10-24-2007

A Dynamic Inventory/Maintenance Model

Jonathan J. Bates
University of South Florida

A Dynamic Inventory/Maintenance Model

by

Jonathan J. Bates

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
Department of Industrial and Management Systems Engineering
College of Engineering
University of South Florida

Major Professor: Jose L. Zayas-Castro, Ph.D.
Tapas K. Das, Ph.D.
Thom Hodgson, Ph.D.
Louis Martin-Vega, Ph.D.
Alex Savachkin, Ph.D.
Carlos Smith, Ph.D.

Date of Approval:
October 24, 2007

Keywords: Predicted Demand Forecasting, Censoring, Reliability Test, Weibull Estimation, Stocking Policy

DEDICATION

To my beautiful wife Melissa and our daughter Madelynn who brings color to my life.

Praise be to God for making all things possible.

ACKNOWLEDGEMENTS

I would like to thank Dr. Zayas-Castro for his direction and support throughout the course of this research. I also appreciate and support his leadership and vision for the IMSE Department at University of South Florida. I would also like to thank my committee members for their guidance.

Outside of the academic environment, I thank the U.S. Coast Guard for the opportunity to pursue this effort and the continued support throughout the research process. My family, friends, and supervisors were also incredibly supportive while encouraging and prodding me to keep going.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

A DYNAMIC INVENTORY/MAINTENANCE MODEL

Jonathan J. Bates

ABSTRACT

A model is proposed to provide inventory and maintenance guidance for a system of operating parts. This model is capable of handling a system with multiple operating components, unknown part lifetime failure distribution, and separately maintained parts. In this model, part reliability characteristics are used along with system costs to predict the required stocking levels and part replacement times. Two maintenance strategies are presented that have the unique characteristic of allowing flexible scheduling of replacements. A case study is completed comparing developed stocking policies to an existing policy. An estimation selection method is introduced and fit into the model for computing Weibull distribution parameters when part reliability is not well known. An algorithm is displayed that describes the implementation of the system model and data from practical case scenarios are conducted using this algorithm.

CHAPTER 1

INTRODUCTION

1.1 Motivation

Fleet oriented organizations, that is, organizations that utilize mobile, complex assets to perform organizational goals, rely on proper maintenance and suffer greatly if the operational time of those assets is diminished. Examples of "fleet oriented organizations" include airlines, shipping companies (utilizing trucks, airplanes, ships etc.), and military organizations among others. Additionally, as these assets increase in complexity and cost, the intricacy and cost of spare parts will also increase. Costs associated with maintenance and reliability, including inventory costs, are on the rise. The commercial airline industry spent over $36.1 billion in 2004 on maintenance and reliability and has a large amount of money tied up in unnecessary inventory [1]. The market analysis firm AeroStrategy is forecasting a 5.6% growth rate in maintenance and reliability spending for the commercial airline industry over the next decade and by 2014, this type of spending will exceed $62 billion [2]. These problems are not limited to the commercial airline industry. Every year, the U.S. General Accounting Office (GAO) issues reports to the U.S. Congress citing the need for improvement and high costs of maintenance and reliability within the Department of Defense. These reports recommend that Congress and listed government agencies take specific actions to decrease costs while providing the public the same level of service [3-6].

Inventory theory has been studied for years, with a significant amount of work completed since the 1950's. Despite all this effort, successful practical application has not been achieved in most industries. Optimal inventory policies generally require "complete knowledge" of the part's demand distribution. This notion of complete knowledge is a phrase that suggests a distribution's type and parameters are confidently known. However, it is important to be able to explain the reasons for the demand. In the retail goods world, demand represents the active desire of a consumer to own a product. With many products, the consumer's desire may vary over time or with seasons. When dealing with a dynamic market, understanding the events that produce product desirability is just as important as having complete historical knowledge of the demand distribution. These causes for demand can be used to predict changes in demand, rather than updating the distribution with recent historical information. It can be argued that complete knowledge is not obtained until the underlying causes for demand are known.

What are the causes for a demand on inventory? This study will specifically look at the area of spare parts demand. For spare or reparable parts, the underlying causes for demand can be found in the maintenance policies and reliability information of the individual components. A demand is created when either a part is required due to maintenance or the part failed and therefore a replacement is needed. This research is guided by a unique concept, building a methodology to model a spare parts inventory/maintenance system using the causes for demand, the maintenance and failure events. Using this idea, an approach is outlined for a jointly modeled inventory and maintenance system that is subject to existing or expected operational and budgetary constraints.

The building blocks for this methodology include contributions from inventory and reliability theory. Chapter 2 contains an introduction of some inventory theory concepts, a review of existing research, and solution methods to select an optimal stationary inventory policy, approximate solutions, and a non-stationary inventory policy. The selection of a maintenance policy is discussed in Chapter 3 along with a review of relative research in the area of joint inventory/maintenance policies. The construction of the predicted demand forecast is discussed in Chapter 4. Chapter 5 outlines an estimation selection method that will be used to develop the part failure distribution. The system model is assembled, tested, and analyzed in Chapter 6, followed by concluding remarks and a discussion of possible research extensions in Chapter 7.

1.2 System Description

The following is a description of the system being modeled in very general terms; this general description is employed to widen the utilization of the study. The "system" refers to a collection of n operating parts either in various geographical locations or grouped together. The term "parts" can refer to a functioning collection of parts such as an engine or a component level part such as a piston. The parts have generally the same, but not identical operational schedules. Additionally, the "birth-dates" of the parts can vary. The failure rates of the parts are identical and are either of the constant or increasing type, but do not have to be well known. This allowance makes it possible for this model to be employed by organizations that do not have well established reliability programs. The system also includes an inventory of parts that are available for installation if an operating part replacement is necessary. As a part fails or is replaced due to maintenance, a demand is placed on the inventory stock. The costs associated with the maintenance and inventory, along with system variables, are introduced in later chapters.

1.3 Research Objectives and Contributions

Only a limited number of studies have been completed examining the interaction of maintenance, reliability, and inventory concepts. The interaction between them is clearly evident and a holistic study is necessary to ensure an efficient system approach is available to organizations and managers. The application of this study is targeted for use in enterprises such as commercial and public transportation, militaries, and organizations that may operate fleets of vehicles. However, the completed research and resulting decision tool would be useful to any organization that carries part inventories for use in an equipment maintenance program including manufacturing and utility plants.

The objective is to provide a strategy to obtain a combined inventory/maintenance model, where the *primary* input parameters of part reliability and inventory/maintenance costs determine the stocking levels and maintenance policy. This strategy can serve as a decision tool for organizations to make justifiable and cost saving policy changes. The combined model should be dynamic in that the inventory and maintenance policies may vary with changing operational conditions and organizational objectives. The difficult task in this research is to provide a linking mechanism between inventory and maintenance in a way that provides decision makers with the ability to adopt new policies under anticipated or expected conditions. Often, organizations that maintain their own equipment do not have sufficient information to predict when parts will fail. Existing literature in maintenance planning often assumes that failure times are known perfectly. This study addresses this issue and provides a method to implement this model given little or zero part reliability information

The area of combined inventory/maintenance modeling is not a complete field and most studies that have been conducted deal only with simplified and static parameters, such as known failure distributions and limited allowable spares. Existing studies often assume that machines are maintained simultaneously. An integrated approach to determine a maintenance policy and inventory stocking rule for a dynamic system is not available in the existing literature. The approach in this study is to present a solution with a wide range of applicability. The contributions of this research include advancements and additions to the available academic literature in inventory theory and reliability analysis, as well as a practical solution to a significant problem that exists throughout many industries. Although the resulting methodology will utilize several existing ideas, the manner in which these ideas are weaved together along with some unique concepts has not been accomplished to date.

CHAPTER 2

INVENTORY THEORY

2.1 (*s, S*) Inventory Policy

A common aspect of most spare parts inventories is the demand of these parts is relatively low or intermittent. The (*s, S*) inventory system has been shown in several studies to be the best performing for items with such demand [7]. The (*s, S*) model is periodic-single item inventory system and is an optimal policy for systems that meet the following assumptions: independent and identically distributed demand, ordering costs are linear plus a fixed setup cost, and all other periodic costs are linear (shortage and holding) [8].

The contributions of the existing (*s, S*) inventory model literature can be grouped into three general categories: model formulation and characteristics, optimal solutions, and approximate solutions. Arrow et al. [9] first formulated this model by examining a dynamic system with demand as a random variable with known distribution. The order-up-to and reorder levels are determined as functions of ordering cost, penalty cost, and demand distribution. [10] showed that the (*s, S*) policy is always the optimal policy if holding and shortage costs are linear [8, 11, 12] contribute to the area of model characteristics and derive bounds on the model's parameters, *s* and *S*.

In this model, *s* is the safety stock level and *S* is the maximum replenishment level. If inventories are reviewed on a periodic basis, the on-hand level plus the amount on order, hereby called s_r, could have the range s_r greater than s or s_r less than or equal to s, where s again is the safety stock level. If s_r falls into the second range, an order should be placed of quantity $S - s_r$, where S is the maximum replenishment level. In contrast to a periodic review model, if the inventory is continuously reviewed, the order quantity is always equal to $S - s$ and s_r will never be less then s. Figure 1 shows an example of an (s, S) inventory model where $S = 15$ and $s = 5$. At time unit 13, the value s_r is equal to s so an order of $S - s_r$ (10 units) is placed. At time unit 14, s_r is equal to 15, 5 units on-hand and 10 units on order. This order is delivered and is added to the on-hand amount at time unit 17.

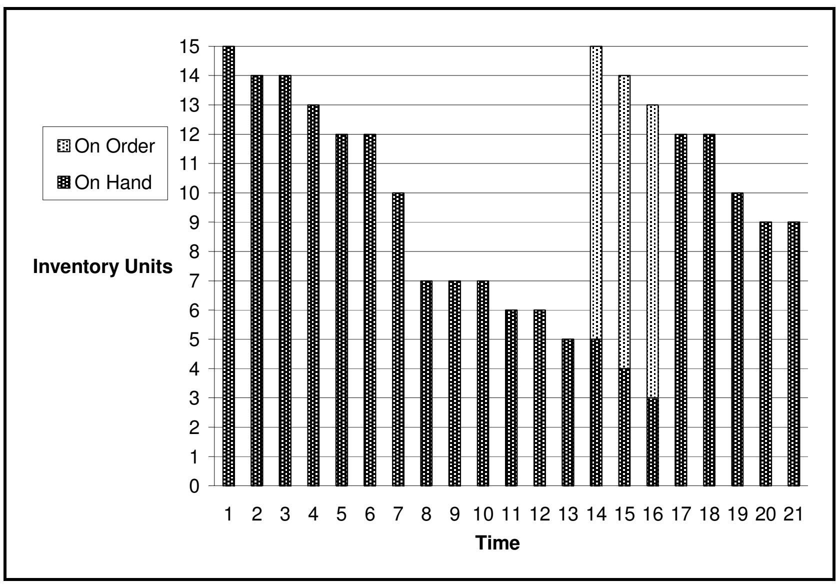

Figure 1 (*s, S*) Inventory Example

The solution to the single-item, stochastic inventory problem was first conceived through a small inventory control conference at RAND during the summer of 1950 [13]. This conference was organized by the Office of Naval Research and it brought together

Jacob Marschak and Kenneth J. Arrow. From the discussion during this conference, Arrow and Marschak thought it was important to investigate a realistic model that was a combination of the two main kinds of existing inventory models. The first existing inventory model was one where inventory could be carried over from one period to the next to fulfill a constant demand. The other model was a one-period model with stochastic demand. The combination of these two models represented a more realistic scenario where inventory is carried over from one period to the next and demand is random. At this time, only solutions of the *(s, S)* form were considered and the underlying problem was identified as a Markov Decision Process. The third author of [9], Theodore Harris, was brought on to analyze this Markov process and subsequently provided the method to determine the discounted expected cost under a fixed *(s, S)* policy. "Optimal Inventory Policy" inspired a large amount of research in dynamic inventory policy including several papers showing the theoretical optimality of the *(s, S)* policy as well as several variations of optimal and approximate solutions to the problem. Additionally the topic of dynamic programming, as coined by Richard Bellman, has an origin credited to this paper [13]. It is interesting to note that Kenneth J. Arrow first referred to this two-bin policy as the *(S, s)* policy and this remained the models title until Donald Iglehart's paper in 1963 re-titled it as the *(s, S)* policy [8]. This rearranged title seems to currently dominate although Arrow seems to prefer the latter *(S, s)* designation [13]. Many papers were written related to the topic of optimal inventory theory over the next few years, including [11, 14-23], some of which are part of the now fabled "Stanford Studies", but it was not until [10], that a proof was offered for the optimality of the *(s, S)* model.

2.2 Optimal Stationary Solutions

In 1959, Herbert Scarf offered a proof of optimality by showing that if holding and shortage costs can be shown or assumed to be linear, there exists an optimal pair *(s, S)* that minimizes the expected costs over an infinite horizon, thus making the *(s, S)* policy optimal. He then proposed to solve for this optimal pair, the minimum of the one period expected cost *G(y)* must be determined, where $G(y) = c\left(z\right) + L(y)$ is equal to the ordering cost of z units and the holding and penalty costs of incurred for y units. The value y^* that minimizes *G(y)*, is the optimal order-up-to parameter *S*. To solve for the reorder level *s*, simply solve the following equation $G(s) = K + G(S)$, where K is the fixed setup/ordering cost. This solution method is shown in Figure 2. Throughout this chapter and the rest of this study, the following inventory cost definitions will be used:

h – inventory holding cost
p – inventory shortage cost
K – inventory ordering cost
λ – part lead-time

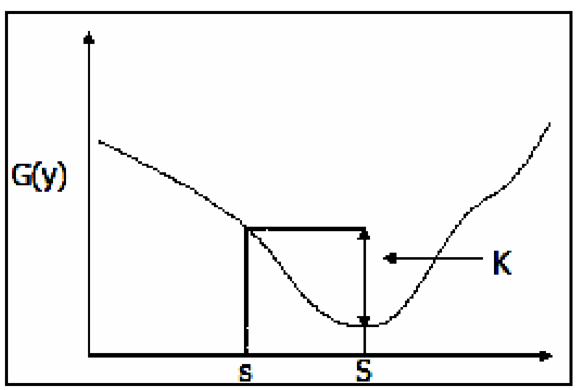

Figure 2 (*s, S*) Inventory Solution Method

In a recent paper, Feng and Xiao [24] developed a new algorithm to search for the optimal pair of *(s, S)* following a tradition of papers including [25-29]. This new algorithm is compared to the most previous approach provided by [29] with numerical tests that show an average savings in evaluation effort of 30%. The proposed algorithm can be successfully implemented for a system with zero lead-time. A clarification to this algorithm was presented in [30] providing a simple modification to implement Feng and Xiao's algorithm when a fixed lead-time exists. This modification is outlined in the remainder of this section.

Following [24], let $G(y)$ equal the average one-period holding and shortage cost when the inventory position at the beginning of the period is *y*; and y_0 equal the largest *y* minimizing $G(y)$. The modification to this algorithm is in the selection of y_0. As stated in [24], y_0 can be obtained by solving a one-period newsboy problem or the solution to the inequality:

$$\Phi\left(y_0\right) < \frac{p}{p+h} \le \Phi\left(y_0+1\right), \tag{2.1}$$

where $\Phi(\cdot)$ is an arbitrary CDF of demand and unit ordering costs are ignored. When a fixed lead-time exists, [24] states that the cost function $G(y)$ can be redefined as in [25]:

$$G^n(y) = \begin{cases} h\sum_{k=0}^{y}(y-k)\varphi^{\lambda+1}(k) + p\sum_{k=y+1}^{\infty}(k-y)\varphi^{\lambda+1}(k) & y \ge 1 \\ p\sum_{k=0}^{\infty}(k-y)\varphi^{\lambda+1}(k) & y \le 0 \end{cases} \tag{2.2}$$

where $\varphi^n(k)$ is the n-fold convolution of $\varphi(k)$ or the demand distribution. When no lead-time is present or $\lambda = 0$, $n = 1$ and it is common to drop the superscripts of this cost function. However, [24] fails to state that using the solution to a one-period newsboy will not find the largest *y* minimizing $G^n(y)$. To obtain y_0 when a lead-time exists, the lower bound of *S* or $\underline{S}$ must first be determined. This can be done by solving for $\underline{S}$ using the inequality given in [25]:

$$\Phi^{\lambda+1}(\underline{S}-1) < \frac{p}{p+h} \le \Phi^{\lambda+1}(\underline{S}). \tag{2.3}$$

[8] showed that $G^n(y)$ is a convex function with a minimum value at y_0. Therefore, with $\underline{S}$ known, y_0 can be described as

$$y_0 = \underline{S} + \Delta_y, \tag{2.4}$$

where Δ_y is the largest positive integer such that

$$G^n(\underline{S}+\Delta_y) > G^n(\underline{S}+\Delta_y+1). \tag{2.5}$$

Thus, y_0 can be found as shown in Figure 3 and Feng and Xiao's algorithm can be run to completion to solve for *s** and *S**, the optimal pair. This modification provides a more general case for a system with a fixed lead-time or with zero lead-time.

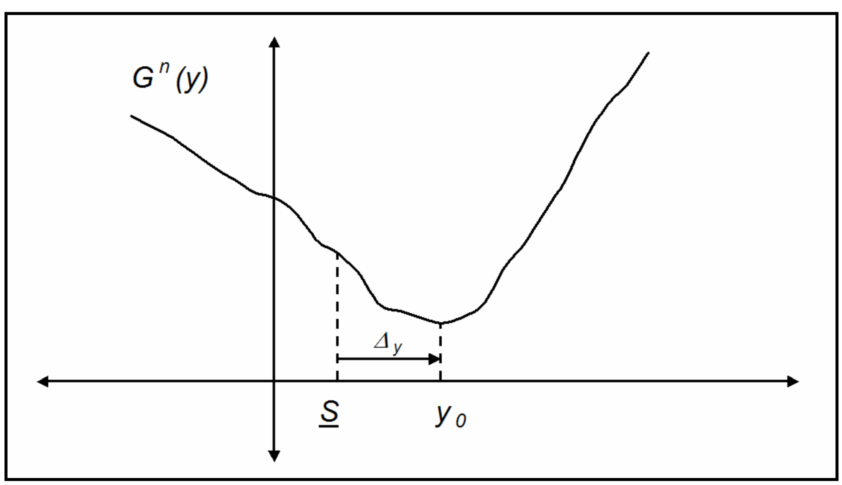

Figure 3 (*s, S*) Solution Modification

Using the modified Feng and Xiao algorithm, the case study shown in Table 3 of [31] is duplicated here as Table 1 where s* and S* are the optimal pair minimizing the long-run average cost function $c(s, S)$. Table 2 duplicates the numerical results of four cases from Feng and Xiao with the additional results for fixed lead-times of 1 and 2.

Table 1 Wagner, O'Hagan. and Lundh Solutions

λ	$\underline{S}$	y_0	s^*	S^*	$c(s^*,S^*)$
0	14	14	9	14	13.7929
1	17	26	20	27	15.9607

Table 2 Modified Feng and Xiao Solutions

$\lambda=0$					
μ	$\underline{S}$	y_0	s^*	S^*	$c(s^*,S^*)$
10	14	14	6	40	35.0215
15	20	20	10	49	42.6978
20	26	26	14	62	49.173
25	32	32	19	56	54.2621
	$\lambda=1$				
μ	$\underline{S}$	y_0	s^*	S^*	$c(s^*,S^*)$
10	17	26	16	51	36.0974
15	17	37	25	65	44.0712
20	35	48	35	83	50.7336
25	44	59	44	83	56.4817
	$\lambda=2$				
μ	$\underline{S}$	y_0	s^*	S^*	$c(s^*,S^*)$
10	23	37	26	62	37.0426
15	36	54	41	82	45.2529
20	48	70	55	104	52.0842
25	61	86	70	109	58.3301

2.3 Approximate Stationary Solutions

Approximate solutions to the (*s, S*) inventory policy general require only limited demand information to determine the inventory parameters. Many approximate methods have been developed and three of the more commonly studied approximate solutions are presented here [31-37]. These methods make use of the demand mean μ and variance σ, along with the inventory costs in determining the inventory values. The first approximate (*s, S*) solution, known as the Normal Approximation, was presented by Donald Roberts in [38]. This approximation utilizes an iterative procedure to solve for the inventory parameters *s* and *S*. This iterative procedure is described in following five steps.

1) Set $Q = \sqrt{\dfrac{2 \cdot K \cdot \mu}{h}}$.

2) Solve $u = \left(1 - \dfrac{h \cdot Q}{\dfrac{h \cdot \mu_L}{2} + \mu \cdot p} \right)$, where $\mu_L = (\lambda + 1) \cdot \mu$.

3) Solve $s = \mu_L + u \cdot \sigma_L$, where $\sigma_L = \sigma \cdot \sqrt{\lambda + 1}$.

4) Redefine Q as $Q = \sqrt{\dfrac{2 \cdot K \cdot \mu}{h} + \left(\mu_L + \dfrac{2 \cdot \mu \cdot p}{h} \right) \cdot \sigma_L \cdot I_N(u)}$, where

   $I_N(\mu) = \int_u^{\infty} (t - u) \cdot \dfrac{1}{\sqrt{2 \cdot p}} \cdot e^{-.5 \cdot t^2} \, dt$ or the standardized Normal loss integral.

5) Repeat steps 2, 3, and 4 until the inventory values *s* and $S = s + Q$ converge.

Additional remarks concerning the implementation of the Normal approximation are given in [39].

The second approximation provides heuristic decision rules for the operating parameters s and S [40]. Five examples from [41] are solved and results from this heuristic are compared to the optimal values. In this method, defining

$$q=\sqrt{\frac{2\cdot K\cdot\mu\cdot\left(1+\frac{h}{p}\right)}{h}}, \tag{2.6}$$

the inventory values are

$$s=\mu_L+\frac{1-q}{2}+\frac{p}{p+h}\cdot\sqrt{\sigma_L^2-\frac{\sigma^2}{2}+\frac{q^2-\mu^2}{12}+\frac{q^2\cdot\sigma^2}{2\cdot\mu}} \tag{2.7}$$

and

$$S=\mu_L+\frac{q-\mu}{2}+\frac{p}{p+h}\cdot\sqrt{\sigma_L^2-\frac{\sigma^2}{2}+\frac{q^2-\mu^2}{12}+\frac{q^2\cdot\sigma^2}{2\cdot\mu}}. \tag{2.8}$$

[42] published a revision of an earlier method [43] known as the Revised Power Approximation. In the previous approximation, a fitted regression model was developed by adjusting the method of Roberts [38] to 288 known inventory policies. [44] shows how the original Power Approximation can be adjusted to prove useful under the following differing conditions; non-stationary demand, correlated demand, or stochastic lead-times. Solving for this method, the initial inventory values are set as

$$s_0=.973\cdot\mu_L+\sigma_L\cdot\left[\frac{.183}{\sqrt{\frac{D_p}{\left(\frac{\sigma_L\cdot p}{h}\right)}}}+1.063-2.192\cdot\sqrt{\frac{D_p}{\left(\frac{\sigma_L\cdot p}{h}\right)}}\right] \tag{2.9}$$

and

$$S_0 = \mu_L + \left(\frac{p}{p+h}\right) \cdot \sigma_L, \tag{2.10}$$

where

$$D_p = 1.3 \cdot \mu^{.494} \cdot \left(\frac{K}{h}\right)^{.506} \cdot \left(1 + \frac{\sigma_L^2}{\mu_L^2}\right)^{.116}. \tag{2.11}$$

The final approximate inventory values are then determined as

$$s = \begin{cases} s_0 \text{ if } \dfrac{D_p}{\mu} > 1.5 \\ \min\left(s_0, S_0\right) \text{ if } \dfrac{D_p}{\mu} \le 1.5 \end{cases} \tag{2.12}$$

and

$$S = \begin{cases} \left(s_0 + D_p\right) \text{ if } \dfrac{D_p}{\mu} > 1.5 \\ \min\left(s_0 + D_p, S_0\right) \text{ if } \dfrac{D_p}{\mu} \le 1.5 \end{cases}. \tag{2.13}$$

2.4 Non-stationary Solution

The optimality of the (*s, S*) inventory policy under linear purchasing costs was extended to non-stationary demand distributions by Samuel Karlin [23]. Karlin presented a policy that utilizes critical numbers to determine if an order is placed or not. This policy allows the critical number to vary from period to period as demand changes.

Only a limited number of studies have been completed that present unique solutions to this non-stationary inventory policy and a dynamic programming solution is the only known optimal solution. Sethi and Cheng [45] present a dynamic programming solution for both finite and infinite horizon problems under Markovian demand. A number of nearly optimal heuristics have been presented [46, 47]. Bollapragada and Morton [46] compared their heuristic with Askin [47] and an optimal solution obtained through dynamic programming. This study found the Bollapragada and Morton method differed from the optimal solution only 1.7% over the cases studied while the Askin method had an error of 2%. Additionally, the study showed that the Bollapragada and Morton heuristic is computationally more efficient.

To solve for the optimal inventory policy under non-stationary demand, a recursive dynamic programming method can be utilized [46]. Let $J_{k,N}(x)$ be the optimal cost from period k to period N if during each period an optimal policy is followed. Therefore,

$$J_{k,N}(x) = \min\left\{L_{k,\lambda}(x) + E\left[J_{k+1,N}(x-\xi_k)\right]\right\}, k \neq N \tag{2.14}$$

and

$$J_{N,N}(x) = L_{N,\lambda}(x), \tag{2.15}$$

where

$$E\left[J_{k+1,N}(x-\xi_k)\right] = \sum_{\xi_k=0}^{\infty}\begin{bmatrix} L_{k+1,\lambda}(x-\xi_k) + \sum_{\xi_{k+1=0}}^{\infty} L_{k+2,\lambda}(S_{k+1}-\xi_{k+1})\cdot\varphi_{k+1}(\xi_{k+1}) + \ldots \\ + \sum_{\xi_{N-1=0}}^{\infty} L_{N,\lambda}(S_{N-1}-\xi_{N-1})\cdot\varphi_{N-1}(\xi_{N-1}) \end{bmatrix}\cdot\varphi_k(\xi_k). \tag{2.16}$$

Also, $L_{k,\lambda}(x)$ is the expected one-period costs where orders placed in period k will be received at the beginning of l+k. While $\lambda \geq k+1$, this expected one-period cost is

$$L_{k,\lambda}(x) = \begin{cases} \sum_{\xi_k=0}^{x} \left[h\cdot(x-\xi_k) + L_{k+1,\lambda}(x-\zeta_k) \right]\cdot\varphi_k(x) + \sum_{\xi_k=x+1}^{\infty} \left[p\cdot(\xi_k - x) + L_{k+1,\lambda}(x-\zeta_k) \right]\cdot\varphi_k(x) & x>0 \\ \sum_{\xi_k=0}^{\infty} \left[p\cdot(x-\xi_k) + L_{k+1,\lambda}(x-\zeta_k) \right]\cdot\varphi_k(x) & x\le 0 \end{cases} \tag{2.17}$$

and when $\lambda < k+1$

$$L_{\lambda,\lambda}(x) = \begin{cases} \sum_{\xi_k=0}^{x} h\cdot(x-\xi_k)\cdot\varphi_k(x) + \sum_{\xi_k=x+1}^{\infty} p\cdot(\xi_k - x)\cdot\varphi_k(x) & x>0 \\ \sum_{\xi_k=0}^{\infty} p\cdot(x-\xi_k)\cdot\varphi_k(x) & x\le 0 \end{cases}. \tag{2.18}$$

To determine the optimal cost, a recursive dynamic programming approach is used. First, $J_{N,N}(x)$ is evaluated for all values of x between the lower and upper bounds of S_N and the lowest value obtained yields the order-up-to parameter S_N. Then

$$J_{N-1,N}(x) = L_{N-1,\lambda}(x) + E\left[J_{N,N}(S_N - \xi_k) \right] \tag{2.19}$$

is evaluated to determine S_{N-1} and this is repeated for $J_{N-2,N}(x)$, $J_{N-3,N}(x)$, …, $J_{k,N}(x)$ to determine S_{N-2}, S_{N-3}, …, S_k. The reorder level parameter is determined by solving for the largest x less than S_k such that

$$J_{k,N}(x) \le K + L_{k,\lambda}(S_k) + E\left[J_{k+1,N}(S_k - \xi_k) \right]. \tag{2.20}$$

Due to the property of K-convexity originally shown by Scarf [10], this value is the reorder level for period k.

CHAPTER 3

MAINTENANCE POLICIES

3.1 Maintenance Policy

Spare parts inventories exist to allow maintenance personnel access to the parts required to properly sustain equipment operations. The demand rate of spare parts is largely dictated by the usage of the equipment in terms of both time and operational environment. The maintenance policy that is applied to the system also plays a key role in the demand rate of a spare part. Obviously, as the operational time increases on a piece of equipment, the possible chance of failure and subsequent repair also increase. "Run to failure" or Failure Based Maintenance (FBM) is a common maintenance philosophy. Other maintenance policies include preventative maintenance and condition based maintenance. Preventative maintenance is a policy that assigns replacement or repair of system parts at assigned units of time and/or cycles regardless of the condition. In condition-based maintenance, inspections or measurements are conducted on the equipment and replacement occurs when a certain condition is found.

As stated in Chapter 2, the optimal inventory computations require exact knowledge of the demand distribution. The ideas presented in this chapter and Chapter 4, are an effort to define the relationship shared with a component's maintenance, failure rate, and inventory policy. To do this, let us examine a simple component that is governed by an increasing failure rate and due to costs associated with a failure event, it is less expensive to replace the component prior to failure. The maintenance and failure events are mutually exclusive - either the part is maintained prior to failure or the part fails and requires replacement. The probability of this part being replaced is determined by the addition of these two mutually exclusive events. Using the probability axiom for the addition of mutually exclusive events,

$$P\{M \cup F\} = P\{M\} + P\{F\},$$

a combined distribution can be obtained. This combined distribution describes the probability of a part replacement or the probability of a part demand. This idea will be expanded upon in Chapter 4.

The following section identifies some of the studies that have been completed in the area of joint inventory/maintenance policies. If a component's time to failure is known confidently, the optimal timing for the maintenance event can be determined to minimize maintenance costs. Section three of this chapter describes how the optimal replacement time is determined. Once the optimal replacement time is determined, a maintenance schedule can be developed for the system of n parts. The maintenance schedule for each part is described in section four of this chapter.

3.2 Joint Inventory/Maintenance Policies

Attempts have been made to link inventory and maintenance policies but additional research is required. Most existing studies are limited and are not suitable for a dynamic environment. Some limitations found in existing studies include assumptions that the failure distribution is known and constant, spares or machines are limited to one, or all operating machines are maintained at the same time.

One of the first papers in the combined study of inventory and maintenance policies was presented by Falkner in 1968 [48]. In this model, a single component has an increasing failure rate and it is considered more economical to replace a component prior to failure than to allow the component to fail. A dynamic programming problem is derived and solved to minimize the expected machine operating cost over a finite horizon. Given a known lifetime distribution function, an optimal inventory level and reorder level is calculated to minimize costs.

Thomas and Osaki [49] evaluated a system comprised of a maintained component with one allowable spare unit. The expected cost per unit time is derived using ordering (normal and emergency), shortage, and holding costs. An increasing failure rate is used to determine an expected cycle time for preventative replacement. Results show that the optimal policy is either to replace the unit as soon as the ordered spare is delivered or not to replace until after failure. The results depend on given conditions on shortage costs, ordering costs, and lifetime distribution. Another study by Thomas and Osaki [50], sought an optimal ordering policy, or time to order t_0, for a one-machine system with one available spare. Cases are presented where, given varying failure rates, it is optimal to order just prior, just after, or some time after a part is placed in service. This research is limited to systems with one allowable spare and known lifetime distributions. Osaki, Dohi, and Kaio have completed a long line of papers dealing with the stocking policies for a one unit system [51-58].

[59] is the first of two articles by Armstrong and Atkins discussing the joint optimization of maintenance and inventory policies. Their initial model consists of a single component subject to random failure with an allowance of one spare unit. Using a

constant lead-time, maintenance costs for replacement and breakage occur along with inventory costs for holding and shortage. The objective function, or Joint Cost Function (JCF), is derived giving the expected operating cost per unit time. The JCF is composed of the expected cost per cycle in the numerator and the expected cycle length in the denominator. The expected cost per cycle is developed through the addition of the expected replacement and breakage costs per cycle, along with the expected shortage and holding costs per cycle. The JCF is minimized to produce the optimal ordering and maintenance policies. Characteristics of the JCF are given under certain conditions. Results show that the joint optimization gives an average improvement of 3% over a sequentially optimized system. Armstrong and Atkins strongly recommend joint optimization when the inventory costs dominate the maintenance costs and when the lead- time is large. Additionally, they found sequential optimization in some cases can yield good results and recommend that inventory managers be aware of maintenance policies. Armstrong and Atkins [60] offered an extension to include replacing the fixed replacement cost with a cost function and set up the problem with a service level constraint. This work also incorporated separate lead-times for scheduled versus unscheduled orders as well as random lead-times.

Sarker and Haque [61] used simulation to show jointly optimized polices produce better results than separately or sequentially optimized policies. Cost savings of 2.81 – 8.77% are shown for case data. A jointly optimal approach lowered ordering cost, holding cost, and failure replacement costs. Sarker and Haque commented on the research conducted in optimal inventory/maintenance stating, “relatively little effort has been exerted to their (maintenance and inventory) joint optimization…” Their simulation model uses the following assumptions: operating units are statistically identical (failure is revealed instantly), spares do not deteriorate, replacement time is stochastic, increasing failure rate (non-exponential), unit cost is constant and is ignored, emergency orders are placed when stocking level is at or below zero, maintenance policy is of the block replacement type, and inventory is of the continuous review type.

A significant amount of papers have examined maintenance and provisioning policies under a block replacement policy [62, 63]. Chelbi and Dound proposed a computational procedure to determine the optimal replacement period T and optimal inventory threshold s for one unit or a set of identical units under a block replacement policy. The optimal policies were determined by minimizing the total average cost per unit time over an infinite horizon. Also presented are expressions for inventory costs including: average total holding cost, average total shortage cost, and total inventory management cost per unit time [64]. In a block replacement policy, a failed component is replaced at the time of failure and all n operating components are replaced simultaneously at some predetermined time interval. This type of policy is useful for applications where all operating mechanisms can be offline during the same period to perform maintenance. However, it may not be feasible to have all mechanisms offline at the same time. This policy also assumes that the machinery began operating at the same time and continue to follow the same operational schedule.

Kabir and Al-Olayan [65] states joint ordering and maintenance policies commonly are based on a single machine system and a maximum spare allowance of 1. A simulation study of ordering and maintenance policies for multiple machines under a (s, S) ordering policy is conducted and results indicate that the expected cost under separate policies are higher than if policies are jointly derived. Cases show a percent savings between 0 and 21 percent, where the amount of savings depended on the values of inventory and replacement costs and the failure distribution parameters. Using the results, regression analysis determined that holding and shortage inventory costs have the greatest influence on the optimal policies. Additionally, the preventive and failure replacement costs, as well as the failure distribution shape have considerable influence on the stocking policy.

[66] further explored the Barlow and Proschan age-based preventative replacement policy. Optimal values of the decision variables (t_1, s, S) are sought through minimization of the expected total cost per period, where t_1 is the time of preventative replacement. This research compared results between a jointly optimal (t_1, s, S) policy and a Barlow-Proschan age replacement policy supported by an optimal (s, S) inventory policy. The system studied allows for multiple spares and assumes a known lifetime distribution. This is the most relevant work to the system described in this study. The policies presented in this study are unique in that they are developed for a multiple component system where the parts are not maintained by block replacement; rather, each part is maintained separately. The policies presented in this paper will be used in Chapter 6 for a comparison case study.

3.3 Optimal Replacement Policy

Let C_f equal the cost incurred when a part is replaced due to failure, and C_p be the cost when a part is replaced due to preventive maintenance. If $C_p < C_f$, the general maintenance policy is to replace the part at failure or at an optimal replacement age t^*, whichever occurs first. The optimal replacement age is solved by minimizing the expected maintenance cost per unit of time as defined by Barlow and Hunter [67]. The expected maintenance cost per unit of time is found by dividing the expected maintenance cost per cycle by the expected length of the cycle. Let the expected maintenance cost per cycle equal

$$E\left[C\right] = C_f \int_0^{t^*} f(t)dt + C_p \int_{t^*}^{\infty} f(t)dt = C_f F(t) + C_p R(t) \tag{3.1}$$

and the expected cycle length equal

$$E\left[T\right] = \int_0^{t^*} t \cdot f(t)dt + t^* \int_{t^*}^{\infty} f(t)dt = \int_0^{t^*} R(t)dt. \tag{3.2}$$

The expected maintenance cost per unit of time is then

$$Z(t) = \frac{E\left[C\right]}{E\left[T\right]} = \frac{C_f F(t) + C_p R(t)}{\int_0^t R(t')dt'}. \tag{3.3}$$

The optimal replacement age may also be solved by setting the ratio $\dfrac{C_f}{C_f - C_p}$ equal to the equation:

$$L(t) = h(t) \int_0^{t^*} R(t)dt + R(t). \tag{3.4}$$

For a Weibull distribution, this expression is then

$$L(t) = \frac{\beta}{\eta}\left(\frac{t^*}{\eta}\right)^{\beta-1} \int_0^{t^*} \exp\left[-\left(\frac{t}{\eta}\right)^{\beta}\right]dt + \exp\left[-\left(\frac{t}{\eta}\right)^{\beta}\right]. \tag{3.5}$$

A closed-form solution to the integral of *R(t)* is not available and it is suggested that the Simpson's Rule be used to evaluate this expression [68].

3.4 System Maintenance Policy

This study will implement a practical maintenance policy using the guidance of the optimal replacement policy described in the previous section. In most applications, it is impractical to guarantee that a part will be replaced at the exact optimal time. The following reasons are offered to justify this statement: 1) Maintenance personnel may not be available at the exact replacement time, 2) The component may be in an operational state and not available for replacement, and 3) Replacement or repair may need to be delayed due to a spare not being available.

Two maintenance policies were introduced in [69] and will be used to allow for flexible maintenance scheduling. In first policy, called the Normal maintenance strategy, the replacement time is a random variable following a Normal Distribution with a mean equal to the optimal replacement time. The variance of the distribution can be adjusted to ensure the probability of replacements occur in a desired range of time. Additionally, the variance of this distribution should be adjusted so that the distribution is an adequate representation of actual replacement times. The second policy will be called the Uniform maintenance strategy and it utilizes a Uniform distribution to model the maintenance performance. In this strategy, the optimal replacement time will be again be used to establish the time range for maintenance completion.

For the Normal maintenance strategy, a maintenance planner is given two decision variables to develop the maintenance schedule. These variables are PC, or probability of completion, and CR, completion range. CR describes the time range that a replacement should be completed, given as a percentage of the mean or optimal completion time. For example, if the mean is 3, a $CR = .10$ yields a completion range of

$$\mu - \mu \cdot CR \le t^* \le \mu + \mu \cdot CR, \tag{3.6}$$

or $2.7 \le t^* \le 3.3$. *PC* describes the desired or actual probability of replacement during the completion range and is defined using the expression

$$PC = P\left\{\mu - \mu \cdot CR \le t^* \le \mu + \mu \cdot CR\right\}. \tag{3.7}$$

The Completion Range (*CR*) can be used to set a up a confidence limit for the Normal maintenance strategy and using this limit the strategy variance can be determined. For a Normal Distribution, the two-sided 95% confidence interval can be represented as: $\mu \pm \Phi^{-1}(.975) \cdot \sigma$, where Φ^{-1} is the inverse standardized normal function. If we let $PC = .95$ the value $\mu \cdot CR$ is then equal to $\Phi^{-1}(.975) \cdot \sigma$ and $\sigma = \dfrac{\mu \cdot CR}{\Phi^{-1}(.975)}$.

The optimal replacement time is at three years. A maintenance policy is established to ensure there is a 95% probability that the replacement will occur within $\pm 10\%$ of the optimal replacement time. This maintenance policy can be described by a Normal Distribution with $\mu = 3$ and $\sigma = \dfrac{\mu \cdot CR}{\Phi^{-1}(.975)} = \dfrac{3 \cdot .1}{1.96} = .153$.

For the Uniform maintenance strategy, a variable *UR* or Uniform Range will be employed to define the Uniform distribution parameters as $t^* - t^* \cdot UR$ and $t^* + t^* \cdot UR$ respectively. Using the same optimal replacement time from the previous example and $UR = .10$, this strategy would be modeled with a random variable uniformally distributed with parameters 2.9 and 3.1.

CHAPTER 4

REPLACEMENT AND DEMAND DISTRIBUTION

4.1 Calculating the Replacement Probabilities

As first described in Chapter 3, operating parts in the system can either fail or be removed through maintenance. These two stochastic events govern the demand placed on inventory. Only one of these events can occur on each part, for if one of the events transpires, the part is removed from service. In probability theory these events are said to be mutually exclusive. Let the removal due to a maintenance event be designated as *M* and a removal due to failure be *F*, then $P\left(M \cap F\right) = 0$ and the solution to $M \cup F$ is the third axiom of probability: $P(M \cup F) = P\left(M\right) + P\left(F\right)$. Therefore, the replacement probability of each part is determined by summing the probability of being maintained and the probability of failing.

If part P_i has been in operation for a length of time t_i, we can determine the probability of maintenance or failure during the next period Δ through conditioning. The conditional probability of event A occurring given event B has occurred, as long as $P\left(B\right) > 0$, is $P\left(A \middle| B\right) = \dfrac{P\left(AB\right)}{P\left(B\right)}$, where $P\left(AB\right)$ is the probability of A and B occurring simultaneously. Let A indicate the replacement of part *i* due to maintenance or failure during period Δt and B denote that part *i* has not been maintained or failed up to time t_i, the probabilities of maintenance and failure of part *i* during the next period Δt are,

$$P\left(M_i\right)=P_i(t_i<M_i\le t_i+\Delta t\mid M_i>t_i)=\frac{P_i(t_i<M_i\le t_i+\Delta t)}{P_i(M_i>t_i)}=\int_{t_i}^{t_i+\Delta t}\frac{f_M\left(t\right)dt}{1-F_M\left(t\right)}=$$
$$\int_{t_i}^{t_i+\Delta t}\frac{f_M\left(t\right)dt}{R_M\left(t\right)}=\int_{t_i}^{t_i+\Delta t}\lambda_M\left(t\right)dt \quad (4.1)$$

and

$$P\left(F_i\right)=P_i(t_i<F_i\le t_i+\Delta t\mid F_i>t_i)=\frac{P_i(t_i<F_i\le t_i+\Delta t)}{P_i(F_i>t_i)}=\int_{t_i}^{t_i+\Delta t}\frac{f_F\left(t\right)dt}{1-F_F\left(t\right)}=$$
$$\int_{t_i}^{t_i+\Delta t}\frac{f_F\left(t\right)dt}{R_F\left(t\right)}=\int_{t_i}^{t_i+\Delta t}\lambda_F\left(t\right)dt, \quad (4.2)$$

where $R\left(t\right)$ and $\lambda\left(t\right)$ are the reliability and failure rate of maintenance and failure. Recalling the third axiom of probability, the replacement probability of part *i* is,

$$P\left(M_i\right)+P\left(F_i\right)=\int_{t_i}^{t_i+\Delta t}\lambda_M\left(t\right)dt+\int_{t_i}^{t_i+\Delta t}\lambda_F\left(t\right)dt. \quad (4.3)$$

In order to solve for the *k*-Period non-stationary inventory policy as discussed in Chapter 2.4, the demand distributions for *k* future periods are required. The *k*-period demand distributions will be constructed using the replacement probabilities of each individual part as described above. The replacement probabilities can be solved for *k* periods using the expression,

$$P\left(M_{i,k+j}\right)+P\left(F_{i,k+j}\right)=\int_{t_i+j\cdot\Delta t}^{t_i+(j+1)\cdot\Delta t}\lambda_M\left(t\right)dt+\int_{t_i+j\cdot\Delta t}^{t_i+(j+1)\cdot\Delta t}\lambda_F\left(t\right)dt, \quad (4.4)$$

where k is the current period and $j=0,1,\ldots n$. When applying this expression the following rules are used:

1) If $P\left(M_{i,k+j}\right)+P\left(F_{i,k+j}\right)=1$ for any *j*, set $t_{i,k+j+1}=0,$ and
2) Let $R_{i,k+j}$ represent any replacement or $P\left(R_{i,k+j}\right)=P\left(M_{i,k+j}\right)+P\left(F_{i,k+j}\right),$ if $P\left(R_{i,k+j}\right)>1$ set $P\left(R_{i,k+j}\right)=1.$

Rule 1 indicates a predicted replacement during period *k+j* and subsequent part renewal at the beginning of period *k+j+1*. Rule 2 is needed to ensure that the second axiom of probability is maintained for $P\left(R_{i,k+j}\right)$.

4.2 Constructing the Demand Distribution: Gray's Code

Once $P_{i,k+j}$ has been determined for all *i* and *j*, the period demand can be determined. The period demand will be represented as a discrete probability distribution, since the inventory can only be depleted by discrete units. To construct a discrete demand distribution, the probability of the random variable *X*, where *X* represents a discrete part demand, is calculated for each possible *X*. Determining the probability of demand in this case is not a trivial matter, remember, $P_{i,k+j} \neq P_{i+1,k+j}$ for $i = 1 \ldots n-1$. For each *X*, all the combinations of part replacements that yield a demand of *X* must be considered, a collection of $\frac{n!}{X!(n-X)!}$ combinations; making it necessary to construct a total of 2^n unique combinations. For each possible x, where $x = 0,1,2,\ldots,n,$ the products of each combination are summed to determine the value for the probability of demand $\varphi_{k+j}\left(x\right)$. The general expression to determine $\varphi_{k+j}\left(x\right)$ is

$$P(X = x) = \sum_{m=1}^{q}\left[\left(P_1\right)^{\delta_{1,m}} \ldots \left(P_n\right)^{\delta_{n,m}} \left(1-P_1\right)^{1-\delta_{1,m}} \ldots \left(1-P_n\right)^{1-\delta_{n,m}}\right], \quad (4.5)$$

where $q = \frac{n!}{x!(n-x)!}$, $\delta_{i,m} = \begin{cases} 1 & \textit{if part i fails in combination m} \\ 0 & \textit{otherwise} \end{cases}$, and $\sum_{i=1}^{n} \delta_{i,m} = x \quad \forall m$.

The following example is provided to further explain.

For $n = 4$, the combinations of all demand possibilities are shown below where $P_{i,k}$ is the probability of replacement and $\left(1-P_{i,k}\right)$ is the probability of survival of part i during period k:

$$P_k(X=0)=\sum_{m=1}^{1}\left[\left(P_{1,k}\right)^{\delta_{1,1}}\ldots\left(P_{4,k}\right)^{\delta_{4,1}}\left(1-P_{1,k}\right)^{1-\delta_{1,1}}\ldots\left(1-P_{4,k}\right)^{1-\delta_{4,1}}\right]=$$
$$\left(P_{1,k}\right)^{0}\ldots\left(P_{4,k}\right)^{0}\left(1-P_{1,k}\right)^{1-0}\ldots\left(1-P_{4,k}\right)^{1-0}=\left(1-P_1\right)^1\left(1-P_2\right)^1\left(1-P_3\right)^1\left(1-P_4\right)^1$$

$$P_k(X=1)=\sum_{m=1}^{4}\left[\left(P_{1,k}\right)^{\delta_{1,m}}\ldots\left(P_{4,k}\right)^{\delta_{4,m}}\left(1-P_{1,k}\right)^{1-\delta_{1,m}}\ldots\left(1-P_{4,k}\right)^{1-\delta_{4,m}}\right]=$$
$$=\left(P_{1,k}\right)^{\delta_{1,1}}\left(P_{2,k}\right)^{\delta_{2,1}}\left(P_{3,k}\right)^{\delta_{3,1}}\left(P_{4,k}\right)^{\delta_{4,1}}\left(1-P_{1,k}\right)^{1-\delta_{1,1}}\left(1-P_{2,k}\right)^{1-\delta_{2,1}}\left(1-P_{3,k}\right)^{1-\delta_{3,1}}\left(1-P_{4,k}\right)^{1-\delta_{4,1}}+\ldots$$
$$\left(P_{1,k}\right)^{\delta_{1,4}}\left(P_{2,k}\right)^{\delta_{2,4}}\left(P_{3,k}\right)^{\delta_{3,4}}\left(P_{4,k}\right)^{\delta_{4,4}}\left(1-P_{1,k}\right)^{1-\delta_{1,4}}\left(1-P_{2,k}\right)^{1-\delta_{2,4}}\left(1-P_{3,k}\right)^{1-\delta_{3,4}}\left(1-P_{4,k}\right)^{1-\delta_{4,4}}$$
$$=\left(P_{1,k}\right)^{1}\left(P_{2,k}\right)^{0}\left(P_{3,k}\right)^{0}\left(P_{4,k}\right)^{0}\left(1-P_{1,k}\right)^{1-1}\left(1-P_{2,k}\right)^{1-0}\left(1-P_{3,k}\right)^{1-0}\left(1-P_{4,k}\right)^{1-0}+$$
$$\left(P_{1,k}\right)^{0}\left(P_{2,k}\right)^{1}\left(P_{3,k}\right)^{0}\left(P_{4,k}\right)^{0}\left(1-P_{1,k}\right)^{1-0}\left(1-P_{2,k}\right)^{1-1}\left(1-P_{3,k}\right)^{1-0}\left(1-P_{4,k}\right)^{1-0}+$$
$$\left(P_{1,k}\right)^{0}\left(P_{2,k}\right)^{0}\left(P_{3,k}\right)^{1}\left(P_{4,k}\right)^{0}\left(1-P_{1,k}\right)^{1-0}\left(1-P_{2,k}\right)^{1-0}\left(1-P_{3,k}\right)^{1-1}\left(1-P_{4,k}\right)^{1-0}+$$
$$\left(P_{1,k}\right)^{0}\left(P_{2,k}\right)^{0}\left(P_{3,k}\right)^{0}\left(P_{4,k}\right)^{1}\left(1-P_{1,k}\right)^{1-0}\left(1-P_{2,k}\right)^{1-0}\left(1-P_{3k}\right)^{1-0}\left(1-P_{4,k}\right)^{1-1}$$

$$P_k(X=2)=\sum_{m=1}^{6}\left[\left(P_{1,k}\right)^{\delta_{1,m}}\ldots\left(P_{4,k}\right)^{\delta_{4,m}}\left(1-P_{1,k}\right)^{1-\delta_{1,m}}\ldots\left(1-P_{4,k}\right)^{1-\delta_{4,m}}\right]=$$
$$=\left(P_{1,k}\right)^{\delta_{1,1}}\left(P_{2,k}\right)^{\delta_{2,1}}\left(P_{3,k}\right)^{\delta_{3,1}}\left(P_{4,k}\right)^{\delta_{4,1}}\left(1-P_{1,k}\right)^{1-\delta_{1,1}}\left(1-P_{2,k}\right)^{1-\delta_{2,1}}\left(1-P_{3k}\right)^{1-\delta_{3,1}}\left(1-P_{4,k}\right)^{1-\delta_{4,1}}+\ldots$$
$$\left(P_{1,k}\right)^{\delta_{1,6}}\left(P_{2,k}\right)^{\delta_{2,6}}\left(P_{3,k}\right)^{\delta_{3,6}}\left(P_{4,k}\right)^{\delta_{4,6}}\left(1-P_{1,k}\right)^{1-\delta_{1,6}}\left(1-P_{2,k}\right)^{1-\delta_{2,6}}\left(1-P_{3,k}\right)^{1-\delta_{3,6}}\left(1-P_{4,k}\right)^{1-\delta_{4,6}}$$
$$=\left(P_{1,k}\right)^{1}\left(P_{2,k}\right)^{1}\left(P_{3,k}\right)^{0}\left(P_{4,k}\right)^{0}\left(1-P_{1,k}\right)^{1-1}\left(1-P_{2,k}\right)^{1-1}\left(1-P_{3,k}\right)^{1-0}\left(1-P_{4,k}\right)^{1-0}+$$
$$\left(P_{1,k}\right)^{0}\left(P_{2,k}\right)^{1}\left(P_{3,k}\right)^{1}\left(P_{4,k}\right)^{0}\left(1-P_{1,k}\right)^{1-0}\left(1-P_{2,k}\right)^{1-1}\left(1-P_{3,k}\right)^{1-1}\left(1-P_{4,k}\right)^{1-0}+$$
$$\left(P_{1,k}\right)^{0}\left(P_{2,k}\right)^{0}\left(P_{3,k}\right)^{1}\left(P_{4,k}\right)^{1}\left(1-P_{1,k}\right)^{1-0}\left(1-P_{2,k}\right)^{1-0}\left(1-P_{3,k}\right)^{1-1}\left(1-P_{4,k}\right)^{1-1}+$$
$$\left(P_{1,k}\right)^{1}\left(P_{2,k}\right)^{0}\left(P_{3,k}\right)^{0}\left(P_{4,k}\right)^{1}\left(1-P_{1,k}\right)^{1-1}\left(1-P_{2,k}\right)^{1-0}\left(1-P_{3,k}\right)^{1-0}\left(1-P_{4,k}\right)^{1-1}+$$
$$\left(P_{1,k}\right)^{1}\left(P_{2,k}\right)^{0}\left(P_{3,k}\right)^{1}\left(P_{4,k}\right)^{0}\left(1-P_{1,k}\right)^{1-1}\left(1-P_{2,k}\right)^{1-0}\left(1-P_{3,k}\right)^{1-1}\left(1-P_{4,k}\right)^{1-0}+$$
$$\left(P_{1,k}\right)^{0}\left(P_{2,k}\right)^{1}\left(P_{3,k}\right)^{0}\left(P_{4,k}\right)^{1}\left(1-P_{1,k}\right)^{1-0}\left(1-P_{2,k}\right)^{1-1}\left(1-P_{3,k}\right)^{1-0}\left(1-P_{4,k}\right)^{1-1}+$$

$$P_k(X=3)=\sum_{m=1}^{4}\left[\left(P_{1,k}\right)^{\delta_{1,m}}\ldots\left(P_{4,k}\right)^{\delta_{4,m}}\left(1-P_{1,k}\right)^{1-\delta_{1,m}}\ldots\left(1-P_{4,k}\right)^{1-\delta_{4,m}}\right]=$$

$$=\left(P_{1,k}\right)^{\delta_{1,1}}\left(P_{2,k}\right)^{\delta_{2,1}}\left(P_{3,k}\right)^{\delta_{3,1}}\left(P_{4,k}\right)^{\delta_{4,1}}\left(1-P_{1,k}\right)^{1-\delta_{1,1}}\left(1-P_{2,k}\right)^{1-\delta_{2,1}}\left(1-P_{3,k}\right)^{1-\delta_{3,1}}\left(1-P_{4,k}\right)^{1-\delta_{4,1}}+\ldots$$

$$\left(P_{1,k}\right)^{\delta_{1,4}}\left(P_{2,k}\right)^{\delta_{2,4}}\left(P_{3,k}\right)^{\delta_{3,4}}\left(P_{4,k}\right)^{\delta_{4,4}}\left(1-P_{1,k}\right)^{1-\delta_{1,4}}\left(1-P_{2,k}\right)^{1-\delta_{2,4}}\left(1-P_{3,k}\right)^{1-\delta_{3,4}}\left(1-P_{4,k}\right)^{1-\delta_{4,4}}$$

$$=\left(P_{1,k}\right)^{1}\left(P_{2,k}\right)^{1}\left(P_{3,k}\right)^{1}\left(P_{4,k}\right)^{0}\left(1-P_{1,k}\right)^{1-1}\left(1-P_{2,k}\right)^{1-1}\left(1-P_{3,k}\right)^{1-1}\left(1-P_{4,k}\right)^{1-0}+$$

$$\left(P_{1,k}\right)^{0}\left(P_{2,k}\right)^{1}\left(P_{3,k}\right)^{1}\left(P_{4,k}\right)^{1}\left(1-P_{1,k}\right)^{1-0}\left(1-P_{2,k}\right)^{1-1}\left(1-P_{3,k}\right)^{1-1}\left(1-P_{4,k}\right)^{1-1}+$$

$$\left(P_{1,k}\right)^{1}\left(P_{2,k}\right)^{0}\left(P_{3,k}\right)^{1}\left(P_{4,k}\right)^{1}\left(1-P_{1,k}\right)^{1-1}\left(1-P_{2,k}\right)^{1-0}\left(1-P_{3,k}\right)^{1-1}\left(1-P_{4,k}\right)^{1-1}+$$

$$\left(P_{1,k}\right)^{1}\left(P_{2,k}\right)^{1}\left(P_{3,k}\right)^{0}\left(P_{4,k}\right)^{1}\left(1-P_{1,k}\right)^{1-1}\left(1-P_{2,k}\right)^{1-1}\left(1-P_{3,k}\right)^{1-0}\left(1-P_{4,k}\right)^{1-1}$$

$$P_k(X=4)=\sum_{m=1}^{1}\left[\left(P_{1,k}\right)^{\delta_{1,1}}\ldots\left(P_{4,k}\right)^{\delta_{4,1}}\left(1-P_{1,k}\right)^{1-\delta_{1,1}}\ldots\left(1-P_{4,k}\right)^{1-\delta_{4,1}}\right]=$$

$$=\left(P_{1,k}\right)^{1}\left(P_{2,k}\right)^{1}\left(P_{3,k}\right)^{1}\left(P_{4,k}\right)^{1}$$

Note: A total of $2^4=16$ unique combinations are listed above.

With an exponential increase in possibilities, a combinatorial technique known as Gray code of order n, or Gray's code, is necessary to ensure all combinations are identified as *n* becomes large (the possible combinations exceed 1 million for $n=20$). This code makes use of a *n-tuple* of 0's and 1's along with a systematic algorithm to produce all unique combinations. For this case, the digit 1 will represent the probability of a part replacement and 0 will represent the probability of a part survival during the next period. For each combination, the sum of the *n-tuple* will identify the demand amount produced by that combination.

Gray's code was first demonstrated in 1878 by a French engineer Emile Baudot and was patented by the Bell Labs researcher Frank Gray in 1953 [70]. The algorithm to generate Gray's code is:

Begin with *n-tuple* $a_1a_2\cdots a_n = 00\cdots0$. While $a_1a_2\cdots a_n \neq 10\cdots0$, follow the following steps:

1) Add the series of digits. If the sum is even, change a_n.
2) Else, find j where $a_j = 1$ and $a_i = 0$ for all i where $i = n, n-1, \cdots, j+1$ and change a_{j-1}. Note: if $j = n$, change a_{n-1}.

The following example will illustrate Gray's code and the use of the above algorithm for $n = 4$.

a_1	a_2	a_3	a_4	
0	0	0	0	(sum is even change a_n)
0	0	0	1	(see note)
0	0	1	1	(sum is even change a_n)
0	0	1	0	($a_3 = 1$ and $a_4 = 0$, change a_2)
0	1	1	0	(sum is even change a_n)
0	1	1	1	(see note)
0	1	0	1	(sum is even change a_4)
0	1	0	0	($a_2 = 1$, $a_3 = 0$, and $a_4 = 0$ change a_1)
1	1	0	0	(sum is even change a_n)
1	1	0	1	(see note)
1	1	1	1	(sum is even change a_n)
1	1	1	0	(see note)
1	0	1	0	(sum is even change a_n)
1	0	1	1	(see note)
1	0	0	1	(sum is even change a_n)
1	0	0	0	(end).

Assigning the values $P_{1,k} = .1$, $P_{2,k} = .05$, $P_{3,k} = .15$, $P_{4,k} = .75$ and using the series produced by Gray's code, the values for the discrete demand distribution are determined as:

$P_{1,k}$	$P_{2,k}$	$P_{3,k}$	$P_{4,k}$	
0	0	0	0	$=(.9)(.95)(.85)(.25)=.1817$
0	0	0	1	$=(.9)(.95)(.85).75=.5451$
0	0	1	1	$=(.9)(.95).15\cdot.75=.0962$
0	0	1	0	$=(.9)(.95).15(.25)=.0321$
0	1	1	0	$=(.9).05\cdot.15(.25)=.0017$
0	1	1	1	$=(.9).05\cdot.15\cdot.75=.0051$
0	1	0	1	$=(.9).05(.85).75=.0287$
0	1	0	0	$=(.9).05(.85)(.25)=.0096$
1	1	0	0	$=.1\cdot.05(.85)(.25)=.0011$
1	1	0	1	$=.1\cdot.05(.85).75=.0032$
1	1	1	1	$=.1\cdot.05\cdot.15\cdot.75=.0006$
1	1	1	0	$=.1\cdot.05\cdot.15(.25)=.0002$
1	0	1	0	$=.1(.95).15(.25)=.0036$
1	0	1	1	$=.1(.95).15\cdot 75=.0107$
1	0	0	1	$=.1(.95)(.85).75=.0606$
1	0	0	0	$=.1(.95)(.85)(.25)=.0202$

$$\varphi_k(0)=.1817$$
$$\varphi_k(1)=.5451+.0321+.0096+.0202=.6070$$
$$\varphi_k(2)=.0962+.0017+.0287+.0011+.0036+.0606=.1919$$
$$\varphi_k(3)=.0051+.0032+.0002+.0107=.0192$$
$$\varphi_k(4)=.0006$$

The increase in computational effort to assemble the collection of 2^n combinations requires a short discussion on the upper bound of n in the application of this model. The (s, S) inventory policy is best suited for intermittent or low demand items and this type of demand supports a system where n is relatively small. Throughout the development of this model, a general system size of $n \le 30$ was envisioned. This "upper bound" is not a restricting number but is provided as a practical measure.

4.3 Replacement Probability and Demand Distribution Example

As shown in the previous section, the demand distribution from period to period is influenced by the replacement probabilities and ages of each part. If the replacement probabilities change from period to period, the demand distribution will also change. To further examine how the replacement probabilities and demand distribution change, let us examine a series of consecutive periods for a system of four parts. These four parts have an identical failure rate following a Weibull distribution with shape parameter β of 3 and scale parameter η of 6. The expected time of the maintenance event is found by solving for the optimal replacement time as shown in Chapter 3.3. For this example, let $C_f = 2$ and $C_p = 1$. Using the equation

$$\frac{C_f}{C_f - C_p} = \frac{\beta}{\eta}\left(\frac{t^*}{\eta}\right)^{\beta-1} \int_0^{t^*} \exp\left[-\left(\frac{t}{\eta}\right)^{\beta}\right] dt + \exp\left[-\left(\frac{t}{\eta}\right)^{\beta}\right], \tag{4.6}$$

the optimal replacement time t^* is found to be 4.69. Letting the decision variable *PC* and *CR* equal .95 and .10, the variation of the maintenance policy is .24. If the age of the parts are $t_1 = .5, t_2 = 1.7, t_3 = 3.1,$ and $t_3 = 4.6,$ the probability of replacement for each part during the next period of length .25 is:

$$P_{1,k} = \int_{.5}^{.75} \frac{1}{\sqrt{2\pi}\sigma} \exp\left[-\frac{(t-\mu)^2}{2\sigma^2}\right]\left[1-\Phi\left(\frac{t-\mu}{\sigma}\right)\right]^{-1} dt + \int_{.5}^{.75} \frac{\beta}{\eta}\left(\frac{t}{\eta}\right)^{\beta-1} dt = .0014,$$

$$P_{2,k} = \int_{1.7}^{1.95} \frac{1}{\sqrt{2\pi}\sigma} \exp\left[-\frac{(t-\mu)^2}{2\sigma^2}\right]\left[1-\Phi\left(\frac{t-\mu}{\sigma}\right)\right]^{-1} dt + \int_{1.7}^{1.95} \frac{\beta}{\eta}\left(\frac{t}{\eta}\right)^{\beta-1} dt = .0116,$$

$$P_{3,k} = \int_{3.1}^{3.35} \frac{1}{\sqrt{2\pi}\sigma} \exp\left[-\frac{(t-\mu)^2}{2\sigma^2}\right]\left[1-\Phi\left(\frac{t-\mu}{\sigma}\right)\right]^{-1} dt + \int_{3.1}^{3.35} \frac{\beta}{\eta}\left(\frac{t}{\eta}\right)^{\beta-1} dt = .0361,$$

and

$$P_{4,k} = \int_{4.6}^{4.85} \frac{1}{\sqrt{2\pi}\sigma} \exp\left[-\frac{(t-\mu)^2}{2\sigma^2} \right] \left[1 - \Phi\left(\frac{t-\mu}{\sigma} \right) \right]^{-1} dt + \int_{4.6}^{4.85} \frac{\beta}{\eta} \left(\frac{t}{\eta} \right)^{\beta-1} dt = 1.0.$$

Using the probabilities, the demand distribution for the next period is determined to be $\varphi_k(0) = 0$, $\varphi_k(1) = .9514$, $\varphi_k(2) = .0481$, $\varphi_k(3) = .0005$, and $\varphi_k(4) = 0$. Using this same process, the probability of replacement of each part and demand distribution were determined for next four periods and the results are shown in Tables 4 and 5 respectively. Note: The age of Part 4 was initially close to the optimal replacement time and $P_{4,k} = 1.0$. Using the rules given in Chapter 4.1, the part age at the beginning of the next period is 0.

Table 3 Probability of Replacement

	j=0	*j=1*	*j=2*	*j=3*	*j=4*
$P_{1,k+j}$	0.0014	0.0027	0.0044	0.0066	0.0092
$P_{2,k+j}$	0.0116	0.015	0.0188	0.023	0.0277
$P_{3,k+j}$	0.0361	0.042	0.0484	0.0617	0.1365
$P_{4,k+j}$	1	0.0001	0.0005	0.0014	0.0027

Table 4 Demand Distribution

	j=0	*j=1*	*j=2*	*j=3*	*j=4*
$\varphi_{k+j}(0)$	0	0.9411	0.9291	0.9094	0.8296
$\varphi_{k+j}(1)$	0.9514	0.0581	0.0697	0.0885	0.1647
$\varphi_{k+j}(2)$	0.0481	0.0008	0.0012	0.0021	0.0056
$\varphi_{k+j}(3)$	0.0005	0	0	0	0
$\varphi_{k+j}(4)$	0	0	0	0	0

4.4 Stationarity Analysis

Once the demand distribution has been estimated, the inventory policy can be determined using either a stationary or non-stationary solution. A stationary demand is one that follows the same demand distribution from period to period. A non-stationary demand has a varying demand distribution from period to period. To distinguish between a stationary and non-stationary demand, the Kolmogorov-Smirnov or KS test can be used to test the distribution from one period to the next. The hypothesis of the KS test is that the distributions are the same and the test statistic is $KS = \max\left|D_{k+j} - D_{k+j+1}\right|$, where D_{k+j} is the demand distribution cdf for period $k+j$. The rejection criteria for a two-sided test is $KS < \chi^2_{k,p}$. A KS test was performed using the distributions from the previous section and the KS statistic, p-value, and result are displayed in Table 5.

Table 5 Stationarity Test

	KS Stat	p-value	Result
Period k to $k+1$	0.9411	0.02	Hypothesis rejected
Period $k+1$ to $k+2$	0.012	1	Could not reject
Period $k+2$ to $k+3$	0.0197	1	Could not reject

In this scenario, the demand distribution is not stationary and the non-stationary solution method from Chapter 2.4 should be used to determine the inventory policy.

CHAPTER 5

PARAMETRIC ESTIMATION

5.1 Censoring

Part lifetime data is typically in the form of a complete or censored data set. A complete data set is obtained if all items are allowed to operate until failure. However, it is often impractical in reliability testing to allow all items to operate until failure. Censoring occurs when an item is removed from service for any reason prior to failure. The most common forms of censorship are right, left, and interval censoring, with right being the most common. Right censored data describes a data set where the initial time placed in operation is known for all items but the exact failure times of one or more items are unknown due to removal from service. Left censoring is less common then right and occurs when failure times are known for all items but the exact times that one or more items were placed into service are unknown. Lastly, interval censoring occurs if failure times are not exactly known but grouped into intervals.

Right censoring is further divided into three sub-categories: Type I or time censoring, Type II or order statistic censoring, and random censoring. In Type I censoring, one or more items are removed from service at a specified time. Type II censoring occurs when one or more items are removed from service after a specified number of failures have occurred. Random censoring occurs when one or more items are removed from service before failure at independently random times. With random censoring, there are two or more probabilistic distributions in play and this type of censoring is also referred to as a competing risks model. These competing risks are the two or more stochastic events that could end the service life of each item. In this study, the competing risks are failure and random removal from service prior to failure. In the system being modeled, there are two stochastic events that will cause the part to be removed from operation. Replacement due to a failure may occur if a part stops adequately performing its desired function. The second competing risk is replacement at some stochastic interval due to a maintenance policy.

The purpose of reliability testing is to gain knowledge of the governing failure distribution for a particular item. The data collected is used to determine the distribution type and parameters that best describes the expected life of that item. Once lifetime data is collected and the type of censorship is determined, the distribution type and parameters can be estimated. In this study, the Weibull distribution is assumed to best describe the lifetime of the item. The Weibull distribution is often used in reliability analysis due to its flexibility in modeling failure rates. It has been used to model the life of electronic components, semi-conductors, pumps, motors, ball bearings, fatigued materials, as well as various biological organisms. Using this assumption, several methods can be used to determine the estimated Weibull parameters.

5.2 Methods of Estimation

Reineke completed a dissertation in 1998 that examined estimation methods for lifetime data under random right censoring levels of 25%, 50%, and 75% [71]. This study tested seven parametric, nonparametric, and semi-parametric methods and concluded that the parametric Maximum Likelihood Estimator (MLE) is the best method given that the distribution type is correctly specified. Additionally, the Piecewise Exponential Estimator (PEXE) [72] and Földes, Rejtő, and Winter Estimator (FRWE) [73] methods are the best nonparametric estimators for all tested censoring levels. With the exception of the Klein, Lee, and Moeschberger (KLM) Partial Parametric Estimator [74], the semi-parametric methods performed poorly and were not recommended for use. The MLE, PEXE, FRWE, and Klein, Lee, and Moeschberger Partially Parametric (KLM) methods will be further tested, along with the commonly used Kaplan-Meier Estimator (KME) [75], and are described in the following sections.

Due to the nature of this study, it is important to know the best performing estimator given a high level of censoring, lower values of *n*, and a shape parameter that best models mechanical parts. Reineke states that the censoring level does have an effect on estimator performance; however, the study did not comment on how the part sample size *n* and the shape parameter affect the performance. The analysis described later in this chapter will add to the work completed by Reineke, specifically for highly-censored cases. The following sections outline several estimators that are used in this study.

5.2.1 Maximum Likelihood Estimation (MLE)

Given a censored set of data where t_i represents a failure of part *i* and c_i represents a censored time of part *i*, let $\delta_i = 1$ if $t_i \le c_i$ and $\delta_i = 0$ if $t_i > c_i$. The likelihood function is then

$$L(u,\theta) = \prod_{i=1}^{n} f\left(u_i,\theta\right)^{\delta_i} R\left(u_i,\theta\right)^{1-\delta_i}, \tag{5.1}$$

where θ is the vector of parameters to be estimated and u includes all failure and censored times. The log likelihood function is then

$$\ln L(u,\theta)=\sum_{i=1}^{n}\ln f\left(u_i,\theta\right)^{\delta_i}+\sum_{i=1}^{n}\ln R\left(u_i,\theta\right)^{1-\delta_i}. \tag{5.2}$$

Since the density function is the product of the hazard and reliability functions or $f(t)=h(t)R(t)$, the log likelihood function can be rewritten in the form

$$\ln L(u,\theta)=\sum_{i=1}^{n}\ln h\left(u_i,\theta\right)^{\delta_i}+\sum_{i=1}^{n}\ln R\left(u_i,\theta\right)^{\delta_i}+\sum_{i=1}^{n}\ln R\left(u_i,\theta\right)^{1-\delta_i} \tag{5.3}$$

and simplifies to

$$\ln L(u,\theta)=\sum_{i=1}^{n}\ln h\left(u_i,\theta\right)^{\delta_i}+\sum_{i=1}^{n}\ln R\left(u_i,\theta\right). \tag{5.4}$$

If set U contains all values of u where $\delta_i=1$ and using the property $H(t)=-\log R(t)$, the expression can again be rewritten as

$$\ln L(u,\theta)=\sum_{i\in U}\ln h\left(u_i,\theta\right)-\sum_{i=1}^{n}H\left(u_i,\theta\right). \tag{5.5}$$

In the case of the Weibull distribution, the log likelihood function is

$$\ln L\left(u,\theta\right)=\sum_{i\in U}\ln\left[\beta\lambda\left(\lambda u_i\right)^{\beta-1}\right]-\sum_{i=1}^{n}\left(\lambda u_i\right)^{\beta}, \tag{5.6}$$

where β and λ are the shape and scale parameters respectively. This expression can then be simplified to

$$\begin{aligned}\ln L\left(u,\theta\right)&=\sum_{i\in U}\left[\ln\beta+\ln\lambda+\left(\beta-1\right)\ln\lambda+(\beta-1)\ln u_i\right]-\lambda^{\beta}\sum_{i=1}^{n}u_i^{\beta}\\&=\sum_{i\in U}\left[\ln\beta+\beta\ln\lambda+(\beta-1)\ln u_i\right]-\lambda^{\beta}\sum_{i=1}^{n}u_i^{\beta}.\end{aligned} \tag{5.7}$$

If there is r observed failures or values in set U, the log likelihood function becomes

$$\ln L(u,\theta) = r \ln \beta + \beta r \ln \lambda + (\beta - 1)\sum_{i \in U} \ln u_i - \lambda^{\beta} \sum_{i=1}^{n} u_i^{\beta}. \tag{5.8}$$

To solve for the estimates of the shape and scale parameters, the partial derivatives are determined for each parameter and solved for zero using the following expressions:

$$\frac{\partial \ln L(u,\theta)}{\partial \lambda} = \frac{\beta r}{\lambda} - \beta \lambda^{\beta - 1} \sum_{i=1}^{n} u_i^{\beta} = 0 \tag{5.9}$$

and

$$\frac{\partial \ln L(u,\theta)}{\partial \beta} = \frac{r}{\beta} + r \ln \lambda + \sum_{i \in U} \ln u_i - \sum_{i=1}^{n} (\lambda u)_i^{\beta} \ln(\lambda u_i) = 0. \tag{5.10}$$

Unfortunately, there is no closed form solution to the simultaneous solution of these two equations. However, λ can be solved in terms of β using the expression

$$\lambda = \left(\frac{r}{\sum_{i=1}^{n} u_i^{\beta}} \right)^{\frac{1}{\beta}}. \tag{5.11}$$

Applying this expression for λ to the second partial shown above and simplifying yields

$$g(\beta) = \frac{r}{\beta} + \sum_{i \in U} \ln u_i - \frac{r \sum_{i=1}^{n} u_i^{\beta} \ln u_i}{\sum_{i=1}^{n} u_i^{\beta}} = 0. \tag{5.12}$$

To solve for β, the iterative Newton-Raphson procedure could be used. In this case, each subsequent β_{i+1} is

$$\beta_{i+1} = \beta_i - \frac{g(\beta_i)}{g'(\beta_i)}, \tag{5.13}$$

where

$$g'(\beta) = -\frac{r}{\beta^2} - \frac{r}{\left(\sum_{i=1}^{n} u_i^{\beta}\right)^2}\left[\left(\sum_{i=1}^{n} u_i^{\beta}\right)\left(\sum_{i=1}^{n} (\ln u_i)^2 u_i^{\beta}\right) - \left(\sum_{i=1}^{n} u_i^{\beta} \ln u_i\right)^2\right]. \qquad (5.14)$$

The procedure is terminated when $|\beta_{i+1} - \beta_i|$ is less than some small value ε. To determine the initial value for β_0, the following process recommended in Leemis [76] can be used: while the observed number of failures or $r \geq 2$, set

$$\beta_0 = \frac{\sum_{i \in U}\left(X_i - \bar{X}\right)\left(Y_i - \bar{Y}\right)}{\sum_{i \in U}\left(X_i - \bar{X}\right)^2}, \qquad (5.15)$$

where $X_i = \ln u_i$, $Y_i = \ln\left[\ln\left(\frac{n+1}{n(u_i)}\right)\right]$, $\bar{X}$ and $\bar{Y}$ are the sample means, and $n(u_i)$ is the number of items operating prior to time u_i. The Maximum Likelihood Estimator of the lifetime distribution is

$$\widehat{F}_{MLE}(x) = 1 - \exp\left[-(\lambda x)^{\beta}\right]. \qquad (5.16)$$

5.2.2 Kaplan-Meier Estimator (KME)

The KME, also known as the product-limit estimator, was introduced in 1958 and is one of the most significant contributions in reliability theory. This paper is ranked as one of the most cited papers [77] of all time, having been cited over 29,000 times [78]. When no censoring is present, the KME simply reduces to the empirical distribution function. The KME is defined as:

$$\widehat{F}_{KME}(x) = \begin{cases} 0 & 0 < u_1 \\ 1 - \prod_{j=1}^{m} \frac{n_j - r_j}{n_j} & u_1 \le x \le u_m \\ 1 & x > u_m \end{cases}, \tag{5.17}$$

where m is the total number of failures, $n_i = n - \sum_{j=0}^{i-1} s_j - \sum_{j=0}^{i-1} r_j$, and s_j and r_j are given the value of 1 if the *jth* time represents a censored and failed item respectively and zero otherwise.

5.2.3 Piecewise Exponential Estimator (PEXE)

Let u_i represent the ordered failure times, c_{ij} represent the ordered censoring times, and k_i be the total number of censored observations between failures.

$$0 < c_{1,1} < \cdots < c_{1,k_1} < t_1 < c_{2,1} < \cdots < c_{2,k_2} < t_2 < \cdots < t_{r-1} < c_{r,1} < \cdots < c_{r,k_r} < t_r$$
$$< c_{r+1,1} < \cdots < c_{r+1,k_{r+1}}.$$

For the interval between successive failures, a constant failure rate is estimated and this rate is used to fit an exponential estimator of the lifetime distribution on each interval. These fitted functions are then pieced together to form a piecewise function from 0 to the time of the r^{th} failure. Beyond the r^{th} failure, an exponential fit with a hazard rate equal to the previous interval is often used. The constant failure rate for each interval is:

$$z_i = \frac{1}{\sum_{j=1}^{k} \left(c_{i,j} - u_{i-1}\right) + \left(n - i + 1 - \sum_{j=1}^{i} k_j\right) \cdot \left(u_i - u_{i-1}\right)} \tag{5.18}$$

and the PEXE of the lifetime function is

$$\widehat{F}_{PEXE}(x)=\begin{cases}\exp(-z_1 x) & 0\le x\le u_i\\ \exp\{-[z_1u_1+z_2(x-u_1)]\} & u_1\le x\le u_2\\ \exp\{-[z_1u_1+z_2(u_2-u_1)+\ldots+z_i(x-t_{i-1})]\} & u_{i-1}\le x\le u_i, i=3,\ldots,r\\ \exp\{-[z_1u_1+z_2(u_2-u_1)+\ldots+z_r(x-t_{r-1})]\} & x>u_r\end{cases}.$$

(5.19)

5.2.4 Földes, Rejtő, and Winter Estimator (FRWE)

The FRWE kernel density estimator is defined as

$$f_{FRWE}(x)=\frac{1}{h_n}\cdot\sum_{i=1}^{n}\Delta_i k\left(\frac{x-t_i}{h_n}\right), \tag{5.20}$$

where $h_n=\sigma\cdot n^{-\frac{1}{5}}$, σ is the standard deviation of the failure set, Δ_i is the vertical jump of the KME from t_{i-1} to t_i, and $k(x)$ is the standard normal kernel estimator or

$$k(x)=\frac{1}{\sqrt{2\pi}}\exp\left(-\frac{1}{2}\cdot x^2\right). \tag{5.21}$$

The FRWE estimator of the lifetime distribution function is then

$$\widehat{F}_{FRWE}(x)=\int_{-\infty}^{x} f_{FRWE}(t)\,dt. \tag{5.22}$$

5.2.5 Klein, Lee, and Moeschberger Partially Parametric Estimator (KLM)

Let $R(\cdot|\theta)$ represent a parametric survivor function where θ is a vector of estimated parameters from some previous estimation method. In this case, the MLE survivor estimate will be used, where for θ consists of the estimates for β and λ.

Recalling that $\delta_i = 1$ if $t_i \le c_i$ and $\delta_i = 0$ if $t_i > c_i$, the KLM estimator of the lifetime distribution function is

$$\widehat{F}_{KLM}(x) = \frac{\sum_{i=1}^{n} \phi_i\left(x \middle| R(\cdot|\theta)\right)}{n}, \tag{5.23}$$

where

$$\theta_i\left(x \middle| R(\cdot|\theta)\right) = \begin{cases} 1 \, if \;\; X_i > x \\ 0 \, if \;\; X_i \le x \; and \; \delta_i = 1 \\ \dfrac{R(x|\theta)}{R(X_i|\theta)} \, if \;\; X_i \le x \; and \; \delta_i = 0 \end{cases} . \tag{5.24}$$

5.2.6 White Estimator (WH)

The equations to estimate the Weibull shape and scale parameters for the White Estimator (WH) are

$$\frac{1}{\beta_{WH}} = b_{WH} = \frac{\sum_{i=1}^{m} \left(X_i - \bar{X}\right)\left(Y_i - \bar{Y}\right) w_i}{\sum_{i=1}^{m} \left(X_i - \bar{X}\right)^2 w_i} \tag{5.25}$$

and

$$\ln \alpha_{WH} = u_{WH} = \bar{Y} - b_{WH} \bar{X}, \tag{5.26}$$

where

$$\bar{X} = \frac{\sum_{i=1}^{m} X_i w_i}{\sum_{i=1}^{m} w_i}, \tag{5.27}$$

$$\bar{Y} = \frac{\sum_{i=1}^{m} Y_i w_i}{\sum_{i=1}^{m} w_i}, \tag{5.28}$$

$$X_i = \ln\left[\ln\left(\frac{1}{-\exp\left\{-\exp\left[E\left(X_i\right)\right]\right\}}\right)\right], \tag{5.29}$$

$$Y_i = \ln x_i, \tag{5.30}$$

and

$$w_i = \frac{1}{\operatorname{var}\left(X_i\right)}. \tag{5.31}$$

$E\left(X_i\right)$ and $\operatorname{var}\left(X_i\right)$ can be found in Table 1 and Table 2 respectively in [79]. The WH estimator of the lifetime distribution function is then:

$$\widehat{F}_{WH}\left(x\right) = 1 - \exp\left[-\left(\frac{x}{\alpha_{WH}}\right)^{\beta_{WH}}\right]. \tag{5.32}$$

5.2.7 Bain and Engelhardt Estimator (BE)

Using the same notation as above, the equations to estimate the Weibull shape and scale parameters for the Bain and Engelhardt Estimator (BE) [80] are

$$\frac{1}{\beta_{BE}} = b_{BE} = -\sum_{i=1}^{m-1} \frac{Y_i - Y_r}{nk_{r,n}} \tag{5.33}$$

and

$$\ln \alpha_{BE} = u_{BE} = Y_m - E\left[X_m\right] \cdot b_{BE}, \tag{5.34}$$

where

$$k_{m,n} = -\left(\frac{1}{n}\right)\sum_{i=1}^{m-1}\left(E\left[X_i\right] - E\left[X_m\right]\right). \tag{5.35}$$

The BE estimator of the lifetime distribution function is then

$$\widehat{F}_{BE}\left(x\right) = 1 - \exp\left[-\left(\frac{x}{\alpha_{BE}}\right)^{\beta_{BE}}\right]. \tag{5.36}$$

Note: The WH and BE methods are undefined at a censoring level of *n*-1.

5.2.8 Modified Profile Maximum Likelihood Estimator (MPMLE)

The modified profile score function for β in the case of censored data is

$$S_m\left(\beta\right) = \frac{r-2}{\beta} - \left(r\sum_{i=1}^{r}{}^{*} y_i^{\beta}\log y_i\right)\left(\sum_{i=1}^{r}{}^{*} y_i^{\beta}\right)^{-1} + \sum_{i=1}^{r}\log y_i, \tag{5.37}$$

where *r* is the total number of censored observations and the notation

$$\sum_{i=1}^{r}{}^{*} w_i = \sum_{i=1}^{r} w_i + (n-r)w_r \tag{5.38}$$

is used. The modified β is then found by solving $S_m\left(\beta\right) = 0$. The scale parameter can be solved in the same manner as in the MLE method [81].

5.2.9 Ross Estimator (ROSS)

The Ross Maximum Likelihood [82] unbiasing method consists of the same scale parameter found using the MLE method and shape parameter

$$\beta_{ROSS} = \frac{\beta_{MLE}}{1 + \frac{1.37}{r-1.92}\sqrt{\frac{n}{r}}}. \tag{5.39}$$

The ROSS estimator of the lifetime distribution function is then

$$\hat{F}_{ROSS}(x) = 1 - \exp\left[-\left(\frac{x}{\alpha_{MLE}}\right)^{\beta_{ROSS}}\right]. \tag{5.40}$$

5.3 Estimation Analysis

Expanding Reineke's [71] analysis, the MLE, KME, FRWE, PEXE, and KLM methods were evaluated under the following censoring levels, sample sizes, and Weibull shape parameters:

Censoring Levels – (60%-70%], (70%-80%], (80%-90%], (90%-100%]

Sample Size – 5, 6, 8, 10, 15, 20, 25, 30, 35, 40, 50, 60, 70, 80, 90, 100

Weibull Shape Parameter – 3, 4, 5

Data was simulated for all 765 combinations and the Integrated Square Error (ISE) was determined for each case. ISE is defined here as the integrated squared difference between the estimated and true distribution functions or

$$ISE\left(\hat{F}\right) = \int_{-\infty}^{+\infty} \left[\hat{F}(x) - F(x)\right]^2 dx. \tag{5.41}$$

This process was then repeated up to 10,000 times to collect a suitable sample size to determine which method performed best for each case. The ISE values for each case were then evaluated using a pair-wise Kruskal-Wallis (KW) test comparison. The KW test is a nonparametric version of a one-way analysis of variance and is used to determine if the values of one sample are different then the values of another sample. To perform a KW test, the samples are combined into a single sample, the values are sorted from smallest to largest, and a rank is assigned to each value. The average rank of each sample is then compared using the test statistic

$$KW = \frac{12}{N\left(N+1\right)}\sum_{i=1}^{K} n_i \left(\bar{R}_i - \frac{N+1}{2}\right)^2, \qquad (5.42)$$

where K is the number of samples under comparison, n is the size of the sample, and $\bar{R}_i$ is the average rank of the sample. The null hypothesis for this test is that the distributions are the same and it is rejected if $KW > \chi^2_{K-1}$. A total of 10 pair-wise comparisons were made for each test combination and the methods were ranked from best performing to worst. If the hypothesis for two or more methods could not be rejected, these methods were given an equal rank. If the hypothesis was rejected, the method with the smaller mean was given a higher rank.

The analysis was conducted initially replicating each test case 1,000 times. This initial analysis showed that for values of n greater than 30, the MLE method is the best performing regardless of censoring level or shape parameter. For these same test cases, the FRWE method had the second best performance. The results including mean ISE, ISE standard deviation, and rank for all 765 combinations are shown in Appendix A. However, for values of n equal to or less than 30, such a simple conclusion was not apparent. To increase the ability of the KW test to distinguish between paired methods, a second analysis incorporating 10,000 replications was conducted for all combinations of $n \leq 30$ and the best performing method for each combination is shown in Table 6 [83].

Table 6 Initial Estimation Method Selection Results

n=5			
Censoring Level	β=2	β=3	β=4
(60%-70%]	FRWE	FRWE	FRWE
(80%-90%]	FRWE	FRWE	FRWE

n=6			
Censoring Level	β=2	β=3	β=4
(60%-70%]	PEXE	PEXE	PEXE
(80%-90%]	PEXE	PEXE	PEXE

n=10			
Censoring Level	β=2	β=3	β=4
(60%-70%]	PEXE	MLE	MLE
(70%-80%]	PEXE	MLE	FRWE
(80%-90%]	KLM	KLM	FRWE

n=10			
Censoring Level	β=2	β=3	β=4
(60%-70%]	PEXE	MLE	MLE
(70%-80%]	PEXE	MLE	FRWE
(80%-90%]	KLM	KLM	FRWE
(90%-100%]	KLM	KLM	FRWE

n=15			
Censoring Level	β=2	β=3	β=4
(60%-70%]	MLE	MLE	MLE
(70%-80%]	PEXE	MLE	MLE
(80%-90%]	PEXE	MLE	MLE
(90%-100%]	KLM	KME	KME

n=20			
Censoring Level	β=2	β=3	β=4
(60%-70%]	MLE	FRWE	FRWE
(70%-80%]	MLE	MLE	FRWE
(80%-90%]	PEXE	MLE	MLE
(90%-100%]	PEXE	PEXE	KLM

n=25			
Censoring Level	β=2	β=3	β=4
(60%-70%]	MLE	FRWE	FRWE
(70%-80%]	MLE	MLE	MLE
(80%-90%]	MLE	MLE	MLE
(90%-100%]	PEXE	PEXE	KLM

n=30			
Censoring Level	β=2	β=3	β=4
(60%-70%]	MLE	MLE	FRWE
(70%-80%]	MLE	MLE	MLE
(80%-90%]	MLE	MLE	MLE
(90%-100%]	PEXE	KLM	KLM

An examination of this table fails to provide simple rules to follow in choosing an estimation method except that as the MLE begins to dominate as *n* increases. The well known bias increase with the MLE explains the decreased performance with smaller values of *n*. In practice, this table may not be useful since one of the entering arguments, β, is an unknown value in a reliability test. However, the range of the shape parameter could be estimated using expert selection or through existing component values from published data as found in [84] and displayed in Appendix B. An improved application would be to provide a tool for selecting an estimation method given an assumed shape parameter range, n, and known level of censoring. This tool could be used at the completion of a reliability test, when the censored amount is known.

Using this strategy, a second analysis was performed to determine the best performing method given a value of n, censoring amount, and shape parameter range. Four additional methods were incorporated into this analysis to allow for increased competition. These methods were outlined earlier in 5.2.6-5.2.9. Montanari et al. [85] tested a total of six methods including the WH, BE, and ROSS methods for sample sizes of 6, 10, and 20; shape parameters of .5, 1, and 10; and for censoring levels of 30% and 50%. Their results show that the WH and BE estimator performed well at higher levels of censoring. The ROSS method was shown to perform satisfactorily but was not recommended for higher levels of censoring. Yang and Xie's [86] MPMLE method was shown to have less bias than the MLE and ROSS methods and had increased efficiency with lower sample sizes and heavier censoring.

A total of nine estimation methods were tested for all combinations of the following values:

Censoring Levels – $n-1$, $n-2$,…, 1

Sample Size (n) – 5, 10, 15, 20, 30

Weibull Shape Parameter (β) ranges – 1-1.45, 1.5-1.95, 2-2.45, 2.5-2.95, 3-3.45, 3.5-3.95, 4-4.45

Once again, the ISE for each value combination was calculated using 10,000 simulated estimations and a KW test was completed for all 36 pair-wise comparisons. Throughout the simulation, the shape parameter values were uniformly distributed between the specified range values over the 10,000 replications. The best performing method was again determined by evaluating the rank value in the KW test. The p-value was calculated for each comparison and used to determine the significance of the results.

The results of this second analysis are displayed in Appendix C. In these tables, an asterisk indicates that a method is not significantly different than one or more other methods at a 95% confidence level. Generally for these cases, the best performing method displayed slightly outperforms a method that is the best performing method at a neighboring censoring level. For example, in the case of $n = 20$, $\beta = 1.5 - 1.95$, and censored number of 8, the rank values indicated the MPMLE method has the best performing, followed by ROSS, and then MLE. The p-value for the comparison of MPMLE to ROSS was .78 and .54 for the comparison of MPMLE to MLE. The p-values indicate that the MPMLE is not significantly different than the ROSS or MLE; however, it was selected as the best performing due to a lower rank value. The ROSS and MLE method are selected as the best performing at the neighboring censoring value. To obtain more significant results, the replication number could be increased, but 10,000 was used as a reasonable choice for all tests.

The results shown in Appendix C show that for heavier censoring levels, the KME and PEXE methods tend to dominate as best performing. The FRWE also performs well at high censoring levels at lower values of *n*. On the opposite end of the censoring level, the MPMLE and ROSS methods perform well and the MLE is overwhelmingly favored in the middle ranges. At smaller values of *n* and lower censoring levels, the WH, BE, and to a lesser degree FRWE perform well. For n greater than 10 and censoring less than 45%, the pair-wise comparison of the MPMLE and

ROSS methods yielded high p-values indicating that the ISE ranks of these two methods are not significantly distinguishable.

This analysis shows that for varying value combinations, several estimation methods should be employed and carefully selected given the exact scenario. This is clearly evident as the censoring level changes within a fixed β value range. This occurrence will happen often in a practical scenario as the number of censored items will fluctuate from test to test.

For a reliability test of randomly right censored Weibull distributed data, the results obtained by this analysis should be used to select the best performing estimation method. In the cases where the shape parameter range does not match that tested in this analysis, an similar comparison of several estimation methods should be conducted. A further examination of selection methods for a particularly important when high or low censoring is found.

CHAPTER 6

SYSTEM ANALYSIS

6.1 System Introduction

Combining the information from Chapters 2-4, a generalized approach using component failure distribution to determine the stocking policy for any multiple-component system subject to individual replacement times is introduced in this Chapter. As earlier stated in Chapter 3.2, Kabir and Al-Olayan's study [66] presented one of the few policies using multiple components that do not follow a block replacement strategy. Using this study as a baseline, a total of 49 developed stocking policies are tested against the policy values from [66] and the policy costs are compared. In this analysis, the lifetime failure distribution of the components is assumed to be well known. Section 6.2.2 will expand upon these policies when the component failure information is unknown.

6.2.1 Stocking Policy Analysis

To develop our stocking policies, three methods of compiling inventory demand information is used. Two of the three methods employ the predictive forecasting method introduced in Chapter 4. The first method will be called the Failure Forecast and uses only the lifetime failure parameters (equation 4.2) while the second, called the Combined

Forecast, uses both the failure information and the maintenance distribution information (equation 4.3). The third method simply uses historical demand information.

Using the information from these three methods, the Power, Normal, and Naddor (*s, S*) approximations introduced in Chapter 2.3 were used to produce the following inventory policies:

1) Historical Power Approximation
2) Historical Normal Approximation
3) Historical Naddor Approximation
4) Forecasted Failure Power Approximation
5) Forecasted Failure Normal Approximation
6) Forecasted Failure Naddor Approximation
7) Combined Forecast Power Approximation
8) Combined Forecast Normal Approximation
9) Combined Forecast Naddor Approximation.

As shown in Chapter 2.3, these approximations require the holding cost, penalty cost, order cost, lead-time, demand mean μ, and demand variation σ to determine the (*s, S*) inventory parameters. For the Forecasted Failure and Combined Forecast policies, the expected demand and variation was calculated for the future $k = 30$ periods.

A second group of inventory policies were developed using these approximations by determining the expected μ and σ during only the future $\lambda + 1$ periods. These six policies are called the:

10) Forecasted Failure Lead Power Approximation
11) Forecasted Failure Lead Normal Approximation

12) Forecasted Failure Lead Naddor Approximation

13) Combined Forecast Lead Power Approximation

14) Combined Forecast Lead Normal Approximation

15) Combined Forecast Lead Naddor Approximation.

The recursive dynamic programming approach introduced in Chapter 2.4 was combined with the Forecasted Failure and Combined Forecast for $k = 30$ periods providing two additional inventory policies:

16) Forecasted Failure Non-Stationary Solution

17) Combined Forecast Non-Stationary Solution.

Lastly, the optimal (*s, S*) solution presented in Chapter 2.2 was solved to obtain the following inventory policies:

18) Forecasted Failure Optimal Solution

19) Combined Forecast Optimal Solution.

For these policies, the discrete probability density functions for the future *k* periods were determined and used to find a single average discrete probability density function. For the inventory policies tested in this Chapter, the stationarity characteristic of the demand distribution was ignored. For the majority of the scenarios studied, the demand from period to period was shown to be non-stationary using the techniques presented in Chapter 4.4.

These 19 inventory policies were then coupled with two or three maintenance strategies. The first maintenance strategy, called the Fixed strategy, is to replace the part at the optimal replacement time determined by equation 3.5 or upon failure. The second and third maintenance strategies were introduced in Chapter 3.4 and allow for practical maintenance scheduling. Using the Fixed, Normal, and Uniform maintenance strategies, a total of 49 stocking policies were developed. Inventory policies 1 – 6, 10 – 12, 16, and 18 were coupled with all three maintenance strategies. Policies 7 – 9, 13 – 15, 17, and 19 were tested under the Normal and Uniform strategies only. These policies all use the Combined Forecast approach which incorporates these two strategies into the demand prediction. The 49 stocking policies will be referred to by the numerical listing shown previously for the inventory policy following by the maintenance strategy. For example, Policy 1 Normal refers to a stocking policy whose inventory parameters are calculated by the Revised Power Approximation using historical demand that is subject to a Normal maintenance strategy.

The 49 stocking policies were tested against the stocking policies presented in [66] under varying inventory and maintenance costs. These costs, along with the optimal replacement time t^* and inventory policy values, are displayed in Table 7. These cases were repeated for Weibull parameter (β,η) values of (1.5,100), (2,100), and (3,100). The comparison results located in Appendix D, E, and F display the cost difference between the stocking policies presented in this study and the stocking policy of Kabir and Al-Olayan. The optimal replacement time determined by equation 3.4 and the inventory policy values are also displayed in these Appendices. The inventory policy values displayed are the minimum and maximum values calculated during the comparison, as these values vary from period to period as latest information is used to determine the inventory policies.

The inventory cost accounting for stocking policies developed in this study differ slightly from that used by Kabir and Al-Olayan. The policies developed in this study are all periodically reviewed where the policies in Kabir and Al-Olayan are continuous. In a periodic policy, holding and shortage costs are totaled for full periods. Holding costs are not accumulated for parts that are removed from inventory during the period. Similarly, shortage costs will be added for each period a part is short and for the period that the part is received. In a continuous policy, the costs are computed for the exact time it is held or short. The cost accounting differences between these two policies does not give either policy a significant advantage.

Table 7: Kabir and Al-Olayan Policy Values

	Inventory values			Maintenance costs			Inventory values	
Run	Ordering (K)	Penalty (p)	Holding (h)	Maintenance (C_m)	Failure (C_f)	$t*$	s	S
1	8.75	13.5	0.6875	25	55	210	0	2
2	16.25	28.5	0.6875	25	55	100	0	3
3	16.25	13.5	1.5625	25	55	210	0	2
4	16.25	13.5	0.6875	25	55	210	0	2
5	16.25	13.5	0.6875	35	85	100	0	3
6	8.75	28.5	1.5625	25	55	210	0	2
7	8.75	28.5	0.6875	35	55	220	1	2
8	8.75	28.5	0.6875	25	85	100	0	3
9	8.75	13.5	1.5625	35	55	210	0	2
10	8.75	13.5	1.5625	35	85	210	0	2
11	8.75	13.5	0.6875	35	85	210	0	2
12	16.25	28.5	1.5625	35	55	210	0	2
13	16.25	28.5	1.5625	25	85	100	0	3
14	16.25	28.5	0.6875	35	85	100	0	3
15	16.25	13.8	1.5625	35	85	210	0	2
16	8.75	28.5	1.5625	35	85	210	0	2
17	20	21	1.125	30	70	210	0	2
18	5	21	1.125	30	70	210	0	2
19	12.5	36	1.125	30	70	100	0	3
20	12.5	6	1.125	30	70	210	0	2
21	12.5	21	2	30	70	210	0	2
22	12.5	21	0.25	30	70	220	1	3
23	12.5	21	1.125	40	70	210	0	2
24	12.5	21	1.125	20	70	100	0	3
25	12.5	21	1.125	30	100	100	0	3
26	12.5	21	1.125	30	40	210	0	2
27	12.5	21	1.125	30	70	210	0	2

The lead-time for part orders in the Kabir and Al-Olayan policies is stochastic and defined by the Weibull parameters (3.2,10). The period length for the periodically reviewed policies is the mean value of this stochastic lead-time. This value is the solution to $\beta\Gamma\left(1+\frac{1}{\eta}\right)$. For this period length, the lead-time for the periodic policies was typically one or two periods.

The comparison results were obtained by compiling stocking policy costs for a simulated 1,000 periods which is equivalent to a time length of $1000\cdot\beta\Gamma\left(1+\frac{1}{\eta}\right)$. For the studied Weibull parameters, the simulation length was: 1.5 – 1491 time units, 2.0 – 1989 time units, and 3 – 2983 time units. A beginning span of 15 periods was used to initialize part ages. At the conclusion of this beginning span all costs and inventory orders were reset to zero. The historical demand information began with information from the previous ten periods and accumulated information up until the 1000th period. As stated before, the conditional probabilities of failure used in the predicted demand forecasting for policies 4 – 9 and 16 – 17 were calculated for $k = 30$ future periods.

In this comparison, the values used for the Normal and Uniform maintenance strategies as discussed in Chapter 3.4 were $PC = .99$, $CR = .025$, and $UR = .025$. Using these parameters for the first run of $\beta = 1.5$, the Normal maintenance strategy allows for maintenance scheduling where 99% of the replacement times are normally distributed between 177.1 – 186.2 time units. Likewise, the Uniform maintenance strategy allows replacement times to occur uniformly distributed between 177.1 and 186.2 time units. If this maintenance window is unrealistic or needs to be widened or reduced, the *CR* value could be increased or decreased.

The results in Appendix D – F show that stocking Policy 10 Fixed and 12 Fixed have the best performance against the Kabir and Al-Olayan policies over the entirety of the runs. Policy 10 Fixed has a maximum average difference of 8% for $\beta = 1.5$ and returns an average cost savings of 1% for $\beta = 3.0$. Policy 12 Fixed has a maximum average difference of 6% for $\beta = 2.0$ and equals the Kabir and Al-Olayan cost for $\beta = 3.0$. In general, Policies 4 – 6, 13 – 16, 18, and 19 performed poorly for all maintenance strategies and cost groups. Although not as cost effective as when coupled with a Fixed maintenance strategy, Policy 10 and 12 show strong performances when a flexible maintenance policy is applied as seen in the results for Policy 10 Normal/Uniform and Policy 12 Normal/Uniform.

Appendix G shows additional cost analysis results for *CR/UR* values of .05, .075, and .1 with $\beta = 3.0$. These additional parameter values increase the scheduling window up to $\pm 10\%$ of the optimal replacement time. A similar comparison could be used to make an informed decision when altering maintenance policy to allow for a wider scheduling window. In these results, the cost difference between Policy 12 Normal with *CR* value of .05 is not substantially different than the same policy with a *CR* value of .025 although it expands the scheduling window by 100%.

Several of the policies developed in this study performed poorly against the Kabir and Al-Olayan policy. Policy 10 and 12 performed well with all maintenance strategies and will be used along with Policy 11 in the following section. A total of five operating parts were used in this stocking policy comparison, which is a relatively small system. A larger system would provide an increased sample size of parts that would better reflect the predicted demand forecasts used to build the presented stocking policies.

Note: The analysis conducted in this section was first presented in [69].

6.2.2 Stocking Policy with Failure Estimation

The system outlined in the following sections bring together the ideas presented throughout this study. The comparison in the previous section was completed with the assumption that the part lifetime failure distribution is known. Using this same periodic review of the inventory/maintenance system, the estimation selection method presented in Chapter 5 can be applied to a system where the failure distribution is unknown. For any system matching the description given in Chapter 1.2, a dynamic inventory/maintenance system can be employed to select the inventory and maintenance policies from one period to the next.

6.2.2.1 System Algorithm

The proposed system model is outlined in Figure 4. The system values shown on the right side of the figure should be known prior to implementing the algorithm. The system cost values such as C_f, C_p, h, p, and, K can be updated as they may change over the life of the model. This study does not discuss how these cost values are assigned, but all are commonly used and their values should be available in any existing system. To track the ages of each part, an identifier such as a serial number should be assigned to each part. When Part i is replaced, the replacement part is then assigned the same identifier i.

The variables on the left side of the figure are called decision variables. These values can be adjusted by a maintenance manager to ensure adequate time is available to perform the maintenance. The maintenance manager should then ensure these values accurately reflect the maintenance performance.

The following is a further description of each step in the algorithm. Step 0 is carried out only when the model is first applied to a system. Upon the completion of Step 5, the algorithm returns to Step 1 and Step 0 is never visited again.

Step 0:

This model was developed under the assumption that the reliability information for the part is not well known. However, values for the failure distribution parameters need to be entered to initiate the algorithm. These values can be determined as suggested in Chapter 5.3.

Step 1:

If the failure distribution parameters or maintenance cost values have changed since the previous iteration, the optimal replacement time will be re-calculated. If no changes to the distribution parameters or cost values have occurred, the optimal replacement time will not change and this step can be bypassed.

Step 2:

If the optimal replacement time or decision variables have changed since the previous iteration, the system maintenance policy will be re-calculated. The new optimal replacement time t^* becomes the mean or expected time for part maintenance and the maintenance strategy values are determined as shown in Chapter 3.4. This new maintenance policy should be applied to the current period.

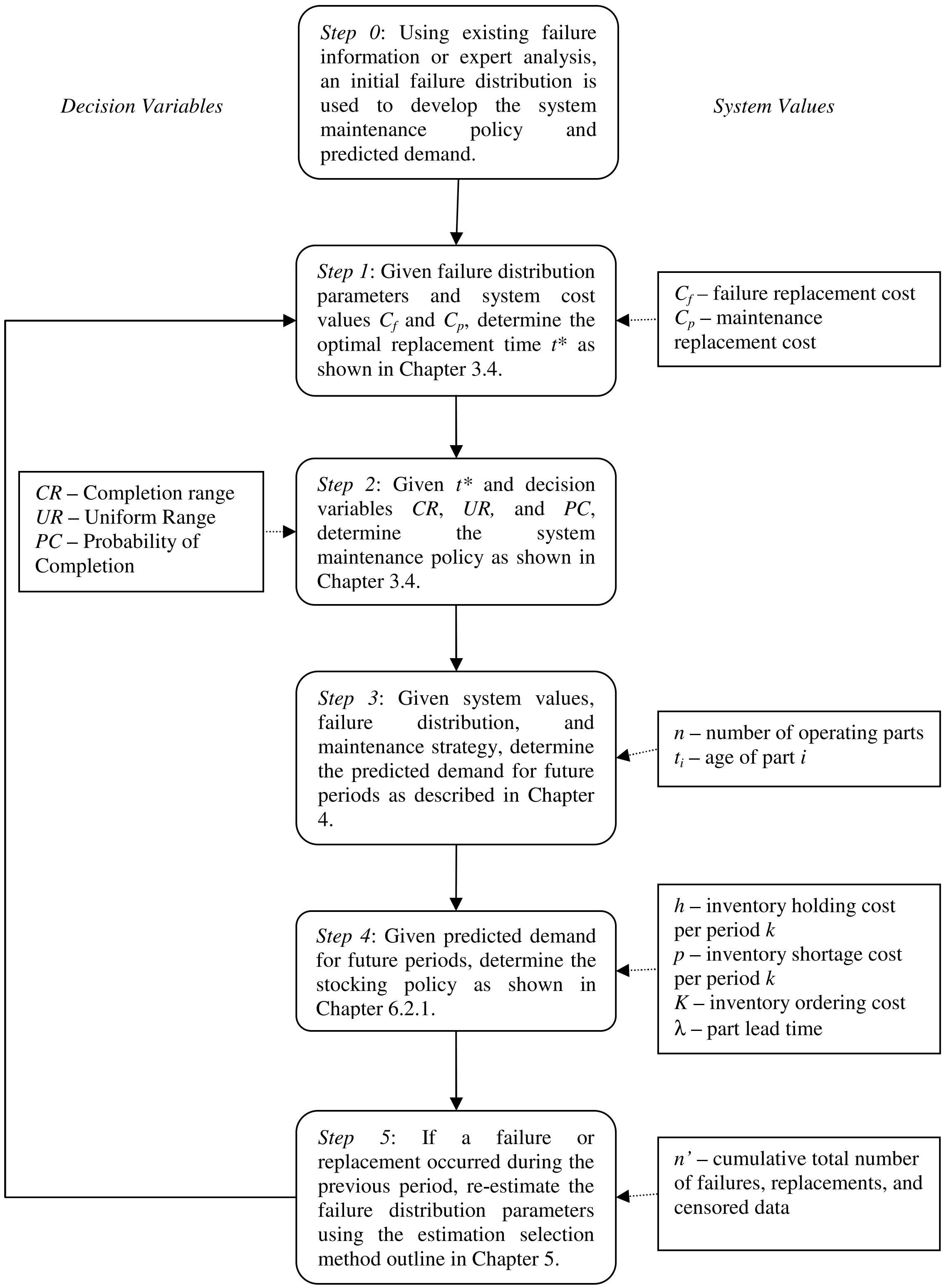

Figure 4 System Model

Step 3:

If any values from Step 1 – 2 or any system values have changed, the stocking policy values will need to be re-calculated. If all these values have not changed, the conditional failure probabilities and predicted demand remain the same as calculated during the previous iteration. The age of each part should be updated and recorded after every iteration.

Step 4:

If the predicted demand or any system values have changed, the inventory policy will need to be re-calculated using the appropriate inventory solution method. The new reorder level and order-up-to amount should be applied for the current period.

Step 5:

At the conclusion of any period containing a part replacement, Step 5 or the re-estimation of the failure distribution parameters should be completed. The completion of Step 5 then triggers Steps 1 – 4 to be completed at the beginning of the following period. The need to complete Step 5 could also triggered by a significant number of failures. This immediate re-estimation would provide increased model performance prior to the completion of the current period. If a part replacement occurred during the previous period, either due to failure or maintenance, a new estimate for the distribution parameters will be calculated using the appropriate estimation method for the value of n and the cumulative censored number. Through each replacement, the reliability data increases and the distribution parameter estimation improves. The cumulative reliability data, n', contains the age of a part at each maintenance replacement, the age of a part at each failure, and the age of the censored parts at the time of a part failure. The censored data dilutes the parameter estimation. Therefore, whenever a part is replaced due to maintenance, the censored times that were recorded earlier during that part's life are removed from *n'*. Each record in *n'* consists of the time of failure or censor and the mode indicator, where 1 indicates a failure and 0 indicates a censor. Additionally, each record in *n*' has a subscript *i,k*, where *i* indicates the part identifier and *k* indicates the period when the data was collected. The following example is provided to further explain how *n'* is updated.

At the beginning of period 1, four parts have ages $t_1 = 6.1, t_2 = 2.6, t_3 = 5.5,$ and $t_4 = 5.9.$ The period length is .25 years. During period 1, Part 3 fails and reliability data is gathered with Parts 1, 2, and 4 being censored.

$$n' = (6.35,0)_{1,1}, (2.85,0)_{2,1}, (5.75,1)_{3,1}, (6.15,0)_{4,1}$$

During period 2, Part 4 fails and reliability data is gathered with Parts 1, 2, and 3 being censored.

$$n' = (6.35,0)_{1,1}, (2.85,0)_{2,1}, (5.75,1)_{3,1}, (6.15,0)_{4,1}, (6.60,0)_{1,2}, (3.10,0)_{2,2}, (.25,0)_{3,2}, (6.40,1)_{4,2}$$

During period 3, Part 1 is replaced through maintenance and this censored time is included in the reliability data while the censored values for Part 1 at the failure of Part 3 and 4 are discarded. The discarded data was previously listed with the subscripts 1,1 and 1,2.

$$n' = (2.85,0)_{2,1}, (5.75,1)_{3,1}, (6.15,0)_{4,1}, (3.10,0)_{2,2}, (.25,0)_{3,2}, (6.40,1)_{4,2}, (6.85,0)_{1,3}$$

6.2.2.2 Policy Analysis

Using the system algorithm described in the previous section, Policies 10 – 12 coupled with the Fixed, Normal, and Uniform maintenance strategies were tested using the inventory and maintenance costs from Run 1 listed in Table 7. The Matlab code for the analysis completed in this section can be found in Appendix H. For the initial distribution parameters required in Step 0, 11 shape and scale parameter pairs were used. The accumulated costs using these parameter pairs for a simulated length of 100 periods are displayed in Table 8 and Table 9.

Run 1 of Table 8 is the base run using the system algorithm with known failure distribution parameters. In this run, the distribution estimation procedure was bypassed

Table 8: System Results

	Step 0 Values		Fixed Maintenance Strategy			Normal Maintenance Strategy			Uniform Maintenance Strategy		
	Shape	Scale	P10	P11	P12	P10	P11	P12	P10	P11	P12
Run 1	3.0	100	5553.1	6787.0	5288.6	9640.1	7524.5	9174.7	4948.9	5121.8	4837.1
Run 2	2.5	100	4926.0	5833.2	5654.8	14578.2	8052.3	12087.3	13101.2	6212.8	11201.0
Run 3	3.5	100	4518.5	4911.3	4498.2	5161.2	5397.2	5058.5	5607.9	5543.1	5546.1
Run 4	3.0	90	8262.7	7336.7	5709.3	5101.7	5263.3	5372.4	6119.9	6619.2	5590.3
Run 5	2.5	90	4818.2	5865.1	4510.6	4988.1	5536.4	5404.4	5609.8	5490.6	5726.6
Run 6	3.5	90	5780.0	6551.1	5578.1	6689.3	4954.4	6513.3	9481.9	5393.6	8327.3
Run 7	3.0	110	6043.9	7326.7	5757.0	6187.6	5334.3	5826.7	4904.7	4846.6	4646.9
Run 8	2.5	110	4864.9	6781.9	5349.0	11043.8	7225.5	7926.1	5112.7	5048.9	4634.0
Run 9	3.5	110	13429.8	7613.1	9408.2	5848.0	5490.5	5582.6	5492.8	5255.6	5212.1
Mean Cost (Run 2-9)=			6580.5	6527.4	5808.2	7449.7	5906.7	6721.4	6928.8	5551.3	6360.5
Std Dev (Run 2-9)=			3016.8	931.2	1542.7	3490.4	1105.9	2351.4	2883.8	590.8	2273.9
Δ (Run 1 - Mean)=			19%	-4%	10%	-23%	-21%	-27%	40%	8%	31%

and the true distribution parameters were used to develop the predicted demand forecast. In Runs 2 – 9 of Table 8, the Step 0 values were varied to allow Step 5 of the algorithm to be tested. The Step 0 values represent the initial parameter estimates that could be used when applying this system and were varied $\pm 17\%$ for the shape parameter and $\pm 10\%$ for the scale parameter in Runs 2 – 9. These values were used to represent a well estimated initial distribution. The mean values for all policies are shown along with the standard deviation and percent cost difference between the base run cost and mean value of Runs 2 – 9. A negative value indicates a cost savings over the base run while a positive value represents a cost increase.

From the information displayed in Table 8, Policy 11 Uniform shows excellent performance when an initial estimate for the distribution parameters is used. Policy 11 Uniform returns the lowest mean value and standard deviation for Runs 2 – 9. The low value for standard deviations indicates that the policy has the most stable performance across varying initial estimates. This policy also returned a low average cost increase as compared to an identical system with known parameters.

Table 9: Poorly Estimated Parameters

	Step 0 Values		Fixed Maintenance Strategy			Normal Maintenance Strategy			Uniform Maintenance Strategy		
	Shape	Scale	P10	P11	P12	P10	P11	P12	P10	P11	P12
Run 10	3.0	50	11192.6	8556.5	8810.3	14756.1	10215.3	10364.7	14852.1	9977.5	10970.9
Run 11	3.0	150	5256.1	6286.4	5084.0	4398.6	4675.2	4536.1	4674.0	4718.8	4674.7

The run scenarios in Table 9 began with a poorly estimated initial scale parameter. These cost figures show that a conservative estimate of the scale paramter returns inflated costs. In this run, many components were replaced due to maintenance which led to a highly-censored distribution estimation. The operational lifetime of the parts were shortened and frequent preventive replacements increased the inventory and maintenance costs which led to a high total cost. Whereas in Run 11, the high scale value allowed more early period failures and subsequently an improved distribution estimation.

Appendix I shows the distribution parameters and optimal replacement times as estimated for each period of Run 10. Likewise, Appendix J shows these values for Run 11. An examination of these Appendices reveals that the over-estimated scale value in Run 11 allows the estimation procedure to quickly close in on the actual parameter values. This comparison lends evidence to suggest that an over-estimated scale parameter may be an effective strategy when the distribution parameters are unknown.

CHAPTER 7

CONCLUSION AND FUTURE RESEARCH

7.1 Conclusion

This research effort first presented ideas and then proposed a system model to further expand the body of work in combined or jointly modeled inventory/maintenance systems. Several unique concepts were introduced and applied to build this system model, including using the lifetime failure characteristics of parts to develop a predicted demand forecast, maintenance strategies that allow flexible scheduling, and an estimation selection method for highly-censored Weibull data.

Several stocking policies were tested against existing policies presented in a study by Kabir and Al-Olayan. This testing showed that the predicted demand over a lead-time coupled with a flexible maintenance strategy provides adequate results as highlighted in Chapter 6.2.1. This comparison adds to the limited amount of existing studies dealing with multiple components subject to differing replacement times. As indicated earlier, most stocking policy studies examining multiple components limit the replacement strategy to a block replacement type.

Applying these stocking policies to a part system with unknown failure distribution parameters, the estimation selection method described in Chapter 5 allowed this dynamic system to produce inventory and maintenance guidance. Section 6.2.2.1 presented an algorithm that can be applied to any system matching the description outlined in Section 1.2. The proposed system algorithm performed well in several scenarios that began with limited or non-existent part reliability data. The best results were obtained from Policy 11 Uniform as shown in Table 8. This policy returned the most stable performance and lowest costs for a stocking policy that allows for flexible maintenance scheduling.

Flexible maintenance strategies are used to provide a scheduling window for part replacements. This idea is not often used in literature but is almost always used in application. Small expansions of this scheduling window are shown in Appendix G to have limited effect on costs.

7.2 Research Extensions

Additional complexity could be included to further the applicability of this study. Such complexity could include adding to and refining the system cost values. For example, this study utilized a single variable to account for the cost of a part failure. This single variable could be broken down into sub-costs including, but not limited to a cost to account for the operational time lost, cost to determine the cause of failure, and emergency travel costs of maintenance personnel. Other system costs, such as the inventory holding costs, could also be expanded. Further analysis with multiple replications of the proposed algorithm could be completed using these additional costs to build a sensitivity analysis.

A supplementary extension to this research would include adding increased complexity to the stocking policy. A more complex policy would include spares that are new and/or comprised of previously used parts that have been overhauled. The inventory stock would then be replenished by orders placed and by parts returned from overhaul. The failure characteristic of these overhauled parts may or may not then vary from that of a new one.

This system utilized a rather simple policy where parts either fail or are replaced prior to failure. A major extension to this research would include the introduction of Condition Based Monitoring (CBM), Predictive Maintenance, and Reliability Centered Maintenance techniques into the system model. These techniques are widely used in industry and their inclusion would provide even further applicability. The idea of imperfect maintenance would also increase the complexity of this model. Imperfect maintenance is any maintenance action that is performed on the part, whether it is the action of replacing the part or minor preventive maintenance in between replacements, which causes damage or a condition that accelerates failure.

Table 9 suggests that the performance of the system algorithm is excellent when an over-estimated scale parameter is used for the initial parameter value. This over-estimation would normally allow a larger amount of early failures. These failures however provide early reliability data to determine better parameter estimates. Additional testing of this or other initial estimation strategies would also be beneficial to this study. If the over-estimation strategy is shown to be an effective option, failure costs and a increased demand on inventory could be anticipated and planned for.

A few ideas discussed in this study could have some application in medical studies. Organ transplant surgeries can be viewed as part replacements, where the transplanted organ takes on the role of a “spare part”. The system algorithm could be applied to a medical system of *n* patients that have failing parts or organs that place a demand on the inventory stock. Such a system is similar to the one in this study in that there are multiple operating components that have varying replacement times.

Finally, a case study using existing data or real-time data should be completed to validate this system model. Analysis of real-time data would allow the model to showcase the dynamic characteristics, specifically the ability to allow flexible maintenance scheduling. The data obtained from the actual replacement times could then be used to determine if the modeled maintenance strategy accurately reflects the observed replacement times.

REFERENCES

[1] A. A. Ghobbar and C. H. Friend, "Evaluation of Forecasting Methods for Intermittent Parts Demand in the Field of Aviation: A Predictive Model," *Computers and Operations Research*, vol. 30, pp. 2097-2114, 2003.

[2] P. Flint, "MRO: Maintaining Optimism," *Air Transport World*, pp. 38, 2005.

[3] "Army Needs Plan to Implement Depot Maintenance Report's Recommendations," United States General Accounting Office 2004.

[4] "Air Force Plans and Initiatives to Mitigate Spare Parts Shortages Need Better Implementation," United States General Accounting Office 2003.

[5] "The Army Needs a Plan to Overcome Critical Spare Parts Shortages," United States General Accounting Office 2003.

[6] "Overall Inventory and Requirements Are Increasing, but Some Reductions in Navy Requirements Are Possible," United States General Accounting Office 2003.

[7] B. Sani and B. G. Kingsman, "Selecting the best periodic inventory control and demand forecasting methods for low demand items," *The Journal of the Operational Research Society*, vol. 48, pp. 700-713, 1997.

[8] D. L. Iglehart, "Optimality of (s,S) Policies in the Infinite Horizon Dynamic Inventory Problem," *Management Science*, vol. 9, pp. 259-267, 1963.

[9] K. J. Arrow, T. Harris, and J. Marschak, "Optimal Inventory Policy," *Econometrica*, vol. 19, pp. 250-272, 1951.

[10] H. Scarf, "The Optimality of (S,s) Policies in the Dynamic Inventory Problem," in *Mathematical Methods in the Social Sciences*, vol. IV, K. J. Arrow, S. Karlin, and P. Suppes, Eds. Stanford: Stanford University Press, 1959, pp. 196-202.

[11] K. J. Arrow, S. Karlin, and H. Scarf, *Studies in the Mathematical Theory of Inventory and Production*, vol. I. Stanford: Standford University Press, 1958.

[12] E. Zabel, "A Note on the Optimality of (S,s) Policies in Inventory Theory," *Management Science*, vol. 9, pp. 123-125, 1962.

[13] K. J. Arrow, "The Genesis of "Optimal Inventory Policy"," *Operations Research*, vol. 50, pp. 1-2, 2002.

[14] A. Dvoretsky, J. Kiefer, and J. Wolfowitz, "The Inventory Problem: 1. Case of Known Distributions of Demand," *Econometrica*, vol. 20, pp. 187-222, 1952.

[15] A. Dvoretsky, J. Kiefer, and J. Wolfowitz, "The Inventory Problem: II. Case of Unknown Distributions of Demand," *Econometrica*, vol. 20, pp. 450-466, 1952.

[16] A. Dvoretsky, J. Kiefer, and J. Wolfowitz, "On the Optimal Character of the (s,S) Policy in Inventory Theory," *Econometrica*, vol. 21, pp. 586-596, 1953.

[17] R. Bellman, I. Glicksberg, and O. Gross, "On the Optimal Inventory Equation," *Management Science*, vol. 2, pp. 83-104, 1955.

[18] T. M. Whitin and J. W. T. Youngs, "A Method for Calculating Optimal Inventory Levels and Delivery Time," *Naval Research Logistics Quarterly*, vol. 2, pp. 157-174, 1955.

[19] H. M. Wagner and T. M. Whitin, "Dynamic Version of the Economic Lot Size Model," *Management Science*, vol. 5, pp. 89-96, 1958.

[20] S. Karlin and A. Fabens, "The (s,S) inventory model under Markovian demand process," in *Mathematical Methods in the Social Sciences*, K. J. Arrow, S. Karlin, and P. Suppes, Eds. Stanford: Stanford University Press, 1959.

[21] H. Scarf, "Bayes Solutions of the Statistical Inventory Problem," *The Annals of Mathematical Statistics*, vol. 30, pp. 490-508, 1959.

[22] H. P. Galliher, P. M. Morse, and M. Simond, "Dynamics of Two Classes of Continuous-Review Inventory Systems," *Operations Research*, vol. 7, pp. 362-294, 1959.

[23] S. Karlin, "Dynamic Inventory Policy with Varying Stochastic Demands," *Management Science*, vol. 6, pp. 231-258, 1960.

[24] Y. Feng and B. Xiao, "A New Algorithm for Computing Optimal (s,S) Policies in a Stochastic Single Item/Location Inventory System," *IIE Transactions*, vol. 32, pp. 1081-1090, 2000.

[25] A. F. Veinott and H. M. Wagner, "Computing Optimal (s,S) Inventory Policies," *Management Science*, vol. 11, pp. 525-552, 1965.

[26] B. C. Archibald and E. A. Silver, "(s,S) Policies Under Continuous Review and Discrete Compound Poisson Demand," *Management Science*, vol. 24, pp. 899-909, 1978.

[27] A. Federgruen and P. Zipkin, "An Efficient Algorithm for Computing Optimal (s,S) Policies," *Operations Research*, vol. 32, pp. 1268-1285, 1982.

[28] A. Federgruen and P. Zipkin, "Computing Optimal (s,S) Policies in Inventory Models with Continuous Demands," *Advances in Applied Probability*, vol. 17, pp. 424-442, 1985.

[29] Y.-S. Zheng and A. Federgruen, "Finding Optimal (s,S) Policies is about as Simple as Evaluating a Single Policy," *Operations Research*, vol. 39, pp. 654-665, 1991.

[30] J. J. Bates and J. L. Zayas-Castro, "A Note on "A New Algorithm for Computing the Optimal (s, S) Policies in a Stochastic Single Item/Location Inventory System"," *IMSE University of South Florida Working Paper*, 2007.

[31] H. M. Wagner, M. O'Hagan, and B. Lundh, "An Empirical Study of Exactly and Approximately Optimal Inventory Policies," *Management Science*, vol. 11, pp. 690-723, 1965.

[32] R. D. Snyder, "Computation of (S,s) Ordering Policy Parameters," *Management Science*, vol. 21, pp. 223-229, 1974.

[33] H. Schneider, "Methods for Determining the Re-Order Point of an (s,S) Ordering Policy when a Service Level is Specified," *The Journal of the Operational Research Society*, vol. 29, pp. 1181-1193, 1978.

[34] J. R. Freeland and E. L. Porteus, "Evaluating the effectiveness of a new method for computing approximately optimal (s,S) inventory policies," *Operations Research*, vol. 28, pp. 353-364, 1980.

[35] H. C. Tijms and H. Groenevelt, "Simple Approximations for the Reorder Point in Periodic Inventory Systems with Service Level," *European Journal of Operational Research*, vol. 17, pp. 175-190, 1984.

[36] E. L. Porteus, "Numerical comparisons of inventory policies for periodic review systems," *Operations Research*, vol. 33, pp. 134-152, 1985.

[37] I. Sahin and D. Sinha, "On Asymptotic Approximations for (s,S) Policies," *Stochastic Analysis and Applications*, vol. 5, pp. 189-212, 1987.

[38] D. M. Roberts, "Approximations to optimal policies in a dynamic inventory model," in *Studies in applied probability and management science*, K. J. Arrow, S. Karlin, and H. Scarf, Eds. Stanford, California: Stanford University Press, 1962, pp. 207-229.

[39] H. M. Wagner, *Principles of Management Science*. USA: Prentice-Hall, 1975.

[40] E. Naddor, "Optimal and heuristic decisions on single and multi-item inventory systems," *Management Science*, vol. 21, pp. 1234-1249, 1975.

[41] E. Naddor, *Inventory Systems*. New York: John Wiley & Sons, Inc., 1966.

[42] R. Ehrhardt and C. Mosier, "A revision of the power approximation for computing (s,S) policies," *Management Science*, vol. 30, pp. 618-622, 1984.

[43] R. Ehrhardt, "The power approximation for computing (s,S) policies," *Management Science*, vol. 25, pp. 777-786, 1979.

[44] R. Ehrhardt and H. M. Wagner, "Inventory Models and Practice," in *Advanced Techniques in the Practice of Operations Research*, H. J. Greenberg, F. H. Murphy, and S. H. Shaw, Eds. New York, New York: Elsevier Science Publishing Co., Inc., 1982, pp. 251-332.

[45] S. P. Sethi and F. Cheng, "Optimality of (s,S) Policies in Inventory Models with Markovian Demand," *Operations Research*, vol. 45, pp. 931-939, 1997.

[46] S. Bollapragada and T. E. Morton, "A Simple Heuristic for Computing Nonstationary (s,S) Policies," *Operations Research*, vol. 47, pp. 576-584, 1999.

[47] R. G. Askin, "A Procedure for Production Lot Sizing with Probalisitic Dynamic Demand," *IIE Transactions*, vol. 13, pp. 132-137, 1981.

[48] C. H. Falkner, "Jointly Optimal Inventory and Maintenance Policies for Stochastically Failing Equipment," *Operations Research*, vol. 16, pp. 587-601, 1968.

[49] L. C. Thomas and S. Osaki, "An optimal ordering policy for a spare unit with lead time," *European Journal of Operational Research*, vol. 2, pp. 409-419, 1978.

[50] L. C. Thomas and S. Osaki, "A Note on Ordering Policy," *IEEE Transactions on Reliability*, vol. 27, pp. 380-381, 1978.

[51] S. Osaki, "An ordering policy with lead time," *International Journal of Systems Science*, vol. 8, pp. 1091-1095, 1977.

[52] T. Dohi, N. Kaio, and S. Osaki, "On the optimal ordering policies in maintenance theory - survey and applications," *Applied Stochastic Models and Data Analysis*, vol. 14, pp. 309-321, 1998.

[53] T. Dohi, N. Kaio, and S. Osaki, "Optimal ordering policies with time-dependent delay structure," *Journal of Quality in Maintenance Engineering*, vol. 2, pp. 50-62, 1996.

[54] N. Kaio and S. Osaki, "Optimal ordering policies with lead time for an operating unit in preventive maintenance," *IEEE Transactions on Reliability*, vol. R-27, pp. 270-271, 1978.

[55] N. Kaio and S. Osaki, "Optimal planned maintenance with salvage cost," *International Journal of Production Research*, vol. 16, pp. 249-257, 1978.

[56] N. Kaio and S. Osaki, "Discrete-time ordering policies," *IEEE Transactions on Reliability*, vol. R-29, pp. 405-406, 1979.

[57] N. Kaio and S. Osaki, "Optimal ordering policies with two types of randomized lead times," *Computers for Mathematical Applications*, vol. 19, pp. 43-52, 1990.

[58] N. Kaio and S. Osaki, "Discrete time ordering policies with minimal repair," *RAIRO Operations Research*, vol. 18, pp. 515-523, 1980.

[59] M. J. Armstrong and D. A. Atkins, "Joint optimization of maintenance and inventory policies for a simple system," *IIE Transactions*, vol. 28, pp. 415-424, 1996.

[60] M. J. Armstrong and D. A. Atkins, "A note on joint optimization of maintenance and inventory," *IIE Transactions*, vol. 30, pp. 143-149, 1998.

[61] R. Sarker and A. Haque, "Optimization of maintenance and spare provisioning policy using simulation," *Applied Mathematical Modeling*, vol. 24, pp. 721-760, 2000.

[62] T. S. Vaughan, "Failure replacement and preventive maintenance spare parts ordering policy," *Euroupean Journal of Operational Research*, vol. 161, pp. 183-190, 2005.

[63] Y. K. Yoo, K.-J. Kim, and J. Seo, "Optimal Joint Spare Stocking and Block Replacement Policy (Cost Modeling of Spare Stocking and Block Replacement)," *International Journal of Advanced Manufacturing Technology*, vol. 18, pp. 906-909, 2001.

[64] A. Chelbi and A. Daoud, Kadi, "Spare provisioning strategy for preventively replaced systems subjected to random failure," *International Journal of Production Economics*, vol. 74, pp. 183-189, 2001.

[65] A. B. M. Z. Kabir and A. S. Al-Olayan, "Joint Optimization of Age Replacement and Continuous Review Spare Provisioning Policy," *International Journal of Operations & Production Management*, vol. 14, pp. 53-69, 1994.

[66] A. B. M. Z. Kabir and A. S. Al-Olayan, "A stocking policy for spare part provisioning under age based preventive replacement," *European Journal of Operational Research*, vol. 90, pp. 171-181, 1996.

[67] R. Barlow and L. Hunter, "Optimum Preventive Maintenance Policies," *Operations Research*, vol. 8, pp. 90-100, 1960.

[68] R. E. Barlow and F. Proschan, *Mathematical Theory of Reliability*. New York: Wiley, 1965.

[69] J. J. Bates and J. L. Zayas-Castro, "Stocking Policies for Multiple Operating Components," *Journal of Quality in Maintenance Engineering (in review)*, 2007.

[70] R. A. Brualdi, *Introduction to Combinatorics*, 4 ed. Upper Saddle River, New Jersey: Pearson Education, Inc., 2004.

[71] D. M. Reineke, "Estimation and Goodness-of-Fit in the case of Randomly Censored Lifetime Data," in *Department of Mathematics and Statistics*: Air Force Institute of Technology, 1999, pp. 215.

[72] J. Kitchen, N. A. Langberg, and F. Proschan, "A New Method for Estimating Life Distributions from Incomplete Data," *Statistics & Decisions*, vol. 1, pp. 241-255, 1983.

[73] A. Földes, L. Rejto, and B. B. Winter, "Strong Consistency Properties of Nonparametric Estimators for Randomly Censored Data, II: Estimation of Density and Failure Rate," *Periodica Mathematica Hungarica*, vol. 12, pp. 15-29, 1981.

[74] J. P. Klein, S.-C. Lee, and M. L. Moeschberger, "A Partially Parametric Estimator of Survival in the Presence of Randomly Censored Data," *Biometrics*, vol. 46, pp. 795-811, 1990.

[75] E. L. Kaplan and P. Meier, "Nonparametric Estimation from Incomplete Observations," *Journal of the American Statistical Association*, vol. 53, pp. 457-481, 1958.

[76] L. M. Leemis, *Reliability: probabilistic models and statistical methods*. Englewood Cliffs, New Jersey: Prentice-Hall, Inc., 1995.

[77] E. Garfield, "The Most-Cited Papers of All Time, SCI 1945-1988. Part 1B. Superstars New to the SCI Top 100," *Current Comments*, vol. 8, pp. 3-13, 1990.

[78] "ISI Web of Knowledge," 2006.

[79] J. S. White, "The Moments of Log-Weibull Order Statistic," General Motors Corporation Research Laboratories, Warren, Michigan, 1967.

[80] M. Engelhardt and L. J. Bain, "Simplified Statistical Procedures for the Weibull or Extreme-Value Distribution," *Technometrics*, vol. 19, pp. 323-331, 1977.

[81] Z. Yang and M. Xie, "Efficient Estimation of the Weibul Shape Parameter Based on a Modified Profile Likelihood," *Journal of Statistical Computation and Simulation*, vol. 73, pp. 115-123, 2003.

[82] R. Ross, "Bias and Standard Deviation due to Weibull Parameter Estimation for Small Data Sets," *IEEE Transactions on Dielectrics and Electrical Insulation*, vol. 3, pp. 28-42, 1996.

[83] J. J. Bates and J. L. Zayas-Castro, "Selecting the Best Performing Weibull Estimation Method for Randomly Right Censored Data," *International Journal of Quality, Reliability, and Safety Engineering (in review)*, 2007.

[84] P. Bloch, Heinz, and F. Geitner, *Practical Machinery Management for Process Plants*, vol. 2: Machinery Failure Analysis and Troubleshooting, 2 ed. Houston, TX: Gulf Publishing Company, 1994.

[85] G. C. Montanari, G. Mazzanti, M. Cacciari, and J. C. Fothergill, "Optimum Estimators for the Weibull Distribution of Censored Data: Singularly-censored Tests," *IEEE Transactions on Dielectrics and Electrical Insulation*, vol. 4, pp. 462-469, 1997.

[86] Z. Yang and M. Xie, "Efficient Estimation of the Weibull Shape Parameter based on a Modified Profile Likelihood," *Journal of Statistical Compuation and Simulation*, vol. 73, pp. 115-123, 2003.

APPENDICES

Appendix A: Initial Estimation Selection Analysis

		n=5														
		Beta=2					Beta=3					Beta=4				
		MLEise	KMEise	FRWEise	PEXEise	KLMise	MLEise	KMEise	FRWEise	PEXEise	KLMise	MLEise	KMEise	FRWEise	PEXEise	KLMise
.6-.7	Mean	0.0750	0.1023	0.0716	0.0789	0.0968	0.0613	0.0730	0.0484	0.0646	0.0680	0.0378	0.0465	0.0306	0.0451	0.0428
	St Dev	0.1265	0.1343	0.0998	0.1182	0.1281	0.0872	0.0844	0.0620	0.0766	0.0809	0.0535	0.0495	0.0369	0.0492	0.0488
	Rank	3	5	1	2	4	1	4	1	2	3	1	4	1	3	2
.8-.9	Mean	0.1984	0.0861	0.0709	0.0930	0.0881	0.1399	0.0639	0.0449	0.1029	0.0660	0.1960	0.0846	0.0704	0.0901	0.0865
	St Dev	0.1862	0.1096	0.0756	0.1298	0.1150	0.1654	0.0923	0.0589	0.1163	0.0961	0.1851	0.1093	0.0747	0.1240	0.1144
	Rank	3	2	1	2	1	4	2	1	3	2	3	1	1	2	1

		n=6														
		Beta=2					Beta=3					Beta=4				
		MLEise	KMEise	FRWEise	PEXEise	KLMise	MLEise	KMEise	FRWEise	PEXEise	KLMise	MLEise	KMEise	FRWEise	PEXEise	KLMise
.6-.7	Mean	0.0627	0.0796	0.0599	0.0586	0.0744	0.0505	0.0589	0.0394	0.0503	0.0540	0.0326	0.0397	0.0258	0.0370	0.0354
	St Dev	0.0922	0.0994	0.0714	0.0758	0.0940	0.0754	0.0743	0.0505	0.0641	0.0700	0.0453	0.0458	0.0319	0.0431	0.0424
	Rank	3	5	2	1	4	1	4	1	2	3	1	4	1	3	2
.8-.9	Mean	0.1620	0.0729	0.0701	0.0679	0.0746	0.1068	0.0509	0.0396	0.0738	0.0529	0.0731	0.0348	0.0254	0.0805	0.0367
	St Dev	0.1696	0.0952	0.0629	0.0876	0.1000	0.1298	0.0773	0.0497	0.0816	0.0813	0.0827	0.0483	0.0307	0.0829	0.0518
	Rank	3	2	2	1	2	3	1	1	2	1	3	2	1	4	2

		n=8														
		Beta=2					Beta=3					Beta=4				
		MLEise	KMEise	FRWEise	PEXEise	KLMise	MLEise	KMEise	FRWEise	PEXEise	KLMise	MLEise	KMEise	FRWEise	PEXEise	KLMise
.6-.7	Mean	0.2055	0.2564	0.3453	0.1762	0.2304	0.0665	0.0845	0.1035	0.0678	0.0755	0.0322	0.0496	0.0366	0.0485	0.0429
	St Dev	0.2990	0.3438	0.2920	0.2572	0.3263	0.0798	0.0931	0.0917	0.0836	0.0863	0.0431	0.0486	0.0420	0.0510	0.0441
	Rank	2	4	5	1	3	1	4	5	2	3	1	5	2	4	3
.7-.8	Mean	0.1423	0.1721	0.4446	0.1025	0.1469	0.0451	0.0589	0.1208	0.0510	0.0503	0.0387	0.0496	0.0375	0.0464	0.0422
	St Dev	0.2277	0.2680	0.2881	0.1710	0.2470	0.0637	0.0725	0.0878	0.0716	0.0646	0.0464	0.0493	0.0402	0.0504	0.0433
	Rank	2	4	5	1	3	1	3	4	2	1	2	5	1	4	3
.8-.9	Mean	0.0841	0.0950	0.5656	0.0835	0.0807	0.0419	0.0307	0.1313	0.1140	0.0288	0.1028	0.0503	0.0441	0.1018	0.0536
	St Dev	0.2172	0.1883	0.2437	0.1534	0.2003	0.0586	0.0523	0.0746	0.1513	0.0556	0.1066	0.0629	0.0428	0.1007	0.0688
	Rank	3	4	5	2	1	3	2	5	4	1	4	2	1	3	2

		n=10														
		Beta=2					Beta=3					Beta=4				
		MLEise	KMEise	FRWEise	PEXEise	KLMise	MLEise	KMEise	FRWEise	PEXEise	KLMise	MLEise	KMEise	FRWEise	PEXEise	KLMise
.6-.7	Mean	0.0384	0.0539	0.0427	0.0431	0.0505	0.0299	0.0460	0.0354	0.0416	0.0410	0.0292	0.0398	0.0256	0.0413	0.0341
	St Dev	0.0527	0.0585	0.0475	0.0509	0.0571	0.0396	0.0445	0.0399	0.0457	0.0423	0.0382	0.0409	0.0298	0.0454	0.0371
	Rank	1	5	2	3	4	1	5	2	4	3	1	2	1	3	2
.7-.8	Mean	0.0424	0.0521	0.0481	0.0398	0.0476	0.0335	0.0452	0.0370	0.0397	0.0394	0.0290	0.0376	0.0256	0.0372	0.0318
	St Dev	0.0536	0.0575	0.0478	0.0474	0.0542	0.0403	0.0443	0.0398	0.0439	0.0405	0.0410	0.0434	0.0292	0.0452	0.0389
	Rank	2	4	3	1	3	1	5	2	4	3	3	4	1	5	2
.8-.9	Mean	0.0450	0.0485	0.0619	0.0352	0.0443	0.0397	0.0458	0.0439	0.0371	0.0399	0.0361	0.0438	0.0363	0.0402	0.0366
	St Dev	0.0669	0.0672	0.0539	0.0513	0.0632	0.0516	0.0527	0.0417	0.0444	0.0477	0.0454	0.0478	0.0366	0.0458	0.0419
	Rank	1	3	4	1	2	2	5	4	1	3	3	5	1	4	2
.9-1	Mean	0.0720	0.0376	0.0793	0.0297	0.0382	0.0766	0.0398	0.0564	0.0485	0.0412	0.0762	0.0392	0.0429	0.0691	0.0417
	St Dev	0.1014	0.0651	0.0546	0.0429	0.0672	0.0938	0.0592	0.0467	0.0503	0.0625	0.0860	0.0537	0.0383	0.0659	0.0582
	Rank	5	3	2	1	4	4	1	4	3	2	4	2	1	3	2

		n=15														
		Beta=2					Beta=3					Beta=4				
		MLEise	KMEise	FRWEise	PEXEise	KLMise	MLEise	KMEise	FRWEise	PEXEise	KLMise	MLEise	KMEise	FRWEise	PEXEise	KLMise
.6-.7	Mean	0.1100	0.1506	0.3149	0.1057	0.1255	0.0388	0.0516	0.0870	0.0403	0.0436	0.0204	0.0328	0.0218	0.0413	0.0263
	St Dev	0.1470	0.1655	0.2321	0.1686	0.1638	0.0407	0.0453	0.0682	0.0561	0.0435	0.0231	0.0274	0.0245	0.0409	0.0242
	Rank	1	3	5	4	2	1	4	5	2	3	1	4	2	5	3
.7-.8	Mean	0.0872	0.1216	0.3834	0.0806	0.0979	0.0299	0.0413	0.0981	0.0325	0.0344	0.0238	0.0339	0.0271	0.0385	0.0269
	St Dev	0.1327	0.1479	0.2443	0.1463	0.1442	0.0352	0.0401	0.0673	0.0486	0.0375	0.0284	0.0331	0.0280	0.0422	0.0281
	Rank	2	4	5	1	3	1	4	5	2	3	1	4	3	5	2
.8-.9	Mean	0.0455	0.0622	0.5412	0.0314	0.0458	0.0150	0.0215	0.1182	0.0173	0.0176	0.0280	0.0363	0.0304	0.0356	0.0288
	St Dev	0.0741	0.0847	0.2113	0.0706	0.0784	0.0220	0.0251	0.0563	0.0244	0.0219	0.0379	0.0420	0.0299	0.0438	0.0358
	Rank	2	5	4	1	3	1	4	5	2	3	1	5	3	4	2
.9-1	Mean	0.0229	0.0321	0.5964	0.0301	0.0209	0.0252	0.0125	0.1183	0.0332	0.0125	0.0435	0.0248	0.0436	0.0330	0.0260
	St Dev	0.0593	0.0530	0.1545	0.0498	0.0535	0.0223	0.0164	0.0559	0.0378	0.0185	0.0561	0.0385	0.0336	0.0303	0.0413
	Rank	2	4	5	3	1	3	1	5	4	2	4	1	5	3	2

		n=20														
		Beta=2					Beta=3					Beta=4				
		MLEise	KMEise	FRWEise	PEXEise	KLMise	MLEise	KMEise	FRWEise	PEXEise	KLMise	MLEise	KMEise	FRWEise	PEXEise	KLMise
.6-.7	Mean	0.0232	0.0320	0.0294	0.0319	0.0283	0.0165	0.0226	0.0156	0.0265	0.0187	0.0110	0.0162	0.0099	0.0228	0.0126
	St Dev	0.0276	0.0298	0.0308	0.0331	0.0289	0.0229	0.0250	0.0176	0.0309	0.0222	0.0143	0.0165	0.0114	0.0239	0.0139
	Rank	1	5	3	4	2	2	4	1	5	3	1	3	1	4	2
.7-.8	Mean	0.0246	0.0309	0.0388	0.0288	0.0267	0.0149	0.0198	0.0169	0.0216	0.0161	0.0106	0.0152	0.0106	0.0198	0.0116
	St Dev	0.0344	0.0364	0.0359	0.0365	0.0333	0.0215	0.0235	0.0174	0.0269	0.0205	0.0151	0.0174	0.0119	0.0231	0.0144
	Rank	1	4	5	3	2	1	4	3	4	2	2	4	1	5	3
.8-.9	Mean	0.0210	0.0248	0.0547	0.0209	0.0215	0.0130	0.0166	0.0210	0.0159	0.0134	0.0097	0.0132	0.0126	0.0146	0.0101
	St Dev	0.0357	0.0367	0.0423	0.0334	0.0340	0.0201	0.0220	0.0190	0.0222	0.0191	0.0148	0.0170	0.0121	0.0196	0.0142
	Rank	2	4	5	1	3	1	4	5	3	2	1	4	3	3	2
.9-1	Mean	0.0153	0.0156	0.0798	0.0110	0.0139	0.0104	0.0110	0.0303	0.0084	0.0093	0.0094	0.0100	0.0165	0.0084	0.0083
	St Dev	0.0294	0.0286	0.0443	0.0225	0.0270	0.0175	0.0170	0.0210	0.0137	0.0157	0.0154	0.0147	0.0132	0.0128	0.0135
	Rank	4	3	5	1	2	2	3	4	1	1	1	2	3	1	1

Appendix A (continued): Initial Estimation Selection Analysis

		n=25														
		Beta=2					Beta=3					Beta=4				
		MLEise	KMEise	FRWEise	PEXEise	KLMise	MLEise	KMEise	FRWEise	PEXEise	KLMise	MLEise	KMEise	FRWEise	PEXEise	KLMise
.6-.7	Mean	0.020647	0.028307	0.02675504	0.0302448	0.0244202	0.013573	0.0191203	0.01325738	0.0247811	0.0153808	0.009182	0.0140245	0.0084783	0.022389	0.010482
	St Dev	0.024442	0.026388	0.02678505	0.0309316	0.0246725	0.018291	0.0201326	0.01441355	0.0268333	0.0176922	0.011833	0.0141333	0.0097115	0.0228892	0.0115547
	Rank	1	4	3	4	2	2	4	1	5	3	1	3	1	4	2
.7-.8	Mean	0.019958	0.025138	0.03901861	0.0243856	0.0212693	0.011578	0.0159449	0.01570991	0.0185129	0.0125195	0.008477	0.0124704	0.0093268	0.0176844	0.0091983
	St Dev	0.029729	0.031579	0.03338719	0.0325897	0.0282826	0.016991	0.0190797	0.01555229	0.0227224	0.0163718	0.011697	0.0138982	0.0097074	0.0198566	0.0112829
	Rank	1	4	5	3	2	1	3	3	4	2	1	4	3	5	2
.8-.9	Mean	0.016454	0.020011	0.05238397	0.0178305	0.0169448	0.009969	0.0131902	0.01999854	0.0135695	0.0103748	0.007893	0.0111823	0.0114213	0.0133928	0.0082419
	St Dev	0.027727	0.029033	0.03855994	0.0279882	0.0264954	0.015254	0.0168202	0.01710252	0.0182021	0.0144977	0.012162	0.0142665	0.0108232	0.0173316	0.011704
	Rank	1	3	4	2	2	1	3	5	4	2	1	3	4	4	2
.9-1	Mean	0.009712	0.00997	0.08491585	0.0068791	0.008642	0.007252	0.0077227	0.03215622	0.0057487	0.0064269	0.006419	0.0070095	0.0175109	0.0058013	0.0056904
	St Dev	0.019266	0.018884	0.03852355	0.0152485	0.0176355	0.012825	0.012416	0.01926152	0.0102875	0.0114413	0.010616	0.0106486	0.0122491	0.0094359	0.0094779
	Rank	4	5	2	1	3	1	2	3	1	1	1	2	3	1	1

		n=30														
		Beta=2					Beta=3					Beta=4				
		MLEise	KMEise	FRWEise	PEXEise	KLMise	MLEise	KMEise	FRWEise	PEXEise	KLMise	MLEise	KMEise	FRWEise	PEXEise	KLMise
.6-.7	Mean	0.018002	0.024948	0.02470402	0.028173	0.021111	0.01091	0.0160273	0.01141231	0.0225253	0.0124462	0.007795	0.0121579	0.0073506	0.0217936	0.0089015
	St Dev	0.021758	0.023782	0.02416676	0.0293633	0.0221104	0.014659	0.0166912	0.01237187	0.0235469	0.0143204	0.009764	0.0117865	0.0080738	0.0209329	0.0095907
	Rank	1	4	3	5	2	1	4	2	5	3	2	4	1	5	3
.7-.8	Mean	0.017356	0.022628	0.03410705	0.0235334	0.0187467	0.010258	0.0145546	0.01361385	0.018642	0.0111277	0.007515	0.0115634	0.0080138	0.0181662	0.0081658
	St Dev	0.025838	0.027685	0.0293926	0.0304267	0.0247257	0.014738	0.016704	0.01347501	0.0216337	0.0142752	0.010013	0.0123737	0.008253	0.0186767	0.0096457
	Rank	1	2	5	4	2	1	4	3	5	2	1	3	2	4	2
.8-.9	Mean	0.012937	0.016213	0.05106991	0.0150406	0.0133897	0.008306	0.0113918	0.01899658	0.0124679	0.0086806	0.006532	0.0095804	0.0105049	0.0126238	0.0068649
	St Dev	0.02167	0.022959	0.03546445	0.0228961	0.0207066	0.013217	0.0148297	0.01540221	0.0170139	0.0126667	0.009867	0.0118665	0.0095183	0.0153563	0.0095428
	Rank	1	4	5	3	2	1	3	4	3	2	1	3	4	4	2
.9-1	Mean	0.008142	0.009596	0.07761005	0.0071745	0.0079829	0.005768	0.0072794	0.0289169	0.0061824	0.0056996	0.005203	0.0068442	0.0153815	0.0065677	0.0051157
	St Dev	0.015514	0.016155	0.03704289	0.0139117	0.0147702	0.01008	0.0108691	0.01766735	0.0105115	0.0096629	0.008611	0.0094658	0.0108931	0.0096891	0.0081654
	Rank	3	4	5	1	2	2	4	5	3	1	2	4	5	3	1

		n=35														
		Beta=2					Beta=3					Beta=4				
		MLEise	KMEise	FRWEise	PEXEise	KLMise	MLEise	KMEise	FRWEise	PEXEise	KLMise	MLEise	KMEise	FRWEise	PEXEise	KLMise
.6-.7	Mean	0.022837	0.032104	0.02478715	0.0313624	0.0308916	0.009482	0.0147236	0.01077164	0.0170244	0.0139207	0.006344	0.0096908	0.0071372	0.0128705	0.0091548
	St Dev	0.026389	0.0277	0.02601286	0.0286175	0.0278334	0.010133	0.0110505	0.01033694	0.0126323	0.0111174	0.007277	0.0077792	0.0073124	0.0106415	0.0077901
	Rank	1	3	2	3	3	1	4	2	5	3	1	4	2	5	3
.7-.8	Mean	0.022243	0.031476	0.02398846	0.0311234	0.0301342	0.009795	0.015087	0.01114674	0.0191558	0.0140221	0.005839	0.0090422	0.0066411	0.0129029	0.0083432
	St Dev	0.024159	0.026054	0.02432263	0.0280682	0.0261744	0.011218	0.0123307	0.0114162	0.0160068	0.0121354	0.007007	0.0072183	0.0068209	0.0098381	0.0072092
	Rank	1	3	2	3	3	1	4	2	5	3	1	4	2	5	3
.8-.9	Mean	0.023277	0.031927	0.02483957	0.0320981	0.0303493	0.009775	0.0149056	0.01075772	0.0195017	0.0137255	0.006027	0.0093044	0.0067589	0.0149645	0.0085018
	St Dev	0.029864	0.029971	0.02790052	0.032479	0.0301002	0.011194	0.011984	0.01131678	0.0151398	0.0121002	0.007287	0.0077246	0.0071992	0.0117275	0.0076638
	Rank	1	3	2	3	3	1	4	2	5	3	1	4	2	5	3
.9-1	Mean	0.021849	0.030641	0.02422734	0.0310086	0.0289205	0.009681	0.01489	0.01052219	0.0203669	0.0136087	0.006273	0.0097136	0.0070978	0.0163927	0.008719
	St Dev	0.025195	0.026959	0.02539105	0.0296965	0.0271726	0.012072	0.0130098	0.01173561	0.0175472	0.0129696	0.006807	0.0074501	0.0070296	0.0125688	0.0072808
	Rank	1	3	2	3	3	1	4	2	5	3	1	4	2	5	3

		n=40														
		Beta=2					Beta=3					Beta=4				
		MLEise	KMEise	FRWEise	PEXEise	KLMise	MLEise	KMEise	FRWEise	PEXEise	KLMise	MLEise	KMEise	FRWEise	PEXEise	KLMise
.6-.7	Mean	0.019258	0.028094	0.02140267	0.0274713	0.0271071	0.008551	0.0131744	0.00986824	0.015687	0.0125097	0.005126	0.0079826	0.0059795	0.010781	0.007563
	St Dev	0.021229	0.023615	0.02196411	0.0244746	0.0235643	0.010126	0.0109427	0.01028421	0.0126235	0.0109744	0.0058	0.006325	0.0059859	0.0078457	0.0063342
	Rank	1	3	2	3	3	1	4	2	5	3	1	4	2	5	3
.7-.8	Mean	0.020161	0.028787	0.0219845	0.0284839	0.0276273	0.00865	0.0131766	0.00976408	0.016102	0.0123469	0.005098	0.0079473	0.0058224	0.0116433	0.0074149
	St Dev	0.021388	0.022485	0.02121311	0.0244448	0.0227206	0.009808	0.0104326	0.00994585	0.0126207	0.0103394	0.005715	0.0060134	0.005763	0.0086948	0.0060184
	Rank	1	3	2	3	3	1	4	2	5	3	1	4	2	5	3
.8-.9	Mean	0.019379	0.027875	0.02140126	0.0284052	0.0265231	0.008777	0.0135063	0.00996902	0.0173866	0.0125097	0.005089	0.0079947	0.005857	0.0127224	0.0073839
	St Dev	0.020429	0.02154	0.02035806	0.0239894	0.0217242	0.010448	0.0112548	0.01052833	0.0135957	0.011324	0.005571	0.0060483	0.0056538	0.0093792	0.0060072
	Rank	1	3	2	3	3	1	4	2	5	3	1	4	2	5	3
.9-1	Mean	0.019581	0.027355	0.02137764	0.0279495	0.0258759	0.008579	0.0131159	0.00957233	0.0179118	0.0119929	0.005259	0.0082148	0.0060238	0.0142334	0.0074578
	St Dev	0.022818	0.023754	0.02235748	0.0269376	0.0239125	0.010179	0.0109929	0.01026518	0.0144926	0.0108881	0.005963	0.0064271	0.0061626	0.0107712	0.0063658
	Rank	1	3	2	3	3	1	4	2	5	3	1	4	2	5	3

		n=50														
		Beta=2					Beta=3					Beta=4				
		MLEise	KMEise	FRWEise	PEXEise	KLMise	MLEise	KMEise	FRWEise	PEXEise	KLMise	MLEise	KMEise	FRWEise	PEXEise	KLMise
.6-.7	Mean	0.015943	0.023855	0.0184906	0.0240908	0.0231722	0.006611	0.0102313	0.00785067	0.0123045	0.0098298	0.00429	0.0066253	0.0051298	0.0089877	0.0063263
	St Dev	0.018209	0.020126	0.01922453	0.0210933	0.0202291	0.007403	0.0079487	0.00761908	0.0093503	0.0080183	0.004767	0.0051267	0.0049647	0.0064098	0.0051475
	Rank	1	3	2	3	3	1	4	2	5	3	1	4	2	5	3
.7-.8	Mean	0.017143	0.024759	0.019337	0.0252161	0.023865	0.007266	0.011051	0.00840277	0.01382	0.0104811	0.004241	0.0066447	0.0050674	0.0099403	0.0062914
	St Dev	0.018913	0.019861	0.01882568	0.0207372	0.0198613	0.008122	0.0085098	0.00815815	0.010112	0.0085619	0.004757	0.0052224	0.0049455	0.0071244	0.0052205
	Rank	1	3	2	3	3	1	4	2	5	3	1	4	2	5	3
.8-.9	Mean	0.016566	0.023886	0.01844379	0.0244393	0.0229427	0.007355	0.0112019	0.00843334	0.0145453	0.0105358	0.004298	0.0066105	0.0050183	0.0107553	0.0061819

Appendix A (continued): Initial Estimation Selection Analysis

		n=60														
		Beta=2					Beta=3					Beta=4				
		MLEise	KMEise	FRWEise	PEXEise	KLMise	MLEise	KMEise	FRWEise	PEXEise	KLMise	MLEise	KMEise	FRWEise	PEXEise	KLMise
.6-.7	**Mean**	0.014312	0.020501	0.01627139	0.0206335	0.0199969	0.005758	0.0088604	0.00688883	0.0104718	0.0085869	0.003524	0.005444	0.0042866	0.0073029	0.0052298
	St Dev	0.017085	0.018081	0.01741973	0.0181274	0.0181491	0.006914	0.0073066	0.00701781	0.0079702	0.0073497	0.003816	0.0041023	0.0040381	0.0049567	0.0041096
	Rank	1	3	2	3	3	1	3	2	4	3	1	3	2	4	3
.7-.8	**Mean**	0.013993	0.020573	0.01612016	0.0212413	0.019898	0.00582	0.0090403	0.00706286	0.0115108	0.0086553	0.003853	0.0057265	0.0045139	0.0082915	0.0054656
	St Dev	0.015797	0.016764	0.0162153	0.0177713	0.016795	0.006418	0.0071173	0.00688582	0.0086258	0.0071383	0.004719	0.004848	0.0047031	0.0060596	0.0048544
	Rank	1	3	2	3	3	1	4	2	5	3	1	4	2	5	3
.8-.9	**Mean**	0.014442	0.02116	0.01696192	0.0225981	0.0205232	0.005785	0.0089167	0.0068596	0.0119186	0.0084384	0.003746	0.0057426	0.0044527	0.0090056	0.0054614
	St Dev	0.016779	0.01848	0.01776915	0.0200319	0.0185386	0.006278	0.0068767	0.00652923	0.0085138	0.0068715	0.004423	0.0047249	0.00457	0.006699	0.0047533
	Rank	1	3	2	3	3	1	4	2	5	3	1	4	2	5	3
.9-1	**Mean**	0.014349	0.020636	0.01629828	0.0222302	0.0198007	0.014082	0.0203379	0.01622096	0.0221148	0.0195311	0.003389	0.0054226	0.0041116	0.0097515	0.0050452
	St Dev	0.016316	0.016864	0.0162539	0.0186293	0.0168896	0.015685	0.0168889	0.01620009	0.0189669	0.0169214	0.003847	0.0042112	0.003971	0.0065968	0.0042137
	Rank	1	3	2	3	3	1	4	2	4	3	1	4	2	5	3

		n=80														
		Beta=2					Beta=3					Beta=4				
		MLEise	KMEise	FRWEise	PEXEise	KLMise	MLEise	KMEise	FRWEise	PEXEise	KLMise	MLEise	KMEise	FRWEise	PEXEise	KLMise
.6-.7	**Mean**	0.010032	0.015058	0.01222835	0.0156427	0.014737	0.004403	0.0068699	0.00551307	0.0081297	0.0066891	0.002567	0.0039605	0.0031766	0.005271	0.0038463
	St Dev	0.011439	0.012319	0.01208807	0.0125873	0.0123587	0.004894	0.005263	0.00513964	0.0056652	0.0052694	0.002887	0.0030947	0.0029987	0.0036951	0.0031046
	Rank	1	3	2	3	3	1	3	2	4	3	1	3	2	4	3
.7-.8	**Mean**	0.010163	0.015283	0.0121591	0.0160244	0.0148586	0.004122	0.0065481	0.00514251	0.0081767	0.0063296	0.002668	0.0041172	0.0032927	0.0058682	0.0039789
	St Dev	0.01129	0.012186	0.01167096	0.0127259	0.0122004	0.004997	0.0053436	0.00518764	0.0059483	0.0053482	0.002862	0.0031071	0.0029868	0.0038619	0.0031053
	Rank	1	3	2	3	3	1	3	2	4	3	1	2	3	4	5
.8-.9	**Mean**	0.010784	0.0157	0.01256925	0.0169887	0.0151998	0.004451	0.006782	0.00534435	0.0090421	0.0065152	0.010693	0.0156444	0.0126936	0.0171061	0.0152082
	St Dev	0.011991	0.012825	0.01231857	0.0136017	0.0128225	0.005374	0.0059777	0.00573638	0.0069168	0.0059949	0.011802	0.0126076	0.012314	0.0137134	0.012658
	Rank	1	3	2	4	3	1	4	2	5	3	1	3	2	4	3
.9-1	**Mean**	0.010263	0.015221	0.01206688	0.0168057	0.0146811	0.010228	0.0153342	0.01228874	0.0170549	0.0148332	0.010614	0.0157588	0.0126084	0.0175955	0.0151984
	St Dev	0.011585	0.01227	0.01171826	0.0135168	0.0122873	0.011192	0.0118138	0.0114715	0.0125782	0.0118879	0.012865	0.0135953	0.0130441	0.0146347	0.0135931
	Rank	1	3	2	4	3	1	3	2	4	3	1	3	2	4	3

		n=100														
		Beta=2					Beta=3					Beta=4				
		MLEise	KMEise	FRWEise	PEXEise	KLMise	MLEise	KMEise	FRWEise	PEXEise	KLMise	MLEise	KMEise	FRWEise	PEXEise	KLMise
.6-.7	**Mean**	0.008092	0.012379	0.01028929	0.0128844	0.012156	0.003416	0.0054007	0.00437207	0.0063167	0.0052806	0.002127	0.0032958	0.0027297	0.00431	0.0032151
	St Dev	0.009111	0.01026	0.00998485	0.0104741	0.0102708	0.004345	0.0045543	0.00445256	0.0046968	0.004555	0.002645	0.0027325	0.0027373	0.0031931	0.0027325
	Rank	1	3	2	3	3	1	3	2	4	3	1	3	2	4	3
.7-.8	**Mean**	0.008817	0.01308	0.0107134	0.0138992	0.0128155	0.008774	0.0129305	0.01071988	0.0138171	0.0126471	0.00208	0.0032739	0.0027221	0.0047082	0.0031781
	St Dev	0.010299	0.011037	0.01089948	0.0115958	0.0110658	0.010837	0.0111506	0.01111268	0.0115716	0.0111653	0.00228	0.0024844	0.0024016	0.0029516	0.0024936
	Rank	1	3	2	4	3	1	3	2	4	3	1	3	2	4	3
.8-.9	**Mean**	0.008517	0.012888	0.01063833	0.0140348	0.012574	0.008814	0.0129252	0.01066286	0.0140859	0.0125787	0.008421	0.0125661	0.0101726	0.0135474	0.0122496
	St Dev	0.009433	0.010353	0.00991519	0.0109411	0.010374	0.010935	0.0112066	0.01113956	0.0119757	0.0112173	0.009664	0.01061	0.010429	0.0109284	0.0106093
	Rank	1	3	2	4	3	1	3	2	4	3	1	3	2	4	3

Appendix B: Weibull Parameter Database

	Weibull Shape			Weibull Scale Eta (hours)		
Component	**Low**	**Typical**	**High**	**Low**	**Typical**	**High**
Ball bearing	0.7	1.3	3.5	14,000	40,000	250,000
Roller bearings	0.7	1.3	3.5	9,000	50,000	125,000
Sleeve bearing	0.7	1	3	10,000	50,000	143,000
Belts, drive	0.5	1.2	2.8	9,000	30,000	91,000
Bellows, hydraulic	0.5	1.3	3	14,000	50,000	100,000
Bolts	0.5	3	10	125,000	300,000	100,000,000
Clutches, friction	0.5	1.4	3	67,000	100,000	500,000
Clutches, magnetic	0.8	1	1.6	100,000	150,000	333,000
Couplings	0.8	2	6	25,000	75,000	333,000
Couplings, gear	0.8	2.5	4	25,000	75,000	1,250,000
Cylinders, hydraulic	1	2	3.8	9,000,000	900,000	200,000,000
Diaphragm, metal	0.5	3	6	50,000	65,000	500,000
Diaphragm, rubber	0.5	1.1	1.4	50,000	60,000	300,000
Gaskets, hydraulics	0.5	1.1	1.4	700,000	75,000	3,300,000
Filter, oil	0.5	1.1	1.4	20,000	25,000	125,000
Gears	0.5	2	6	33,000	75,000	500,000
Impellers, pumps	0.5	2.5	6	125,000	150,000	1,400,000
Joints, mechanical	0.5	1.2	6	1,400,000	150,000	10,000,000
Knife edges, fulcrum	0.5	1	6	1,700,000	2,000,000	16,700,000
Liner, recip. comp. cyl.	0.5	1.8	3	20,000	50,000	300,000
Nuts	0.5	1.1	1.4	14,000	50,000	500,000
"O"-rings, elastomeric	0.5	1.1	1.4	5,000	20,000	33,000
Packings, recip. comp. rod	0.5	1.1	1.4	5,000	20,000	33,000
Pins	0.5	1.4	5	17,000	50,000	170,000
Pivots	0.5	1.4	5	300,000	400,000	1,400,000
Pistons, engines	0.5	1.4	3	20,000	75,000	170,000
Pumps, lubricators	0.5	1.1	1.4	13,000	50,000	125,000
Seals, mechanical	0.8	1.4	4	3,000	25,000	50,000
Shafts, cent. pumps	0.8	1.2	3	50,000	50,000	300,000
Springs	0.5	1.1	3	14,000	25,000	5,000,000
Vibration mounts	0.5	1.1	2.2	17,000	50,000	200,000
Wear rings, cent. pumps	0.5	1.1	4	10,000	50,000	90,000
Valves, recip comp.	0.5	1.4	4	3,000	40,000	80,000
Machinery Equipment						
Circuit breakers	0.5	1.5	3	67,000	100,000	1,400,000
Compressors, centrifugal	0.5	1.9	3	20,000	60,000	120,000
Compressor blades	0.5	2.5	3	400,000	800,000	1,500,000
Compressor vanes	0.5	3	4	500,000	1,000,000	2,000,000
Diaphgram couplings	0.5	2	4	125,000	300,000	600,000
Gas turb. comp. blades/vanes	1.2	2.5	6.6	10,000	250,000	300,000
Gas turb. blades/vanes	0.9	1.6	2.7	10,000	125,000	160,000
Motors, AC	0.5	1.2	3	1,000	100,000	200,000
Motors, DC	0.5	1.2	3	100	50,000	100,000

Appendix C: Final Estimation Selection Analysis

n=5							
Censored	Beta Values						
Number	**1-1.45**	**1.5-1.95**	**2-2.45**	**2.5-2.95**	**3-3.45**	**3.5-4**	**4-4.45**
1	MPMLE*	FRWE	FRWE*	FRWE*	FRWE*	WH*	WH*
2	MLE	MLE	MLE	MLE	MLE*	MLE*	MLE
3	PEXE	PEXE	PEXE*	FRWE	MLE*	MLE*	MLE*
4	PEXE	FRWE*	FRWE	FRWE	FRWE	FRWE	FRWE

n=10							
Censored	Beta Values						
Number	**1-1.45**	**1.5-1.95**	**2-2.45**	**2.5-2.95**	**3-3.45**	**3.5-4**	**4-4.45**
1	MPMLE*	WH*	WH*	WH*	WH*	WH*	WH*
2	MPMLE*	BE*	BE	BE*	WH*	WH*	WH*
3	MPMLE*	MPMLE*	MPMLE*	ROSS*	ROSS*	ROSS*	ROSS*
4	MLE	MLE*	MLE*	MLE*	MLE*	MLE*	MLE*
5	MLE	MLE	MLE	MLE*	MLE	MLE	MLE
6	MLE	MLE	MLE	MLE	MLE	MLE	MLE
7	PEXE	PEXE	PEXE*	FRWE	FRWE	FRWE	FRWE
8	PEXE	PEXE	PEXE	PEXE	PEXE*	FRWE	FRWE
9	PEXE	PEXE*	KME	KME	KME	KME	KME

n=15							
Censored	Beta Values						
Number	**1-1.45**	**1.5-1.95**	**2-2.45**	**2.5-2.95**	**3-3.45**	**3.5-4**	**4-4.45**
1	MPMLE*	MPMLE*	MPMLE*	MPMLE*	MPMLE*	MPMLE*	MPMLE*
2	MPMLE*	MPMLE*	WH*	WH*	ROSS*	ROSS*	ROSS*
3	ROSS*	ROSS*	ROSS*	ROSS*	ROSS*	ROSS*	ROSS*
4	ROSS*	ROSS*	ROSS*	ROSS*	ROSS*	ROSS*	ROSS*
5	MPMLE*	MPMLE*	ROSS*	ROSS*	ROSS*	ROSS*	ROSS*
6	MLE*	MPMLE*	MPMLE*	MPMLE*	MPMLE*	MPMLE*	MPMLE*
7	MLE	MLE	MLE	MLE	MLE	MLE	MLE
8	MLE	MLE	MLE	MLE	MLE	MLE	MLE
9	MLE	MLE	MLE	MLE	MLE	MLE	MLE
10	MLE	MLE	MLE	MLE	MLE	MLE	MLE
11	MLE*	MLE	MLE	MLE	MLE	MLE	MLE*
12	PEXE	PEXE	MLE*	MLE	MLE	MLE	MLE
13	PEXE	PEXE	PEXE	PEXE	PEXE	PEXE	PEXE*
14	PEXE	KME	KME	KME	KME	KME	KME

Appendix C (continued): Final Estimation Selection Analysis

n=20							
Censored	Beta Values						
Number	**1-1.45**	**1.5-1.95**	**2-2.45**	**2.5-2.95**	**3-3.45**	**3.5-4**	**4-4.45**
1	MPMLE*	MPMLE*	MPMLE*	MPMLE*	MPMLE*	MPMLE*	MPMLE*
2	MPMLE*	MPMLE*	ROSS*	ROSS*	ROSS*	ROSS*	ROSS*
3	ROSS*	ROSS*	ROSS*	ROSS*	ROSS*	ROSS*	ROSS*
4	ROSS*	ROSS*	ROSS*	ROSS*	ROSS*	ROSS*	ROSS*
5	ROSS*	ROSS*	ROSS*	ROSS*	ROSS*	ROSS*	ROSS*
6	ROSS*	ROSS*	ROSS*	ROSS*	ROSS*	ROSS*	ROSS*
7	MPMLE*	ROSS*	ROSS*	ROSS*	ROSS*	ROSS*	ROSS*
8	MLE*	MPMLE*	MPMLE*	ROSS*	ROSS*	ROSS*	ROSS*
9	MLE	MLE	MLE*	MLE*	MPMLE*	MPMLE*	MPMLE*
10	MLE	MLE	MLE	MLE	MLE	MLE	MLE
11	MLE	MLE	MLE	MLE	MLE	MLE	MLE
12	MLE	MLE	MLE	MLE	MLE	MLE	MLE
13	MLE	MLE	MLE	MLE	MLE	MLE	MLE
14	MLE	MLE	MLE	MLE	MLE	MLE	MLE
15	MLE	MLE	MLE	MLE	MLE	MLE	MLE
16	MLE	MLE	MLE	MLE	MLE	MLE	MLE
17	PEXE	PEXE*	MLE*	MLE	MLE	MLE	MLE
18	PEXE	PEXE	PEXE	PEXE	PEXE	PEXE	PEXE*
19	PEXE	PEXE	KME	KME	KME	KME	KME

n=30							
Censored	Beta Values						
Number	**1-1.45**	**1.5-1.95**	**2-2.45**	**2.5-2.95**	**3-3.45**	**3.5-4**	**4-4.45**
1	MPMLE*	MPMLE*	MPMLE*	MPMLE*	MPMLE*	MPMLE*	MPMLE*
2	MPMLE*	ROSS*	MPMLE*	ROSS*	MPMLE*	MPMLE*	ROSS*
3	ROSS*	ROSS*	ROSS*	ROSS*	ROSS*	ROSS*	ROSS*
4	ROSS*	ROSS*	ROSS*	ROSS*	ROSS*	ROSS*	ROSS*
5	ROSS*	ROSS*	ROSS*	ROSS*	ROSS*	ROSS*	ROSS*
6	ROSS*	ROSS*	ROSS*	ROSS*	ROSS*	ROSS*	ROSS*
7	ROSS*	ROSS*	ROSS*	ROSS*	ROSS*	ROSS*	ROSS*
8	ROSS*	ROSS*	ROSS*	ROSS*	ROSS*	ROSS*	ROSS*
9	ROSS*	ROSS*	ROSS*	ROSS*	ROSS*	ROSS*	ROSS*
10	ROSS*	ROSS*	ROSS*	ROSS*	ROSS*	ROSS*	ROSS*
11	MPMLE*	ROSS*	ROSS*	ROSS*	ROSS*	ROSS*	ROSS*
12	MPMLE*	MPMLE*	MPMLE*	ROSS*	ROSS*	ROSS*	ROSS*
13	MLE	MLE*	MPMLE*	MPMLE*	ROSS*	ROSS*	ROSS*
14	MLE	MLE	MLE*	MLE*	MPMLE*	MPMLE*	MPMLE*
15	MLE	MLE	MLE	MLE	MLE	MLE*	MLE*
16	MLE	MLE	MLE	MLE	MLE	MLE	MLE
17	MLE	MLE	MLE	MLE	MLE	MLE	MLE
18	MLE	MLE	MLE	MLE	MLE	MLE	MLE
19	MLE	MLE	MLE	MLE	MLE	MLE	MLE
20	MLE	MLE	MLE	MLE	MLE	MLE	MLE
21	MLE	MLE	MLE	MLE	MLE	MLE	MLE
22	MLE	MLE	MLE	MLE	MLE	MLE	MLE
23	MLE	MLE	MLE	MLE	MLE	MLE	MLE
24	MLE	MLE	MLE	MLE	MLE	MLE	MLE
25	MLE	MLE	MLE	MLE	MLE	MLE	MLE
26	MLE	MLE	MLE	MLE	MLE	MLE	MLE
27	PEXE	PEXE	PEXE*	MLE*	MLE*	MLE*	MLE*
28	PEXE	PEXE	PEXE	PEXE	PEXE	PEXE	PEXE*
29	PEXE	PEXE	PEXE	KME	KME	KME	KME

Appendix D: Kabir and Al-Olayan Comparison for β=*1.5*

Policy (P)		P1 Fixed	P1 Normal	P1 Uniform			P2 Fixed	P2 Normal	P2 Uniform			P3 Fixed	P3 Normal	P3 Uniform		
Run	t^*	Δ	Δ	Δ	s	S	Δ	Δ	Δ	s	S	Δ	Δ	Δ	s	S
1	181.7	13%	13%	16%	1.3-2.4	4.8-6.7	5%	6%	9%	0.7-1.2	4.3-5.8	10%	7%	10%	1.5-2.6	4.3-6.1
2	181.7	-5%	-5%	-9%	0.7-1.8	4.2-6.8	-7%	-4%	-9%	0.4-0.9	4.0-6.2	-11%	-10%	-13%	0.7-1.8	3.6-6.0
3	181.7	35%	37%	35%	1.1-1.9	3.6-5.1	24%	34%	30%	0.5-1.3	3.0-4.8	34%	38%	34%	1.5-2.1	3.3-4.8
4	181.7	28%	25%	25%	1.8-2.3	6.4-7.5	21%	18%	15%	1.1-1.6	5.8-7.2	22%	20%	20%	1.7-2.2	5.7-6.8
5	154.6	7%	6%	6%	0.4-1.6	3.9-6.9	1%	1%	0%	0.3-0.9	3.8-6.5	4%	4%	5%	0.6-2.1	3.5-6.7
6	181.7	37%	39%	38%	1.8-2.8	3.9-5.2	22%	27%	26%	1.1-1.5	3.3-4.3	17%	20%	21%	1.6-2.2	3.0-3.9
7	412.4	16%	17%	16%	1.2-3.0	4.4-7.1	11%	12%	11%	0.6-1.2	4.1-5.6	10%	9%	9%	1.0-2.7	3.6-5.9
8	99.8	1%	4%	3%	1.7-2.8	5.1-7.0	-7%	-5%	-4%	0.8-1.2	4.5-6.0	-3%	0%	-1%	1.5-2.5	4.3-5.9
9	412.4	45%	44%	43%	1.5-2.5	3.6-5.0	31%	28%	25%	1.0-1.5	3.3-4.2	36%	33%	32%	1.6-2.4	3.2-4.2
10	154.6	33%	30%	30%	0.8-1.6	2.8-4.3	22%	24%	20%	0.5-1.0	2.7-4.0	32%	28%	28%	1.2-2.0	2.6-4.0
11	154.6	18%	22%	16%	2.2-3.5	5.5-7.5	10%	17%	8%	1.2-1.8	4.7-6.1	13%	18%	10%	1.9-2.9	4.6-6.1
12	412.4	22%	25%	23%	1.2-2.6	3.6-5.9	8%	9%	12%	0.8-1.5	3.3-5.0	11%	12%	14%	1.1-2.2	3.0-4.8
13	99.8	29%	31%	28%	3.1-7.1	6.6-11.3	11%	12%	11%	1.7-2.6	5.4-7.2	12%	16%	10%	2.6-5.3	5.4-8.3
14	154.6	1%	2%	1%	1.6-2.4	6.4-8.3	-4%	-1%	-2%	0.8-1.2	5.8-7.5	0%	1%	0%	1.6-2.3	5.7-7.4
15	154.6	18%	21%	19%	0.6-1.3	3.5-4.8	16%	21%	17%	0.5-0.8	3.6-4.5	23%	26%	23%	1.0-2.0	3.4-4.9
16	154.6	21%	18%	20%	1.2-1.9	3.5-4.6	13%	9%	11%	0.7-1.0	3.2-4.0	12%	10%	13%	1.2-1.9	2.9-3.8
17	164.6	27%	23%	21%	0.9-2.3	4.6-7.4	17%	16%	13%	0.6-1.2	4.4-6.5	25%	24%	20%	1.2-3.0	4.3-7.2
18	164.6	18%	21%	23%	1.3-3.0	2.8-5.1	7%	13%	15%	0.7-1.5	2.4-4.1	6%	11%	15%	1.2-2.2	2.1-3.7
19	164.6	9%	7%	9%	2.3-4.1	5.3-7.6	-2%	-4%	-2%	1.2-1.8	4.3-5.7	-4%	-7%	-4%	1.8-2.9	4.0-5.6
20	164.6	20%	21%	18%	0.6-1.7	3.1-5.3	17%	18%	14%	0.2-1.0	2.8-4.9	33%	31%	31%	1.2-2.6	3.2-5.6
21	164.6	18%	18%	20%	0.6-1.3	2.3-3.8	8%	12%	15%	0.3-0.8	2.0-3.5	17%	17%	20%	1.0-1.7	2.1-3.5
22	164.6	9%	11%	10%	2.4-4.2	8.8-12.1	4%	5%	7%	1.2-1.9	7.9-10.1	3%	5%	7%	1.6-3.3	7.3-10.2
23	296.8	21%	22%	25%	1.2-2.9	3.4-6.5	15%	12%	17%	0.6-1.6	2.8-5.4	15%	13%	18%	1.2-2.5	2.7-5.3
24	99.7	23%	27%	26%	2.1-3.1	5.2-6.8	12%	11%	13%	1.2-1.8	4.5-5.9	12%	16%	13%	1.8-2.6	4.3-5.7
25	102.1	16%	21%	20%	1.2-2.0	4.5-5.7	12%	15%	13%	0.7-1.0	4.1-5.0	10%	14%	15%	1.3-2.2	4.0-5.1
26	872.6	11%	12%	11%	1.1-1.7	4.2-5.2	3%	10%	4%	0.7-0.9	4.0-4.6	10%	11%	10%	1.2-1.9	3.7-4.7
27	164.6	26%	29%	25%	1.7-2.8	4.8-6.4	18%	20%	17%	1.1-1.6	4.4-5.5	19%	21%	16%	1.5-2.4	4.1-5.3
Avg Difference=		19%	20%	19%			11%	12%	11%			14%	14%	14%		

	P4 Fixed	P4 Normal	P4 Uniform			P5 Fixed	P5 Normal	P5 Uniform			P6 Fixed	P6 Uniform	P6 Normal			P7 Normal		
Run	Δ	Δ	Δ	s	S	Δ	Δ	Δ	s	S	Δ	Δ	Δ	s	S	Δ	s	S
1	17%	16%	41%	1.3-1.9	5.6-6.7	12%	11%	31%	0.8-1.1	5.4-6.3	11%	39%	13%	1.0-1.4	5.4-6.3	12%	1.3-2.1	5.3-6.6
2	0%	0%	27%	1.4-2.0	7.2-8.7	-3%	0%	21%	0.8-1.1	7.0-8.3	-5%	16%	-7%	0.8-1.1	6.1-7.3	-1%	1.4-2.1	6.9-8.4
3	59%	61%	80%	2.0-2.9	5.8-7.3	48%	54%	62%	1.7-2.2	5.8-7.0	46%	71%	54%	1.9-2.6	5.3-6.6	48%	1.7-3.2	5.4-7.4
4	37%	34%	49%	2.0-2.9	7.7-9.6	32%	31%	44%	1.6-2.2	7.8-9.3	28%	37%	24%	1.5-2.1	6.8-8.4	29%	1.9-3.0	7.5-9.2
5	11%	12%	23%	1.3-1.8	7.0-8.5	9%	9%	17%	0.8-1.1	7.0-8.2	4%	13%	5%	0.7-1.0	6.0-7.2	10%	1.2-2.1	6.7-8.5
6	38%	42%	96%	2.2-3.0	5.0-6.2	28%	41%	78%	1.7-2.2	4.7-5.7	28%	100%	36%	2.2-2.9	4.4-5.4	45%	2.1-4.2	4.8-7.2
7	21%	22%	66%	1.5-2.1	5.7-7.1	17%	21%	56%	0.8-1.2	5.4-6.6	13%	71%	18%	1.1-1.6	4.8-6.0	24%	1.4-2.1	5.6-7.2
8	-6%	0%	38%	1.4-1.8	5.6-6.5	0%	-2%	27%	0.8-1.0	5.4-6.0	-8%	41%	-4%	1.1-1.3	4.7-5.4	3%	1.7-3.8	5.9-8.6
9	55%	55%	79%	2.0-3.0	4.8-6.3	45%	44%	65%	1.6-2.3	4.7-5.8	47%	79%	47%	2.1-2.9	4.4-5.6	55%	2.0-3.2	4.8-6.6
10	39%	39%	61%	1.3-1.8	4.1-5.0	32%	31%	46%	0.8-1.1	3.9-4.5	32%	60%	30%	1.3-1.7	3.6-4.3	37%	1.1-2.5	3.8-5.6
11	20%	25%	34%	2.1-2.8	6.4-7.6	16%	21%	27%	1.7-2.1	6.2-7.3	15%	33%	20%	1.9-2.4	5.6-6.7	23%	2.0-5.1	6.1-9.9
12	34%	37%	71%	2.2-3.2	6.0-7.8	26%	28%	59%	1.7-2.3	5.8-7.3	25%	69%	27%	2.0-2.8	5.3-6.8	35%	2.1-3.2	5.9-7.7
13	19%	21%	52%	2.1-2.7	5.9-7.0	12%	17%	35%	1.6-2.0	5.7-6.6	9%	50%	12%	1.9-2.4	5.2-6.1	29%	2.4-7.2	6.2-11.8
14	2%	6%	29%	1.4-1.9	7.2-8.6	-1%	3%	22%	0.8-1.1	7.0-8.2	-2%	18%	1%	0.7-1.1	6.1-7.2	4%	1.4-2.5	7.0-8.8
15	32%	34%	50%	1.2-1.7	5.0-6.1	26%	27%	37%	0.8-1.1	4.9-5.7	23%	46%	28%	1.0-1.4	4.4-5.3	30%	1.0-1.7	4.6-5.9
16	21%	20%	65%	1.4-2.0	4.2-5.2	14%	19%	48%	0.8-1.1	3.9-4.6	13%	67%	12%	1.4-1.8	3.5-4.3	21%	1.4-2.4	4.1-5.4
17	30%	31%	47%	1.3-1.8	6.3-7.6	25%	22%	38%	0.8-1.1	6.2-7.2	19%	39%	19%	0.9-1.2	5.4-6.5	27%	1.2-2.0	6.0-7.4
18	21%	27%	73%	2.2-2.9	4.7-5.8	13%	20%	57%	1.7-2.2	4.4-5.3	15%	76%	19%	2.3-2.9	4.1-5.0	25%	2.1-4.7	4.5-7.5
19	3%	2%	44%	2.2-3.0	6.2-7.5	3%	0%	33%	1.7-2.2	6.0-7.1	-1%	47%	-4%	2.0-2.6	5.4-6.5	5%	2.2-6.4	6.0-11.0
20	40%	47%	52%	1.8-2.7	5.8-7.3	36%	37%	44%	1.6-2.2	5.9-7.1	32%	43%	39%	1.7-2.4	5.5-6.7	29%	1.4-2.1	5.2-6.4
21	29%	33%	67%	1.3-1.9	4.3-5.3	21%	28%	50%	0.8-1.1	4.0-4.8	25%	68%	31%	1.3-1.7	3.7-4.6	32%	1.1-1.9	3.9-5.1
22	9%	11%	31%	2.2-3.1	10.6-13.0	8%	11%	28%	1.6-2.2	10.6-12.8	4%	26%	7%	1.0-1.5	9.0-11.1	10%	2.2-4.5	10.3-13.8
23	30%	32%	59%	2.2-3.1	6.2-7.8	25%	25%	51%	1.7-2.3	6.0-7.4	23%	58%	25%	2.0-2.7	5.4-6.8	29%	2.1-3.4	6.1-7.8
24	19%	22%	56%	2.1-2.7	6.1-7.0	17%	21%	41%	1.7-2.0	5.9-6.7	11%	54%	16%	1.9-2.3	5.4-6.2	34%	2.5-7.2	6.5-11.9
25	17%	21%	49%	1.4-1.7	5.3-6.1	13%	19%	38%	0.8-1.0	5.1-5.7	9%	48%	14%	1.1-1.4	4.5-5.2	23%	1.4-3.2	5.3-7.7
26	22%	25%	60%	1.4-2.2	5.3-7.1	16%	21%	42%	0.8-1.2	5.4-6.5	12%	56%	16%	1.1-1.7	4.6-6.0	23%	1.3-2.0	5.3-6.7
27	32%	33%	57%	2.2-2.9	6.1-7.4	24%	29%	46%	1.7-2.2	6.0-7.0	23%	57%	29%	2.0-2.5	5.4-6.5	33%	2.0-3.7	5.8-8.1
	24%	26%	54%			19%	22%	42%			17%	51%	19%			25%		

Appendix D (continued): Kabir and Al-Olayan Comparison for β=1.5

Policy (P)		P7 Uniform			P8 Normal			P8 Uniform			P9 Normal			P9 Uniform		
Run	t^*	Δ	s	S	Δ	s	S	Δ	s	S	Δ	s	S	Δ	s	S
1	181.7	15%	0.8-2.6	4.1-7.4	11%	0.8-1.2	5.3-6.6	12%	0.6-1.4	4.1-6.8	8%	1.0-2.0	4.6-6.0	12%	0.7-2.7	3.6-6.8
2	181.7	-3%	1.0-3.6	5.8-10.5	-1%	0.8-1.2	6.8-8.0	-4%	0.6-1.7	5.8-9.3	-6%	0.8-1.6	5.9-7.3	-7%	0.6-3.6	4.9-9.6
3	181.7	52%	1.2-3.7	4.4-8.2	45%	1.6-2.2	5.5-6.9	45%	1.2-2.5	4.6-7.5	47%	1.8-3.3	5.1-6.9	48%	1.3-3.8	4.1-7.6
4	181.7	30%	1.2-4.0	5.8-10.7	27%	1.6-2.1	7.6-8.9	27%	1.0-2.7	5.9-10.2	20%	1.4-2.7	6.6-8.3	21%	0.9-3.8	5.1-9.7
5	154.6	9%	0.7-2.0	5.2-8.4	8%	0.8-1.3	6.7-8.2	8%	0.5-1.3	5.3-8.1	5%	0.7-2.3	5.9-8.1	5%	0.5-2.2	4.6-7.9
6	181.7	41%	1.6-4.0	3.9-7.1	34%	1.6-2.4	4.6-6.1	33%	1.2-2.5	3.7-6.2	33%	2.1-3.3	4.2-5.6	31%	1.6-3.3	3.3-5.7
7	412.4	23%	1.5-2.2	5.8-7.3	19%	0.8-1.2	5.4-6.7	18%	0.9-1.3	5.5-6.9	15%	1.1-1.6	4.7-6.0	14%	1.1-1.7	4.8-6.1
8	99.8	-2%	0.4-3.4	2.7-8.2	-3%	1.0-1.6	5.7-7.2	-3%	0.2-1.6	2.7-7.0	-3%	1.3-3.4	5.0-7.2	-5%	0.4-3.0	2.3-6.9
9	412.4	54%	2.1-3.2	4.9-6.6	44%	1.6-2.6	4.6-6.4	46%	1.7-2.6	4.7-6.3	46%	2.1-3.3	4.3-6.0	46%	2.2-3.2	4.4-6.0
10	154.6	35%	0.6-2.0	2.8-5.0	30%	0.8-1.4	3.7-4.9	27%	0.5-1.2	2.8-4.7	29%	1.3-2.9	3.4-5.3	29%	0.8-2.3	2.5-4.7
11	154.6	14%	0.9-4.0	3.7-8.6	20%	1.6-2.7	6.0-8.0	10%	0.8-2.4	3.8-7.5	18%	1.8-4.2	5.4-8.2	8%	0.9-3.4	3.3-7.2
12	412.4	36%	2.1-3.5	5.9-8.1	27%	1.7-2.4	5.8-7.4	30%	1.6-2.6	5.7-7.8	24%	1.9-2.9	5.2-6.8	26%	1.9-3.1	5.2-7.1
13	99.8	17%	0.9-7.3	3.4-11.8	21%	1.8-3.2	6.1-8.5	12%	0.7-3.0	3.4-8.2	18%	2.1-5.6	5.4-9.1	11%	1.0-5.5	3.0-9.0
14	154.6	1%	0.8-2.5	5.1-8.7	4%	0.9-1.3	6.8-8.2	2%	0.5-1.3	5.1-8.1	1%	0.8-2.3	5.9-7.9	0%	0.5-2.4	4.4-7.8
15	154.6	25%	0.5-1.7	3.3-5.9	28%	0.8-1.2	4.7-5.8	24%	0.4-1.2	3.4-5.8	26%	1.0-2.3	4.2-5.9	24%	0.7-2.2	3.1-5.9
16	154.6	20%	0.9-2.6	3.1-5.7	13%	0.9-1.3	3.8-4.7	15%	0.6-1.4	3.0-5.0	12%	1.4-2.2	3.5-4.5	14%	1.0-2.4	2.6-4.8
17	164.6	25%	0.8-3.6	4.9-9.8	22%	0.8-1.2	6.0-7.1	24%	0.6-1.8	4.9-8.4	17%	0.9-2.1	5.2-6.9	20%	0.7-4.6	4.3-9.8
18	164.6	31%	1.6-5.9	3.5-8.8	17%	1.6-2.6	4.2-6.0	23%	1.1-2.9	3.3-6.6	18%	2.1-3.6	3.9-5.5	22%	1.6-4.1	2.9-6.1
19	164.6	9%	1.4-8.1	4.5-13.0	-1%	1.6-3.0	5.8-8.5	4%	1.1-3.5	4.4-9.3	-5%	1.9-4.5	5.1-8.2	-1%	1.2-5.6	3.8-9.4
20	164.6	29%	0.7-2.6	3.7-7.2	33%	1.5-2.0	5.6-6.8	33%	0.8-2.4	4.0-7.5	31%	1.6-2.5	5.2-6.5	34%	1.0-3.3	3.6-7.4
21	164.6	34%	0.5-3.1	2.5-6.5	24%	0.8-1.2	3.8-4.8	28%	0.4-1.6	2.5-5.4	20%	1.2-2.1	3.5-4.6	28%	0.7-3.4	2.2-6.0
22	164.6	13%	1.3-6.6	7.2-16.5	10%	1.6-2.5	10.2-12.6	10%	0.9-3.0	7.1-14.0	5%	1.0-3.1	8.7-11.7	10%	0.5-5.2	6.0-14.2
23	296.8	37%	2.0-3.7	5.9-8.3	25%	1.7-2.4	5.9-7.4	32%	1.6-2.5	5.8-7.7	19%	1.9-2.9	5.4-6.7	30%	1.8-3.2	5.2-7.1
24	99.7	22%	0.7-7.2	2.9-12.0	24%	1.9-3.1	6.3-8.4	16%	0.6-3.2	3.0-8.5	21%	2.2-5.6	5.7-9.2	13%	0.8-5.6	2.6-9.3
25	102.1	16%	0.3-1.8	2.3-6.0	20%	0.9-1.6	5.2-6.5	14%	0.2-1.1	2.4-5.8	18%	1.3-3.4	4.6-7.0	11%	0.4-1.7	2.0-5.3
26	872.6	24%	1.4-2.0	5.3-6.7	17%	0.8-1.2	5.1-6.2	16%	0.8-1.2	5.1-6.2	15%	1.1-1.6	4.5-5.7	14%	1.1-1.6	4.6-5.7
27	164.6	31%	1.0-4.7	3.8-9.3	29%	1.6-2.4	5.7-7.2	24%	0.8-2.8	3.8-8.0	27%	1.8-3.2	5.1-6.8	21%	1.0-3.9	3.3-7.7
Avg Difference=		24%			20%			20%			18%			18%		

	P10 Fixed	P10 Normal	P10 Uniform			P11 Fixed	P11 Normal	P11 Uniform			P12 Fixed	P12 Normal	P12 Uniform			P13 Normal		
Run	Δ	Δ	Δ	s	S	Δ	Δ	Δ	s	S	Δ	Δ	Δ	s	S	Δ	s	S
1	5%	5%	7%	0.6-1.5	3.4-5.6	21%	3%	6%	0.3-0.8	3.3-5.2	-1%	-1%	0%	0.5-1.0	2.9-4.7	11%	0.6-6.7	3.3-13.0
2	-11%	-8%	-6%	0.7-1.5	4.5-7.3	15%	-8%	-8%	0.4-0.8	4.5-7.0	-13%	-9%	-5%	0.3-0.7	3.8-6.1	-6%	0.6-7.5	4.4-16.0
3	26%	36%	31%	1.0-2.4	3.7-6.3	50%	34%	26%	0.8-1.7	3.7-5.8	19%	28%	24%	0.9-1.9	3.3-5.4	46%	0.9-14.5	3.3-21.1
4	20%	16%	16%	0.9-2.3	4.8-8.0	51%	16%	12%	0.7-1.6	4.7-7.7	22%	16%	8%	0.6-1.5	4.1-6.9	20%	1.0-6.5	4.8-13.9
5	0%	4%	11%	0.6-1.5	4.3-7.2	22%	1%	9%	0.3-0.8	4.3-6.8	-4%	-3%	5%	0.2-0.6	3.6-6.0	14%	0.6-13.8	4.5-24.4
6	14%	27%	20%	1.0-2.3	2.9-5.1	35%	28%	16%	0.7-1.6	2.8-4.6	14%	23%	14%	1.1-2.2	2.5-4.4	34%	1.1-7.9	3.0-11.5
7	8%	13%	22%	0.7-1.7	3.5-6.1	28%	15%	24%	0.4-0.9	3.3-5.6	8%	9%	20%	0.5-1.2	2.9-5.0	9%	0.6-1.7	3.3-6.1
8	-7%	-5%	-6%	0.6-1.2	3.3-5.1	7%	-1%	-5%	0.3-0.7	3.2-4.7	-5%	-5%	-5%	0.5-0.9	2.8-4.2	47%	0.6-20.0	3.4-27.8
9	28%	28%	28%	1.0-2.4	2.9-5.3	40%	24%	17%	0.7-1.7	2.8-4.8	22%	22%	18%	1.1-2.2	2.5-4.6	26%	1.0-2.6	2.8-5.6
10	25%	24%	29%	0.6-1.4	2.5-4.1	29%	19%	30%	0.4-0.7	2.4-3.6	18%	17%	25%	0.7-1.2	2.1-3.4	34%	0.6-5.6	2.5-9.7
11	9%	13%	11%	1.0-2.1	4.0-6.2	19%	15%	10%	0.8-1.5	3.9-5.9	5%	11%	8%	0.9-1.7	3.4-5.3	24%	1.0-7.2	3.9-12.7
12	12%	16%	12%	1.1-2.6	3.8-6.7	30%	16%	3%	0.8-1.8	3.7-6.2	5%	10%	2%	1.0-2.2	3.3-5.7	14%	1.0-3.1	3.6-7.6
13	2%	10%	10%	1.0-2.0	3.5-5.6	21%	15%	19%	0.7-1.4	3.4-5.2	1%	7%	4%	0.9-1.7	3.0-4.8	98%	1.0-31.7	3.6-38.6
14	-6%	-2%	-3%	0.6-1.5	4.4-7.2	17%	-3%	-6%	0.3-0.8	4.4-6.8	-9%	-2%	-4%	0.3-0.7	3.0-6.0	10%	0.7-7.7	4.5-16.4
15	15%	23%	20%	0.6-1.4	3.1-5.1	30%	15%	19%	0.3-0.7	3.0-4.7	6%	11%	15%	0.5-1.0	2.6-4.2	26%	0.6-4.9	3.1-10.5
16	8%	10%	9%	0.6-1.5	2.5-4.3	17%	5%	5%	0.3-0.8	2.3-3.8	2%	5%	2%	0.7-1.4	2.1-3.5	33%	0.6-17.5	2.4-22.6
17	13%	15%	18%	0.6-1.5	3.9-6.3	43%	13%	12%	0.3-0.8	3.9-6.0	11%	12%	9%	0.4-0.8	3.3-5.3	34%	0.7-14.3	4.1-23.4
18	7%	16%	12%	1.1-2.2	2.8-4.7	15%	14%	9%	0.8-1.6	2.7-4.2	2%	13%	7%	1.2-2.2	2.4-4.0	23%	1.0-8.5	2.6-11.7
19	-8%	-9%	-1%	1.1-2.3	3.8-6.2	7%	-3%	-2%	0.8-1.6	3.7-5.8	-10%	-13%	-4%	1.0-1.9	3.3-5.3	15%	1.0-19.6	3.6-25.7
20	20%	26%	23%	1.0-2.3	3.9-6.2	38%	23%	20%	0.8-1.6	3.8-5.9	14%	20%	18%	0.9-1.7	3.4-5.4	26%	1.0-4.5	3.7-10.1
21	12%	19%	22%	0.7-1.5	2.7-4.5	25%	12%	22%	0.4-0.8	2.5-4.0	6%	10%	16%	0.7-1.3	2.3-3.7	32%	0.7-13.9	2.7-19.3
22	4%	6%	3%	1.0-2.4	6.6-11.1	31%	7%	4%	0.7-1.7	6.6-10.8	17%	19%	10%	0.3-1.0	5.5-9.3	14%	1.1-21.0	7.0-34.0
23	12%	14%	18%	1.1-2.5	3.9-6.7	32%	10%	14%	0.8-1.8	3.8-6.2	9%	11%	9%	1.0-2.1	3.4-5.7	9%	1.1-3.3	3.8-8.1
24	1%	10%	14%	1.1-1.9	3.8-5.6	23%	14%	21%	0.8-1.4	3.7-5.2	0%	10%	13%	1.0-1.6	3.3-4.8	83%	0.9-31.5	3.6-38.7
25	4%	11%	12%	0.6-1.3	3.3-4.9	19%	12%	18%	0.4-0.7	3.2-4.5	2%	8%	14%	0.6-0.9	2.8-4.1	68%	0.6-15.6	3.1-22.9
26	6%	13%	10%	0.7-1.9	3.3-6.2	26%	5%	5%	0.4-1.0	3.2-5.6	-3%	1%	0%	0.6-1.3	2.8-5.1	6%	0.6-1.7	3.2-5.8
27	12%	22%	13%	1.1-2.2	3.9-6.1	30%	19%	8%	0.8-1.6	3.8-5.7	10%	15%	4%	1.0-1.9	3.4-5.2	26%	1.0-7.8	3.8-13.5
	8%	13%	13%			27%	12%	11%			5%	9%	8%			29%		

Appendix D (continued): Kabir and Al-Olayan Comparison for β=1.5

Policy (P)		P13 Uniform			P14 Normal			P14 Uniform			P15 Normal			P15 Uniform			P16 Fixed	P16 Normal
Run	t^*	Δ	s	S	Δ	s	S	Δ	s	S	Δ	s	S	Δ	s	S	Δ	Δ
1	181.7	11%	0.6-6.6	3.4-12.8	4%	0.3-2.9	3.2-9.9	7%	0.3-2.9	3.3-9.8	5%	0.5-7.3	2.8-12.3	5%	0.5-7.2	2.8-12.1	29%	24%
2	181.7	-3%	0.6-7.5	4.3-15.8	-6%	0.3-2.9	4.3-12.2	-4%	0.3-2.9	4.2-12.5	-9%	0.3-8.6	3.7-15.5	-7%	0.3-8.5	3.6-15.3	35%	37%
3	181.7	37%	1.0-7.0	3.5-12.7	40%	0.7-6.0	3.3-14.0	33%	0.7-4.3	3.5-10.8	38%	0.8-12.8	2.9-17.8	30%	0.9-6.8	3.1-11.6	49%	53%
4	181.7	29%	1.0-6.6	4.7-14.0	19%	0.7-3.4	4.8-11.8	20%	0.7-3.7	4.7-12.5	17%	0.6-6.4	4.2-12.7	22%	0.6-6.5	4.0-12.8	35%	32%
5	154.6	8%	0.5-5.8	3.9-14.4	6%	0.4-4.6	4.4-15.9	7%	0.3-2.7	3.9-12.0	18%	0.3-20.1	3.8-28.0	9%	0.2-8.4	3.3-15.6	14%	13%
6	181.7	28%	0.8-7.7	2.5-11.1	30%	0.8-4.3	2.8-9.4	22%	0.6-3.7	2.4-8.4	21%	1.2-5.7	2.6-8.8	21%	1.0-5.1	2.2-7.9	67%	70%
7	412.4	9%	0.7-1.7	3.7-6.0	15%	0.3-0.9	3.2-5.6	6%	0.4-0.9	3.5-5.5	9%	0.5-1.2	2.8-5.0	7%	0.6-1.2	3.1-5.0	7%	9%
8	99.8	21%	0.4-19.9	2.6-27.7	4%	0.3-5.6	3.3-15.9	3%	0.2-5.6	2.6-15.8	26%	0.5-16.4	2.9-21.8	9%	0.4-16.4	2.2-21.8	5%	-2%
9	412.4	25%	1.1-2.5	3.1-5.4	23%	.7-1.8	2.7-5.0	22%	0.8-1.7	2.9-4.9	19%	1.1-2.5	2.5-4.8	20%	1.2-2.3	2.7-4.6	74%	73%
10	154.6	31%	0.6-5.6	2.4-9.7	22%	0.4-2.6	2.3-7.4	22%	0.3-2.6	2.2-7.4	29%	0.745.9	2.1-8.9	27%	0.7-5.9	2.0-8.9	39%	34%
11	154.6	17%	1.0-7.3	3.8-12.7	16%	0.7-3.9	3.8-10.9	9%	0.7-3.9	3.7-10.8	17%	0.9-5.9	3.4-10.5	9%	0.9-5.9	3.3-10.4	24%	28%
12	412.4	14%	1.0-2.4	3.5-6.3	8%	0.8-2.4	3.5-7.3	15%	0.7-1.7	3.4-5.9	8%	1.0-2.8	3.1-6.6	9%	0.9-2.0	3.1-5.4	46%	46%
13	99.8	60%	0.7-31.7	2.8-38.7	29%	0.7-7.9	3.5-17.4	21%	0.5-8.0	2.7-17.5	49%	0.9-19.3	3.1-23.9	28%	0.7-19.4	2.4-24.0	27%	29%
14	154.6	4%	0.6-7.5	4.4-16.0	2%	0.4-3.1	4.5-12.7	-1%	0.3-2.9	4.3-12.3	6%	0.8-8.7	3.8-15.9	0%	0.3-8.6	3.7-15.5	36%	42%
15	154.6	28%	0.5-8.9	2.9-15.8	20%	0.3-2.4	3.0-8.7	15%	0.3-3.7	2.8-11.5	28%	0.5-6.6	2.6-11.1	23%	0.5-11.4	2.5-16.6	20%	23%
16	154.6	34%	0.6-17.4	2.3-22.6	10%	0.3-5.1	2.3-12.4	14%	0.3-5.1	2.2-12.4	17%	0.7-12.7	2.0-15.9	20%	0.7-12.7	1.9-15.8	5%	4%
17	164.6	22%	0.6-6.1	3.9-13.5	20%	0.4-4.7	4.0-14.8	17%	0.3-2.8	3.8-10.7	32%	0.4-18.1	3.4-24.8	20%	0.4-7.8	3.3-13.9	41%	34%
18	164.6	23%	0.9-8.3	2.4-11.5	16%	0.7-4.0	2.5-8.5	19%	0.6-3.9	2.3-8.4	11%	1.1-5.5	2.3-7.9	16%	1.0-5.5	2.1-7.8	46%	49%
19	164.6	5%	0.8-8.5	3.3-13.5	-5%	0.7-6.1	3.5-14.8	-1%	0.6-4.0	3.2-10.7	-5%	0.9-11.7	3.1-16.0	-4%	0.8-5.8	2.8-10.3	16%	14%
20	164.6	24%	0.8-4.3	3.4-9.8	24%	0.7-3.6	3.6-9.9	23%	0.6-3.4	3.3-9.6	31%	0.8-5.4	3.3-10.5	23%	0.7-5.2	3.0-10.1	36%	38%
21	164.6	26%	0.5-5.9	2.3-10.2	16%	0.4-4.5	2.5-11.2	18%	0.3-2.6	2.1-7.7	26%	0.7-12.6	2.3-16.1	19%	0.6-6.0	1.9-9.1	26%	28%
22	164.6	16%	0.9-16.7	6.4-30.1	8%	0.8-6.1	7.0-20.8	10%	0.7-6.0	6.4-21.6	18%	0.3-18.6	5.9-28.9	17%	0.2-14.7	5.4-26.2	4%	7%
23	296.8	15%	1.0-3.6	3.8-8.5	10%	0.8-2.5	3.7-7.8	15%	0.8-2.7	3.7-8.1	7%	1.0-2.9	3.3-7.1	11%	1.0-3.2	3.3-7.4	40%	38%
24	99.7	54%	0.6-31.5	2.8-38.7	27%	0.7-7.9	3.5-17.6	24%	0.5-7.9	2.7-17.6	41%	0.9-19.7	3.1-24.4	27%	0.7-19.7	2.4-24.4	29%	31%
25	102.1	47%	0.5-15.5	2.6-22.7	26%	0.3-4.9	3.0-13.7	20%	0.2-4.9	2.6-13.5	61%	0.5-15.5	2.6-20.5	37%	0.4-15.4	2.2-20.4	43%	45%
26	872.6	8%	0.7-1.6	3.3-5.7	3%	0.3-0.9	3.1-5.3	2%	0.4-0.9	3.2-5.2	0%	0.5-1.2	2.7-4.8	-3%	0.6-1.2	2.8-4.7	28%	32%
27	164.6	30%	0.9-16.2	3.4-22.2	21%	0.8-4.3	3.7-11.1	16%	0.7-5.7	3.3-13.6	20%	1.0-6.4	3.2-11.1	17%	0.8-11.3	2.9-15.6	41%	43%
Avg Difference=		23%			15%			14%			20%			15%			32%	32%

	P16 Uniform			P17 Normal			P17 Uniform			P18 Fixed	P18 Normal	P18 Uniform			P19 Normal	P19 Uniform		
Run	Δ	s	S	Δ	s	S	Δ	s	S	Δ	Δ	Δ	s	S	Δ	Δ	s	S
1	30%	0.0	1.0-2.0	17%	0.0	1.0-3.0	26%	0.0-1.0	1.0-3.0	26%	26%	30%	1.0-2.0	7.0-8.0	20%	23%	1.0-2.0	6.0-8.0
2	35%	0.0	2.0-3.0	31%	0.0-1.0	2.0-4.0	26%	0.0-1.0	2.0-4.0	9%	11%	9%	2.0	9.0-10.0	11%	8%	2.0	9.0-10.0
3	50%	3.0-4.0	5.0-6.0	55%	3.0-7.0	5.0-9.0	54%	3.0-5.0	5.0-7.0	60%	66%	66%	2.0-3.0	7.0-8.0	62%	63%	2.0-3.0	6.0-8.0
4	31%	3.0-4.0	6.0	33%	3.0-5.0	6.0-8.0	33%	3.0-5.0	5.0-8.0	49%	47%	47%	2.0-3.0	9.0-11.0	34%	35%	2.0-3.0	9.0-10.0
5	15%	0.0	1.0-2.0	5%	0.0-1.0	0.0	11%	0.0-1.0	1.0-4.0	17%	17%	19%	1.0-2.0	8.0-10.0	12%	15%	0.0-2.0	0.0-9.0
6	69%	4.0-5.0	6.0	70%	4.0-6.0	5.0-7.0	70%	4.0-6.0	5.0-7.0	64%	68%	68%	3.0-4.0	6.0-8.0	57%	57%	2.0-4.0	6.0-8.0
7	8%	0.0	2.0-3.0	8%	0.0	2.0-3.0	4%	0.0	2.0-3.0	34%	38%	35%	2.0-3.0	7.0-9.0	37%	36%	2.0-3.0	7.0-8.0
8	-1%	0.0	2.0	-11%	0.0-3.0	2.0-5.0	-4%	0.0-2.0	1.0-4.0	2%	8%	7%	2.0	7.0-8.0	6%	6%	0.0-3.0	0.0-9.0
9	72%	4.0	5.0-6.0	72%	4.0	5.0-6.0	72%	4.0	5.0-6.0	72%	75%	74%	2.0-3.0	6.0-7.0	71%	73%	2.0-3.0	6.0-8.0
10	35%	0.0	1.0-2.0	22%	0.0-1.0	1.0-3.0	27%	0.0-1.0	1.0-3.0	39%	40%	40%	1.0-2.0	5.0-6.0	39%	39%	1.0-2.0	4.0-6.0
11	21%	4.0	6.0	30%	4.0-6.0	5.0-8.0	24%	4.0-6.0	5.0-8.0	29%	34%	26%	3.0	8.0-9.0	29%	24%	2.0-4.0	7.0-9.0
12	47%	4.0-5.0	6.0	46%	4.0-5.0	6.0-7.0	47%	4.0	5.0-6.0	54%	58%	55%	0.0-4.0	0.0-9.0	55%	57%	0.0-4.0	0.0-9.0
13	27%	4.0	5.0-6.0	37%	4.0-7.0	5.0-8.0	33%	4.0-7.0	5.0-8.0	33%	37%	34%	3.0	7.0-8.0	33%	31%	3.0	7.0-9.0
14	39%	0.0	2.0-3.0	30%	0.0-1.0	2.0-4.0	31%	0.0-1.0	2.0-4.0	11%	13%	13%	2.0	9.0-10.0	13%	12%	2.0	9.0-10.0
15	18%	0.0	1.0-2.0	16%	0.0-1.0	1.0-3.0	14%	0.0-1.0	1.0-3.0	33%	36%	35%	1.0	6.0-7.0	34%	35%	1.0	5.0-7.0
16	4%	0.0	1.0-2.0	-1%	0.0-2.0	1.0-4.0	2%	0.0-2.0	1.0-4.0	35%	38%	39%	2.0	5.0-6.0	27%	31%	1.0-2.0	5.0-6.0
17	30%	0.0	1.0-2.0	24%	0.0-2.0	1.0-4.0	22%	0.0-1.0	1.0-3.0	41%	39%	40%	1.0-2.0	8.0-9.0	30%	30%	1.0-2.0	7.0-8.0
18	52%	4.0-5.0	6.0	49%	4.0-6.0	5.0-7.0	52%	4.0-6.0	5.0-8.0	39%	46%	47%	3.0-4.0	6.0-7.0	36%	38%	0.0-4.0	0.0-7.0
19	17%	4.0-5.0	6.0-7.0	14%	4.0-7.0	6.0-9.0	17%	4.0-6.0	6.0-8.0	21%	21%	23%	3.0-4.0	8.0-9.0	15%	16%	0.0-5.0	0.0-10.0
20	37%	2.0-3.0	5.0	38%	2.0-4.0	5.0-7.0	38%	2.0-4.0	5.0-7.0	42%	46%	45%	1.0-2.0	7.0-8.0	32%	32%	1.0-2.0	6.0-7.0
21	32%	0.0	1.0-2.0	19%	0.0-2.0	1.0-4.0	28%	0.0-1.0	1.0-3.0	39%	45%	47%	1.0-2.0	5.0-6.0	32%	35%	0.0-2.0	0.0-6.0
22	7%	4.0-5.0	6.0-7.0	7%	4.0-6.0	6.0-8.0	7%	4.0-6.0	6.0-8.0	18%	20%	21%	3.0-4.0	13.0-15.0	16%	16%	3.0-4.0	12.0-15.0
23	41%	4.0	6.0	38%	4.0-5.0	6.0-7.0	41%	4.0-5.0	6.0-7.0	47%	47%	49%	3.0-4.0	8.0-9.0	46%	47%	0.0-4.0	0.0-9.0
24	30%	4.0	6.0	39%	4.0-7.0	5.0-8.0	37%	4.0-8.0	5.0-9.0	35%	37%	37%	3.0	8.0	37%	35%	2.0-4.0	7.0-9.0
25	47%	0.0	1.0-2.0	19%	0.0-2.0	1.0-4.0	28%	0.0-2.0	1.0-4.0	30%	34%	33%	1.0-2.0	7.0	25%	27%	1.0-2.0	6.0-8.0
26	27%	0.0	1.0-2.0	34%	0.0	1.0-2.0	26%	0.0	1.0-2.0	39%	42%	40%	1.0-2.0	7.0-8.0	45%	40%	1.0-2.0	7.0-8.0
27	39%	4.0	6.0	46%	4.0-6.0	5.0-7.0	43%	4.0-7.0	5.0-8.0	48%	51%	45%	3.0	8.0	43%	41%	2.0-3.0	7.0-9.0
	32%			29%			30%			36%	39%	38%			33%	33%		

Appendix E: Kabir and Al-Olayan Comparison for β=2.0

Policy (P)		P1 Fixed	P1 Normal	P1 Uniform			P2 Fixed	P2 Normal	P2 Uniform			P3 Fixed	P3 Normal	P3 Uniform		
Run	t^*	Δ	Δ	Δ	s	S	Δ	Δ	Δ	s	S	Δ	Δ	Δ	s	S
1	98.1	17%	19%	16%	1.0-1.7	4.3-5.6	15%	19%	13%	1.0-1.7	4.3-5.6	15%	17%	13%	1.0-1.9	3.8-5.1
2	98.1	18%	18%	18%	3.1-4.9	8.1-10.5	7%	9%	2%	3.1-4.9	8.1-10.5	7%	7%	7%	2.4-4.0	6.6-8.5
3	98.1	33%	34%	35%	0.9-1.4	4.0-5.0	27%	26%	27%	0.9-1.4	4.0-5.0	33%	34%	35%	1.6-2.3	4.0-5.0
4	98.1	18%	18%	22%	1.2-1.8	5.7-7.2	16%	16%	18%	1.2-1.8	5.7-7.2	20%	19%	23%	1.8-2.8	5.5-7.2
5	88.9	21%	16%	13%	1.6-5.6	7.2-12.8	15%	10%	6%	1.6-5.6	7.2-12.8	26%	22%	19%	2.1-9.1	7.0-14.3
6	98.1	19%	22%	20%	1.9-4.7	4.5-7.8	2%	5%	3%	1.9-4.7	4.5-7.8	5%	9%	7%	1.9-4.0	3.7-6.1
7	154.4	39%	36%	41%	2.7-4.7	6.2-8.7	23%	20%	26%	2.7-4.7	6.2-8.7	22%	20%	28%	1.9-3.2	4.7-6.3
8	66.9	5%	5%	7%	2.5-4.2	6.6-8.9	-3%	-6%	0%	2.5-4.2	6.6-8.9	-1%	0%	2%	2.1-3.8	5.5-7.4
9	154.4	45%	42%	43%	1.5-2.4	3.8-4.9	32%	34%	32%	1.5-2.4	3.8-4.9	44%	40%	41%	1.7-2.3	3.4-4.2
10	98.1	25%	25%	26%	1.3-1.9	3.6-4.7	12%	13%	11%	1.3-1.9	3.6-4.7	23%	24%	23%	1.8-2.4	3.4-4.4
11	98.1	8%	10%	10%	1.2-2.2	4.6-6.3	4%	7%	8%	1.2-2.2	4.6-6.3	9%	11%	12%	1.5-2.6	4.1-5.8
12	154.4	19%	23%	20%	1.1-1.6	4.0-4.9	17%	15%	17%	1.1-1.6	4.0-4.9	18%	22%	18%	1.3-1.7	3.5-4.4
13	66.9	42%	37%	40%	1.7-3.3	5.0-7.5	24%	19%	23%	1.7-3.3	5.0-7.5	30%	25%	29%	2.0-3.6	4.5-6.8
14	98.1	20%	18%	23%	3.4-8.2	8.9-14.4	12%	13%	11%	3.4-8.2	8.9-14.4	8%	8%	11%	2.6-6.6	7.3-11.1
15	98.1	38%	38%	39%	2.4-3.7	6.0-8.2	23%	22%	22%	2.4-3.7	6.0-8.2	29%	26%	28%	2.6-3.8	5.6-7.6
16	98.1	40%	43%	41%	3.4-5.7	5.9-8.5	18%	18%	23%	3.4-5.7	5.9-8.5	13%	17%	16%	2.6-3.7	4.3-5.6
17	92.4	21%	16%	16%	1.6-3.6	6.3-8.6	17%	12%	12%	1.6-3.6	6.3-8.6	28%	25%	27%	2.0-5.3	6.1-8.9
18	92.4	18%	16%	18%	1.5-2.2	3.5-4.5	7%	1%	5%	1.5-2.2	3.5-4.5	5%	1%	5%	1.5-2.0	2.8-3.6
19	92.4	19%	18%	20%	2.1-2.9	5.7-7.3	13%	10%	8%	2.1-2.9	5.7-7.3	15%	16%	15%	1.9-2.6	4.8-6.2
20	92.4	15%	13%	16%	0.4-0.9	3.6-4.8	15%	12%	16%	0.4-0.9	3.6-4.8	24%	21%	26%	1.3-2.2	4.1-5.5
21	92.4	39%	41%	40%	2.3-3.5	5.0-6.5	22%	23%	21%	2.3-3.5	5.0-6.5	27%	28%	26%	2.2-3.1	4.3-5.4
22	92.4	16%	20%	17%	3.0-3.9	10.5-12.0	11%	19%	18%	3.0-3.9	10.5-12.0	10%	13%	11%	1.9-2.8	8.8-10.1
23	129.8	31%	31%	31%	0.7-3.6	2.6-6.9	21%	20%	23%	0.7-3.6	2.6-6.9	21%	22%	23%	0.8-3.0	2.2-5.3
24	65.4	47%	49%	52%	2.8-5.7	6.2-10.1	24%	23%	29%	2.8-5.7	6.2-10.1	32%	30%	34%	2.4-4.5	5.0-7.9
25	67.9	27%	29%	28%	1.9-3.0	5.9-7.3	15%	15%	15%	1.9-3.0	5.9-7.3	20%	19%	21%	2.0-3.2	5.3-6.7
26	225.7	40%	39%	39%	2.3-3.2	5.5-6.9	28%	29%	29%	2.3-3.2	5.5-6.9	27%	27%	29%	2.0-2.7	4.5-5.7
27	92.4	19%	20%	17%	2.7-4.2	6.0-8.1	4%	6%	5%	2.4-4.2	6.0-8.1	7%	9%	7%	2.3-3.4	4.8-6.5
Avg Difference=		26%	26%	26%			15%	15%	16%			19%	19%	20%		

	P4 Fixed	P4 Normal	P4 Uniform			P5 Fixed	P5 Normal	P5 Uniform			P6 Fixed	P6 Normal	P6 Uniform			P7 Normal		
Run	Δ	Δ	Δ	s	S	Δ	Δ	Δ	s	S	Δ	Δ	Δ	s	S	Δ	s	S
1	59%	62%	82%	2.9-3.9	8.5-10.6	43%	47%	65%	1.8-2.4	8.0-9.8	48%	50%	74%	2.7-3.5	7.6-9.2	27%	1.5-4.2	5.7-9.2
2	60%	66%	92%	5.1-6.9	12.7-16.0	41%	48%	71%	3.3-4.6	11.7-14.9	41%	43%	73%	3.8-5.3	10.7-13.6	22%	2.6-8.4	8.4-15.3
3	95%	95%	110%	2.3-3.3	7.4-9.3	79%	85%	101%	1.7-2.3	7.2-8.9	88%	93%	113%	2.6-3.4	7.1-8.7	48%	1.1-3.0	5.0-7.5
4	62%	63%	80%	2.6-3.7	10.2-12.8	52%	52%	72%	1.7-2.4	10.0-12.3	49%	49%	67%	2.4-3.1	9.4-11.5	26%	1.3-3.8	7.1-10.7
5	44%	44%	53%	2.6-3.7	10.1-12.8	38%	37%	44%	1.7-2.4	9.9-12.3	38%	35%	45%	2.4-3.1	9.4-11.6	17%	1.4-3.6	7.3-10.4
6	66%	69%	115%	3.2-4.2	6.8-8.6	40%	43%	88%	1.8-2.4	6.1-7.5	52%	52%	108%	2.9-3.8	5.7-7.1	23%	1.6-4.7	4.4-7.9
7	100%	103%	149%	5.5-8.3	11.1-15.6	72%	77%	116%	3.4-5.5	10.0-14.2	72%	73%	127%	4.2-6.7	9.0-12.9	41%	2.7-6.8	7.1-11.9
8	18%	19%	60%	3.4-4.1	9.0-10.4	6%	8%	47%	1.8-2.2	8.2-9.5	8%	10%	59%	2.7-3.3	7.5-8.7	12%	1.9-11.7	6.3-17.5
9	148%	150%	173%	4.4-6.5	8.1-11.1	116%	121%	146%	3.3-4.9	7.6-10.3	130%	135%	161%	4.3-6.3	7.2-9.9	60%	2.3-4.0	5.2-7.3
10	66%	73%	92%	2.5-3.6	6.2-8.0	50%	53%	70%	1.7-2.4	5.8-7.3	63%	67%	91%	2.7-3.7	5.7-7.2	30%	1.3-3.5	4.2-6.8
11	36%	38%	56%	2.9-4.0	8.4-10.6	24%	28%	45%	1.7-2.4	7.9-9.8	27%	30%	52%	2.6-3.5	7.5-9.3	16%	1.5-4.2	5.8-9.2
12	95%	99%	125%	2.9-4.5	7.8-10.9	71%	77%	104%	1.7-2.7	7.3-9.9	80%	85%	118%	2.7-4.0	6.9-9.5	37%	1.5-2.8	5.4-7.3
13	65%	61%	103%	2.9-3.7	7.8-9.4	41%	45%	77%	1.7-2.2	7.3-8.6	49%	47%	95%	2.7-3.3	6.9-8.2	40%	1.6-9.0	5.6-14.3
14	52%	53%	83%	5.2-6.7	12.8-15.8	36%	37%	64%	3.3-4.5	11.9-14.7	34%	35%	67%	3.9-5.2	10.8-13.4	21%	2.7-8.6	8.6-15.5
15	90%	92%	109%	4.0-5.4	9.0-11.3	77%	77%	90%	3.1-4.3	8.7-10.9	81%	83%	100%	4.0-5.3	8.4-10.5	40%	2.1-5.6	6.0-10.1
16	93%	100%	142%	5.1-6.6	8.8-10.9	63%	68%	103%	3.3-4.5	7.7-9.7	68%	74%	122%	4.3-5.7	7.1-9.0	49%	2.7-8.3	5.5-11.6
17	62%	62%	83%	2.7-3.7	9.3-11.5	48%	48%	66%	1.7-2.3	9.0-10.9	49%	49%	67%	2.5-3.2	8.5-10.3	21%	1.4-3.8	6.4-9.7
18	51%	52%	93%	3.2-4.2	6.5-8.1	33%	33%	75%	1.8-2.4	5.6-7.0	38%	38%	97%	2.9-3.8	5.3-6.6	16%	1.6-7.7	4.1-10.9
19	54%	56%	101%	3.2-4.4	8.3-10.6	35%	37%	80%	1.8-20.4	7.6-9.5	38%	42%	95%	2.7-3.6	7.0-8.9	19%	1.6-5.1	5.6-9.8
20	57%	57%	72%	1.8-2.6	7.0-8.8	51%	52%	64%	1.5-2.1	7.3-8.9	62%	60%	73%	2.4-3.1	7.3-8.8	18%	0.8-3.2	4.7-8.2
21	121%	127%	153%	4.4-6.0	8.2-10.6	90%	100%	121%	3.2-4.4	7.6-9.7	100%	110%	139%	4.1-5.6	7.2-9.3	52%	2.3-6.6	5.2-10.1
22	40%	45%	61%	5.5-7.1	16.6-20.4	31%	36%	51%	3.3-4.6	15.7-19.4	27%	31%	50%	3.6-4.7	14.1-17.4	24%	2.7-10.1	11.2-20.3
23	101%	105%	129%	4.8-7.0	10.0-13.5	77%	84%	103%	3.3-5.0	9.2-12.5	79%	89%	110%	4.1-6.1	8.5-11.7	43%	2.5-5.2	6.6-9.6
24	85%	88%	125%	4.7-5.8	9.9-11.7	60%	62%	96%	3.3-4.1	9.1-10.8	65%	65%	110%	4.1-5.0	8.5-10.1	55%	2.6-9.6	6.8-14.6
25	47%	48%	79%	2.8-3.6	8.0-9.6	31%	33%	59%	1.7-2.2	7.4-8.8	34%	36%	69%	2.6-3.3	7.0-8.4	29%	1.6-5.8	5.7-11.2
26	148%	161%	186%	4.9-7.8	10.1-14.1	113%	125%	151%	3.3-5.6	9.3-13.6	118%	126%	162%	4.2-6.9	8.7-12.8	82%	3.0-4.7	7.6-9.3
27	56%	60%	77%	4.7-6.3	9.9-12.4	39%	46%	57%	3.2-4.5	9.1-11.5	43%	48%	66%	4.0-5.5	8.5-10.7	23%	2.4-7.3	6.4-12.0
	73%	76%	103%			54%	58%	82%			59%	61%	93%			33%		

Appendix E (continued): Kabir and Al-Olayan Comparison for β=2.0

Policy (P)		P7 Uniform			P8 Normal			P8 Uniform			P9 Normal			P9 Uniform				
Run	t^*	Δ	Δ	Δ	s	S	Δ	Δ	Δ	s	S	Δ	Δ	Δ	s	S		
1	98.1	22%	0.4-4.4	2.8-9.7	18%	1.0-1.8	5.6-7.4	14%	0.3-2.0	2.9-7.9	21%	1.3-4.8	5.0-8.6	14%	0.5-5.0	2.5-9.2		
2	98.1	17%	1.2-6.2	5.1-14.6	9%	1.8-3.3	8.2-11.3	10%	0.8-4.0	5.0-13.4	8%	1.8-6.7	7.2-12.2	3%	0.8-4.9	4.3-12.3		
3	98.1	42%	0.4-2.9	3.0-7.5	42%	0.9-1.6	5.0-7.1	36%	0.4-1.7	3.2-7.4	44%	1.2-4.4	4.6-8.0	40%	0.6-4.3	2.8-7.9		
4	98.1	26%	0.5-2.9	4.4-11.1	22%	0.9-1.7	7.2-9.2	22%	0.4-10.9	4.5-10.8	24%	1.0-5.7	6.4-11.4	19%	0.4-3.9	3.9-9.8		
5	88.9	11%	0.3-2.8	3.4-10.8	16%	1.0-1.6	7.3-9.2	9%	0.2-1.8	3.5-10.6	17%	1.0-5.6	6.5-11.1	8%	0.3-3.3	3.0-9.7		
6	98.1	15%	0.4-4.8	1.1-8.1	13%	1.0-1.9	4.1-5.9	9%	0.1-2.1	1.2-6.6	12%	1.5-4.0	3.7-6.2	7%	0.5-4.1	1.0-6.4		
7	154.4	43%	1.9-5.4	5.3-10.9	32%	1.9-3.2	6.8-9.3	38%	1.3-3.5	5.1-10.2	27%	2.2-4.7	6.0-8.8	31%	1.5-4.0	4.4-9.2		
8	66.9	4%	0.4-11.7	2.3-17.5	-3%	1.1-3.3	6.0-9.9	-1%	0.2-3.3	2.3-9.9	2%	1.4-10.0	5.3-13.9	-3%	0.5-10.0	1.9-13.9		
9	154.4	57%	1.4-4.5	3.7-8.5	55%	1.9-2.7	5.1-6.4	48%	1.2-3.6	3.7-8.1	51%	2.4-3.7	4.7-6.3	48%	1.6-4.5	3.4-7.7		
10	98.1	20%	0.0-3.0	0.8-7.2	26%	1.0-1.6	4.1-5.5	16%	0.0-2.0	0.8-6.6	28%	1.5-4.0	3.8-6.3	19%	0.4-3.1	0.7-6.4		
11	98.1	12%	0.3-2.7	2.6-8.1	11%	1.0-1.8	5.7-7.8	8%	0.3-1.6	2.7-7.7	13%	1.3-4.8	5.1-8.6	8%	0.5-2.9	2.3-7.1		
12	154.4	34%	1.0-2.7	4.3-7.8	33%	1.0-1.6	5.3-6.6	32%	0.7-1.7	4.3-7.2	30%	1.3-2.8	4.7-6.6	28%	0.9-2.5	3.8-6.7		
13	66.9	28%	0.3-6.0	2.3-11.2	22%	1.0-2.7	5.4-8.4	21%	0.2-2.5	2.3-8.2	31%	1.4-9.2	4.9-12.7	21%	0.5-6.3	2.0-10.2		
14	98.1	21%	0.6-6.2	3.6-13.6	11%	1.9-3.3	8.4-11.6	14%	0.5-3.7	3.6-12.8	12%	1.9-6.8	7.3-12.4	13%	0.5-5.0	3.0-11.6		
15	98.1	34%	0.5-4.2	2.6-9.4	32%	1.8-3.0	6.1-8.6	27%	0.5-3.4	2.7-9.3	37%	2.1-5.4	5.6-8.9	30%	0.7-4.2	2.4-8.7		
16	98.1	32%	0.0-6.1	0.3-9.3	25%	1.9-3.2	5.1-7.4	21%	0.0-3.6	0.3-8.3	27%	2.4-5.2	4.7-7.5	18%	0.5-4.6	0.3-7.7		
17	92.4	19%	0.1-4.1	1.5-10.5	17%	0.9-1.7	6.4-8.6	14%	0.1-2.0	1.6-9.6	18%	1.1-5.2	5.7-9.9	14%	0.3-5.2	1.3-10.6		
18	92.4	15%	0.3-5.0	1.6-7.9	5%	1.0-2.4	3.8-6.5	9%	0.2-2.1	1.5-6.2	6%	1.6-5.6	3.4-7.5	8%	0.6-4.0	1.3-6.0		
19	92.4	17%	0.4-3.7	2.6-8.8	10%	1.0-1.9	5.3-7.5	12%	0.3-1.9	2.6-8.1	10%	1.4-4.7	4.7-8.2	8%	0.5-3.3	2.2-7.4		
20	92.4	22%	0.1-1.9	2.2-7.2	20%	0.7-1.3	5.4-6.6	25%	0.2-1.6	2.4-7.4	29%	1.1-6.5	4.8-10.3	27%	0.4-3.1	2.2-7.1		
21	92.4	38%	0.5-5.4	2.0-9.8	39%	1.8-3.1	5.1-7.4	31%	0.4-4.1	2.1-9.1	39%	2.3-5.1	4.7-7.6	31%	0.8-5.1	1.8-8.7		
22	92.4	18%	0.8-7.0	5.3-17.3	18%	1.9-3.5	11.0-14.3	17%	0.5-3.6	5.3-16.7	16%	1.4-9.0	9.4-17.4	12%	0.3-5.9	4.4-14.7		
23	129.8	38%	1.2-6.3	4.1-12.6	32%	1.9-2.6	6.4-7.8	30%	1.0-4.6	4.1-11.8	32%	2.2-4.2	5.7-7.8	29%	1.2-5.5	3.6-11.0		
24	65.4	32%	0.2-4.3	1.3-9.6	32%	2.0-3.5	6.6-9.4	39%	0.1-3.2	1.3-9.1	35%	2.3-7.2	5.9-10.8	23%	0.4-3.7	1.1-8.3		
25	67.9	20%	0.3-5.5	2.3-10.5	18%	1.0-2.4	5.6-8.3	13%	0.2-2.2	2.3-8.1	21%	1.4-6.3	5.0-10.3	15%	0.5-6.0	2.0-9.7		
26	225.7	77%	2.7-4.5	6.8-9.9	61%	2.3-2.9	7.4-8.3	61%	2.0-3.3	6.4-9.3	58%	2.7-3.9	6.7-7.8	54%	2.3-3.9	5.8-8.6		
27	92.4	14%	0.7-5.4	2.8-10.2	13%	1.8-3.1	6.2-8.7	9%	0.5-3.2	2.9-9.0	13%	2.1-5.7	5.5-9.2	8%	0.7-4.4	2.5-8.5		
Avg Difference=		27%			23%			22%			24%			20%				

	P10 Fixed	P10 Normal	P10 Uniform			P11 Fixed	P11 Normal	P11 Uniform			P12 Fixed	P12 Normal	P12 Uniform			P13 Normal		
Run	Δ	Δ	Δ	s	S	Δ	Δ	Δ	s	S	Δ	Δ	Δ	s	S	Δ	s	S
1	5%	14%	9%	0.4-1.6	2.9-5.9	24%	11%	6%	0.3-0.8	2.8-5.5	4%	11%	5%	0.4-1.1	2.4-4.9	84%	0.4-31.2	2.9-40.5
2	-1%	1%	1%	0.7-2.6	3.9-8.7	27%	6%	2%	0.5-1.8	4.0-8.4	-1%	4%	0%	0.4-1.8	3.3-7.5	73%	0.8-38.0	4.4-48.6
3	18%	25%	26%	0.4-1.6	2.6-5.5	42%	21%	19%	0.3-0.9	2.6-5.1	10%	16%	20%	0.4-1.1	2.2-4.7	80%	0.4-12.0	2.6-19.1
4	8%	13%	16%	0.4-1.6	3.5-7.6	41%	10%	16%	0.2-0.9	3.6-7.2	4%	8%	11%	0.2-0.7	3.0-6.3	67%	0.4-14.2	3.5-25.0
5	4%	8%	1%	0.3-1.6	3.3-7.6	31%	6%	2%	0.2-0.9	3.4-7.3	4%	3%	1%	0.2-0.7	2.8-6.4	63%	0.4-27.8	3.7-40.5
6	-11%	-2%	6%	0.4-1.6	2.0-4.4	6%	-5%	1%	0.3-0.8	1.9-3.9	-11%	-3%	-4%	0.6-1.4	1.7-3.7	129%	0.4-35.1	2.0-41.2
7	20%	24%	27%	0.8-3.8	3.4-9.2	40%	23%	27%	0.6-2.7	3.3-8.5	17%	14%	29%	0.8-3.2	2.9-7.9	33%	0.9-14.1	3.6-20.9
8	-9%	0%	2%	0.5-1.2	2.8-5.0	11%	4%	5%	0.3-0.7	2.8-4.7	-4%	5%	6%	0.4-0.9	2.4-4.2	147%	0.4-69.3	2.6-80.1
9	29%	36%	37%	0.8-3.1	2.5-6.3	39%	31%	30%	0.6-2.2	2.5-5.8	23%	31%	31%	1.0-2.9	2.3-5.6	44%	0.6-6.4	2.2-9.9
10	5%	13%	10%	0.4-1.6	1.8-4.5	16%	9%	7%	0.2-0.9	1.8-4.0	1%	7%	6%	0.5-1.4	1.6-3.8	79%	0.3-13.6	1.8-18.7
11	-2%	4%	4%	0.4-1.6	2.6-6.0	12%	4%	3%	0.2-0.9	2.6-5.6	-4%	2%	2%	0.4-1.1	2.2-5.0	78%	0.4-31.2	2.7-40.5
12	14%	24%	21%	0.4-2.2	2.4-6.8	34%	17%	16%	0.2-1.2	2.3-6.1	6%	15%	14%	0.4-1.6	2.0-5.6	34%	0.4-10.0	2.6-17.0
13	6%	11%	17%	0.4-1.2	2.3-4.7	25%	8%	17%	0.2-0.7	2.3-4.4	12%	3%	14%	0.4-1.0	2.0-4.0	192%	0.3-53.7	2.3-63.4
14	3%	5%	9%	0.7-2.4	4.1-8.4	30%	14%	12%	0.6-1.7	4.2-8.1	8%	18%	12%	0.5-1.7	3.5-7.2	65%	07-19.5	4.2-28.5
15	11%	15%	15%	0.7-2.2	2.9-6.1	29%	15%	15%	0.5-1.7	2.9-5.8	8%	9%	11%	0.7-1.9	2.6-5.4	101%	0.7-21.6	3.0-28.6
16	7%	17%	14%	0.8-2.4	2.4-5.2	16%	22%	19%	0.6-1.7	2.3-4.8	5%	14%	15%	0.9-2.3	2.1-4.5	125%	0.7-36.8	2.4-41.8
17	4%	11%	8%	0.4-1.5	3.2-6.6	33%	5%	4%	0.2-0.8	3.2-6.3	3%	1%	2%	0.3-0.9	2.7-5.5	89%	0.4-28.8	3.0-39.8
18	-5%	-2%	3%	0.4-1.5	1.8-4.0	10%	-6%	5%	0.2-0.8	1.7-3.5	-9%	-6%	1%	0.6-1.5	1.5-3.3	111%	0.4-36.9	1.8-42.3
19	-4%	5%	6%	0.4-1.6	2.3-5.7	20%	3%	9%	0.2-0.9	2.3-5.2	-2%	4%	6%	0.4-1.2	2.0-4.7	113%	0.4-37.9	2.5-46.6
20	9%	13%	15%	0.4-1.4	2.7-5.4	28%	7%	11%	0.3-0.8	2.7-5.0	5%	5%	9%	0.3-0.9	2.3-4.6	70%	0.3-32.4	2.2-42.4
21	9%	20%	19%	0.6-2.4	2.2-5.4	27%	20%	17%	0.5-1.7	2.2-5.0	5%	16%	16%	0.8-2.2	2.0-4.7	104%	0.8-16.0	2.7-20.6
22	7%	14%	12%	0.7-2.5	5.7-11.4	39%	21%	18%	0.6-1.8	5.8-11.2	31%	37%	32%	0.2-1.1	4.8-9.7	78%	0.8-48.4	6.0-63.1
23	15%	24%	21%	0.7-3.1	3.0-7.8	32%	20%	21%	0.6-2.3	3.0-7.2	11%	19%	17%	0.7-2.7	2.7-6.7	46%	0.8-16.8	3.2-22.9
24	15%	18%	27%	0.7-1.9	3.0-5.5	44%	38%	41%	0.5-1.4	3.0-5.2	13%	24%	27%	0.7-1.6	2.6-4.7	180%	0.8-32.9	3.2-40.1
25	3%	4%	11%	0.3-1.2	2.3-4.8	24%	5%	11%	0.2-0.7	2.3-4.5	7%	7%	8%	0.4-0.9	2.0-4.1	107%	0.4-31.4	2.7-40.1
26	29%	41%	42%	0.8-4.0	3.2-9.3	50%	34%	35%	0.6-2.9	3.2-8.5	19%	31%	29%	0.8-3.4	2.8-8.0	30%	0.7-3.7	3.0-8.8
27	-2%	5%	1%	0.7-2.4	2.9-6.4	13%	6%	2%	0.5-1.7	2.9-6.1	-2%	2%	0%	0.7-2.0	2.5-5.6	65%	0.7-28.6	3.1-35.9
	7%	13%	14%			28%	13%	14%			6%	11%	12%			88%		

Appendix E (continued): Kabir and Al-Olayan Comparison for β=2.0

Policy (P)		P13 Uniform			P14 Normal			P14 Uniform			P15 Normal			P15 Uniform			P16 Fixed	P16 Normal
Run	t^*	Δ	s	S	Δ	s	S	Δ	s	S	Δ	s	S	Δ	s	S	Δ	Δ
1	98.1	45%	0.3-15.8	2.1-23.5	30%	0.3-7.9	2.8-20.0	19%	0.2-4.8	2.1-13.8	71%	0.4-29.7	2.4-35.7	36%	0.3-16.5	1.8-21.7	51%	51%
2	98.1	29%	0.5-19.3	3.3-28.1	15%	0.6-8.7	4.4-22.2	12%	0.4-5.9	3.3-16.8	40%	0.6-26.6	3.8-33.8	15%	0.4-14.6	2.8-21.2	9%	12%
3	98.1	64%	0.3-11.9	2.3-18.9	45%	0.3-4.1	2.5-12.1	41%	0.2-4.0	2.3-12.0	95%	0.4-15.0	2.2-20.0	71%	0.4-14.9	2.0-19.9	47%	41%
4	98.1	42%	0.4-14.2	3.3-24.5	31%	0.2-4.8	3.6-16.2	24%	0.2-4.5	3.3-15.2	89%	0.2-20.7	3.0-28.7	43%	0.2-20.7	2.8-28.2	45%	41%
5	88.9	32%	0.3-14.2	2.9-24.9	24%	0.3-7.5	3.7-21.6	14%	0.2-4.7	2.9-16.1	87%	0.2-37.9	3.1-46.6	45%	0.2-20.7	2.4-28.6	41%	33%
6	98.1	48%	0.3-17.9	1.7-23.0	27%	0.2-0.8	1.9-18.0	11%	0.5-5.0	1.7-12.3	61%	0.6-22.3	1.7-25.7	22%	0.5-12.8	1.5-15.9	-2%	10%
7	154.4	32%	0.7-8.9	3.2-14.3	22%	0.7-5.8	3.6-14.8	29%	0.5-4.0	3.2-11.2	19%	0.8-9.4	3.1-14.8	24%	0.7-5.9	2.8-10.5	46%	47%
8	66.9	75%	0.3-69.3	2.0-80.1	22%	0.2-12.8	2.6-31.7	11%	0.1-12.8	2.0-31.7	79%	0.4-47.3	2.2-53.7	39%	0.4-47.3	1.7-53.7	20%	36%
9	154.4	42%	0.6-6.4	2.2-10.4	32%	0.5-3.7	2.2-8.2	33%	0.5-4.1	2.2-8.9	35%	0.8-5.3	2.0-8.2	35%	0.8-5.7	2.0-8.9	74%	76%
10	98.1	48%	0.2-13.7	1.4-18.9	34%	0.2-4.3	1.7-10.7	25%	0.1-4.4	1.4-11.0	66%	0.5-12.4	1.5-15.7	42%	0.5-12.6	1.2-15.9	34%	34%
11	98.1	24%	0.4-6.7	2.6-12.9	21%	0.2-7.9	2.7-19.9	12%	0.2-2.9	2.6-9.9	68%	0.4-29.7	2.3-35.7	22%	0.4-7.3	2.2-12.3	33%	34%
12	154.4	24%	0.4-6.7	2.3-12.3	18%	0.3-4.4	2.6-12.8	15%	0.2-2.8	2.3-9.1	28%	0.5-10.1	2.2-15.5	15%	0.4-6.9	2.0-11.2	33%	33%
13	66.9	92%	0.2-31.5	1.9-39.8	50%	0.2-11.3	2.3-26.0	34%	0.1-7.8	1.9-19.0	150%	0.4-43.0	1.9-48.7	84%	0.4-27.5	1.6-32.7	106%	89%
14	98.1	54%	0.4-19.6	3.1-28.3	16%	0.6-6.6	4.2-18.6	18%	0.3-6.1	3.1-16.9	36%	0.4-14.8	3.6-21.7	31%	0.4-14.8	2.6-21.5	10%	10%
15	98.1	53%	0.5-12.8	2.3-18.4	42%	0.6-6.8	3.0-15.5	30%	0.4-5.2	2.3-12.4	79%	0.7-17.3	2.7-22.2	44%	0.5-10.9	2.0-14.9	29%	28%
16	98.1	68%	0.2-19.0	1.2-23.3	33%	0.6-8.1	2.3-16.5	27%	0.2-5.8	1.3-12.2	48%	0.9-16.8	2.1-19.7	29%	0.6-10.4	1.1-13.2	43%	45%
17	92.4	52%	0.1-28.8	1.7-40.0	30%	0.2-7.5	3.0-20.4	22%	0.1-7.7	1.7-20.8	105%	0.3-33.7	2.5-41.0	57%	0.3-33.8	1.4-41.3	53%	42%
18	92.4	61%	0.3-18.9	1.4-23.5	19%	0.2-8.1	1.7-18.0	15%	0.2-5.3	1.3-12.5	43%	0.6-20.9	1.5-23.8	26%	0.5-12.3	1.2-15.0	0%	1%
19	92.4	44%	0.3-10.1	2.3-16.3	28%	0.2-8.5	2.5-21.7	16%	0.2-3.5	2.2-11.0	71%	0.4-28.6	2.1-34.0	27%	0.4-9.0	1.9-13.5	22%	25%
20	92.4	38%	0.3-9.0	2.2-16.2	27%	0.5-5.3	2.3-18.1	24%	0.2-2.7	2.2-10.6	104%	0.3-41.2	2.0-47.6	49%	0.3-14.6	1.9-19.9	47%	41%
21	92.4	88%	0.5-17.3	2.1-21.9	46%	0.6-5.5	2.6-11.7	35%	0.4-5.2	2.0-11.2	61%	1.0-10.6	2.4-13.7	48%	0.7-11.0	1.8-14.0	55%	61%
22	92.4	30%	0.5-21.6	4.8-34.7	23%	0.6-10.9	6.0-31.7	18%	0.4-6.3	4.9-23.0	57%	0.2-38.9	5.0-51.3	22%	0.1-19.2	4.0-29.5	10%	14%
23	129.8	54%	0.7-22.1	2.9-29.3	25%	0.6-5.3	3.1-12.8	29%	0.5-7.6	2.9-17.1	30%	0.8-11.7	2.8-15.9	33%	0.7-15.4	2.6-20.7	43%	43%
24	65.4	87%	0.4-16.8	2.1-22.9	55%	0.6-8.0	3.2-17.8	39%	0.3-5.7	2.1-13.5	97%	0.8-20.2	2.8-24.8	53%	0.5-11.7	1.8-16.0	33%	33%
25	67.9	73%	0.1-20.8	1.4-28.9	30%	0.3-7.8	2.7-19.3	25%	0.1-6.3	1.4-17.0	87%	0.5-28.2	2.3-33.7	58%	0.3-20.0	1.1-25.5	71%	64%
26	225.7	28%	0.8-4.0	3.2-9.3	33%	0.6-2.8	3.0-8.4	25%	0.6-3.1	3.2-8.9	25%	0.7-3.3	2.6-7.7	23%	0.8-3.6	2.8-8.1	64%	67%
27	92.4	33%	0.4-16.9	2.2-23.0	19%	0.6-8.0	3.0-17.8	12%	0.3-5.6	2.2-13.5	34%	0.8-18.5	2.7-23.5	17%	0.5-11.8	1.9-16.1	18%	21%
Avg Difference=		50%			30%			23%			65%			37%			38%	38%

	P16 Uniform			P17 Normal			P17 Uniform			P18 Fixed	P18 Normal	P18 Uniform			P19 Normal	P19 Uniform		
Run	Δ	s	S	Δ	s	S	Δ	s	S	Δ	Δ	Δ	s	S	Δ	Δ	s	S
1	44%	0	1.0-2.0	15%	0.0-3.0	1.0-5.0	22%	0.0-2.0	1.0-4.0	65%	68%	67%	0.0-4.0	0.0-11.0	32%	29%	1.0-3.0	6.0-9.0
2	11%	4.0-5.0	6.0-7.0	16%	4.0-7.0	5.0-9.0	14%	4.0-7.0	5.0-8.0	67%	71%	71%	0.0-7.0	0.0-17.0	24%	23%	0.0-5.0	0.0-13.0
3	42%	0	1.0-2.0	17%	0.0-1.0	1.0-4.0	27%	0.0-1.0	1.0-4.0	106%	109%	108%	0.0-3.0	0.0-10.0	48%	47%	0.0-2.0	0.0-8.0
4	47%	0	1.0-2.0	15%	0.0-2.0	1.0-4.0	33%	0.0-1.0	1.0-4.0	66%	68%	71%	3.0-4.0	11.0-14.0	29%	32%	0.0-3.0	0.0-11.0
5	28%	0	1.0-2.0	11%	0.0-2.0	1.0-5.0	15%	0.0-1.0	1.0-4.0	50%	48%	44%	3.0-4.0	11.0-14.0	18%	16%	0.0-3.0	0.0-10.0
6	10%	0	1.0-2.0	-11%	0.0-3.0	1.0-4.0	1%	0.0-2.0	1.0-4.0	73%	76%	78%	0.0-4.0	0.0-9.0	24%	25%	1.0-3.0	5.0-7.0
7	51%	4.0-6.0	6.0-7.0	46%	4.0-6.0	6.0-8.0	51%	4.0-6.0	6.0-8.0	108%	110%	114%	0.0-9.0	0.0-16.0	54%	60%	3.0-5.0	9.0-11.0
8	33%	0	1.0-2.0	-12%	0.0-5.0	1.0-6.0	17%	0.0-5.0	1.0-6.0	24%	25%	27%	0.0-4.0	0.0-11.0	4%	7%	0.0-5.0	0.0-11.0
9	77%	4	5.0-6.0	76%	3.0-6.0	5.0-7.0	78%	3.0-6.0	5.0-7.0	156%	165%	163%	0.0-7.0	0.0-12.0	72%	77%	2.0-3.0	6.0-7.0
10	32%	0	1.0-2.0	4%	0.0-2.0	1.0-4.0	10%	0.0-2.0	1.0-4.0	68%	74%	75%	0.0-4.0	0.0-9.0	28%	27%	1.0-2.0	4.0-6.0
11	34%	0	1.0-2.0	5%	0.0-3.0	1.0-5.0	18%	0.0-1.0	1.0-3.0	39%	41%	44%	0.0-4.0	0.0-11.0	17%	18%	1.0-3.0	6.0-9.0
12	38%	0	1.0-3.0	36%	0.0-1.0	1.0-4.0	29%	0.0-1.0	1.0-3.0	105%	110%	109%	3.0-5.0	9.0-11.0	54%	54%	1.0-2.0	6.0-8.0
13	95%	0	1.0-2.0	1%	0.0-5.0	1.0-6.0	40%	0.0-3.0	1.0-5.0	68%	68%	72%	3.0-4.0	9.0-10.0	30%	38%	0.0-4.0	0.0-10.0
14	13%	4.0-5.0	6.0-7.0	15%	4.0-7.0	5.0-9.0	18%	4.0-7.0	5.0-9.0	59%	59%	62%	0.0-7.0	0.0-17.0	21%	25%	0.0-5.0	0.0-13.0
15	28%	3.0-4.0	5.0-6.0	40%	3.0-7.0	5.0-9.0	40%	3.0-6.0	5.0-8.0	98%	101%	102%	0.0-5.0	0.0-12.0	37%	41%	2.0-4.0	6.0-9.0
16	45%	4.0-5.0	5.0-6.0	51%	4.0-7.0	5.0-8.0	48%	4.0-7.0	5.0-8.0	100%	108%	108%	5.0-7.0	9.0-12.0	37%	36%	2.0-4.0	6.0-9.0
17	42%	0	1.0-2.0	4%	0.0-3.0	1.0-5.0	23%	0.0-3.0	1.0-5.0	68%	67%	67%	0.0-4.0	0.0-12.0	22%	27%	1.0-3.0	7.0-10.0
18	8%	0	1.0-2.0	-13%	0.0-3.0	1.0-5.0	-1%	0.0-2.0	1.0-4.0	61%	60%	63%	3.0-4.0	7.0-9.0	16%	19%	1.0-4.0	5.0-7.0
19	25%	0	1.0-3.0	-3%	0.0-3.0	1.0-5.0	22%	0.0-2.0	1.0-4.0	62%	63%	63%	0.0-5.0	0.0-11.0	24%	26%	1.0-3.0	6.0-9.0
20	46%	0	1.0-2.0	15%	0.0-5.0	0.0-6.0	28%	0	0.0-3.0	61%	59%	65%	2.0-3.0	8.0-9.0	22%	28%	0.0-3.0	5.0-7.0
21	58%	4	5.0-6.0	72%	4.0-7.0	5.0-8.0	66%	3.0-7.0	5.0-8.0	130%	139%	139%	4.0-6.0	9.0-11.0	49%	48%	2.0-4.0	6.0-9.0
22	12%	4.0-5.0	6.0-7.0	16%	4.0-7.0	6.0-9.0	13%	4.0-7.0	5.0-9.0	45%	49%	46%	6.0-8.0	18.0-22.0	25%	21%	3.0-6.0	12.0-16.0
23	43%	4.0-5.0	5.0-7.0	47%	4.0-7.0	5.0-8.0	47%	4.0-7.0	5.0-8.0	108%	114%	113%	0.0-7.0	0.0-14.0	50%	50%	3.0-4.0	8.0-9.0
24	36%	4	5.0-6.0	45%	4.0-7.0	5.0-8.0	43%	4.0-7.0	5.0-8.0	92%	95%	97%	0.0-6.0	0.0-12.0	45%	49%	0.0-5.0	0.0-11.0
25	68%	0	1.0-2.0	-1%	0.0-3.0	1.0-5.0	28%	0.0-3.0	1.0-5.0	51%	55%	55%	0.0-4.0	0.0-10.0	24%	26%	1.0-3.0	6.0-9.0
26	66%	4.0-5.0	5.0-7.0	65%	4.0-5.0	5.0-7.0	65%	4.0-5.0	5.0-7.0	158%	167%	166%	0.0-8.0	0.0-15.0	96%	96%	0.0-4.0	0.0-10.0
27	18%	4	5.0-6.0	26%	4.0-7.0	5.0-8.0	23%	4.0-7.0	5.0-8.0	62%	67%	65%	0.0-7.0	0.0-13.0	24%	22%	2.0-4.0	7.0-10.0
	39%			22%			30%			80%	83%	83%			34%	36%		

Appendix F: Kabir and Al-Olayan Comparison for β=3.0

Policy (P)		P1 Fixed	P1 Normal	P1 Uniform			P2 Fixed	P2 Normal	P2 Uniform			P3 Fixed	P3 Normal	P3 Uniform		
Run	t^*	Δ	Δ	Δ	s	S	Δ	Δ	Δ	s	S	Δ	Δ	Δ	s	S
1	76.0	11%	11%	12%	1.7-2.5	5.3-6.8	6%	8%	9%	0.8-1.2	4.5-5.7	10%	11%	11%	4.9-6.3	1.8-2.9
2	76.0	19%	20%	17%	2.0-3.0	7.0-8.8	12%	13%	13%	0.9-1.3	5.9-7.4	17%	19%	18%	6.3-8.2	1.8-3.3
3	76.0	55%	51%	52%	1.8-3.0	5.1-6.8	33%	28%	31%	1.3-1.7	4.9-5.9	40%	35%	37%	4.8-6.3	2.0-3.2
4	76.0	25%	25%	25%	2.8-5.5	8.1-11.8	15%	15%	16%	1.4-2.4	6.4-8.9	19%	19%	18%	7.2-10.7	2.6-5.6
5	71.5	17%	17%	21%	1.8-7.6	7.4-15.1	12%	9%	15%	1.0-2.1	6.2-9.1	23%	22%	27%	7.3-17.2	2.3-12.1
6	76.0	28%	24%	25%	2.0-3.8	4.4-6.8	16%	10%	9%	0.9-1.5	3.5-4.8	22%	19%	20%	3.5-5.3	1.9-3.4
7	99.2	18%	18%	18%	1.6-2.2	5.1-6.0	12%	18%	14%	0.8-1.0	4.5-5.1	7%	8%	7%	4.2-5.0	1.4-2.0
8	59.8	13%	11%	13%	2.8-5.9	7.2-10.9	-1%	-5%	0%	1.3-2.1	5.5-7.7	3%	2%	5%	6.1-9.0	2.4-5.3
9	99.2	23%	21%	24%	0.8-1.4	3.0-3.9	16%	12%	16%	0.6-0.9	2.6-3.7	23%	21%	24%	2.8-3.7	1.1-1.8
10	71.5	17%	19%	13%	1.4-2.9	3.8-6.0	14%	16%	10%	0.7-1.3	3.2-4.6	25%	27%	23%	3.5-5.5	1.9-3.4
11	71.5	15%	13%	10%	1.6-2.7	5.7-7.1	5%	3%	1%	0.9-1.3	4.9-6.0	11%	10%	6%	5.1-6.6	1.6-3.1
12	99.2	23%	23%	25%	1.2-1.7	4.3-5.1	17%	16%	15%	0.7-0.9	4.0-4.6	16%	15%	19%	3.8-4.6	1.3-1.9
13	59.8	93%	92%	96%	4.6-19.1	8.7-24.4	31%	32%	34%	1.9-4.5	5.8-10.9	58%	58%	60%	7.0-15.4	3.7-12.1
14	71.5	6%	6%	7%	1.6-2.4	6.9-8.1	1%	0%	4%	0.9-1.2	6.0-7.3	5%	4%	6%	6.0-7.3	1.3-2.4
15	71.5	23%	26%	24%	1.3-3.5	4.9-7.9	9%	11%	9%	0.6-1.3	4.1-5.8	24%	28%	26%	4.9-8.4	1.8-5.2
16	71.5	32%	33%	34%	3.8-8.8	6.1-12.0	6%	9%	9%	1.5-2.9	4.1-6.9	7%	8%	8%	4.1-7.3	2.6-5.3
17	73.3	37%	37%	41%	2.8-5.4	7.1-10.1	22%	20%	20%	1.4-2.3	5.8-7.8	33%	30%	31%	6.1-9.2	2.7-5.0
18	73.3	38%	38%	38%	4.0-8.7	6.4-11.4	3%	6%	2%	1.7-2.8	4.2-6.3	4%	6%	3%	4.3-6.5	2.9-4.8
19	73.3	53%	50%	51%	3.7-8.9	7.1-13.3	21%	17%	18%	1.6-2.9	5.2-8.0	25%	22%	21%	5.1-8.8	2.6-5.7
20	73.3	15%	16%	14%	0.8-2.9	4.3-7.6	11%	12%	10%	0.0-0.7	3.5-5.0	36%	37%	34%	5.2-9.5	1.9-6.1
21	73.3	65%	65%	62%	3.1-7.1	5.9-10.3	34%	33%	31%	1.4-2.4	4.1-6.1	39%	39%	36%	4.7-7.3	2.8-5.1
22	73.3	11%	11%	6%	3.8-6.6	10.8-15.3	2%	1%	-4%	1.5-2.5	8.5-11.6	3%	4%	-1%	9.0-13.0	2.9-5.9
23	89.8	19%	20%	17%	1.7-3.9	5.2-7.7	8%	11%	6%	0.9-1.4	4.3-5.4	14%	10%	8%	4.7-6.9	1.8-4.5
24	59.0	40%	39%	39%	2.2-6.1	6.3-11.1	20%	17%	19%	1.2-2.1	5.4-7.4	29%	27%	34%	5.7-10.2	2.3-6.7
25	60.4	30%	31%	27%	2.3-5.1	6.4-9.9	14%	16%	10%	1.2-2.0	5.1-7.2	31%	32%	27%	5.8-9.1	2.4-5.6
26	121.7	32%	34%	33%	1.5-2.9	4.4-6.5	23%	25%	26%	1.0-1.7	4.1-5.6	21%	23%	24%	3.7-5.4	1.4-2.5
27	73.3	32%	32%	33%	3.1-4.6	6.6-8.6	14%	13%	14%	1.6-2.1	5.3-6.5	19%	17%	19%	5.3-6.8	2.6-3.7
Avg Difference=		29%	29%	29%			14%	14%	13%			21%	20%	20%		

	P4 Fixed	P4 Normal	P4 Uniform			P5 Fixed	P5 Normal	P5 Uniform			P6 Fixed	P6 Normal	P6 Uniform			P7 Normal		
Run	Δ	Δ	Δ	s	S	Δ	Δ	Δ	s	S	Δ	Δ	Δ	s	S	Δ	s	S
1	72%	74%	95%	3.7-7.9	9.8-14.4	38%	41%	61%	2.2-3.2	8.9-10.9	68%	71%	98%	3.5-8.6	8.8-13.7	17%	1.4-5.7	5.4-11.5
2	91%	89%	121%	4.1-9.1	12.4-18.1	47%	49%	81%	2.2-3.4	11.4-13.8	87%	85%	122%	3.4-10.5	11.0-17.7	26%	1.5-6.7	7.1-14.5
3	201%	192%	208%	5.2-10.1	10.6-16.1	126%	126%	143%	2.8-5.0	10.0-12.1	179%	176%	198%	5.1-9.4	9.9-14.0	53%	1.8-12.0	5.5-17.2
4	124%	123%	142%	5.6-11.6	13.8-20.4	72%	76%	89%	3.9-5.3	13.0-15.6	114%	112%	129%	5.0-11.3	12.5-18.6	41%	2.1-14.3	7.7-22.2
5	70%	65%	86%	3.7-7.0	11.9-16.0	43%	42%	59%	2.2-3.1	11.2-13.0	85%	79%	96%	4.1-10.3	11.6-17.6	41%	1.9-8.9	8.8-17.6
6	131%	130%	179%	4.1-8.4	8.0-12.7	61%	58%	99%	2.2-3.3	6.9-8.8	100%	93%	156%	3.7-6.9	6.7-9.8	30%	1.5-6.8	4.2-10.5
7	98%	97%	137%	3.8-10.2	10.0-16.8	52%	55%	88%	2.1-3.5	9.1-11.8	76%	75%	123%	3.0-8.9	8.3-13.9	31%	1.8-6.4	6.2-11.7
8	69%	63%	108%	5.4-10.2	11.5-16.9	24%	23%	61%	2.5-3.5	9.1-11.8	50%	49%	100%	4.6-8.9	9.7-14.0	14%	1.7-15.4	6.0-21.6
9	126%	125%	147%	3.0-6.2	7.1-10.5	82%	85%	105%	2.0-3.0	6.6-8.4	120%	119%	146%	3.2-6.5	6.4-9.5	38%	1.4-3.9	4.2-7.3
10	106%	104%	119%	3.5-6.8	7.5-11.2	51%	57%	72%	2.2-3.0	6.7-8.3	98%	96%	119%	3.8-7.0	6.9-10.1	62%	1.9-8.3	5.2-12.6
11	57%	55%	70%	4.0-8.1	10.1-14.7	28%	29%	41%	2.3-3.3	9.2-10.9	56%	55%	72%	4.0-8.9	9.2-13.9	33%	2.1-9.7	7.1-16.2
12	120%	120%	151%	3.4-7.5	8.9-13.4	74%	75%	105%	2.1-3.2	8.2-10.2	110%	107%	147%	3.1-7.7	7.8-12.2	37%	1.5-3.8	5.5-8.5
13	196%	195%	233%	7.5-15.5	13.1-21.6	93%	96%	129%	4.3-5.7	10.8-13.4	136%	137%	181%	6.2-11.0	10.7-15.5	83%	2.4-12.4	6.3-17.5
14	63%	58%	86%	4.5-5.5	12.8-17.4	28%	28%	53%	2.3-3.3	11.6-13.4	59%	57%	89%	4.0-9.6	11.6-16.9	35%	2.3-10.2	9.1-18.9
15	91%	93%	107%	3.1-6.1	8.6-12.1	61%	64%	77%	2.1-2.9	8.1-9.6	110%	109%	128%	3.7-8.1	8.4-12.8	60%	1.7-7.1	6.2-13.1
16	169%	168%	209%	7.0-15.3	11.1-19.6	69%	70%	105%	4.2-5.6	9.3-11.9	91%	93%	140%	5.6-9.1	8.7-12.1	135%	3.6-29.7	6.9-34.4
17	148%	144%	166%	5.9-12.4	13.2-20.1	83%	80%	103%	4.0-5.5	12.0-14.6	123%	119%	146%	5.2-11.0	11.7-17.1	42%	2.2-14.6	7.0-21.6
18	166%	162%	198%	7.0-14.7	10.6-18.6	64%	66%	96%	4.1-5.5	8.6-11.4	83%	80%	136%	5.5-8.5	8.0-11.1	36%	2.5-18.9	4.9-22.3
19	176%	168%	215%	7.2-18.0	12.8-24.2	81%	79%	119%	4.1-6.1	10.9-14.7	103%	103%	159%	5.4-11.1	10.2-15.7	47%	2.6-12.2	6.4-17.2
20	70%	69%	78%	2.3-4.2	7.9-10.4	57%	58%	64%	1.8-2.3	8.0-9.3	108%	106%	112%	3.4-7.5	8.6-12.6	19%	0.7-4.7	4.5-10.1
21	232%	227%	256%	5.7-12.4	10.0-17.0	121%	123%	147%	4.0-5.3	8.9-11.3	168%	163%	199%	5.2-9.0	8.6-12.2	64%	2.1-9.6	5.0-13.3
22	64%	63%	76%	7.2-17.6	19.4-30.6	28%	31%	41%	4.1-6.0	17.8-21.9	52%	50%	66%	5.1-15.6	16.7-26.5	13%	2.7-13.3	10.8-24.4
23	95%	91%	113%	3.5-8.1	9.2-14.3	55%	56%	76%	2.1-3.3	8.5-10.6	90%	86%	111%	3.2-8.5	8.2-13.2	37%	1.7-5.4	6.0-10.5
24	116%	111%	153%	4.4-8.0	10.0-14.1	60%	57%	96%	2.3-3.3	8.7-10.3	110%	106%	155%	4.5-8.4	9.2-13.0	37%	1.5-8.8	5.5-14.7
25	94%	92%	120%	4.3-8.0	10.0-14.1	48%	49%	71%	2.3-3.3	8.7-10.5	89%	88%	118%	4.4-8.4	9.1-13.0	37%	1.5-9.7	5.5-15.6
26	191%	194%	224%	5.2-12.6	10.9-18.6	118%	120%	150%	3.8-5.8	10.2-13.7	149%	149%	175%	4.5-9.4	9.5-14.3	63%	2.6-5.2	6.7-9.8
27	153%	148%	175%	6.3-13.6	12.0-19.7	77%	75%	100%	4.0-5.7	10.6-13.5	109%	105%	138%	5.3-10.1	10.1-14.7	39%	2.3-10.1	6.1-15.0
	122%	119%	147%			64%	65%	90%			101%	99%	132%			43%		

Appendix F (continued): Kabir and Al-Olayan Comparison for β=3.0

Policy (P)		P7 Uniform			P8 Normal			P8 Uniform			P9 Normal			P9 Uniform		
Run	t^*	Δ	D	D	σ	S	D	Δ	D	s	Σ	D	D	Δ	s	S
1	76.0	-3%	0.0-5.9	1.0-11.6	8%	0.9-2.5	5.3-9.1	-3%	0.0-2.4	1.0-8.6	15%	1.2-6.5	4.8-10.8	-7%	0.3-6.6	0.9-11.0
2	76.0	2%	0.0-6.8	1.3-14.7	14%	0.9-2.6	6.9-11.6	2%	0.0-2.6	1.4-11.0	21%	1.0-8.0	6.1-14.0	0%	0.3-7.9	1.1-14.2
3	76.0	26%	0.1-8.6	1.2-13.9	34%	1.6-3.1	5.6-9.0	26%	0.1-3.6	1.3-9.7	45%	1.9-10.2	5.2-13.8	19%	0.4-8.0	1.1-12.1
4	76.0	15%	0.1-5.4	1.5-12.2	19%	1.7-3.9	7.6-12.0	10%	0.1-2.8	1.6-10.0	31%	1.7-3.7	6.8-19.3	5%	0.3-5.4	1.3-11.1
5	71.5	19%	0.0-6.5	1.1-15.1	26%	1.3-3.3	8.7-12.3	15%	0.0-2.8	1.2-11.8	49%	1.4-13.6	7.8-20.1	22%	0.2-9.6	1.0-16.6
6	76.0	-4%	0.0-4.1	0.7-7.4	10%	0.9-2.4	3.9-6.9	-6%	0.0-1.8	0.7-5.6	17%	1.5-5.6	3.6-8.0	-14%	0.4-3.6	0.6-6.0
7	99.2	25%	0.5-4.9	2.7-10.3	20%	1.0-2.3	5.8-8.3	16%	0.3-2.1	2.7-8.3	21%	1.3-5.7	5.1-9.7	15%	0.5-4.3	2.3-8.7
8	59.8	7%	0.0-10.6	0.5-16.7	-3%	1.0-3.7	5.7-11.7	-7%	0.0-3.3	0.5-10.9	3%	1.3-12.7	5.0-16.7	-3%	0.4-9.2	0.4-13.5
9	99.2	30%	0.3-3.7	1.7-7.1	33%	1.0-1.8	4.1-5.7	22%	0.2-1.9	1.8-6.0	40%	1.5-4.3	3.9-6.7	28%	0.6-4.2	1.6-6.5
10	71.5	17%	0.0-5.2	0.6-9.2	34%	1.3-2.9	4.9-8.1	5%	0.0-2.4	0.6-7.1	58%	2.0-8.2	4.7-11.0	16%	0.4-5.5	0.5-8.4
11	71.5	5%	0.0-8.5	0.7-15.0	14%	1.3-3.3	6.8-10.6	-3%	0.0-3.3	0.8-10.4	32%	1.8-10.5	6.2-15.3	1%	0.3-9.3	0.6-14.1
12	99.2	29%	0.3-3.8	2.4-8.5	28%	1.0-1.8	5.3-7.0	20%	0.3-1.8	2.4-7.1	29%	1.4-4.1	4.8-7.7	21%	0.5-4.1	2.1-7.7
13	59.8	65%	0.0-15.4	0.6-21.3	30%	1.8-4.6	6.1-11.3	26%	0.0-5.5	0.6-12.8	45%	2.2-8.7	5.5-12.3	35%	0.4-10.8	0.5-15.2
14	71.5	10%	0.0-7.7	1.1-16.1	12%	1.3-3.5	8.8-13.6	3%	0.0-3.0	1.1-12.3	33%	1.5-12.1	7.8-18.6	7%	0.3-8.8	0.9-15.6
15	71.5	19%	0.0-5.4	0.3-11.1	37%	1.3-2.8	6.2-9.3	12%	0.0-2.5	0.3-8.8	74%	1.8-9.5	5.8-13.9	23%	0.4-7.3	0.2-11.8
16	71.5	43%	0.1-11.5	0.6-15.6	39%	2.6-7.0	6.3-14.6	12%	0.1-4.6	0.6-10.1	55%	3.2-14.2	5.8-17.0	11%	0.5-7.3	0.5-10.1
17	73.3	40%	0.5-8.9	3.1-14.7	18%	1.7-4.1	7.0-11.7	24%	0.4-4.0	3.1-11.8	26%	1.8-12.4	6.2-17.2	29%	0.5-0.8	2.7-12.6
18	73.3	37%	0.1-12.9	0.6-16.5	7%	1.7-4.5	4.5-8.6	8%	0.1-4.8	0.6-10.2	8%	2.3-8.7	4.0-10.7	5%	0.5-7.5	0.5-9.9
19	73.3	43%	0.6-13.8	2.4-19.8	20%	1.8-3.9	6.0-10.1	16%	0.4-5.4	2.3-13.3	19%	2.1-7.6	5.3-11.2	18%	0.6-9.1	1.9-13.8
20	73.3	15%	0.0-3.4	1.8-9.2	17%	0.3-1.5	4.8-7.1	16%	0.1-1.8	2.0-8.1	36%	1.1-8.9	4.6-12.7	25%	0.4-6.2	1.8-11.1
21	73.3	54%	0.4-9.8	1.8-13.8	34%	1.7-3.4	4.9-8.1	32%	0.3-4.0	1.8-9.0	41%	2.2-6.4	4.5-9.3	37%	0.7-7.0	1.6-9.8
22	73.3	12%	0.1-16.6	1.7-29.4	4%	1.8-4.2	10.5-16.2	1%	0.1-5.6	1.7-20.6	6%	1.4-11.9	9.1-20.9	6%	0.2-14.7	1.4-25.4
23	89.8	17%	0.6-4.1	3.1-9.1	24%	1.1-2.2	5.7-7.7	8%	0.4-2.0	3.1-7.5	31%	1.4-5.9	5.2-9.6	12%	0.7-4.3	2.7-8.3
24	59.0	3%	0.0-5.1	0.6-10.3	18%	1.0-3.0	5.3-9.7	11%	0.0-2.3	0.6-8.1	33%	1.3-9.2	4.8-13.4	-1%	0.4-5.5	0.5-9.5
25	60.4	-2%	0.0-5.7	0.5-11.0	16%	1.0-3.2	5.3-10.2	1%	0.0-2.4	0.5-8.1	34%	1.3-10.1	4.8-14.1	-5%	0.4-6.2	0.4-10.1
26	121.7	55%	1.1-6.5	3.6-11.6	47%	2.0-2.8	6.6-8.1	42%	0.8-3.4	3.5-9.2	47%	2.3-4.3	5.9-8.0	41%	1.0-5.3	3.1-9.4
27	73.3	39%	0.5-9.2	2.2-14.3	13%	1.7-3.7	5.8-9.8	16%	0.4-4.2	2.2-10.8	17%	2.0-7.4	5.3-10.9	20%	0.6-7.0	1.9-11.3
Avg Difference=		23%			21%			12%			32%			14%		

	P10 Fixed	P10 Normal	P10 Uniform			P11 Fixed	P11 Normal	P11 Uniform			P12 Fixed	P12 Normal	P12 Uniform			P13 Normal		
Run	Δ	Δ	Δ	s	S	Δ	Δ	Δ	s	S	Δ	Δ	Δ	s	S	Δ	s	S
1	-12%	-2%	-2%	0.2-1.6	1.7-6.2	14%	-4%	-1%	0.1-0.9	1.7-5.8	-11%	-2%	-3%	0.3-1.2	1.4-5.2	146%	0.1-31.6	1.4-40.9
2	-4%	8%	6%	0.2-1.7	2.2-7.9	37%	9%	11%	0.1-0.9	2.3-7.6	10%	5%	5%	0.2-0.9	1.8-6.6	197%	0.1-36.9	1.9-49.5
3	9%	19%	23%	0.3-2.4	2.0-6.6	34%	19%	24%	0.3-2.0	2.0-6.5	5%	14%	19%	0.5-2.2	1.8-6.0	165%	0.3-39.4	2.0-47.4
4	4%	10%	11%	0.4-2.6	2.9-9.1	38%	16%	11%	0.3-2.0	3.0-8.9	3%	16%	10%	0.3-1.9	2.5-8.0	175%	0.3-51.6	2.5-63.7
5	3%	-1%	8%	0.2-1.5	2.2-7.4	31%	-3%	6%	0.1-0.9	2.3-7.2	1%	-1%	4%	0.2-0.7	1.9-6.3	89%	0.2-28.1	2.4-40.8
6	-13%	-2%	1%	0.2-1.7	1.3-4.6	6%	2%	1%	0.1-0.9	1.3-4.2	-14%	-4%	-2%	0.5-1.6	1.1-3.9	253%	0.2-35.9	1.1-42.0
7	3%	18%	13%	0.2-2.5	1.9-7.9	23%	18%	17%	0.1-1.4	1.9-7.2	0%	16%	11%	0.4-1.9	1.6-6.6	111%	0.2-40.6	1.7-50.0
8	-5%	0%	4%	0.1-1.2	1.5-5.0	17%	2%	5%	0.1-0.7	1.5-4.7	2%	3%	5%	0.3-0.9	1.3-4.2	155%	0.1-70.5	1.4-81.3
9	9%	20%	22%	0.2-2.4	1.3-6.0	20%	11%	17%	0.1-1.4	1.3-5.3	7%	15%	17%	0.5-2.2	1.2-5.1	116%	0.2-27.1	1.4-33.2
10	-13%	0%	-5%	0.1-1.4	1.0-4.2	0%	0%	-6%	0.1-0.8	1.0-3.9	-13%	-3%	-8%	0.4-1.3	0.9-3.6	155%	0.2-27.3	1.2-33.4
11	-11%	-4%	-4%	0.1-1.5	1.5-5.8	8%	-4%	-5%	0.1-0.9	1.5-5.5	-8%	-3%	-9%	0.3-1.1	1.2-4.9	107%	0.2-31.8	1.8-41.0
12	4%	17%	19%	0.2-2.5	1.6-7.4	24%	11%	14%	0.1-1.4	1.7-6.7	-3%	12%	13%	0.4-1.9	1.4-6.2	86%	0.1-16.1	1.2-23.0
13	7%	19%	26%	0.2-2.3	1.5-6.2	44%	36%	50%	0.2-1.7	1.6-6.0	8%	34%	42%	0.5-2.0	1.3-5.4	252%	0.4-34.1	2.0-40.9
14	-9%	-6%	-3%	0.2-1.5	2.5-7.4	20%	-2%	-7%	0.1-0.8	2.6-7.1	-2%	-2%	3%	0.2-0.8	2.1-6.2	100%	0.2-36.9	2.6-49.5
15	-7%	5%	1%	0.2-1.5	1.6-5.5	14%	4%	0%	0.1-0.9	1.6-5.2	-8%	2%	-4%	0.3-1.2	1.4-4.8	151%	0.2-24.3	1.6-32.6
16	-13%	12%	2%	0.3-2.6	1.5-5.6	3%	20%	13%	0.3-1.9	1.5-5.2	-16%	6%	-2%	0.6-2.6	1.3-4.9	222%	0.3-44.5	1.3-50.4
17	7%	6%	12%	0.3-2.7	2.4-8.3	36%	10%	21%	0.3-2.1	2.5-8.1	5%	12%	15%	0.4-2.1	2.1-7.3	150%	0.3-45.1	2.3-55.7
18	-12%	1%	0%	0.4-2.7	1.6-5.4	5%	9%	13%	0.3-2.0	1.6-4.9	-12%	-3%	-4%	0.7-2.7	1.4-4.7	184%	0.3-68.5	1.3-73.7
19	3%	12%	15%	0.3-3.2	1.8-8.0	29%	24%	23%	0.2-2.4	1.8-7.5	-1%	17%	23%	0.5-2.8	1.6-6.9	196%	0.4-40.8	2.0-47.9
20	-2%	8%	3%	0.1-1.5	1.5-5.7	18%	6%	2%	0.1-0.9	1.6-5.4	-3%	3%	-1%	0.3-1.1	1.4-5.0	117%	0.1-32.9	1.3-43.0
21	1%	22%	15%	0.3-2.6	1.6-5.9	19%	25%	21%	0.3-2.0	1.6-5.6	2%	17%	7%	0.6-2.6	1.4-5.3	232%	0.3-30.8	1.5-36.0
22	-5%	1%	-2%	0.3-2.9	3.5-12.4	31%	12%	10%	0.2-2.1	3.6-12.2	32%	23%	17%	0.1-1.4	2.9-10.5	71%	0.4-43.8	4.0-59.0
23	-1%	11%	5%	0.2-2.0	1.7-6.7	21%	6%	2%	0.1-1.1	1.7-6.1	-3%	6%	-1%	0.4-1.6	1.4-5.6	108%	0.2-31.9	1.8-40.5
24	-2%	9%	11%	0.2-1.0	1.6-4.5	17%	9%	10%	0.1-0.6	1.6-4.3	4%	7%	11%	0.4-0.8	1.3-3.8	195%	0.1-32.0	1.4-40.6
25	-3%	11%	4%	0.1-1.2	1.5-4.9	21%	13%	3%	0.1-0.7	1.5-4.7	3%	14%	1%	0.4-1.0	1.2-4.2	196%	0.1-32.0	1.5-40.6
26	32%	43%	47%	0.4-4.9	2.2-10.8	54%	37%	37%	0.3-3.7	2.3-10.0	22%	33%	34%	0.6-4.4	2.0-9.4	64%	0.3-13.9	2.0-20.0
27	0%	3%	7%	0.3-2.8	2.0-7.2	23%	7%	11%	0.3-2.1	2.0-6.9	-3%	6%	1%	0.5-2.5	1.7-6.3	157%	0.4-34.0	2.2-41.0
	-1%	9%	9%			22%	11%	11%			0%	9%	8%			154%		

Appendix F (continued): Kabir and Al-Olayan Comparison for β=3.0

Policy (P)		P13 Uniform			P14 Normal			P14 Uniform			P15 Normal			P15 Uniform			P16 Fixed	P16 Normal
Run	t^*	Δ	s	S	Δ	s	S	Δ	s	S	Δ	s	S	Δ	s	S	Δ	Δ
1	76.0	65%	0.1-16.1	1.3-23.8	33%	0.1-7.9	1.4-19.9	16%	0.1-5.0	1.3-14.2	129%	0.3-30.0	1.2-35.9	56%	0.3-16.7	1.1-22.1	52%	49%
2	76.0	83%	0.1-18.8	1.7-29.2	43%	0.1-8.6	1.9-24.7	29%	0.1-5.5	1.8-17.8	186%	0.2-38.9	1.6-47.3	81%	0.2-21.3	1.4-28.8	106%	111%
3	76.0	81%	0.3-13.0	1.7-19.3	63%	0.3-8.7	2.1-19.4	47%	0.2-5.6	1.7-13.2	125%	0.5-28.0	1.8-33.1	73%	0.4-11.5	1.5-16.5	41%	40%
4	76.0	62%	0.3-23.4	2.5-34.0	45%	0.2-10.4	2.6-25.1	26%	0.2-7.6	2.6-19.8	134%	0.3-42.3	2.2-50.1	52%	0.3-21.7	2.1-29.8	18%	19%
5	71.5	54%	0.1-14.3	1.9-24.8	31%	0.1-7.4	2.4-21.5	24%	0.1-4.7	1.9-15.9	127%	0.2-38.2	2.0-46.8	71%	0.2-20.9	1.6-28.5	76%	58%
6	76.0	74%	0.1-18.2	1.0-23.2	52%	0.1-8.2	1.1-18.6	18%	0.1-5.0	1.0-12.3	131%	0.4-22.6	0.9-26.1	37%	0.4-12.9	0.8-16.0	31%	39%
7	99.2	71%	0.2-20.6	1.7-28.3	29%	0.1-6.8	1.7-23.5	24%	0.1-5.7	1.7-16.1	72%	0.3-30.3	1.4-36.3	45%	0.3-16.8	1.4-22.2	12%	34%
8	59.8	88%	0.1-56.8	11-67.1	24%	0.1-12.9	1.4-31.9	7%	0.1-11.2	1.2-28.5	96%	0.3-48.0	1.1-54.3	45%	0.3-40.2	1.0-46.5	94%	89%
9	99.2	55%	0.1-13.8	0.7-18.8	39%	0.1-6.6	1.4-15.3	25%	0.0-4.1	0.7-10.3	89%	0.5-21.8	1.2-25.4	47%	0.4-12.5	0.6-15.6	24%	26%
10	71.5	62%	0.1-14.0	0.7-19.0	41%	0.1-6.7	1.2-15.3	17%	0.0-4.4	0.7-10.9	118%	0.4-21.9	1.0-25.4	50%	0.4-12.6	0.6-15.8	32%	32%
11	71.5	23%	0.1-16.1	1.4-23.7	18%	0.1-7.8	1.8-19.7	0%	0.1-5.0	1.4-14.6	92%	0.3-30.1	1.5-35.9	19%	0.3-16.7	1.1-22.0	45%	42%
12	99.2	84%	0.2-16.3	1.7-23.2	30%	0.1-4.9	1.3-13.7	33%	0.1-4.9	1.7-13.5	71%	0.4-15.4	1.1-20.1	68%	0.4-15.6	1.4-20.2	30%	39%
13	59.8	162%	0.2-39.8	1.2-47.1	67%	0.3-8.4	2.1-18.4	40%	0.1-8.9	1.2-19.3	129%	0.5-20.2	1.8-24.5	74%	0.4-23.2	1.0-27.8	27%	31%
14	71.5	59%	0.1-37.2	1.6-49.8	24%	0.1-8.6	2.6-24.6	11%	0.1-8.5	1.7-24.4	101%	0.2-38.9	2.2-47.3	58%	0.2-39.2	1.4-47.6	84%	78%
15	71.5	67%	0.0-14.6	1.1-21.7	44%	0.1-6.1	1.6-16.2	23%	0.0-4.1	1.1-12.2	166%	0.3-27.0	1.3-32.3	81%	0.3-17.5	1.0-22.3	49%	48%
16	71.5	103%	0.2-37.1	1.1-42.3	38%	0.2-10.9	1.3-21.7	21%	0.2-8.5	1.1-17.2	68%	0.6-21.0	1.2-24.6	27%	0.5-17.2	0.9-20.2	23%	25%
17	73.3	92%	0.3-28.5	2.3-37.5	45%	0.3-10.6	2.4-24.3	31%	0.3-7.4	2.4-18.1	104%	0.4-33.5	2.0-40.5	65%	0.4-22.5	2.0-28.6	22%	23%
18	73.3	87%	0.2-39.8	1.1-44.2	29%	0.3-11.6	1.3-22.8	14%	0.2-8.2	1.1-16.4	41%	0.6-24.6	1.2-27.3	17%	0.6-16.2	0.9-18.6	26%	28%
19	73.3	81%	0.3-21.4	1.7-27.3	46%	0.3-8.6	2.0-19.5	24%	0.2-6.1	1.7-14.6	74%	0.5-20.8	1.7-25.3	34%	0.5-12.2	1.5-16.2	29%	29%
20	73.3	35%	0.1-9.1	1.3-16.2	35%	0.1-5.9	1.3-18.1	19%	0.1-3.1	1.3-11.3	186%	0.3-41.8	1.1-48.2	66%	0.3-14.8	1.1-19.9	55%	47%
21	73.3	102%	0.3-18.6	1.4-23.4	71%	0.3-7.5	1.5-15.4	39%	0.5-5.5	1.5-11.9	116%	0.6-17.1	1.3-20.3	56%	0.6-11.8	1.3-14.9	54%	57%
22	73.3	42%	0.1-44.0	2.0-59.1	14%	0.3-9.7	4.1-28.5	6%	0.1-9.2	2.0-27.6	51%	0.1-36.5	3.4-47.4	27%	0.1-36.7	1.6-47.2	-1%	1%
23	89.8	56%	0.2-16.4	1.6-23.4	32%	0.1-7.9	1.8-19.5	20%	0.1-4.9	1.6-13.4	93%	0.4-28.5	1.5-33.9	48%	0.4-16.0	1.3-20.7	42%	46%
24	59.0	93%	0.1-21.0	0.9-28.9	52%	0.1-7.8	1.5-19.2	33%	0.0-6.1	0.9-16.1	161%	0.4-28.6	1.2-33.9	80%	0.4-20.1	0.8-25.4	116%	113%
25	60.4	81%	0.1-28.1	1.3-36.7	53%	0.1-7.9	1.5-19.6	19%	0.1-7.4	1.4-18.7	159%	0.4-28.6	1.3-33.9	66%	0.3-25.8	1.1-31.3	95%	91%
26	121.7	55%	0.4-8.6	2.1-15.2	40%	0.3-5.3	2.0-12.8	37%	0.3-5.5	2.1-13.3	43%	0.5-10.2	1.7-14.8	38%	0.5-7.3	1.8-12.9	62%	64%
27	73.3	61%	0.3-17.0	1.8-22.9	40%	0.3-8.1	2.3-17.9	17%	0.2-5.6	1.9-13.5	77%	0.5-20.6	1.9-25.1	32%	0.5-11.7	1.6-16.2	24%	24%
Avg Difference=		73%			40%			23%			109%			52%			47%	47%

	P16 Uniform			P17 Normal			P17 Uniform			P18 Fixed	P18 Normal	P18 Uniform			P19 Normal	P19 Uniform		
Run	Δ	s	S	Δ	s	S	Δ	s	S	Δ	Δ	Δ	s	S	Δ	Δ	s	S
1	52%	0	1.0-2.0	-12%	0.0-3.0	0.0-5.0	11%	0.0-2.0	0.0-4.0	53%	56%	56%	0.0-4.0	0.0-11.0	15%	17%	1.0-3.0	6.0-10.0
2	109%	0	1.0-3.0	3%	0.0-3.0	1.0-5.0	47%	0.0-2.0	1.0-4.0	64%	67%	66%	0.0-5.0	0.0-14.0	24%	27%	1.0-4.0	8.0-13.0
3	40%	3.0-4.0	5.0-6.0	61%	3.0-8.0	5.0-10.0	56%	3.0-6.0	5.0-8.0	140%	135%	138%	0.0-6.0	0.0-12.0	39%	50%	0.0-4.0	0.0-10.0
4	20%	3.0-4.0	5.0-6.0	23%	3.0-7.0	5.0-9.0	20%	3.0-7.0	5.0-9.0	86%	86%	88%	0.0-6.0	0.0-16.0	28%	32%	2.0-5.0	8.0-15.0
5	66%	0	1.0-2.0	19%	0.0-2.0	1.0-5.0	48%	0.0-1.0	1.0-4.0	56%	54%	57%	3.0-4.0	12.0-14.0	29%	35%	1.0-4.0	9.0-13.0
6	36%	0	1.0-2.0	-16%	0.0-3.0	1.0-5.0	13%	0.0-2.0	1.0-4.0	84%	85%	86%	0.0-5.0	0.0-9.0	18%	22%	1.0-3.0	4.0-8.0
7	29%	0.0-1.0	1.0-3.0	10%	0.0-3.0	1.0-5.0	6%	0.0-2.0	1.0-4.0	72%	74%	74%	0.0-5.0	0.0-12.0	31%	36%	0.0-4.0	0.0-10.0
8	101%	0	1.0-2.0	-18%	0.0-5.0	1.0-6.0	36%	0.0-4.0	1.0-5.0	39%	38%	38%	4.0-6.0	11.0-12.0	3%	7%	0.0-8.0	0.0-15.0
9	26%	0	0.0-2.0	-7%	0.0-3.0	0.0-4.0	10%	0.0-2.0	0.0-4.0	104%	103%	103%	0.0-4.0	0.0-9.0	38%	42%	1.0-2.0	5.0-6.0
10	28%	0	0.0-2.0	-14%	0.0-3.0	0.0-4.0	-3%	0.0-3.0	0.0-4.0	69%	74%	69%	0.0-4.0	0.0-8.0	38%	37%	0.0-3.0	0.0-9.0
11	38%	0	1.0-2.0	-12%	0.0-3.0	1.0-5.0	6%	0.0-2.0	0.0-4.0	40%	40%	36%	0.0-4.0	0.0-12.0	20%	17%	0.0-4.0	0.0-12.0
12	45%	0	1.0-3.0	7%	0.0-2.0	0.0-4.0	18%	0.0-2.0	1.0-4.0	98%	102%	103%	3.0-4.0	9.0-11.0	42%	48%	1.0-3.0	6.0-8.0
13	30%	4	5.0-6.0	48%	4.0-8.0	5.0-9.0	42%	4.0-8.0	5.0-10.0	113%	113%	114%	6.0-7.0	12.0-13.0	39%	51%	0.0-6.0	0.0-11.0
14	86%	0	1.0-2.0	-7%	0.0-3.0	1.0-5.0	25%	0.0-4.0	1.0-5.0	42%	42%	44%	0.0-5.0	0.0-14.0	21%	25%	2.0-5.0	9.0-14.0
15	44%	0	0.0-2.0	13%	0.0-2.0	0.0-4.0	28%	0.0-1.0	0.0-4.0	72%	75%	72%	0.0-3.0	0.0-10.0	46%	45%	0.0-3.0	0.0-10.0
16	26%	4.0-5.0	5.0-6.0	35%	4.0-8.0	5.0-10.0	32%	4.0-7.0	5.0-8.0	93%	96%	94%	6.0-7.0	10.0-12.0	49%	54%	3.0-8.0	7.0-11.0
17	26%	3.0-4.0	5.0-6.0	28%	3.0-8.0	5.0-10.0	30%	3.0-7.0	5.0-9.0	96%	98%	101%	0.0-7.0	0.0-15.0	27%	38%	0.0-5.0	0.0-14.0
18	26%	4.0-5.0	5.0-6.0	34%	4.0-7.0	5.0-8.0	30%	4.0-8.0	5.0-9.0	86%	87%	87%	0.0-7.0	0.0-11.0	17%	19%	0.0-8.0	0.0-10.0
19	28%	4.0-5.0	5.0-7.0	37%	4.0-7.0	5.0-9.0	32%	4.0-7.0	5.0-9.0	104%	104%	102%	0.0-8.0	0.0-15.0	29%	34%	0.0-6.0	0.0-11.0
20	49%	0	0.0-2.0	28%	0.0-5.0	0.0-6.0	38%	0.0-1.0	0.0-4.0	61%	59%	60%	2.0-3.0	8.0-10.0	19%	21%	0.0-3.0	5.0-10.0
21	54%	3.0-4.0	5.0-6.0.	74%	3.0-7.0	5.0-9.0	66%	3.0-6.0	5.0-8.0	148%	150%	148%	5.0-6.0	10.0-11.0	43%	48%	0.0-4.0	0.0-9.0
22	-5%	4.0-5.0	6.0-7.0	4%	4.0-7.0	5.0-9.0	-2%	4.0-8.0	5.0-10.0	36%	37%	31%	6.0-8.0	19.0-22.0	8%	5%	2.0-7.0	11.0-17.0
23	37%	0	1.0-3.0	0%	0.0-3.0	1.0-5.0	10%	0.0-2.0	1.0-4.0	74%	76%	72%	0.0-5.0	0.0-11.0	37%	34%	1.0-3.0	6.0-9.0
24	120%	0	1.0-2.0	-16%	0.0-3.0	1.0-5.0	37%	0.0-3.0	0.0-4.0	79%	77%	82%	3.0-4.0	10.0-11.0	24%	35%	1.0-4.0	5.0-11.0
25	84%	0	1.0-2.0	-8%	0.0-3.0	1.0-5.0	23%	0.0-2.0	1.0-4.0	63%	65%	61%	3.0-5.0	10.0-11.0	25%	25%	1.0-4.0	5.0-11.0
26	65%	4.0-5.0	5.0-7.0	64%	4.0-7.0	5.0-8.0	64%	4.0-6.0	5.0-8.0	147%	148%	151%	5.0-7.0	11.0-14.0	69%	68%	0.0-4.0	0.0-9.0
27	26%	4.0-5.0	5.0-6.0	35%	4.0-7.0	5.0-8.0	33%	4.0-7.0	5.0-8.0	92%	90%	94%	5.0-7.0	12.0-14.0	23%	27%	0.0-5.0	0.0-10.0
	48%			15%			28%			82%	83%	82%			30%	33%		

Appendix G: Cost Comparison with Additional CR/UR Values

Policy (P) CR/UR Value	P1 Normal Δ	P1 Uniform Δ	P2 Normal Δ	P2 Uniform Δ	P3 Normal Δ	P3 Uniform Δ
0.025	29%	29%	14%	13%	20%	20%
0.05	29%	29%	13%	13%	21%	21%
0.075	30%	31%	14%	15%	21%	21%
0.1	28%	28%	13%	13%	19%	19%

CR/UR Value	P4 Normal Δ	P4 Uniform Δ	P5 Normal Δ	P5 Uniform Δ	P6 Normal Δ	P6 Uniform Δ
0.025	119%	147%	65%	90%	99%	132%
0.05	123%	150%	66%	91%	99%	133%
0.075	125%	153%	68%	94%	102%	136%
0.1	124%	152%	67%	92%	100%	133%

CR/UR Value	P7 Normal Δ	P7 Uniform Δ	P8 Normal Δ	P8 Uniform Δ	P9 Normal Δ	P9 Uniform Δ
0.025	43%	23%	21%	12%	32%	14%
0.05	45%	30%	22%	16%	32%	21%
0.075	48%	55%	24%	28%	34%	40%
0.1	51%	61%	24%	31%	35%	44%

CR/UR Value	P10 Normal Δ	P10 Uniform Δ	P11 Normal Δ	P11 Uniform Δ	P12 Normal Δ	P12 Uniform Δ
0.025	9%	9%	11%	11%	9%	8%
0.05	10%	10%	12%	13%	9%	9%
0.075	10%	12%	12%	13%	11%	11%
0.1	11%	11%	14%	14%	10%	11%

CR/UR Value	P13 Normal Δ	P13 Uniform Δ	P14 Normal Δ	P14 Uniform Δ	P15 Normal Δ	P15 Uniform Δ
0.025	154%	73%	40%	23%	109%	52%
0.05	185%	115%	43%	34%	124%	78%
0.075	184%	129%	47%	44%	123%	90%
0.1	196%	128%	47%	45%	124%	86%

CR/UR Value	P16 Normal Δ	P16 Uniform Δ	P17 Normal Δ	P17 Uniform Δ	P18 Normal Δ	P18 Uniform Δ
0.025	47%	48%	15%	28%	83%	82%
0.05	43%	43%	16%	24%	84%	83%
0.075	46%	47%	18%	24%	86%	86%
0.1	39%	40%	18%	22%	84%	84%

CR/UR Value	P19 Normal Δ	P19 Uniform Δ
0.025	30%	33%
0.05	31%	34%
0.075	33%	37%
0.1	33%	37%

Appendix H: Matlab Code

Text in italics is added for code explanation

Parameter definition and values

```
Penalty_Cost=13.5;
Holding_Cost=.6875;
Ordering_Cost=8.75;
Cf=55;
Cp=25;
Demand_History_Fixed=xlsread('demand data.xls','Demand');
h=Holding_Cost;
p=Penalty_Cost;
K=Ordering_Cost;
Shape=3;
Shape_Fixed=3;
Shape_Normal=3;
Shape_Uniform=3;
Scale=100;
Scale_Fixed=90;
Scale_Normal=90;
Scale_Uniform=90;
PC=.99;
CR=.025;
UR=.025;
k=30;
Scale_Lead=10;
Shape_Lead=3.2;
Period_Maintenance_Fixed=0;
Period_Failure_Fixed=0;
Period_Maintenance_Normal=0;
Period_Failure_Normal=0;
Period_Maintenance_Uniform=0;
Period_Failure_Uniform=0;
Period_Holding_FO_Lead_Fixed_PowerAppx=0;
Period_Holding_FO_Lead_Fixed_NormalAppx=0;
Period_Holding_FO_Lead_Fixed_NaddorAppx=0;
Period_Holding_FO_Lead_Normal_PowerAppx=0;
Period_Holding_FO_Lead_Normal_NormalAppx=0;
Period_Holding_FO_Lead_Normal_NaddorAppx=0;
Period_Holding_FO_Lead_Uniform_PowerAppx=0;
Period_Holding_FO_Lead_Uniform_NormalAppx=0;
Period_Holding_FO_Lead_Uniform_NaddorAppx=0;
Period_Shortage_FO_Lead_Fixed_PowerAppx=0;
Period_Shortage_FO_Lead_Fixed_NormalAppx=0;
```

Appendix H (continued): Matlab Code

```
Period_Shortage_FO_Lead_Fixed_NaddorAppx=0;
Period_Shortage_FO_Lead_Normal_PowerAppx=0;
Period_Shortage_FO_Lead_Normal_NormalAppx=0;
Period_Shortage_FO_Lead_Normal_NaddorAppx=0;
Period_Shortage_FO_Lead_Uniform_PowerAppx=0;
Period_Shortage_FO_Lead_Uniform_NormalAppx=0;
Period_Shortage_FO_Lead_Uniform_NaddorAppx=0;
Period_Order_FO_Lead_Fixed_PowerAppx=0;
Period_Order_FO_Lead_Fixed_NormalAppx=0;
Period_Order_FO_Lead_Fixed_NaddorAppx=0;
Period_Order_FO_Lead_Normal_PowerAppx=0;
Period_Order_FO_Lead_Normal_NormalAppx=0;
Period_Order_FO_Lead_Normal_NaddorAppx=0;
Period_Order_FO_Lead_Uniform_PowerAppx=0;
Period_Order_FO_Lead_Uniform_NormalAppx=0;
Period_Order_FO_Lead_Uniform_NaddorAppx=0;
On_Hand_FO_Lead_Fixed_PowerAppx=ceil(S_FO_Lead_Fixed_PowerAppx);
On_Hand_FO_Lead_Fixed_NormalAppx=ceil(S_FO_Lead_Fixed_NormalAppx);
On_Hand_FO_Lead_Fixed_NaddorAppx=ceil(S_FO_Lead_Fixed_NaddorAppx);
On_Hand_FO_Lead_Normal_PowerAppx=ceil(S_FO_Lead_Normal_PowerAppx);
On_Hand_FO_Lead_Normal_NormalAppx=ceil(S_FO_Lead_Normal_NormalAppx);
On_Hand_FO_Lead_Normal_NaddorAppx=ceil(S_FO_Lead_Normal_NaddorAppx);
On_Hand_FO_Lead_Uniform_PowerAppx=ceil(S_FO_Lead_Uniform_PowerAppx);
On_Hand_FO_Lead_Uniform_NormalAppx=ceil(S_FO_Lead_Uniform_NormalAppx);
On_Hand_FO_Lead_Uniform_NaddorAppx=ceil(S_FO_Lead_Uniform_NaddorAppx);
On_Order_FO_Lead_Fixed_PowerAppx=0;
On_Order_FO_Lead_Fixed_NormalAppx=0;
On_Order_FO_Lead_Fixed_NaddorAppx=0;
On_Order_FO_Lead_Normal_PowerAppx=0;
On_Order_FO_Lead_Normal_NormalAppx=0;
On_Order_FO_Lead_Normal_NaddorAppx=0;
On_Order_FO_Lead_Uniform_PowerAppx=0;
On_Order_FO_Lead_Uniform_NormalAppx=0;
On_Order_FO_Lead_Uniform_NaddorAppx=0;
Z_FO_Lead_Fixed_PowerAppx=On_Hand_FO_Lead_Fixed_PowerAppx+On_Order_F
O_Lead_Fixed_PowerAppx;
Z_FO_Lead_Fixed_NormalAppx=On_Hand_FO_Lead_Fixed_NormalAppx+On_Order_
FO_Lead_Fixed_NormalAppx;
Z_FO_Lead_Fixed_NaddorAppx=On_Hand_FO_Lead_Fixed_NaddorAppx+On_Order_
FO_Lead_Fixed_NaddorAppx;
Z_FO_Lead_Normal_PowerAppx=On_Hand_FO_Lead_Normal_PowerAppx+On_Order
_FO_Lead_Normal_PowerAppx;
Z_FO_Lead_Uniform_PowerAppx=On_Hand_FO_Lead_Uniform_PowerAppx+On_Ord
```

```
er_FO_Lead_Uniform_PowerAppx;
Z_FO_Lead_Normal_NormalAppx=On_Hand_FO_Lead_Normal_NormalAppx+On_Or
der_FO_Lead_Normal_NormalAppx;
Z_FO_Lead_Uniform_NormalAppx=On_Hand_FO_Lead_Uniform_NormalAppx+On_
Order_FO_Lead_Uniform_NormalAppx;
Z_FO_Lead_Normal_NaddorAppx=On_Hand_FO_Lead_Normal_NaddorAppx+On_Or
der_FO_Lead_Normal_NaddorAppx;
Z_FO_Lead_Uniform_NaddorAppx=On_Hand_FO_Lead_Uniform_NaddorAppx+On_O
rder_FO_Lead_Uniform_NaddorAppx;
Simulation_Length=100;
reset=15;
history=10;
Lead_Orders_FO_Lead_Fixed_PowerAppx=zeros(Simulation_Length+lamda+1,1);
Lead_Orders_FO_Lead_Normal_PowerAppx=zeros(Simulation_Length+lamda+1,1);
Lead_Orders_FO_Lead_Uniform_PowerAppx=zeros(Simulation_Length+lamda+1,1);
Lead_Orders_FO_Lead_Fixed_NormalAppx=zeros(Simulation_Length+lamda+1,1);
Lead_Orders_FO_Lead_Normal_NormalAppx=zeros(Simulation_Length+lamda+1,1);
Lead_Orders_FO_Lead_Uniform_NormalAppx=zeros(Simulation_Length+lamda+1,1);
Lead_Orders_FO_Lead_Fixed_NaddorAppx=zeros(Simulation_Length+lamda+1,1);
Lead_Orders_FO_Lead_Normal_NaddorAppx=zeros(Simulation_Length+lamda+1,1);
Lead_Orders_FO_Lead_Uniform_NaddorAppx=zeros(Simulation_Length+lamda+1,1);
```

Determine initial maintenance strategy

```
tstar=Optimal_Replacement_Time(Shape_Fixed,Scale_Fixed,Cp,Cf);
tstar_Fixed=tstar;
tstar_Normal=tstar;
tstar_Uniform=tstar;
sigma=Maintenance_Policy(PC,CR,tstar);
Uniform_a=tstar-UR*tstar;
Uniform_b=tstar+UR*tstar;
t=xlsread('part data.xls','Part Ages');
n=length(t);
```

Determine period length

```
[M,V] = weibstat(1/(Scale_Lead^Shape_Lead),Shape_Lead);
deltat=M;
lamda=ceil((weibrnd((1/(Scale_Lead^Shape_Lead)),Shape_Lead))/deltat);
```

Determine initial replacement modes (failed or maintained) for all parts

```
t_Fixed=Replacement_mode_Fixed(t,Shape,Scale,tstar);
Data_Combined_Fixed(:,1)=t_Fixed(:,1);
Data_Combined_Fixed(:,2)=0;
t_Normal=Replacement_mode_Normal(t,Shape,Scale,tstar,sigma);
```

Appendix H (continued): Matlab Code

```
Data_Combined_Normal(:,1)=t_Normal(:,1);
Data_Combined_Normal(:,2)=0;
t_Uniform=Replacement_mode_Uniform(t,Shape,Scale,tstar,Uniform_a,Uniform_b);
Data_Combined_Uniform(:,1)=t_Uniform(:,1);
Data_Combined_Uniform(:,2)=0;
```

Determine the conditional probabilities and demand distribution for future periods

```
Pconditional_FO_Fixed=Conditional_Failure_Only(n,t_Fixed,k,deltat,Shape_Fixed,Scale
_Fixed);
Pconditional_FO_Normal=Conditional_Failure_Only(n,t_Normal,k,deltat,Shape_Normal
,Scale_Normal);
Pconditional_FO_Uniform=Conditional_Failure_Only(n,t_Uniform,k,deltat,Shape_Unifo
rm,Scale_Uniform);
Demand_Failure_Only_Fixed=Demand_Distribution(n,k,Pconditional_FO_Fixed);
Demand_Failure_Only_Normal=Demand_Distribution(n,k,Pconditional_FO_Normal);
Demand_Failure_Only_Uniform=Demand_Distribution(n,k,Pconditional_FO_Uniform);
FO_Demand_pdf_Fixed=kExpected_Demand(n,k,Demand_Failure_Only_Fixed);
FO_Demand_pdf_Normal=kExpected_Demand(n,k,Demand_Failure_Only_Normal);
FO_Demand_pdf_Uniform=kExpected_Demand(n,k,Demand_Failure_Only_Uniform);
```

Determine the initial inventory values

```
Demand_FO_Lead_Fixed=Expected_Demand(n,lamda+1,Demand_Failure_Only_Fixed)
;
 [s_FO_Lead_Fixed_PowerAppx S_FO_Lead_Fixed_PowerAppx]
=Power_Approximation(lamda,p,h,K,Demand_FO_Lead_Fixed);
[s_FO_Lead_Fixed_NormalAppx S_FO_Lead_Fixed_NormalAppx]
=Normal_Approximation(lamda,p,h,K,Demand_FO_Lead_Fixed);
[s_FO_Lead_Fixed_NaddorAppx S_FO_Lead_Fixed_NaddorAppx]
=Naddor_Approximation(lamda,p,h,K,Demand_FO_Lead_Fixed);
FO_Lead_Fixed_Exp_Demand=Demand_FO_Lead_Fixed(1,1);
Demand_FO_Lead_Normal=Expected_Demand(n,lamda+1,Demand_Failure_Only_Nor
mal);
[s_FO_Lead_Normal_PowerAppx S_FO_Lead_Normal_PowerAppx]
=Power_Approximation(lamda,p,h,K,Demand_FO_Lead_Normal);
[s_FO_Lead_Normal_NormalAppx S_FO_Lead_Normal_NormalAppx]
=Normal_Approximation(lamda,p,h,K,Demand_FO_Lead_Normal);
[s_FO_Lead_Normal_NaddorAppx S_FO_Lead_Normal_NaddorAppx]
=Naddor_Approximation(lamda,p,h,K,Demand_FO_Lead_Normal);
FO_Lead_Normal_Exp_Demand=Demand_FO_Lead_Normal(1,1);
Demand_FO_Lead_Uniform=Expected_Demand(n,lamda+1,Demand_Failure_Only_Uni
form);
[s_FO_Lead_Uniform_PowerAppx S_FO_Lead_Uniform_PowerAppx]
=Power_Approximation(lamda,p,h,K,Demand_FO_Lead_Uniform);
```

Appendix H (continued): Matlab Code

```
[s_FO_Lead_Uniform_NormalAppx S_FO_Lead_Uniform_NormalAppx]
=Normal_Approximation(lamda,p,h,K,Demand_FO_Lead_Uniform);
[s_FO_Lead_Uniform_NaddorAppx S_FO_Lead_Uniform_NaddorAppx]
=Naddor_Approximation(lamda,p,h,K,Demand_FO_Lead_Uniform);
FO_Lead_Uniform_Exp_Demand=Demand_FO_Lead_Uniform(1,1);
```

Begin simulated periods

```
Iteration=1;
while Iteration<Simulation_Length
```

Define/reset period values

```
Replacement_Data_Normal=0;
Replacement_Data_Uniform=0;
Replacement_Data_Fixed=0;
Period_Censored_Normal=0;
Period_Censored_Uniform=0;
Period_Censored_Fixed=0;
Failure_Data_Normal=0;
Failure_Data_Uniform=0;
Failure_Data_Fixed=0;
Period_Maintenance_Fixed=0;
Period_Failure_Fixed=0;
Period_Maintenance_Normal=0;
Period_Failure_Normal=0;
Period_Maintenance_Uniform=0;
Period_Failure_Uniform=0;
Period_Holding_FO_Lead_Fixed_PowerAppx=0;
Period_Shortage_FO_Lead_Fixed_PowerAppx=0;
Period_Order_FO_Lead_Fixed_PowerAppx=0;
Period_Holding_FO_Lead_Fixed_NormalAppx=0;
Period_Shortage_FO_Lead_Fixed_NormalAppx=0;
Period_Order_FO_Lead_Fixed_NoralAppx=0;
Period_Holding_FO_Lead_Fixed_NaddorAppx=0;
Period_Shortage_FO_Lead_Fixed_NaddorAppx=0;
Period_Order_FO_Lead_Fixed_NaddorAppx=0;
Period_Holding_FO_Lead_Normal_PowerAppx=0;
Period_Shortage_FO_Lead_Normal_PowerAppx=0;
Period_Order_FO_Lead_Normal_PowerAppx=0;
Period_Holding_FO_Lead_Normal_NormalAppx=0;
Period_Shortage_FO_Lead_Normal_NormalAppx=0;
Period_Order_FO_Lead_Normal_NormalAppx=0;
Period_Holding_FO_Lead_Normal_NaddorAppx=0;
Period_Shortage_FO_Lead_Normal_NaddorAppx=0;
```

Appendix H (continued): Matlab Code

```
Period_Order_FO_Lead_Normal_NaddorAppx=0;
Period_Holding_FO_Lead_Uniform_PowerAppx=0;
Period_Shortage_FO_Lead_Uniform_PowerAppx=0;
Period_Order_FO_Lead_Uniform_PowerAppx=0;
Period_Holding_FO_Lead_Uniform_NormalAppx=0;
Period_Shortage_FO_Lead_Uniform_NormalAppx=0;
Period_Order_FO_Lead_Uniform_NormalAppx=0;
Period_Holding_FO_Lead_Uniform_NaddorAppx=0;
Period_Shortage_FO_Lead_Uniform_NaddorAppx=0;
Period_Order_FO_Lead_Uniform_NaddorAppx=0;
```

Check to see if orders are received

```
if Lead_Orders_FO_Lead_Fixed_PowerAppx(Iteration)>0
        On_Hand_FO_Lead_Fixed_PowerAppx=On_Hand_FO_Lead_Fixed_PowerApp
        x+Lead_Orders_FO_Lead_Fixed_PowerAppx(Iteration);
        On_Order_FO_Lead_Fixed_PowerAppx=On_Order_FO_Lead_Fixed_PowerApp
        x-Lead_Orders_FO_Lead_Fixed_PowerAppx(Iteration);
end
if Lead_Orders_FO_Lead_Normal_PowerAppx(Iteration)>0
        On_Hand_FO_Lead_Normal_PowerAppx=On_Hand_FO_Lead_Normal_Power
        Appx+Lead_Orders_FO_Lead_Normal_PowerAppx(Iteration);
        On_Order_FO_Lead_Normal_PowerAppx=On_Order_FO_Lead_Normal_Power
        Appx-Lead_Orders_FO_Lead_Normal_PowerAppx(Iteration);
end
if Lead_Orders_FO_Lead_Uniform_PowerAppx(Iteration)>0
        On_Hand_FO_Lead_Uniform_PowerAppx=On_Hand_FO_Lead_Uniform_Powe
        rAppx+Lead_Orders_FO_Lead_Uniform_PowerAppx(Iteration);
        On_Order_FO_Lead_Uniform_PowerAppx=On_Order_FO_Lead_Uniform_Pow
        erAppx-Lead_Orders_FO_Lead_Uniform_PowerAppx(Iteration);
end
if Lead_Orders_FO_Lead_Fixed_NormalAppx(Iteration)>0
        On_Hand_FO_Lead_Fixed_NormalAppx=On_Hand_FO_Lead_Fixed_NormalA
        ppx+Lead_Orders_FO_Lead_Fixed_NormalAppx(Iteration);
        On_Order_FO_Lead_Fixed_NormalAppx=On_Order_FO_Lead_Fixed_NormalA
        ppx-Lead_Orders_FO_Lead_Fixed_NormalAppx(Iteration);
end
if Lead_Orders_FO_Lead_Normal_NormalAppx(Iteration)>0
        On_Hand_FO_Lead_Normal_NormalAppx=On_Hand_FO_Lead_Normal_Norm
        alAppx+Lead_Orders_FO_Lead_Normal_NormalAppx(Iteration);
        On_Order_FO_Lead_Normal_NormalAppx=On_Order_FO_Lead_Normal_Norm
        alAppx-Lead_Orders_FO_Lead_Normal_NormalAppx(Iteration);
end
```

Appendix H (continued): Matlab Code

```
if Lead_Orders_FO_Lead_Uniform_NormalAppx(Iteration)>0
    On_Hand_FO_Lead_Uniform_NormalAppx=On_Hand_FO_Lead_Uniform_Nor
    malAppx+Lead_Orders_FO_Lead_Uniform_NormalAppx(Iteration);
    On_Order_FO_Lead_Uniform_NormalAppx=On_Order_FO_Lead_Uniform_Nor
    malAppx-Lead_Orders_FO_Lead_Uniform_NormalAppx(Iteration);
end
if Lead_Orders_FO_Lead_Fixed_NaddorAppx(Iteration)>0
    On_Hand_FO_Lead_Fixed_NaddorAppx=On_Hand_FO_Lead_Fixed_NaddorAp
    px+Lead_Orders_FO_Lead_Fixed_NaddorAppx(Iteration);
    On_Order_FO_Lead_Fixed_NaddorAppx=On_Order_FO_Lead_Fixed_NaddorA
    ppx-Lead_Orders_FO_Lead_Fixed_NaddorAppx(Iteration);
end
if Lead_Orders_FO_Lead_Normal_NaddorAppx(Iteration)>0
    On_Hand_FO_Lead_Normal_NaddorAppx=On_Hand_FO_Lead_Normal_Naddo
    rAppx+Lead_Orders_FO_Lead_Normal_NaddorAppx(Iteration);
    On_Order_FO_Lead_Normal_NaddorAppx=On_Order_FO_Lead_Normal_Nadd
    orAppx-Lead_Orders_FO_Lead_Normal_NaddorAppx(Iteration);
end
if Lead_Orders_FO_Lead_Uniform_NaddorAppx(Iteration)>0
    On_Hand_FO_Lead_Uniform_NaddorAppx=On_Hand_FO_Lead_Uniform_Nad
    dorAppx+Lead_Orders_FO_Lead_Uniform_NaddorAppx(Iteration);
    On_Order_FO_Lead_Uniform_NaddorAppx=On_Order_FO_Lead_Uniform_Na
    ddorAppx-Lead_Orders_FO_Lead_Uniform_NaddorAppx(Iteration);
end
```

Check to see if parts fail or maintained during period

```
[output t_Fixed]=
Check_t(t_Fixed,Shape,Scale,tstar_Fixed,deltat,On_Hand_FO_Lead_Fixed_PowerAppx,
On_Hand_FO_Lead_Fixed_NormalAppx,On_Hand_FO_Lead_Fixed_NaddorAppx);
Period_Failure_Fixed=output(1);
Period_Maintenance_Fixed=output(2);
On_Hand_FO_Lead_Fixed_PowerAppx=output(3);
On_Hand_FO_Lead_Fixed_NormalAppx=output(4);
On_Hand_FO_Lead_Fixed_NaddorAppx=output(5);
Z_FO_Lead_Fixed_PowerAppx=On_Hand_FO_Lead_Fixed_PowerAppx+On_Order_F
O_Lead_Fixed_PowerAppx;
Z_FO_Lead_Fixed_NormalAppx=On_Hand_FO_Lead_Fixed_NormalAppx+On_Order_
FO_Lead_Fixed_NormalAppx;
Z_FO_Lead_Fixed_NaddorAppx=On_Hand_FO_Lead_Fixed_NaddorAppx+On_Order_
FO_Lead_Fixed_NaddorAppx;

[output t_Normal]
=Check_t_Normal(t_Normal,Shape,Scale,tstar_Normal,sigma,deltat,On_Hand_FO_Lead
```

```
_Normal_PowerAppx,On_Hand_FO_Lead_Normal_NormalAppx,On_Hand_FO_Lead_
Normal_NaddorAppx);
Period_Failure_Normal=output(1);
Period_Maintenance_Normal=output(2);
On_Hand_FO_Lead_Normal_PowerAppx=output(3);
On_Hand_FO_Lead_Normal_NormalAppx=output(4);
On_Hand_FO_Lead_Normal_NaddorAppx=output(5);
Z_FO_Lead_Normal_PowerAppx=On_Hand_FO_Lead_Normal_PowerAppx+On_Order
_FO_Lead_Normal_PowerAppx;
Z_FO_Lead_Normal_NormalAppx=On_Hand_FO_Lead_Normal_NormalAppx+On_Or
der_FO_Lead_Normal_NormalAppx;
Z_FO_Lead_Normal_NaddorAppx=On_Hand_FO_Lead_Normal_NaddorAppx+On_Or
der_FO_Lead_Normal_NaddorAppx;

[output t_Uniform]=
Check_t_Uniform(t_Uniform,Shape,Scale,Uniform_a,Uniform_b,deltat,On_Hand_FO_L
ead_Uniform_PowerAppx,On_Hand_FO_Lead_Uniform_NormalAppx,On_Hand_FO_L
ead_Uniform_NaddorAppx);
Period_Failure_Uniform=output(1);
Period_Maintenance_Uniform=output(2);
On_Hand_FO_Lead_Uniform_PowerAppx=output(3);
On_Hand_FO_Lead_Uniform_NormalAppx=output(4);
On_Hand_FO_Lead_Uniform_NaddorAppx=output(5);
Z_FO_Lead_Uniform_PowerAppx=On_Hand_FO_Lead_Uniform_PowerAppx+On_Ord
er_FO_Lead_Uniform_PowerAppx;
Z_FO_Lead_Uniform_NormalAppx=On_Hand_FO_Lead_Uniform_NormalAppx+On_
Order_FO_Lead_Uniform_NormalAppx;
Z_FO_Lead_Uniform_NaddorAppx=On_Hand_FO_Lead_Uniform_NaddorAppx+On_O
rder_FO_Lead_Uniform_NaddorAppx;
```

If a replacement occurs, re-estimate distribution values and optimal replacement times

```
if Period_Maintenance_Fixed+Period_Failure_Fixed>0
        [t_Fixed Failure_Data_Fixed Replacement_Data_Fixed
        Period_Censored_Fixed]=Add_Data_Fixed(t_Fixed,Shape,Scale,tstar_Fixed);
        Data_Combined_Fixed=Combine_Data2(n,Data_Combined_Fixed,Failure_Data_
        Fixed,Replacement_Data_Fixed,Period_Censored_Fixed);
        failures=0;
        censored=0;
        [max b]=size(Data_Combined_Fixed);
        Combined_Fixed=Data_Combined_Fixed;
        scrub=0;
        for i=1:max
                if Data_Combined_Fixed(i,2)==1
```

```
                    failures=failures+1;
            else
            if Data_Combined_Fixed(i,2)>=0
                    censored=censored+1;
            else
                    Combined_Fixed(i-scrub,:)=[];
            scrub=scrub+1;
            end
            end
    end
    Combined_Fixed=Combined_Fixed
    censored=censored
    failures=failures
    Shape_Fixed=Shape_Fixed
    best_method=input('Enter preferred estimation method (1-MLE, 2-KME, 3-
    PEXE, 4-FRWE, 5-KLM, 6-ROSS, 7-WH, 8-BE, 9-MPMLE, 10-Simulate, 11-
    NA): ');
    if best_method==1
            Shape_Old=Shape_Fixed;
            Scale_Old=Scale_Fixed;
            [Shape_Fixed Scale_Fixed]=MLE_Estimator(Combined_Fixed);
            ok=input('check: 1-Yes, 2-No: ')
            if ok==1
                    tstar_Fixed=Optimal_Replacement_Time(Shape_Fixed,Scale_Fixe
                    d,Cp,Cf);
            else
                    Shape_Fixed=Shape_Old
                    Scale_Fixed=Scale_Old
                    best_method=input('Enter preferred estimation method (1-MLE, 2-
                    KME, 3-PEXE, 4-FRWE, 5-KLM, 6-ROSS, 7-WH, 8-BE, 9-
                    MPMLE, 10-Simulate): ');
            end
    end
    if best_method==10
            [max b]=size(Combined_Fixed);
            flag=0;
            while flag==0
                    Total_Iterations=input('Enter simulation length: ');
                    Best_Estimator(max,Shape_Fixed,censored,Total_Iterations)
                    flag=input('Simulation sufficient? (0-No, 1-Yes): ');
            end
            best_method=input('Enter preferred estimation method (1-MLE, 2-KME,
            3-PEXE, 4-FRWE, 5-KLM, 6-ROSS, 7-WH, 8-BE, 9-MPMLE): ');
```

```
        end
        if best_method==2
                tstar_Fixed=Optimal_Replacement_Time_KME(Combined_Fixed,Cf,Cp);
        end
        if best_method==3

        tstar_Fixed=Optimal_Replacement_Time_PEXE(Combined_Fixed,Cf,Cp);
        end
        if best_method==4
                tstar_Fixed=Optimal_Replacement_Time_FRWE(Combined_Fixed,Cf,Cp
                );
        end
        if best_method==5
                tstar_Fixed=Optimal_Replacement_Time_KLM(Combined_Fixed,Cf,Cp);
        end
        if best_method==6
                [Shape_Fixed Scale_Fixed]=ROSS_Estimator(Combined_Fixed);
                tstar_Fixed=Optimal_Replacement_Time(Shape_Fixed,Scale_Fixed,Cp,C
                f);
        end
        if best_method==7
                [Shape_Fixed Scale_Fixed]=WH_Estimator(Combined_Fixed);
                tstar_Fixed=Optimal_Replacement_Time(Shape_Fixed,Scale_Fixed,Cp,C
                f);
        end
        if best_method==8
                [Shape_Fixed Scale_Fixed]=BE_Estimator(Combined_Fixed);
                tstar_Fixed=Optimal_Replacement_Time(Shape_Fixed,Scale_Fixed,Cp,C
                f);
        end
        if best_method==9
                [Shape_Fixed Scale_Fixed]=MPMLE_Estimator(Combined_Fixed);
                tstar_Fixed=Optimal_Replacement_Time(Shape_Fixed,Scale_Fixed,Cp,C
                f);
        end
end

if Period_Maintenance_Uniform+Period_Failure_Uniform>0
        [t_Uniform Failure_Data_Uniform Replacement_Data_Uniform
        Period_Censored_Uniform]=Add_Data_Uniform(t_Uniform,Shape,Scale,Unifor
        m_a,Uniform_b);
```

Appendix H (continued): Matlab Code

```
Data_Combined_Uniform=Combine_Data2(n,Data_Combined_Uniform,Failure_
Data_Uniform,Replacement_Data_Uniform,Period_Censored_Uniform);
failures=0;
censored=0;
[max b]=size(Data_Combined_Uniform);
Combined_Uniform=Data_Combined_Uniform;
scrub=0;
for i=1:max
        if Data_Combined_Uniform(i,2)==1
                failures=failures+1;
        else
        if Data_Combined_Uniform(i,2)>=0
                censored=censored+1;
        else
                Combined_Uniform(i-scrub,:)=[];
                scrub=scrub+1;
        end
        end
end
Combined_Uniform=Combined_Uniform
censored=censored
failures=failures
Shape_Uniform=Shape_Uniform
best_method=input('Enter preferred estimation method (1-MLE, 2-KME, 3-
PEXE, 4-FRWE, 5-KLM, 6-ROSS, 7-WH, 8-BE, 9-MPMLE, 10-Simulate, 11-
NA): ');
if best_method==1
        Shape_Old=Shape_Uniform;
        Scale_Old=Scale_Uniform;
        [Shape_Uniform Scale_Uniform]=MLE_Estimator(Combined_Uniform);
        ok=input('check: 1-Yes, 2-No: ')
        if ok==1
                tstar_Uniform=Optimal_Replacement_Time(Shape_Uniform,Scale
                _Uniform,Cp,Cf);
        else
                Shape_Uniform=Shape_Old
                Scale_Uniform=Scale_Old
                best_method=input('Enter preferred estimation method (1-MLE, 2-
                KME, 3-PEXE, 4-FRWE, 5-KLM, 6-ROSS, 7-WH, 8-BE, 9-
                MPMLE, 10-Simulate): ');
        end
end
if best_method==10
```

```
            [max b]=size(Combined_Uniform);
            flag=0;
            while flag==0
                  Total_Iterations=input('Enter simulation length: ');
                  Best_Estimator(max,Shape_Uniform,censored,Total_Iterations)
                  flag=input('Simulation sufficient? (0-No, 1-Yes): ');
            end
            best_method=input('Enter preferred estimation method (1-MLE, 2-KME,
            3-PEXE, 4-FRWE, 5-KLM, 6-ROSS, 7-WH, 8-BE, 9-MPMLE): ');
      end
      if best_method==2
            tstar_Uniform=Optimal_Replacement_Time_KME(Combined_Uniform,C
            f,Cp);
      end
      if best_method==3
            tstar_Uniform=Optimal_Replacement_Time_PEXE(Combined_Uniform,
            Cf,Cp);
      end
      if best_method==4
            tstar_Uniform=Optimal_Replacement_Time_FRWE(Combined_Uniform,
            Cf,Cp);
      end
      if best_method==5
            tstar_Uniform=Optimal_Replacement_Time_KLM(Combined_Uniform,C
            f,Cp);
      end
      if best_method==6
            [Shape_Uniform Scale_Uniform]=ROSS_Estimator(Combined_Uniform);
            tstar_Uniform=Optimal_Replacement_Time(Shape_Uniform,Scale_Unifo
            rm,Cp,Cf);
      end
      if best_method==7
            [Shape_Uniform Scale_Uniform]=WH_Estimator(Combined_Uniform);
            tstar_Uniform=Optimal_Replacement_Time(Shape_Uniform,Scale_Unifo
            rm,Cp,Cf);
      end
      if best_method==8
            [Shape_Uniform Scale_Uniform]=BE_Estimator(Combined_Uniform);
            tstar_Uniform=Optimal_Replacement_Time(Shape_Uniform,Scale_Unifo
            rm,Cp,Cf);
      end
      if best_method==9
```

```
            [Shape_Uniform Scale_Uniform]=
            MPMLE_Estimator(Combined_Uniform);
            tstar_Uniform=Optimal_Replacement_Time(Shape_Uniform,Scale_Unifo
            rm,Cp,Cf);
        end
        Uniform_a=tstar_Uniform-UR*tstar_Uniform;
end

if Period_Maintenance_Normal+Period_Failure_Normal>0
        [t_Normal Failure_Data_Normal Replacement_Data_Normal
        Period_Censored_Normal]=Add_Data_Normal(t_Normal,Shape,Scale,tstar_Nor
        mal,sigma);

        Data_Combined_Normal=Combine_Data2(n,Data_Combined_Normal,Failure_D
        ata_Normal,Replacement_Data_Normal,Period_Censored_Normal);
        failures=0;
        censored=0;
        [max b]=size(Data_Combined_Normal);
        Combined_Normal=Data_Combined_Normal;
        scrub=0;
        for i=1:max
            if Data_Combined_Normal(i,2)==1
                failures=failures+1;
            else
            if Data_Combined_Normal(i,2)>=0
                censored=censored+1;
            else
                Combined_Normal(i-scrub,:)=[];
                scrub=scrub+1;
            end
            end
        end
        Combined_Normal=Combined_Normal
        censored=censored
        failures=failures
        Shape_Normal=Shape_Normal
        best_method=input('Enter preferred estimation method (1-MLE, 2-KME, 3-
        PEXE, 4-FRWE, 5-KLM, 6-ROSS, 7-WH, 8-BE, 9-MPMLE, 10-Simulate, 11-
        NA): ');
        if best_method==10
            [max b]=size(Combined_Normal);
            flag=0;
            while flag==0
```

```
                Total_Iterations=input('Enter simulation length: ');
                Best_Estimator(max,Shape_Normal,censored,Total_Iterations)
                flag=input('Simulation sufficient? (0-No, 1-Yes): ');
        end
        best_method=input('Enter preferred estimation method (1-MLE, 2-KME,
        3-PEXE, 4-FRWE, 5-KLM, 6-ROSS, 7-WH, 8-BE, 9-MPMLE): ');
        end
if best_method==1
        Shape_Old=Shape_Normal;
        Scale_Old=Scale_Normal;
        [Shape_Normal Scale_Normal]=MLE_Estimator(Combined_Normal);
        ok=input('check: 1-Yes, 2-No: ')
        if ok==1
        tstar_Normal=Optimal_Replacement_Time(Shape_Normal,Scale_Normal,
        Cp,Cf);
        else
                Shape_Normal=Shape_Old
                Scale_Normal=Scale_Old
                best_method=input('Enter preferred estimation method (1-MLE, 2-
                KME, 3-PEXE, 4-FRWE, 5-KLM, 6-ROSS, 7-WH, 8-BE, 9-
                MPMLE, 10-Simulate): ');
        end
        end
if best_method==2
        tstar_Normal=Optimal_Replacement_Time_KME(Combined_Normal,Cf,
        Cp);
end
if best_method==3
        tstar_Normal=Optimal_Replacement_Time_PEXE(Combined_Normal,Cf
        ,Cp);
end
if best_method==4
        tstar_Normal=Optimal_Replacement_Time_FRWE(Combined_Normal,C
        f,Cp);
end
if best_method==5
        tstar_Normal=Optimal_Replacement_Time_KLM(Combined_Normal,Cf,
        Cp);
end
if best_method==6
        [Shape_Normal Scale_Normal]=ROSS_Estimator(Combined_Normal);
        tstar_Normal=Optimal_Replacement_Time(Shape_Normal,Scale_Normal,
        Cp,Cf);
```

```
        end
        if best_method==7
                [Shape_Normal Scale_Normal]=WH_Estimator(Combined_Normal);
                tstar_Normal=Optimal_Replacement_Time(Shape_Normal,Scale_Normal,
                Cp,Cf);
        end
         if best_method==8
                [Shape_Normal Scale_Normal]=BE_Estimator(Combined_Normal);
                tstar_Normal=Optimal_Replacement_Time(Shape_Normal,Scale_Normal,
                Cp,Cf);
        end
        if best_method==9
                [Shape_Normal Scale_Normal]=MPMLE_Estimator(Combined_Normal);
                tstar_Normal=Optimal_Replacement_Time(Shape_Normal,Scale_Normal,
                Cp,Cf);
        end

        sigma=Maintenance_Policy(PC,CR,tstar_Normal);
end
```

Determine the conditional probabilities and demand distribution for future periods

```
Pconditional_FO_Fixed=Conditional_Failure_Only(n,t_Fixed,k,deltat,Shape_Fixed,Scale
_Fixed);
Pconditional_FO_Normal=Conditional_Failure_Only(n,t_Normal,k,deltat,Shape_Normal
,Scale_Normal);
Pconditional_FO_Uniform=Conditional_Failure_Only(n,t_Uniform,k,deltat,Shape_Unifo
rm,Scale_Uniform);

Demand_Failure_Only_Fixed=Demand_Distribution(n,k,Pconditional_FO_Fixed);
Demand_Failure_Only_Normal=Demand_Distribution(n,k,Pconditional_FO_Normal);
Demand_Failure_Only_Uniform=Demand_Distribution(n,k,Pconditional_FO_Uniform);

FO_Demand_pdf_Fixed=kExpected_Demand(n,k,Demand_Failure_Only_Fixed);
FO_Demand_pdf_Normal=kExpected_Demand(n,k,Demand_Failure_Only_Normal);
FO_Demand_pdf_Uniform=kExpected_Demand(n,k,Demand_Failure_Only_Uniform);
```

Determine the initial inventory values

```
Demand_FO_Lead_Fixed=Expected_Demand(n,lamda+1,Demand_Failure_Only_Fixed)
;
[s_FO_Lead_Fixed_PowerAppx S_FO_Lead_Fixed_PowerAppx]=
Power_Approximation(lamda,p,h,K,Demand_FO_Lead_Fixed);
[s_FO_Lead_Fixed_NormalAppx S_FO_Lead_Fixed_NormalAppx]=
Normal_Approximation(lamda,p,h,K,Demand_FO_Lead_Fixed);
```

Appendix H (continued): Matlab Code

```
[s_FO_Lead_Fixed_NaddorAppx S_FO_Lead_Fixed_NaddorAppx]=
Naddor_Approximation(lamda,p,h,K,Demand_FO_Lead_Fixed);
FO_Lead_Fixed_Exp_Demand=Demand_FO_Lead_Fixed(1,1);
Demand_FO_Lead_Normal=Expected_Demand(n,lamda+1,Demand_Failure_Only_Nor
mal);
[s_FO_Lead_Normal_PowerAppx S_FO_Lead_Normal_PowerAppx]=
Power_Approximation(lamda,p,h,K,Demand_FO_Lead_Normal);
[s_FO_Lead_Normal_NormalAppx S_FO_Lead_Normal_NormalAppx]=
Normal_Approximation(lamda,p,h,K,Demand_FO_Lead_Normal);
[s_FO_Lead_Normal_NaddorAppx S_FO_Lead_Normal_NaddorAppx]=
Naddor_Approximation(lamda,p,h,K,Demand_FO_Lead_Normal);
FO_Lead_Normal_Exp_Demand=Demand_FO_Lead_Normal(1,1);
Demand_FO_Lead_Uniform=Expected_Demand(n,lamda+1,Demand_Failure_Only_Uni
form);
[s_FO_Lead_Uniform_PowerAppx S_FO_Lead_Uniform_PowerAppx]=
Power_Approximation(lamda,p,h,K,Demand_FO_Lead_Uniform);
[s_FO_Lead_Uniform_NormalAppx S_FO_Lead_Uniform_NormalAppx]=
Normal_Approximation(lamda,p,h,K,Demand_FO_Lead_Uniform);
[s_FO_Lead_Uniform_NaddorAppx S_FO_Lead_Uniform_NaddorAppx]=
Naddor_Approximation(lamda,p,h,K,Demand_FO_Lead_Uniform);
FO_Lead_Uniform_Exp_Demand=Demand_FO_Lead_Uniform(1,1);
```

Check if order needs to be placed

```
if Z_FO_Lead_Fixed_PowerAppx<=s_FO_Lead_Fixed_PowerAppx
        Lead_Orders_FO_Lead_Fixed_PowerAppx(Iteration+lamda)=ceil(S_FO_Lead_F
        ixed_PowerAppx-Z_FO_Lead_Fixed_PowerAppx);
        Period_Order_FO_Lead_Fixed_PowerAppx=1;
end
if Z_FO_Lead_Normal_PowerAppx<=s_FO_Lead_Normal_PowerAppx
        Lead_Orders_FO_Lead_Normal_PowerAppx(Iteration+lamda)=ceil(S_FO_Lead
        _Normal_PowerAppx-Z_FO_Lead_Normal_PowerAppx);
        Period_Order_FO_Lead_Normal_PowerAppx=1;
end
if Z_FO_Lead_Uniform_PowerAppx<=s_FO_Lead_Uniform_PowerAppx
        Lead_Orders_FO_Lead_Uniform_PowerAppx(Iteration+lamda)=ceil(S_FO_Lead
        _Uniform_PowerAppx-Z_FO_Lead_Uniform_PowerAppx);
        Period_Order_FO_Lead_Uniform_PowerAppx=1;
end
if Z_FO_Lead_Fixed_NormalAppx<=s_FO_Lead_Fixed_NormalAppx
        Lead_Orders_FO_Lead_Fixed_NormalAppx(Iteration+lamda)=ceil(S_FO_Lead_
        Fixed_NormalAppx-Z_FO_Lead_Fixed_NormalAppx);
        Period_Order_FO_Lead_Fixed_NormalAppx=1;
end
```

```
if Z_FO_Lead_Normal_NormalAppx<=s_FO_Lead_Normal_NormalAppx
        Lead_Orders_FO_Lead_Normal_NormalAppx(Iteration+lamda)=ceil(S_FO_Lea
        d_Normal_NormalAppx-Z_FO_Lead_Normal_NormalAppx);
        Period_Order_FO_Lead_Normal_NormalAppx=1;
end
if Z_FO_Lead_Uniform_NormalAppx<=s_FO_Lead_Uniform_NormalAppx
        Lead_Orders_FO_Lead_Uniform_NormalAppx(Iteration+lamda)=ceil(S_FO_Lea
        d_Uniform_NormalAppx-Z_FO_Lead_Uniform_NormalAppx);
        Period_Order_FO_Lead_Uniform_NormalAppx=1;
end
if Z_FO_Lead_Fixed_NaddorAppx<=s_FO_Lead_Fixed_NaddorAppx
        Lead_Orders_FO_Lead_Fixed_NaddorAppx(Iteration+lamda)=ceil(S_FO_Lead_
        Fixed_NaddorAppx-Z_FO_Lead_Fixed_NaddorAppx);
        Period_Order_FO_Lead_Fixed_NaddorAppx=1;
end
if Z_FO_Lead_Normal_NaddorAppx<=s_FO_Lead_Normal_NaddorAppx
        Lead_Orders_FO_Lead_Normal_NaddorAppx(Iteration+lamda)=ceil(S_FO_Lea
        d_Normal_NaddorAppx-Z_FO_Lead_Normal_NaddorAppx);
        Period_Order_FO_Lead_Normal_NaddorAppx=1;
end
if Z_FO_Lead_Uniform_NaddorAppx<=s_FO_Lead_Uniform_NaddorAppx
        Lead_Orders_FO_Lead_Uniform_NaddorAppx(Iteration+lamda)=ceil(S_FO_Lea
        d_Uniform_NaddorAppx-Z_FO_Lead_Uniform_NaddorAppx);
        Period_Order_FO_Lead_Uniform_NaddorAppx=1;
end
```

Total up holding costs

```
if On_Hand_FO_Lead_Fixed_PowerAppx>0
        Period_Holding_FO_Lead_Fixed_PowerAppx=On_Hand_FO_Lead_Fixed_Powe
        rAppx;
end
if On_Hand_FO_Lead_Normal_PowerAppx>0
        Period_Holding_FO_Lead_Normal_PowerAppx=On_Hand_FO_Lead_Normal_P
        owerAppx;
end
if On_Hand_FO_Lead_Uniform_PowerAppx>0
        Period_Holding_FO_Lead_Uniform_PowerAppx=On_Hand_FO_Lead_Uniform
        _PowerAppx;
end
if On_Hand_FO_Lead_Fixed_NormalAppx>0
        Period_Holding_FO_Lead_Fixed_NormalAppx=On_Hand_FO_Lead_Fixed_Nor
        malAppx;
end
```

```
if On_Hand_FO_Lead_Normal_NormalAppx>0
        Period_Holding_FO_Lead_Normal_NormalAppx=On_Hand_FO_Lead_Normal_
        NormalAppx;
end
if On_Hand_FO_Lead_Uniform_NormalAppx>0
        Period_Holding_FO_Lead_Uniform_NormalAppx=On_Hand_FO_Lead_Unifor
        m_NormalAppx;
end
if On_Hand_FO_Lead_Fixed_NaddorAppx>0
        Period_Holding_FO_Lead_Fixed_NaddorAppx=On_Hand_FO_Lead_Fixed_Nad
        dorAppx;
end
if On_Hand_FO_Lead_Normal_NaddorAppx>0
        Period_Holding_FO_Lead_Normal_NaddorAppx=On_Hand_FO_Lead_Normal_
        NaddorAppx;
end

if On_Hand_FO_Lead_Uniform_NaddorAppx>0
        Period_Holding_FO_Lead_Uniform_NaddorAppx=On_Hand_FO_Lead_Unifor
        m_NaddorAppx;
end
Total up shortage costs
if On_Hand_FO_Lead_Fixed_PowerAppx<0
        Period_Shortage_FO_Lead_Fixed_PowerAppx=abs(On_Hand_FO_Lead_Fixed_
        PowerAppx);
end
On_Order_FO_Lead_Fixed_PowerAppx=On_Order_FO_Lead_Fixed_PowerAppx+sum(
Lead_Orders_FO_Lead_Fixed_PowerAppx(Iteration+lamda:Simulation_Length+lamda))
;
Z_FO_Lead_Fixed_PowerAppx=On_Hand_FO_Lead_Fixed_PowerAppx+On_Order_F
O_Lead_Fixed_PowerAppx;
if On_Hand_FO_Lead_Normal_PowerAppx<0
        Period_Shortage_FO_Lead_Normal_PowerAppx=abs(On_Hand_FO_Lead_Nor
        mal_PowerAppx);
end
On_Order_FO_Lead_Normal_PowerAppx=On_Order_FO_Lead_Normal_PowerAppx+s
um(Lead_Orders_FO_Lead_Normal_PowerAppx(Iteration+lamda:Simulation_Length+la
mda));
Z_FO_Lead_Normal_PowerAppx=On_Hand_FO_Lead_Normal_PowerAppx+On_Order
_FO_Lead_Normal_PowerAppx;
if On_Hand_FO_Lead_Uniform_PowerAppx<0
        Period_Shortage_FO_Lead_Uniform_PowerAppx=abs(On_Hand_FO_Lead_Unif
        orm_PowerAppx);
```

```
end
On_Order_FO_Lead_Uniform_PowerAppx=On_Order_FO_Lead_Uniform_PowerAppx
+sum(Lead_Orders_FO_Lead_Uniform_PowerAppx(Iteration+lamda:Simulation_Lengt
h+lamda));
Z_FO_Lead_Uniform_PowerAppx=On_Hand_FO_Lead_Uniform_PowerAppx+On_Ord
er_FO_Lead_Uniform_PowerAppx;
if On_Hand_FO_Lead_Fixed_NormalAppx<0
        Period_Shortage_FO_Lead_Fixed_NormalAppx=abs(On_Hand_FO_Lead_Fixed
        _NormalAppx);
end
On_Order_FO_Lead_Fixed_NormalAppx=On_Order_FO_Lead_Fixed_NormalAppx+su
m(Lead_Orders_FO_Lead_Fixed_NormalAppx(Iteration+lamda:Simulation_Length+lam
da));
Z_FO_Lead_Fixed_NormalAppx=On_Hand_FO_Lead_Fixed_NormalAppx+On_Order_
FO_Lead_Fixed_NormalAppx;
if On_Hand_FO_Lead_Normal_NormalAppx<0
        Period_Shortage_FO_Lead_Normal_NormalAppx=abs(On_Hand_FO_Lead_Nor
        mal_NormalAppx);
end
On_Order_FO_Lead_Normal_NormalAppx=On_Order_FO_Lead_Normal_NormalAppx
+sum(Lead_Orders_FO_Lead_Normal_NormalAppx(Iteration+lamda:Simulation_Lengt
h+lamda));
Z_FO_Lead_Normal_NormalAppx=On_Hand_FO_Lead_Normal_NormalAppx+On_Or
der_FO_Lead_Normal_NormalAppx;
if On_Hand_FO_Lead_Uniform_NormalAppx<0
        Period_Shortage_FO_Lead_Uniform_NormalAppx=abs(On_Hand_FO_Lead_Un
        iform_NormalAppx);
end
On_Order_FO_Lead_Uniform_NormalAppx=On_Order_FO_Lead_Uniform_NormalAp
px+sum(Lead_Orders_FO_Lead_Uniform_NormalAppx(Iteration+lamda:Simulation_Le
ngth+lamda));
Z_FO_Lead_Uniform_NormalAppx=On_Hand_FO_Lead_Uniform_NormalAppx+On_
Order_FO_Lead_Uniform_NormalAppx;
if On_Hand_FO_Lead_Fixed_NaddorAppx<0
        Period_Shortage_FO_Lead_Fixed_NaddorAppx=abs(On_Hand_FO_Lead_Fixed
        _NaddorAppx);
end
On_Order_FO_Lead_Fixed_NaddorAppx=On_Order_FO_Lead_Fixed_NaddorAppx+su
m(Lead_Orders_FO_Lead_Fixed_NaddorAppx(Iteration+lamda:Simulation_Length+lam
da));
Z_FO_Lead_Fixed_NaddorAppx=On_Hand_FO_Lead_Fixed_NaddorAppx+On_Order_
FO_Lead_Fixed_NaddorAppx;
if On_Hand_FO_Lead_Normal_NaddorAppx<0
```

Appendix H (continued): Matlab Code

```
        Period_Shortage_FO_Lead_Normal_NaddorAppx=abs(On_Hand_FO_Lead_Nor
        mal_NaddorAppx);
end
On_Order_FO_Lead_Normal_NaddorAppx=On_Order_FO_Lead_Normal_NaddorAppx
+sum(Lead_Orders_FO_Lead_Normal_NaddorAppx(Iteration+lamda:Simulation_Lengt
h+lamda));
Z_FO_Lead_Normal_NaddorAppx=On_Hand_FO_Lead_Normal_NaddorAppx+On_Or
der_FO_Lead_Normal_NaddorAppx;
if On_Hand_FO_Lead_Uniform_NaddorAppx<0
        Period_Shortage_FO_Lead_Uniform_NaddorAppx=abs(On_Hand_FO_Lead_Un
        iform_NaddorAppx);
end
On_Order_FO_Lead_Uniform_NaddorAppx=On_Order_FO_Lead_Uniform_NaddorAp
px+sum(Lead_Orders_FO_Lead_Uniform_NaddorAppx(Iteration+lamda:Simulation_Le
ngth+lamda));
Z_FO_Lead_Uniform_NaddorAppx=On_Hand_FO_Lead_Uniform_NaddorAppx+On_O
rder_FO_Lead_Uniform_NaddorAppx;
```

Record inventory values and costs

```
Iteration=Iteration+1
Inventory_Parameters(Iteration,1)=On_Hand_FO_Lead_Fixed_PowerAppx;
Inventory_Parameters(Iteration,2)=s_FO_Lead_Fixed_PowerAppx;
Inventory_Parameters(Iteration,3)=S_FO_Lead_Fixed_PowerAppx;
Inventory_Parameters(Iteration,4)=On_Order_FO_Lead_Fixed_PowerAppx;
Inventory_Parameters(Iteration,5)=Period_Holding_FO_Lead_Fixed_PowerAppx;
Inventory_Parameters(Iteration,6)=Period_Shortage_FO_Lead_Fixed_PowerAppx;
Inventory_Parameters(Iteration-1,7)=Period_Order_FO_Lead_Fixed_PowerAppx;
Inventory_Parameters(Iteration,8)=On_Hand_FO_Lead_Fixed_NormalAppx;
Inventory_Parameters(Iteration,9)=s_FO_Lead_Fixed_NormalAppx;
Inventory_Parameters(Iteration,10)=S_FO_Lead_Fixed_NormalAppx;
Inventory_Parameters(Iteration,11)=On_Order_FO_Lead_Fixed_NormalAppx;
Inventory_Parameters(Iteration,12)=Period_Holding_FO_Lead_Fixed_NormalAppx;
Inventory_Parameters(Iteration,13)=Period_Shortage_FO_Lead_Fixed_NormalAppx;
Inventory_Parameters(Iteration-1,14)=Period_Order_FO_Lead_Fixed_NormalAppx;
Inventory_Parameters(Iteration,15)=On_Hand_FO_Lead_Fixed_NaddorAppx;
Inventory_Parameters(Iteration,16)=s_FO_Lead_Fixed_NaddorAppx;
Inventory_Parameters(Iteration,17)=S_FO_Lead_Fixed_NaddorAppx;
Inventory_Parameters(Iteration,18)=On_Order_FO_Lead_Fixed_NaddorAppx;
Inventory_Parameters(Iteration,19)=Period_Holding_FO_Lead_Fixed_NaddorAppx;
Inventory_Parameters(Iteration,20)=Period_Shortage_FO_Lead_Fixed_NaddorAppx;
Inventory_Parameters(Iteration-1,21)=Period_Order_FO_Lead_Fixed_NaddorAppx;
Inventory_Parameters(Iteration,22)=On_Hand_FO_Lead_Normal_PowerAppx;
Inventory_Parameters(Iteration,23)=s_FO_Lead_Normal_PowerAppx;
```

```
Inventory_Parameters(Iteration,24)=S_FO_Lead_Normal_PowerAppx;
Inventory_Parameters(Iteration,25)=On_Order_FO_Lead_Normal_PowerAppx;
Inventory_Parameters(Iteration,26)=Period_Holding_FO_Lead_Normal_PowerAppx;
Inventory_Parameters(Iteration,27)=Period_Shortage_FO_Lead_Normal_PowerAppx;
Inventory_Parameters(Iteration-1,28)=Period_Order_FO_Lead_Normal_PowerAppx;
Inventory_Parameters(Iteration,29)=On_Hand_FO_Lead_Normal_NormalAppx;
Inventory_Parameters(Iteration,30)=s_FO_Lead_Normal_NormalAppx;
Inventory_Parameters(Iteration,31)=S_FO_Lead_Normal_NormalAppx;
Inventory_Parameters(Iteration,32)=On_Order_FO_Lead_Normal_NormalAppx;
Inventory_Parameters(Iteration,33)=Period_Holding_FO_Lead_Normal_NormalAppx;
Inventory_Parameters(Iteration,34)=Period_Shortage_FO_Lead_Normal_NormalAppx;
Inventory_Parameters(Iteration-1,35)=Period_Order_FO_Lead_Normal_NormalAppx;
Inventory_Parameters(Iteration,36)=On_Hand_FO_Lead_Normal_NaddorAppx;
Inventory_Parameters(Iteration,37)=s_FO_Lead_Normal_NaddorAppx;
Inventory_Parameters(Iteration,38)=S_FO_Lead_Normal_NaddorAppx;
Inventory_Parameters(Iteration,39)=On_Order_FO_Lead_Normal_NaddorAppx;
Inventory_Parameters(Iteration,40)=Period_Holding_FO_Lead_Normal_NaddorAppx;
Inventory_Parameters(Iteration,41)=Period_Shortage_FO_Lead_Normal_NaddorAppx;
Inventory_Parameters(Iteration-1,42)=Period_Order_FO_Lead_Normal_NaddorAppx;
Inventory_Parameters(Iteration,43)=On_Hand_FO_Lead_Uniform_PowerAppx;
Inventory_Parameters(Iteration,44)=s_FO_Lead_Uniform_PowerAppx;
Inventory_Parameters(Iteration,45)=S_FO_Lead_Uniform_PowerAppx;
Inventory_Parameters(Iteration,46)=On_Order_FO_Lead_Uniform_PowerAppx;
Inventory_Parameters(Iteration,47)=Period_Holding_FO_Lead_Uniform_PowerAppx;
Inventory_Parameters(Iteration,48)=Period_Shortage_FO_Lead_Uniform_PowerAppx;
Inventory_Parameters(Iteration-1,49)=Period_Order_FO_Lead_Uniform_PowerAppx;
Inventory_Parameters(Iteration,50)=On_Hand_FO_Lead_Uniform_NormalAppx;
Inventory_Parameters(Iteration,51)=s_FO_Lead_Uniform_NormalAppx;
Inventory_Parameters(Iteration,52)=S_FO_Lead_Uniform_NormalAppx;
Inventory_Parameters(Iteration,53)=On_Order_FO_Lead_Uniform_NormalAppx;
Inventory_Parameters(Iteration,54)=Period_Holding_FO_Lead_Uniform_NormalAppx;
Inventory_Parameters(Iteration,55)=Period_Shortage_FO_Lead_Uniform_NormalAppx;
Inventory_Parameters(Iteration-1,56)=Period_Order_FO_Lead_Uniform_NormalAppx;
Inventory_Parameters(Iteration,57)=On_Hand_FO_Lead_Uniform_NaddorAppx;
Inventory_Parameters(Iteration,58)=s_FO_Lead_Uniform_NaddorAppx;
Inventory_Parameters(Iteration,59)=S_FO_Lead_Uniform_NaddorAppx;
Inventory_Parameters(Iteration,60)=On_Order_FO_Lead_Uniform_NaddorAppx;
Inventory_Parameters(Iteration,61)=Period_Holding_FO_Lead_Uniform_NaddorAppx;
Inventory_Parameters(Iteration,62)=Period_Shortage_FO_Lead_Uniform_NaddorAppx;
Inventory_Parameters(Iteration-1,63)=Period_Order_FO_Lead_Uniform_NaddorAppx;
```

Record maintenance values and costs

```
Policy_Parameters(Iteration-1,1)=Period_Maintenance_Fixed;
```

Appendix H (continued): Matlab Code

```
Policy_Parameters(Iteration-1,2)=Period_Failure_Fixed;
Policy_Parameters(Iteration,3)=FO_Lead_Fixed_Exp_Demand;
Policy_Parameters(Iteration,4)=abs(Period_Maintenance_Fixed+Period_Failure_Fixed-
FO_Lead_Fixed_Exp_Demand);
Policy_Parameters(Iteration-1,5)=Period_Maintenance_Fixed;
Policy_Parameters(Iteration-1,6)=Period_Failure_Fixed;
Policy_Parameters(Iteration,7)=FO_Lead_Fixed_Exp_Demand;
Policy_Parameters(Iteration,8)=abs(Period_Maintenance_Fixed+Period_Failure_Fixed-
FO_Lead_Fixed_Exp_Demand);
Policy_Parameters(Iteration-1,9)=Period_Maintenance_Fixed;
Policy_Parameters(Iteration-1,10)=Period_Failure_Fixed;
Policy_Parameters(Iteration,11)=FO_Lead_Fixed_Exp_Demand;
Policy_Parameters(Iteration,12)=abs(Period_Maintenance_Fixed+Period_Failure_Fixed-
FO_Lead_Fixed_Exp_Demand);
Policy_Parameters(Iteration-1,13)=Period_Maintenance_Normal;
Policy_Parameters(Iteration-1,14)=Period_Failure_Normal;
Policy_Parameters(Iteration,15)=FO_Lead_Normal_Exp_Demand;
Policy_Parameters(Iteration,16)=abs(Period_Maintenance_Normal+Period_Failure_Nor
mal-FO_Lead_Normal_Exp_Demand);
Policy_Parameters(Iteration-1,17)=Period_Maintenance_Normal;
Policy_Parameters(Iteration-1,18)=Period_Failure_Normal;
Policy_Parameters(Iteration,19)=FO_Lead_Normal_Exp_Demand;
Policy_Parameters(Iteration,20)=abs(Period_Maintenance_Normal+Period_Failure_Nor
mal-FO_Lead_Normal_Exp_Demand);
Policy_Parameters(Iteration-1,21)=Period_Maintenance_Normal;
Policy_Parameters(Iteration-1,22)=Period_Failure_Normal;
Policy_Parameters(Iteration,23)=FO_Lead_Normal_Exp_Demand;
Policy_Parameters(Iteration,24)=abs(Period_Maintenance_Normal+Period_Failure_Nor
mal-FO_Lead_Normal_Exp_Demand);
Policy_Parameters(Iteration-1,25)=Period_Maintenance_Uniform;
Policy_Parameters(Iteration-1,26)=Period_Failure_Uniform;
Policy_Parameters(Iteration,27)=FO_Lead_Uniform_Exp_Demand;
Policy_Parameters(Iteration,28)=abs(Period_Maintenance_Uniform+Period_Failure_Uni
form-FO_Lead_Uniform_Exp_Demand);
Policy_Parameters(Iteration-1,29)=Period_Maintenance_Uniform;
Policy_Parameters(Iteration-1,30)=Period_Failure_Uniform;
Policy_Parameters(Iteration,31)=FO_Lead_Uniform_Exp_Demand;
Policy_Parameters(Iteration,32)=abs(Period_Maintenance_Uniform+Period_Failure_Uni
form-FO_Lead_Uniform_Exp_Demand);
Policy_Parameters(Iteration-1,33)=Period_Maintenance_Uniform;
Policy_Parameters(Iteration-1,34)=Period_Failure_Uniform;
Policy_Parameters(Iteration,35)=FO_Lead_Uniform_Exp_Demand;
```

Appendix H (continued): Matlab Code

```
Policy_Parameters(Iteration,36)=abs(Period_Maintenance_Uniform+Period_Failure_Uniform-FO_Lead_Uniform_Exp_Demand);
```

Record distribution values

```
Distribution_Parameters(Iteration,1)=Shape_Fixed;
Distribution_Parameters(Iteration,2)=Scale_Fixed;
Distribution_Parameters(Iteration,3)=tstar_Fixed;
Distribution_Parameters(Iteration,4)=Shape_Normal;
Distribution_Parameters(Iteration,5)=Scale_Normal;
Distribution_Parameters(Iteration,6)=tstar_Normal;
Distribution_Parameters(Iteration,7)=Shape_Uniform;
Distribution_Parameters(Iteration,8)=Scale_Uniform;
Distribution_Parameters(Iteration,9)=tstar_Uniform;
```

Simulation initialization procedure

```
if Iteration==reset
    Inventory_Parameters(1,:)=Inventory_Parameters(Iteration,:);
    Policy_Parameters(1,:)=Policy_Parameters(Iteration,:);
    Iteration=1;
    reset=0;
    On_Hand_FO_Lead_Fixed_PowerAppx=ceil(S_FO_Lead_Fixed_PowerAppx);
    On_Hand_FO_Lead_Fixed_NormalAppx=ceil(S_FO_Lead_Fixed_NormalAppx)
    On_Hand_FO_Lead_Fixed_NaddorAppx=ceil(S_FO_Lead_Fixed_NaddorAppx);
    On_Hand_FO_Lead_Normal_PowerAppx=ceil(S_FO_Lead_Normal_PowerAppx);
    On_Hand_FO_Lead_Normal_NormalAppx=ceil(S_FO_Lead_Normal_NormalAppx);
    On_Hand_FO_Lead_Normal_NaddorAppx=ceil(S_FO_Lead_Normal_NaddorAppx);
    On_Hand_FO_Lead_Uniform_PowerAppx=ceil(S_FO_Lead_Uniform_PowerAppx);
    On_Hand_FO_Lead_Uniform_NormalAppx=ceil(S_FO_Lead_Uniform_NormalAppx);
    On_Hand_FO_Lead_Uniform_NaddorAppx=ceil(S_FO_Lead_Uniform_NaddorAppx);
    On_Order_FO_Lead_Fixed_PowerAppx=0;
    On_Order_FO_Lead_Fixed_NormalAppx=0;
    On_Order_FO_Lead_Fixed_NaddorAppx=0;
    On_Order_FO_Lead_Normal_PowerAppx=0;
    On_Order_FO_Lead_Normal_NormalAppx=0;
    On_Order_FO_Lead_Normal_NaddorAppx=0;
    On_Order_FO_Lead_Uniform_PowerAppx=0;
    On_Order_FO_Lead_Uniform_NormalAppx=0;
    On_Order_FO_Lead_Uniform_NaddorAppx=0;
```

Appendix H (continued): Matlab Code

```
        Z_FO_Lead_Fixed_PowerAppx=On_Hand_FO_Lead_Fixed_PowerAppx+On_O
        rder_FO_Lead_Fixed_PowerAppx;
        Z_FO_Lead_Fixed_NormalAppx=On_Hand_FO_Lead_Fixed_NormalAppx+On
        _Order_FO_Lead_Fixed_NormalAppx;
        Z_FO_Lead_Fixed_NaddorAppx=On_Hand_FO_Lead_Fixed_NaddorAppx+On_
        Order_FO_Lead_Fixed_NaddorAppx;
        Z_FO_Lead_Normal_PowerAppx=On_Hand_FO_Lead_Normal_PowerAppx+O
        n_Order_FO_Lead_Normal_PowerAppx;
        Z_FO_Lead_Uniform_PowerAppx=On_Hand_FO_Lead_Uniform_PowerAppx+
        On_Order_FO_Lead_Uniform_PowerAppx;
        Z_FO_Lead_Normal_NormalAppx=On_Hand_FO_Lead_Normal_NormalAppx+
        On_Order_FO_Lead_Normal_NormalAppx;
        Z_FO_Lead_Uniform_NormalAppx=On_Hand_FO_Lead_Uniform_NormalApp
        x+On_Order_FO_Lead_Uniform_NormalAppx;
        Z_FO_Lead_Normal_NaddorAppx=On_Hand_FO_Lead_Normal_NaddorAppx+
        On_Order_FO_Lead_Normal_NaddorAppx;
        Z_FO_Lead_Uniform_NaddorAppx=On_Hand_FO_Lead_Uniform_NaddorAppx
        +On_Order_FO_Lead_Uniform_NaddorAppx;
        Lead_Orders_FO_Lead_Fixed_PowerAppx=zeros(Simulation_Length+lamda+1,
        1);
        Lead_Orders_FO_Lead_Normal_PowerAppx=zeros(Simulation_Length+lamda+
        1,1);
        Lead_Orders_FO_Lead_Uniform_PowerAppx=zeros(Simulation_Length+lamda
        +1,1);
        Lead_Orders_FO_Lead_Fixed_NormalAppx=zeros(Simulation_Length+lamda+1
        ,1);
        Lead_Orders_FO_Lead_Normal_NormalAppx=zeros(Simulation_Length+lamda
        +1,1);
        Lead_Orders_FO_Lead_Uniform_NormalAppx=zeros(Simulation_Length+lamda
        +1,1);
        Lead_Orders_FO_Lead_Fixed_NaddorAppx=zeros(Simulation_Length+lamda+1
        ,1);
        Lead_Orders_FO_Lead_Normal_NaddorAppx=zeros(Simulation_Length+lamda
        +1,1);
        Lead_Orders_FO_Lead_Uniform_NaddorAppx=zeros(Simulation_Length+lamda
        +1,1);
        t=t_Fixed(:,1);
end
```

End of simulation

```
End
```

Sum up all costs and record

```
j=1;
```

Appendix H (continued): Matlab Code

```
for i=0:4:32
        Output(5,j)=sum(Policy_Parameters(:,i+1));
        Output(6,j)=sum(Policy_Parameters(:,i+2));
        Output(7,j)=sum(Policy_Parameters(:,i+4))/Iteration;
        j=j+1;
end
j=1;
for i=0:7:56
        Output(1,j)=sum(Inventory_Parameters(:,i+5))*h*deltat+sum(Inventory_Paramet
        ers(:,i+6))*p*deltat+sum(Inventory_Parameters(:,i+7))*K+Output(5,j)*Cp+Outp
        ut(6,j)*Cf;
        Output(2,j)=sum(Inventory_Parameters(:,i+5));
        Output(3,j)=sum(Inventory_Parameters(:,i+6));
        Output(4,j)=sum(Inventory_Parameters(:,i+7));
        Output(8,j)=tstar;
        j=j+1;
end
```

Appendix I: Run 10 Distribution Parameters

	Policy 10			Policy 11			Policy 12		
Period	Shape	Scale	t*	Shape	Scale	t*	Shape	Scale	t*
1	3.0	50.0	38.0	3.0	50.0	38.0	3.0	50.0	38.0
2	3.0	50.0	38.0	3.0	50.0	38.0	3.0	50.0	38.0
3	3.0	50.0	38.0	3.0	50.0	45.5	3.0	50.0	38.0
4	3.0	50.0	38.0	3.0	50.0	45.5	3.0	50.0	38.0
5	3.0	50.0	38.0	3.0	50.0	45.5	3.0	50.0	38.0
6	3.0	50.0	38.0	3.0	50.0	45.5	3.0	50.0	38.0
7	3.0	50.0	38.0	3.0	50.0	45.5	3.0	50.0	38.0
8	3.0	50.0	38.0	3.0	50.0	45.5	3.0	50.0	38.0
9	3.0	50.0	38.0	3.0	50.0	45.5	3.0	50.0	38.0
10	3.0	50.0	38.0	3.0	50.0	45.5	3.0	50.0	38.0
11	3.0	50.0	38.0	3.0	50.0	45.5	3.0	50.0	38.0
12	3.0	50.0	38.0	3.0	50.0	45.5	3.0	50.0	38.0
13	3.0	50.0	38.0	3.0	50.0	45.5	3.0	50.0	38.0
14	3.0	50.0	123.5	3.0	50.0	45.5	3.0	50.0	38.0
15	3.0	50.0	123.5	3.0	50.0	45.5	3.0	50.0	38.0
16	3.0	50.0	30.5	3.0	50.0	45.5	3.0	50.0	38.0
17	3.0	50.0	30.5	3.0	50.0	45.5	3.0	50.0	38.0
18	3.1	77.3	58.5	3.0	50.0	45.5	3.0	50.0	38.0
19	3.1	77.3	58.5	3.0	50.0	131.5	3.0	50.0	38.0
20	3.3	77.2	57.5	3.0	50.0	131.5	3.0	50.0	38.0
21	3.3	77.2	57.5	3.0	50.0	131.5	3.0	50.0	38.0
22	3.3	77.2	57.5	3.0	50.0	131.5	3.0	50.0	38.0
23	2.5	105.2	86.0	3.0	50.0	131.5	3.0	50.0	38.0
24	2.5	105.2	86.0	3.0	50.0	34.5	3.0	50.0	38.0
25	2.5	105.2	29.5	3.0	50.0	34.5	3.0	50.0	38.0
26	2.5	105.2	29.5	4.3	86.8	63.5	3.0	50.0	38.0
27	2.5	105.2	29.5	4.3	86.8	63.5	3.0	50.0	38.0
28	2.5	105.2	29.5	4.3	86.8	63.5	3.0	50.0	38.0
29	1.9	130.6	136.5	4.3	86.8	34.5	3.0	50.0	38.0
30	1.9	130.6	136.5	4.3	86.8	34.5	3.0	50.0	38.0
31	2.5	98.3	80.0	4.3	86.8	34.5	3.0	50.0	38.0
32	2.5	98.3	80.0	4.3	86.8	34.5	3.0	50.0	38.0
33	3.5	77.4	57.5	4.3	86.8	34.5	3.0	50.0	38.0
34	3.5	77.4	57.5	4.3	86.8	34.5	3.0	50.0	118.0
35	3.5	77.4	57.5	2.8	143.9	112.0	3.0	50.0	118.0
36	3.5	77.4	57.5	2.8	143.9	112.0	3.0	50.0	118.0
37	3.3	82.5	61.5	2.8	141.2	110.0	3.0	50.0	118.0
38	3.3	82.5	61.5	2.8	141.2	110.0	4.7	77.7	57.0
39	3.3	82.5	61.5	2.9	141.0	108.5	4.7	77.7	57.0
40	2.5	106.8	87.5	2.9	141.0	108.5	4.0	90.4	66.5
41	2.5	106.8	87.5	2.9	141.0	108.5	4.0	90.4	66.5
42	2.3	111.2	95.5	2.9	141.0	108.5	4.0	90.4	34.0
43	2.3	111.2	95.5	3.0	142.9	109.0	4.0	90.4	34.0
44	2.3	111.2	95.5	3.0	142.9	109.0	2.4	168.9	143.0
45	2.0	131.4	126.5	3.1	124.6	94.5	2.4	168.9	143.0
46	2.0	131.4	126.5	3.1	124.6	94.5	3.0	128.5	97.5
47	2.2	115.7	104.5	3.0	132.1	101.0	3.0	128.5	97.5
48	2.2	115.7	104.5	3.0	132.1	101.0	3.4	118.2	88.0
49	2.2	115.7	104.5	3.3	122.7	92.0	3.4	118.2	88.0
50	2.2	115.7	104.5	3.3	122.7	92.0	3.1	128.2	97.0
51	2.3	109.8	95.0	3.3	115.4	86.0	3.1	128.2	97.0

Appendix I (continued): Run 10 Distribution Parameters

	Policy 10			Policy 11			Policy 12		
Period	Shape	Scale	t*	Shape	Scale	t*	Shape	Scale	t*
52	2.3	109.8	95.0	3.3	115.4	86.0	3.1	128.2	97.0
53	2.5	103.7	85.5	3.4	110.4	82.0	3.3	124.6	93.0
54	2.5	103.7	85.5	3.4	110.4	82.0	3.3	124.6	93.0
55	2.5	102.9	84.0	3.4	110.4	82.0	3.3	127.0	95.0
56	2.5	102.9	84.0	3.4	115.7	86.0	3.3	127.0	95.0
57	2.5	102.9	84.0	3.4	115.7	86.0	3.2	133.8	100.5
58	2.5	106.5	87.5	3.5	111.0	82.0	3.2	133.8	100.5
59	2.5	106.5	87.5	3.5	111.0	82.0	3.8	110.5	81.0
60	2.5	107.8	87.5	3.6	105.6	78.0	3.8	110.5	81.0
61	2.5	107.8	87.5	3.6	105.6	78.0	3.8	110.5	81.0
62	2.5	110.9	91.0	3.5	103.2	76.5	3.8	110.5	81.0
63	2.5	110.9	91.0	3.5	103.2	76.5	3.7	110.9	81.5
64	2.4	113.9	94.5	3.6	104.1	77.0	3.7	110.9	81.5
65	2.4	113.9	94.5	3.6	104.1	77.0	3.7	110.9	81.5
66	2.5	110.5	89.5	3.6	104.1	77.0	3.7	110.9	81.5
67	2.5	110.5	89.5	3.6	104.1	77.0	3.7	110.9	81.5
68	2.5	110.5	89.5	3.7	105.8	78.0	3.8	107.7	79.0
69	2.5	111.7	91.5	3.7	105.8	78.0	3.8	107.7	79.0
70	2.5	111.7	91.5	3.6	107.0	79.0	3.8	108.1	79.5
71	2.4	111.7	93.0	3.6	107.0	79.0	3.8	108.1	79.5
72	2.4	111.7	93.0	3.6	106.7	78.5	3.9	103.8	76.0
73	2.4	111.7	93.0	3.6	106.7	78.5	3.9	103.8	76.0
74	2.4	114.7	95.5	3.4	106.3	79.0	3.9	106.3	78.0
75	2.4	114.7	95.5	3.4	106.3	79.0	3.9	106.3	78.0
76	2.5	112.8	93.0	3.4	107.7	80.0	3.9	106.3	78.0
77	2.5	112.8	93.0	3.4	107.7	80.0	4.0	104.5	76.5
78	2.5	113.6	93.5	3.4	107.6	80.0	4.0	104.5	76.5
79	2.5	113.6	93.5	3.4	107.6	80.0	4.1	105.2	77.0
80	2.6	106.7	85.0	3.4	106.2	79.0	4.1	105.2	77.0
81	2.6	106.7	85.0	3.4	106.2	79.0	4.1	105.4	77.0
82	2.6	104.8	83.5	3.4	106.2	79.0	4.1	105.4	77.0
83	2.6	104.8	83.5	3.4	107.8	80.0	4.1	106.4	78.0
84	2.7	105.0	83.5	3.4	107.8	80.0	4.1	106.4	78.0
85	2.7	105.0	83.5	3.5	105.1	78.0	4.1	106.4	78.0
86	2.7	106.8	84.5	3.5	105.1	78.0	3.8	106.1	78.0
87	2.7	106.8	84.5	3.5	102.2	75.5	3.8	106.1	78.0
88	2.7	108.1	86.0	3.5	102.2	75.5	3.9	106.4	78.0
89	2.7	108.1	86.0	3.5	102.2	75.5	3.9	106.4	78.0
90	2.6	109.3	87.5	3.6	101.2	75.0	3.9	106.4	78.0
91	2.6	109.3	87.5	3.6	101.2	75.0	3.7	105.5	78.0
92	2.6	108.8	87.0	3.6	101.9	75.5	3.7	105.5	78.0
93	2.6	108.8	87.0	3.6	101.9	75.5	3.6	104.4	77.0
94	2.6	108.8	87.0	3.6	101.9	75.5	3.6	104.4	77.0
95	2.7	105.8	83.5	3.6	102.4	75.5	3.4	104.0	77.0
96	2.7	105.8	83.5	3.6	102.4	75.5	3.4	104.0	77.0
97	2.7	107.5	85.0	3.6	101.3	75.0	3.4	104.0	77.0
98	2.7	107.5	85.0	3.6	101.3	75.0	3.5	105.6	78.5
99	2.7	107.5	85.0	3.5	101.4	75.0	3.5	105.6	78.5
Mean=	2.5	108.7	88.3	3.5	105.9	78.5	3.7	109.9	81.1

Appendix J: Run 11 Distribution Parameters

	Policy 10			Policy 11			Policy 12		
Period	Shape	Scale	*t**	Shape	Scale	*t**	Shape	Scale	*t**
1	3.0	150.0	79.0	6.7	91.1	68.5	4.8	95.6	70.0
2	2.0	169.9	169.0	7.3	92.8	70.5	4.8	94.4	69.5
3	2.0	169.9	169.0	7.3	92.8	70.5	4.8	94.4	69.5
4	2.1	172.3	162.5	7.3	92.8	70.5	4.8	94.4	69.5
5	2.1	172.3	162.5	7.3	92.8	70.5	4.3	105.5	77.0
6	2.2	148.9	131.5	5.5	113.8	84.0	4.3	105.5	77.0
7	2.2	148.9	131.5	5.5	113.8	84.0	4.1	103.7	76.0
8	2.2	151.2	133.0	5.2	112.1	82.5	4.1	103.7	76.0
9	2.2	151.2	133.0	5.2	112.1	82.5	3.9	107.0	78.5
10	2.2	122.4	107.5	5.2	111.9	82.5	3.9	107.0	78.5
11	2.2	122.4	107.5	5.2	111.9	82.5	3.0	106.6	81.5
12	2.2	114.4	101.0	5.2	111.9	82.5	3.0	106.6	81.5
13	2.2	114.4	101.0	5.8	112.3	83.5	3.1	108.6	82.5
14	2.4	109.1	92.0	5.8	112.3	83.5	3.1	108.6	82.5
15	2.4	109.1	92.0	5.9	113.3	84.5	3.1	108.6	82.5
16	2.4	109.1	92.0	5.9	113.3	84.5	3.3	102.3	76.5
17	2.4	109.1	92.0	5.7	105.4	78.0	3.3	102.3	76.5
18	2.6	107.9	86.0	5.7	105.4	78.0	3.3	102.3	76.5
19	2.6	107.9	86.0	5.8	105.4	78.5	3.5	101.8	75.5
20	2.6	107.9	86.0	5.8	105.4	78.5	3.5	101.8	75.5
21	2.6	107.9	86.0	5.8	105.4	78.5	3.5	103.3	76.5
22	2.4	117.5	99.5	5.8	105.4	78.5	3.5	103.3	76.5
23	2.4	117.5	99.5	5.8	106.3	79.0	3.5	103.3	76.5
24	2.3	122.2	107.0	5.8	106.3	79.0	3.5	106.4	78.5
25	2.3	122.2	107.0	5.7	106.6	79.0	3.5	106.4	78.5
26	2.3	118.8	101.5	5.7	106.6	79.0	3.5	106.4	78.5
27	2.3	118.8	101.5	5.7	106.6	79.0	3.4	106.0	78.5
28	2.3	118.8	101.5	5.6	106.3	79.0	3.4	106.0	78.5
29	2.3	118.8	101.5	5.6	106.3	79.0	3.4	106.0	78.5
30	2.4	123.2	104.0	5.9	106.7	79.5	3.4	106.0	78.5
31	2.4	123.2	104.0	5.9	106.7	79.5	3.4	106.0	78.5
32	2.5	118.5	98.0	4.9	108.6	80.0	3.6	107.2	79.0
33	2.5	118.5	98.0	4.9	108.6	80.0	3.6	107.2	79.0
34	2.5	118.5	98.0	4.9	108.4	79.5	3.5	108.8	80.5
35	2.5	121.6	99.5	4.9	108.4	79.5	3.5	108.8	80.5
36	2.5	121.6	99.5	4.9	108.4	79.5	3.5	106.9	79.5
37	2.5	119.8	98.0	5.1	105.4	77.5	3.5	106.9	79.5
38	2.5	119.8	98.0	5.1	105.4	77.5	3.5	107.8	80.0
39	2.5	119.8	98.0	5.4	103.3	76.5	3.5	107.8	80.0
40	2.6	119.9	96.5	5.4	103.3	76.5	3.4	106.9	79.5
41	2.6	119.9	96.5	5.6	102.7	76.0	3.4	106.9	79.5
42	2.6	119.3	96.0	5.6	102.7	76.0	3.4	106.9	79.5
43	2.6	119.3	96.0	5.6	102.5	76.0	3.5	105.4	78.0
44	2.6	119.3	96.0	5.6	102.5	76.0	3.5	105.4	78.0
45	2.6	118.0	95.0	5.6	102.5	76.0	3.7	102.7	75.5
46	2.6	118.0	95.0	5.7	103.0	76.5	3.7	102.7	75.5
47	2.6	118.0	95.0	5.7	103.0	76.5	3.7	102.7	75.5
48	2.6	115.0	92.5	5.8	103.4	77.0	3.7	105.5	77.5
49	2.6	115.0	92.5	5.8	103.4	77.0	3.7	105.5	77.5
50	2.6	115.0	92.5	5.9	104.4	77.5	3.7	105.8	78.0
51	2.6	116.4	92.5	5.9	104.4	77.5	3.7	105.8	78.0

Appendix J (continued): Run 11 Distribution Parameters

	Policy 10			Policy 11			Policy 12		
Period	Shape	Scale	$t*$	Shape	Scale	$t*$	Shape	Scale	$t*$
52	2.6	116.4	92.5	5.9	104.4	77.5	3.7	105.8	78.0
53	2.6	116.4	92.5	6.0	104.4	78.0	3.7	107.0	79.0
54	2.6	116.8	93.5	6.0	104.4	78.0	3.7	107.0	79.0
55	2.6	116.8	93.5	6.0	104.4	78.0	3.6	105.7	78.0
56	2.7	115.4	91.5	5.5	104.6	77.5	3.6	105.7	78.0
57	2.7	115.4	91.5	5.5	104.6	77.5	3.6	105.7	78.0
58	2.6	115.8	92.5	5.4	103.5	76.5	3.7	106.8	78.5
59	2.6	115.8	92.5	5.4	103.5	76.5	3.7	106.8	78.5
60	2.7	115.8	91.5	5.4	103.5	76.5	3.7	107.4	79.0
61	2.7	115.8	91.5	5.4	103.5	76.5	3.7	107.4	79.0
62	2.7	115.8	91.5	5.5	103.4	76.5	3.7	108.3	79.5
63	2.7	115.8	91.5	5.5	103.4	76.5	3.7	108.3	79.5
64	2.7	115.8	91.5	5.5	103.4	76.5	3.7	105.0	77.5
65	2.8	113.3	88.0	5.5	103.8	77.0	3.7	105.0	77.5
66	2.8	113.3	88.0	5.5	103.8	77.0	3.8	101.7	75.0
67	2.8	113.3	88.0	5.5	102.6	76.0	3.8	101.7	75.0
68	2.8	114.3	89.0	5.5	102.6	76.0	3.8	101.7	75.0
69	2.8	114.3	89.0	5.6	102.7	76.0	3.8	101.7	75.0
70	2.8	113.9	88.5	5.6	102.7	76.0	3.9	102.9	75.5
71	2.8	113.9	88.5	5.6	102.7	76.0	3.9	102.9	75.5
72	2.8	113.6	88.0	5.7	103.2	76.5	3.5	102.1	75.5
73	2.8	113.6	88.0	5.7	103.2	76.5	3.5	102.1	75.5
74	2.8	113.6	88.0	5.7	103.2	76.5	3.5	102.4	76.0
75	2.8	113.6	88.0	5.2	102.3	75.5	3.5	102.4	76.0
76	2.8	113.6	88.0	5.2	102.3	75.5	3.5	101.8	75.5
77	2.8	113.7	88.5	5.2	102.1	75.0	3.5	101.8	75.5
78	2.8	113.7	88.5	5.2	102.1	75.0	3.6	102.6	76.0
79	2.8	114.0	88.0	5.2	102.2	75.5	3.6	102.6	76.0
80	2.8	114.0	88.0	5.2	102.2	75.5	3.6	101.2	74.5
81	2.8	114.0	88.0	5.2	101.5	75.0	3.6	101.2	74.5
82	2.9	113.5	87.5	5.2	101.5	75.0	3.6	102.3	75.5
83	2.9	113.5	87.5	5.2	101.5	75.0	3.6	102.3	75.5
84	2.9	114.5	88.0	5.3	100.9	74.5	3.6	101.6	75.0
85	2.9	114.5	88.0	5.3	100.9	74.5	3.6	101.6	75.0
86	3.0	110.1	84.5	5.3	100.9	74.5	3.6	101.6	75.0
87	3.0	110.1	84.5	5.3	100.9	74.5	3.6	101.6	75.0
88	3.0	110.1	84.5	5.4	101.4	75.0	3.7	101.5	74.5
89	3.0	110.3	84.0	5.4	101.4	75.0	3.7	101.5	74.5
90	3.0	110.3	84.0	5.4	101.4	75.0	3.7	101.6	75.0
91	3.0	110.3	84.0	5.3	100.9	74.5	3.7	101.6	75.0
92	3.0	112.5	85.5	5.3	100.9	74.5	3.7	102.6	75.5
93	3.0	112.5	85.5	5.3	100.9	74.5	3.7	102.6	75.5
94	3.0	112.5	85.5	5.1	100.7	74.0	3.7	102.6	75.5
95	3.0	113.5	86.5	5.1	100.7	74.0	3.7	102.6	75.5
96	3.0	113.5	86.5	5.1	100.7	74.0	3.7	102.6	75.5
97	3.0	110.9	84.0	5.1	100.7	74.0	3.7	102.6	75.5
98	3.0	110.9	84.0	5.2	99.9	73.5	3.6	102.7	75.5
99	3.0	110.9	84.0	5.2	99.9	73.5	3.6	102.7	75.5
Mean=	2.6	119.2	96.8	5.6	104.0	77.1	3.6	104.1	77.0

ABOUT THE AUTHOR

Jonathan J. Bates received a B.S. in Naval Architecture and Marine Engineering from the United States Coast Guard Academy in 1998. As an officer in the United States Coast Guard, he has served two years as Assistant Engineer Officer on USCGC VIGILANT (WMEC 617), three years as Port Engineer at Naval Engineering Support Unit Portsmouth, Virginia, and is currently serving as Engineer Officer onboard USCGC CONFIDENCE (WMEC 619). In 2003, he earned a M.S. in Ocean Engineering from Virginia Tech and a Professional Engineer license from the State of Florida in 2004. Upon completion of this dissertation, he received a Ph.D. in Industrial Engineering from the University of South Florida.