ABSTRACT

Title of Document: A "DESIGN FOR AVAILABILITY" METHODOLOGY FOR SYSTEMS DESIGN AND SUPPORT

Taoufik Jazouli, Doctor of Philosophy, 2011

Directed By: Professor, Peter Sandborn, Mechanical Engineering

Prognostics and Health Management (PHM) methods are incorporated into systems for the purpose of avoiding unanticipated failures that can impact system safety, result in additional life cycle cost, and/or adversely affect the availability of a system. Availability is the probability that a system will be able to function when called upon to do so. Availability depends on the system's reliability (how often it fails) and its maintainability (how efficiently and frequently it is pro-actively maintained, and how quickly it can be repaired and restored to operation when it does fail). Availability is directly impacted by the success of PHM. Increasingly, customers of critical systems are entering into "availability contracts" in which the customer either buys the availability of the system (rather than actually purchasing the system itself) or the amount that the system developer/manufacturer is paid is a function of the availability achieved by the customer. Predicting availability based on known or predicted system reliability, operational parameters, logistics, etc., is relatively straightforward and can be accomplished using several methods and many existing tools. Unfortunately in these approaches availability is an output of the analysis. The prediction of system's parameters (i.e., reliability, operational parameters, and/or logistics management) to meet an availability requirement is difficult and cannot be generally done using today's existing methods. While determining the availability that results from a set of events is straightforward, determining the events that result in a desired availability is not.

This dissertation presents a "design for availability" methodology that starts with an availability requirement and uses it to predict the required design, logistics and operations parameters. The method is general and can be applied when the inputs to the problem are uncertain (even the availability requirement can be represented as a probability distribution). The method has been demonstrated on several examples with and without PHM.

A "DESIGN FOR AVAILABILITY" METHODOLOGY FOR SYSTEMS DESIGN AND SUPPORT

By

Taoufik Jazouli

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park, in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2011

Advisory Committee:
Professor Peter Sandborn, Chair
Professor Martin Dresner
Professor Abhijit Dasgupta
Professor Bilal M. Ayyub
Professor Donald Barker

Dedication

To my parents, my wife, my brother and my sisters. Also, I would like to dedicate this work to my officemates at CALCE.

Acknowledgements

I would like to acknowledge my advisor, Prof. Peter Sandborn, for his support, guidance, and encouragement. I would like to acknowledge the National Science Foundation, Division of Design and Manufacturing Innovation (Grant Number: CMMI-1129697), for their support. I would also like to thank the more than 100 companies and organizations that support research at the Center for Advanced Life Cycle Engineering (CALCE) at the University of Maryland annually. Specifically, I would like to acknowledge the members of the Prognostics and Health Management (PHM) Consortium at CALCE who provided valuable feedback and insight into practical applications of the design for availability methodology.

Table of Contents

Dedication ii
Acknowledgements iii
Table of Contents iv
List of Tables v
List of Figures vi
Chapter 1: Introduction 1
1.1. Background 4
1.2. Research Scope and Objectives 11
Chapter 2: Maintenance ROI Model 14
2.1. Discrete Event Simulation Maintenance Planning Model 16
2.2. Fixed-Schedule Maintenance Interval 18
2.3. Data-Driven (Precursor to Failure Monitoring) Methods 18
2.4. Model-Based (LRU-Independent) Methods 20
2.5. Discrete Event Simulation Implementation Details 23
2.6. Spares Management 26
2.7. Operational Profile 27
2.8. Implementation Costs 29
2.9. Return on Investment (ROI) Calculation 30
2.10. Summary 33
Chapter 3: Design for Availability 34
3.1. Domain of Applicability – An Example Design for Availability Problem Statement 34
3.2. Introduction to Design for Availability 36
3.3. Design for Availability Approach 38
Chapter 4: Application of the Methodology 53
4.1. Case Study Data Inputs 54
4.2. Use of Design for Availability to Determine System Parameters Affecting either Uptime or Downtime 55
4.3. Use of Design for Availability to Determine System Parameters Affecting both Uptime and Downtime 71
Chapter 5: Summary and Contributions 87
5.1. Summary 87
5.2. Contributions 89
5.3. Potential Broader Impacts of this Work 90
5.4. Future Work 91
Appendix 93
Bibliography 99

List of Tables

Table 2.1: Data Defining Unscheduled Maintenance Operational Profile …………28

Table 4.1: Implementation Costs ……………………………………………….....54

Table 4.2: Operational Profile ………………………….…………………………55

Table 4.3: Spares Inventory ………………………………………………………56

Table 4.4: Spares Inventory ………………………………………………………73

Table A.1: Spares Inventory ……………………………………………………...95

Table A.2: Spares Inventory ……………………………………………………...97

List of Figures

Figure 2.1: Data-driven (precursor to failure monitoring) modeling approach. Symmetric triangular distributions are chosen for illustration. Note, the LRU TTF PDF (left) and the data-driven TTF PDF (right) are not the same (they could have different shapes and sizes) …………………………………………………………..18

Figure 2.2: Model-based (LRU-independent) modeling approach. Symmetric triangular distributions are chosen for illustration. Note, the LRU TTF PDF (left) and the model-based method TTF PDF (right) are not the same (they could have different shapes and sizes) ………………………………………………………………….22

Figure 2.3: Temporal ordering of implementation cost inclusion in the discrete event simulation ……………………………………………………………………………23

Figure 2.4: Spares management for a single socket …………………………………26

Figure 3.1: Design for Availability Methodology …………………………………..38

Figure 3.2: Availability variations as a function of time ……………………………40

Figure 3.3: Scenarios of downtimes requirement …………………………………...46

Figure 3.4: General case of updating downtime requirement ………………………48

Figure 3.5: General case of updating uptime requirement …………………………..48

Figure 4.1: Weibull distribution of TTFs (β=1.1, η=200 and γ=9000) ……………...57

Figure 4.2: Implication of the inventory model parameters on the timeline ………..58

Figure 4.3: (a) Required availability distribution (input to the model). (b) Computed maximum allowable Inventory Lead Time (ILT) in operational hours, for an unscheduled maintenance policy. (c) Availability probability distribution generated using the computed ILT (Figure 4.3.b) ………………………………………………...60

Figure 4.4: Solution process ………………………………………………………...60

Figure 4.5: Computed maximum allowable inventory lead time (ILT) for unscheduled maintenance policy, and for a data-driven PHM approach …………………………62

Figure 4.6: Variation of life cycle cost and inventory lead time with data-driven PHM prognostic distance …………………………………………………………………..63

Figure 4.7: Computed TTF cumulative distribution function for unscheduled maintenance and data-driven PHM ………………………………………………….66

Figure 4.8: Life cycle cost per socket for a 1000 socket population. (a) Life cycle cost per socket for an unscheduled maintenance. (b) Life cycle cost per socket for a data-driven PHM approach ……………………………………………………………….68

Figure 4.9: Histogram of ROI for a 1000-socket population, for a data-driven PHM relative to unscheduled maintenance that meets the availability requirement ………69

Figure 4.10: Histogram of ROI for a 1000-socket population, for a data-driven PHM relative to unscheduled maintenance that uses an ILT distribution generated by the data-driven PHM ……………………………………………………………………70

Figure 4.11: TTF implication on the operational timeline …………………………..74

Figure 4.12: Updating the TTFs, UTs and DTs ……………………………………..76

Figure 4.13: (a) Required availability distribution (input to the model). (b) Computed minimum allowable reliability (TTF) in operational hours, for an unscheduled maintenance policy. (c) Availability probability distribution generated using the computed TTF (Figure 4.13.b) ……………………………………………………………77

Figure 4.14: Variations of life cycle cost and minimum allowable reliability (TTF) with data-driven PHM prognostic distance …………………………………………78

Figure 4.15: Computed minimum allowable reliability (TTF) distribution for unscheduled maintenance and data-driven PHM ……………………………………80

Figure 4.16: Computed TTF cumulative distribution function for unscheduled maintenance and data-driven PHM ………………………………………………….82

Figure 4.17: (a) Life cycle cost per socket for an unscheduled maintenance policy. (b) Life cycle cost per socket for a data-driven PHM approach ………………………...83

Figure 4.18: Histogram of ROI for a 1000-socket population, for a data-driven PHM relative to unscheduled maintenance that meets the availability requirement ………85

Figure 4.19: Histogram of ROI for a 1000-socket population, for a data-driven PHM relative to unscheduled maintenance that uses a TTF distribution generated by a data-driven PHM ……………………………………………………………………….86

Figure A.1: Weibull distribution of TTFs (β=1.1, η=200 and γ=9000) ……………..94

Figure A.2: (a) Required availability distribution (input to the model). (b) Computed spares threshold (ST), for a data-driven PHM approach. (c) Availability probability distribution generated using the computed ST (Figure A.2.b) ………………………95

Figure A.3: (a) Required availability distribution (input to the model). (b) Computed replenishment spares (RS), for an unscheduled maintenance policy. (c) Availability probability distribution generated using the computed RS (Figure A.3.b) …………97

Chapter 1: Introduction

Availability is the ability of a service or a system to be functional when it is requested for use or operation. The availability of an item is a function of its reliability and logistics management (including repairs, replacements and inventory management). Availability accounts for both the frequency of failure (reliability) and the ability to restore the service or system to operation after a failure (maintainability). The dependency on maintenance generally translates into how quickly it can be repaired upon failure and are usually driven by the logistics management and are directly influenced by Prognostics and Health Management (PHM) approaches that may be used. The frequency of the failure is related to the reliability of the system, i.e., how long it will be operational (i.e., "up") before it fails.

Availability is a significant issue for many real world systems. A failure, i.e., a decrease of availability, of an ATM machine causes inconvenience to customers; the unavailability of a point-of-sale system to retail outlets can generate a huge financial loss; the unavailability a medical device or of hospital equipment can result in loss of life; unavailability of servers causes loss of data; poor availability of alternative energy generation (e.g., wind farms) can make them non-viable; and the unavailability of aircraft cause airlines to cancel or delay flights and military missions to be canceled. In these systems, insuring the availability of the system is important and the owners of the systems are often willing to pay a premium for higher availability. For many safety, mission, and infrastructure critical systems, availability is a more important design criteria than acquisition cost.

Several different types of availability can be measured (e.g., inherent, achieved, operational, etc.) [1]. This dissertation is focused on the operational availability since it implicitly incorporates other forms of availability and it is the most commonly used form of availability in availability contracts; however, the methodology could be easily adjusted to incorporate other types of availability.

Operational availability is the probability that a system or piece of equipment operates ordinarily, i.e. functional and available for operation when requested, over a specific period of time under stated conditions [2, 3]. Operational availability (A_0) accounts for all types of maintenance and logistics downtimes. It is computed as the ratio of the accumulated uptime and the sum of the accumulated uptime and downtime:

$$A_O = \frac{uptime}{uptime + downtime} \qquad (1.1)$$

where uptime is the total accumulated operational time during which the system is up and running and able to perform the tasks that are expected from it; while downtime is generated when the system is down and not operating when requested due to repair, replacement, waiting for spares or any other logistics delay time. The summation of the accumulated uptimes and downtimes represents the total operation time for the system.

Customers of avionics, large scale production lines, servers, and infrastructure service providers with high availability requirements are increasingly interested in buying the availability of a system, instead of actually buying the system itself; therefore, the concept of "availability-based contracting" has been introduced. Availability based contracts are a subset of outcome based contracts [4]; where the customer pays for the delivered outcome, instead of paying for specific logistics activities, system reliability management or other tasks. Basically, in this type of contract, the customer pays the service or system provider to ensure a specific availability requirement. For example the Availability Transformation: Tornado Aircraft Contract–ATTAC [5], is an availability contract, where BAE Systems has agreed to support the Tornado GR4 aircraft fleet at a specified availability level throughout the fleet service life for the UK Ministry of Defense. The agreement aims to implement a new cost-effective approach to improve the availability of the fleet while minimizing the life cycle cost [5]. Another form of outcome-based contracting that is used by the U.S. Department of Defense is called performance-based contracting (or PBL – Performance Based Logistics). In PBL contracts the contractor is paid based on the results achieved, not on the methods used to perform the tasks [6, 7]. Availability contracts, and most outcome-based contacting, include cost penalties that could be assessed for failing to fulfill a specified availability requirement within a defined time frame (or a contract payment schedule that is based on the achieved availability).

The evaluation of an availability requirement is a challenging task for both suppliers and customers. From a suppliers' perspective, it is not trivial to estimate the cost of delivering a specific availability. Entering into an availability contract is a non-traditional way of doing business for the suppliers of many types of safety- and mission-critical systems. For example, the traditional avionics supply chain business model is to sell the system; and then separately provide the sustainment of the system. As a result, the avionics suppliers sell the system for whatever they have to in order to obtain the business, knowing that they will make their money on the long-term sustainment of the system. From a customers' perspective, the amount of money that should be spent on a specific availability contract is also a mysterious quantity – if a choice has to be made between two offers of availability contracts where the value of the promised availabilities are close (e.g., one contract offers an availability of 95% at any defined time, and the other one offers 97%), how much money should the customer be willing to spend for a specific availability improvement?

1.1. Background

Types of Availability

Many different types of availability measures exist [1]; the most commonly used are: instantaneous, mean, steady-state, inherent, achieved, and operational availability. The main differences between these forms of availability are the types of activities that are excluded or included in the accumulated downtime and uptime values.

The instantaneous availability is the probability that a system will be operational at any time during its entire operational support life. Note, this probability could change after every repair event, since the reliability of the system either decreases or increases, due to the repair renewal function. The mean availability is related to the instantaneous availability; it is the mean value of the instantaneous availability over a defined period of time. The steady-state availability is defined as the limit of the instantaneous availability as time approaches infinity, i.e., after a significant number of repair events. Inherent availability is purely determined by the design of the system and the unscheduled maintenance actions. It assumes that the logistics management does not generate any delay time; i.e., the system is used under ideal logistics management. Similar to the inherent availability, the achieved availability assumes an ideal logistics management. However, it incorporates both scheduled and unscheduled maintenance into the accumulated downtimes. Basically, it is the probability that a system will operate satisfactorily in an ideal support environment. Finally, the operational availability, which is the most common type of availability appearing in availability contracts, is a measure of the availability that the customer actually experiences. Operational availability includes all sources of downtime (when the system is down while requested for operation), e.g., repair, replacement, waiting for spares replenishment, administrative downtime, or any other logistics downtime.

Readiness is closely related to availability and is a widely used metric for military applications. For availability, “downtime” is only operational downtime, while for readiness, “downtime” includes operational downtime, free time and storage time, [8]. Generally, the concept of readiness is broader than availability as it includes the operational availability of the system, the availability of the people who are needed to operate the system, and the availability of the infrastructure and other resources needed to support the operation of the system.

PHM (Prognostics and Health Management)

Most systems are repaired or replaced upon failure. However, for safety-critical systems, a failure could be very costly, even catastrophic. PHM [9] provides advanced warning of failures as well as appropriate decision making processes for maintenance planning. Prognostics is defined as the process of predicting the future life of a system or a product. Whereas, health management is the capability to make appropriate decisions about maintenance actions and/or other logistics parameters, based on prognostics information. PHM provides an opportunity for lowering sustainment costs, improving maintenance decision-making, maximizing availability, and providing product usage feedback into the product design and validation process. PHM implementation represents a potential transition from the unscheduled maintenance policy, where the system or component is repaired or replaced upon failure, to a scheduled maintenance policy, where a sustainment approach methodology is adopted to repair or replace the system or component before failure.

A subset of PHM that is only focused on reducing maintenance costs is referred to as Condition Based Maintenance (CBM) [10]. CBM is a set of maintenance processes and capabilities derived from real-time assessment of a system's condition. The goal of CBM is to perform maintenance only upon evidence of need.

Applicability to Electronic Systems

This dissertation's primary target application is electronic systems. While PHM and CBM have been performed on non-electronic systems for many years (sometimes known as structural health monitoring), it is far less prevalent for electronics. This is due to several factors including difficulties in identifying precursors to failure in electronics and the larger number of different failure mechanisms (and potential failure locations). Performing PHM for electronics is also difficult because of the high rate of electronics evolution compared to non-electronic systems (e.g., by the time one learns the warning signs of failure, the technology changes). Electronic systems maintenance culture is also historically based on "unscheduled maintenance" where systems are run to failure and then replaced or repaired. While the majority of the work in this dissertation is generally applicable to non-electronic systems, the case study examples considered will be electronic systems.

Logistics Management

Logistics is the management of the flow of the existing resources of a process or a service to perform a specific operational task or to meet a customers' requirement [11]. Logistics management provides a means to evaluate and control information flow. In an operational system environment, logistics management usually includes inventory, man power, administrative processing, transportation, etc. Each one of these activities could induce a delay time resulting in an operational downtime of the system. However, this dissertation will focus on the logistics delays related to maintenance and inventory management. The maintenance actions could generate operational downtimes based on the adopted maintenance policy, repair time, and replacement time. The inventory management could produce a delay when the system is down (while it is requested for operation) waiting for spares replenishment [12].

Availability Modeling

Logistics, maintenance, and availability can be evaluated using several different methods; one common method is discrete event simulation. A discrete-event simulation represents a set of chronological events where each event occurs at an instant in time and marks a change of state in the system. Time-based and event-based simulations are considered the two primary discrete-simulation modeling techniques. Time-based models follow the progress of the process as it occurs at discrete points in simulation time. At each time step the state of the process is observed precisely; however, its progress between any two consecutive time steps is assumed to be negligible and undetermined by any external observer. Thus, time-based modeling techniques assume that important changes only occur at the discrete time steps and the choice of the time step is based on the succession of the events as they occur in the simulation time. In event-based models, the occurrence of the events drives the progress of the modeled process, i.e., the process is event-dependent not time-dependent. In event-based modeling, the simulation is tracking the occurrence of the events as they happen. At every event the progress step, i.e., time step, is determined based on the occurrence of the next event, where the event refers to any significant incident associated with the state of the modeled process [13, 14]. Discrete-event simulation is commonly used to predict the availability and life cycle cost for systems design and support, e.g., [15].

Availability predictions used during the design and support of real systems are also performed using Markov models [16, 17].[1] However, while discrete event simulators track the current state of the system, and based on the present events, predict the occurrence of future events; Markov models do not explicitly embrace the concept of future events, rather they track the model state at each time step and sample how long the model will be in the current state before it switches to the next state [18]. Basically, each event in a discrete event simulator depends on the time spent in that event and the path that led to it, while Markov models depend only on the current state of the model regardless of the duration spent in the current state and the path that led to it [19]. Discrete-event simulators accumulate the outcomes resulting from the type and duration of previous events; and then use only the set of data inputs that are necessary at a specific point on the timeline to predict future events. Markov models incorporate all provided data to generate an analytical solution and use it to determine the current model state and to move to the next state.

Discrete-event simulators are generally more efficient than Markov models for modeling complex systems with large numbers of variables, specifically in data capturing without aggregation [19, 20]. In general, discrete-event simulators order the failure and maintenance events for a system temporally, and the durations associated with the failure and maintenance events can be readily accumulated to estimate availability. Thus, it is straightforward for a discrete-event simulation to compute the

[1] Other methods that are not discrete-event simulator based or Markov models for determining availability exist as well (e.g., [21], [22], [23], and many others), but most of these are confined to the evaluation of extremely simple systems that while preserving the essence of real problems, often have a too limited scope or are too oversimplified to be of practical value to problems where every input is uncertain and there are potentially 1000s of components within a single system to manage and maintain concurrently.

availability based on a particular sequence of failures, logistics and maintenance events. This dissertation will focus on discrete-event simulation based availability calculations.

Discrete-event simulation is often a preferred approach to modeling the maintenance of real systems when many different failure mechanisms (and/or different parts), all characterized by different failure distributions must be concurrently included within the model. This complexity is compounded by the necessity to consider a large population of systems in order to generate viable summary statistics.

Two common mechanisms that may include elements of availability contracting are Product Service Systems (PSS) and leasing models. PSS provide both the product and its service/support based on the customer's requirements [24], which could include an availability requirement. Lease contracts [25] are use-oriented PSS, where the ownership of the product is usually retained by the service provider. A lease contract may indicate not only the basic product and service provided but also other use and operation constraints such as the failure rate threshold. In leasing agreements the customer has an implicit expectation of a minimum availability, but the availability is generally not quantified contractually.

1.2. Research Scope and Objectives

The objective of this dissertation was to develop a “Design for Availability” methodology that enables the prediction of the system reliability, operational parameters, and/or logistics management parameters to meet a general availability requirement. The methodology must applicable to problems where the availability requirement is expressed as a probability distribution and uncertainties may be present in all the design and logistics properties of the system. In addition, the design for availability methodology must allow the determination of prognostics and health management (PHM) parameters so as to enable the assessment of cost/availability tradeoffs associated with the inclusion of PHM within systems, where cost includes the assessment of return on investment (ROI).

In order to achieve the objectives, described above, the following tasks have been completed:

Task 1: Construction of a maintenance model that has the ability to incorporate reliability information, implementation cost, and accommodate different maintenance policies. This model is a discrete-event simulator that allows the calculation of life cycle cost and ROI for different PHM approaches. This task was completed by another student prior to the start of this dissertation, see [26].

Task 2: Extend the maintenance model (Task 1) to include logistics management elements, i.e., specifically detailed spares management. This includes initial spares, spare replenishment criteria, lead time for spare replenishment, spares carrying cost, etc. Inclusion of a spares management model allows the calculation of availability concurrent with life cycle cost and ROI. Note this task creates the ability to calculate availability as an output (not an input).

Task 3: Formulate a general Design for Availability model that uses an availability input to generate the required system parameters to meet an availability requirement. This task is focused on the prediction of parameters affecting either uptime or downtime (not both). The following additional activities have been performed in this task:

a) Implemented the methodology in software so it can be tested.

b) Example application of the methodology to determine the inventory parameters (e.g., inventory lead time, threshold for spares replenishment, etc.) for a specific availability requirement.

c) Performed formal verification of the method by using the generated inventory parameters as inputs into the maintenance model to compute availability, and compare the resultant availability (output) to the availability that was the original requirement (input).

Task 4: Formulate a Design for Availability method to determine parameters that affect both uptime and downtime, for example, determining the reliability of the system for a specific availability requirement. This task will focus on determining a single consolidated reliability distribution describing a composite of the reliability associated with all relevant failure mechanisms for a system or subsystem.

Task 5: Application of the Design for Availability method to the performance of tradeoffs between different maintenance approaches, primarily, unscheduled maintenance and a data-driven PHM approach.

Task 6: Compute the life cycle cost and perform an ROI analysis concurrent with the application of the Design for Availability methodology. This included the following two subtasks:

a) Automated the ROI calculation and development of a stochastic ROI analysis, using the Task 2 maintenance model.

b) Performed life cycle cost and ROI analysis using the Design for Availability model.

Chapter 2: Maintenance ROI Model

This chapter[2] describes a maintenance model that can be used to assess life cycle cost tradeoffs, including the Return on Investment (ROI), associated with various maintenance approaches. This model was specifically formulated to allow PHM approaches to be evaluated and traded off. This model forms a necessary basis for verifying the design for availability methodology and performing cost assessment concurrent with the determination of parameters satisfying availability requirements. This chapter represents Tasks 1 and 2 described in Section 1.2.3

The modeling described in this chapter targets finding the optimum balance between avoiding failures and throwing away remaining useful life (RUL). Two systems, fielded and used under similar conditions, will not generally fail at exactly the same time due to differences in their manufacturing and materials, and due to differences in the environmental stress history they experience. Therefore, system reliability is generally represented as a probability distribution over time or in relation to a specific environmental stress driver. Likewise, the ability of a PHM approach to accurately predict RUL is not perfect due to sensor uncertainties, sensor gaps, sensor locations, and/or uncertainties in algorithms and models used. Practically speaking, these uncertainties make 100% failure avoidance impossible to obtain; optimal maintenance planning for systems effectively becomes a trade-off between the potentially high costs of failure and the costs of throwing away remaining system life in order to avoid failures.

[2] Portions of this chapter describe work performed prior to this dissertation, see [13, 27].

Although many applicable models for single- and multiunit maintenance planning have appeared [28, 29], the majority of the models assume that monitoring information is perfect (without uncertainty) and complete (all units are monitored identically), that is, maintenance planning can be performed with perfect knowledge as to the state of each unit. For many types of systems, and especially electronic systems, these are not good assumptions and maintenance planning, if possible at all, becomes an exercise in decision making under uncertainty with sparse data. The perfect monitoring assumption is especially problematic when the PHM approach is model-based because model-based approaches do not depend on precursors. A detailed discussion of model-based (LRU-independent) PHM methods is provided in Section 2.4. In model-based (LRU-independent) PHM methods, the PHM structure (or sensor) is independent of the LRUs, that is, the PHM structures are not coupled to a particular LRU's manufacturing or material variations. Thus, for electronics, model-based processes do not deliver any measures that correspond exactly to the state of a specific instance of a system. Previous work that treats imperfect monitoring includes [30, 31]. Perfect but partial monitoring has been previously treated [32].

This chapter describes a stochastic decision model that enables the optimal interpretation of model-based damage accumulation or data-driven precursor data and applies to failure events that appear to be random or appear to be clearly caused by defects.

2.1. Discrete Event Simulation Maintenance Planning Model

The maintenance planning model discussed here accommodates variable time to failure (TTF) of LRUs and variable RUL estimates associated with PHM approaches implemented within LRUs.[4] The model considers both single and multiple sockets[5] within a larger system. Discrete event simulation is used to follow the life of individual socket instances from the start of their field lives to the end of their operation and support.[6] Discrete event simulation allows for the modeling of a system as it evolves over time by capturing the system's changes as separate events (as opposed to continuous simulation where the system evolves as a continuous function). The evolutionary unit need not be time; it could be thermal cycles, or some other unit relevant to the particular failure mechanisms addressed by the PHM approach. Discrete event simulation has the advantage of defining the problem in terms of an intuitive basis, that is, a sequence of events, thus avoiding the need for formal specification. Discrete event simulation is widely used in maintenance and operations modeling [e.g., 33, 34 and 15] and has also previously been used to model PHM activities [35, 36 and 37].

[4] LRU refers to Line Replaceable Unit that represents the lowest-level item that is replaceable or repairable in the system.

[5] A socket is a unique instance of an installation location for an LRU. One instance of a socket occupied by an engine controller is its location on a particular engine. The socket may be occupied by a single LRU during its lifetime (if the LRU never fails), or multiple LRU if one or more LRU fail, and needs to be replaced.

[6] Alternatively, one could follow the lifetime of LRUs through their use, repair, reuse in other sockets, and disposal. CBM models generally following LRUs. The advantage of following sockets is that it enables the calculation of ROI, life-cycle cost and availability for sockets, however, the disadvantage for following sockets is that it implicitly assumes a stable population of LRUs and assumes that all LRUs returned to sockets after repair are approximately equivalent. For system integrators and sustainers, following sockets is generally preferable to following LRUs, however, for subsystem manufacturers and sustainers, following LRUs may be preferable.

The model discussed in this chapter treats all inputs to the discrete event simulation as probability distributions, that is, a stochastic analysis is used, implemented as a Monte Carlo simulation. Various maintenance interval and PHM approaches are distinguished by how sampled TTF values are used to model PHM RUL forecasting distributions. To assess PHM, relevant failure mechanisms are segregated into two types. Failure mechanisms that are random from the viewpoint of the PHM methodology are failure mechanisms that the PHM methodology is not collecting any information about (non-detection events). These failure mechanisms may be predictable but are outside the scope of the PHM methods applied. The second type refers to failure mechanisms that are predictable from the viewpoint of the PHM methodology—probability distributions can be assigned for these failure mechanisms.

For the purposes of cost model formulation, PHM approaches are categorized as (a) a fixed-schedule maintenance interval; (b) a variable maintenance interval schedule for LRU instances that is based on inputs from a data-driven (precursor to failure) methodology; and (c) a variable maintenance interval schedule for LRU instances that is based on a model-based methodology. Note, for simplicity, the model formulation is presented based on “time” to failure measured in operational hours; however, the relevant quantity could be a non-time measure.

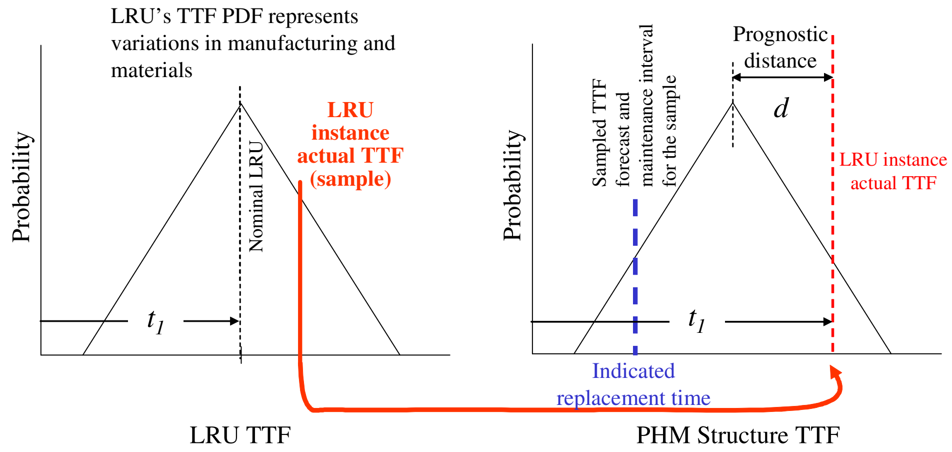


Figure 2.1. Data-driven (precursor to failure monitoring) modeling approach. Symmetric triangular distributions are chosen for illustration. Note, the LRU TTF PDF (left) and the data-driven TTF PDF (right) are not the same (they could have different shapes and sizes).

2.2. Fixed-Schedule Maintenance Interval

A fixed-schedule maintenance interval is selected that is kept constant for all instances of the LRU that occupy a socket throughout the system life cycle. In this case the LRU is replaced on a fixed interval (measured in operational hours), that is, time-based prognostics. This is analogous to mileage-based oil changes in automobiles.

2.3. Data-Driven (Precursor to Failure Monitoring) Methods

Data-driven (precursor to failure monitoring) approaches are defined as a fuse or other monitored structure that is manufactured with or within the LRUs, or as a monitored precursor variable that represents a nonreversible physical process [38].[7] Health monitoring and LRU-dependent fuses are examples of data-driven methods.

[7] In either case, the structure or parameter is coupled or correlated in some way to the manufacturing or material variations of a particular LRU.

The parameter to be determined (optimized) is prognostic distance. The prognostic distance is a measure of how long before system failure the prognostic structures or prognostic cell is expected to indicate failure (e.g., in operational hours). The data-driven methodologies forecast a unique TTF distribution for each instance of an LRU based on the instance's TTF.[8] For illustration purposes, the data-driven forecast is represented as a symmetric triangular distribution with a most likely value (mode) set to the TTF of the LRU instance minus the prognostic distance, Figure 2.1.

The data-driven distribution has a fixed width measured in the relevant environmental stress units (e.g., operational hours) representing the probability of the prognostic structure indicating the precursor to a failure. As a simple example, if the prognostic structure was a LRU-dependent fuse that was designed to fail at some prognostic distance earlier than the system it protects, then the distribution on the right side of Figure 2.1 represents the distribution of fuse failures (the TTF distribution of the fuse). The parameter to be optimized in this case is the prognostic distance assumed for the precursor to failure monitoring forecasted TTF.

The model proceeds in the following way: for each LRU TTF distribution sample (t_1) taken from the left side of Figure 2.1, a precursor to failure monitoring TTF distribution is created that is centered on the LRU TTF minus the prognostic distance ($t_1 – d$). The precursor to failure monitoring TTF distribution is then sampled, and if the precursor to failure monitoring TTF sample is less than the actual TTF of the LRU instance, the precursor to failure monitoring is deemed successful. If the precursor to failure monitoring distribution TTF sample is greater than the actual TTF

[8] In this model, all failing LRUs are assumed to be maintained via replacement or good-as-new repair; therefore, the time between failure and the time to failure are the same.

of the LRU instance, then precursor to failure monitoring was unsuccessful. If successful, a scheduled maintenance activity is performed and the timeline for the socket is incremented by the precursor to failure monitoring sampled TTF. If unsuccessful, an unscheduled maintenance activity is performed and the timeline for the socket is incremented by the actual TTF of the LRU instance. At each maintenance activity, the relevant costs are accumulated.

2.4. Model-Based (LRU-Independent) Methods

In model-based (LRU-independent) PHM methods, the PHM structure (or sensor) is independent of the LRUs, that is, the PHM structures are not coupled to a particular LRU's manufacturing or material variations. An example of a model-based method is life consumption monitoring (LCM) [39]. LCM is the process by which a history of environmental stresses (e.g., thermal, vibration) is used in conjunction with PoF models to compute damage accumulated and thereby forecast RUL. The model-based methodology forecasts a unique TTF distribution for each instance of an LRU based on its unique environmental stress history. For illustration purposes, the model-based TTF forecast is represented as a symmetric triangular distribution with a most likely value (mode) set relative to the TTF of the nominal LRU and a fixed width measured in operational hours, Figure 2.2. Other distributions may be chosen and Vichare et al. [40] have shown how this distribution may also be derived from recorded environment history. The shape and width of the model-based method distribution depend on the uncertainties associated with the sensing technologies and uncertainties in the prediction of the damage accumulated (data and model uncertainty). The variable to be optimized in this case is the safety margin assumed on the LRU-independent method forecasted TTF, that is, the length of time (e.g., in operation hours) before the LRU-independent method forecasted TTF the unit should be replaced.

The model-based method proceeds in the following way: for each LRU TTF distribution sample (left side of Figure 2.2), an LRU-independent method TTF distribution is created that is centered on the TTF of the nominal LRU minus the safety margin—right side of Figure 2.2 (note, the model-based methods only know about the nominal LRU, not about how a specific instance of an LRU varies from the nominal). The LRU-independent method TTF distribution is then sampled, and if the LRU-independent method TTF sample is less than the actual TTF of the LRU instance, then the LRU-independent method was successful (failure avoided). If the LRU-independent method TTF distribution sample is greater than the actual TTF of the LRU instance, then the LRU-independent method was unsuccessful. If successful, a scheduled maintenance activity is performed and the timeline for the socket is incremented by the LRU-independent method sampled TTF. If unsuccessful, an unscheduled maintenance activity is performed and the timeline for the socket is incremented by the actual TTF of the LRU instance.[9]

[9] LRU-independent fuses and canary devices may require replacement for each alert that they provide whether that alert is a false positive or not. After the PHM devices are removed for maintenance, to download data, or for other activities, reinstallation follows.

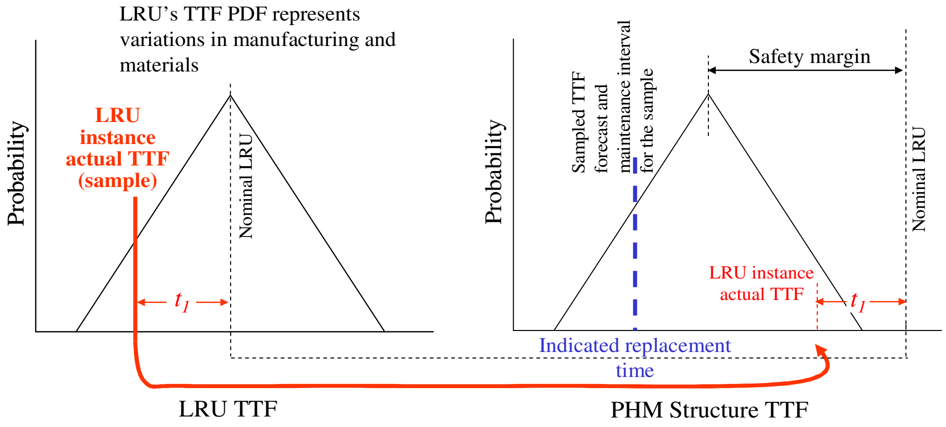

LRU's TTF PDF represents variations in manufacturing and materials
Probability
LRU instance actual TTF (sample)
Nominal LRU
t_1
LRU TTF
Probability
Sampled TTF forecast and maintenance interval for the sample
Safety margin
Nominal LRU
LRU instance actual TTF
t_1
Indicated replacement time
PHM Structure TTF

Figure 2.2. Model-based (LRU-independent) modeling approach. Symmetric triangular distributions are chosen for illustration. Note, the LRU TTF PDF (left) and the model-based method TTF PDF (right) are not the same (they could have different shapes and sizes).

In the maintenance models discussed, a random failure component may also be superimposed as discussed in [27]. The fixed-schedule maintenance, data-driven and model-based method models are implemented as stochastic simulations, in which a statistically relevant number of sockets are considered in order to construct histograms of costs, availability, and failures avoided. Again, at each maintenance activity, the relevant costs are accumulated.

The fundamental difference between the data-driven and model-based methods is that in the data-driven method the TTF distribution associated with the PHM structure (or sensor) is unique to each LRU instance, whereas in the model-based method the TTF distribution associated with the PHM structure (or sensor) is tied to the nominal LRU and is independent of any manufacturing or material variations between LRU instances.

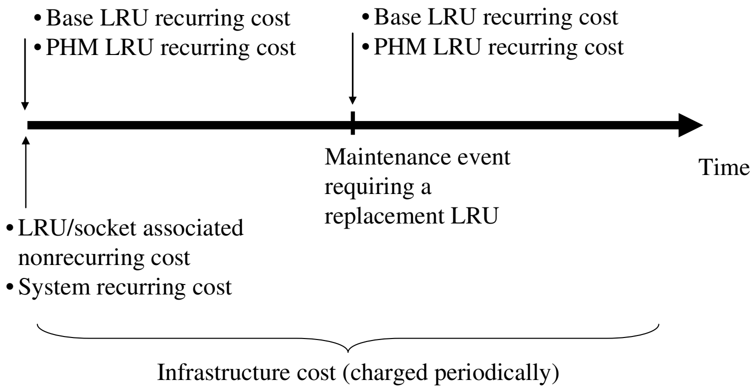

• Base LRU recurring cost
• PHM LRU recurring cost
• Base LRU recurring cost
• PHM LRU recurring cost
Maintenance event requiring a replacement LRU
Time
• LRU/socket associated nonrecurring cost
• System recurring cost
Infrastructure cost (charged periodically)

Figure 2.3. Temporal ordering of implementation cost inclusion in the discrete event simulation.

2.5. Discrete Event Simulation Implementation Details

The model follows the history of a single socket or a group of sockets from time zero to the end of support life for the system. To generate meaningful results, a statistically relevant number of sockets (or systems of sockets) are modeled and the resulting cost and other metrics are presented in the form of histograms. The scheduled and unscheduled costs computed for the sockets at each maintenance event are given by

$$C_{socket\,i} = fC_{LRU\,i} + (1-f)C_{LRU\,repair\,i} + fT_{replace\,i}V + (1-f)T_{repair\,i}V \qquad (2.1)$$

where $C_{socket\ i}$ is the life-cycle cost of socket i; $C_{LRU\ i}$ is the cost of procuring a new LRU; $C_{LRU\ repair\ i}$ is the cost of repairing an LRU in socket i; f is the fraction of maintenance events on socket i that require replacement of the LRU in socket i with a new LRU; $T_{replace\ i}$ is the time to replace the LRU in socket i; $T_{repair\ i}$ is the time to repair the LRU in socket i; and V is the value of time out of service.

Note, the values of f and V generally differ depending on whether the maintenance activity is scheduled or unscheduled.

As the discrete event simulation tracks the actions that affect a particular socket during its life cycle, the implementation costs are inserted at the appropriate locations, Figure 2.3. At the beginning of the life cycle, the non-recurring cost is applied. The recurring costs at the LRU level and at the system level are first applied here and subsequently applied at each maintenance event that requires replacement of an LRU ($C_{LRU\,i}$, as in equation (2.1)). The recurring LRU-level costs include the base cost of the LRU regardless of the maintenance approach. Discrete event simulations that compare alternative maintenance approaches to determine the ROI of PHM must include the base cost of the LRU itself without any PHM-specific hardware. If discrete event simulation is used to calculate the life-cycle cost for a socket under an unscheduled maintenance policy, then the recurring LRU-level cost is reduced to the cost of replacing or repairing an LRU upon failure. Under a policy involving PHM, the failure of an LRU results in additional costs for the hardware, assembly, and installation of the components used to perform PHM. The infrastructure costs are distributed over the course of the socket's life cycle and are charged periodically.

The model assumes that the TTF distribution represents manufacturing and material variations from LRU to LRU. The range of possible environmental stress histories that sockets may see are modeled using an environmental stress history distribution. Note, the environmental stress history distribution need not be used if the TTF distribution for the LRUs includes environmental stress variations. The environmental stress history distribution is not used with the data-driven or model-based methods. Random TTFs are characterized by a uniform distribution with a height equal to the average random failure rate per year and a width equal to the inverse of the average random failure rate.

Uncertainty, which must be propagated throughout the life-cycle simulations of systems, is present at multiple levels in the calculation of RUL. The data collected by the prognostic devices, the material inputs reliability modeling depends on, and the underlying assumptions of electronic failure behavior that are applied to produce reliability estimates may not always be accurate.

Uncertainties can be handled using different approaches; however, the most general method of handling uncertainties is to use a Monte Carlo analysis approach in which each input parameter is optionally represented as a probability distribution. The implementation of the maintenance modeling discussed in this chapter is implemented as a Monte Carlo analysis that follows a statistically relevant number of sockets over their support lives.

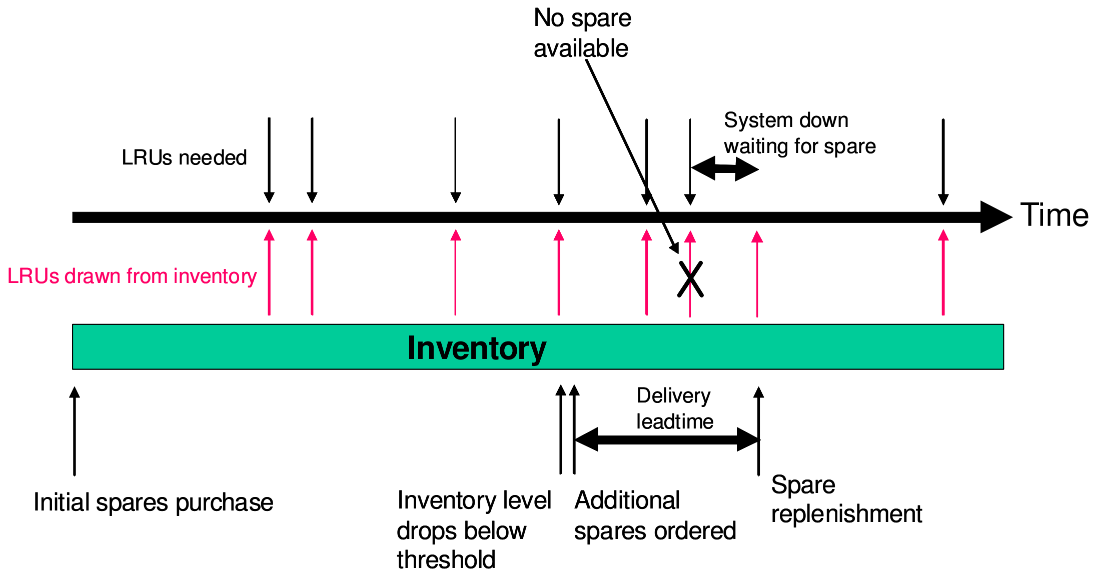

No spare available
System down waiting for spare
LRUs needed
Time
LRUs drawn from inventory
X
Inventory
Delivery leadtime
Initial spares purchase
Inventory level drops below threshold
Additional spares ordered
Spare replenishment

Figure 2.4. Spares management for a single socket.

2.6. Spares Management

The maintenance planning simulation can be performed assuming that spares can be purchased as needed, or that spares reside in an inventory. Figure 2.4 shows a graphical representation of the sparing process in this model. The spares inventory model includes the purchase of an initial quantity of spares (the purchase is assumed to happen at the start of the simulation). As the LRUs in sockets fail and require spares, they are drawn from the inventory. An inventory carrying cost is assessed per year based on the number of spares that reside in the inventory at the beginning of the year. When the number of spares in the inventory drops below a defined threshold, additional spares are automatically purchased (this is called a spare replenishment). The replenishment spares become available in the inventory for use after a lead-time. Cost of money is assessed on all spares purchases, inventory, and replenishment activities.

Only a fraction of the LRU failures require permanent spares because they cannot be repaired. Repairable LRU failures only require spares for the time during their repair.

Each socket is assumed to have its own independent inventory. In other words, I am assuming that these individual inventories could be equivalent to one or multiple large inventories that are used by the whole population of sockets. This means that each socket in the population is subject to the same spares management. This assumption is most appropriate for large populations of sockets where LRUs have the same *TTF* distributions, since on average each socket will use the same number of spares throughout its support life. This assumption doesn't hold for a population of LRUs with different *TTF* distributions that draw from the same inventories; since each socket could use on average a different number of spares throughout its support life, thus the spares management will be different for each socket.

Logistics management models that include detailed treatments of inventories and spares in the context of PHM appear in [36, 37, 41].

2.7. Operational Profile

The operational profile of systems equipped with PHM dictates how the information provided by PHM may be used to affect the maintenance and usage schedules. The effective costs associated with maintenance actions depend on when (and where) actions are indicated relative to some operational cadence. Cadences may be proscribed by business constraints, regulations, or mission requirements and may be subject to change as user requirements shift. The cadence may be best described according to a probabilistic model rather than a timeline, that is, a defined probability of a maintenance request being issued before, during, or after a mission or particular type of use. The implications of the safety margins or prognostics distances will vary with the difference in cadence to affect the timing of maintenance actions.

The operational profile is reflected in the maintenance modeling by varying the value of the parameter *V* in equation (2.1). The value of an hour out of service, *V*, is set to a specific value if the maintenance is scheduled, but if the maintenance is unscheduled, the value of *V* is given by the data in Table 2.1.

Table 2.1. Data Defining Unscheduled Maintenance Operational Profile

Maintenance Event	**Probability**	***V***
Maintenance event before mission (during preparation)	P_b	V_b
Maintenance event during mission	P_d	V_d
Maintenance event after mission (during downtime)	P_a	V_a

“Before mission” represents maintenance requirements that occur while preparing to place the system into service, that is, while loading passengers onto the aircraft for a scheduled commercial flight. “During mission” means that the maintenance requirement occurs while the system is performing a service and may result in interruption of that service, for example, making an emergency landing or abandoning a HMMWV by the side of the road during a convoy. “After mission” represents time that the system is not needed, that is, the period of time from midnight to 6:00 am when the commercial aircraft could sit idle at a gate.

When an unscheduled maintenance event occurs, a random number generator is used to determine the portion of the operational profile the event is in and the corresponding value (*V*) used in the analysis. This type of valuation in the discrete event simulation is only useful if a stochastic analysis that follows the life of a statistically relevant number of sockets is used.

2.8. Implementation Costs

Implementation costs are the costs associated with the realization of PHM in a system, the technologies and support necessary to integrate and incorporate PHM into new or existing systems. The costs of implementing PHM can be categorized as recurring, non-recurring, or infrastructural depending on the frequency, and role of the corresponding activities. The implementation cost is the cost of enabling the determination of Remaining Useful Life (RUL) for the system.

Non-recurring costs are associated with one-time only activities that typically occur at the beginning of the timeline of a PHM program, although disposal or recycling non-recurring costs would occur at the end. Non-recurring costs can be calculated on a per-LRU, per-socket, or per a group of LRU or sockets basis. The specific non-recurring cost is calculated as

$$C_{NRE} = C_{dev\_hard} + C_{dev\_soft} + C_{training} + C_{doc} + C_{\mathrm{int}} + C_{qual} \qquad (2.2)$$

Recurring costs are associated with activities that occur continuously or regularly during the PHM program. As with non-recurring costs, some of these costs can be viewed as an additional charge for each instance of a LRU, or for each socket (or for a group of LRU or sockets). The recurring cost is calculated as

$$C_{REC} = C_{hard\_add} + C_{assembly} + C_{test} + C_{install} \qquad (2.3)$$

Unlike recurring and non-recurring costs, infrastructure costs are associated with the support features and structures necessary to sustain PHM over a given activity period, and are characterized in terms of the ratio of money to a period of activity (i.e., dollars per operational hour, dollars per mission, dollars per year). The infrastructure costs are calculated as

$$C_{INF} = C_{prognostic\ maintenance} + C_{decision} + C_{retraining} + C_{data} \qquad (2.4)$$

See [42] for a detailed discussion of the various implementation cost contributions.

2.9. Return on Investment (ROI) Calculation

In general, ROI is the ratio of gain to investment. Equation (2.5) is a way of defining ROI over a system's life cycle.

$$ROI = \frac{Return - Investment}{Investment} = \frac{Avoided\ Cost}{Investment} - 1 \qquad (2.5)$$

The central ratio in equation (2.5) is the classical ROI definition, and the ratio on the right is the form of ROI that is applicable to PHM assessment. In the case of PHM, the investment includes all the costs necessary to develop, install, and support a PHM approach in a system; while the avoided cost is a quantification of the benefit realized through the use of a PHM approach. Note that not all researchers that quote ROI numbers for the application of PHM to systems define ROI in the same way; therefore, published ROI may not be directly comparable in all cases. Equation (2.5) is the standard definition used by the financial world for ROI.

Viable business cases for PHM do not necessarily require that the ROI be greater than zero. ROI > 0, implies that there is a cost benefit. In some cases, the value of PHM is not directly quantifiable in monetary terms, but is necessary in order to meet a system requirement that could not otherwise be attained, e.g., an availability requirement. However, the evaluation of ROI (whether greater than or less than zero) is still a necessary part of any business case developed for PHM [42].

For PHM, ROI must be measured relative to whatever methodology is currently used to manage the system. For electronic systems, a common management approach is unscheduled maintenance. Following an unscheduled maintenance policy, systems are operated until failure, and are then repaired or replaced. Applying equation (2.5) to measure ROI relative to unscheduled maintenance gives

$$ROI = \frac{(C_{us} - I_{us}) - (C_{PHM} - I_{PHM})}{(I_{PHM} - I_{us})} - 1 \qquad (2.6)$$

In my case, I define I_{us} = 0, i.e., the investment cost in unscheduled maintenance is indexed to zero by definition. This does not imply that the cost of performing maintenance in the unscheduled case is zero (the cost of performing maintenance is part of C_{us}), but reflects that a maintenance approach relying purely on unscheduled maintenance makes no investment in PHM. Setting I_{us} = 0, then equation (2.6) becomes

$$ROI = \frac{C_{us} - (C_{PHM} - I_{PHM})}{I_{PHM}} - 1 \qquad (2.7)$$

Equation (2.7) measures ROI of a PHM approach relative to unscheduled maintenance; if C_{PHM} is equal to C_{us}, then ROI equals 0, the breakeven point.[10]

[10] Equation (2.7) is only valid for comparison of ROI to unscheduled maintenance, which is a convenient well defined solution to measure ROI. Using equation (2.7), one can compare the relative ROI of multiple PHM approaches measured from unscheduled maintenance; however, the ROI of one PHM approach relative to another is not given by the difference between their ROI relative to unscheduled maintenance. To evaluate ROI relative to a baseline other than unscheduled maintenance, appropriate values of Avoided Cost and Investment must be substituted into equation (2.5).

The investment cost is the effective cost per socket of implementing PHM, and then using the knowledge it creates to guide maintenance actions, and planning. The PHM investment cost is calculated as

$$I_{PHM} = C_{NRE} + C_{REC} + C_{INF} \qquad (2.8)$$

The costs of false alarm resolution, procurement of a different quantity of LRU than the number required by an unscheduled maintenance approach, and maintenance costs that differ from unscheduled maintenance are not included in the investment cost because they are the result of the investment, and are reflected in C_{PHM}. C_{PHM} must also include the cost of money differences associated with purchasing LRU at maintenance events between unscheduled maintenance, and a PHM approach; i.e., even if both approaches end up purchasing the same number of replacement LRU for a socket, they may purchase them at different points in time resulting in different effective costs if the discount rate is non-zero. If replacement LRU are drawn from an inventory of spares (as opposed to purchased as needed), then there may be no cost of money impact on ROI associated with the procurement of spares.

The ROI in equation (2.7) can be calculated statically using values of C_{us}, C_{PHM}, and I_{PHM} that are averaged over an entire population of sockets. However, in reality, a population of sockets will result in a distribution of ROI (every socket potentially having a different ROI). To calculate the distribution of ROI, each member of the population has to be independently tracked through its lifetime assuming first an unscheduled maintenance policy, and then assuming a PHM maintenance approach (using identical samples from the distributions that represent the member’s characteristics and maintenance costs in a Monte Carlo analysis). In this manner, a separate ROI is calculated for each member of the population. When the process is repeated on an entire population of sockets, a histogram of ROI is generated from which business case parameters can be extracted.

2.10. Summary

This chapter provided an overview of the base maintenance model used in this dissertation. Additional details on the formulation of the model can be found in [27] and [42]. Applications of this model appear in [27, 42, and 43]. This model has also been released as a software tool that is supported by CALCE for members of the CALCE PHM Consortium [44].

This model represents a forward discrete event simulator that can determine life cycle cost, ROI, and availability (as an output not an input). In this dissertation, this simulation will be used: (a) to order events for the design for availability solution; (b) to generate availability outputs for verification of the design for availability; and (c) to concurrently calculate life cycle cost and ROI with the design for availability activities.

Chapter 3: Design for Availability

This chapter describes the formulation of a general design for availability approach that uses an availability requirement (input) to generate system parameters. Parameters that depend on either uptime or downtime (but not both) and parameters that depend on both uptime and downtime are addressed. A detailed description of each step of the formulation and the application of the methodology is provided (for both types of system parameters).

3.1. Domain of Applicability – An Example Design for Availability Problem Statement

While availability is critical to the many different types of systems mentioned in Chapter 1, in this section I will briefly describe one particular design example so as to better define design for availability's domain of applicability and to address how it could be incorporated into a design problem statement.

Consider a multifunctional display (MFD) that is used in the cockpits of Boeing 737 aircraft. If the MFD is non-operational, the aircraft cannot takeoff, resulting in flight delays and/or cancelations, both of which cost an airline money. An airline decides that they wish to impose a 97% mean availability requirement on the MFD units across their entire fleet of 737s over some prescribed period of time subject to some range of operational conditions and operational schedule. Presumably this availability requirement is arrived at based on a flow down of other business requirements and targets. The airline's availability requirement (and other operational assumptions) represents a design problem that may be imposed on: the manufacturer of the MFD, the airline's own procurement and support organizations, or both.

First, consider the manufacturer of the MFD. One design attribute that the manufacturer can affect is the reliability of the MFD, which is actually represented by a composite of failure distributions corresponding to relevant failure mechanisms. Using a "design for availability" methodology, the manufacturer could use the 97% mean availability requirement coupled with the range of environment stress conditions that they expect the MFDs to encounter during operation to determine the required reliability distributions corresponding to one or more of the relevant failure mechanisms. The manufacturer could then modify the material selection, component selection, and/or design rules in order to meet the required reliability.

Alternatively, the airline could use the mean availability requirement to enable selection between competing suppliers of the MFD unit. Obviously, the availability requirement could be used to determine the necessary composite reliability of a MFD unit and this could be used as a selection criteria. However, design of availability-centric systems is not all about designing the object, it's just as much about designing how you will support the object. To this end, consider the Inventory Lead Time (*ILT*). The *ILT* is the amount of time it takes to receive spares after they are ordered. The airline could impose their mean availability requirement along with the spares inventory (threshold for replenishment, inventory holding costs, maximum number of spares held, etc.) to determine a maximum allowable *ILT* for the MFD unit and then select the supplier of the MFD based on the supplier's capability to provide spares that satisfy the required *ILT*.

3.2. Introduction to Design for Availability

Most availability and life cycle cost predictions used during the design and support of real systems are performed using discrete event simulators, e.g., Chapter 2 and [15]. In general, discrete event simulators order the failure and maintenance events for a system temporally, and the times associated with the failure and maintenance events can be readily accumulated to estimate availability. Thus, it is straightforward for the simulation to compute the availability based on a particular sequence of failures, logistics and maintenance events.

The requirements for PHM are based on a set of reliability, operational profile, and logistics parameters, which in turn are dependent on availability objectives. Thus, for the same availability requirement different sets of system parameters could be determined based on the selected PHM sustainment approach; i.e., the prediction of system parameters (to meet availability requirements) is dependent on the applied PHM sustainment approach. However, in general this prediction is a stochastic reverse simulation problem.

Availability requirements can be satisfied by running discrete event simulators in the forward direction (forward in time) for many permutations of the system parameters and then selecting the inputs that generate the required availability output, e.g., [45, 46]. These "brute force" search-based approaches are computationally impractical for real problems (particularly for real-time problems), are unable to deal with general uncertainties, and can't accommodate an availability requirement that is represented as a probability distribution. There have also been attempts to perform reverse simulation (run discrete event simulators backwards in time) [47, 48], but this has only been demonstrated on extremely simple problems with no applicability to the real world systems. While determining the availability that results from a sequence of events is straightforward, determining the events that result in a desired availability is not, and has not in general been done. Alternatively stated, availability is straightforward to predict based on the system's reliability, operation, sparing, etc., however, the general prediction of the system parameters to meet a required availability ("design for availability") has never been done and it is the goal of this dissertation

To meet an availability requirement one can change the reliability, logistics or operation of a system or a process. For example, I may keep the same reliability and operational profile of the system, and change its logistics management (e.g., reduce the maintenance downtime by stocking spare parts in more locations). Similarly, reliability or operational characteristics (or some combination of them) could be modified to meet an availability requirement. Deriving the appropriate system parameters for a specific availability requirement, will ease logistics management, avoid availability cost penalties, and predict the required reliability of the system.

The goal of this work is to reverse the problem setup; this means, instead of solving for availability for a specific set of system parameters, I will solve for system parameters for a specific availability. The design for availability model could be used to generate system reliability, operation, sparing, etc., for a specific availability, i.e., for a specific uptime (time that the system is up and running as requested) and downtime (time that the system is down undergoing a repair or waiting for spares). The approach presented here is not based on running backwards discrete event

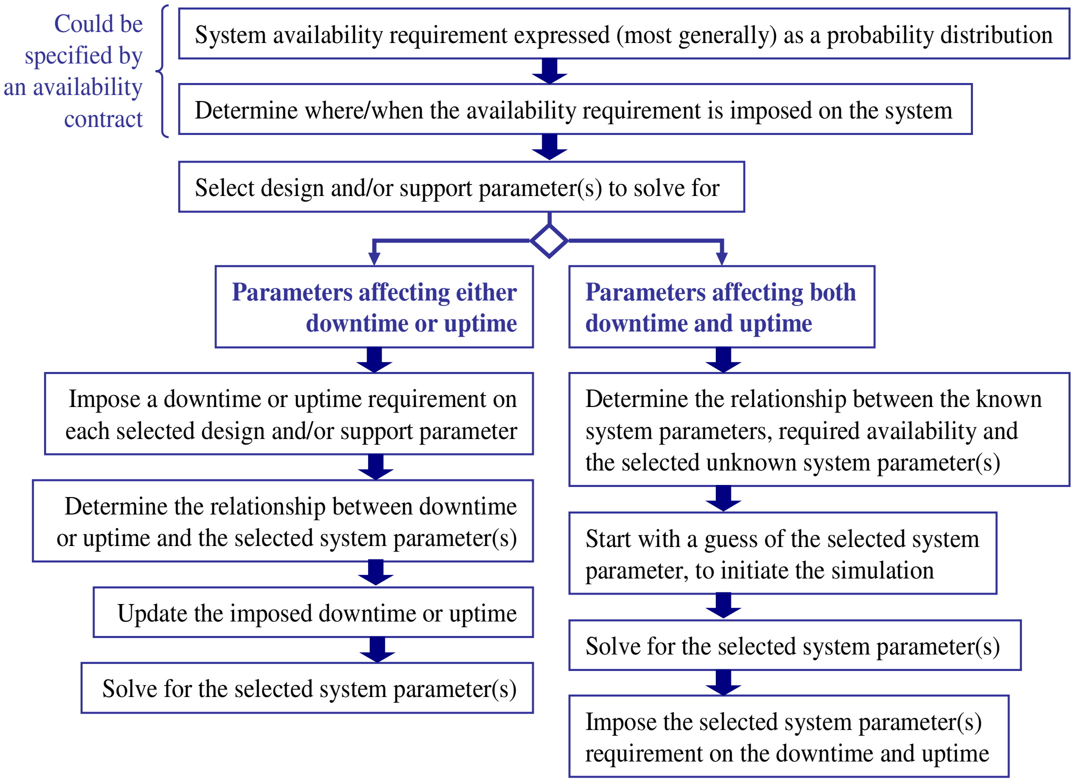

Could be specified by an availability contract
System availability requirement expressed (most generally) as a probability distribution
Determine where/when the availability requirement is imposed on the system
Select design and/or support parameter(s) to solve for
Parameters affecting either downtime or uptime
Parameters affecting both downtime and uptime
Impose a downtime or uptime requirement on each selected design and/or support parameter
Determine the relationship between downtime or uptime and the selected system parameter(s)
Update the imposed downtime or uptime
Solve for the selected system parameter(s)
Determine the relationship between the known system parameters, required availability and the selected unknown system parameter(s)
Start with a guess of the selected system parameter, to initiate the simulation
Solve for the selected system parameter(s)
Impose the selected system parameter(s) requirement on the downtime and uptime

Figure 3.1. Design for Availability Methodology.

simulation, i.e., the model runs a forward discrete event simulation, but, instead of using system parameters to compute uptimes or downtimes, the new model uses the specified availability requirement (input) to impose the appropriate uptimes and downtimes, and solve for the selected unknown system parameters (output).

3.3. Design for Availability Approach

Figure 3.1 shows the steps to formulate and execute the design for availability solution. Details for each step of the process of determining the selected unknown system parameters concurrently (for a specific availability requirement) are discussed in the following sections.

Interpreting the Availability Requirement

The design for availability methodology is applicable to any type of input of the availability requirement (e.g., single value, probability distribution, range of values, etc.). A realistic availability requirement is generally expressed as a probability distribution. Since, even when a contract specifies the availability requirement as a single value, the interpretation of this single value either leads to considering the average availability of a population of systems, i.e., the average of a distribution; or the single value is the minimum availability of all system instances within the population. These interpretations are consistent with the fact that the reliability of the product or system is represented as a probability distribution (or, more accurately a set of probability distributions each corresponding to a different relevant failure mechanism); thus using a logistics management plan that is common across the population, each system instance will have a different availability value depending on the failure dates of the subsystem instances that occupy it and the operational profile variations.

Determine Where/When the Availability Requirement is Imposed

To generalize the design for availability model, I adopt a conservative approach by fulfilling an availability requirement at all times during the entire support life. In other words, the model satisfies any availability contract requirement, regardless of the availability evaluation time intervals specified by the contract terms. However, if needed, the model could be adjusted to evaluate the availability requirement only at the contract's defined times (which would be less conservative).

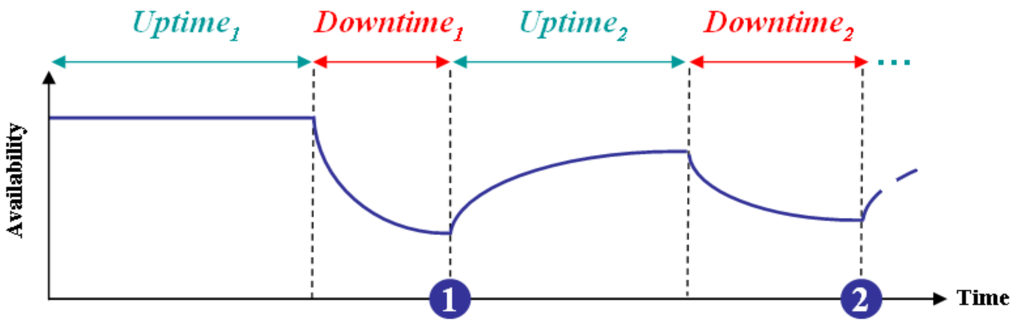


Figure 3.2. Availability variations as a function of time.

For the remainder of this discussion I will assume that the availability requirement implies that the operational availability (A_o) should not drop below the availability requirement value at any time during the entire support life.

By analyzing the A_o variations based on equation (1.1), the A_o keeps decreasing during downtimes and increasing during uptimes. In other words, A_o reaches its local minimum values at the end of every downtime (e.g., points 1 and 2 in Figure 3.2). Thus, if the availability requirement is satisfied at the end of every downtime (minimum A_o values satisfy the requirement), it will be satisfied at all times during the support life of the system. Therefore my approach is to impose the availability requirement at the end of every downtime.

Select the Design and/or Support Parameter to Solve for

Different values of a system parameter could generate different downtimes and/or uptimes, resulting in different availability values. For example, to meet a specific availability requirement, the reliability of the system could be improved, and/or the logistics management could be modified. This means, once the availability requirement is defined, a decision has to be made upfront regarding which system parameter the system manufacturer, provider or user is willing to change to meet the availability requirement. Once the system parameter that will be modified to meet the availability requirement is selected, the availability requirement will be used to solve for it, i.e., the availability requirement is used as an input to the model, and the selected unknown system parameter is one of the resulting outputs of the model.

Determine the Type of System Parameter

In the context of design for availability, there exist two distinct types of system parameters: parameters affecting either uptime or downtime (not both) and parameters concurrently affecting both uptime and downtime. As shown in Figure 3.1 the steps to formulate and execute the design for availability solution are different for each type of system parameter.

In the case of parameters affecting either uptime or downtime (not both), one of the two quantities (uptime or downtime) is known and can be determined from the known system parameters, while the other quantity is unknown. In other words, a change in the value of the selected unknown system parameter produces a change in only one of the two quantities (either uptime or downtime), while the other quantity is exclusively dependent on the other known system parameters. For example, if uptime is the known quantity (determined from the known system parameters), while the downtime is the unknown quantity that is imposed based on the required availability and the system generated uptimes. Then, the selected unknown system parameter solely depends on the downtime and is computed based on the imposed downtime.

However, in the case of parameters concurrently affecting both uptime and downtime, a change in the value of this type of parameter could produce a change in both uptime and downtime. Both quantities (uptime and downtime) are dependent on this type of parameters (reliability is a prime example). When one of these system parameters is unknown, then both uptime and downtime are unknown. Therefore, I cannot impose exclusively a downtime or uptime requirement as described in the previous paragraph. This means, a relationship between the known system parameters, required availability and the selected unknown system parameter needs to be defined, to solve for this type of system parameter.

Parameters Affecting either Uptime or Downtime (not both)

Impose an Uptime or Downtime requirement on the selected system parameter

The A_o is a function of accumulated uptimes and downtimes. Therefore, imposing an availability requirement means either imposing an uptime requirement while the downtime is automatically generated by the system parameters; or imposing a downtime requirement while the uptime is automatically generated by the system parameters. Note, for this type of system parameters (i.e., parameters affecting either uptime or downtime) either an uptime or a downtime requirement is imposed, not both. In both cases, the imposed downtimes or uptimes are computed at defined times or events as a function of the required A_o.

Assuming that to satisfy an availability requirement I need to impose downtimes. In this case, I can ignore all uptimes and only focus on the required downtimes to fulfill the availability requirement; and then determine a relationship that allows us to derive the system parameters that produce these downtimes. In this case, by generating the appropriate downtimes, only the system parameters that are responsible for generating the downtimes could be computed as outputs; other system parameters that are not involved in generating downtimes should be fixed, i.e., used only as inputs. Equations (3.1) and (3.2), which are derived from equation (1.1), express the relationships that are used to impose the first downtime (DT_1) and a k^{th} downtime (DT_k), respectively.

$$DT_1 = \frac{(UT_1)(1 - Availability)}{Availabi\,l\,ity} \qquad (3.1)$$

$$DT_k = \frac{\sum_{j=1}^{k} UT_j \;\; (1 - Availability)}{Availability} - \sum_{j=1}^{k-1} DT_j \qquad (3.2)$$

where *Availability* represents the specified availability requirement, and UT_j corresponds to the j^{th} uptime.

Basically, the criteria of imposing either an uptime or downtime requirement is based on the unknown system parameter that I desire to determine to fulfill a specific availability requirement. For example, if the uptime remains constant while varying a selected unknown system parameter, meaning that the uptime is independent of this unknown parameter; while, the downtime values are changing. Then I must impose the downtime to meet the availability requirement; and vice versa.

Determine the Relationship between Uptime or Downtime and the Selected System Parameter

Assume that the set of system parameter that I am interested in computing to meet the availability requirement is explicitly related to the downtimes (based on the criteria mentioned in the previous section), instead of the uptimes. Thus, I want to impose the downtime requirement. Then, I need to establish a relationship between the unknown set of system parameter and the imposed downtimes. Based on this relationship, the missing system parameters could be computed and updated as soon as the downtimes are determined.

In this case, a relationship between the selected unknown system parameters and the required availability needs to be determined in order to solve for the unknown system parameter. Basically, I have two unknown quantities: the unknown system parameter, and the downtime. Therefore, I need to establish two relationships to be able to solve for the two unknown quantities. Since availability is by definition a function of uptime and downtime; therefore I can solve for downtime as a function of the required availability and uptime, where uptime is a known quantity generated through the known systems parameters. This generates the first relationship; which is used to solve for the downtime. Also, since these types of system parameters explicitly affect the downtime, then the selected unknown system parameter could be expressed as a function of downtime; hence the second relationship; which is used to solve for the unknown system parameter. To summarize, the downtime is imposed through the availability requirement, then the imposed downtime quantity is used to solve for the selected unknown system parameter.

Defining a relationship between the selected system parameter and the downtime, allows the inclusion of the availability requirement into the computation of the selected system parameter. Notice, that in this section, for the purpose of illustration, I have assumed that the selected unknown system parameter affects the downtime; a similar approach could be adopted if the selected system parameter affects the uptime (instead of downtime).

Update the Imposed Uptime or Downtime

In this section, for demonstration purposes, I will assume that the selected unknown system parameter is explicitly dependent on the required downtime to satisfy the availability requirement, while the rest of the system parameters are given and responsible for fully generating the uptime. However, before the end of this section I will provide an analogy for fulfilling the availability requirement by imposing the required uptime (instead of downtime), where the selected unknown system parameter is explicitly dependent on the uptime.

The model uses an availability requirement (input) to solve for a specific system parameter (output). My goal is to derive a unique system parameter value for a given availability requirement. Therefore, the model imposes the appropriate downtimes that satisfy the availability requirement; then, the system parameter is derived based on each imposed downtime. The procedure of defining the relationship between system parameter and required downtimes is discussed in Section 3.5.2. However, the challenge here is the fact that availability is not determined by a single downtime value, but rather a sequence of downtime values that are not necessarily

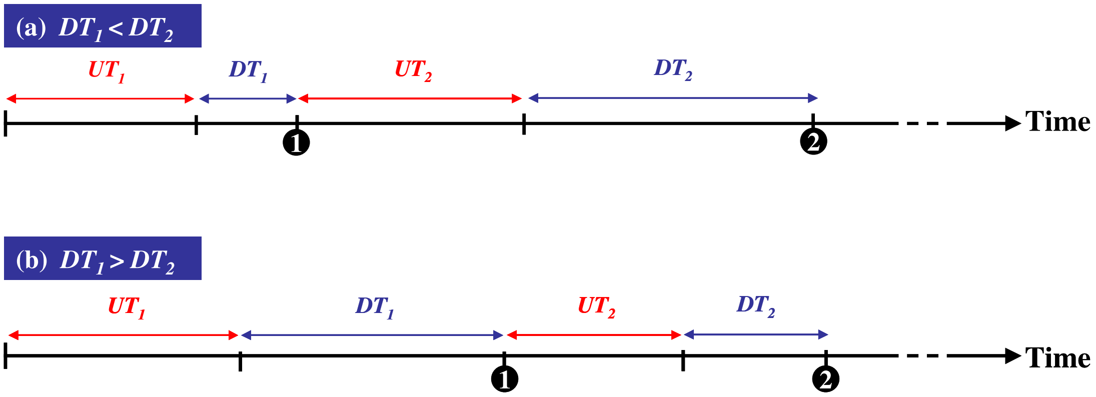


Figure 3.3. Scenarios of downtimes requirement.

identical; each resulting in different computed values for the system parameter. As a result, by the end of the simulation I could generate multiple values for the same system parameter with no way to determine which value to use to fulfill the availability requirement.

In the simplest case, if all downtimes were identical, the same value for the system parameter would have been generated at the conclusion of every downtime. To achieve this, I wish to select a single downtime value that is the maximum allowable downtime to meet a specific availability requirement, and then use this quantity as a constant downtime value that will fulfill the availability requirement at every point throughout the entire support life. To derive the maximum allowable downtime value, I have explored two scenarios.

The first scenario is illustrated in Figure 3.3a, where the first imposed downtime (DT_1) duration is shorter than the second imposed downtime (DT_2), where both downtimes have been imposed based on the same availability requirement. In this case, averaging the two downtimes would generate an average downtime ($DT_{Average}$) that is larger than DT_1, thus the availability requirement will not be fulfilled at the end of DT_1, equation (3.3); i.e., in this situation the maximum allowable downtime duration that the system can accommodate without failing to satisfy the availability requirement is constrained by the value of DT_1. Therefore, DT_2 value should be substituted for DT_1.

$$\frac{UT_1}{UT_1 + DT_{Average}} < \frac{UT_1}{UT_1 + DT_1} \qquad (3.3)$$

The second scenario is illustrated in Figure 3.3b, where DT_1 is larger than DT_2. In this case, averaging the two downtimes would generate a $\mathrm{DT}_{\mathrm{Average}}$ smaller than DT_1, thus the availability requirement will be fulfilled at the end of DT_1, equation (3.4). Also, notice that the availability requirement at any specific time includes all accumulated previous downtimes. Thus, when using the average value at the end of DT_2, the availability requirement will still be satisfied, equation (3.5), since the accumulated averages are just the accumulated downtimes, i.e., summation of DT_1 and DT_2. Therefore, in this situation the maximum allowable downtime duration that the system can accommodate without negating the availability requirement is the average downtime.

$$\frac{UT_1}{UT_1 + DT_{Average}} > \frac{UT_1}{UT_1 + DT_1} \qquad (3.4)$$

$$\frac{UT_1 + UT_2}{UT_1 + UT_2 + DT_{Average} + DT_{Average}} = \frac{UT_1 + UT_2}{UT_1 + UT_2 + DT_1 + DT_2} \qquad (3.5)$$

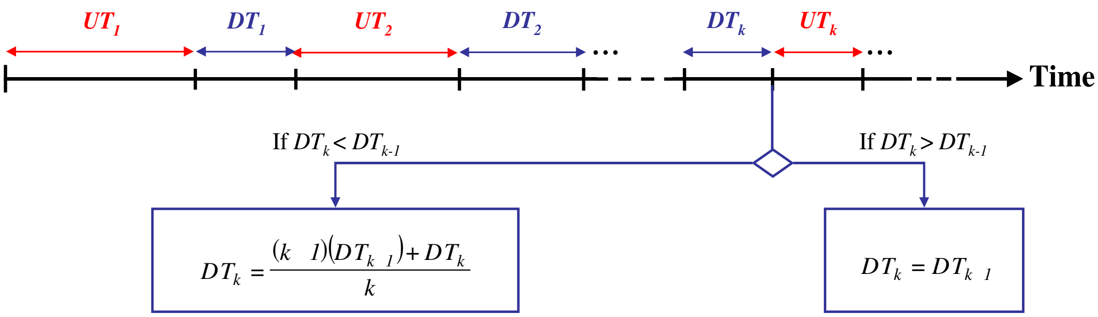


Figure 3.4. General case of updating downtime requirement.

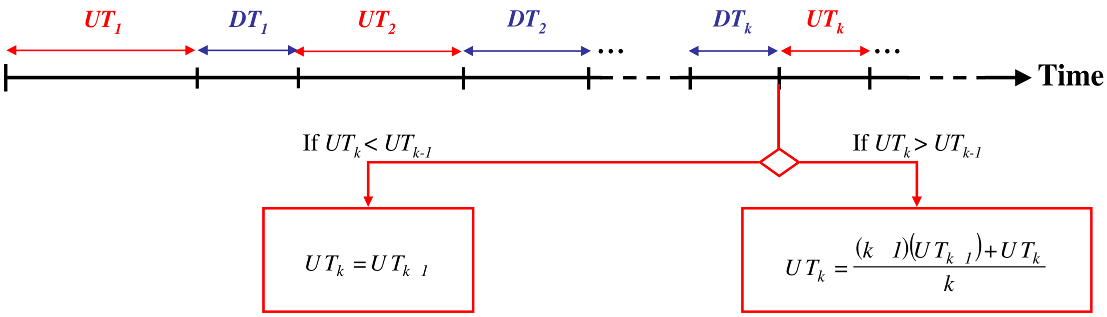


Figure 3.5. General case of updating uptime requirement.

Figure 3.4 summarizes both cases in one general case. The model imposes the required downtime to meet the availability requirement, and then it evaluates the current downtime with respect to the previous one. If the currently imposed downtime is larger than the previous one, then the model substitutes the current downtime value for the previous one. But if the currently imposed downtime is shorter than the previous one, then the model averages the current downtime value with all previous ones. The goal of this procedure is to generate one unique value of the maximum allowable downtime that meets the availability requirement. Note, during this procedure, the unknown system parameter that is determined based on the downtime requirement, gets updated as soon as the downtime values are updated.

Note, if the unknown system parameters are explicitly generating the uptime, instead of the downtime, then, by analogy, I can use a similar procedure to impose and update the uptimes (Figure 3.5) to derive one unique value of the system parameters. In this case, I will derive the minimum allowable uptime that meets the availability requirement; then use the derived quantity to compute the corresponding system parameters.

Solve for the Selected System Parameter

Once the final value of the downtime (or uptime) is imposed and updated, the selected unknown system parameter is solved for, using the relationship defined in section 3.5.2. Basically, this system parameter value was derived based on the specified availability requirement.

Finally, the design for availability model uses this final value of the selected system parameter that is necessary to meet the A_o requirement, to compute other quantities of interest (e.g., life cycle cost, investment cost, avoided failures, etc.).

Parameters Affecting both Uptime and Downtime

Determine the relationship between the known system parameters, required availability and the selected unknown system parameter

When the selected unknown system parameter concurrently affects downtime and uptime, a relationship between the known system parameters, required availability and the selected unknown system parameter needs to be determined in order to solve for the unknown system parameter.

In this case, I have three unknown quantities: 1) the unknown system parameter, 2) the downtime and 3) the uptime. Therefore, I need to establish three relationships to be able to solve for the three unknown quantities. Since this type of system parameters explicitly affects the uptime, therefore, the uptime could be expressed as a function of the unknown system parameter; this generates the first relationship. Similarly, the unknown system parameter explicitly affects the downtime, thus, downtime could be expressed as a function of the unknown system parameter; generating the second relationship. Also, availability is by definition a function of uptime and downtime; hence the third relationship. At this point, I have defined three relationships, with three unknowns (the unknown system parameter, the downtime and the uptime), thus I can solve for the unknown system parameter, uptime and downtime.

Start with a guess of the selected system parameter, to initiate the simulation

In general, a closed-form analytical solution cannot be determined when solving for the unknown system parameter as a function of known quantities (availability requirement and other known system parameters), since the sequences of the accumulated event outcomes are only generated in real simulation time. Also, when modeling real complex systems, probabilistic models are usually used where quantities include uncertainties (probability distributions).

The event outcomes associated with the sampled values can only be accumulated by sampling the known quantities in real simulation time. Basically, an event outcome generated by the same known system parameter is not generally repeated (i.e., it does not generally reoccur in an identical form at a regular interval). Each sample of the same quantity, i.e., system parameter, could result in a different event outcome, producing a different sequence of events and results for every system instance.

As a result of the situation described above, a conservative guess of the initial value of the selected unknown system parameter is required in order to launch the simulation, i.e., launch the sampling of the known quantities and accumulate the events outcomes. However, this guess is only used to initiate the simulation; it does not affect the final results of the analysis.

Solve for the selected system parameter

The model uses the initial guessed value of the selected unknown system parameter to generate the first uptime and downtime values. Then, the unknown system parameter value is computed and updated, at the end of the first downtime, while accounting for the accumulated events type and duration. The same process is repeated at the end of every downtime. Basically, the selected unknown system parameter is solved for or imposed using the known system parameters and the availability requirement.

Impose the selected system parameter requirement on the uptime and downtime

The computed system parameter value is used to compute and update the uptime and downtime values. Notice that the availability requirement was imposed through the selected system parameter, then the requirement has been transferred through the selected system parameter to impose the uptime and downtime values that are necessary to meet the availability requirement. Once all quantities are computed and updated, the process continues forward in time to the next event. The same computational process is performed at the end of every downtime, until the timeline reaches the end of the system's support life.

It is important to note that the described process is not iterative. Updating the unknown system parameter once at the end of every downtime is not the same as using multiple values of the unknown system parameter and continually iterating the entire process until the availability requirement is met. Because it is not iterative, it has the following advantages: computationally simple and straightforward, an exact solution could be determined, and a real-time assessment could be performed.

Finally, the model uses the updated selected system parameter, uptimes, and downtimes to compute other quantities of interest (e.g., life cycle cost, avoided failures, etc.).

Chapter 4: Application of the Methodology

The design for availability methodology described in Chapter 3 has been implemented within the maintenance ROI model described in Chapter 2, for demonstration and testing (i.e., verification). This chapter presents the application of the methodology to a case study example, to determine the necessary system parameters to meet a specific availability requirement. First, a derivation of the maximum allowable inventory lead time is presented as an example of determining system parameters affecting either uptime or downtime. Then a derivation of the minimum allowable system reliability is presented as an example of determining system parameters affecting both uptime and downtime. For both examples, a cost analysis of the system management using an unscheduled maintenance policy and a data-driven PHM approach is provided, this includes return on investment (ROI) and life cycle cost analysis.

The maintenance ROI model, i.e., PHM ROI model, described in Chapter 2 is a discrete-event simulation that follows a population of sockets (a socket is an instance of an installation location for an LRU) through their lifetime from first line replaceable unit (LRU) installation in the socket to retirement of the socket. This is implemented as a Monte Carlo simulation. The simulation is a stochastic simulation of a timeline where specific events are added to the timeline and the resulting event order and timing can be used to analyze throughput, cost, availability (as an output), etc.

The prediction of the remaining useful life (RUL) is determined by the sampling of both the time-to-failure (TTF) values and the distributions that are used to model the effectiveness of a particular PHM approach. The sampling of the TTF values is defined differently for each PHM sustainment approach (e.g., data-driven, model-based, fixed interval scheduled maintenance and unscheduled maintenance) – see Chapter 2. The PHM ROI model includes the modeling of other quantities as well (e.g., operational profile, false positives, cost of money, inventory management, etc.).

4.1. Case Study Data Inputs

This section provides model inputs and assumptions that are used for the case study examples presented in this chapter. The LRU used in this example is an avionics multifunction display (MFD). The implementation costs are summarized in Table 4.1. The discount rate on money used is 0.07.

Table 4.1. Implementation Costs

Frequency	Type	Value
Recurring Costs	Base cost of an LRU (without PHM)	$25,000 per LRU
Recurring Costs	Recurring PHM cost	$155 per LRU $90 per socket ($C_{REC}$)
Recurring Costs	Annual Infrastructure	$450 per socket ($C_{INF}$)
Non-Recurring Engineering	PHM cost	$700 per LRU ($C_{NRE}$)

The cost per hour out of service is $500 for scheduled maintenance and $5092 for unscheduled maintenance assuming during mission failures. However, I assume that if the multifunction display (MFD) is not functional and the inventory is out of spares, thus the aircraft is grounded for more than 24 hours waiting for spares replenishment; then the value of an hour out of service drops to 10% of the cost of the original aircraft being out of service. The operational profile is summarized in Table 4.2 [42, 49], and a 20 years support life was chosen based on [50].

Table 4.2. Operational Profile

Factor	Multiplier	Total
Support life: 20 years	2,429 flights per year	48,580 flights over support life
7 flights per day	125 minutes per flight	875 minutes in flight per day
45 minutes turnaround between flights [51]	6 preparation periods per day (between flights)	270 minutes between flights per day

4.2. Use of Design for Availability to Determine System Parameters Affecting either Uptime or Downtime

Demonstration and Verification of the Methodology: Logistics (Inventory) Parameters

In this section the design for availability methodology will be demonstrated for the derivation of the first type of system parameters, i.e., system parameters affecting either uptime or downtime (not both). An example for a logistics parameter derivation is presented. The objective in this case study example is to determine the appropriate spares replenishment lead time, i.e., inventory lead time (*ILT*), to fulfill a specific availability requirement. In order to use this demonstration as a qualitative verification of the methodology, I will perform the following steps:

- Using the availability distribution requirement as an input to the design for availability model, determine the required *ILT* distribution (output).
- Use the generated *ILT* distribution as an input to the existing PHM ROI simulation (described in the introduction to this section) to predict an availability distribution as an output.

- Compare the availability distribution input requirement to the availability distribution determined as an output.

Notice that the first step is sufficient to achieve the design for availability task, since the *ILT* will be determined for a specific contract availability requirement. The second and third steps are for verification of the methodology.

A detailed description of all of the case study inputs is provided in Section 4.1, including LRU description, implementation and maintenance costs, and operational profile.

The reliability information and inventory management parameters are provided in Figure 4.1 and Table 4.3, respectively. Table 4.3 summarizes the spares inventory (per socket) assumptions that are used for this specific example. Also, note that the spares carrying costs are incorporated into the LRU recurring costs. Figure 4.1 shows the assumed reliability for this case example, i.e., time-to-failure (TTF), of the LRU based on [43] and [52].

Table 4.3. Spares Inventory

Factor	Quantity
Initial spares purchased for each socket	5
Threshold for spare replenishment	≤ 1 spares in the inventory per socket
Number of spares to purchase per socket at replenishment	4
Spare replenishment lead time	*Solved for in this section case study*
Spares carrying cost	10% of the beginning of year inventory value per year
Billing due date when ordering additional spares	2 years from purchase date

To determine the appropriate *ILT* to ensure meeting a specific availability requirement I need to define a relationship between the downtime requirement and *ILT*. In this example, the *ILT* is the unknown system parameter, where *ILT* is the amount of time it takes to receive replenishment spares (*RS*) when additional spares are ordered at the inventory spares threshold (*ST*) value (i.e., minimum quantity of held spares); this example case assumes that the inventory downtime (when the inventory runs out of spares, and the system is down waiting for replenishment spares) is larger than any concurrent maintenance downtime. Also, once the spares are received, the part can be immediately installed in the system. The *ILT* requires imposing a downtime requirement since varying the *ILT* only affects the downtime values, i.e., how long the system will be down waiting for spares to be replenished.

In this case (Figure 4.2), the decision to impose the inventory downtime (*IDT*) to meet the availability requirement, instead of imposing uptime, is based on the fact that the unknown system parameter, i.e., *ILT*, is only dependent on the downtimes; and it is independent of the uptimes. Varying the *ILT* generates different *IDT* values; however the uptime values remain constant since they are only a function of the inventory spares threshold (*ST*) and maintenance downtimes (*MDT*). Where the

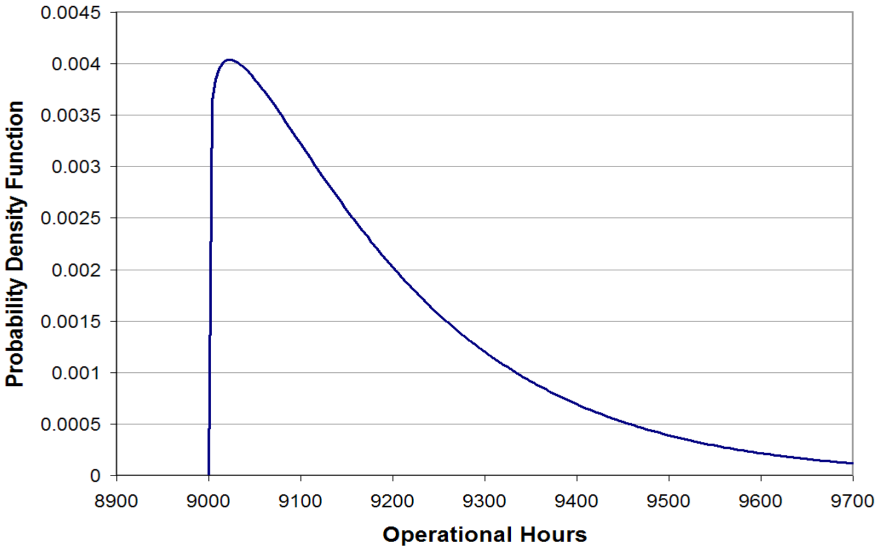


Figure 4.1. Weibull distribution of TTFs (β=1.1, η=200 and γ=9000).

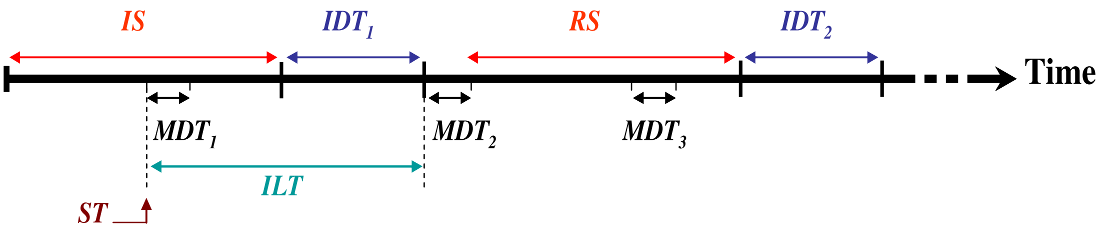


Figure 4.2. Implication of the inventory model parameters on the timeline.

inventory held spares is a function of the quantity of initial spares (*IS*) and quantity of *RS*. The ILT only defines the start of the next uptime, but it does not define the uptime duration.

Because, the *IDT* is purely a function of *ILT* and spares threshold time (*STT*), where *STT* is the corresponding period of time to use all remaining spares; the *IDT* only depends on how low the inventory level is allowed to drop before ordering additional spares and how long it will take to receive those spares,

$$ILT = STT + IDT_1 \qquad (4.1)$$

For this example I assume that the *MDT* are given and cannot be modified, i.e., the maintenance lead time, replacement time and repair time are already specified as inputs. To fulfill the availability requirement at the end of the first *IDT*; IDT_1 should satisfy the A_o requirement (as defined in equation (1.1); where $IS\text{-}MDT_1$ corresponds to the accumulated uptime and $IS+IDT_1$ corresponds to the sum of the accumulated uptime and downtime), thus satisfy the following equation:

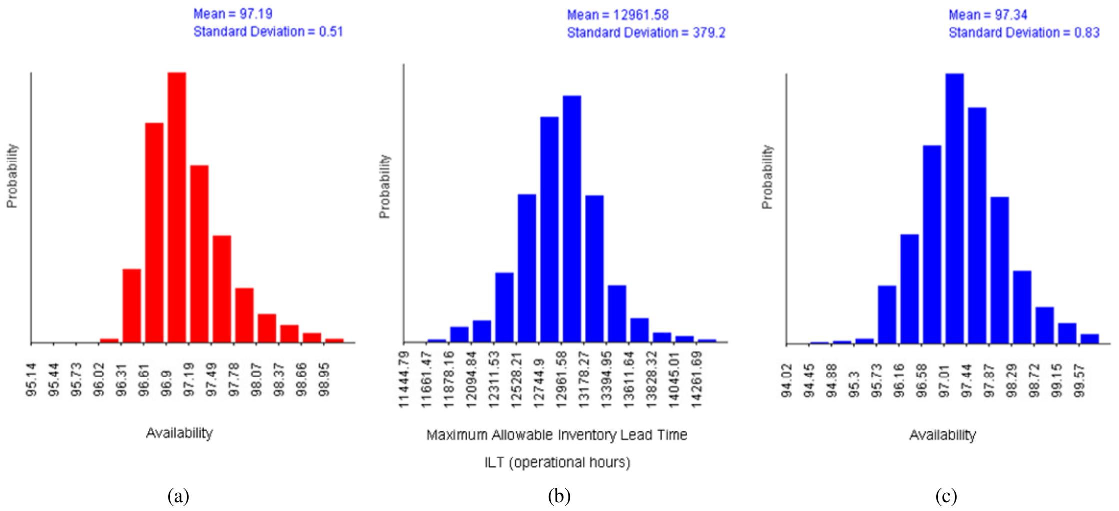
$$IDT_1 = \frac{IS - MDT_1}{A_o} - IS \qquad (4.2)$$

Once IDT_1 is determined, the *ILT* can be computed by satisfying equation (4.1).

Finally, the *ILT* is updated as the downtime requirement gets updated throughout the entire support life. Similar process is generally used to apply the approach to any system parameter that is explicitly related to either downtime or uptime, to fulfill a specific availability requirement.

The availability distribution considered for this example case is shown in Figure 4.3.a. This distribution could represent the requirement of an availability contract. Note, availability contracts may specify the availability requirement as a single value, but, to accommodate more general problems, for all example cases I will use availability requirements that are represented as a probability distribution.

Considering an availability requirement that is expressed as a probability distribution makes the process of determining the necessary system parameters to meet the availability requirement challenging, since every system instance could have a different availability requirement based on the sampled value from the probability distribution. Figure 4.4 shows the process used to generate a distribution of system parameter values using a discrete-event simulator (the PHM ROI model described earlier). The Monte Carlo implementation of the model samples the required availability distribution and other quantities that may be described as probability distributions, and then uses the quantities to solve for a value of the system parameter using the design for availability methodology. This process is repeated for each socket (a socket is an instance of an installation location for an LRU) in the population, resulting in histograms of system parameter values (e.g., *ILTs*). Figure 4.3.b shows the *ILT* required to meet the availability requirement determined using the design for availability methodology.



(a) (b) (c)

Figure 4.3. (a) Required availability distribution (input to the model). (b) Computed maximum allowable Inventory Lead Time (*ILT*) in operational hours, for an unscheduled maintenance policy. (c) Availability probability distribution generated using the computed *ILT* (Figure 4.3.b).

In order to qualitatively verify the methodology, the maximum allowable *ILT* distribution (Figure 4.3.b) was used as an input to the PHM ROI model. The PHM ROI model used the maximum allowable *ILT* distribution along with the other inputs (see Section 4.1) and generated a resulting availability distribution. Figure 4.3.a shows the original availability input (mean = 97.19% and standard deviation =

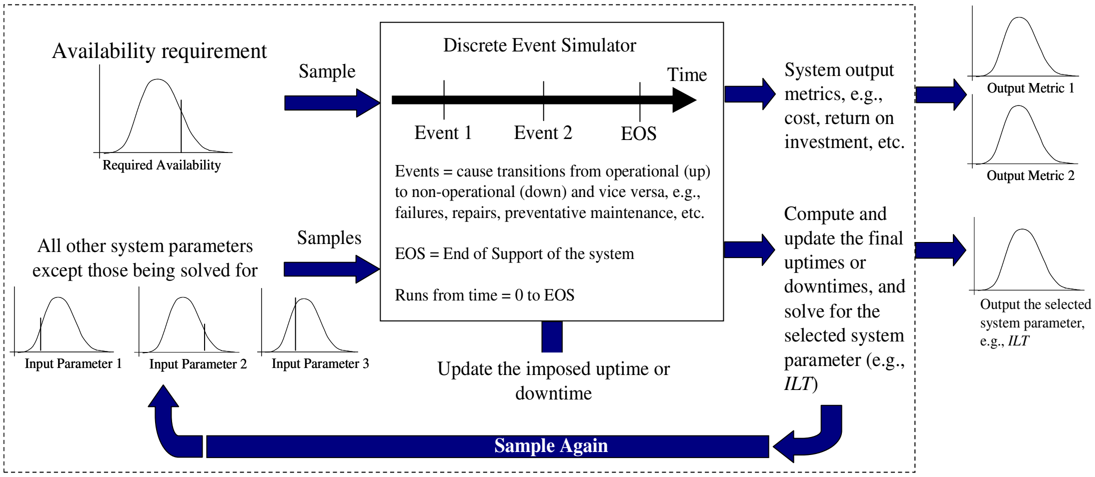

Availability requirement
Required Availability
Sample
Discrete Event Simulator
Time
Event 1
Event 2
EOS
Events = cause transitions from operational (up) to non-operational (down) and vice versa, e.g., failures, repairs, preventative maintenance, etc.
EOS = End of Support of the system
Runs from time = 0 to EOS
System output metrics, e.g., cost, return on investment, etc.
Output Metric 1
Output Metric 2
All other system parameters except those being solved for
Samples
Input Parameter 1
Input Parameter 2
Input Parameter 3
Update the imposed uptime or downtime
Compute and update the final uptimes or downtimes, and solve for the selected system parameter (e.g., ILT)
Output the selected system parameter, e.g., ILT
Sample Again

Figure 4.4. Solution process.

0.51%), while Figure 4.3.c shows the availability prediction that resulted from the PHM ROI model (mean = 97.34% and standard deviation = 0.83%). The two results are not expected to be absolutely identical (since this is a stochastic solution), but the means and standard deviations are very similar. This simple example case validates the methodology and demonstrates qualitatively that the design for availability approach is truly satisfying the input availability requirement.

Unscheduled Maintenance vs. Data-driven PHM

In this Section I compare the maximum allowable *ILT* for a specific availability requirement for unscheduled maintenance and a data-driven PHM approach.

Determining the maximum allowable *ILT* for a specific availability requirement could be used to improve logistics management and potentially reduce life cycle cost. If the availability drops below a specified threshold value, a cost penalty could be assessed; determining upfront the appropriate *ILT* could avoid these potential cost penalties. Also, knowing the maximum allowable *ILT* information, customers could require their suppliers to deliver within a specific lead time.

For the assumed set of system parameters and assumptions (see Section 4.1) I want to determine the appropriate spares replenishment lead time, i.e., inventory lead time (*ILT*), to fulfill the availability requirement specified in Figure 4.3a for an unscheduled maintenance approach and a data-driven PHM approach applied to the same system (a detailed explanation of how the data-driven PHM approach is modeled is provided in Section 2.3).

Reducing the delivery time, i.e., *ILT* (considered as the only variable input in this example) would increase the availability. However, I also want to maximize the *ILT* to reduce the cost. Basically, I want to generate an optimal solution that produces the maximum allowable *ILT* (to minimize cost) that keeps the availability value at or above the availability requirement.

By running the simulation with the imposed availability (input) requirement shown in Figure 4.3.a, the *ILT* (output) satisfying this requirement was determined for the unscheduled maintenance policy, and the generated maximum allowable *ILT* probability distribution is shown in Figure 4.5, the light gray histogram bars (this distribution is the same as Figure 4.3.b).

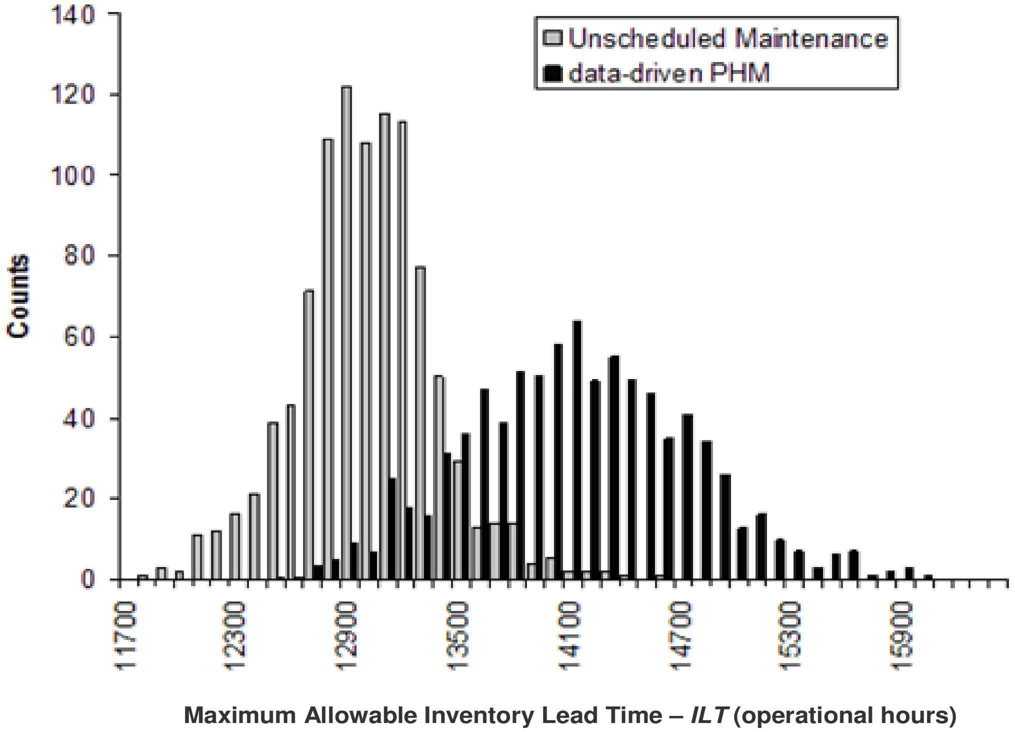


Figure 4.5. Computed maximum allowable inventory lead time (*ILT*) for unscheduled maintenance policy, and for a data-driven PHM approach.

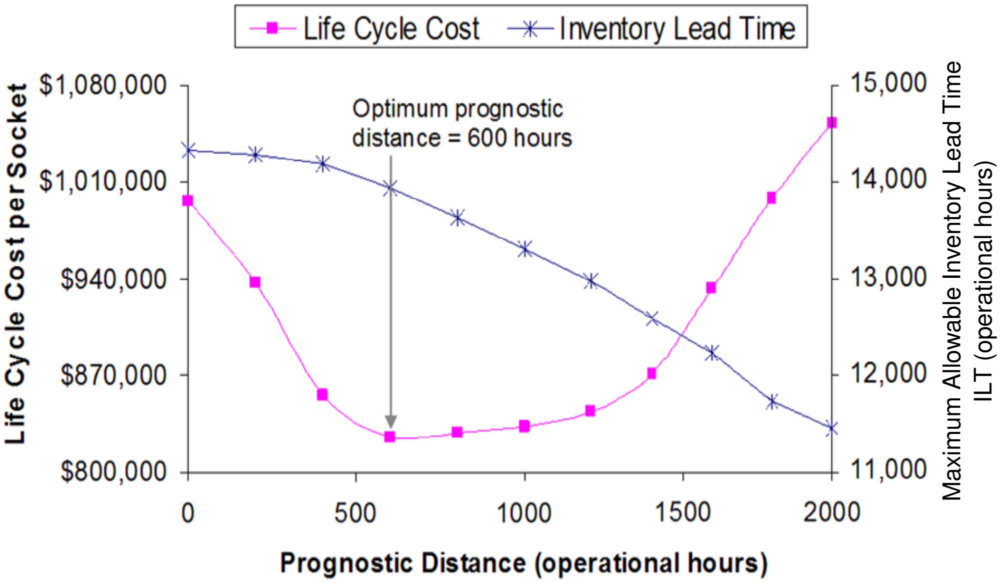


Figure 4.6. Variation of life cycle cost and inventory lead time with data-driven PHM prognostic distance.

I now need to apply the methodology to the data-driven PHM approach. Figure 4.6 shows the results of the analysis to determine the optimal (lowest life cycle cost) prognostic distance for the data-driven PHM approach used; where the prognostic distance is defined as the measure (e.g., operational hours) of how long the prognostic structures or prognostic cell is expected to indicate failure, before the system actually fails (see Section 2.3).

Small prognostic distances may miss failures while large prognostic distances may throw away significant remaining useful life. Each prognostic distance generates a corresponding *ILT* and life cycle cost. Note, the *ILT* values shown in Figure 4.6 are the mean values of the generated maximum allowable *ILT* distributions. Large prognostic distances may provide additional time to order spares ahead of failures; however, they could also produce solutions that require more spares, which will increase the accumulated *IDT* (inventory downtime, i.e., time that the system is down waiting for spares) since every delayed spares replenishment event generates an additional *IDT*. Therefore, to accommodate these downtimes, while fulfilling the availability requirement, the *ILT* has to be reduced (i.e., reduce the *IDT*, thus faster delivery). This is illustrated in Figure 4.6; where large prognostic distances generate shorter *ILT*.

Using the input data (provided in Section 4.1) with the data-driven PHM approach, an optimal prognostic distance of 600 operational hours results in the minimum life cycle cost over the entire support life. Also, a symmetric triangular distribution with a width of 500 hours was assumed to represent the effectiveness of the data-driven PHM approach.

After running the simulation with the imposed availability requirement shown in Figure 4.3.a, the maximum allowable *ILT* satisfying the contract requirement was determined, and the generated *ILT* probability distribution is shown in Figure 4.5 (black histogram bars).

In this example, the data-driven PHM approach allows for a larger *ILT* (mean = 13,936 operational hours), compared to the unscheduled maintenance case (mean = 12,961 operational hours)[11]. In other words, using a data-driven PHM approach allows a given availability requirement to be met if *ILTs* are longer, or alternatively stated, the use of PHM would allow a supply chain with longer *ILTs* to be used. The use of a data-driven PHM approach has shifted the maximum allowable *ILT* distribution by approximately 1000 hours to the right. This result is due the fact that

[11] The overlap area of *ILT* distributions in Figure 4.5 is negligible compared to the main distributions areas. This justifies the considerable difference (≈ 1,000 operational hours) in distributions' mean values. Also, Figure 4.7 supports these results, since it shows that for the same *ILT* value an availability requirement could be met with higher confidence level, when using a data-driven PHM approach.

data-driven PHM has provided early warning of failures; therefore creating the opportunity to switch maintenance actions from unscheduled to scheduled events reducing the accumulated operational downtime. For a fixed *ILT*, this would result in an improved operational availability of the system. However, since in this problem the same availability requirement was imposed for both cases (unscheduled maintenance and data-driven PHM), thus the accumulated operational downtime was used as a fixed quantity (imposed by the contract availability requirement); then the avoided unscheduled maintenance downtime was added to the *IDT*, resulting in a larger allowed *ILT*.

To summarize, the design for availability methodology was applied to the case study example described in Section 4.1, for two different maintenance approaches, to satisfy a specific contract availability requirement. For both approaches, unscheduled maintenance and data-driven PHM, I was able to determine the unknown system parameter (the maximum allowable *ILT* in this case) satisfying the availability requirement. Then a comparison of the results showed that data-driven PHM allows larger *ILTs* compared to the unscheduled maintenance policy case.

Figure 4.7 shows how the maximum allowable *ILT* results could be practically interpreted. For example, if the *ILT* of each spares replenishment order is equal or greater than 12,700 operational hours, then the system manager would be 95% confident to meet the availability requirement under a data-driven PHM approach, and only 78% confident to meet the same availability requirement under an unscheduled maintenance approach.

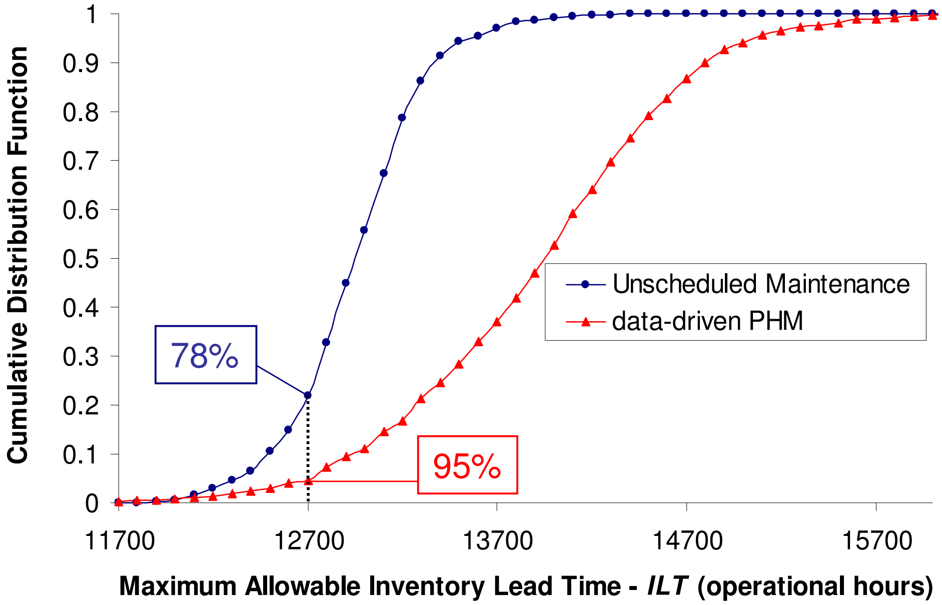
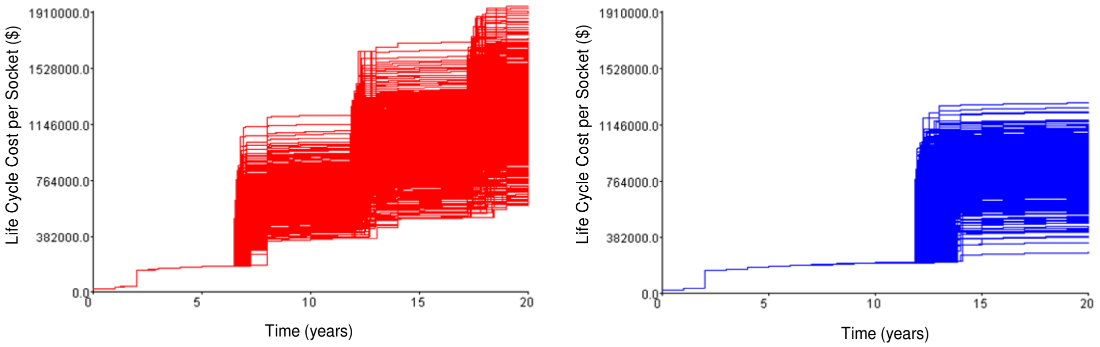


Figure 4.7. Computed *TTF* cumulative distribution function for unscheduled maintenance and data-driven PHM.

Notice that, the application of the design for availability methodology doesn't only determine the unknown system parameter (e.g., *ILT*) but also illustrates the effect of each adopted sustainment approach on the selected system parameter. In other words, for a specific PHM approach or any other maintenance policy, one could expect the variation of the selected system parameter, and react accordingly to maintain the availability requirement.

Cost Analysis

In the analysis described in this section, the PHM data-driven case produced lower life cycle cost (mean = $848,089) compared to the unscheduled maintenance case (mean = $1,241,238). This is due to the cost of purchasing additional spares replenishment in the unscheduled maintenance case. Unscheduled maintenance approaches generally require the minimum number of spares because the unscheduled maintenance events are performed upon the actual failure of the LRUs, thus maximizing the useful life of the LRUs. However, in this case, I have assumed that early warning of failures, in the data-driven PHM case, provides an opportunity to schedule and perform on-site maintenance events that resulted in repairing most LRUs, i.e., PHM provides the capability to intervene before a complete deterioration of the LRUs, resulting in replacing fewer LRUs (and thus ordering fewer spares). While in the unscheduled maintenance case, I assumed that all failures were resolved by replacing LRUs rather than repairing them. Since the PHM data-driven case required fewer spares, it required only one spare replenishment event (the large step in Figure 4.8.b); while the unscheduled maintenance case required more spares and three replenishment events (the three large steps in Figure 4.8.a). Also, for the PHM data-driven sustainment approach the billing due date for the spares replenishment events happened on a later date (~12th year in Figure 4.8.b) than the unscheduled maintenance case (~7th, 12th and 18th year in Figure 4.8.a), making the PHM data-driven approach less costly because of the non-zero discount rate.



(a) (b)

Figure 4.8. Life cycle cost per socket for a 1000 socket population. (a) Life cycle cost per socket for an unscheduled maintenance. (b) Life cycle cost per socket for a data-driven PHM approach.

Note, the cost per hour out of service was the main cost driver for both cases, data-driven PHM and unscheduled maintenance; since the contract availability requirement value was relatively low (relatively large downtimes were allowed). The life cycle cost for both cases (data-driven PHM and unscheduled maintenance) would have been dramatically reduced for a higher availability requirement (e.g., 99.99%).

Return on Investment (ROI)

In this subsection, the return on investment (ROI) of a data-driven PHM approach relative to unscheduled maintenance is analyzed. The total life cycle cost per socket, for a data-driven PHM approach, was $848,089 (mean), with an effective investment cost per socket of $6,891 (mean), representing the cost of developing, supporting, and installing PHM. This cost was compared to the unscheduled maintenance approach, where the total life cycle cost per socket was $1,241,238 (mean). Note that the investment cost for the unscheduled maintenance policy is by definition zero; since the ROI is computed to support an economic justification in investing in PHM, as opposite to the unscheduled maintenance case where there is no investment (i.e., zero investment) in PHM. A detailed description of the methodology of determining ROI for PHM systems is provided in Section 2.9.

Figure 4.9 shows the histogram of the computed ROI for 1000-socket population, using the inputs data provided in Section 4.1. In this example case, the computed mean ROI of investing in a data-driven PHM approach for the population of sockets was 57.0. This is a relatively large value of ROI, which is the result of the small PHM investment cost. Notice that some ROI values in Figure 4.9 become negative. This means that there is a risk that implementing a data-driven PHM approach that meets the specified availability requirement for the system specified in Section 4.1, could result in an economic loss, i.e., I could end up being worse off than unscheduled maintenance. Based on Figure 4.9, this example predicts that a data-driven PHM approach would result in a positive ROI (cost benefit) with a 93.9%

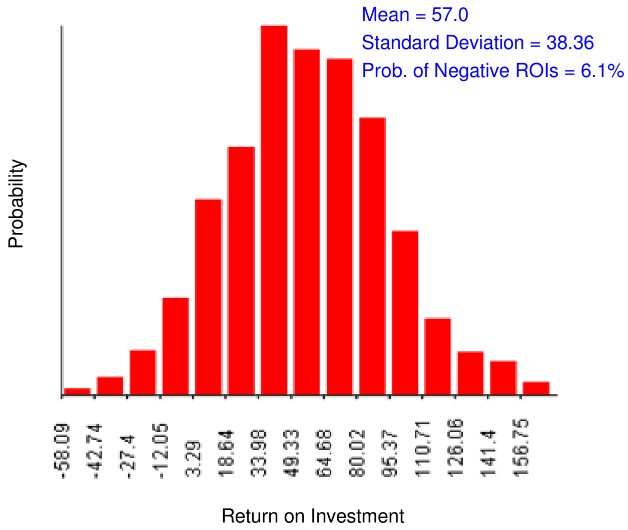


Figure 4.9. Histogram of ROI for a 1000-socket population, for a data-driven PHM relative to unscheduled maintenance that meets the availability requirement.

confidence. Notice that in this case the ROI was computed for a data-driven PHM relative to an unscheduled maintenance, with two different *ILT* distributions. An *ILT* distribution that was generated by a data-driven PHM to meet the availability requirement, and an *ILT* distribution that was generated by an unscheduled maintenance to meet the same availability requirement.

Figure 4.10 shows the histogram of the computed ROI for 1000-socket population, for a data-driven PHM relative to an unscheduled maintenance policy using the same *ILT* distribution, which was generated to meet the availability requirement with a data-driven approach. In this case, the unscheduled maintenance approach does not meet the availability requirement. The computed mean ROI of investing in a data-driven PHM approach for the population of sockets was 167.64. This larger ROI value is explained by the fact that the unscheduled maintenance case used larger values of *ILT* (which were generated by the data-driven PHM approach to meet the availability requirement).

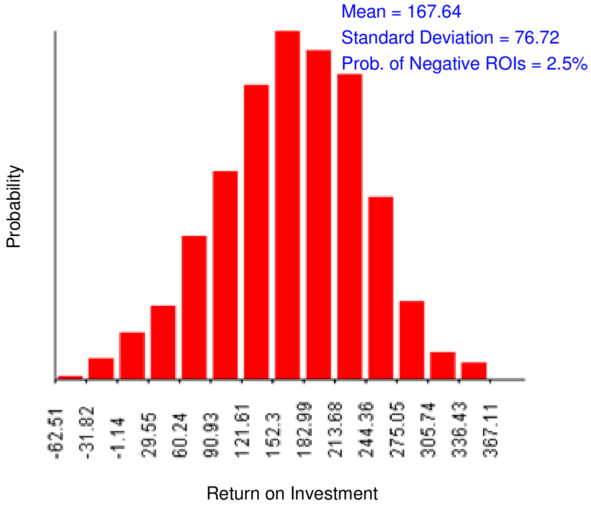


Figure 4.10. Histogram of ROI for a 1000-socket population, for a data-driven PHM relative to unscheduled maintenance that uses an *ILT* distribution generated by the data-driven PHM.

4.3. Use of Design for Availability to Determine System Parameters Affecting both Uptime and Downtime

Demonstration and Verification of the Methodology: Reliability (*TTF*)

In this section, the objective of the application of the design for availability methodology is to determine the minimum[12] allowable reliability, i.e., time-to-failure (*TTF*), of the LRUs that is necessary to meet the availability requirement. In this example case, the reliability of each LRU is represented by its *TTF*, where each *TTF* corresponds to the period of time until the occurrence of the next actual failure.

TTF is used as an example to demonstrate the application of the methodology to system parameters concurrently affecting both uptime and downtime. For example, consider the following scenario (which illustrates how *TTF* could concurrently affect both uptime and downtime): the replenishment spares will be delivered one year from now for a specific inventory, while this inventory is currently out of spares. The system (the socket), that is drawing spares from this specific inventory as needed, will be up and running as long as the currently used spare by the system doesn't require replacement, thus the system uptime is dependent on the *TTF* of this spare. Also, the system downtime could be minimized if the spare being used does not require replacement until the replenishment spares are delivered (e.g., one year from now). However, as soon as the spare requires replacement, the system will be down until additional spares are received. Thus, the system downtime is dependent on the *TTF* of

[12] The required availability distribution and other quantities (inputs) that may be described as probability distributions, are sampled and used to solve for a single value of *TTF*. This value represents the minimum *TTF* value (minimum allowable reliability) that is necessary to meet the sampled required availability in the environment defined by the sampled values of all the other input quantities. This process is repeated for each socket in the population, resulting in a histogram of minimum allowable *TTFs*. Each individual in the histogram represents one socket in the population of sockets under one possible set of life cycle conditions. This solution process was illustrated in Figure 4.4.

this spare. This simple scenario demonstrates how the *TTF* of the LRUs could affect both the uptime and downtime. Notice that a detailed discussion and assumptions of the spares management model is provided in Section 2.6.

To demonstrate and verify the derivation of the *TTF* for a specific availability requirement, the design for availability methodology has been implemented within the PHM ROI model.

In order to use the application of the methodology on *TTF* as a qualitative verification of the methodology, I will perform the same verification process steps as used with system parameters affecting either uptime or downtime. This means, first, using the availability distribution requirement as an input, determine the distribution of the minimum allowable *TTF*. Then, for verification purposes, use the generated *TTF* distribution as an input to the existing PHM ROI simulation to predict an availability distribution as an output. Finally, compare the availability distribution input requirement to the availability distribution determined as an output – they should be equivalent.

A detailed description of all of the case study inputs is provided in Section 4.1, including LRU description, implementation and maintenance costs, and operational profile.

The reliability information is not provided, since in this example case the reliability (*TTF*) is the selected unknown system parameter being solved for. The inventory management parameters are provided in Table 4.4, which summarizes the spares inventory (per socket) assumptions that are used for this specific example.

Also, note that the spares carrying costs are incorporated into the LRU recurring costs.

Table 4.4. Spares Inventory

Factor	Quantity
Initial spares purchased for each socket	3
Threshold for spare replenishment	≤ 1 spares in the inventory per socket
Number of spares to purchase per socket at replenishment	2
Spare replenishment lead time	18 calendar months
Spares carrying cost	10% of the beginning of year inventory value per year
Billing due date when ordering additional spares	2 years from purchase date

Both, the *TTF* values and the distributions modeling the effectiveness of a particular PHM approach are used to predict the remaining useful life (RUL) of the LRUs (see Chapter 2). For each PHM sustainment approach (e.g., data-driven, model-based – also known as physics of failure, fixed-interval scheduled maintenance and unscheduled maintenance), the sampling of the *TTF* values is defined differently. The sampled TTF values are used to predict the maintenance events and to determine whether the selected PHM approach detected (or failed to detect) a failure.

In the unscheduled maintenance case, the sampling of the *TTF* values predict the date of the next maintenance event associated with a failure of a system instance. Spares are drawn from the inventory as needed to support maintenance. Once the inventory reaches a threshold value, additional spares are ordered, and the replenishment spares are delivered after a delivery lead time. Figure 4.11 illustrates this scenario, where *MDT* is maintenance downtime, *ILT* is the inventory lead time, *ST* is the spares threshold (once the inventory level drops below this value, additional spares are ordered), and *IDT* is inventory downtime (when the inventory runs out of spares, and the system is down waiting for spares).

Notice that the accumulated uptime (*UT*) accounts for all system's uptimes. This includes the system's uptime while using the inventory initial spares (*IS*) and the system's uptime while using inventory replenishment spares (*RS*). The *RS* could be ordered multiple times as needed,

$$\sum UT = (IS)(TTF) + \sum (RS)(TTF) \qquad (4.3)$$

The accumulated downtime (*DT*) includes the maintenance downtime (*MDT*) and the inventory downtime (*IDT*),

$$\sum DT = \sum IDT + \sum MDT = \sum (ILT - (ST)(TTF)) + \sum MDT \qquad (4.4)$$

Notice that the summations in equations (4.3) and (4.4) do not necessarily refer to the analytical summations, but to the accumulation of events. Since these relationships are based on the accumulation of the event outcomes and sequences, that are only determined in real simulation time. Also, the model is probabilistic, this means each sample of the same quantity, i.e., system parameter, could result in a different event outcome. A detailed explanation is provided in Section 3.6.

The operational availability is, by definition, the accumulated uptime over the total operational time (i.e., sum of the total accumulated uptime and downtime),

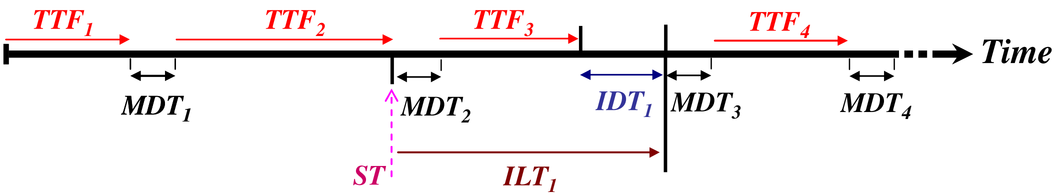


Figure 4.11. TTF implication on the operational timeline.

$$A_o = \frac{\sum UT}{\sum UT + \sum DT} \qquad (4.5)$$

where *UT* is the accumulated uptime and *DT* is the accumulated downtime.

For example, the k^{th} *TTF* value could be derived by combining equations (4.3), (4.4) and (4.5). The k^{th} *TTF* corresponds to the k^{th} downtime, where the k^{th} downtime could be a maintenance downtime, inventory downtime, or any other logistics downtime. Once again, the summations in equation (4.6) do not refer to analytical summations, but to the accumulation of events outcomes and sequences. Therefore, the right side of equation (4.6) does not explicitly include the "*k*" subscript,

$$TTF_k = \frac{\sum ILT + \sum MDT}{\frac{1-A_o}{A_o}\left(IS + \sum RS\right) + \sum ST} \qquad (4.6)$$

Notice that equations (4.3)-(4.6) could be slightly different for each problem set up or model. The modeling of the operational timeline illustrated in this section is by no means unique. However, different models could provide different equations, but, the steps of the procedure remain the same. Thus, the application of the design for availability methodology is general and could be apply to any problem, independently of the setup of these equations.

After every downtime, the *TTF* is computed using the procedure described above. However, the methodology derives the minimum allowable *TTF* that is necessary to meet the availability requirement. Figure 4.12 illustrates the process of updating the computed *TTFs*. Basically, after every downtime, the computed *TTF* is compared to the previous value, if the current value is greater than the previous one,

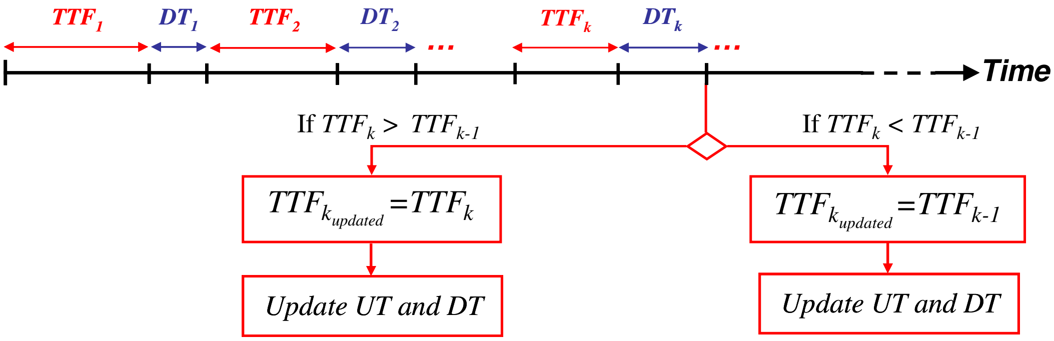
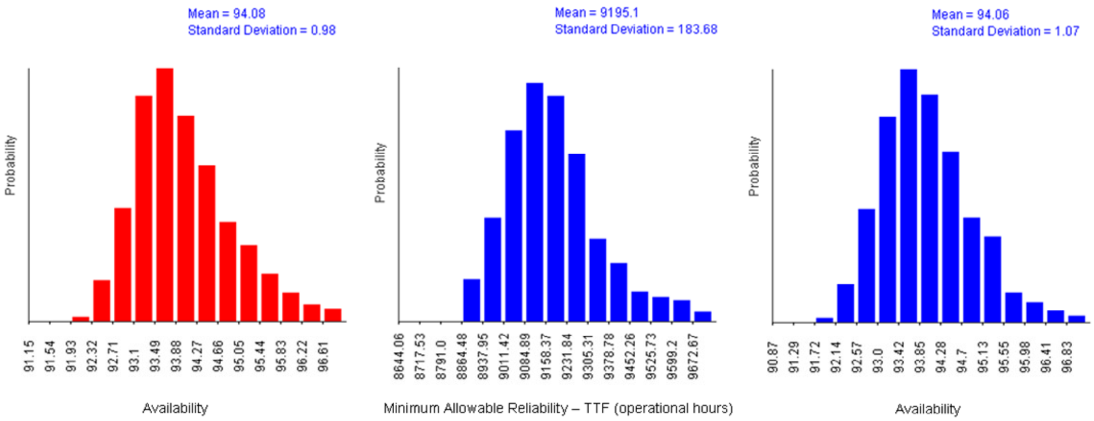


Figure 4.12. Updating the *TTFs*, *UTs* and *DTs*.

then the current value is substituted for the previous value. But if the current value is less than the previous one, then the current one is used. Once, the current *TTF* value is updated, this new *TTF* requirement is imposed on the uptime and downtime values through equations (4.3) and (4.4). Finally, the model uses the updated *TTFs*, *UTs*, and *DTs* to compute other quantities of interest.

While considering an availability requirement that is expressed as a probability distribution is more realistic, it makes the process of determining the necessary system parameters to meet the availability requirement challenging, since every system instance could have a different availability requirement based on the sampled value from the probability distribution. Figure 4.4 shows the process used to generate a distribution of system parameter values using a discrete event simulator (the PHM ROI model described earlier).



(a) (b) (c)

Figure 4.13. (a) Required availability distribution (input to the model). (b) Computed minimum allowable reliability (*TTF*) in operational hours, for an unscheduled maintenance policy. (c) Availability probability distribution generated using the computed *TTF* (Figure 4.13.b).

The availability requirement considered in this example case, for an unscheduled maintenance policy, is shown in Figure 4.13.a. This availability requirement has been used as an input to the design for availability model. Figure 4.13.b show the resulting TTF distribution. The TTF distribution was generated through the process illustrated in Figure 4.4 and using the input data provided in Section 4.1.

In order to qualitatively verify the methodology, the TTF distribution (Figure 4.13.b) was used as an input to the PHM ROI model, while using an unscheduled maintenance approach. The PHM ROI model used the TTF distribution along with the other data inputs and generated a resulting availability distribution. Figure 4.13.c shows the availability prediction that resulted from the PHM ROI model. The two availability distributions (Figures 4.13.a and 4.13.c) are not expected to be absolutely identical (since this is a stochastic solution), but the means and standard deviations are very similar. This qualitatively validates the design for availability model.

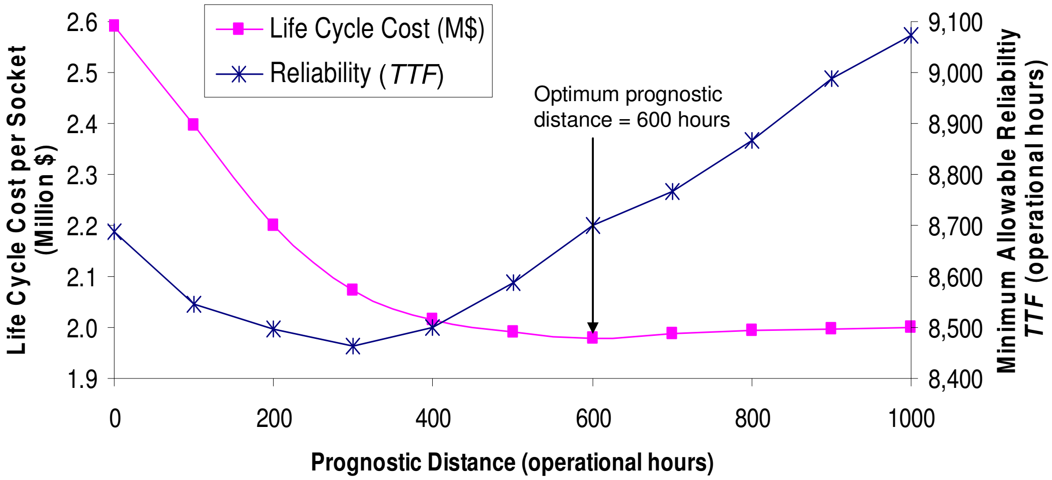


Figure 4.14. Variations of life cycle cost and minimum allowable reliability (*TTF*) with data-driven PHM prognostic distance.

Unscheduled Maintenance vs. Data-Driven PHM Approach

A detailed description of the inputs data used for this example is provided in Section 4.1. For this example case study the optimal data-driven PHM prognostic distance was determined by selecting the prognostic distance resulting in a minimum life cycle cost. Where the prognostic distance is defined as the measure of how long the prognostic structure or prognostic cell is expected to indicate failure before the system actually fails (see Section 2.3). This analysis has resulted an optimal prognostic distance of 600 operational hours (see Figure 4.14).

For each prognostic distance there is a corresponding minimum allowable *TTF* distribution and life cycle cost distribution (see Figure 4.14). However, the *TTF* and life cycle cost values shown on Figure 4.14 are the means of the generated *TTF* and life cycle cost distributions respectively.

Small prognostic distances maximize the LRUs useful life, but result more unanticipated failures (more unscheduled maintenance events, i.e., expensive and larger maintenance time), thus, potentially increasing the maintenance downtime. Consequently, they require larger *TTFs* to produce larger uptime durations, since the pre-imposed uptime-downtime relationship has to be maintained in order to satisfy the availability requirement. In this case, the life cycle cost is increased because of the cost of the unscheduled maintenance, i.e., unanticipated failures.

On the other hand, large prognostic distances throw away considerable remaining useful life of the LRUs. Thus, increase the number of spares drawn from inventory and spares sent to the repair process, and potentially increase the inventory downtime. However, more failures are avoided (more scheduled maintenance, i.e., less expensive and shorter maintenance time). Similarly, to maintain the uptime-downtime relationship defined by the availability requirement, larger *TTFs* are required. In this case, the life cycle cost is increased by the cost of the repair process and inventory downtime.

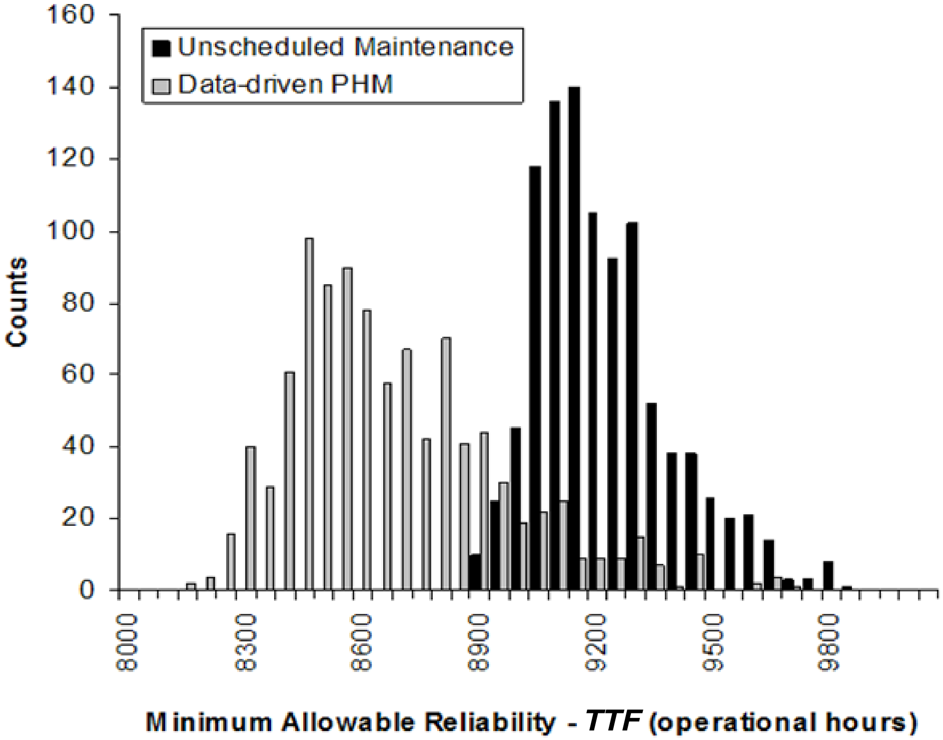


Figure 4.15. Computed minimum allowable reliability (*TTF)* distribution for unscheduled maintenance and data-driven PHM.

The availability requirement considered in this subsection is shown on Figure 4.13.a. This availability requirement has been used as an input to the design for availability model. Figure 4.15 show the resulting *TTF* distributions using unscheduled maintenance and data-driven PHM, in light grey and black colors respectively. The *TTF* distributions were generated through the process illustrated in Figure 4.4 and using the input data provided in Section 4.1.

By comparing the resulting *TTF* distributions for unscheduled maintenance and data-driven PHM approaches (Figure 4.15), data-driven PHM has allowed a lower *TTF* requirement. This means, in this example case, using a data-driven PHM approach relaxes (relative to unscheduled maintenance) the required *TTF* to meet the imposed availability requirement. This is a powerful result from the design for availability methodology, since the methodology doesn't only derive the necessary system parameters for a specific availability requirement, but it also reflects the impact of a PHM approach on the selected system parameters, thus, providing a better understanding of the relationship of a PHM implementation and the system parameters. Also, the methodology emphasizes the fact that a PHM implementation selection should incorporate all design, support and logistics parameters. In other words, based on the design, support, or logistics management, one PHM approach could be more feasible than the other.

Predicting the *TTF* distribution could be used to avoid the contract availability penalties, since a cost penalty could be assessed for not fulfilling the availability requirement specified in the contract. Also, the minimum allowable *TTF* information could be used to define requirements and provide feedback to the design process, since it is more expensive to design LRUs with larger *TTFs*. Finally, explicitly expressing the *TTF* distribution could be used to predict and understand the system's behavior.

Figure 4.16 shows how the *TTF* results could be practically interpreted. For example, if the reliability (*TTF*) of each LRU is equal or greater than 9,000 operational hours, then the system manager would be 87% confident to meet the availability requirement under if a data-driven PHM approach is used, and only 8% confident to meet the same availability requirement under an unscheduled maintenance approach.

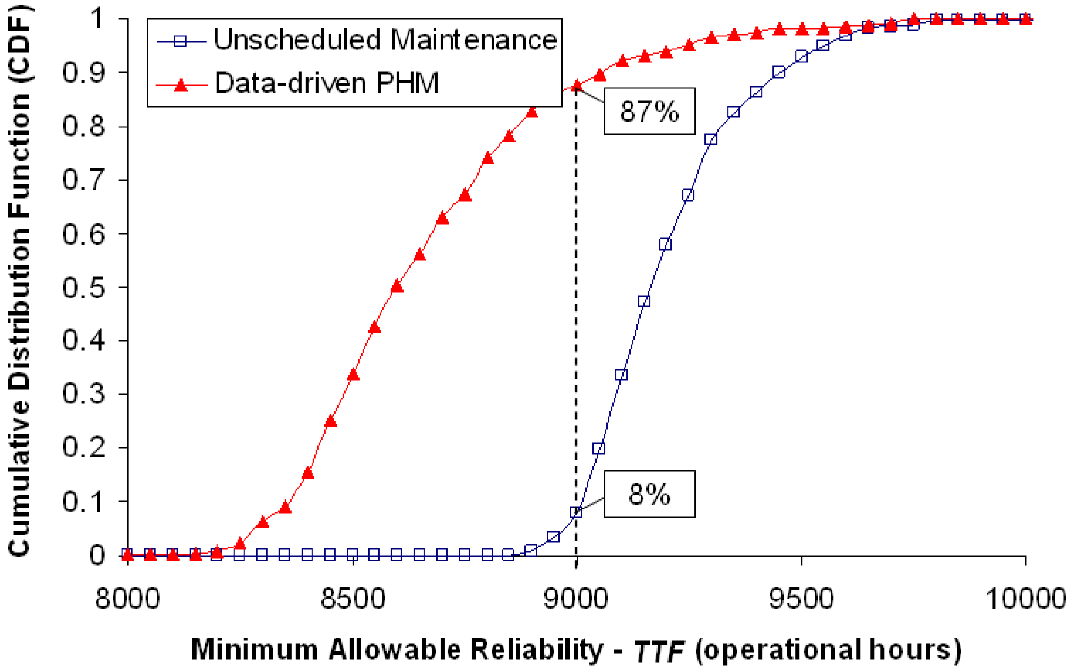


Figure 4.16. Computed *TTF* cumulative distribution function for unscheduled maintenance and data-driven PHM.

Cost Analysis

Figure 4.17 represents the accumulation of the life cycle cost per socket for both the data-drive PHM and unscheduled maintenance case. The data-driven PHM case resulted in an overall lower life cycle cost (mean = $1,973,625) compared to the unscheduled maintenance case (mean = $2,469,334). In this example, the data-driven PHM approach case required fewer spares throughout the support life of the system, compared to the unscheduled maintenance policy case. This result is due to the ability to repair versus replace.

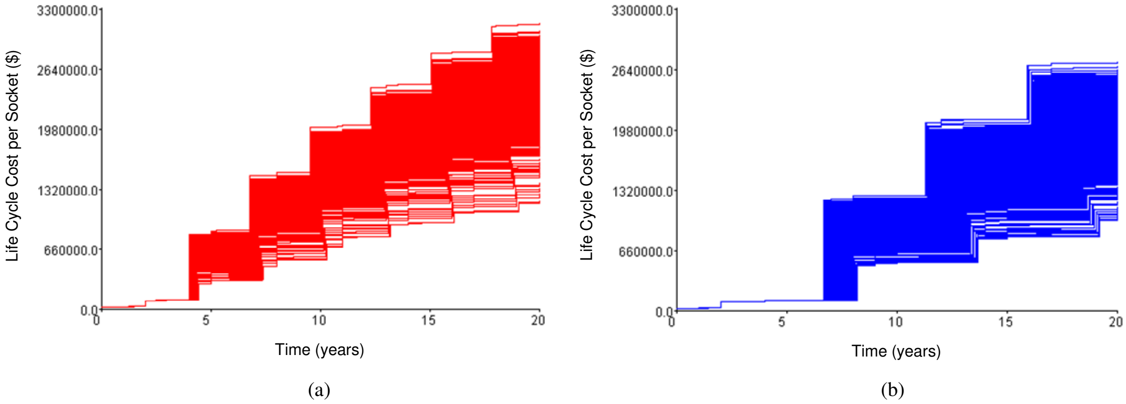


Figure 4.17. (a) Life cycle cost per socket for an unscheduled maintenance policy. (b) Life cycle cost per socket for a data-driven PHM approach.

In the data-driven PHM sustainment approach case the billing due date for the initial and most spare replenishment events occurred on a later date than the unscheduled maintenance case, therefore the cost of purchasing additional spares was smaller because due to the discount rate on money. The annual steps seen in Figure 4.17, are relatively larger for the data-driven PHM approach, because: more spares are held in the inventory (higher annual spares carrying cost), expensive spares (PHM recurring costs are added to LRU purchase price) and PHM infrastructure costs are annually accumulated. Finally, notice that the total accumulated downtime is constant for both cases (imposed by the availability requirement); this explains the small steps in Figure 4.17 for unscheduled maintenance case during the replenishment events at approximately years 4, 7, 9, etc. (frequent short, i.e., less expensive, inventory downtimes), compared to the data-driven PHM case large steps at approximately years 7, 12 and 17 (less frequent longer, i.e. expensive, inventory downtimes). On the other hand, the maintenance downtimes generated by the unscheduled maintenance case have been larger (unanticipated and unscheduled events) compared to the maintenance downtimes generated by the data-driven PHM case (anticipated and scheduled events).

This cost analysis could have been even more favorable to the data-driven PHM case, since in this example case I did not include the modeling of the cost associated with improving an LRU's reliability, i.e., *TTF*. Figure 4.15 shows that, in this example, the unscheduled maintenance case required larger *TTFs* compared to the data-driven PHM case to meet the same availability requirement (Figure 4.13.a). Thus, if the cost of improving *TTFs* was included, then the larger *TTFs* requirement in the unscheduled maintenance case would have cost more, resulting in a larger life cycle cost for the unscheduled maintenance approach.

Return on Investment (ROI)

In this subsection, the return on investment (ROI) of a data-driven PHM approach relative to unscheduled maintenance is analyzed. The total life cycle cost per socket, for a data-driven PHM approach, was $1,973,625 (mean), with an effective investment cost per socket of $6,749 (mean), representing the cost of developing, supporting, and installing PHM. This cost was compared to the unscheduled maintenance approach, where the total life cycle cost per socket was $2,469,334 (mean). Note that the investment cost for the unscheduled maintenance policy is by definition zero; since the ROI is computed to support an economic justification in investing in PHM, as opposite to the unscheduled maintenance case where there is no investment (i.e., zero investment) in PHM. A detailed description of the methodology of determining ROI for PHM systems is provided in Section 2.9.

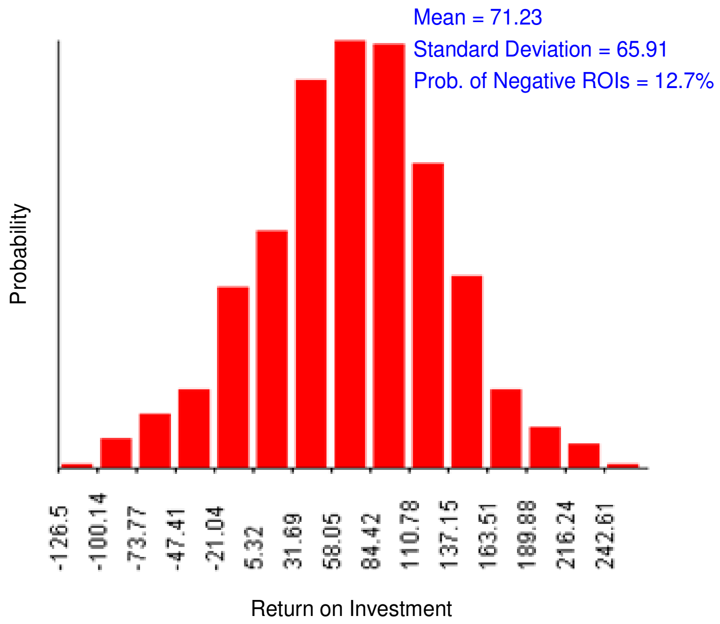


Figure 4.18. Histogram of ROI for a 1000-socket population, for a data-driven PHM relative to unscheduled maintenance that meets the availability requirement.

Figure 4.18 shows the histogram of the computed ROI for 1000-socket population, using the inputs data provided in Section 4.1. In this example case, the computed mean ROI of investing in a data-driven PHM approach for the population of sockets was 71.23. This is relatively a large value of ROI, which is justified by the small PHM investment cost. Notice that some ROI values in Figure 4.18 become negative. This means that there is a risk that implementing a data-driven PHM approach that meets the specified availability requirement for the system specified in Section 4.1, could result in an economic loss, i.e., I could end up being worse off than unscheduled maintenance. Based on Figure 4.18, this example predicts that a data-driven PHM approach would result in a positive ROI (cost benefit) with an 87.3% confidence.

Notice that in this case the ROI was computed for a data-driven PHM relative to an unscheduled maintenance where each of the maintenance approaches has a

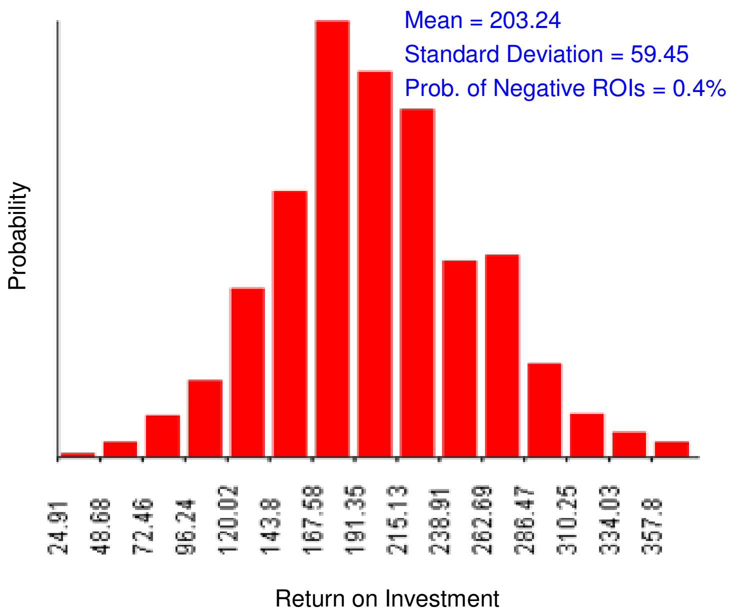


Figure 4.19. Histogram of ROI for a 1000-socket population, for a data-driven PHM relative to unscheduled maintenance that uses a TTF distribution generated by a data-driven PHM.

different *TTF* distribution - *TTF* distribution that was generated by a data-driven PHM to meet the availability requirement, and a *TTF* distribution that was generated by an unscheduled maintenance to meet the same availability requirement.

Figure 4.19 shows the histogram of the computed ROI for 1000-socket population, for a data-driven PHM relative to an unscheduled maintenance policy using the same *TTF* distribution, which was generated to meet the availability requirement with a data-driven approach. The computed mean ROI of investing in a data-driven PHM approach for the population of sockets was 203.24. This larger ROI value is explained by the fact that the unscheduled maintenance case used shorter TTFs (which were generated by the data-driven PHM approach to meet the availability requirement).

Chapter 5: Summary and Contributions

5.1. Summary

In this dissertation a new methodology for determining unknown system parameters to fulfill a specific availability requirement has been presented. The design for availability methodology is not a "search-based" approach and is capable of calculating unknown system parameters directly from the availability requirement even when the inputs (e.g., design and support parameters) are uncertain and the availability requirement is represented as a probability distribution.

Case study examples were presented to demonstrate the methodology, as well as providing a means for verification and qualitative validation purposes. Case study results for an example system managed using both unscheduled maintenance and a data-driven PHM approach were also included. The model predicted a larger allowable *ILT* for the data-driven PHM case for the same availability requirement. The conclusions in this dissertation about the *ILT* associated with PHM and unscheduled maintenance approaches are specific to the example data assumed and should not be interpreted as a general conclusion. However, the example demonstrates that the use of PHM, in cases where availability requirements are imposed, can provide value beyond the commonly articulated failure avoidance and minimization of lost remaining useful life.

The determination of *ILT* for a specific availability requirement was provided as a demonstration of the design for availability methodology operation; where the *ILT* was used as an example of system parameters affecting either uptime or downtime. The methodology can be applied to determine any system parameters that can be explicitly related to the timeline downtimes or uptimes, for a contract availability requirement.

A demonstration of the derivation of the reliability (*TTF*), as a parameter affecting both uptime and downtime, is provided. The demonstration shows how the minimum reliability of a system or subsystem could be determined, to meet a specific availability requirement. This reliability information could be crucial to availability contracts and to any system with high availability requirement.

The reliability analysis, for a data-driven PHM approach versus an unscheduled maintenance approach, shows that the computed minimum reliability to meet a specific availability requirement is explicitly dependent on the PHM approach used to maintain the system. The analysis also shows that each PHM approach produces a different life cycle cost. Basically, for the same availability requirement a system would require different reliability management based on the adopted maintenance policy. The design for availability application results demonstrate not only deriving the system parameters that are necessary to meet a specific availability requirement, but also provide a critical tool to understand the impact of a PHM implementation on each system parameter. In the case study examples, the PHM data-driven case has produced a lower life cycle cost compared to the unscheduled maintenance case. This is caused by: 1) the ability to repair versus replace, 2) the number of spares required to support the system, and 3) the discount rate on money. The cost analysis reflects the complexity of a true understanding of a PHM implementation and its impact on the life cycle management of the system. Only by adopting a complete approach that takes into consideration all system design, support and logistics parameters, that a realistic assessment of a PHM implementation could be performed.

5.2. Contributions

The research work presented in this dissertation makes the following contributions:

1) Creation of the first general "design for availability" methodology that is not a "search-based" method and is capable of calculating unknown system parameters directly from an availability requirement when the inputs are uncertain and the availability requirement is represented as a probability distribution.
    a. The new methodology was demonstrated on system parameters affecting either downtime or uptime, e.g., logistics parameters.
    b. The new methodology was demonstrated on system parameters affecting both uptime and downtime, e.g., reliability.
2) Integration of the design for availability method into the process of designing PHM into systems. This integration provides a key means to quantify application-specific PHM implementation impacts on specific system parameters – a capability that has not been previously available
3) Creation of a methodology for performing life cycle cost (and ROI) versus availability tradeoffs for systems that incorporate PHM. This

dissertation is the first reported work to quantitatively produce life cycle cost (and ROI) versus availability for systems that incorporate PHM.

5.3. Potential Broader Impacts of this Work

The design for availability methodology, applied to PHM systems, provides the possibility for significant new capability to: a) perform (in conjunction with prognostics and health management) real-time pro-active availability analysis; b) determine requirements flow down for the development of prognostics and system health management and flow down to the supply chain; and c) perform pro-active reliability versus logistics tradeoffs, and assess the cost and resources required to deliver and support systems subject to availability contracts (e.g., Performance-Based Logistics contracts).

This method will enable the use of advances in the detection of performance anomalies and degradation of systems (including prognostics), to assess (and mitigate) logistics risks that result in system downtime. Providing health assessment and advanced warning of impending failure coupled with real-time design for availability control enables decision support actions that when communicated to maintenance and logistics operations will insure timely forecasting of maintenance and logistics actions that meet required availability levels, while providing valuable feedback to the design process.

The methodology is part of a disciplined supportability analysis strategy that could be applied early in the system development process, thus exerting influence on the system (and system supportability) design by suggesting where appropriate PHM monitors and data collection mechanisms should be included in the design.

5.4. Future Work

Future work could extend the design for availability methodology to address the concurrent determination of multiple system parameters that are dependent, i.e., if a relationship between the unknown system parameters is known. However, when the unknown system parameters are independent, the inclusion of an optimization approach may be required; since the relationship between the imposed downtime (or uptime) and the unknown system parameters could accept more than one unique solution. But, even in this case (i.e., multiple independent system parameters), the methodology is still efficient in terms of reducing the large and complex optimization problem, i.e., determining the unknown system parameters for multiple non-identical downtime (or uptime) values that generate different availability quantities, which may or may not satisfy the availability requirement; to determining the unknown system parameters for a single downtime (or uptime) value that has been imposed to satisfy the availability requirement.

The work presented in this dissertation could be extended by the inclusion of a redundancy analysis. In other words, analyze cases where a unit's failure does not necessarily generate an operational downtime of the system, since other redundant units could substitute the failed unit and maintain the ordinary operation of the system, while the failed unit is being repaired or replaced. This will directly affect the availability of the system (i.e., improve the availability), since the downtime duration will be reduced because of the immediate accessibility to the redundant units. Therefore, the design for availability derivation process, to determine the unknown system parameters, will be different for systems with redundant units.

This dissertation is focused on availability contracts that consist of meeting an operational availability requirement; this is indeed the general case (i.e., operational availability). Since the operational availability is the actual system availability that the customer sees, it implicitly incorporates other forms of availability and it is the most commonly used form of availability in this type of contracts. However, future work could extend the design for availability methodology to incorporate other types of availability (e.g., instantaneous, mean, steady-state, inherent and achieved). The main difference between these forms of availability is the type of activities that are excluded or included in the accumulated downtime and uptime values. This means when applying the design for availability methodology, using the same system design and support parameters to meet a specific availability requirement, the imposed downtimes and uptimes requirement is going to be different based on the considered type of availability.

A conservative approach has been adopted, since the availability requirement was satisfied at any time during the entire system's life. Future work could consider satisfying the availability requirement at specific time periods. Since different contracts could define an availability requirement over different periods of time. In this case, the application of the methodology would be similar to the conservative case (i.e., satisfying the availability requirement at all times), with the difference of imposing the uptime and downtime requirements at a contract-specified periods of time, instead of imposing the uptime and downtime requirement after every downtime.

Appendix

This Appendix presents additional example cases for the demonstration and verification of the design for availability methodology.

The design for availability methodology will be demonstrated for the derivation of two different logistics parameters: 1) spares threshold (*ST*) and 2) replenishment spares (*RS*). Where *ST* is the minimum quantity of held spares, and *RS* is the number of replenishment spares ordered at the at *ST* value. Similar to Chapter 4, in order to use this demonstration as a qualitative verification of the methodology, I will perform the following steps:

- Using the availability distribution requirement as an input to the design for availability model, determine the required *ST* distribution (output).
- Use the generated *ST* distribution as an input to the existing PHM ROI simulation (described in the introduction to this section) to predict an availability distribution as an output.
- Compare the availability distribution input requirement to the availability distribution determined as an output.

Notice that the first step is sufficient to achieve the design for availability task, since the *ST* will be determined for a specific contract availability requirement. The second and third steps are for verification of the methodology. The same three-step process is adopted for the demonstration and verification of the methodology applied to the *RS*. A detailed description of how *ST* and *RS* affect the operational timeline is provided in Section 4.2.

Spares Threshold

All case study inputs are provided in Section 4.1, including LRU description, implementation and maintenance costs, and operational profile.

The reliability information and inventory management parameters are provided in Figure A.1 and Table A.1, respectively. Table A.1 summarizes the spares inventory (per socket) assumptions that are used for this specific example. Also, note that the spares carrying costs are incorporated into the LRU recurring costs. Figure A.1 shows the assumed reliability for this case example, i.e., time-to-failure (TTF), of the LRU based on [43] and [52].

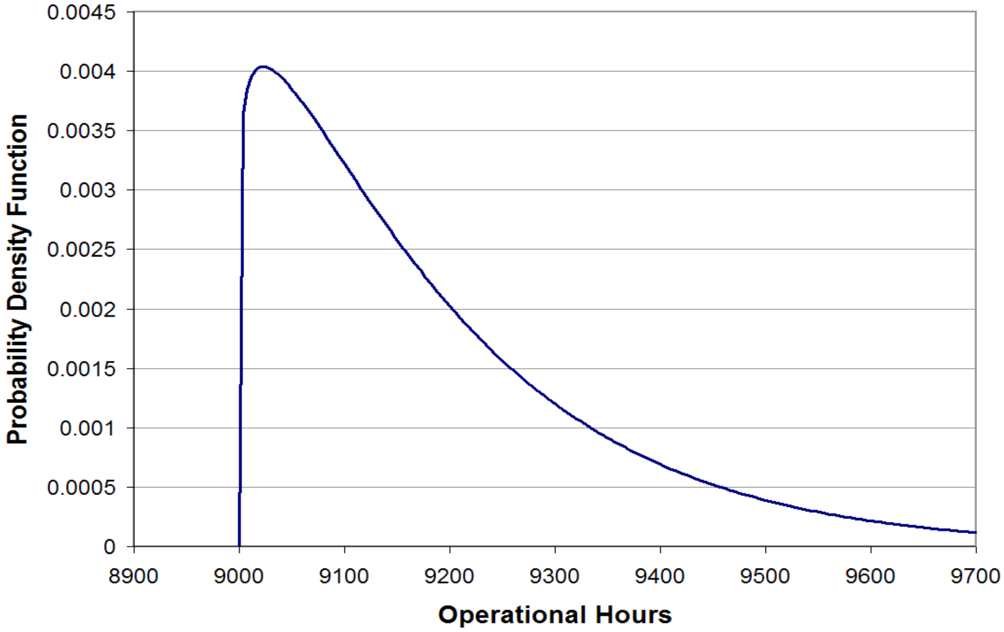


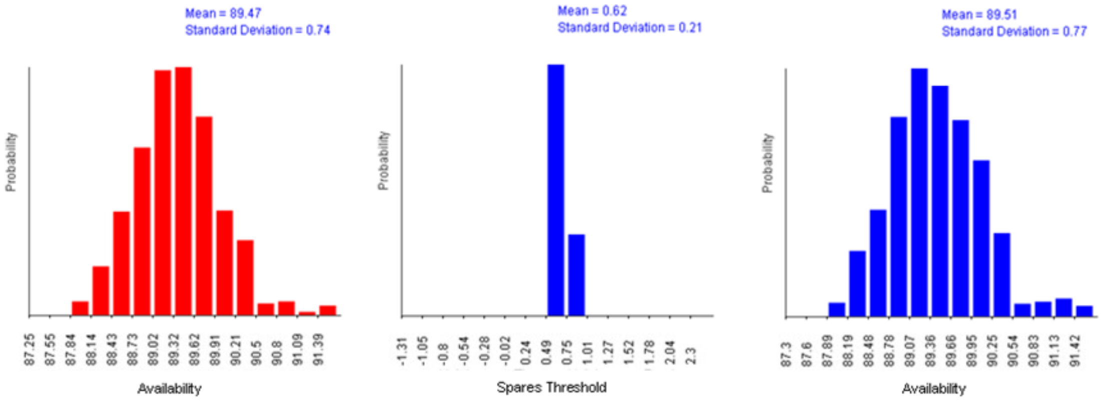
Figure A.1. Weibull distribution of TTFs (β=1.1, η=200 and γ=9000).

Table A.1. Spares Inventory

Factor	Quantity
Initial spares purchased for each socket	4
Threshold for spare replenishment	*Solved for in this section case study*
Number of spares to purchase per socket at replenishment	3
Spare replenishment lead time	27 calendar months
Spares carrying cost	10% of the beginning of year inventory value per year
Billing due date when ordering additional spares	2 years from purchase date

The availability distribution considered for this example case is shown in Figure A.2.a. This distribution could represent the requirement of an availability contract. Note, availability contracts may specify the availability requirement as a single value, but, to accommodate more general problems, for all example cases I will use availability requirements that are represented as a probability distribution.

Figure A.2.b shows the *ST* required to meet the availability requirement determined using the design for availability methodology.



(a) (b) (c)

Figure A.2. (a) Required availability distribution (input to the model). (b) Computed spares threshold (*ST*), for a data-driven PHM approach. (c) Availability probability distribution generated using the computed *ST* (Figure A.2.b).

In order to qualitatively verify the methodology, the spares threshold *ST* distribution (Figure A.2.b) was used as an input to the PHM ROI model. The PHM ROI model used the spares threshold *ST* distribution along with the other inputs (see Section 4.1) and generated a resulting availability distribution. Figure A.2.a shows the original availability input (mean = 89.47% and standard deviation = 0.74%), while Figure A.2.c shows the availability prediction that resulted from the PHM ROI model (mean = 89.51 and standard deviation = 0.77). The two results are not expected to be absolutely identical (since this is a stochastic solution), but the means and standard deviations are very similar. This simple example case validates the methodology and demonstrates qualitatively that the design for availability approach is truly satisfying the input availability requirement.

Replenishment Spares

All case study inputs are provided in Section 4.1, including LRU description, implementation and maintenance costs, and operational profile.

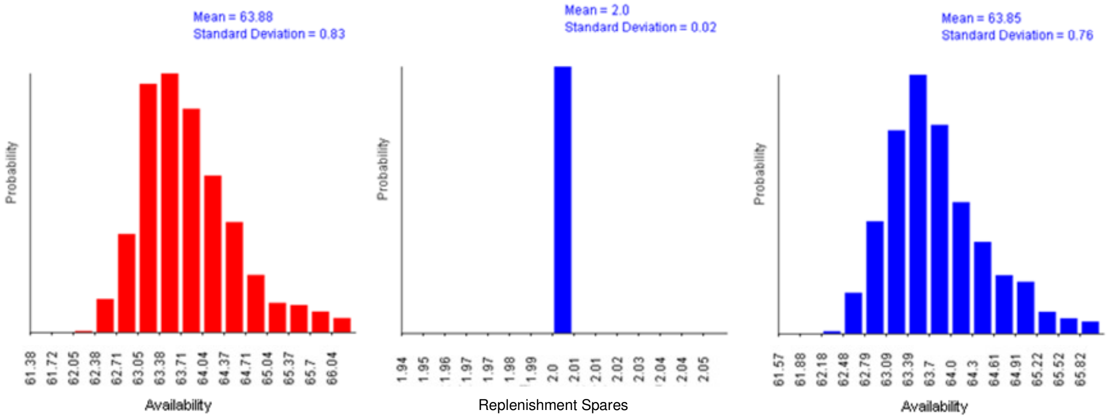
The reliability information and inventory management parameters are provided in Figure A.1 and Table A.2, respectively. Table A.2 summarizes the spares inventory (per socket) assumptions that are used for this specific example. Also, note that the spares carrying costs are incorporated into the LRU recurring costs. Figure A.1 shows the assumed reliability for this case example, i.e., time-to-failure (TTF).

Table A.2. Spares Inventory

Factor	Quantity
Initial spares purchased for each socket	3
Threshold for spare replenishment	≤ 1 spares in the inventory per socket
Number of spares to purchase per socket at replenishment	*Solved for in this section case study*
Spare replenishment lead time	36 calendar months
Spares carrying cost	10% of the beginning of year inventory value per year
Billing due date when ordering additional spares	2 years from purchase date

The availability distribution considered for this example case is shown in Figure A.3.a. This distribution could represent the requirement of an availability contract. Note, availability contracts may specify the availability requirement as a single value, but, to accommodate more general problems, for all example cases I will use availability requirements that are represented as a probability distribution.

Figure A.3.b shows the *ST* required to meet the availability requirement determined using the design for availability methodology.



(a) (b) (c)

Figure A.3. (a) Required availability distribution (input to the model). (b) Computed replenishment spares (*RS*), for an unscheduled maintenance policy. (c) Availability probability distribution generated using the computed *RS* (Figure A.3.b).

In order to qualitatively verify the methodology, the replenishment spares *RS* distribution (Figure A.3.b) was used as an input to the PHM ROI model. The PHM ROI model used the replenishment spares *RS* distribution along with the other inputs (see Section 4.1) and generated a resulting availability distribution. Figure A.3.a shows the original availability input (mean = 63.88% and standard deviation = 0.83%), while Figure A.3.c shows the availability prediction that resulted from the PHM ROI model (mean = 63.85 and standard deviation = 0.76). The two results are not expected to be absolutely identical (since this is a stochastic solution), but the means and standard deviations are very similar. This simple example case validates the methodology and demonstrates qualitatively that the design for availability approach is truly satisfying the input availability requirement.

Bibliography

[1] D. Kececioglu, *Maintainability, Availability, Operational Readiness, Engineering Handbook, Volume 1*, Prentice Hall PTR, 1995.

[2] F. E. Oliveto, "An Optimal Sparing Model for the Operational Availability to Approach the Inherent Availability," in *Proceedings of the Annual Reliability and Maintainability Symposium*, 2001, pp. 252-257.

[3] Y. Macheret, P. Koehn, and D. Sparrow, "Improving reliability and operational availability of military systems," in *Proceedings of the IEEE Aerospace Conference*, 2005.

[4] I. C. L. Ng, R. Maull, and N. Yip, "Outcome-based Contracts as a driver for Systems thinking and Service-Dominant Logic in Service Science: Evidence from the Defence industry", *European Management Journal*, vol. 27, no. 6, pp. 377-387, December 2009.

[5] BAE 61972 BAE Annual Report (2008).

[6] R. L. Beanum, "Performance - Based logistics and contractor support methods," in *Proceedings of the IEEE Systems Readiness Technology Conference (AUTOTESTCON)*, September 2006.

[7] W. A. Hyman, "Performance-based contracting for maintenance," NCHRP Synthesis 389, ISSN:0547-5570, 2009.

[8] M. Pecht, *Product Maintainability Supportability Handbook*, 2nd Edition, CRC Press, 2009.

[9] M. G. Pecht, Prognostics and Health Management of Electronics, Wiley-Interscience, September 2008.

[10] S. V. Amari and L. McLaughlin, "Optimal design of a condition-based maintenance model", Reliability and Maintainability, 2004 Annual Symposium – RAMS, pp. 528-533, January 2004.

[11] F. Nilsson, J. Waidringer, "Toward Adaptive Logistics Management," Proceedings of the 38th Annual Hawaii International Conference on System Sciences, pp. 81c, January 2005.

[12] E. C. Mercado, *Hands-On Inventory Management*, Auerbach Publications, 2007.

[13] B. Hruz and M. C. Zhou, *Modeling and Control of Discrete-Event Dynamic Systems*, Springer, 2007.

[14] G. S. Fishman, *Discrete-Event Simulation: Modeling, Programming, and Analysis*, Springer, 2001.

[15] M. Bazargan and R. N. McGrath, "Discrete event simulation to improve aircraft availability and maintainability," in *Proceedings of the Annual Reliability and Maintainability Symposium*, 2003, pp. 63-67.

[16] L. Yin, R. M. Fricks, and K. S. Trivedi, Application of Semi-Markov Process and CTMC to Evaluation of UPS System Availability, in Proceedings of the Annual Reliability and Maintainability Symposium, 2002, pp. 584-591.

[17] P. Choonhapran and G. Balzar, Availability of HV Circuit-Breakers: The Application of Markov Model, in Proceedings of the Power Engineering Society General Meeting, 2007.

[18] T. DelSole, A Fundamental Limitation of Markov Models, Journal of the Atmospheric Sciences, Vol. 57, pp. 2158-2168, 2000.

[19] A. S. Noetzel, The Reduction of a Discrete Event Simulation to a Markov Chain, Proceedings of the Winter Simulation Conference, Dec. 1979, pp. 391-397.

[20] J. R. Clymer, Simulation-Based Engineering of Complex Systems, Wiley-Inter-Science, 2nd Edition, 2009.

[21] Y. Min, H. Zhengyou, and Q. Qingquan, Fuzzy Availability Based on Affine Arithmetic and Markov Process, in Proceedings of the Second International Conference on Intelligent Computation Technology and Automation, 2009, pp. 704-707.

[22] A. Wood, Availability Calculations with Exhaustible Spares, IEEE Transactions on Reliability, Vol. 38, No. 3, 1989, pp. 388-391.

[23] (Yang and Nachlas 2001) S-C. Yang and J. A. Nachlas, Bivariate Reliability and Availability Modeling, IEEE Transactions on Reliability, Vol. 50, No. 1, 2001, pp. 26-35.

[24] O. O. Bankole, R. Roy, E. Shehab, P. Wardle, Affordability Assessment of Industrial Product-Service System in the Aerospace Defense Industry, proceedings of the 1st CIRP Industrial Product-Service Systems (IPS2) Conference, Cranfield University, 1-2 April 2009, pp. 230.

[25] R. H. Yeh and W. L. Chang, Optimal threshold value of failure-rate for leased products with preventive maintenance actions, Mathematical and Computer Modeling, Vol. 46, 2007, pp. 730-737.

[26] K. Feldman, *Electronic Prognostics and Health Management: A Return on Investment Analysis*, Master of Science Thesis, Mechanical Engineering, University of Maryland, May 2008, URL: http://hdl.handle.net/1903/8323.

[27] P. A. Sandborn and C. Wilkinson, "A Maintenance Planning and Business Case Development Model for the Application of Prognostics and Health Management (PHM) to Electronic Systems," *Microelectronics Reliability*, vol. 47, no. 12, pp. 1889-1901, December 2007.

[28] C. Valdez-Flores and R. Feldman, "A Survey of Preventative Maintenance Models for Stochastically Determining Single-Unit Systems," *Naval Research Logistics*, Vol. 36, pp. 419–446, 1989.

[29] D. Cho and M. Parlar, "A Survey of Preventative Maintenance Models for Multi-Unit Systems," *European Journal of Operational Research*, Vol. 51, pp. 1–23, 1991.

[30] W. Wang, "A Model to Determine the Optimal Critical Level and the Monitoring Intervals in Condition-Based Maintenance," *International Journal of Production Research*, Vol. 38, No. 6, pp. 1425–1436, 2000.

[31] A. Barros, C. Berenguer, and A. Grall, "Optimization of Replacement Times Using Imperfect Monitoring Information," *IEEE Transactions on Reliability*, Vol. 52, No. 4, pp. 523–533, 2003.

[32] G. Heinrich and U. Jensen, "Bivariate Lifetime Distributions and Optimal Replacement," *Mathematical Methods of Operations Research*, Vol. 4, pp. 31–47, 1996.

[33] T. Raivio, E. Kuumola, V. A. Mattila, K. Virtanen, and R. P. Hämäläinen, “A Simulation Model for Military Aircraft Maintenance and Availability,” *Proceedings of the European Simulation Multiconference*, pp. 190–194, September 2001.

[34] L. Warrington, J. A. Jones and N. Davis, “Modeling of Maintenance, within Discrete Event Simulation,” *Proceedings of the Reliability and Maintainability Symposium (RAMS)*, Seattle, WA, pp. 260–265, January 2002.

[35] Y. Lin, A. Hsu, and R. Rajamani, “A Simulation Model for Field Service with Condition-Based Maintenance,” *Proceedings of the Winter Simulation Conference*, San Diego, CA, pp. 1885–1890, December 2002.

[36] J. J. Luna, “A probabilistic model for evaluating PHM effectiveness,” *Proceedings of the International Conference on Prognostics and Health Management*, Denver, CO, October 2008.

[37] J. J. Luna, “Metrics, Models, and Scenarios for Evaluating PHM Effects on Logistics Support,” *Proceedings of the 1st Annual Conference of the Prognostics and Health Management Society*, San Diego, CA, Oct 2009.

[38] C. Byington, M. Watson, and D. Edwards D., “Data-Driven Neural Network Methodology to Remaining Life Predictions for Aircraft Actuator Components,” IEEE Aerospace Conference, Big Sky, MT, March 2004.

[39] S. Mishra, S. Ganesan, M. Pecht, J. Xie, "Life consumption monitoring for electronics prognostics," IEEE Aerospace Conference Proceedings, volume 5, pp. 3455-3466, December 2004.

[40] N. Vichare, P. Rodgers, and M. Pecht, “Methods for Binning and Density Estimation of Load Parameters for Prognostics and Health Management,” *International Journal of Performability Engineering*, Vol. 2, No. 2, pp. 149–161, 2006.

[41] A. Khalak and J. Tierno, “Influence of Prognostics Health Management on Logistics Supply Chain,” *Proceedings of the American Control Conference*, Minneapolis, MN, pp. 3737-3742, June 2006.

[42] K. Feldman, T. Jazouli, and P. Sandborn, “A Methodology for Determining the Return on Investment Associated with Prognostics and Health Management,” *IEEE Trans. on Reliability*, vol. 58, no. 2, pp. 305-316, June 2009.

[43] E. Scanff, K. Feldman, S. Ghelam, P. Sandborn, M. Glade, and B. Foucher, “Life cycle cost estimation of using prognostic health management for helicopter avionics,” *Microelectronic Reliability*, vol. 47, no. 12, pp. 1857-1864, December 2007.

[44] http://www.prognostics.umd.edu/membersOnly/ROITool/PHM_ROI.htm

[45] G. Janakiraman, J. R. Santos, and Y. Turner, “Automated System Design for Availability,” in *Proceedings of the International Conference on Dependable Systems and Networks (DSN'04)*, 2004.

[46] H. F. Castro and K. L. Cavalca, “Availability optimization with genetic algorithm,” *Reliability Engineering & System Safety*, vol. 91, no. 4, pp. 413-420, April 2006.

[47] D. M. Raffo and S. Setamanit, “Supporting Software Process Decisions Using Bi-directional Simulation,” *International Journal of Software Engineering and Knowledge Engineering (IJSEKE)*, vol. 13, no. 5, pp. 513-530, 2003.

[48] A. P. Reynolds and G. P. McKeown, “Construction of factory schedules using reverse simulation,” *European Journal of Operational Research*, vol. 179, no. 3, pp. 656-676, June 2007.

[49] A. Henkle, C. Lindsey, and M. Bernson, "Southwest Airlines: A review of the operational and cultural aspects of Southwest Airlines," Operations Management Course Presentation, Sloan School of Management, Summer 2002.

[50] Federal Aviation Administration, "Investment Analysis Benefit Guidelines: Quantifying Flight Efficiency Benefits," Version 3.0, Investment Analysis and Operations Research Group, June 2001.

[51] A. Henkle, C. Lindsey, and M. Bernson, *Southwest Airlines: A review of the operational and cultural aspects of Southwest Airlines*, Operations Management Course Presentation, Sloan School of Management, 2002.

[52] D. Kumar, J. Crocker, J. Knezevic, and M. El-Haram, Reliability, Maintenance and Logistic Support: A Life Cycle Approach, Springer, 2000.