Using PHM to Meet Availability-Based Contracting Requirements

Taoufik Jazouli and Peter Sandborn
CALCE Center for Advanced Life Cycle Engineering
Department of Mechanical Engineering
University of Maryland, College Park, MD 20742
tjazouli@calce.umd.edu
sandborn@calce.umd.edu

***Abstract*—"Availability-based" contracting originated because customers with high availability requirements are in many cases interested in migrating from buying the actual system to buying the availability of the system. A well-known example of availability-based contracting is Performance Based Logistics (PBL).**

Prognostics and Health Management (PHM) methods are incorporated into systems to avoid unanticipated failures that can potentially impact system safety, result in additional life cycle cost, and/or adversely affect the system availability. While predicting the availability of a system based on known or predicted system parameters is relatively straightforward and can be accomplished using existing methods; determining the system parameters that result in a desired availability is not and is generally performed using "brute force" search-based methods that become quickly impractical for designing systems with more than a few variables and when uncertainties are present.

This paper presents the application of PHM within a "design for availability" approach that uses an availability requirement to predict the required logistics, design (including reliability) and operation parameters with and without the application of PHM methods. A life cycle cost analysis is used to quantify trade-offs of using PHM methods versus more traditional maintenance approaches in the context of availability contracts.

Keywords-component; PHM, availability-based contracts, performance-based contracts, performance-based logistics (PBL), life cycle costing, modeling and simulation.

I. Introduction

A Prognostics and Health Management (PHM) implementation is adopted to provide advanced warning of unanticipated failures, enhance the decision making process for maintenance planning, enhance the real-time assessment of system's health, lower sustainment costs, provide product usage feedback into the product design process and potentially improve availability. Thus, a successful PHM implementation would impact the availability of the system. An understanding of the relationship between PHM methods and system's availability is critical in systems with a high-availability requirement. In availability-centric systems (e.g., power generation, product lines, security or safety systems, critical infrastructure, commercial aircraft, weapons systems, etc.), a drop in availability could result in catastrophic loss of life or money.

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 (frequency of a failure) and maintainability (if an item does fail, how quickly could it be restored to operation). Several different types of availability can be measured for repairable systems (e.g., inherent, achieved, operational, etc.) [1]. However, this analysis is focused on the operational availability since it implicitly incorporates other forms of *downtime*-based availability and it is the most commonly used form of availability specified in availability contracts, however, the proposed design for availability methodology is general and could be easily extended 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_O) accounts for all types of maintenance and logistics *downtimes*. It is computed as the ratio of the accumulated *uptime* to the sum of the accumulated *uptime* and *downtime*,

$$A_o = \frac{uptime}{uptime + downtime} \qquad (1)$$

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. *Downtime* is generated when the system is down and not operating when requested due to repair, replacement, preventative maintenance, 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. Other types of *downtime*-based availability differ from operational availability in the specific activities that are included in the *uptimes* and *downtimes* [4].

978-1-4244-9826-0/11/$26.00 ©2011 IEEE

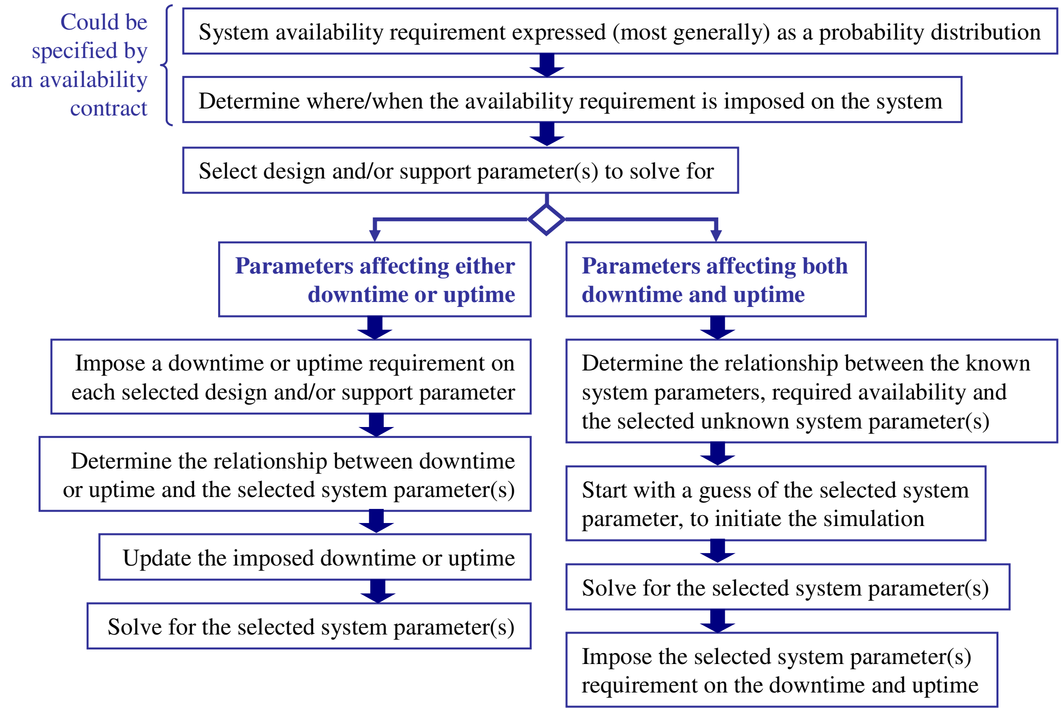


Figure 1. Design for Availability Methodology.

Customers of availability-centric systems, are in many cases interested in buying the availability of a system, instead of actually buying the system itself; or the amount of money that the system manufacturer or system provider is paid for is a function of the availability achieved. This approach illustrates the concept of "availability-based" contracting. A detailed explanation of these types of contracts, and similar types of contracting (e.g., outcome-based contracting [5] and performance-based logistics [6,7]), is provided in [8]. Generally, Availability-based contracts specify an availability requirement that has to be met at all times, or during specific time periods, throughout the system's operational life. Both, the customer and the system manufacturer or provider, have the challenging task of appropriately pricing the contract and determining the necessary system's design and support parameters to meet the availability requirement specified in the contract. To support these activities the concept of "design for availability" has been introduced, which is a methodology that uses a specific availability requirement (as an input) to determine the required logistics, design and operations parameters (as outputs). A detailed description of the methodology and its application is provided in [8]. The preliminary work appearing in [8] was limited to the system parameters affecting either *downtime* (i.e., when the system is down undergoing a repair or waiting for spares) or *uptime* (i.e., when the system is up and running), but not both. In this paper, the application of the methodology has been extended and generalized to include the prediction of system parameters concurrently affecting both *downtime* and *uptime*, to meet a specific availability requirement.

The derivation and application of the design for availability methodology that is used to determine the necessary system parameters (e.g., reliability) to meet an availability requirement are presented in this paper. The next section describes the design for availability approach and the required steps for the application of the approach. Section III illustrates the application of the methodology to determine the reliability of the system (i.e., time-to-failure) for simple case study examples for systems with and without PHM subject to an availability requirement. The example application includes a life cycle cost and return on invest analysis.

II. Design for Availability Approach

Using today's models and tools, availability is computed based on known or estimated system reliability, operational and logistics parameters, etc. However, while determining the availability that results from a set of system parameters and/or events is straightforward, determining the system parameters and/or events that result in a desired availability is not. The goal of the work presented in this paper is to reverse the problem. In other words, to create a methodology that determines the necessary system reliability, operational parameters, and/or logistics management parameters to meet a specific availability requirement; i.e., a "design for availability" methodology.

The methodology introduced uses the availability requirement to compute and impose the necessary *uptimes* and *downtimes* throughout the system's life. Then, it uses these imposed *uptime* and *downtime* values to solve for the unknown system parameters. 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*. Figure 1 shows the steps to formulate and execute the design for availability solution for both types of parameters. Previous work [8] demonstrates the application of the methodology to parameters affecting either *uptime* or *downtime*. The work presented in this paper is focused on the derivation of system parameters that concurrently affect both *uptime* and *downtime*.

Details for each step of the process of determining system parameters concurrently affecting both *uptime* and *downtime* (for a specific availability requirement) are discussed qualitatively in the following subsections. An example application of the process is provided in Section III.

A. 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.). Availability requirements, although often expressed as a single value, really generally represents 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.

B. Determine Where/When the Availability Requirement is Imposed

To generalize the design for availability model, a conservative approach is adopted 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 could be less conservative). For the remainder of this discussion it is assumed 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. Notice that if the availability requirement is interpreted as a probability distribution, then the availability of every member of the population should satisfy a sampled value from the availability distribution requirement. However, if the availability requirement is interpreted as a single minimum value, then the availability of every member of the population should satisfy this availability single minimum value throughout the entire support life.

By analyzing the A_O variations based on (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*. 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 the approach is to impose the availability requirement at the end of every *downtime*.

C. Select 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.

D. Determine the Type of Parameter

As mentioned previously, there exist two types of system parameters: parameters affecting either *uptime* or *downtime* (not both), and parameters concurrently affecting both *uptime* and *downtime*.

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*. This version of the methodology is only applicable to system parameters that explicitly affect either *uptime* or *downtime*, but not both. A detailed description and treatment of this type of parameters was provided in [8].

The focus of this paper is on the other type of system parameters, i.e., 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, a *downtime* or *uptime* requirement cannot be exclusively imposed as described in the previous paragraph and in [8]. 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.

E. Determine the Relationship Between the Known System Parameters, Required Availability and the Selected Unknown System Parameter

As explained in the previous subsection, 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, there are three unknown quantities: 1) the unknown system parameter, 2) the *downtime* and 3) the *uptime*. Therefore, three relationships need to be established in order to solve for the three unknown quantities. Since these types of system parameters explicitly affect 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 affect 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, three relationships have been defined, with three unknowns (the unknown system parameter, the *downtime* and the *uptime*), thus the unknown system parameter, *uptime*, and *downtime*, can be solved for.

F. 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 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.

G. 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. Basically, the selected unknown system parameter is solved for or imposed using the known system parameters and the availability requirement.

H. 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 design variable and continually iterating the entire process until the availability requirement is met. And 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.).

III. APPLICATION OF THE METHODOLOGY: PREDICTING THE RELIABILITY

In this section, the design for availability methodology will be demonstrated for two system sustainment approaches: unscheduled maintenance and data-driven PHM [9], with the same availability distribution requirement (input). The objective is to determine the minimum[1] allowable reliability, i.e., time-to-failure (*TTF*), of the LRUs to meet the availability requirement. In this example, 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.

[1] 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 (a socket is an instance of an installation location for an LRU) 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.

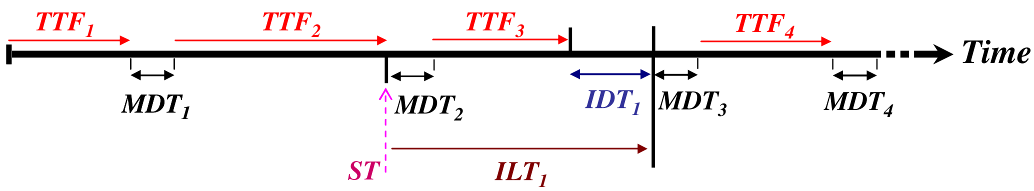


Figure 2. *TTF* implication on the operational timeline.

Notice that the *TTF* is the type of system parameters that concurrently affects both *uptime* and *downtime*. For example, consider the following scenario: the inventory is out of spares, the replenishment spares will be delivered one year from now, and the system is using last spare available from the inventory. The system will be up and running as long as this spare 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 (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 this spare. This simple scenario demonstrates how the *TTF* of the LRUs could affect both the *uptime* and *downtime*.

To demonstrate and verify the derivation of the *TTF* for a specific availability requirement, the design for availability methodology has been implemented within a PHM Return on Investment (PHM ROI) tool. The PHM ROI tool is a discrete event simulation that follows a population of sockets (a socket is an instance of an installation location for an LRU) throughout their entire support life from the first line replaceable unit (LRU) installation in the socket to the retirement of the socket. The tool determines the life cycle cost, return on investment and availability impacts associated with putting PHM structures into systems. The PHM ROI tool includes the modeling of other quantities as well (e.g., operational profile, false positives, cost of money, inventory management, etc). A detailed description of the PHM ROI tool is provided in [9, 10].

In order to use the application of the methodology on *TTF* as a qualitative verification of the application of the methodology, the following three steps are performed: 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 (described in the introduction to this section) 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.

Appendix A to this paper provides all the case study inputs, including LRU description, implementation and maintenance costs, operational profile and inventory management parameters. Notice that the *TTF* information is not provided in Appendix A, since it is an unknown quantity that needs to be determined based on the availability requirement, using the design for availability methodology.

A. *Reliability (TTF) Derivation*

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. 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 [9].

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 2 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* 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,

$$\sum UT = (IS)(TTF) + \sum (RS)(TTF) \qquad (2)$$

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

$$\sum DT = \sum IDT + \sum MDT = \sum (ILT - (ST)(TTF)) + \sum MDT \qquad (3)$$

Notice that the summations in equations (2) and (3) 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.,

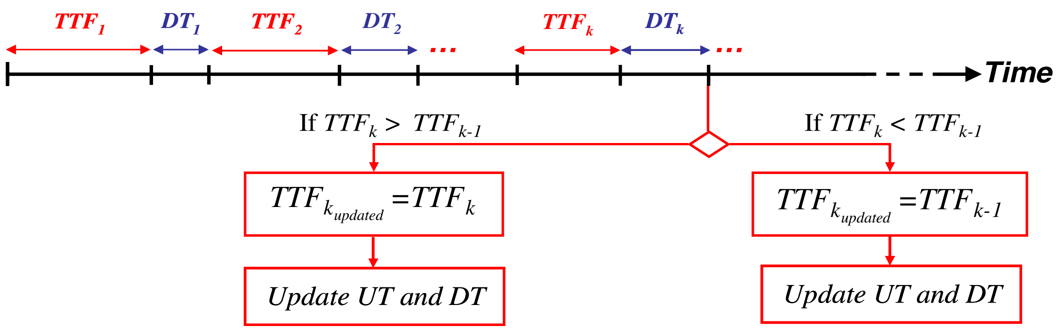


Figure 3. Updating the *TTFs*, *UTs* and *DTs*.

system parameter, could result in a different event outcome. A detailed explanation is provided in Section II.

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

$$A_o = \frac{\sum UT}{\sum UT + \sum DT} \qquad (4)$$

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

For example, the kth *TTF* value could be derived by combining equations (2), (3) and (4). The kth *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 (5) do not refer to analytical summations, but to the accumulation of events outcomes and sequences. Therefore, the right side of equation (5) does not explicitly include the "*k*" subscript,

$$TTF_k = \frac{\sum ILT + \sum MDT}{\frac{1-A_o}{A_o}\left(IS + \sum RS\right) + \sum ST} \qquad (5)$$

Notice that equations (2)-(5) 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 set up 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 3 illustrates the process of updating the computed *TTF*s. Basically, after every *downtime*, the computed *TTF* is compared to the previous value, if the current value is greater than the previous one, 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

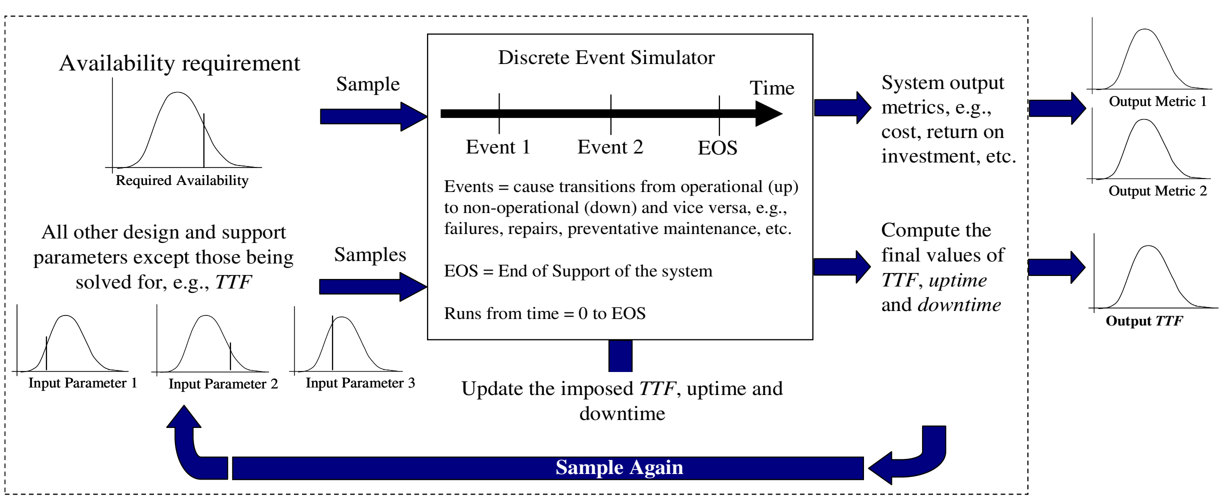


Figure 4. Solution process.

through equations (2) and (3). Finally, the model uses the updated *TTFs*, *uptimes*, and *downtimes* 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 shows the process used to generate a distribution of system parameter values using a discrete event simulator (the PHM ROI tool 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.

B. *Application: Unscheduled Maintenance vs. Data-Driven PHM*

A detailed description of the inputs data used for this example is provided in Appendix A, for both unscheduled maintenance and data-driven approaches. 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 [9]. This analysis has resulted an optimal prognostic distance of 600 operational hours (see Figure 5).

The right vertical axis in Figure 5 corresponds to the mean value of the allowable minimum *TTF* distribution corresponding to each prognostic distance. Since, for each prognostic distance there is a corresponding allowable minimum *TTF* distribution and life cycle cost distribution. However, the *TTF* and life cycle cost values shown on Figure 5 are the means of the generated *TTF* and life cycle cost distributions respectively.

While data-driven PHM could provide an opportunity to avoid unanticipated failures and perform more on-site scheduled repairs (since LRUs are maintained before they actually fail, i.e., a better chance to be repaired, rather than replaced or thrown away), a good understanding of the impact of a PHM implementation on the system parameters (e.g., maintenance planning and inventory management) is indispensable. One of PHM's key parameters is the prognostic distance, which could affect both the maintenance planning and inventory management. 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 *TTF*s 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*.

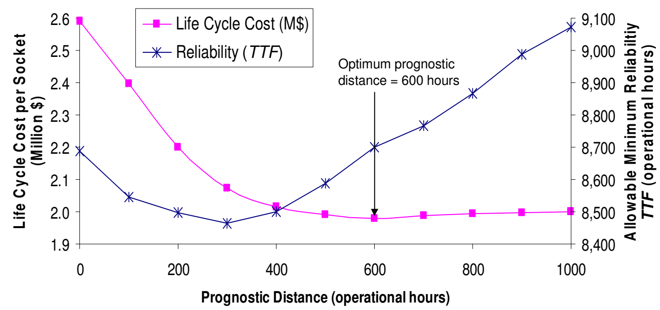


Figure 5. Variations of life cycle with data-driven PHM prognostic distance.

The availability requirement considered in this subsection is shown on Figure 6. This availability requirement has been used as an input to the design for availability model. Figures 7and 8 show the resulting *TTF* distributions using unscheduled maintenance and data-driven PHM. The *TTF* distributions were generated through the process illustrated in Figure 4 and using the input data provided in Appendix A. The qualitative verification process, of the design for availability methodology, is provided in Appendix B.

By comparing the resulting *TTF* distributions for unscheduled maintenance and data-driven PHM approaches (Figure 8), data-driven PHM has allowed a lower *TTF*

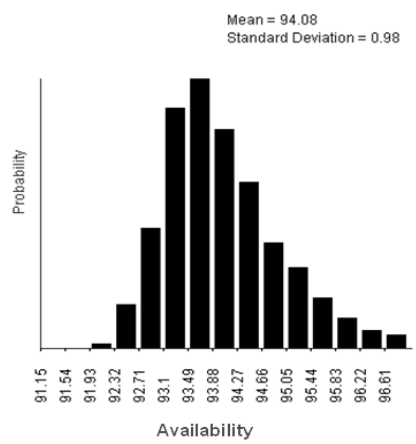


Figure 6. Required availability distribution (input to the model).

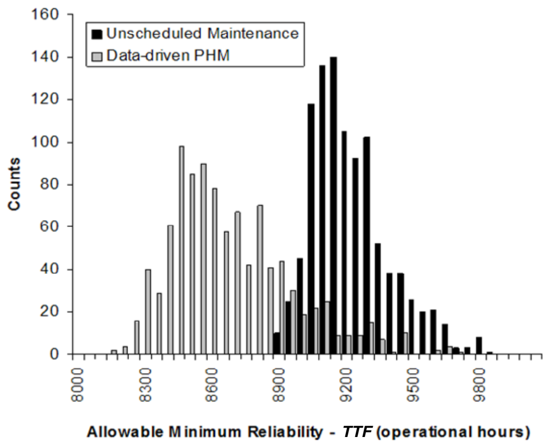


Figure 7. Computed *TTF* distribution for unscheduled maintenance and data-driven PHM.

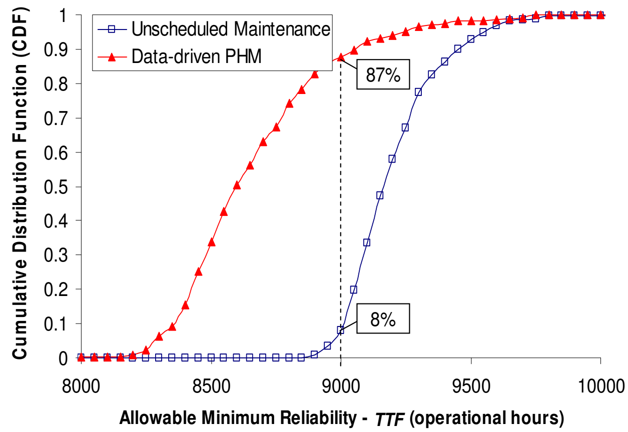


Figure 8. Computed *TTF* cumulative distribution function for unscheduled maintenance and data-driven PHM.

requirement. This means, in this example, 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 requirement 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 system's behavior.

Figure 8 shows how practically the *TTF* results could be 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 a data-driven PHM approach, and only 8% confident to meet the same availability requirement under an unscheduled maintenance approach.

C. Cost Analysis: Unscheduled Maintenance vs. Data-Driven PHM

Figure 9 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). The data-driven PHM case requires fewer spares throughout the support life of the system. This result is not intuitive. It is the unscheduled maintenance approach that would be expected to require the minimum number of spares; since the unscheduled maintenance events are usually performed upon the actual failure of the LRUs, thus maximizing the useful life of the LRUs, which results in the minimum number of spares. However, the counterintuitive result is primarily due to the ability to repair versus replace, i.e., early warning of failures in the data-driven PHM case provide an opportunity to schedule and perform on-site maintenance events that resulted in repairing most LRUs, because of the capability to intervene before a complete deterioration of the LRUs, while the system is not requested for operation (shorter and less expensive maintenance time). In the unscheduled maintenance case, most failures were resolved by replacing LRUs rather than repairing them.

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 9, 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 9 for unscheduled maintenance case during the replenishment events at approximately years 4, 7, etc. (frequent short, i.e., less expensive, inventory *downtimes*), compared to the data-driven PHM case large steps at approximately years 7, 12, etc. (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

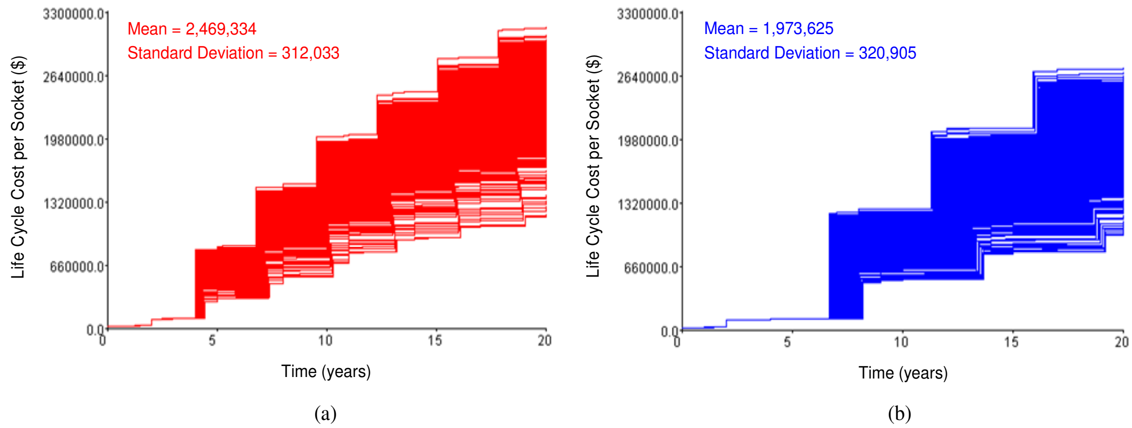


Figure 9. (a) Life cycle cost per socket for a data-driven PHM approach. (b) Life cycle cost per socket for an unscheduled maintenance approach.

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 the modeling of the cost associated with improving an LRU's reliability (i.e., *TTF*) was not included. Figure 7 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 6). 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.

D. Return on Investment (ROI) Analysis

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 out of the scope of this paper, however, an in-depth discussion of this methodology and its implementation are provided in [10].

Figure 10 shows the histogram of the computed ROI for 1000-socket population, using the inputs data provided in Appendix A. In this example, 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 the ROI values in Figure 10 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 Appendix A, could result in an economic loss, i.e., you could end up being worse off than unscheduled maintenance. Based on Figure 10, this example

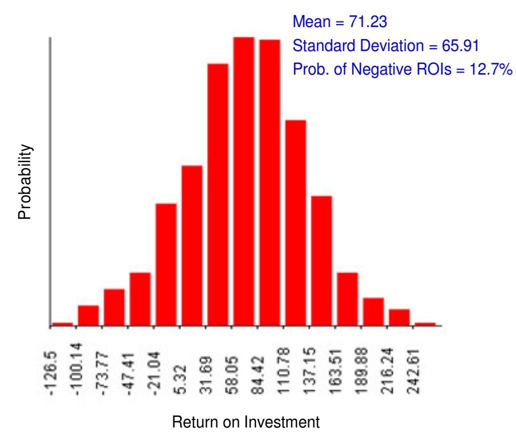


Figure 10. Histogram of ROI for a 1000-socket population.

predicts that a data-driven PHM approach would result in a positive ROI (cost benefit) with an 87.3% confidence.

IV. Discussion and Conclusion

This paper illustrates the application of the design for availability methodology to PHM systems. The methodology uses an availability requirement to determine the unknown system design and support parameters; the approach is general and could be applied to any type of system even when uncertainties are included. The results presented in this work are primarily focused on the system parameters concurrently affecting both *uptime* and *downtime*.

A demonstration of the derivation of the reliability (*TTF*), as a parameter affecting both *uptime* and *downtime*, is provided in detail. The step-by-step demonstration shows how the necessary 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 necessary 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. 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 example, the PHM data-driven case has produced a lower life cycle cost compared to the Unscheduled Maintenance case. This has been motivated 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.

The results presented in this paper and previous work [8] are focused on deriving single system parameters that are necessary to meet a specific availability requirement. However, the design for availability methodology could be extended to address the concurrent determination of multiple system parameters; which would involve two cases. First case, if system parameters are dependent, in other words if an explicit relationship between these unknown system parameters could be determined, then the application of the methodology is straightforward and sufficient to solve for the unknown dependent system parameters. Second case, if the system parameters are independent, then the inclusion of an optimization approach might be required at the conclusion of the 7th step (Section II) of the methodology; since the established relationships that are used to solve for the unknown system parameters might have more unknowns than the actual number of derived equations. However, even in this case (i.e., multiple independent unknown system parameters), the methodology is still efficient in terms of reducing the large and complex optimization "search-based" problem, where every generated set of system parameters may or may not satisfy the availability requirement; to a basic "non-search-based" problem where the unknown system parameters have to be optimized to satisfy one single relationship which is already predefined by the means of the design for availability methodology.

ACKNOWLEDGMENT

The authors would like to thank the more than 100 companies and organizations that support research activities at the Center for Advanced Life Cycle Engineering (CALCE) at the University of Maryland annually. Specifically, the CALCE Prognostics and Health Management Consortium members.

REFERENCES

[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] C. H. Lie, C. L. Hwang, F. A. Tillman, Availability of maintained systems: a state-of-the-art survey. AIIE Trans. 9, 247-259, 1977.

[5] 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.

[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] T. Jazouli and P. Sandborn, A Design for Availability Approach for Use with PHM, Proceedings of the International Conference on Prognostics and Health Management, October 2010.

[9] 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.

[10] 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.

[11] 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.

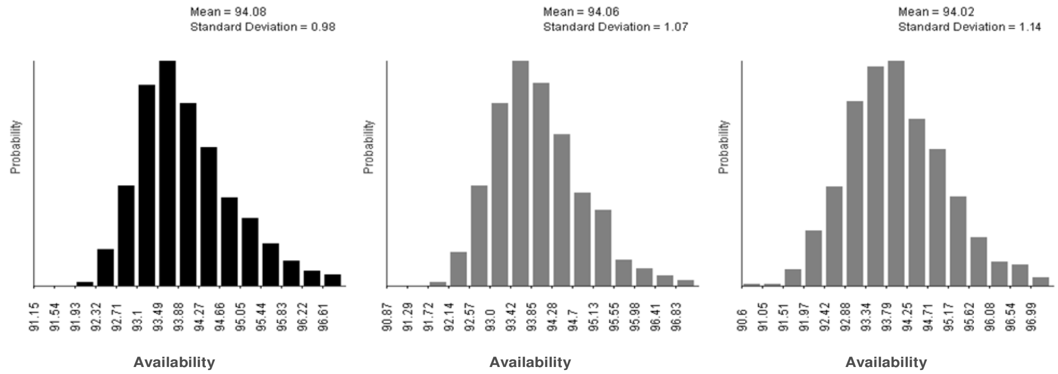
[12] Federal Aviation Administration, Investment Analysis Benefit Guidelines: Quantifying Flight Efficiency Benefits, Version 3.0, Investment Analysis and Operations Research Group, June 2001.

[13] 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.

APPENDIX A – DATA SUMMARY FOR CASE STUDIES

This Appendix represents a simplified version of the case study that appeared in [10]. This Appendix provides model inputs and assumptions that are used for the example analyses presented in Section III. Only the most relevant inputs for this specific application of the design for availability model are provided here; for a more detailed inputs data refer to [10]. The LRU used in this example is an avionics multifunction display (MFD). The implementation costs are summarized in Table A.1. The discount rate on money used was 0.07 (constant over time).

The cost per hour out of service is \$500 and \$5092 for scheduled maintenance and unscheduled maintenance (assuming during mission failures) respectively. However, it is



(a) (b) (c)

Figure B.1. (a) Required availability distribution (input to the model). (b) Availability distribution generated (output of the discrete event simulator) using the computed *TTF* (Figure B.2a) for unscheduled maintenance. (c) Availability distribution generated (output) using the computed *TTF* (Figure B.2b) for data-driven PHM.

assumed 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 A.2 [10, 11], and a 20 years support life was chosen based on [12].

Table A.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 (CREC)
Recurring Costs	Annual Infrastructure	$450 per socket (CINF)
Non-Recurring Engineering	PHM cost	$700 per LRU (CNRE)

Table A.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 [13]	6 preparation periods per day (between flights)	270 minutes between flights/day

Table A.3. 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

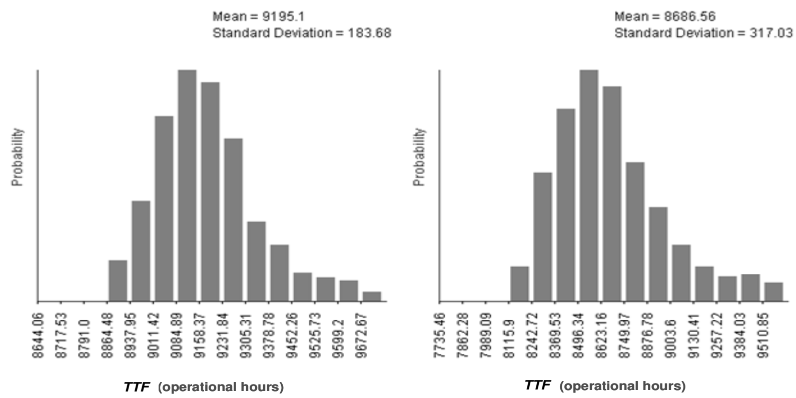
Table A.3 summarizes the spares inventory (per socket) assumptions. This includes the spares carrying costs, which are incorporated into the LRU recurring costs.

Appendix B – Model Verification

This appendix represents a qualitative verification of the design for availability methodology. The following three steps comprise the verification process:

1. Using the availability distribution requirement as an input to the design for availability model, determine the allowable minimum *TTF* distribution.
2. Use the *TTF* distribution generated in step 1 as an input to the existing PHM ROI simulation (described in Section III of this paper) to predict an availability distribution as an output.
3. Compare the availability distribution input requirement (used in step 1) to the availability distribution determined as an output (in step 2) – they should be equivalent.

The availability requirement considered in this Appendix is shown in Figure B.1a; which has been used as an input to the design for availability model. Figures B.2a and B.2b show the resulting *TTF* distributions using unscheduled maintenance and data-driven PHM respectively. In order to qualitatively verify the methodology, the *TTF* distribution (Figure B.2a) was used as an input to the PHM ROI tool discussed in Section III of this paper, while using an unscheduled maintenance approach. The PHM ROI tool used the *TTF* distribution along with the other inputs in the Appendix A and generated a resulting availability distribution (Figure B.1b). The two availability distributions (Figures B.1a and B.1b) are not expected to be absolutely identical (since this is a stochastic solution), but the means and standard deviations are very similar.



(a) (b)

Figure B.2. *TTF* distributions (model outputs) required to satisfy the availability requirement in Figure 6a for various maintenance approaches. (a) *TTF* distribution for an unscheduled maintenance approach. (b) *TTF* distribution for a data-driven approach.