12-2012

On The Maintenance Modeling and Optimization of Repairable Systems: Two Different Scenarios

Suzan Alaswad
University of Arkansas, Fayetteville

ON THE MAINTENANCE MODELING AND OPTIMIZATION OF REPAIRABLE SYSTEMS: TWO DIFFERENT SCENARIOS

ON THE MAINTENANCE MODELING AND OPTIMIZATION OF REPAIRABLE SYSTEMS: TWO DIFFERENT SCENARIOS

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy in Industrial Engineering

By

Suzan M. Alaswad
Jordan University of Science and Technology
Bachelor of Science in Mechanical Engineering, 2000
University of Arkansas
Master of Science in Industrial Engineering, 2005

December 2012
University of Arkansas

ABSTRACT

The use of mathematical modeling for the purpose of analyzing and optimizing the performance of repairable systems is widely studied in the literature. In this dissertation, we study two different scenarios on the maintenance modeling and optimization of repairable systems. First, we study the long-run availability of a traditional repairable system that is subjected to imperfect corrective maintenance. We use Kijima's second virtual age model to describe the imperfect repair process. Because of the complexity of the underlying probability models, we use simulation modeling to estimate availability performance and meta-modeling to convert the reliability and maintainability parameters of the repairable system into an availability estimate without the simulation effort. As a last step, we add age-based, perfect preventive maintenance to our analysis. Second, we optimize a preventive maintenance policy for a two-component repairable system. When either component fails, instantaneous, minimal, and costly corrective maintenance is performed on the component. At equally-spaced, discrete points during the system's useful life, the decision-maker has the option to perform instantaneous, imperfect, and costly preventive maintenance on one or both of the components, to instantaneously replace one or both of the components, or to do nothing. We use a Genetic Algorithm in an attempt to find a cost-optimal set of preventive maintenance and replacement decisions.

This dissertation is approved for recommendation
to the Graduate Council.

Dissertation Director:

____
Dr. C. Richard Cassady

Dissertation Committee:

____
Dr. Darin W. Nutter

____
Dr. Edward A. Pohl

____
Dr. Chase Rainwater

DISSERTATION DUPLICATION RELEASE

I hereby authorize the University of Arkansas Libraries to duplicate this dissertation when needed for research and/or scholarship.

Agreed ____

Suzan M. Alaswad

Refused ____

Suzan M. Alaswad

ACKNOWLEDGMENTS

In the name of Allah, the most gracious, the most merciful

I would like to thank my husband Sinan. His support, encouragement, patience, tolerance of my occasional frustrations, and unconditional love were definitely the bedrock upon which the past ten years of my life have been built. I'm also greatly indebted to my children, Lina, Layann, and Abdurahman. They are my source of happiness. I owe my every achievement to my husband and children.

Moreover, I would like to give my sincere thanks to my advisor, Dr. C. Richard Cassady, for his assistance and guidance through what seemed to be a never-ending process. The completion of this dissertation wouldn't be possible without his support and direction. Special thanks are also extended to my committee members, Dr. Darin W. Nutter, Dr. Edward A. Pohl, and Dr. Chase Rainwater, for their valuable comments and suggestions.

DEDICATION

This dissertation is dedicated to my Mother and Father whom I lost at an early age of my life, may their souls rest in peace. I wish they could have lived long enough to see me accomplish this long-awaited goal.

TABLE OF CONTENTS

1. Introduction

All industrial organizations depend on the effective and efficient operation of systems that are subject to and maintained upon failure. If maintenance options other than system replacement (e.g., repair) are available, then such a system is referred to as a repairable system (RS). The proper maintenance of a RS is a challenge faced by engineers in all industries; unfortunately, this challenge often is not met productively. For example, Mobley (1988) estimates that a manufacturer's maintenance costs represent 15-40% of the cost of goods produced, but approximately one-third of all maintenance costs are associated with unnecessary or incorrect maintenance actions. Among the potential causes of these difficulties are a "necessary evil" view of maintenance – maintain only upon system failure – and an unscientific approach to maintenance decision-making.

1.1 Maintenance Actions

Maintenance actions performed on a RS can be categorized into two groups: corrective maintenance (CM) actions and preventive maintenance (PM) actions. CM actions are performed in response to system failures, and they could correspond to either repair or replacement activities. PM actions are not performed in response to RS failure, but they are intended to delay or prevent RS failures. Note that PM actions may or may not be cheaper and/or faster than CM actions. As with CM actions, PM actions can correspond to either repair or replacement activities

PM actions can be divided into two sub-categories. Scheduled maintenance (SM) actions are planned based on some measure of elapsed time. Condition-based maintenance (CBM) actions are initiated based on data obtained from sensors applied to the RS. Vibration data and chemical analysis data are two examples of the type of data used in CBM. CBM provides the potential for just-in-time maintenance.

1.2 Repairable Systems Modeling

Repairable systems modeling refers to the application of operations research techniques (e.g., probability modeling, optimization, simulation) to problems related to equipment maintenance. Repairable system models are typically used to evaluate the performance of one or more repairable systems and/or design maintenance policies for one or more repairable systems. The literature on the use of mathematical modeling for the purpose of analyzing and optimizing the performance of repairable systems is extensive. McCall (1965), Pierskalla and Voelker (1976), Sherif and Smith (1981), Cho and Parlar (1981), Dekker (1996), and Wang (2002) all provide surveys of this literature. The work summarized in these papers captures a wide variety of underlying assumptions and modeling approaches.

The vast majority of published work in repairable systems modeling treats RS that conform to six assumptions.

1. The RS is comprised of a single component.
2. The RS has binary status. At any point in time, the RS is either functioning or down for maintenance.
3. The RS has self-announcing failures. Inspection is not required to determine the status of the RS.
4. The RS is intended to function continuously. The RS has no planned downtime.
5. The RS is "as good as new" at time zero.
6. The RS is subjected to either no PM or SM.

Given these assumptions, a traditional model of a RS requires specification of six characteristics.

1. A probability distribution that governs the time until the first RS failure must be specified. The Weibull and exponential probability distributions are common choices.
2. A mathematical model of the duration of CM must be specified. Common choices here include instantaneous CM, a generic probability distribution, the exponential probability distribution, and the lognormal probability distribution.
3. A mathematical model of the impact of CM must be specified. The most common choice here is perfect CM. After perfect CM, the RS is in an "as good as new" state. The second most common choice is minimal CM. After minimal CM, the RS is in an "as bad as old" condition – the RS is functioning but its age is equivalent to its age at the instant failure occurred. Imperfect maintenance refers to a class of models that describe CM impact that is worse than perfect but better than minimal (Pham and Wang, 1996).
4. If SM is utilized, then the policy that governs SM must be specified. Two common policies are age-based PM and block-based PM. Under an age-based PM policy, PM is initiated if the RS functions without failure for a specified period of time. Under a block-based policy, PM is initiated at equally-spaced points in time.
5. If SM is utilized, then the duration of CM must be modeled.
6. If SM is utilized, then the impact of PM must be modeled.

1.3 Contribution

In this dissertation, we make two contributions to the repairable systems modeling literature. In Chapter 2, we study the long-run availability of a traditional RS that is subjected to imperfect CM. Because of the complexity of the underlying probability models, we cannot derive a closed-form expression for the availability of the RS. Therefore, we use simulation modeling to estimate availability performance and meta-modeling to convert the reliability and maintainability parameters of the RS into an availability estimate without the simulation effort. As a final step, we add age-based, perfect PM to our analysis.

In Chapter 3, we optimize a PM policy for a two-component RS. When either component fails, instantaneous, minimal, and costly CM is performed on the component. At equally-spaced, discrete points during the system's useful life, the decision-maker has the option to perform instantaneous, imperfect, and costly PM on one or both of the components, to instantaneously replace one or both of the components, or to do nothing. We use a heuristic in an attempt to find a cost-optimal set of PM and replacement decisions.

2. A Model of Limiting Availability under Imperfect Maintenance

Two popular imperfect CM models are based on the concept of "virtual" age as introduced by Kijima et al. (1988). Consider a RS that, at any point in time, is in one of two states, functioning or failed (under CM); and assume that the unit is initially (at time $t = 0$) functioning. Let X_n denote the duration of the period between the $(n-1)^{th}$ CM action completion and the n^{th} failure; and let V_n denote the virtual age of the RS at the time of the CM action completion. Kijima's first model of virtual age is

$$V_n = V_{n-1} + aX_n \tag{2.1}$$

where a is some constant such that $0 \leq a \leq 1$, and $V_0 = 0$. The RS accumulates age during each period of function, i.e. $X_1, X_2, \ldots$. After each failure, CM removes some of the age accumulated during the most recent interval of function. Thus, $1 - a$ captures the degree of RS restoration achieved through CM. Note that perfect CM ($a = 0$) and minimal CM ($a = 1$) are both special cases of this virtual age model.

Let $F_1(x)$ denote the cumulative distribution function of X_1, i.e. the life distribution of a new RS. Let $F_n(x|y)$ denote the conditional cumulative distribution function of X_n given that $V_{n-1} = y$. Then

$$F_n(x|y) = \frac{F_1(x+y)-F_1(y)}{1-F_1(y)} \tag{2.2}$$

Thus, the length of an interval of RS function depends on the virtual age of the equipment at the beginning of the interval. Kijima et al. (1988) use an economic model to evaluate periodic replacement policies for such RS assuming that both CM and replacement are instantaneous activities.

In a second paper, Kijima (1989) expands the study of virtual age in several ways. First, he generalizes (2.1) by modeling virtual age as

$$V_n = V_{n-1} + A_n X_n \tag{2.3}$$

where $\{A_1, A_2,. \dots\}$ is a sequence of independent random variables each distributed over the real interval [0,1]. Second, he presents a second virtual age model

$$V_n = A_n(V_{n-1} + X_n) \tag{2.4}$$

Under this second model, each repair removes a portion of the current virtual age of the RS. For both models of virtual age, he analyzes the behavior of the random variable

$$S_n = \sum_{k=1}^{n} X_k \tag{2.5}$$

for the purpose of studying the same periodic replacement problem as studied by Kijima et al. (1988). Note that we refer to (2.3) the Type I Kijima CM model and (2.4) as the Type II Kijima CM model.

2.1 Additional Studies of Virtual Age

Several other studies have added to the body of knowledge on virtual age. Uematsu and Nishida (1987) use a non-homogenous Poisson process to determine interval reliability and develop cost-optimal replacement models. They use a general repair model (which includes the two Kijima models as special cases) where each interval of RS function is subject to the influence of all previous failure history.

Dagpunar (1997, 1998) suggests an upgraded repair model where minimal repair is performed until a unit of equipment exceeds a specified virtual age. Thereafter, repairs restore the unit to a specified virtual age. He obtains approximations for steady-state measures of RS aging and uptime between maintenance actions.

Scarsini and Shaked (2000) model the total benefit (monetary value) of an item using Kijima's second model. Love et al. (2000) develop a discrete, semi-Markov structure to determine optimal maintenance policies under Kijima's first model. Mettas and Zhao (2005) develop a likelihood-based approach for estimating the parameter of Kijima's second model.

Cassady et al. (2005) use simulation to develop an approximate, analytic expression for RS availability under a special case of Kijima's first model. They use this model to develop near-cost-optimal replacement policies for the RS. Then, they develop and validate meta-models that can be used to convert the RS reliability and maintainability parameters into the parameters of the approximate availability function without the simulation effort.

Doyen and Gaudoin (2006) propose new generalized virtual age models that generalize Kijima's virtual age models to the case in which both CM and PM are used. A generalized virtual age model is defined by both a sequence of effective ages which characterizes the effects of both types of maintenance according to a classical virtual age model, and a usual competing risks model which characterizes the dependency between the two types of maintenance.

Bartholomew-Biggs et al. (2009) consider the problem of scheduling imperfect PM for a RS. The impact of PM is modeled using the Kijima virtual age models.

2.2 A Summary of Cassady et al. (2005)

Consider a RS that is required to operate on a continuous basis for a useful life of L hours. Suppose that at any point in time, the RS is in one of two states, functioning or failed (under CM), and that the RS is initially functioning. Let $Y(t)$ denote the status of the RS at time t; let $Y(t) = 1$ indicate that the RS is functioning at time t, and let $Y(t) = 0$ indicate that the RS is failed at time t.

Suppose that X_1 is a Weibull random variable having shape parameter $\beta > 1$ and scale parameter η. Thus,

$$F_1(x) = 1 - \exp\left(-\left(\frac{x}{\eta}\right)^\beta\right) \tag{2.6}$$

is the cumulative distribution function of X_1. Suppose that X_n has a residual (conditional) Weibull probability distribution (with the same parameters) given survival to age $V_{n-1} = y$ where the accumulation of virtual age is governed by (2.1). Thus,

$$F_n(x|y) = 1 - \exp\left(-\left(\frac{x+y}{\eta}\right)^\beta + \left(\frac{y}{\eta}\right)^\beta\right) \tag{2.7}$$

is the cumulative distribution function of X_n. Finally, suppose that the time required to complete CM is a constant value of t_r, and PM is not performed on the RS.

Let $A(t)$ denote the availability function for the RS where

$$A(t) = \Pr(Y(t) = 1) \tag{2.8}$$

Due to the uncertainty in the number of failures occurring by time t, and the fact that the probability distribution of X_n is dependent upon $\{X_1, X_2, \dots, X_{n-1}\}$, the derivation of a closed form expression for the availability function is not possible. Therefore, Cassady *et al*. (2005) use a discrete-event simulation model to estimate the availability function. The simulation model mimics the function, failure, and CM of the RS using two events: failure, and CM completion. The model collects availability data on the RS at equally-spaced discrete observation points during the RS useful life.

Based on the simulation output, Cassady et al. (2005) propose an approximate, closed-form availability function. Specifically, they use

$$A(t) = \exp(-b_0 t^{b_1}) \tag{2.9}$$

to approximate the availability function. Note that b_0 and b_1 are estimated from the simulation output. They use a factorial design over β, α, and t_r to demonstrate that (2.9) provides a reasonable approximation of RS availability.

Given the approximate model of equipment availability, Cassady *et al*. (2005) determine a near-cost-optimal replacement interval for the RS. Let τ denote the replacement interval for the RS. Note that the replacement action is instantaneous. The average cost per unit time of RS operation, $AvgCost(\tau)$, is a function of τ, and can be defined using two cost parameters: RS acquisition cost (c_a) and the cost per unit time of RS downtime (c_d).

$$AvgCost(\tau) = \left(\frac{c_a}{\tau}\right) + c_d\left(1 - A_{avg}(\tau)\right) \tag{2.10}$$

Note that $A_{avg}(\tau)$ is the average availability over the first time τ units of equipment operation, and note that

$$A_{avg}(\tau) = \frac{1}{\tau}\int_0^\tau A(t)dt \approx \frac{1}{\tau}\int_0^\tau \exp(-b_0 t^{b_1})dt \tag{2.11}$$

Note that the integral in (2.11) must be evaluated numerically.

Finally, in order to eliminate the need to perform simulation to obtain the parameters of the availability model, Cassady et al. (2005) use additional experiments to develop meta-models to convert the RS reliability and maintainability parameters directly into the coefficients of the availability model without requiring the simulation effort. The replacement policy obtained from analysis of the meta-model is compared to the policy obtained directly from the simulation output. The average increase in cost resulting from the sub-optimal replacement policy is only 0.10%. Therefore, the meta-models are robust and provide good estimates of the parameters of the proposed availability function.

The work of Cassady et al. (2005) serves as a starting point for this research. We take a similar approach to studying availability and PM planning for a RS subject to the Type II Kijima CM model.

2.3 RS Definition

We consider the RS studied by Cassady et al. (2005) except that the accumulation of virtual age is governed by

$$V_n = a(V_{n-1} + X_n) \tag{2.12}$$

where a is some constant such that $0 \leq a \leq 1$. In addition, our focus is on the RS limiting availability

$$A = \lim_{t \to \infty} A(t) \tag{2.13}$$

It is not possible to derive a closed-form expression for limiting availability. Therefore, like Cassady et al. (2005), our initial objective is to construct a discrete-event simulation model that can be used to estimate limiting availability.

We constructed a simulation model of the RS cyclical process of function, failure, and CM. The model collects data on RS availability at 50,000 equally-spaced observation points during the RS useful life. During each replication of the simulation model, RS status (functioning or failed) is recorded at each observation point. Since $A(t)$ is the probability that the RS is functioning at time t, availability is estimated at each observation point by dividing the number of replications during which the RS was functioning at the observation point by the number of replications. Since availability is a proportion, we use 153,664 replications to provide 95% confidence intervals on each availability estimate with a worst-case half width of 0.0025.

2.4 Initial Experimentation

As a first step in modeling the system limiting availability function, a simple set of experiments was used to generate sample plots of the estimated system availability function. Eight experiments were formulated with respect to the three parameters: β, a, and t_r. The details of the experimental design for these experiments are summarized in Table 2.1. The values found in Table 2.1 were chosen to provide reasonable coverage of reliability and maintainability parameters found in many repairable mechanical systems. Note that all eight experiments utilize $\eta = 1$, and $L = 200$. Figures 2.1−2.8 represent the behavior observed across these experiments. The availability function achieves a steady-state value greater than zero, rather than degrading over time as in the case considered by Cassady *et al*. (2005).

Table 2.1. Experimental Design

Experiment	β	a	t_r
1	1.5	0.4	0.05
2	1.5	0.4	0.15
3	1.5	0.8	0.05
4	1.5	0.8	0.15
5	3	0.4	0.05
6	3	0.4	0.15
7	3	0.8	0.05
8	3	0.8	0.15

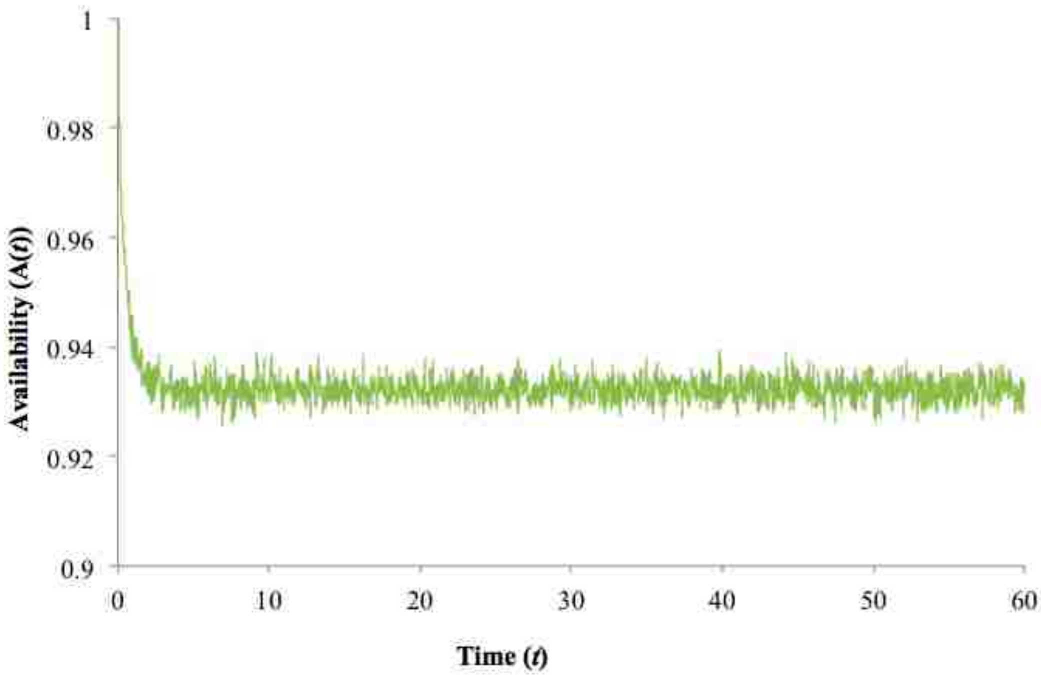

Figure 2.1. Experiment 1 Availability Plot

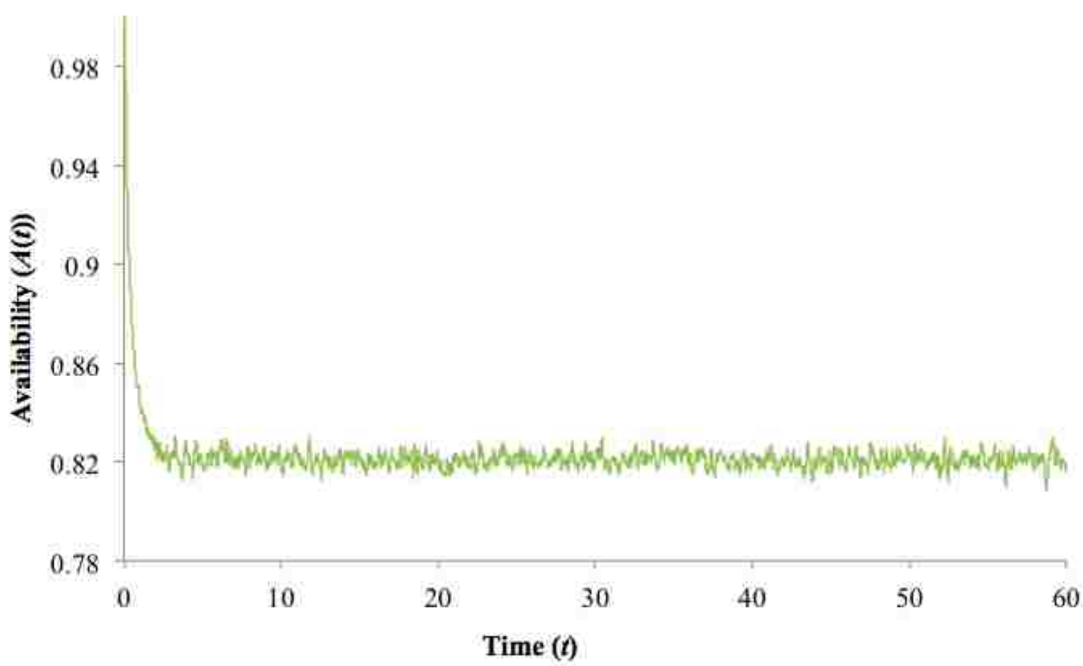

Figure 2.2. Experiment 2 Availability Plot

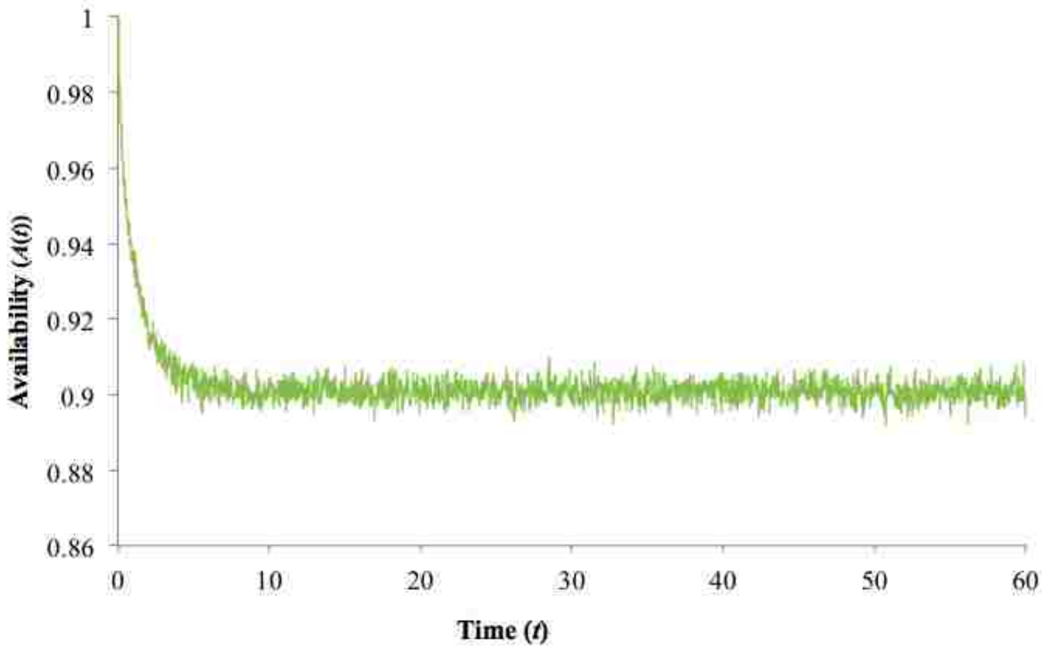

Figure 2.3. Experiment 3 Availability Plot

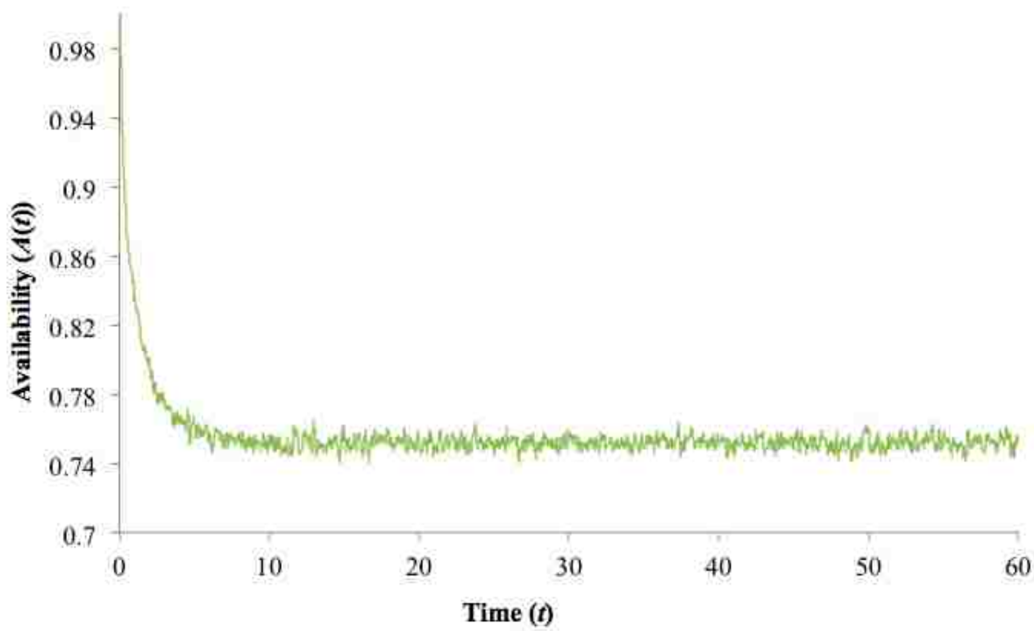

Figure 2.4. Experiment 4 Availability Plot

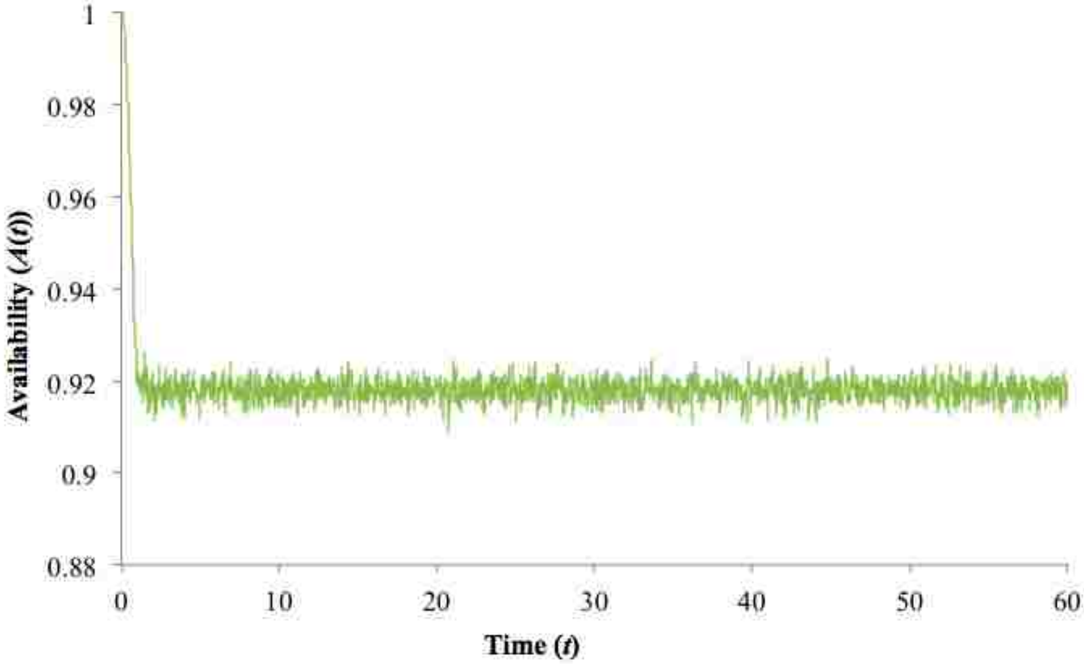

Figure 2.5. Experiment 5 Availability Plot

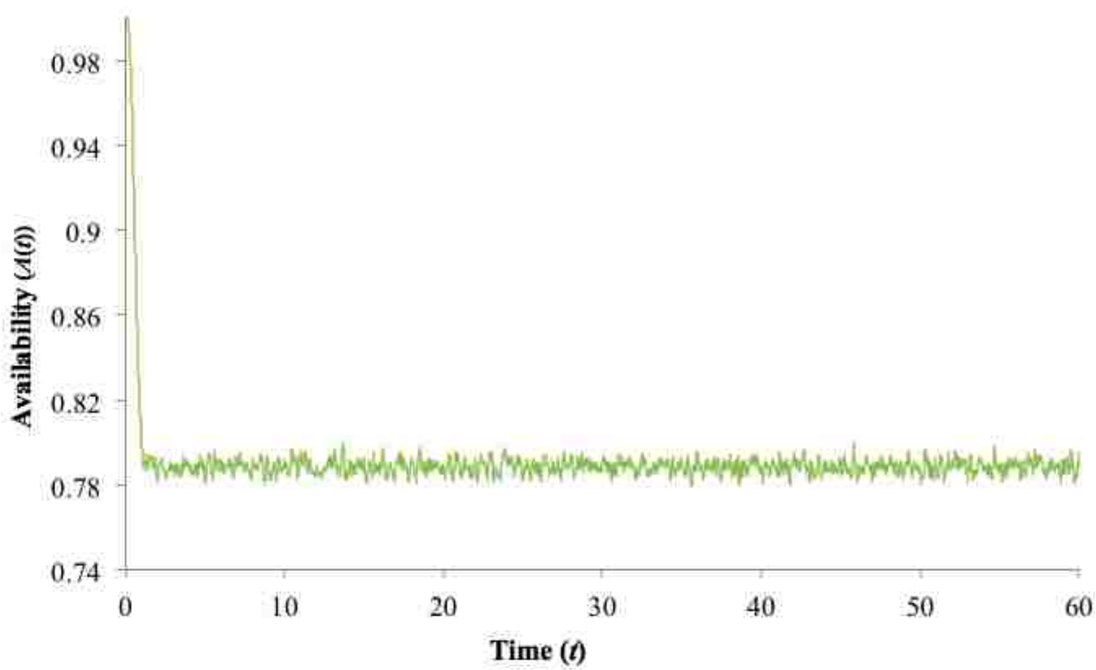

Figure 2.6. Experiment 6 Availability Plot

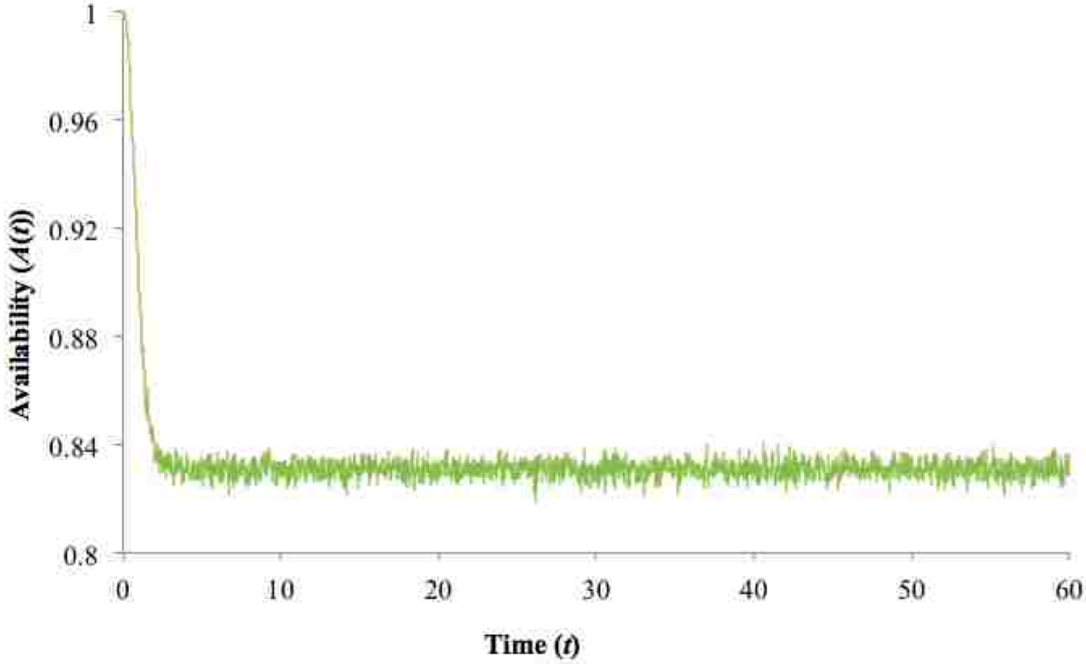

Figure 2.7. Experiment 7 Availability Plot

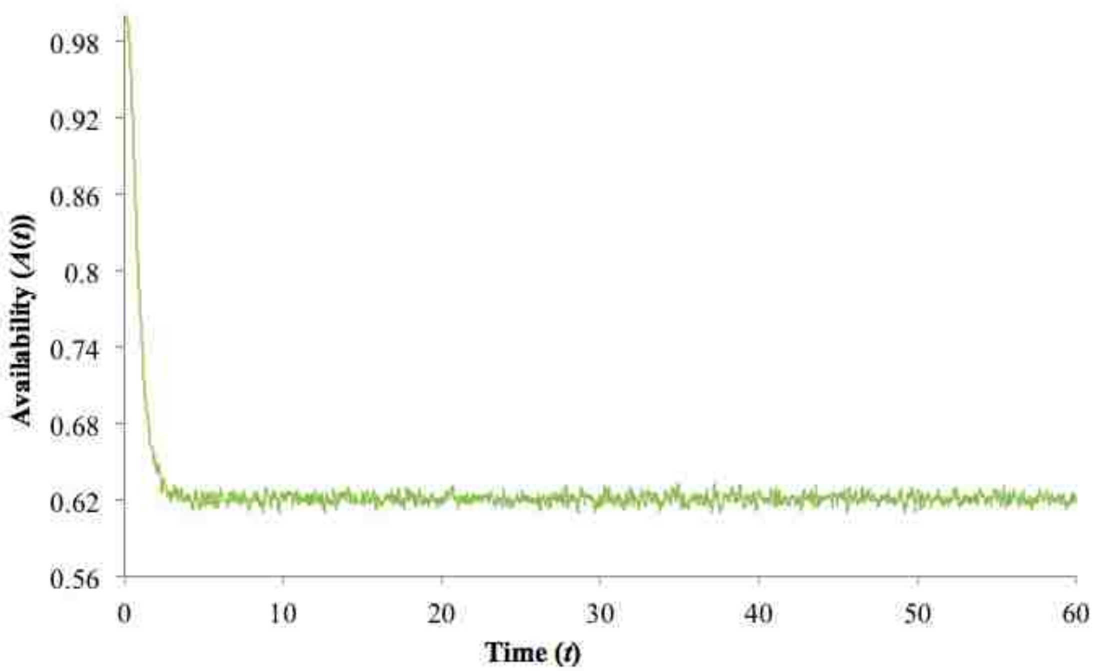

Figure 2.8. Experiment 8 Availability Plot

Consider the RS corresponding to Experiment 1 in Table 2.1, such that $\beta = 1.5$, $a = 0.4$, $t_r = 0.05$, $\eta = 1$, and $L = 200$. Figure 2.1 captures the availability estimates resulting from simulating the RS. The system limiting availability A is the steady-state value of the system availability. The value of A is estimated from the simulation output by computing the average of the availability estimates beyond the initial transient period. The Marginal Standard Error Rule, MSER, by White (1997) was used to determine the truncation point (d), i.e. the point before which the data suggests steady-state has not been achieved. Truncation removes the first $d \ll n$ observations from the average of the availability estimates. The MSER selects a truncation point that minimizes the width of the marginal confidence interval about the truncated sample mean.

For the given experiment, the identified truncation point is $d = 1.86$. The data up to this point were removed and the system limiting availability was estimated by finding the truncated mean ($A = 0.9322$). Figure 2.9 adds the estimate of limiting availability to the truncated data from Figure 2.1. The estimated values of A for the eight experiments are given in Table 2.2.

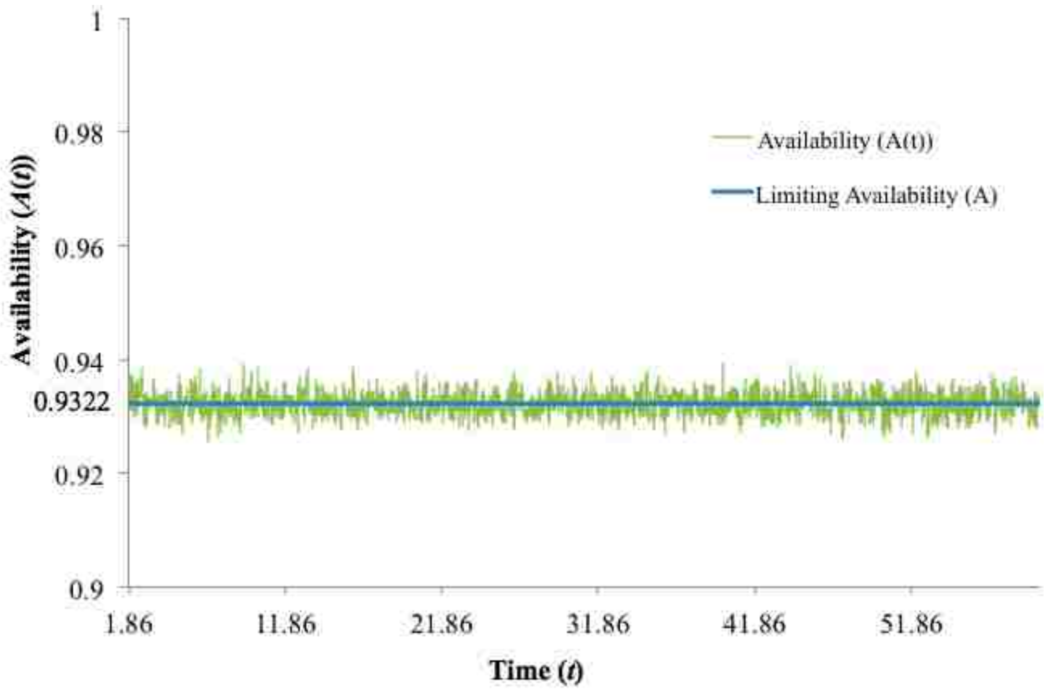

Figure 2.9. Availability vs. Limiting Availability for Experiment 1

Table 2.2. Limiting Availability Estimates from Simulation

Experiment	A
1	0.9322
2	0.8210
3	0.9009
4	0.7518
5	0.9179
6	0.7885
7	0.8309
8	0.6209

2.5 Meta-Modeling of the Availability Function Parameters

The analysis in the previous section provides reasonable approximations to the limiting availability behavior of RS possessing the RAM properties defined in section 2.3. However, each time the value of β, a, or t_r is changed a new simulation experiment must be conducted.

Therefore, a worthwhile next step is to construct an accurate and robust meta-model that converts the system RAM parameters (β, a, t_r) into the system limiting availability without the simulation effort.

We begin by expanding the initial experimental design into a circumscribed central composite (CCC) experimental design to examine the relationship between β, a, and t_r; and A. The CCC design requires five levels of each factor, which were chosen to capture a wide range of system performance under the general system definition. The specific factor levels we used are enumerated in Table 2.3.

Table 2.3. CCC Experimental Design: Factor Settings

	Coded Value			Actual Value		
Experiment	β	a	t_r	β	a	t_r
1	−1	−1	−1	1.5	0.4	0.05
2	−1	−1	1	1.5	0.4	0.15
3	−1	1	−1	1.5	0.8	0.05
4	−1	1	1	1.5	0.8	0.15
5	1	−1	−1	3	0.4	0.05
6	1	−1	1	3	0.4	0.15
7	1	1	−1	3	0.8	0.05
8	1	1	1	3	0.8	0.15
9	0	0	0	2.25	0.6	0.1
10	0	0	−3/2	2.25	0.6	0.025
11	0	0	3/2	2.25	0.6	0.175
12	0	−3/2	0	2.25	0.3	0.1
13	0	3/2	0	2.25	0.90	0.1
14	−3/2	0	0	1.125	0.6	0.1
15	3/2	0	0	3.375	0.6	0.1

For each experiment, the simulation model was executed using 50,000 observations, and 153,664 replications. The estimates of A found from the simulation output can be found in Table 2.5. Analysis of variance is then used to develop a meta-model of A in terms of β, a, and t_r. This involves applying linear regression to the A values obtained from simulation (Table 2.5) for estimating the parameters in equation (2.14).

$$A = g_0 + g_1\beta + g_2 a + g_3 t_r + g_{11}\beta^2 + g_{12}\beta a + g_{13}\beta t_r + g_{22}a^2 + g_{23}a t_r + g_{33}t_r^2 \tag{2.14}$$

The resulting parameters estimates are given in Table 2.4 (note that all the coefficients are statistically significant at a level of significance of 0.05), and the corresponding estimates of A are provided in Table 2.5. The mean absolute error of the 15 estimates is 0.0098.

Table 2.4. Meta-Model Parameter Estimates

g_0	0.8965
g_1	0.0633
g_2	0.7696
g_3	−0.8400
g_{11}	0.0278
g_{12}	−0.1284
g_{13}	−0.2636
g_{22}	−0.4975
g_{23}	−1.4812
g_{33}	4.0742

To test the robustness of our meta-models, we conducted experiments with randomly selected values of β, a, and t_r within the CCC experimental design. We used a pseudorandom number generator to create fifty such experiments. The data from the first 10 of these experiments are shown in Table 2.6. We executed the simulation model for each experiment. Then, we: (1) used the simulation output to estimate A, (2) used the meta-model to estimate A, and (3) compared the meta-model to the simulation output by computing the MAE. The results from the first 10 experiments are shown in Table 2.7. The MAE across the 50 experiments is

0.004. This provides further evidence that the meta-model provides a reasonable approximation of A.

Table 2.5. Limiting Availability Estimates

Experiment	Simulation A	Meta-Model A	Absolute Error
1	0.9322	0.9360	0.0038
2	0.8210	0.8347	0.0136
3	0.9009	0.8983	0.0025
4	0.7518	0.7378	0.0140
5	0.9179	0.9378	0.0198
6	0.7885	0.7969	0.0084
7	0.8309	0.8231	0.0078
8	0.6209	0.6229	0.0020
9	0.8173	0.8170	0.0002
10	0.9470	0.9530	0.0059
11	0.7188	0.7269	0.0081
12	0.8673	0.8516	0.0156
13	0.6631	0.6929	0.0297
14	0.8919	0.8957	0.0038
15	0.7986	0.8109	0.0123

Table 2.6. Randomly Selected Parameters Values

Experiment	β	a	t_r
1	2.2425	0.6429	0.0991
2	2.0318	0.4393	0.1445
3	2.2042	0.4467	0.1418
4	2.8164	0.6837	0.0972
5	1.5908	0.5581	0.1052
6	2.9636	0.6423	0.0579
7	1.8007	0.5093	0.1435
8	1.9000	0.5475	0.1418
9	1.8449	0.5813	0.1371
10	2.4942	0.5494	0.0861

Table 2.7. Limiting Availability Estimates for the Random Parameters

Experiment	Simulation A	Meta-Model A	Absolute Error
1	0.8080	0.8065	0.0016
2	0.7992	0.8059	0.0067
3	0.7968	0.8008	0.0041
4	0.7819	0.7780	0.0039
5	0.8462	0.8544	0.0082
6	0.8662	0.8699	0.0037
7	0.7964	0.8075	0.0111
8	0.7866	0.7951	0.0085
9	0.7880	0.7954	0.0074
10	0.8439	0.8447	0.0008

2.6 Preventive Maintenance Analysis

Since the availability of RS under a Kijima's Type II repair model will quickly approach a steady state value, it may be worthwhile to use PM on the system to improve the steady-state behavior. Age-based PM is often used to improve the availability for RS that achieve steady-state. In this section, we study the impact of age-based PM on RS performance under a Kijima's Type II model. Our goal is to identify an optimal age-based PM policy that maximizes the system's steady-state availability. An age-based PM policy τ implies that PM is performed if the RS functions without failure for a period of τ time units.

Suppose that PM restores RS to "as good as new" condition. Note that PM should be worthwhile if it is cheaper and/or faster than repair. Therefore, we assume the duration of PM, t_{PM}, to be constant such that $t_{PM} = \theta t_r$, where $\theta < 1$. This implies that $t_{PM} < t_r$, which makes PM a viable choice in order to avoid the longer repair associated with diagnosing and repairing a system failure in the field.

As an initial step in PM analysis, we use some of the previously implemented experiments and the simulation model to study the effects associated with age-based PM on the system limiting availability to demonstrate if age-based PM can, in fact, improve A. To study the effects associated with age-based PM, we use the two experiments that yield the lowest and highest steady state availability values in Table 2.5. Experiments 8 and 10 yield the lowest and highest steady state availability values respectively for the fifteen CCC experiments. We construct a set of experiments by varying the values of τ. The range of τ for each experiment was determined based on the 5th and 95th percentile of the underlying Weibull distribution of each experiment. The values of τ, then, were determined by dividing the range in equal intervals. Tables 2.8 and 2.9 define the experimental design used to study the age-based PM on experiment 8 and 10.

Table 2.8. PM Experimental Design for Experiment 8

Experiment	τ
1	0.37
2	0.584
3	0.798
4	1.012
5	1.226
6	1.44

Table 2.9. PM Experimental Design for Experiment 10

Experiment	τ
1	0.27
2	0.54
3	0.81
4	1.08
5	1.35
6	1.625

Figure 2.10 illustrates the effect τ has on the steady state availability for experiment 8 with $\theta = 0.2$. We can see that PM improves the steady state availability significantly for small values of τ. As τ approaches the upper range of the specified interval, we see that the steady state value approaches the steady state value assuming no PM. Although small values of τ improve the steady state value of availability, it does take longer to achieve steady state for the smaller values of τ due to the additional PM actions. For this instance, we see that significant improvements in steady state availability can be made when we use PM at the cost of having lower initial availability until steady state is achieved.

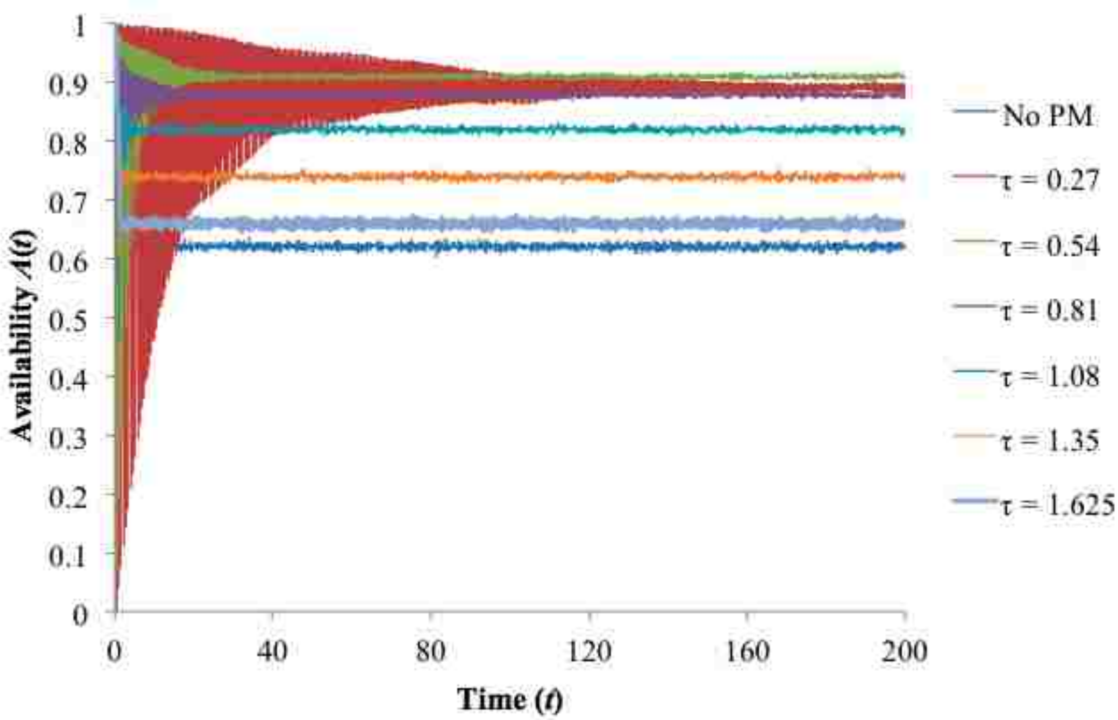

Figure 2.10. Availability Plot for Experiment 8 with $\theta = 0.2$

To examine how the steady state value of the availability changes as a function of θ, we found the limiting availability values for each τ when $\theta = 0.75$ and compared them to limiting availability found when $\theta = 0.2$ (see Table 2.10). The results show that for smaller values of θ we get larger improvements in the steady state value. Figure 2.11 shows how the steady state value improves for Experiment 8 when $\tau = 0.37$ as PM is prescribed and how the improvement is larger for the smaller value of θ.

Table 2.10. Experiment 8 Limiting Availability as a Function of θ

τ	$\theta = 0.2$	$\theta = 0.75$
0.37	0.9081	0.7561
0.584	0.9075	0.8071
0.798	0.8812	0.8131
1.012	0.8375	0.7926
1.226	0.7785	0.7523
1.44	0.7107	0.6983

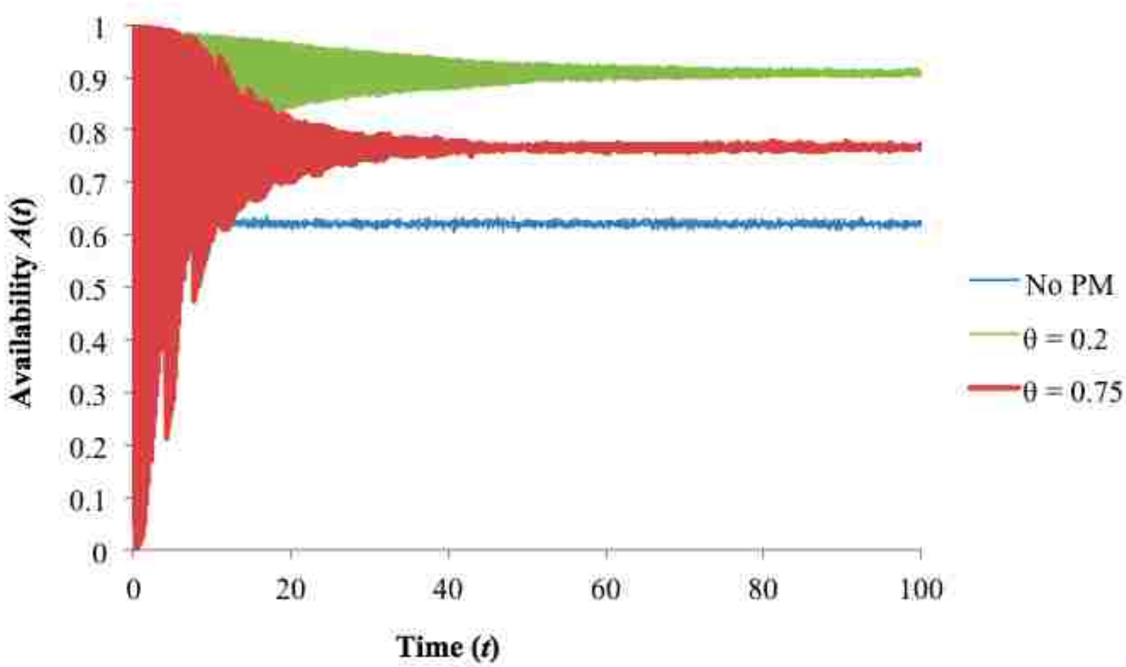

Figure 2.11. Availability Plot for Experiment 8 for $\tau = 0.37$

Figures 2.12 and 2.13 illustrate consistent performance for experiment 10, although the magnitude of improvement is significantly lower because of the already high steady state value associated with this parameter set. The low magnitude of improvement in the steady state values can be also seen in Table 2.11 where the limiting availability with $\theta = 0.75$ and $\theta = 0.2$ were found.

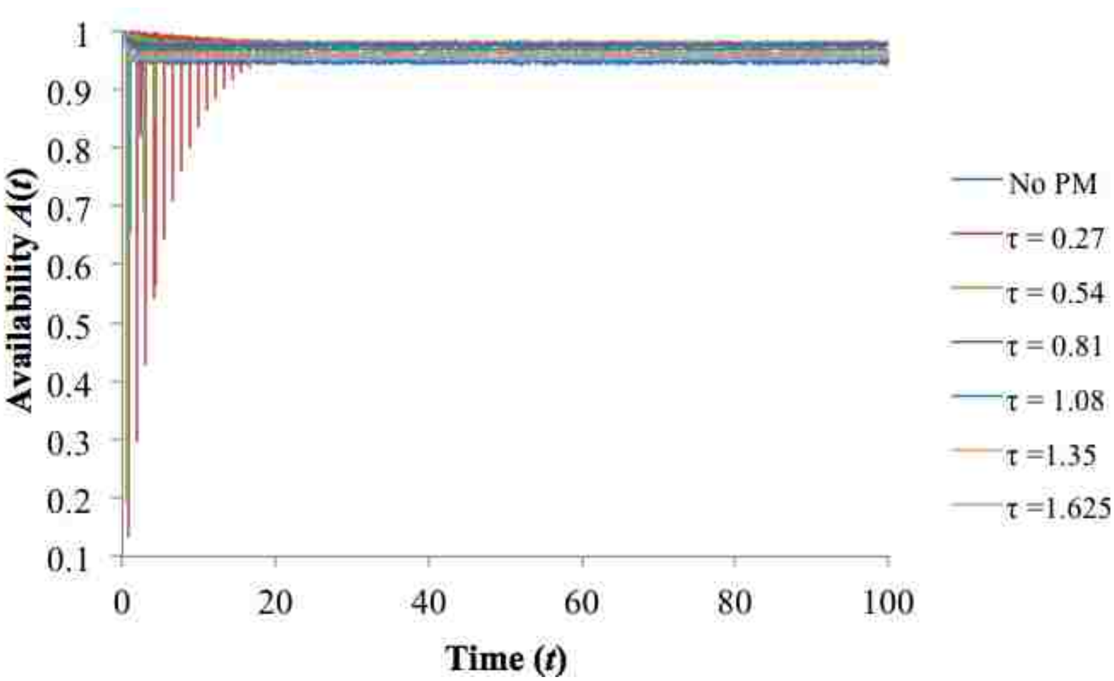

Figure 2.12. Availability Plot for Experiment 10 with $\theta = 0.2$

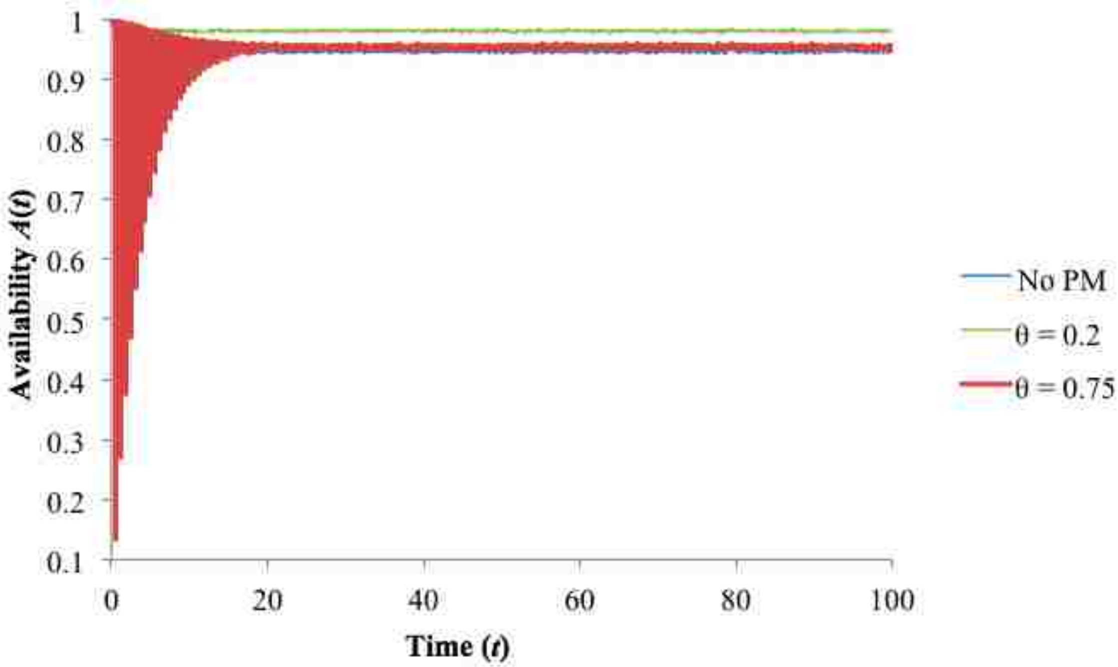

Figure 2.13. Availability Plot for Experiment 10 for $\tau = 0.54$

Table 2.11. Experiment 10 Limiting Availability Based on θ

τ	$\theta = 0.2$	$\theta = 0.75$
0.27	0.9783	0.9316
0.54	0.9800	0.9578
0.81	0.9756	0.9624
1.08	0.9687	0.9609
1.35	0.9610	0.9568
1.625	0.9540	0.9522

Thus, our results indicate that when using a Kijima Type II model for a system repair process, PM can improve the steady state availability value. However, it takes longer to reach steady state when a PM policy is used and the instantaneous availability is significantly lower early on when using a PM policy.

Given the previous results that show that age-based PM, in fact, can improve the steady state availability value, it may be worthwhile to identify an optimal age-based PM policy that maximizes the system's steady-state availability. To find the optimal PM policy, we constructed a second meta-model that includes the PM parameter (τ) as one of the inputs and used that model to optimize PM without the simulation effort. We expanded the previous meta-model by adding τ and t_{PM} into the circumscribed central composite (CCC) experimental design to examine the relationship between β, a, t_r, t_{PM}, and τ; and A. The specific factor levels we used for the five levels of each factor required for the CCC design are enumerated in Table 2.12.

Table 2.12. CCC Experimental Design: Factor Settings

	Coded value					Actual value				
Experiment	β	a	t_r	t_{PM}	τ	β	a	t_r	t_{PM}	τ
1	−1	−1	−1	−1	1	1.5	0.4	0.05	0.01	1.44
2	1	−1	−1	−1	−1	3	0.4	0.05	0.01	0.37
3	−1	1	−1	−1	−1	1.5	0.8	0.05	0.01	0.37
4	1	1	−1	−1	1	3	0.8	0.05	0.01	1.44
5	−1	−1	1	−1	−1	1.5	0.4	0.15	0.01	0.37
6	1	−1	1	−1	1	3	0.4	0.15	0.01	1.44
7	−1	1	1	−1	1	1.5	0.8	0.15	0.01	1.44
8	1	1	1	−1	−1	3	0.8	0.15	0.01	0.37
9	−1	−1	−1	1	−1	1.5	0.4	0.05	0.03	0.37
10	1	−1	−1	1	1	3	0.4	0.05	0.03	1.44
11	−1	1	−1	1	1	1.5	0.8	0.05	0.03	1.44
12	1	1	−1	1	−1	3	0.8	0.05	0.03	0.37
13	−1	−1	1	1	1	1.5	0.4	0.15	0.03	1.44
14	1	−1	1	1	−1	3	0.4	0.15	0.03	0.37
15	−1	1	1	1	−1	1.5	0.8	0.15	0.03	0.37
16	1	1	1	1	1	3	0.8	0.15	0.03	1.44
17	−1.5	0	0	0	0	1.125	0.6	0.1	0.02	0.905
18	1.5	0	0	0	0	3.375	0.6	0.1	0.02	0.905
19	0	−1.5	0	0	0	2.25	0.3	0.1	0.02	0.905
20	0	1.5	0	0	0	2.25	0.9	0.1	0.02	0.905
21	0	0	−1.5	0	0	2.25	0.6	0.025	0.02	0.905
22	0	0	1.5	0	0	2.25	0.6	0.175	0.02	0.905
23	0	0	0	−1.5	0	2.25	0.6	0.1	0.035	0.905
24	0	0	0	1.5	0	2.25	0.6	0.1	0.005	0.905
25	0	0	0	0	−1.5	2.25	0.6	0.1	0.02	0.1025
26	0	0	0	0	1.5	2.25	0.6	0.1	0.02	1.7075
27	0	0	0	0	0	2.25	0.6	0.1	0.02	0.905

For each experiment, the simulation model was executed using 50,000 observations, and 153,664 replications. The values of A estimated from the simulation output. Then, analysis of variance is used to develop a meta-model of A in terms of β, a, t_r, t_{PM}, and τ. This involves applying linear regression to the A values obtained from simulation (Table 2.14) for estimating the parameters in equation 2.15.

$$A = h_0 + h_1\beta + h_2 a + h_3 t_r + h_4 t_{PM} + h_5\tau + h_{11}\beta^2 + h_{12}\beta a + h_{13}\beta t_r + h_{14}\beta t_{PM} + h_{15}\beta\tau + h_{22}a^2 + h_{23}at_r + h_{24}at_{PM} + h_{25}a\tau + h_{33}{t_r}^2 + h_{34}t_r t_{PM} + h_{35}t_r\tau + h_{44}{t_{PM}}^2 + h_{45}t_{PM}\tau + h_{55}\tau^2 \quad (2.15)$$

The resulting parameters estimates are given in Table 2.13, and the corresponding estimates of A are provided in Table 2.14. Note that all the coefficients are statistically significant at a level of significance of 0.05. The mean absolute error of the 27 estimates is 0.0090.

Table 2.13. Meta-Model Parameter Estimates

h_0	0.7058
h_1	0.0545
h_2	0.1035
h_3	–0.0087
h_4	1.6589
h_5	0.3606
h_{11}	0.0082
h_{12}	–0.0493
h_{13}	–0.0398
h_{14}	–0.4221
h_{15}	–0.0595
h_{22}	0.1462
h_{23}	–0.1881
h_{24}	–5.5899
h_{25}	–0.0871
h_{33}	2.4375
h_{34}	–14.1596
h_{35}	–0.8227
h_{44}	45.9857
h_{45}	1.3461
h_{55}	–0.0902

Table 2.14. Limiting Availability Estimates

Experiment	Simulation A	Meta-Model A	Absolute Error
1	0.9413	0.9423	0.0009
2	0.9677	0.9595	0.0082
3	0.9462	0.9367	0.0096
4	0.8806	0.8886	0.0080
5	0.8959	0.8834	0.0124
6	0.8031	0.8083	0.0052
7	0.8384	0.8422	0.0038
8	0.9544	0.9491	0.0054
9	0.9050	0.8960	0.0090
10	0.9223	0.9310	0.0087
11	0.9271	0.9343	0.0073
12	0.9196	0.9177	0.0019
13	0.8424	0.8468	0.0044
14	0.9093	0.9046	0.0048
15	0.8543	0.8482	0.0061
16	0.7108	0.7222	0.0115
17	0.8951	0.9079	0.0128
18	0.9084	0.8987	0.0097
19	0.9094	0.9184	0.0091
20	0.8995	0.8935	0.0060
21	0.9614	0.9629	0.0015
22	0.8486	0.8501	0.0016
23	0.8912	0.8836	0.0077
24	0.9120	0.9228	0.0108
25	0.8329	0.8702	0.0374
26	0.8334	0.7991	0.0343
27	0.9014	0.8928	0.0086

To test the robustness of our meta-models, we conducted experiments with randomly selected values of β, a, t_r, t_{PM}, and τ within the CCC experimental design. We used a pseudorandom number generator to create fifty such experiments. The data from the first 10 of these experiments are shown in Table 2.15. We executed the simulation model for each experiment. Then, we: (1) used the simulation output to estimate A, (2) used the meta-model to estimate A, and (3) compared the meta-model to the simulation output by computing the MAE. The results from the first 10 experiments are shown in Table 2.16. The average MAE across the 50 experiments is 0.013. This provides evidence that the meta-model provides a reasonable approximation of A.

Table 2.15. Randomly Selected Parameters Values

Experiment	β	a	t_r	t_{PM}	τ
1	2.816	0.684	0.097	0.019	0.431
2	2.964	0.642	0.058	0.012	0.914
3	1.945	0.692	0.111	0.022	1.379
4	2.766	0.73	0.081	0.016	0.981
5	2.621	0.748	0.127	0.025	1.080
6	2.739	0.585	0.104	0.021	0.648
7	2.852	0.548	0.093	0.019	0.491
8	2.729	0.558	0.144	0.029	0.610
9	2.982	0.567	0.052	0.010	0.576
10	2.706	0.591	0.133	0.027	0.440

Table 2.16. Limiting Availability Estimates for The Random Parameters

	Simulation	Meta-Model	
Experiment	A	A	Absolute Error
1	0.9386	0.9124	0.0262
2	0.9418	0.9409	0.0010
3	0.8527	0.8426	0.0101
4	0.9101	0.9051	0.0051
5	0.8495	0.8356	0.0139
6	0.9264	0.9024	0.0240
7	0.9421	0.9182	0.0240
8	0.9042	0.8732	0.0310
9	0.9663	0.9603	0.0060
10	0.9164	0.8819	0.0344

The optimal PM time (τ^*) can be found using the meta-model of A by taking the first derivative of the function in 2.15 with respect to τ and then setting the derivative equal to zero and solving it. Note that the second derivative of the function in 2.15 with respect to τ is always negative which implies that any maximum found is a global result. Consider the system corresponding to experiment 1 in Table 2.15. For this example, the meta-model recommended (optimal) PM time is 0.438, and the corresponding limiting availability (based on the simulation model) is 0.939. Figure 2.14 shows a plot of the limiting availability (based on the simulation model) as a function of τ for the given experiment. We can see that the limiting availability based on the meta-model recommended PM time is close to the simulation optimal value of the limiting availability.

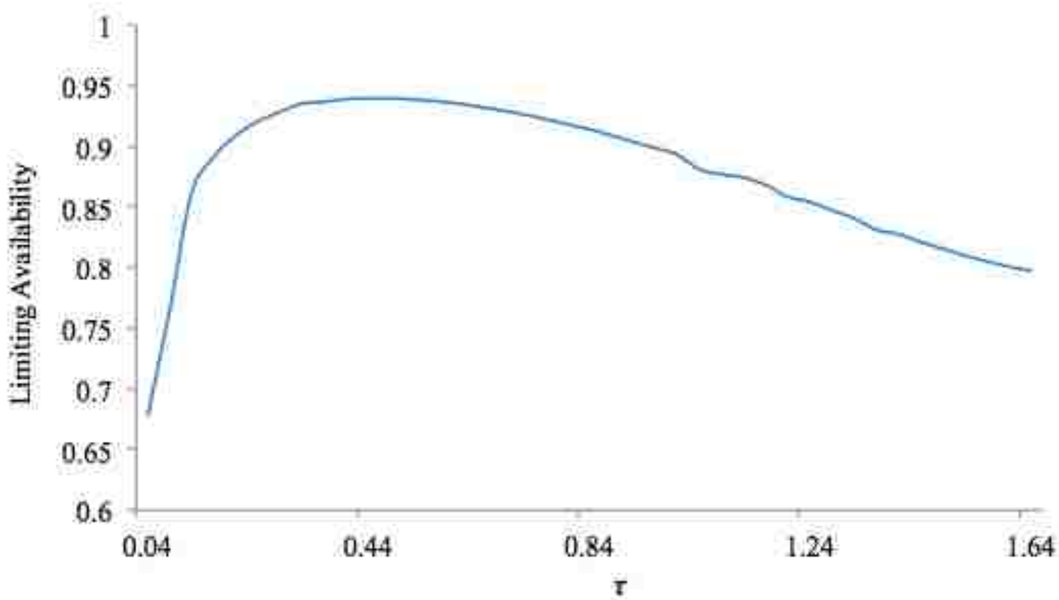

Figure 2.14. Limiting Availability vs τ

Suppose we assume that the simulation values of A are the true values, but the meta-model values are used to determine the optimal PM time (τ^*). We can determine the loss in availability associated with this approximation by comparing the limiting availability value of the optimal PM time based on the meta-model to the limiting availability value of the optimal PM time based on the simulation model. Let τ^*denote the optimal PM time based on the simulation model. Simulation-based optimization can be used to find τ^*. We used Golden section search to find τ^*in this research. let $\hat{\tau}^*$ denote the optimal PM time based on the meta-model, let A_1 denote the limiting availability resulted from using a PM policy of τ^*, and let A_2 denote the limiting availability resulted from using a PM policy of $\hat{\tau}^*$. We can determine the average loss in availability associated with this approximation over the random 50 experiments in Table 2.15. Note that the simulation model is used to compute A_1 and A_2. The loss in availability for each experiment is given by

$$\frac{A_1 - A_2}{A_1} \times 100\% \tag{2.16}$$

The results from the first 10 experiments are given in Table 2.17. The average availability loss associated with all 50 experiments is 0.16%. This indicates that the meta-model presented in 2.15 is robust, and provides good estimates of the availability to determine the optimal PM times for the system.

Table 2.17. System Availability Loss

Experiment	τ^*	A_1	$\hat{\tau}^*$	A_2	Loss
1	0.4489	0.9397	0.4388	0.9396	0.01%
2	0.4723	0.9633	0.5358	0.9627	0.06%
3	0.4689	0.9067	0.6805	0.9023	0.48%
4	0.4689	0.9481	0.4836	0.9479	0.02%
5	0.4544	0.9179	0.3802	0.9163	0.18%
6	0.4578	0.9335	0.4949	0.9333	0.02%
7	0.5212	0.9412	0.5107	0.9411	0.02%
8	0.4505	0.9085	0.3860	0.9082	0.04%
9	0.4689	0.9677	0.5802	0.9662	0.15%
10	0.4578	0.9164	0.4134	0.9158	0.07%

2.7 Conclusion

This paper studies the long-run availability of a traditional RS that is subjected to imperfect CM. Kijima's second virtual age model is used to describe the imperfect repair process. Because of the complexity of the underlying probability models, we cannot derive a closed-form expression for the availability of the RS. Therefore, we use simulation modeling to estimate availability performance and meta-modeling to convert the reliability and maintainability parameters of the RS into an availability estimate without the simulation effort. The system limiting availability is estimated from the simulation output by computing the average of the availability estimates beyond the initial transient period. Using a circumscribed central composite experimental design, we confirm the accuracy of the meta-model using the 15 experiments and 50 random experiments within the design space. The mean absolute error between the simulation output and the meta-model output is 0.0098 for the 15 experiments, and 0.004 for the 50 random experiments. This indicates that the meta-model provides a reasonable approximation of the system limiting availability.

As a final step, we add age-based, perfect PM to our analysis. Our goal is to identify an optimal age-based PM policy that maximizes the system's steady-state availability. To find the optimal PM policy, we construct a second meta-model that includes the PM interval as one of the inputs and use that model to optimize PM without the simulation effort. Using a circumscribed central composite experimental design, we confirm the accuracy of the meta-model using the 27 experiments and 50 random experiments within the design space. For the new 50 experiments, we compare the PM policy obtained from analysis of the meta-model to the policy obtained directly from the simulation output. The average availability loss associated with all 50 experiments is 0.16%. Therefore, we conclude that the meta-model is robust, and provide good estimate of the limiting availability to determine the optimal PM times for the system.

3. Preventive Maintenance and Replacement Scheduling for a Two–Component System

Maintenance and replacement planning for single-component repairable systems has been studied extensively in the literature, and such systems can be found in practice. Chaudhuri and Sahu (1977) are among the first to consider imperfect maintenance in planning PM activities for a deteriorating system. Many extensions have been made to this work including those of Chan and Downs (1978), Malik (1979), and Nakagawa (1983). Malik (1979) proposes a model for finding successive maintenance points using the concept of an "improvement factor".

Jayabalan and Chaudhuri (1992) present a model where a variable improvement factor is utilized. They present a two-phase algorithm for cost optimization of maintenance scheduling. The first phase yields optimal time intervals between PM events. The second phase involves the calculation of the total cost of both maintenance and replacement to determine the optimal time of replacement.

Usher *et al*. (1998) present an optimal maintenance and replacement model for a single component system. They determine an optimal PM schedule for a system subject to deterioration by considering the cost of the rate of occurrence of failure over time, and the use of an improvement factor for the case of imperfect maintenance. Additionally, they provide a comparison of computational results among random search, genetic algorithm, and branch-and-bound algorithms.

Tsai *et al*. (2001) consider two activities, imperfect maintenance and replacement in their maintenance optimization model. Imperfect maintenance activities are modeled based on the concept of improvement factor. They use a genetic algorithm to determine the cost-optimal PM activities.

Although, multi-component repairable systems are more common, mathematical models for maintenance and replacement planning for multi-component repairable systems are rare due to their increased complexity and difficulty to solve. Levitin and Lisnianski (2000) present an optimization model to determine PM actions for a multi-state multi-component system. Their model aims to minimize cost subject to required level of reliability. They apply a universal generating function technique to evaluate multi-state system reliability and use a genetic algorithm to solve the model. Shalaby *et al*. (2004) develop an optimization model for PM scheduling of multi-component multi-state system. They define the sequence of PM activities as the decision variables and the summation of PM, minimal repair, and downtime costs as the objective functions. They use combined genetic algorithm and simulation approach to optimize the model.

The research of Usher *et al*. (1998) serves as the motivation and starting point for this research. Their research focuses on the formulation, solution and analysis of a model for planning PM and replacement activities for a single-component repairable system subject to an increasing rate of occurrence of failure. We extend their research and present a model for planning PM and replacement activities for a two-component repairable system. We present an analysis of a two-component repairable system to better understand multi-component systems and gain insights into the related complexities.

3.1 Model Derivation

Consider a repairable system that is comprised of two components connected in series. The system is to be operated over a fixed interval of time that can be subdivided into a discrete number of equal-length periods. The system is subject to deterioration with age, and this deterioration is modeled by an increasing rate of occurrence of failure (ROCOF). During each period, system failures caused by failure of one of the components may occur and, if they occur, are rectified instantaneously with minimal repair. At the end of each period in the future (except for the last period), one of the following three options is selected and executed instantaneously on each of the components.

1. Do nothing – No action is planned for the end of the period, i.e., the component remains in an "as bad as old" condition at the beginning of the next period.
2. Replacement – The component is replaced at the end of the period, immediately placing in an "as good as new" condition, i.e., its effective age is returned to zero at the beginning of the next period.
3. Preventive maintenance – The component is maintained at the end of the period and returned to a condition somewhere between "as good as new" and "as bad as old". At the beginning of the next period, the component's effective age is reduced by a stated percentage of its age at the end of the period.

Note that replacement and PM can only be performed at these discrete points in time.

3.1.1 Modeling System Maintenance and Aging

Let $[0, t]$ denote the time interval (i.e., maintenance planning horizon) of interest. This interval is segmented into n discrete intervals, each of length t/n. The maintenance decisions are represented using two sets of binary variables. Let

$$p_{ij} = \begin{cases} 1 & \text{if PM is performed on component } i \text{ at the end of period } j \\ 0 & \text{otherwise} \end{cases}, \tag{3.1}$$

and let

$$r_{ij} = \begin{cases} 1 & \text{if replacement of component } i \text{ occurs at the end of period } j \\ 0 & \text{otherwise} \end{cases}, \tag{3.2}$$

$i = 1, 2$, $j = 1, 2, \dots, n-1$. These variables correspond to the decision variables in the optimization model. To prevent the initiation of both PM and replacement in the same period, the following constraint is defined:

$$p_{ij} + r_{ij} \leq 1, \tag{3.3}$$

$i = 1, 2, j = 1, 2, \dots, n-1$.

To account for the instantaneous changes in effective system age that result from system PM or replacement, the following notation is introduced. Let $x_{i,j}$ denote the effective age of component i at the start of period j, and let $y_{i,j}$ denote the age of component i at the end of period j, $j = 1, 2, \dots, n$. Since the system is initially new, $x_{i,1} = 0$, $i = 1, 2$, and because repair is minimal,

$$y_{i,j} = x_{i,j} + \frac{t}{n}, \tag{3.4}$$

$i = 1, 2$, $j = 1, 2, \dots, n$. Consider some component $i \in \{1, 2\}$ and some period $j \in \{1, 2, \dots, n-1\}$. If no action is taken at the end of period j, then $x_{i,j+1} = y_{i,j}$. If the system is replaced at the end of period j, then $x_{i,j+1} = 0$. If PM is performed at the end of period j, then

$$x_{i,j+1} = \alpha_i y_{i,j} \tag{3.5}$$

where α_i is a constant such that $0 < \alpha_i < 1$.

The maintenance decisions and their relationship to the effective age of the system can be summarized using the following equations that serve as functional constraints in the optimization model:

$$x_{i,1} = 0 \qquad i = 1, 2, \tag{3.6}$$

$$y_{i,j} = x_{i,j} + \frac{t}{n} \qquad i = 1, 2, j = 1, 2, \dots, n, \tag{3.7}$$

$$x_{i,j+1} = y_{i,j} - (1 - \alpha_i) y_{i,j} p_{i,j} - y_{i,j} r_{i,j} \qquad i = 1, 2, j = 1, 2, \dots, n. \tag{3.8}$$

3.1.2 Modeling System Maintenance Costs

The objective in the optimization model is to minimize γ, the expected value of the repair, replacement and preventive maintenance costs incurred over the planning horizon. Note that

$$\gamma = \sum_{j=1}^{n} c_j \tag{3.9}$$

where c_j denotes the expected value of the repair, replacement and preventive maintenance costs incurred during period j (including any actions taken at the end of the period), $j = 1, 2, \dots, n$. Furthermore, note that

$$c_j = f_j + m_j \tag{3.10}$$

where f_j denotes the expected value of the cost of failures occurring during period j and m_j denotes the cost of any end-of-period maintenance in period j, $j = 1, 2, \dots, n$. Note that $m_n = 0$.

Let ϕ_i denote the cost of failure of component i, $i = 1, 2$. Let $\lambda_i(a)$, where a denotes the effective age of the component $(a \geq 0)$, denote the mathematical function that captures the increasing ROCOF of component i, $i = 1, 2$. The widely-recognized power law process (Weibull process) is used to model the ROCOF. Therefore,

$$\lambda_i(a) = \frac{\beta_i}{{\eta_i}^{\beta_i}} a^{\beta_i - 1}, \tag{3.11}$$

$i = 1, 2$. Recall that, in this case, both components have an increasing ROCOF, so $\beta_i > 1$, $i = 1, 2$. Since repair is minimal, the non-homogeneous Poisson process governs component failures during each period, and the number of component i failures in period j is a Poisson random variable having mean

$$v_{i,j} = \int_{x_{i,j}}^{y_{i,j}} \lambda_i(a)\, da = \left(\frac{y_{i,j}}{\eta_i}\right)^{\beta_i} - \left(\frac{x_{i,j}}{\eta_i}\right)^{\beta_i}, \tag{3.12}$$

$i = 1, 2, j = 1, 2, \dots, n$, and

$$f_j = \phi_1 v_{1j} + \phi_2 v_{2j}, \tag{3.13}$$

$j = 1, 2, \dots, n$.

The question that comes to mind here is why not to apply Usher *et al*. (1998)'s one-component model independently to the two components and find the optimal maintenance policy. It is because, when planning PM strategies, considerations of the overall benefit for the whole system should supersede the optimum plan for each component separately. Since a RS is almost always comprised of many components that have different maintenance needs, optimizing maintenance planning at the component level is likely to be suboptimal at the system level. For example, sometimes it is less expensive and more convenient to perform PM on a component in a system when performing a repair action on another component in the system rather than at the optimum time for performing the PM for that component. Therefore, we need system-level maintenance strategies for performing component-level maintenance.

Often, components that comprise a system are not independent. This dependence can be either structural or economic. Structural dependence may manifest itself in terms of common-cause failures or maintenance-induced damage. Economic dependence suggests that it is more economical to repair several components together rather than repairing them separately. This is also referred to as opportunistic maintenance. When opportunistic maintenance is performed only a minimal variable cost is added to repair other components but a lot of other fixed cost is saved.

Opportunism can be explained with a simple example. Consider the two-components system described in section 2. Either due to failure or expiration of a PM interval, maintenance is about to be performed on component 1. If component 2 is near the expiration of its PM interval, then it may be worthwhile to go ahead and perform PM on component 2. Such an action is an opportunistic maintenance action. Since performing any replacement or maintenance on the system usually consumes at a fixed cost, performing opportunistic maintenance saves a lot of that fixed cost. Let δ denote the fixed cost of performing any replacements or maintenance. Let π_i denote the incremental cost of performing PM on component i, and let ρ_i denote the incremental cost of replacing component i, $i = 1, 2$. Then,

$$m_j = \pi_{1j}p_{1j} + \pi_{2j}p_{2j} + \rho_{1j}r_{1j} + \rho_{2j}r_{2j} + \delta q_j, \tag{3.14}$$

where

$$q_j = \min\left(1, \sum_{i=1}^{2}\left(p_{ij} + r_{ij}\right)\right), \tag{3.15}$$

$j = 1, 2, \dots, n-1$.

For a two-component system when $n = 2$, there are 9 feasible options for maintenance at the end of the first period corresponding to all combinations of performing maintenance or replacement on each of the two components. Table 2.1 shows those options and their corresponding maintenance total costs for both cases when considering the two components independently and the one two-component system. Note that all the incremental and fixed costs of performing any replacement or maintenance in the table are set to be 1. The results show that considering a single two-component system costs less than considering two one-component systems, which makes the model introduced in this chapter more reasonable.

Table 3.1. Maintenance Options and Their Costs for $n = 2$

Maintenance options				**Comp. 1**	**Comp. 2**	**2-Comp. System**	**Two 1-Comp. System**	**2-Comp. System**
p_{11}	r_{11}	p_{21}	r_{21}	q_1	q_1	q_1	m_1	m_1
0	0	0	0	0	0	0	0	0
0	0	0	1	0	1	1	1	1
1	0	0	0	1	0	1	1	1
0	1	0	0	1	0	1	1	1
0	0	1	0	0	1	1	1	1
0	0	0	1	0	1	1	1	1
0	1	0	1	1	1	1	2	1
1	0	1	0	1	1	1	2	1
0	1	1	0	1	1	1	2	1

The optimal maintenance policy can be obtained by solving the following optimization model:

$$\text{Min} \quad \gamma = \sum_{j=1}^{n} c_j = \sum_{j=1}^{n} \phi_1 \left(\left(\frac{y_{1,j}}{\eta_1}\right)^{\beta_1} - \left(\frac{x_{1,j}}{\eta_1}\right)^{\beta_1} \right) + \phi_2 \left(\left(\frac{y_{2,j}}{\eta_2}\right)^{\beta_2} - \left(\frac{x_{2,j}}{\eta_2}\right)^{\beta_2} \right) + \pi_{1j}p_{1j} + \pi_{2j}p_{2j} + \rho_{1j}r_{1j} + \rho_{2j}r_{2j} + \delta q_j \quad (3.16)$$

s.t

$$x_{i,1} = 0 \qquad i = 1, 2, \quad (3.17)$$

$$y_{i,j} = x_{i,j} + \frac{t}{n} \qquad i = 1, 2, j = 1, 2, \dots, n, \quad (3.18)$$

$$x_{i,j+1} = y_{i,j} - (1 - \alpha_i) y_{i,j} p_{i,j} - y_{i,j} r_{i,j} \qquad i = 1, 2, j = 1, 2, \dots, n, \quad (3.19)$$

$$p_{ij} = 0 \text{ or } 1 \qquad i = 1, 2, j = 1, 2, \dots, n, \quad (3.20)$$

$$r_{ij} = 0 \text{ or } 1 \qquad i = 1, 2, j = 1, 2, \dots, n, \quad (3.21)$$

$$p_{ij} + r_{ij} \leq 1 \qquad i = 1, 2, j = 1, 2, \dots, n, \quad (3.22)$$

where

$$q_j = \min\left(1, \sum_{i=1}^{2} (p_{ij} + r_{ij})\right) \qquad j = 1, 2, \dots, n. \quad (3.23)$$

The optimization model, denoted as problem **P**, reveals a binary programming problem with a nonlinear objective function and linear constraints. The nonlinearity is introduced by the polynomial term in the objective function, and the minimum term in the term q_j. The minimization of the polynomial function subject to linear constraints was shown to be NP-hard by Parrilo and Sturmfels (2003).

Note that problem **P** has $\left(3^{n-1}\right)^2$ feasible solutions corresponding to all combinations of doing nothing, performing PM, or performing replacement on each of the two components at the end of each of the first $n-1$ periods.

3.2 Numerical Experimentation

To demonstrate the implementation of the model and reinforce some of the underlying concepts, a small example is utilized. Consider a system of two components having repair and maintenance (RAM) characteristics such that $\alpha_1 = 0.4$, $\alpha_2 = 0.25$, $\beta_1 = 1.5$, $\beta_2 = 2$, $\eta_1 = 2$, $\eta_2 = 3$, $\phi_1 = \$10$, $\phi_2 = \$15$, $\pi_1 = \$1.5$, $\pi_2 = \$2.5$, $\rho_1 = \$3$, $\rho_2 = \$5$, and $\delta = \$1$. The system is required to operate over a planning horizon of length $t = 12$.

3.2.1 An Enumerative Solution Approach

As a first solution approach, a Visual Basic (VB) application that enumerates all feasible solutions for problem **P** was developed. The application evaluates the total expected cost for each solution and identifies the optimal sequence of actions for each component.

When $n = 4$, there are $3^{2(4-1)} = 729$ feasible solutions. Using total enumeration, the optimal solution (Table 3.1) is identified and results in a total expected cost of \$160.48. If n is increased to 8 (same $t = 12$), the optimal solution (Table 3.2) results in a total expected cost of \$142.46. The enumerative approach was used to solve the defined example for all $n =$ 2, 3, 4, 5, 6, 7, 8, where $n =$ 8 is the highest number of periods we could go with a reasonable computation time of 280 minutes. The results of enumeration showed that when the number of periods is small, replacement is performed more, but as the number of periods increases fewer replacements are needed.

Table 3.2. Optimal Solution for $n = 4$

	Action	
Period	**Component 1**	**Component 2**
1	Replacement	Replacement
2	Replacement	Replacement
3	Replacement	Replacement

Table 3.3. Optimal solution for $n = 8$

	Action	
Period	**Component 1**	**Component 2**
1	Replacement	Maintenance
2	Replacement	Replacement
3	Replacement	Maintenance
4	Replacement	Replacement
5	Replacement	Maintenance
6	Replacement	Replacement
7	Replacement	Maintenance

Enumeration of all feasible solutions is not practical for large instances of **P**. For the defined example, a reasonable limit on computation time is exceeded for $n > 8$. The relationship between run time and n is exponential (see Figure 3.1). Note that the run times portrayed in Figure 3.1 are based on the use of a personal computer having a 2.0 GHz Intel Core2Duo processor and 3GB of RAM.

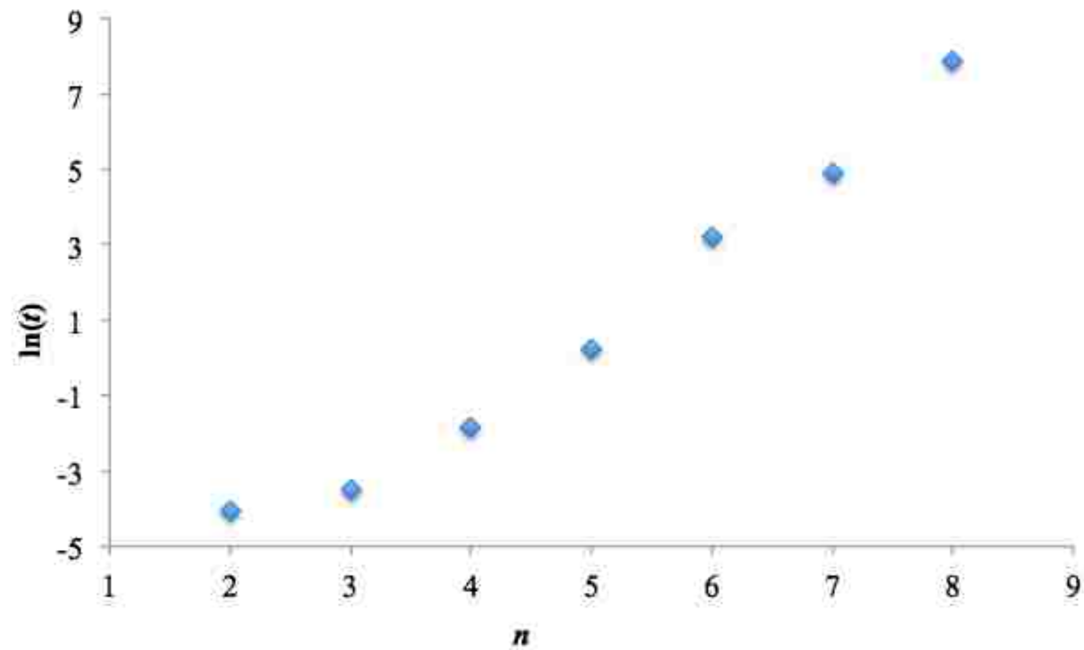

Figure 3.1 Relationship Between Problem Size and Run Time in Enumeration

3.2.2 A Heuristic Solution Approach: Genetic Algorithm

Compared to enumerative approaches, heuristic approaches to solving combinatorial optimization problems usually require shorter run times at the price of reduced solution quality. One commonly-used heuristic approach for these types of problems is the use of genetic algorithms (GA). GA have the advantage of searching extremely large solution spaces for better solutions in a relatively short time and using those found solutions in generating new solutions. In this section, a GA for solving problem **P** is developed and evaluated.

Genetic algorithms (GA) are motivated by the theory of evolution, i.e., “survival of the fittest” (Holland, 1975). GA have been designed as general search strategies and optimization methods working on populations of feasible solutions. Working with populations allows for the identification and exploration of properties which good solutions have in common (Goldberg, 1989).

In GA, individual solutions to an optimization problem are represented by a chromosome of genes. For problem **P**, each chromosome (maintenance plan) consists of $2n$ genes. Each gene represents a planning period action (0, 1 or 2) for one of the components, where 0 denotes do nothing, 1 denotes a PM action and 2 denotes a replacement. For example, if a chromosome for $n = 3$ is 210000, then the planned actions in period 1 are a replacement on component 1 and PM on component 2, and the planned actions in periods 2 and 3 are do nothing on both components. Each chromosome is evaluated by computing its fitness value. In **P**, the solution's fitness is the total expected cost (γ).

The GA begins by randomly creating an initial population of 1000 chromosomes. Each gene in each chromosome is randomly selected from the set $\{0, 1, 2\}$. The first generation of solutions begins with the 1000 randomly-created solutions. The GA uses two simple operators to create subsequent populations. These operators are crossover and mutation. In this paper, 95% (950) of the remaining solutions are created by the crossover operator. In order to apply this operator, two parents are randomly selected from the existing 1000 solutions. Then, the crossover operator is applied by randomly choosing a position in the parent solutions and exchanging the tail (the genes after the chosen position) of the first solution with the tail of the second solution (see Figure 3.2). The remaining 5% (50) of the solutions are generated using mutation as shown in Figure 3.3. In mutation, a gene is randomly selected in a randomly-selected solution (from the original 1000). Then, that gene is replaced by randomly selecting one of the other two feasible values.

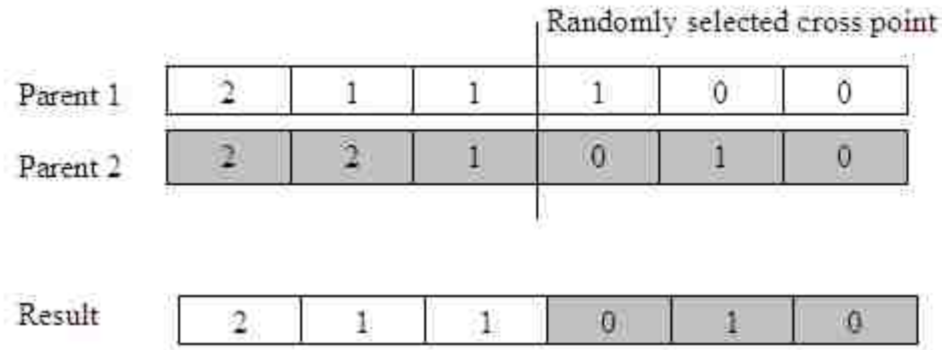

Figure 3.2 Crossover Process

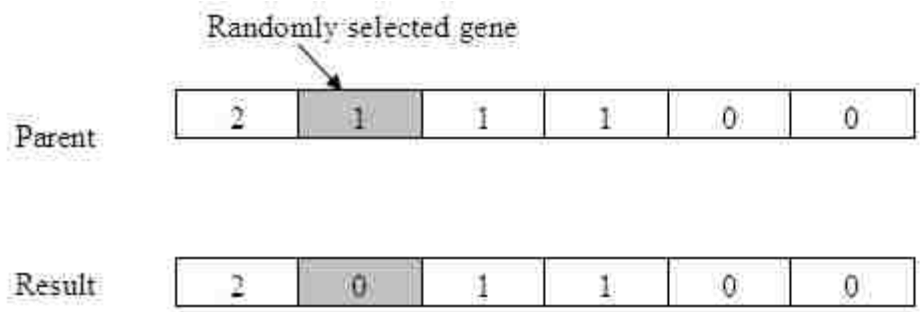

Figure 3.3 Mutation Process

After applying both crossover and mutation, the fitness values of the 2000 solutions are computed. The 1000 (50%) solutions with the lowest fitness (highest cost) are deleted. The remaining 1000 solutions serve as 50% of the next generation. This process continues for 1000 generations, and the best solution in the final generation is the recommended solution.

In an effort to validate the GA application, which is constructed in VB, the GA results are compared to the enumerative approach results for the defined example and all $n = 2, 3, 4, 5, 6, 7, 8$. In all cases, the GA recommends the optimal solution. The execution time for the GA when $n = 8$ was less than a minute, based on the use of a personal computer having a 2.0 GHz Intel Core2Duo processor and 3GB of RAM.

Since the GA appears to be effective for small problems, larger problems are considered. For the defined example and $n = 20$, the GA recommends the solution in Table 3.3, which results in a total expected cost of $142.84. The time it took the GA to solve this problem was 2 minutes. For the same example assuming $n = 52$, the GA recommends the solution in Table 3.4 which results in a total expected cost of $152.90. The GA was able to provide this solution in 5 minutes. Figure 3.4 illustrates the "path" taken by GA for this example. Note that after only 80 generations, the GA converges to its recommended plan. The relationship between run time and n in GA is shown in Figure 3.5.

Table 3.4 GA Solution for $n = 20$

	Action	
Period	**Component 1**	**Component 2**
1	-	-
2	Replacement	Maintenance
3	-	-
4	Replacement	Maintenance
5	-	-
6	Replacement	Maintenance
7	-	-
8	Replacement	Maintenance
9	-	-
10	Replacement	Maintenance
11	-	-
12	Replacement	Maintenance
13	-	-
14	Replacement	Maintenance
15	-	-
16	Replacement	Maintenance
17	-	-
18	Replacement	Maintenance
19	-	-

Table 3.5 GA Solution for $n = 52$

	Action	
Period	**Component 1**	**Component 2**
1	-	-
2	-	-
3	-	-
4	-	-
5	-	-
6	Replacement	Maintenance
7	-	-
8	-	-
9	-	-
10	-	-
11	Replacement	Maintenance
12	-	-
13	-	-
14	-	-
15	-	-
16	Replacement	Maintenance
17	-	-
18	Replacement	Maintenance
19	-	-
20	-	-
21	-	-
22	-	-
23	Maintenance	Maintenance
24	-	-
25	-	-
26	-	-

	Action	
Period	**Component 1**	**Component 2**
27	Replacement	Maintenance
28	-	-
29	-	-
30	-	-
31	-	-
32	-	-
33	Replacement	Replacement
34	-	-
35	-	-
36	-	-
37	-	-
38	-	-
39	Replacement	Maintenance
40	-	-
41	-	-
42	-	-
43	-	-
44	-	-
45	Replacement	Maintenance
46	-	-
47	-	-
48	-	-
49	-	-
50	-	-
51	-	-

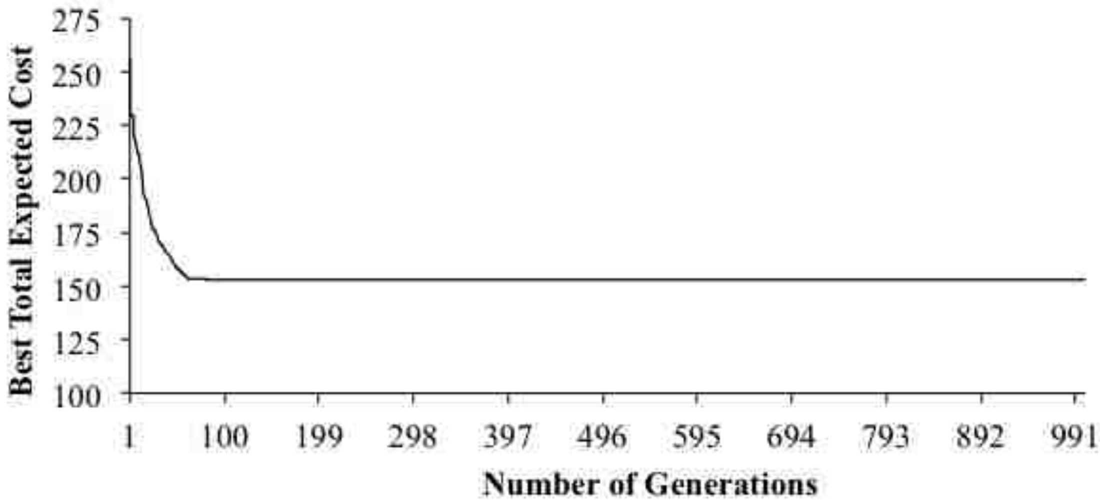

Figure 3.4 Relationship Between Minimum γ and The Number of Generations

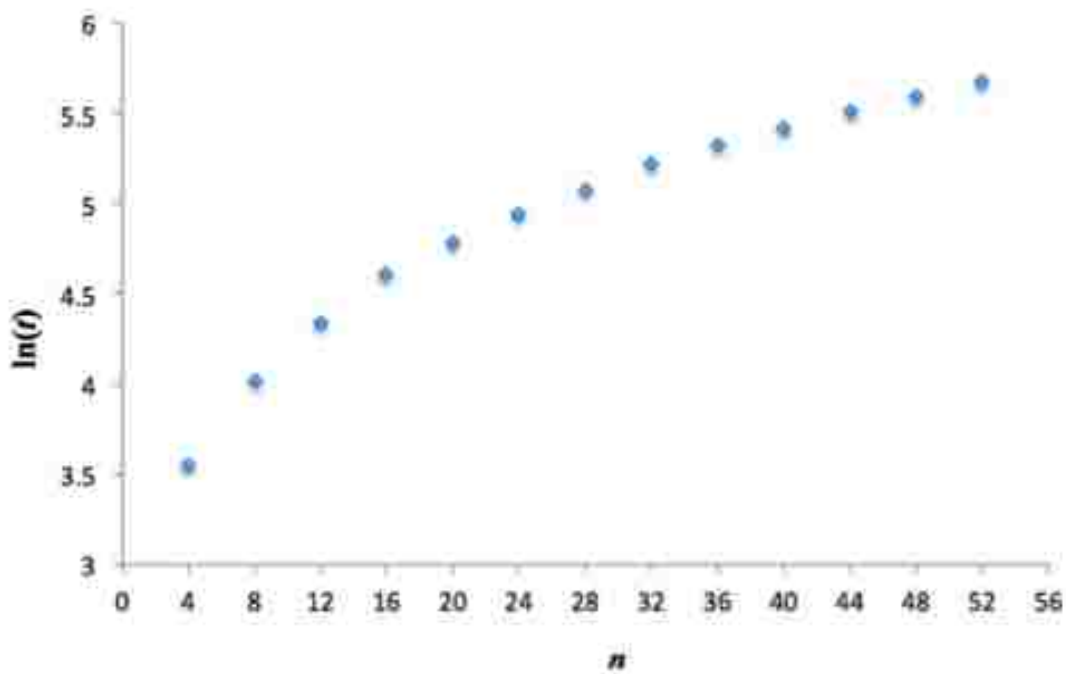

Figure 3.5 Relationship Between Problem Size and Run Time in GA

Since the GA provides no guarantee of optimality, the performance of the GA, using some extreme cases that have obvious or intuitive results, is validated. The baseline example used is the same defined example.

Case 1: $\rho_1 = \rho_2 = \delta = 0$

In this case the optimal solution should be all replacements for both components in each period. If replacement has no cost, then it is always the best action to perform since it places both components in an "as good as new" condition. For this case, the GA was able to provide the optimal solutions for $n = 20$ and $n = 52$.

Case 2: $\rho_1 = \pi_1, \rho_2 = \pi_2$

In this case, the replacement cost is equal to the preventive maintenance cost for each component. Since replacement is more effective than maintenance, the optimal solution for this case should not contain any maintenance action in any period. After setting $\rho_1 = \pi_1 = 1.5$ and $\rho_2 = \pi_2 = 2.5$, and keeping all other parameters the same, the GA recommended the expected solution (all replacements) for $n = 20$ and $n = 52$.

3.3 Model Formulation for *m*–Component System

In this section we extend our previous model to consider a repairable system with m components. A new optimization model for planning the PM and replacements schedules for m–component system is presented.

Similar to the two-component system, suppose there is a RS of m components connected in series. The system is to be operated over a fixed interval of time that can be subdivided into a discrete number of equal-length periods. And each component in the system is subject to deterioration that is modeled by an increasing ROCOF. At the end of each period in the future (except for the last period), one of the three options (do nothing, replacement, or PM) is selected.

Assume the system maintenance, aging, and maintenance costs are modeled in the same manner as for the two-component system. Then, the optimal maintenance policy can be obtained by solving the following optimization model:

$$\text{Min} \quad \gamma = \sum_{j=1}^{n} c_j$$

$$= \sum_{j=1}^{n} \left(\sum_{i=1}^{m} \left[\phi_i \left(\left(\frac{y_{i,j}}{\eta_i} \right)^{\beta_i} - \left(\frac{x_{i,j}}{\eta_i} \right)^{\beta_i} \right) + \pi_{ij} p_{ij} + \rho_{ij} r_{ij} \right] + \delta q_j \right) \tag{3.24}$$

s.t

$$x_{i,1} = 0 \qquad i = 1, 2, \dots, m, \tag{3.25}$$

$$y_{i,j} = x_{i,j} + \frac{t}{n} \qquad i = 1, 2, \dots, m, j = 1, 2, \dots, n, \tag{3.26}$$

$$x_{i,j+1} = y_{i,j} - (1 - \alpha_i) y_{i,j} p_{i,j} - y_{i,j} r_{i,j} \qquad i = 1, 2, \dots, m, j = 1, 2, \dots, n, \tag{3.27}$$

$$p_{ij} = 0 \text{ or } 1 \qquad i = 1, 2, \dots, m, j = 1, 2, \dots, n, \tag{3.28}$$

$$r_{ij} = 0 \text{ or } 1 \qquad i = 1, 2, \dots, m, j = 1, 2, \dots, n, \tag{3.29}$$

$$p_{ij} + r_{ij} \leq 1 \qquad i = 1, 2, \dots, m, j = 1, 2, \dots, n, \tag{3.30}$$

where

$$q_j = \min\left(1, \sum_{i=1}^{m} (p_{ij} + r_{ij})\right) \qquad j = 1, 2, \dots, n. \tag{3.31}$$

The optimization model reveals a NP-hard problem with a nonlinear objective function and linear constraints. Also, the problem has $\left(3^{n-1}\right)^m$ feasible solutions corresponding to all combinations of doing nothing, performing PM, or performing replacement on each of the m components at the end of each of the first $n - 1$ periods. So when $n = 4$ and $m = 3$, there are $3^{3(4-1)} = 19683$ feasible solutions. Note that the problem difficulty grows considerably as the number of components increases, which makes it impossible to solve the problem in a reasonable time using enumeration especially for larger number of periods.

Now if we consider the GA to solve the problem, each chromosome (maintenance plan) will consist of $m \times n$ genes. As the number of components increases, it will take GA longer to find a solution, especially when the number of periods increases. So even when the formulation of the m-component system didn't change significantly from the two-component system, the solution approaches may not work as efficiently due to the increased complexity of the system.

3.4 Conclusion

This paper presents an approach for identifying a cost-optimal maintenance policy for a system comprised of two components connected in series. The system is to be operated over a fixed interval of time that can be subdivided into a discrete number of equal-length periods. The system is subject to deterioration with age, and this deterioration is modeled by an increasing ROCOF. At the end of each period in the future, one of three actions (maintain, replace, or do nothing) is to be executed instantaneously on each of the components such that the total expected costs are minimized.

Two approaches are used to identify a cost-effective preventive maintenance policy: an enumerative approach that guarantees an optimal policy, and a heuristic approach that provides no such guarantees. The enumerative approach is found not to be practical for large size problems because the run time increases exponentially as the number of periods increases. On the other hand, results from using the genetic algorithm appear to indicate that it can be successfully used to find a good solution very quickly. The GA provides the same solutions for small problems as the enumerative approach, and it generates intuitive solutions for the extreme cases of larger problems. The genetic algorithm run time was directly influenced by the problem size, such that large problems need more computation time for a fixed number of generations and population size.

Finally, the model formulation for the m-component system is presented. Although, the formulation of the m-component system didn't change notably from the two-component system, the solution approaches may not work as efficiently due to the increased complexity of the system.

4. Summary of Contributions and Future Work

This dissertation achieves two main contributions. This section summarizes those contributions and future work that if done can enhance this research.

4.1 Contributions

The contributions of this dissertation can be summarized as follows:

1. Study the long-run availability of a traditional RS that is subjected to imperfect corrective maintenance modeled by Kijima's second virtual age model.
    - Use simulation modeling to estimate availability performance due to the difficulty of deriving a closed-form expression for the availability of the RS.
    - Use meta-modeling to convert the reliability and maintainability parameters of the repairable system into an availability estimate without the simulation effort.
    - Add age-based, perfect PM to the analysis to improve system steady-state availability.
2. Optimize a PM policy for a multi-component RS.
    - Use Genetic Algorithm in an attempt to find a cost-optimal set of PM and replacement decisions for a two-component system.
    - Formulate the optimization model for m-component system.

4.2 Future Research Work

Future research could include investigating the transient behavior of the system studied in chapter 2. Although, every practical system in the world has a transient state, even if it is very short, virtually all studies have emphasized on steady state or equilibrium behavior in preference to transient behavior. Our results indicate that using PM on the system improves the steady-state value of availability. However, it takes longer to achieve higher values of the steady state availability, with more fluctuating transient behavior. Therefore, future work could study the tradeoff between higher steady-state availability with longer fluctuated transient behavior, or a faster to stabilize system with lower steady-state availability.

Additionally, one could investigate availability performance when a RS is subject to stochastic values of a, t_r, and t_{PM}.

Finally, future work for the multi-component system in chapter 3 could investigate new solution approaches to solve the m-component RS. Experiments on using GA with an alternative chromosome encoding or other efficient heuristics to solve the problem in reasonable time could be investigated.

References

Bartholomew-Biggs, M., Zuo, M., J., Li, X. 2009. Modelling and Optimizing Sequential Imperfect Preventive Maintenance. *Reliability Engineering and System Safety* **94** 53-62.

Cassady, C., Iyyob, I., Schneider, K., Pohl, E. 2005. A generic model of equipment availability under imperfect maintenance. *IEEE Transactions on Reliability* **54** 564-571.

Chan, P.K.W., Downs, T. 1978. Two Criteria for Preventive Maintenance. *IEEE Transactions on Reliability* **27** 272-273.

Chaudhuri, D., Sahu, K.C. 1977. Preventive Maintenance Intervals for Optimal Reliability of Deteriorating system. *IEEE Transactions on Reliability* **26** 371–372.

Cho, D., Parlar, M. 1991. A survey of maintenance models for multi-unit systems. *European Journal of Operational Research* **51**(1) 1-23.

Dagpunar, J. 1997. Renewal-type equations for a general repair process. *Quality and Reliability Engineering International* **13**(4) 235-245.

Dekker, R. 1996. Applications of maintenance optimization models: A review and analysis. *Reliability Engineering and System Safety* **51**(3) 229-240.

Doyen, I., Gaudoin, O. 2006. Imperfect Maintenance in Generalized Competing Risk Frames. *Journal of Applied Probability* **43**(3) 825-839.

Goldberg, D. 1989. Genetic Algorithms in Search Optimization and Machine Learning. Addison-Wesley Publishing, Reading, MA.

Holland, J. 1975. Adaptation in Natural and Artificial Systems. University of Michigan Press.

Jabayalan, V., Chaudhuri, D. 1992. Optimal maintenance and replacement policy for a deteriorating system with increased mean downtime. *Naval Research Logistics* **39**(1) 67-78.

Kijima, M. 1989. Some results for repairable systems with general repair. *Journal of Applied Probability* **26** 89-102.

Kijima, M., Morimura, H., Suzuki, Y. 1988. Periodic replacement problem without assuming minimal repair. *European Journal of Operational Research* **37**(2) 194-203.

Levitin, G., Lisnianski, A. 2000. Optimization of Imperfect Preventive Maintenance for Multi State Systems. *Reliability Engineering and System Safety* **67** 193-203.

Love, C., Zhang, Z., Zitron, M., Guo, R. 2000. A discrete semi-Markov decision model to determine the optimal repair/replacement policy under general repair. *European Journal of Operational Research* **125** 398-409.

Malik, M. 1979. Reliable Preventive Maintenance Scheduling. *IIE Transactions* **11**(3) 221-228.

McCall, John J. 1965. Maintenance Policies For Stochastically Failing Equipment. *Management Science* **11**(5) 347-358.

Mettas, A., Zhao, W. 2005. Modeling and analysis of repairable systems with general repair. *Proceedings of the Annual Reliability and Maintainability Symposium* 176-182.

Mobley, K. 1988. A Guide to Predictive Maintenance. Oakville **11**(1).

Nakagawa, T. 1983. Mean Time to Failure with Preventive Maintenance. *IEEE Transactions on Reliability* **32** 341.

Parrilo, P., Sturmfels, B. 2003. Minimizing polynomial functions. *Algorithmic and Quantitative Real Algebraic Geometry, DIMACS Series in Discrete Mathematics and Theoretical Computer Science* **60** 83-99.

Pham, H., Wang, H. 1996. Imperfect maintenance. *European Journal of Operational Research* **94** 425-438.

Pierskalla, William P., Voelker, J. 1976. A Survey of Maintenance Models: The Control and Surveillance of Deteriorating Systems. *Naval Research Logistics Quarterly* **23**(3) 353-388.

Scarsini, M. Shaked, M. 2000. On the value of an item subject to general repair or maintenance”, *European Journal of Operational Research* **122** 625-637.

Shalaby, M.A., Gomaa, A.H., Mohib, A.M. 2004. A Genetic Algorithm for Preventive Maintenance Scheduling in a Multiunit Multistate System. *Journal of Engineering and Applied Science* **51**(4) 795-811.

Sherif, Y., Smith, M. 1981. Optimal Maintenance Models for Systems Subject to Failure. *Naval Research Logistics Quarterly* **28**(1) 47-74.

Tsai, Y., Wang,K., Teng, H. 2001. Optimizing Preventive Maintenance for Mechanical Components Using Genetic Algorithms. *Reliability Engineering and System Safety* **74** 89-97.

Uematsu, K., Nishida, T. 1987. The branching nonhomogeneous Poisson process and its application to a replacement model. *Microelectronics Reliability* **27**(4) 685-691.

Usher, J., Kamal, A., Syed, W. 1998. Cost Optimal Preventive Maintenance and Replacement Scheduling. *IIE Transactions* **30**(12) 1121-1128.

Valdez-Flores, C., Feldman, R. 1989. Survey of preventive maintenance models for stochastically deteriorating single-unit systems. *Naval Research Logistics* **36**(4) 419-446.

Wang, H. 2002. A Survey of Maintenance Policies of Deteriorating Systems. *European Journal of Operational Research* **139**(3) 469-489.

White, K.P., Jr. 1997. An Effective Truncation Heuristic for Bias Reduction in Simulation Output. *Simulation* **69**(6) 323-334.

Appendices

A.1 Simulation Availability Model

```
Option Explicit

Const nReps As Long = 153667
Const nObs As Long = 50000
Const a As Double = 0.4
Const Beta As Double = 1.5
Const eta As Double = 1
Const Theta As Double = 0.2
Const tr As Double = 0.05
Const tp As Double = Theta * tr
Const Taw As Double = 200
Const t_end As Long = 200

Dim r As Long
Dim tmr As Double
Dim i As Long
Dim Scale_ As Double

Dim DownType As Long
Dim t_now As Double
Dim x As Long
Dim U As Double
Dim T As Double
Dim NextDown As Double
Dim Age As Double
Dim AgeAtDown As Double
Dim AgeAtUp As Double
Dim NextUp As Double
Dim NextObs As Double
Dim UStart As Double

Dim Sum_X(nObs) As Double
Dim Sum_U(nObs) As Double

Private Function Random() As Double
Dim x As Double
    x = Rnd()
    Do While x = 0
        x = Rnd()
    Loop
    Random = x
```

```
End Function

Public Sub Main()
   Initialize
   For r = 1 To nReps
      Replicate
      If r Mod 1000 = 0 Then
         Debug.Print "Rep#" & r & " completed, estimated remaining time is " & Round((Timer - tmr) / r * (nReps - r) / 60, 1) & " min"
         DoEvents
      End If
   Next
   Output
End Sub

Private Sub Initialize()
Dim i As Long
   Randomize
   tmr = Timer
   For i = 0 To nObs
      Sum_X(i) = 0
      Sum_U(i) = 0
   Next
   Scale_ = t_end / nObs
End Sub

Private Sub Replicate()
   t_now = 0
   x = 1
   Age = 0
   NextUp = t_end + 1
   i = 1
   NextObs = Scale_
   UStart = 0

   T = eta * ((Age / eta) ^ Beta - Log(Random())) ^ (1 / Beta) - Age

   If T > Taw Then
      NextDown = Taw
      AgeAtDown = Taw
      DownType = 1
   Else
      NextDown = T
      AgeAtDown = T
      DownType = 2
   End If
```

```
    Do While i <= nObs
        If NextObs <= NextUp And NextObs <= NextDown Then Observation
        If NextDown < NextUp And NextDown < NextObs Then Down
        If NextUp < NextDown And NextUp < NextObs Then Up
    Loop
End Sub

Private Sub Down()
    t_now = NextDown
    Age = AgeAtDown
    x = 0
    U = U + (t_now - UStart)
    NextDown = t_end + 1

    If DownType = 1 Then
        NextUp = t_now + tp
        AgeAtUp = 0
    Else
        NextUp = t_now + tr
        AgeAtUp = a * AgeAtDown
    End If
End Sub

Private Sub Up()
    t_now = NextUp
    Age = AgeAtUp
    x = 1
    NextUp = t_end + 1
    UStart = t_now

    T = eta * (-Log(Random()) + (Age / eta) ^ Beta) ^ (1 / Beta) - Age

    If Age + T > Taw Then
        NextDown = t_now + (Taw - Age)
        AgeAtDown = Taw
        DownType = 1
    Else
        NextDown = t_now + T
        AgeAtDown = Age + T
        DownType = 2
    End If
End Sub

Private Sub Observation()
    t_now = NextObs
```

```
    If x = 1 Then
        U = U + (t_now - UStart)
        UStart = t_now
    End If

    Sum_X(i) = Sum_X(i) + x
    Sum_U(i) = Sum_U(i) + U
    i = i + 1
    NextObs = t_now + Scale_
End Sub

Private Sub Output()
Dim j As Long
Dim k As Long
Dim t_time(nObs) As Double
Dim Avail(nObs) As Double

    For j = 1 To nObs
        t_time(j) = j * Scale_
         Avail(j) = Sum_X(j) / nReps
    Next

    Worksheets("Results").Range("A:Z").Clear
    Worksheets("Results").Cells(1, 1).Value = "t_time"
    Worksheets("Results").Cells(1, 2).Value = "Avail"
    Worksheets("Results").Cells(2, 1).Value = 0
    Worksheets("Results").Cells(2, 2).Value = 1
    For k = 1 To nObs
        Worksheets("Results").Cells(k + 2, 1).Value = t_time(k)
        Worksheets("Results").Cells(k + 2, 2).Value = Avail(k)
    Next

Debug.Print "Estimating truncation time point"
Dim d As Long
Dim MSER As Double
Dim MSER_min As Double
Dim Y_bar_n_d As Double
Dim Sum As Double

    MSER_min = 10 ^ 30
    d = 0
    For k = 1 To nObs - 2

        Sum = 0
        For j = k + 1 To nObs
            Sum = Sum + Avail(j)
```

```
        Next
        Y_bar_n_d = 1 / (nObs - k) * Sum

        Sum = 0
        For j = k + 1 To nObs
           Sum = Sum + (Avail(j) - Y_bar_n_d) ^ 2
        Next
        MSER = 1 / (nObs - k) ^ 2 * Sum

        If MSER < MSER_min Then
           MSER_min = MSER
           d = k
        End If

    Next
    Worksheets("Results").Cells(1, 3).Value = "MSER*"
    Worksheets("Results").Cells(1, 4).Value = MSER_min
    Worksheets("Results").Cells(2, 3).Value = "d*"
    Worksheets("Results").Cells(2, 4).Value = t_time(d)

    Debug.Print "Done in " & Round((Timer - tmr) / 60, 1) & " Min"
    MsgBox ("Done in " & Round((Timer - tmr) / 60, 1) & " Min")

End Sub
```

A.2 Enumeration Model

```
Option Base 1
Option Explicit

Dim tmr As Single

Dim P1() As Integer
Dim R1() As Integer
Dim x1() As Single
Dim y1() As Single
Dim P2() As Integer
Dim R2() As Integer
Dim x2() As Single
Dim y2() As Single
Dim CostofAction() As Single
Dim ExpNoFailur1() As Single
Dim ExpNoFailur2() As Single
Dim q() As Integer

Dim NumberofPeriods As Integer
Dim TotalTime As Single
Dim Percentage As Single
Dim CostofFailure1 As Single
Dim CostofReplacment1 As Single
Dim CostofEndPM1 As Single
Dim Beta1 As Single
Dim Eta1 As Single
Dim alpha1 As Single
Dim CostofFailure2 As Single
Dim CostofReplacment2 As Single
Dim CostofEndPM2 As Single
Dim Beta2 As Single
Dim Eta2 As Single
Dim alpha2 As Single

Dim delta As Single

Public Sub GetCost()
   tmr = Timer
   NumberofPeriods = 3
   TotalTime = 12
   alpha1 = 0.4
   CostofFailure1 = 10
   CostofReplacment1 = 3
```

```
    CostofEndPM1 = 1.5
    Eta1 = 2
    Beta1 = 1.5
    alpha2 = 0.25
    CostofFailure2 = 15
    CostofReplacment2 = 5
    CostofEndPM2 = 2.5
    Eta2 = 3
    Beta2 = 2
    delta = 1
    Cost (NumberofPeriods)
End Sub

Public Sub Cost(n As Integer)

Dim i As Integer
Dim j As Integer
Dim k As Integer
Dim Text As String
Dim MinimumCost As Single
Dim NPV As Single
Dim Action() As Integer
Dim OptAction() As Integer

Dim FinishedAll As Boolean

    ReDim P1(n) As Integer
    ReDim R1(n) As Integer
    ReDim x1(n) As Single
    ReDim y1(n) As Single
    ReDim CostofAction(n) As Single
    ReDim ExpNoFailur1(n) As Single
    ReDim Action(n * 2) As Integer
    ReDim OptAction(n * 2) As Integer
    ReDim P2(n) As Integer
    ReDim R2(n) As Integer
    ReDim x2(n) As Single
    ReDim y2(n) As Single
    ReDim ExpNoFailur2(n) As Single
    ReDim q(n) As Integer

    MinimumCost = 1e+18
    FinishedAll = False

    x1(1) = 0
    x2(1) = 0
```

```
j = NumberofPeriods + 1

For i = 1 To NumberofPeriods
   Action(i) = 0
Next

Do While Not FinishedAll


   FinishedAll = True
   For i = 1 To NumberofPeriods
      If Action(i) <> 2 Then
         FinishedAll = False
         Exit For
      End If
   Next


   For i = 1 To NumberofPeriods
      If Action(i) = 1 Then
         P1(i) = 1
      Else
         P1(i) = 0
      End If

      If Action(i) = 2 Then
         R1(i) = 1
      Else
         R1(i) = 0
      End If
   Next

   For i = 1 To NumberofPeriods
      If Action(i + NumberofPeriods) = 1 Then
         P2(i) = 1
      Else
         P2(i) = 0
      End If

      If Action(i + NumberofPeriods) = 2 Then
         R2(i) = 1
      Else
         R2(i) = 0
      End If
   Next
```

```
    NPV = 0
    For i = 1 To NumberofPeriods
        y2(i) = x2(i) + TotalTime / NumberofPeriods
        If i < n Then
            x2(i + 1) = y2(i) * (1 - ((1 - alpha2) * P2(i)) - R2(i))
        End If
        ExpNoFailur2(i) = ((y2(i) / Eta2)) ^ Beta2 - ((x2(i) / Eta2)) ^ Beta2
        CostofAction(i) = CostofFailure2 * ExpNoFailur2(i) + CostofEndPM2 * P2(i) +
CostofReplacment2 * R2(i)


        NPV = NPV + CostofAction(i)
    Next

    If NPV < MinimumCost Then
        MinimumCost = NPV
        For i = 1 To NumberofPeriods
            OptAction(i) = Action(i)
        Next
    End If

    Text = ""
    For i = 1 To NumberofPeriods
        Text = Text & Action(i) & ","
    Next
   Debug.Print Text & NPV
    DoEvents

    If FinishedAll = False Then
        For i = 1 To NumberofPeriods
            If Action(i) < 2 Then
                Action(i) = Action(i) + 1
                For k = 1 To i - 1
                    Action(k) = 0
                Next
                Exit For
            End If
        Next
    End If
Loop

Text = ""
For i = 1 To NumberofPeriods
    Text = Text & OptAction(i) & ","
Next
```

```
    For i = 1 To NumberofPeriods
        Worksheets("sheet3").Cells(i + 2, 2).Value = ""
        Worksheets("sheet3").Cells(i + 2, 3).Value = ""
    Next

    For i = 1 To NumberofPeriods
        Worksheets("sheet3").Cells(i + 2, 2).Value = i
        Worksheets("sheet3").Cells(i + 2, 3).Value = OptAction(i)
    Next

    Worksheets("sheet3").Cells(2, 4).Value = MinimumCost
    Worksheets("sheet3").Cells(2, 5).Value = Timer - tmr

    MsgBox "tmr= " & Timer - tmr & ", minimum cost= " & MinimumCost & " at " & Text
    Debug.Print n, Timer - tmr
End Sub
```

A.3 Genetic Algorithm Model

```
Option Base 1
Option Explicit

Const TotalTime As Integer = 12
Const alpha1 As Single = 0.4
Const CostofReplacment1 As Single = 3
Const CostofReplacment2 As Single = 5
Const CostofFailure1 As Single = 10
Const CostofFailure2 As Single = 15
Const CostofEndPM1 As Single = 1.5
Const CostofEndPM2 As Single = 2.5
Const Eta1 As Single = 2
Const Beta1 As Single = 1.5
Const alpha2 As Single = 0.25
Const Eta2 As Single = 3
Const Beta2 As Single = 2
Const delta As Single = 1


Const PopulationSize As Integer = 1000
Const NumberofPeriods As Integer = 4
Const NumberofGenerations As Integer = 1000
Const CrossPointPercentage As Single = 0.1
Const ReplicationPercentage As Single = 0.95

Dim NewChildNumber As Integer
Dim GACost(NumberofGenerations) As Single

Type SolutionObject
   Coding(NumberofPeriods * 2) As Integer
   ObjectiveValue As Single
End Type

Dim tmr As Single

Dim Solution(PopulationSize * 2) As SolutionObject

Private Function UNIF(Lb As Integer, Ub As Integer) As Integer
   UNIF = Int((Ub - Lb + 1) * Rnd + Lb)
End Function

Public Sub GAMain()
Dim i As Integer
```

```
Dim j As Integer

   tmr = Timer
   Randomize

   Initilize
   For i = 1 To NumberofGenerations
      Replication (Fix(PopulationSize * ReplicationPercentage))
      Mutation (PopulationSize - Fix(PopulationSize * ReplicationPercentage))
      'get objective value for all new solutions
      For j = PopulationSize + 1 To PopulationSize * 2
         Solution(j).ObjectiveValue = EvaluateObjectiveValue(j)
      Next
      InsertSort Solution(), 1, PopulationSize * 2
      KillLowerPopulation
      GACost(i) = Solution(1).ObjectiveValue
   Next

   Output

End Sub

Private Sub Initilize()

Dim i As Integer
Dim j As Integer

   For j = 1 To PopulationSize
      For i = 1 To NumberofPeriods * 2 - 2
         Solution(j).Coding(i) = UNIF(0, 2)
      Next
      Solution(j).Coding(NumberofPeriods * 2 - 1) = 2
      Solution(j).Coding(NumberofPeriods * 2) = 2
      Solution(j).ObjectiveValue = EvaluateObjectiveValue(j)
   Next
End Sub

Private Sub Replication(NumberofNewChildren As Integer)
Dim Parent1 As Integer
Dim Parent2 As Integer
Dim CrossPoint As Integer
Dim i As Integer
Dim j As Integer
Dim k As Integer 'debug

  NewChildNumber = PopulationSize + 1
```

```
  For i = NewChildNumber To PopulationSize + NumberofNewChildren
      Parent1 = UNIF(1, PopulationSize)
      Parent2 = UNIF(1, PopulationSize)
      CrossPoint = (UNIF((1 + CrossPointPercentage * NumberofPeriods), (NumberofPeriods -
CrossPointPercentage * NumberofPeriods))) * 2

      For j = 1 To NumberofPeriods * 2
         If j < CrossPoint Then
           Solution(i).Coding(j) = Solution(Parent1).Coding(j)
         Else
            Solution(i).Coding(j) = Solution(Parent2).Coding(j)
         End If
      Next
   Next
   NewChildNumber = PopulationSize + NumberofNewChildren + 1
End Sub

Private Sub Mutation(NumberofNewChildren As Integer)
Dim Parent As Integer
Dim Position As Integer
Dim i As Integer
Dim j As Integer
Dim k As Integer

  For i = NewChildNumber To NewChildNumber + NumberofNewChildren - 1
      Parent = UNIF(1, PopulationSize)
      Position = UNIF(1, NumberofPeriods * 2)
      For j = 1 To NumberofPeriods * 2
         If j = Position And j < (NumberofPeriods * 2 - 2) Then
            'mutate coding
            k = UNIF(0, 2)
            Do While k = Solution(Parent).Coding(j)
               k = UNIF(0, 2)
            Loop
            Solution(i).Coding(j) = k
         Else
            'copy coding
            Solution(i).Coding(j) = Solution(Parent).Coding(j)
         End If
      Next
   Next
End Sub

Private Function EvaluateObjectiveValue(j As Integer) As Single

Dim i As Integer
```

```
Dim NPV As Single
Dim P1(NumberofPeriods * 2) As Integer
Dim R1(NumberofPeriods * 2) As Integer
Dim x1(NumberofPeriods * 2) As Single
Dim y1(NumberofPeriods * 2) As Single
Dim CostofAction(NumberofPeriods * 2) As Single
Dim ExpNoFailur1(NumberofPeriods * 2) As Single
Dim P2(NumberofPeriods * 2) As Integer
Dim R2(NumberofPeriods * 2) As Integer
Dim x2(NumberofPeriods * 2) As Single
Dim y2(NumberofPeriods * 2) As Single
Dim ExpNoFailur2(NumberofPeriods * 2) As Single
Dim q(NumberofPeriods * 2) As Integer

  For i = 1 To NumberofPeriods
    If Solution(j).Coding(1 + ((i - 1) * 2)) = 1 Then
      P1(i) = 1
    Else
      P1(i) = 0
    End If

    If Solution(j).Coding(1 + ((i - 1) * 2)) = 2 Then
      R1(i) = 1
    Else
      R1(i) = 0
    End If
  Next

  For i = 1 To NumberofPeriods
    If Solution(j).Coding(i * 2) = 1 Then
      P2(i) = 1
    Else
      P2(i) = 0
    End If

    If Solution(j).Coding(i * 2) = 2 Then
      R2(i) = 1
    Else
      R2(i) = 0
    End If

  Next

  For i = 1 To NumberofPeriods
    If P1(i) + P2(i) + R1(i) + R2(i) < 1 Then
      q(i) = P1(i) + P2(i) + R1(i) + R2(i)
```

```
      Else
         q(i) = 1
      End If

   Next

   x1(1) = 0
   x2(1) = 0
   NPV = 0
   For i = 1 To NumberofPeriods
      y1(i) = x1(i) + (TotalTime / (NumberofPeriods))
       y2(i) = x2(i) + (TotalTime / (NumberofPeriods))
      If i < NumberofPeriods Then
         x1(i + 1) = y1(i) * (1 - ((1 - alpha1) * P1(i)) - R1(i))
            x2(i + 1) = y2(i) * (1 - ((1 - alpha2) * P2(i)) - R2(i))
         End If
         ExpNoFailur1(i) = ((y1(i) / Eta1)) ^ Beta1 - ((x1(i) / Eta1)) ^ Beta1
         ExpNoFailur2(i) = ((y2(i) / Eta2)) ^ Beta2 - ((x2(i) / Eta2)) ^ Beta2
         CostofAction(i) = CostofFailure1 * ExpNoFailur1(i) + CostofFailure2 * ExpNoFailur2(i)
+ CostofEndPM1 * P1(i) + CostofEndPM2 * P2(i) + CostofReplacment1 * R1(i) +
CostofReplacment2 * R2(i) + delta * q(i)
         NPV = NPV + CostofAction(i)
   Next

   EvaluateObjectiveValue = NPV

End Function

Private Sub KillLowerPopulation()
Dim i As Integer
Dim j As Integer

   For i = PopulationSize + 1 To PopulationSize * 2
      Solution(i).ObjectiveValue = 999999999999#
      For j = 1 To NumberofPeriods * 2
         Solution(i).Coding(j) = -1
      Next
   Next
End Sub

Private Sub InsertSort(ByRef A() As SolutionObject, ByVal Lb As Long, ByVal Ub As Long)
   Dim t As SolutionObject
   Dim i As Long
   Dim j As Long

   ' sort A[Lb..Ub]
```

```
    For i = Lb + 1 To Ub
        t = A(i)

        ' shift elements down until insertion point found
        For j = i - 1 To Lb Step -1
            If A(j).ObjectiveValue <= t.ObjectiveValue Then Exit For
            A(j + 1) = A(j)
        Next j

        ' insert
        A(j + 1) = t
    Next i
End Sub

Private Sub Output()
Dim i As Integer

    Worksheets("GA Results").Range("A:H").Clear

    Worksheets("GA Results").Cells(1, 1).Value = "Gen #"
    Worksheets("GA Results").Cells(1, 2).Value = "Best Sol. Cost"
    For i = 2 To NumberofGenerations + 1
        Worksheets("GA Results").Cells(i, 1).Value = i - 1
        Worksheets("GA Results").Cells(i, 2).Value = GACost(i - 1)
    Next

    Worksheets("GA Results").Cells(1, 4).Value = "Period"
    Worksheets("GA Results").Cells(1, 5).Value = "Machine"
    Worksheets("GA Results").Cells(1, 6).Value = "Action"
    Worksheets("GA Results").Cells(1, 7).Value = "Final Solution Cost"
    Worksheets("GA Results").Cells(2, 7).Value = Solution(1).ObjectiveValue
    Worksheets("GA Results").Cells(1, 8).Value = "Run Time (Sec.)"
    Worksheets("GA Results").Cells(2, 8).Value = Timer - tmr
    For i = 2 To NumberofPeriods * 2 + 1
        Worksheets("GA Results").Cells(i, 4).Value = Fix(i / 2)
        Worksheets("GA Results").Cells(i, 5).Value = (i Mod 2) + 1
        Worksheets("GA Results").Cells(i, 6).Value = Solution(1).Coding(i - 1)
    Next

End Sub
Public Sub Sinan()
Dim i As Integer
Dim k As Integer

    k = 200
    For i = 1 To NumberofPeriods * 2
```

```
        Solution(k).Coding(i) = 2
    Next
    Solution(k).ObjectiveValue = EvaluateObjectiveValue(k)
End Sub
```

A.4 Analysis of Variance from Minitab for *A*

Response Surface Regression: A versus β, a, t_r

Estimated Regression Coefficients for *A*

Term	Coef	SE Coef	T	P
Constant	0.89652	0.16080	5.575	0.003
β	-0.06333	0.07541	-0.840	0.439
a	0.76960	0.28278	2.722	0.042
t_r	-0.84004	0.90137	-0.932	0.394
β*β	0.02787	0.01493	1.867	0.121
a*a	-0.49758	0.20989	-2.371	0.064
t_r*t_r	4.07420	3.35820	1.213	0.279
β*a	-0.12842	0.04641	-2.767	0.040
β*t_r	-0.26367	0.18565	-1.420	0.215
a*t_r	-1.48125	0.69618	-2.128	0.087

S = 0.0196909 PRESS = 0.0167642
R-Sq = 98.54% R-Sq(pred) = 87.39% R-Sq(adj) = 95.92%

Analysis of Variance for *A*

Source	DF	Seq SS	Adj SS	Adj MS	F	P
Regression	9	0.1309	0.1309	0.0145	37.53	0.000
Linear	3	0.1180	0.0064	0.0021	5.53	0.048
Square	3	0.0074	0.0074	0.0024	6.42	0.036
Interaction	3	0.0055	0.0055	0.0018	4.73	0.064
Res. Error	5	0.0019	0.0019	0.0003		
Total	14	0.1329				

A.5 Analysis of Variance from Minitab for *A* (adding PM parameters)

Response Surface Regression: A versus β, a, t_r, t_{PM}, τ

Estimated Regression Coefficients for *A*

Term	Coef	SE Coef	T	P
Constant	0.7058	0.1790	3.942	0.008
β	0.0545	0.0878	0.621	0.558
a	0.1035	0.3293	0.314	0.764
t_r	-0.0088	1.0720	-0.008	0.994
t_{PM}	1.6589	5.3602	0.309	0.767
τ	0.3606	0.0942	3.828	0.009
β*β	0.0083	0.0166	0.499	0.636
a*a	0.1463	0.2339	0.625	0.555
t_r*t_r	2.4375	3.7425	0.651	0.539
t_{PM}*t_{PM}	45.9857	93.5625	0.491	0.641
τ*τ	-0.0903	0.0327	-2.761	0.033
β*a	-0.0494	0.0505	-0.977	0.366
β*t_r	-0.0399	0.2021	-0.197	0.850
β*t_{PM}	-0.4221	1.0104	-0.418	0.691
β*τ	-0.0595	0.0189	-3.151	0.020
a*t_r	-0.1881	0.7578	-0.248	0.812
a*t_{PM}	-5.5899	3.7891	-1.475	0.191
a*τ	-0.0871	0.0708	-1.230	0.265
t_r* t_{PM}	-14.1596	15.1562	-0.934	0.386
t_r*τ	-0.8227	0.2833	-2.904	0.027
t_{PM}*τ	1.3461	1.4165	0.950	0.379

S = 0.0303124 PRESS = 0.264488
R-Sq = 93.16% R-Sq(pred) = 0.00% R-Sq(adj) = 70.36%

Analysis of Variance for *A*

Source	DF	Seq SS	Adj SS	Adj MS	F	P
Regression	20	0.0750	0.0750	0.0037	4.09	0.044
Linear	5	0.0441	0.0136	0.0027	2.97	0.109
Square	5	0.0079	0.0079	0.0015	1.72	0.262
Interaction	10	0.0230	0.0230	0.0023	2.51	0.136
Res. Error	6	0.0055	0.0055	0.0009		
Total	26	0.0805				