUITE GOOD EVEN WHEN KF IS NOT 1 BUT IN THE RANGE OF  *
C**  .99 TO 1.1 .                                                   *
C**                                                                 *
C********************************************************************
C
      IMPLICIT REAL*8(A-H,O-Z)
     COMMON / DYNM2 / ISTEPS
C
C
      DELIME=DABS(DELIME)
C**
C**FCTIME IS THE MULTIPLE OF IM ENDING
C**
      FCTIME=1.D0+DELIME
C**
      QISTM1=ISTEPS-1
C**
C**RNEWM IS THE NEW VALUE OF M
C**
      RNEWM=DEXP(DLOG(FCTIME)/QISTM1)
C**
C**SETDEL IS THE INCREMENT IN KF OR NF.
C**
      SETDEL=RNEWM-1.D0
C
      RETURN
      END
C
C_____________________________________________________________________
```

VITA

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.