Prognostic Algorithm Verification

N. Scott Clements and David S. Bodden

Lockheed Martin Aeronautics Company, Fort Worth, TX, 76131, USA
scott.clements@lmco.com
david.s.bodden@lmco.com

ABSTRACT

A rigorous methodology is presented for both specification and verification of prognostic algorithm performance. The prognostic algorithm specification statement takes the form, "The prognostic algorithm shall provide a minimum of <TTM> hours time-to-maintenance such that between <Lower>% and <Upper>% of failures of component ABC will be avoided with <Confidence>% confidence." The methodology is developed first for a single failure mode case and then extended to the multiple failure mode case. The case of non-prognosable failure modes is also considered. Finally, implications of this approach are presented, including pre-tabulation of confidence bounds, estimation of the minimum amount of data required to reach a given verification confidence, and a method for using a minimum confidence growth curve to account for initial low confidence in a prognostic algorithm.

1. INTRODUCTION

The goal of prognostics is to predict the time to failure (or similar measures, such as remaining useful life or time to maintenance) of a component or system. These predictions, when incorporated into an overall maintenance concept of operation, may provide several benefits, such as increased mission reliability and system availability, optimized spares positioning, and enhanced reliability centered maintenance (Massam & McQuillan, 2002).

At the earliest stages of design, these goals are documented as requirements. Typically, requirements statements are first developed at a higher (system) level, then flowed down to lower sub-systems and, potentially, individual components. As the requirements are developed at the lower tiers, they tend to become more specific and, thus, independently verifiable. Early work in writing requirements for prognostic algorithms relied on basic measurements such as the confidence interval at standard mean time to failure prediction (Kacprzynski et al., 2004), average bias and precision (Roemer, Dzakowic, Orsagh, Byington, & Vachtsevanos, 2005), and minimum time to prediction and minimum improvement of the service interval over legacy methods (Line and Clements, 2006).

More recently, several performance criteria for Prognostics and Health Management (PHM) have been developed, as documented in (Saxena et al., 2008), (Leao, Yoneyama, Rocha, & Fitzgibbon, 2008), and (Wheeler, Kurtoglu, & Poll, 2010). These criteria, though, are usually used as a means of measuring the performance of a prognostic algorithm, often in relation to other algorithms (say, for example, to determine the 'best-performing' algorithm out of a set). While these performance measures could, potentially, be used as the basis for a requirement (Tang, Orchard, Goebel, & Vachtsevanos, 2011), there are two issues with this approach. First, there are not currently accepted performance thresholds related to these measures (particularly for fielded systems). Second, most of these measures require knowing the true state of health of the component being analyzed, or at least the true time of failure. In many fielded systems, it is not acceptable to let a component run to failure, and not cost effective or accurate to determine the remaining life of a component removed before failure.

Often, the data that will be available for verification will be (1) how many instances of a component (across a fleet, for instance) were replaced and (2) of those, how many failed before they were replaced (as opposed to how many were replaced based on a time-to-maintenance prediction).

Due to the inherent uncertainties associated with prognostic algorithms, the remaining useful life prediction is typically given as a probability distribution around a mean predicted time to failure. Instead of trying to characterize and verify the shape of the predicted failure pdf, this paper uses a threshold requirement (i.e., "capture 95% of all failures") as a starting point for verification.

This document addresses two main issues of verification of prognostic algorithms. First, Section 2 discusses what a meaningful and verifiable prognostic requirement statement must include. Then, Section 3 provides a statistical approach to verifying such a requirement by considering the case of a single failure mode. Section 4 extends the analysis presented

Clements et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Approved for Public Distribution, PIRA# AER201305018 (LM Aero)

in Section 3 to the case of multiple failure modes, both prognosable and non-prognosable. Finally, Section 5 discusses some of the real-world implications of this verification approach.

2. Prognostic Requirement Statements

A primary issue with verifying prognostics algorithms is formulating a proper prognostic requirement statement. "Proper" in this sense means that the requirement is verifiable, which implies that the data needed to verify it can realistically be acquired. Unfortunately, the data most likely to be available is rather limited. In the normal course of developing a component, the design, analysis, and perhaps "lab data" will be useful for determining such things as average failure distributions, failure modes and probabilities, and potential precursor signals of failures. While helpful in designing the prognostic algorithm itself, this type of data will usually not be useful in verifying the algorithm. In practice, the data available for verification will be maintenance data, such as how long a component has been in service, how many times a component has been replaced, how many times it has failed, etc.

Consider the example of a flight control surface actuator such as an electro-mechanical actuator (EMA). Suppose a prognostic technique has been developed for this EMA that gives a time-to-maintenance (TTM) indication based on measured performance. Further, assume that maintenance is planned based on the prognostic indication. That is, the part is replaced when indicated, even if it has not failed. In an ideal world, once removed, the part would be analyzed to determine how much useful life remained in the component. Although the component could conceivably be placed in a test bench and operated until failure, doing so would not be economically feasible (not to mention issues such as recreating realistic flight conditions and load profiles). Instead, the part will most likely be repaired, recertified and placed back in the supply chain (or discarded). However, as mentioned before, there is some data available for verification: the number of times that EMA has been removed and whether or not it had failed in place before being removed (i.e., when the prognostic algorithm fails to give a maintenance time before failure). Note that these counts can be aggregated across all aircraft in a squadron (for example) to provide a statistically significant sample.

In addition to having reasonable access to the requisite data, the prognostic requirement statement must be written in such a way that it has an interpretation that is not ambiguous. To demonstrate some of the ambiguities that can arise with interpretation, an initial attempt at a prognostic requirement will be given and then refined as needed.

1st attempt: *The prognostic algorithm shall provide a time-to-maintenance such that at least 95% of failures of component XYZ will be avoided.*

As will be shown, there are several problems with this statement. First, there is no minimum bound on the time to failure of the prediction. Simply declaring "Component XYZ will fail in five minutes" (or some other arbitrarily short time) technically satisfies the requirement, but is practically useless. This minimum time to failure declaration requirement often stems from an analysis of the minimum useful notification, based on factors such as the lead time to procure a replacement component and how often the prognostic algorithm will be run. So, a second attempt is made:

2nd attempt: *The prognostic algorithm shall provide a minimum of 20 hours time-to-maintenance such that at least 95% of failures of component XYZ will be avoided.*

This attempt at a requirement statement at first glance may appear adequate (and, indeed, it is close), but as will be shown more clearly in the next section, there are still two problems with it – the confidence in the prediction and protection against 'overly conservative' predictions. In the next section, the verification approach will be presented as well as further refinements on the requirement statement to address these issues.

3. Verification Approach

It is important to understand that this verification technique is not trying to determine how well the prognostic algorithm is determining the actual remaining useful life distribution of a component. In fact, the prognostic algorithm does not even need to explicitly calculate the remaining life distribution. Rather, this approach to verification is based off the "avoid 95% of failures" portion of the requirement statement. Specifically, this approach evaluates whether the time-to-maintenance value (which the prognostic algorithm does provide) is adequately avoiding the specified percentage of failures.

The basic idea behind this approach is the expectation that the prognostic algorithm is, in fact, expected to "miss" a small percentage of failures. In the example requirement statements given in Section 2, an algorithm that satisfies the requirement will avoid at least 95% of failures; conversely, it will miss at most 5% of failures. So out of 100 replacements, the component can be expected to fail *about* 5 times. If the maintenance records indicate that there were actually 25 failures, the algorithm is probably not meeting the requirement. The rest of this section attempts to apply statistical theory to this approach to better quantify the confidence that the algorithm is meeting the stated requirement.

3.1. Assumptions

The following assumptions are made throughout the rest of this section.

1. The prognostic algorithm being verified provides a minimum time-to-maintenance that satisfies the minimum time constraint of the requirement statement.
2. Maintenance actions are planned based on the time-to-maintenance measures (that is, the part is replaced when indicated, even if it has not failed).
3. The number of times the component has been replaced (both due to failure or prognostic indication) is available.
4. The number of times the component failed before being replaced is available.
5. If a component is removed either due to a failure or based on a prognostic indication of imminent failure, then it will be replaced or serviced to a 'like new' condition before re-entering the supply chain.

There are several measures that may be output from a prognostic algorithm, including a best estimate of the remaining useful life, the shape of the remaining life distribution, and a best estimate of the time-to-maintenance for a given failure avoidance percentage. The first assumption is simply that the TTM value is made available (other measures may or may not be output as well).

The other three assumptions concern maintenance operations. The second assumption is that the time-to-maintenance measure is actually used. As will be discussed in Section 5.3, this assumption can be relaxed a little to provide for a "confidence building" period during which the prognostic algorithm is verified without the risk of excessive failures or unnecessarily maintenance actions. Finally, the last two assumptions provide the data needed for the verification calculations.

3.2. Single Failure Mode Construction

To start the derivation, assume that the prognostic algorithm is perfectly accurate. That is, the algorithm provides a consistent time-to-maintenance measure, which divides the failure rate pdf (whatever its shape) as shown in Figure 1. In this figure, t_0 is the time at which the prediction is made, t_M is the maintenance time, and f is the percentage of failures that would be avoided by performing maintenance at the indicated time.

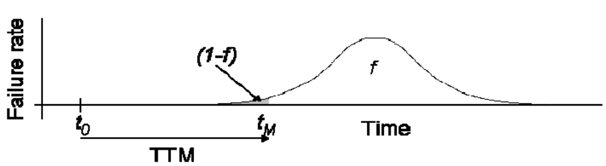

Figure 1. Prognostic Algorithm Time-To-Maintenance Prediction

The following two definitions are now made. Let n be the number of components replaced (failed and not failed) and x be the number of those replaced components that failed before being replaced. With these definitions, the probability of missing exactly x component failures is given by the standard binomial distribution:

$$P(\text{missing}\, x\, \text{failures}) = \binom{n}{x}(1-f)^x f^{n-x}, \tag{1}$$

where $\binom{n}{x}$ is the binomial coefficient:

$$\binom{n}{x} = \frac{n!}{(n-x)!x!}. \tag{2}$$

This expression characterizes the distribution of x given values for n and f. Figure 2 shows the probability mass function (pmf) of this binomial distribution for $n = 50$ and $f = 0.8$. As expected, the highest probability of failure occurs at $x = n \times (1-f) = 50 \times (1-0.8) = 10$. Note that this is a discrete distribution – it is only defined on integer values of x. However, a different distribution can be calculated for every possible value of f between zero and one. For example, Figure 3 shows the binomial distributions for eleven different values of f. The distribution corresponding to Figure 2 ($f = 0.8$) is highlighted in blue. Further, since f can take on any value in the range from zero to one, Figure 3 can be filled in, yielding Figure 4.

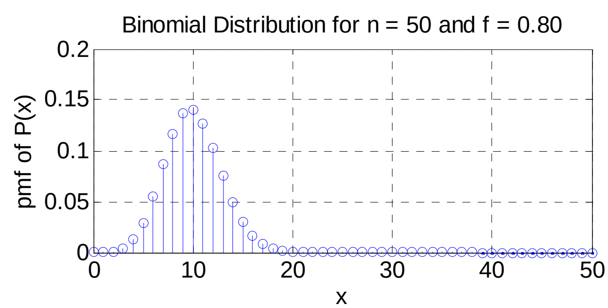

Figure 2. Binomial Distribution for $n = 50$ and $f = 0.80$

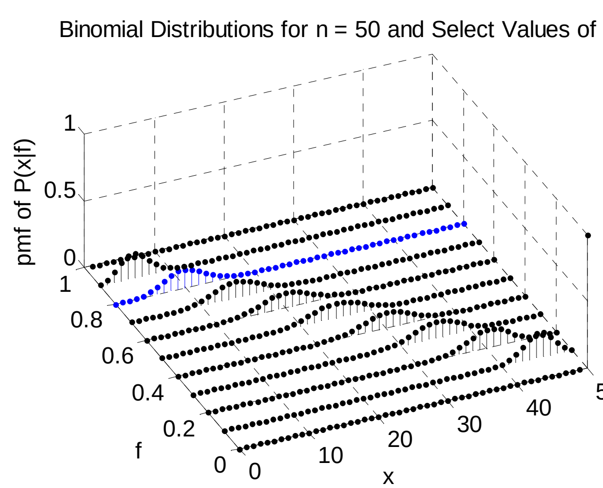

Figure 3. Binomial Distribution for $n = 50$ and Various Values of f

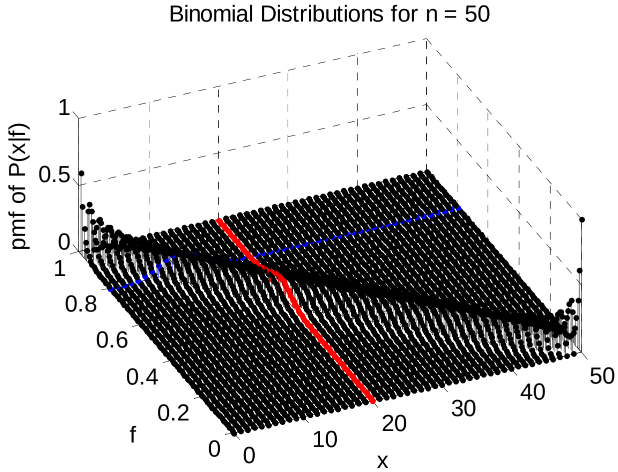

Figure 4. Binomial Distribution for n = 50 and All Values of f

Up to now, the values for f and n have been considered known. Now consider a slightly different problem: given values for n and x, what is the best guess for the value of f? For example, suppose that n = 50 and x = 20, as highlighted by the red line in Figure 4. This red line is the marginalization of the joint distribution of f. Note that this line is continuous in f and only defined over the range $0 \le f \le 1$. Before using this marginalization to develop a confidence measure, it must first be normalized to form a proper pdf. This results in a pdf for the marginalization of the joint distribution of f given by

$$(n+1)\binom{n}{x}(1-f)^x f^{n-x}. \tag{3}$$

As an example, the graph of the pdf of f for n = 50 and x = 20 is given in Figure 5. This corresponds to the red line shown in Figure 4.

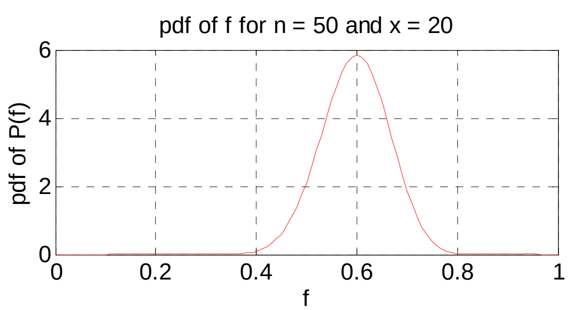

Figure 5. pdf of f for n = 50 and x = 20

Given the pdf for the distribution of f for a given set of values n and x, the confidence (i.e., probability) that the actual value of f is a given value (or, more accurately, that the actual value is within a range of values) can be found by calculating the area under the distribution for that range. Written as a formula, the confidence that the actual value of f is between two values a and b (where $a \le f \le b$) is given as follows:

$$\int_a^b (n+1)\binom{n}{x}(1-f)^x f^{n-x} df. \tag{4}$$

For example, suppose that the flight control surface EMA had been replaced 50 times, and 20 of those times were due to a failure of the EMA, the confidence that the prognostic algorithm was avoiding at least 95% of the failures would be equal to the area under the curve in Figure 5 from 0.95 to 1.00 (or 95% to 100%). Obviously, for this curve the confidence would be very close to zero, which intuitively makes sense given the number of failures that were incurred.

However, suppose there had only been one failure out of 50 replacements. The distribution for f in that scenario is shown in Figure 6. In this case the confidence that the failure avoidance is at least 95% is much higher (73.56%):

$$\int_{0.95}^{1.00} (50+1)\binom{50}{1}(1-f)^1 f^{50-1} df = 0.7356. \tag{5}$$

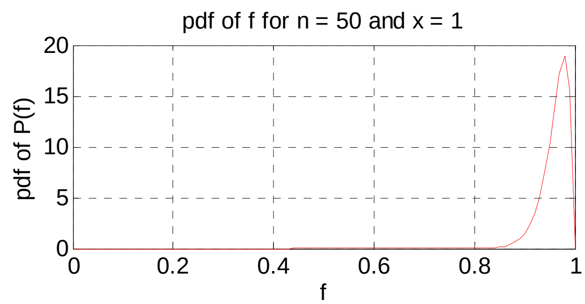

Figure 6. pdf of f for n = 50 and x = 1

3.3. Prognostic Requirement Statement Refinements

In Section 2, the requirement statement was left with two remaining issues. Now that a method of determining the confidence has been shown, the prognostic requirement can be further refined.

3rd attempt: *The prognostic algorithm shall provide a minimum of 20 hours time-to-maintenance such that at least 95% of failures of component XYZ will be avoided with 90% confidence.*

With the addition of the confidence measure, the requirement statement can be verified in a statistically meaningful manner. However, there is no protection against 'overly conservative' predictions. For example, a prognostic algorithm that claims 20 hours time-to-maintenance every time it is run (even if there were actually hundreds or more hours of remaining useful life), would meet the requirement, as it would definitely catch 95% of all failures of that component. Realistically (for components that are not so critical that they should never be allowed to fail in place), there is an expectation that an accurately tuned prognostic algorithm would 'miss' some (small) percentage of failures. Thus, instead of saying "… at least 95% of failures …"

(which implies between 95% and 100%), the upper bound is slightly reduced.

4th attempt: *The prognostic algorithm shall provide a minimum of 20 hours time-to-maintenance such that between 95% and 99% of failures of component XYZ will be avoided with 90% confidence.*

Figure 7 shows a grid plot of the confidence calculation for various values of x and n. Grid coordinates (i.e., combinations of x and n) that meet or exceed 90% confidence (per the example requirement statement above) are colored green. Similarly (for illustrative purposes), yellow indicates a confidence between 70% and 90%, and blue indicates less than 70% confidence. As can be seen, for some values of x, there is no value of n that will satisfy the requirement. Moreover, consider the case where $x = 4$. The requirement is satisfied only if n is between 168 and 237. Thus, if the prognostic algorithm is too conservative (and there have been more than 237 replacements for 4 failures), the confidence will drop below the threshold. Thus, a prognostic requirement written as shown above is not only verifiable from maintenance record data, but it also provides a means of identifying algorithms that are potentially too conservative in their time-to-maintenance predictions.

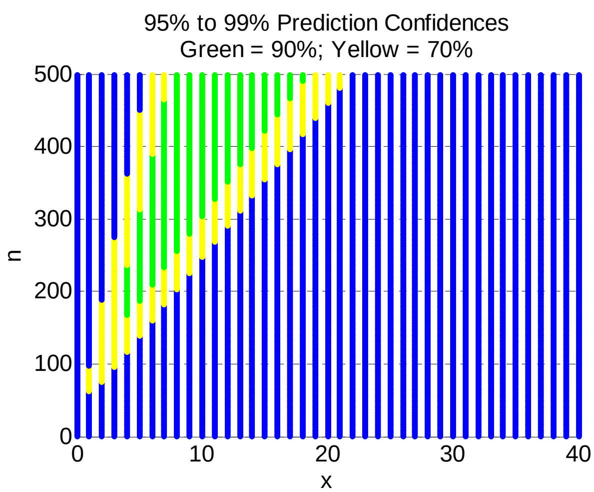

Figure 7. Prediction Confidence Regions

4. Extension to Multiple Failure Modes

The verification approach developed in Section 3 assumed that the component under analysis had a single failure mode. In reality, that is rarely the case, as components often have many failure modes and prognostic algorithms to cover only a few of them (typically, the most severe or frequent one(s)). This section extends the previous approach to verification of multiple failure modes.

As a start, it should be pointed out that the previous (single failure mode) approach still has applicability when considering the *total* prognostic requirement for a component. That is, if the goal is to determine how well a given component is meeting a goal of, say, between 90% and 99% coverage of all failure modes given whatever prognostic algorithm(s) are implemented, the single failure mode verification approach can be used. If, however, the desire is to see how well the individual algorithms (for a single component) are meeting individual goals, a new approach is warranted.

This extension to multiple failure modes is presented in three steps. First, Section 4.1 extends the previous approach to the case of two failure modes for a single component, each of which has its own prognostic algorithm. An overall confidence algorithm is then constructed from probabilistic principles for this two failure mode case. Section 4.2 then further extends this construction to the case of an arbitrary number of failure modes (again, where each has its own prognostic algorithm). Finally, Section 4.3 addresses the situation where one or more failure modes do not have associated prognostic algorithms.

4.1. Two Failure Mode Construction

To describe the approach taken in this extension, consider the case of a component with two failure modes. Further, assume that the component has been replaced four times (n = 4) of which two were due to component failure (x = 2). There are several scenarios that could lead to this result, as shown in Table 1. The first column of the table (n_1) is the number of times the component was replaced due to failure mode m_1. This includes both preemptive replacements based on prognostic indications and replacements required due to a failure of the component due to failure mode m_1. The second column (x_1) is the number of times the component failed due to failure mode m_1 before being replaced (that is, it was not replaced preemptively based on a prognostic indication). The next two columns (n_2 and x_2) represent the corresponding values for failure mode m_2.

Table 1. Possible Scenarios for Two Failure Mode Example

n_1	x_1	n_2	x_2	$n = n_1 + n_2$	$x = x_1 + x_2$
0	0	4	2	4	2
1	0	3	2	4	2
2	0	2	2	4	2
1	1	3	1	4	2
2	1	2	1	4	2
3	1	1	1	4	2
2	2	2	0	4	2
3	2	1	0	4	2
4	2	0	0	4	2

There are several things to note in this table. First, the total number of replacements (n) for each scenario must equal the (known) total number of replacements for the component ($n_1 + n_2$). Similarly, the total number of failures (x) for each scenario must equal the (known) total number of failures for the component ($x_1 + x_2$). Also, for each failure mode, the number of replacements due to that failure mode cannot be less than the number of actual failures due to that failure mode (or, algebraically, $n_i \geq x_i$). Finally, the individual values of n_1, x_1, n_2, and x_2 may not be known.

As shown in the derivation for a single failure mode, the probability of missing x_i out of n_i failures given that the failure mode is m_i (with corresponding value f_i) is given by:

$$P(x_i \mid n_i, m_i) = \binom{n_i}{x_i}(1-f_i)^{x_i} f_i^{n_i - x_i} . \quad (6)$$

The total probability, $P(x,n)$, can then be found as

$$P(x,n) = \sum P(x_1 \mid n_1, m_1) P(x_2 \mid n_2, m_2) P((m_1,n_1),(m_2,n_2)) \quad (7)$$

where the summation is taken over all possible scenarios (as given in Table 1). The last term in the above summation, $P((m_1,n_1),(m_2,n_2))$, is the probability that m_1 occurred n_1 times and m_2 occurred n_2 times and is given by

$$P((m_1,n_1),(m_2,n_2)) = \binom{n}{n_1,n_2} P(m_1)^{n_1} P(m_2)^{n_2} . \quad (8)$$

The term $\binom{n}{n_1,n_2}$ is the multinomial coefficient (a generalization of the binomial coefficient). In general, the multinomial coefficient is given by

$$\binom{n}{n_1,n_2,\cdots,n_k} = \frac{n!}{n_1! n_2! \cdots n_k!} . \quad (9)$$

This coefficient can be thought of as the number of ways that n objects can be placed in k bins with n_1 objects in the first bin, n_2 objects in the second bin, etc.

The term $P(m_i)$ is the *a priori* known relative probability that a failure is due to failure mode m_i. This term can be determined from standard reliability data, such as Mean Time Between Failure (MTBF). If the MTBF for failure mode m_i is given by $MTBF_i$, the relative probability for failure mode m_i is given by

$$P(m_i) = \frac{MTBF_i^{-1}}{\sum_{i=1}^{k} MTBF_i^{-1}} . \quad (10)$$

Returning to the example of two failure modes (Table 1) with $n = 4$, $x = 2$, $f_1 = 0.8$, $f_2 = 0.9$, $MTBF_1 = 5000$ hours, and $MTBF_2 = 2000$ hours, evaluating Equation 7 yields $P(2,4) = 0.07532$. This value is interpreted to mean that there is a 7.53% probability that there will be 2 missed failures out of 4 total replacements, given two failure modes with the given MTBF values and prognostic algorithms with the given f_i values. Figure 8 is a 3-D stem plot of this total probability calculated for values of x between 0 and 10 and values of n between 0 and 50. The example calculated previously is highlighted on this plot.

In Figure 8 the values of f_1 and f_2 are held constant while x and n are varied. In application, the values of x and n will be known and the issue will be to determine the most probable ranges of f_1 and f_2. To address this question, we can plot the probability $P(x,n)$ for various values of f_1 and f_2 for given values of x and n. Figure 9 shows such a plot for the previous example ($x = 2$ and $n = 4$). The point highlighted on the plot (with $f_1 = 0.9$ and $f_2 = 0.8$) corresponds to the same point highlighted in Figure 8.

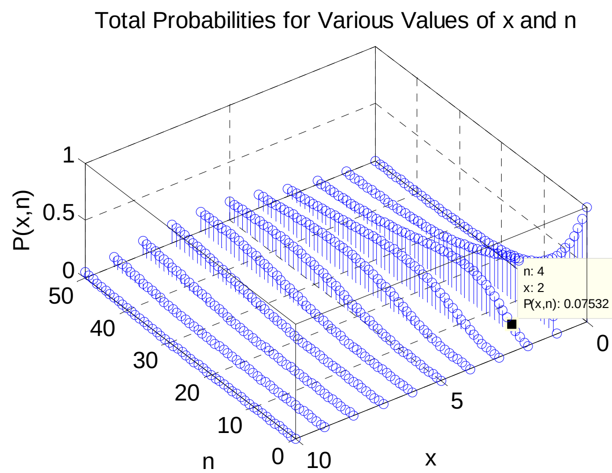

Figure 8. Plot of Total Probabilities for Various Values of x and n with Fixed f_1 and f_2

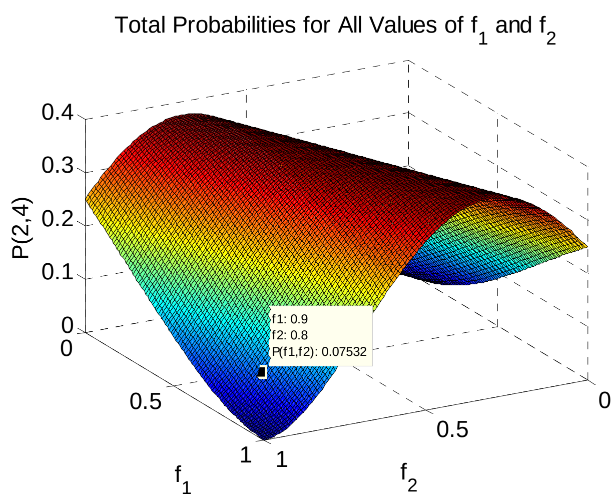

Figure 9. Plot of Total Probabilities for All Values of f_1 and f_2 with $x = 2$ and $n = 4$

Several observations can be made of Figure 9. First, the values f_1 and f_2 are continuous and bounded ($0 \leq f_i \leq 1$). Thus, the plot is a true surface and not discrete values. Second, the values of f_1 and f_2 that have the highest total probability (depicted as red in Figure 9) form a skewed line. This trend is more easily seen when the total probability plot is viewed "straight down", as shown in Figure 10. Such a 2-dimensional plot is called a "heat map", as the 3rd dimension is depicted purely as a gradient color (typically from blue to red). Recall that this example assumes we have "missed" 2 out of 4 failure events ($x = 2$ and $n = 4$). Thus the probability that both prognostic algorithms have f_i values of 50% ($f_1 = 0.5$ and $f_2 = 0.5$) should be high. As shown most readily in the heat map of Figure 10, this is indeed the case. However, since it is not presumed known how the values of x and n break down in relation to each failure mode, there are other scenarios that are just as probable. For example, the algorithm associated with failure mode m_1 may have a better (higher) value of f_1 that is compensated by a worse (lower) value of f_2. This tradeoff is evidenced by the straight banding of colors shown in Figure 10. If the failure rates (or MTBFs) of the two failure modes were equal, this line would be at a 45° angle to the f-axes. In this example, however, failure mode m_2 has a higher failure rate (or, equivalently, a lower MTBF) than failure mode m_1. Thus, a change in the value of f_2 will have a more pronounced effect on the total probability than a change in the value of f_1. This is illustrated by the skewing of the bands of the heat map to the $f_2 = 0.5$ line (or "to the vertical").

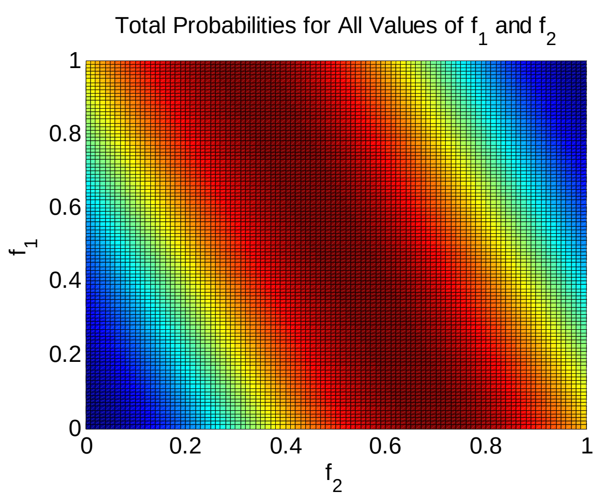

Figure 10. Heat Map of Total Probabilities for All Values of f_1 and f_2 with $x = 2$ and $n = 4$

Continuing with the generalization of the approach outlined for the single failure mode case, the total probability surface shown in Figure 9 is the marginalization of the joint distribution of the values of f_1 and f_2. Thus, if the surface is normalized such that the total volume under the surface is 1.0, the resulting surface will be the joint probability distribution function (pdf) of the values f_1 and f_2. Note that this joint pdf surface is the same shape as that of Figure 9 with the only difference being the scaling of the z-axis.

Having calculated the joint pdf, determining the confidence that the f_i values of the two prognostic algorithms are in given ranges is simply a matter of integrating the joint pdf over the ranges of interest. For example, to determine the probability that $0.45 \le f_1 \le 0.55$ and $0.4 \le f_2 \le 0.6$, the following double integral would be evaluated:

$$\int_{0.4}^{0.6}\int_{0.45}^{0.55} p(f_1, f_2)\,df_1 df_2 \,, \tag{11}$$

where $p(f_1, f_2)$ is the joint pdf. For the case of $x = 2$ and $n = 4$, the integration will yield the following:

$$\int_{0.4}^{0.6}\int_{0.45}^{0.55} p(f_1, f_2)\,df_1 df_2 = 2.87\% \,. \tag{12}$$

This confidence value is rather small, but recall that the total number of replacements (n) in this example is just four. If the number of replacements is increased to $n = 100$ and the number of missed failures kept at 50% ($x = 50$), the resulting pdf is shown in Figure 11. Compared to Figure 9, this pdf has much sharper roll-offs on either side of the "high-probability" line. The resulting confidence is also higher:

$$\int_{0.4}^{0.6}\int_{0.45}^{0.55} p(f_1, f_2)\,df_1 df_2 = 8.46\% \,. \tag{13}$$

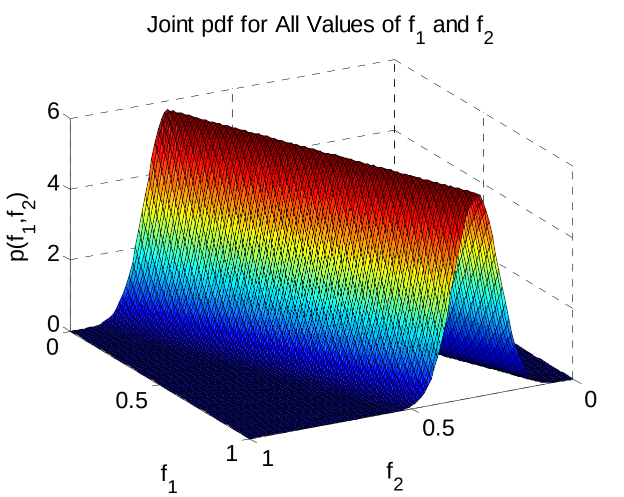

Figure 11. Joint pdf of f_1 and f_2 with Fixed $x = 50$ and $n = 100$

Even with the sharper roll-off from a higher number of replacements (n), the confidence for this example is still only 8.46%. This is primarily due to the large number of combinations of failure modes that will yield around 50% prediction (i.e., $f_1 \approx 0.5$ and $f_2 \approx 0.5$). A perhaps more realistic example, shown in Figure 12, represents five missed failures ($x = 5$) out of 100 replacements ($n = 100$). Calculating the confidence that $0.8 \le f_1 \le 0.99$ and $0.9 \le f_2 \le 0.99$ yields a value of 64.31%:

$$\int_{0.9}^{0.99}\int_{0.8}^{0.99} p(f_1, f_2)\,df_1 df_2 = 64.31\% \,. \tag{14}$$

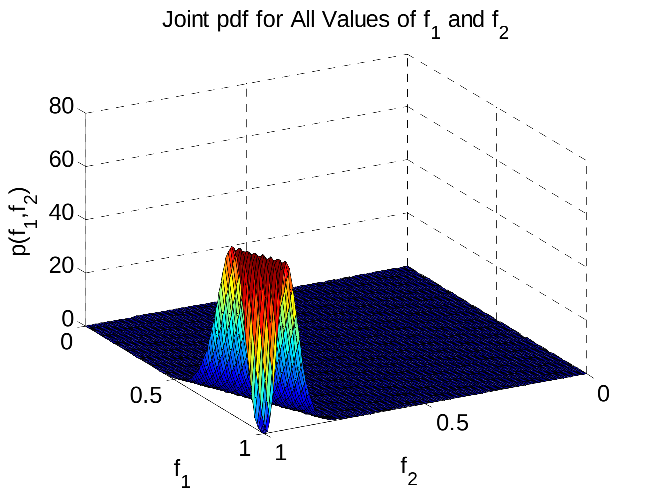

Figure 12. Joint pdf of f_1 and f_2 with Fixed $x = 5$ and $n = 100$

4.2. Generalization to *k* Failure Modes

The generalization from two failure modes to an arbitrary number of failure modes is straightforward. However, the process quickly turns into an exercise in proper indexing. The two primary points to keep in mind are the following. First, all combinations of $P(x_i, n_i)$ must be accounted for and weighted based on their frequency and relative failure rate. Second, for each scenario (i.e., set of values $\{x_i\}$ and $\{n_i\}$), the following must be true:

- $\Sigma_{i=1}^{k} x_i = x$, (15)
- $\Sigma_{i=1}^{k} n_i = n$, and (16)
- $n_i \geq x_i$. (17)

Equation 15 through 17 are simple generalizations of the two-failure mode construction (Section 4.1) to k failure modes. Although the derivation is quite involved (and omitted from this paper for space), one solution is presented below.

$$P(x,n) = \sum_{x_1=0}^{x} \sum_{x_2=0}^{x-x_1} \cdots \sum_{x_{k-1}=0}^{x-\Sigma_{i=1}^{k-2} x_i} \sum_{n_1=x_1}^{x_1+s} \sum_{n_2=x_2}^{x_2+s-s_1} \cdots \sum_{n_{k-1}=x_{k-1}}^{x_{k-1}+s-\Sigma_{i=1}^{k-2} s_i} \left\{ \binom{n}{n_1, n_2, \cdots, n_k} \prod_{i=1}^{k} \left[P(x_i \mid n_i, m_i) P(m_i)^{n_i} \right] \right\} \quad (18)$$

with:

$$\begin{aligned} s &= n - x \\ s_i &= n_i - x_i \\ x_k &= x - \Sigma_{i=1}^{k-1} x_i \\ n_k &= x - \Sigma_{i=1}^{k-1} n_i \\ P(x_i \mid n_i, m_i) &= \binom{n_i}{x_i} (1-f_i)^{x_i} f_i^{n_i - x_i} \end{aligned} \quad (19)$$

The k-dimensional joint pdf is then found by normalizing the k-dimensional integral of $P(x,n)$ to one:

$$p(f_1, f_2, \cdots, f_k) = \frac{P(x,n)}{\int_0^1 \cdots \int_0^1 \int_0^1 P(x,n) df_1 df_2 \cdots df_k} . \quad (20)$$

Finally, the confidence that $a_i \leq f_i \leq b_i$ for $i = 1,\ldots,k$ is given by:

$$\int_{a_k}^{b_k} \cdots \int_{a_2}^{b_2} \int_{a_1}^{b_1} p(f_1, f_2, \cdots, f_k) df_1 df_2 \cdots df_k . \quad (21)$$

4.3. Non-Prognosable Failure Modes

Finally, the case of non-prognosable failure modes is considered. A non-prognosable failure mode is simply a failure mode for which there is no prognostic algorithm in place to predict remaining useful life. Note that the lack of a prognostic algorithm need not imply that such an algorithm could not be developed, only that it isn't in place for the component being analyzed. Further, it is reasonable that many failure modes of a component will not have prognostic algorithms (due to the relative infrequency of occurrence of the failure modes or a lack of technical understanding to develop such algorithms). All of these non-prognosable failure modes can, for the purposes of this analysis, be combined into a single non-prognosable failure mode with a composite MTBF given by:

$$MTBF_{non-prognosable} = \left(\sum MTBF^{-1} \right)^{-1} , \quad (22)$$

where the summation is taken over all non-prognosable failure modes.

A non-prognosable failure mode can then be characterized as a failure mode where $f = 0$. That is, there is zero probability that the failure mode will be predicted before that failure mode occurs. Further, this value of f is not probabilistic (it is deterministic with value zero), so it should not be included as a variable in the joint pdf. The effect of this characterization is to alter the calculation of the total probability function, $P(x,n)$. Without loss of generality, let the non-prognosable failure mode be listed as the last (or k^{th}) failure mode. Thus, $f_k = 0$. Now consider the term $P(x_k|n_k,m_k)$ that occurs in the calculation of $P(x,n)$. This term is given by:

$$\begin{aligned} P(x_k \mid n_k, m_k) &= \binom{n_k}{x_k} (1-f_k)^{x_k} f_k^{n_k - x_k} \\ &= \frac{n_k!}{(n_k - x_k)! x_k!} (1-0)^{x_k} 0^{n_k - x_k} \\ &= \frac{n_k!}{(n_k - x_k)! x_k!} 0^{n_k - x_k} . \end{aligned} \quad (23)$$

From the constraints mentioned in Section 4.2, n_k must be greater than or equal to x_k, so consider the two cases $n_k = x_k$ and $n_k > x_k$ (recalling that $0! = 0^0 = 1$):

$$P(x_k \mid n_k, m_k) = \begin{cases} 1 & , n_k = x_k \\ 0 & , n_k > x_k \end{cases}. \tag{24}$$

This is consistent with the earlier characterization that a non-prognosable failure mode misses all occurrences of that failure mode (that is, when $n_k = x_k$ the probability is one). And for any case where fewer than all of the non-prognosable failure modes are missed ($n_k > x_k$), the probability is zero.

5. IMPLICATIONS

5.1. Tabulation

As discussed in Section 4, the overall confidence equation (Eq. 4) can be used when considering the verification of an overall requirement for a given component (as opposed to the multiple failure mode confidence equation given as Eq. 21). The evaluation of the integral in the overall confidence equation does have a closed-form solution. This closed form of the solution, though unwieldy to write down, is generally quicker (and more accurate) to calculate than to evaluate the integral using numerical techniques. Tabulations can be pre-calculated and stored instead of performing the complex calculation every time a value is needed.

5.2. Minimum Amount of Data Required for Verification

It is often desirable to know how much data will be required to verify a requirement. Such knowledge can be useful when scheduling and allocating resources to the verification task. To show how this information can be derived from this verification technique, consider an electro-mechanical actuator with the following prognostic requirement.

The prognostic algorithm shall provide a minimum of 20 hours time-to-maintenance such that between 95% and 99% of failures of the EMA will be avoided with 90% confidence.

The more failures that occur, the more replacements must have been performed to meet the requirement. Also, as mentioned in Section 3.3, for some numbers of failures, there is no number of replacements that will satisfy the requirement. For example, the least number of replacements that can conceivably be used to verify the requirement is 168, but only if there have been four failures in those 168 replacements. Table 2 shows the minimum number of replacements required for verification for a given number of failure occurrences.

These numbers, combined with the predicted reliability failure rate, can give a minimum value for the amount of data required and the time required to verify a prognostic algorithm. Unfortunately, these values are only minimum values. A more practical approach is given next.

Table 2. Minimum Number of Replacements Required For Verification

No. of Failures	Minimum No. of Replacements Required
0 - 3	N/A
4	168
5	187
6	210
7	234
8	257
9	281
10	> 300

5.3. Confidence Growth Curves

The blue line in Figure 13 shows a typical confidence growth curve for a prognostic algorithm. The black asterisks indicate ‘missed’ failures (i.e., failures that occurred before the indicated TTM). All other replacements were scheduled in accordance with a prognostic algorithm time-to-maintenance prediction.

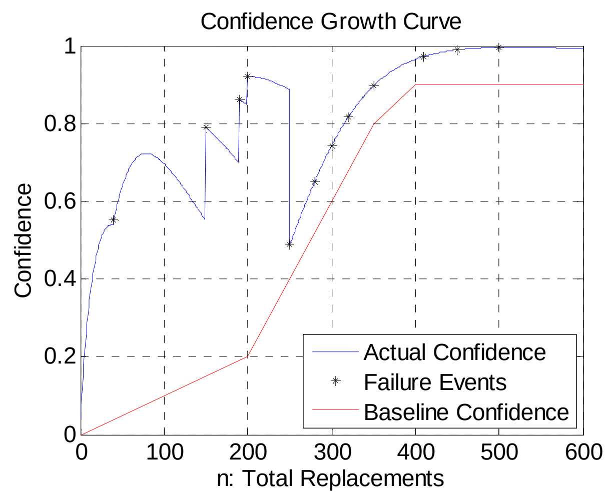

Figure 13. Confidence Growth Curve

As can be seen, the confidence starts out low and tends to increase as more data points are acquired. When a failure event occurs (and the associated un-predicted maintenance replacement), the confidence drops, particularly in the beginning when there are few data points. However, as the number of replacements increases, the effect of a failure event on the confidence curve is dampened.

The confidence growth curve, along with a baseline confidence curve, can be used to bound the time and data required for verification, as well as to provide a means of declaring a verification as failed. The baseline confidence curve is a minimum confidence threshold for the actual confidence curve. The shape of the baseline curve would be specified based on specific knowledge of the algorithm being verified. In general, though, it would tend to be pessimistic initially to allow for large swings in the confidence. An example of a baseline confidence curve is shown in Figure 13 in red.

In order for a prognostic algorithm to be verified, it would not only have to reach the desired confidence, but also do so without going below the baseline curve. If the actual confidence does dip below the baseline curve, the verification could be considered failed. An added benefit of the approach is that the baseline curve can constrain how much time is available for an algorithm to reach verification. For example, in Figure 13, the baseline confidence requires that the algorithm reaches verification (90% confidence) no later than by the 400th replacement.

The confidence growth curve can also be used to determine when to start relying on a prognostic algorithm. Often, particularly for a new prognostic technique, there can be reluctance to schedule maintenance on a part based on the prognostic prediction. In these cases, traditional maintenance concepts can initially be employed while a hypothetical confidence curve is tracked on the side. The hypothetical curve would assume that the prognostic prediction was acted upon. Similarly, if a failure occurred that the algorithm did not predict, the hypothetical confidence curve would be penalized accordingly. When and if the hypothetical confidence reaches a pre-determined threshold of acceptance, maintenance can start being scheduled based on the prognostic prediction instead of the traditional means.

6. CONCLUSION

This paper has addressed two of the central issues concerning verification of prognostic algorithms. First, the question of how to write a meaningful and verifiable prognostic algorithm requirement statement was considered. Through the course of the paper, it was shown that the following requirement statement template is both statistically meaningful and verifiable using available field data.

The prognostic algorithm shall provide a minimum of <TTM> hours time-to-maintenance such that between <LOWER>% and <UPPER>% of failures of component <COMPONENT> will be avoided with <CONFIDENCE>% confidence.

Second, a statistical approach to verifying such a statement was presented. The approach requires very few assumptions and can be easily pre-tabulated for a given requirement's failure threshold. Furthermore, implications of the approach can be used to bound the time and data necessary for verification as well as provide a means of building confidence in an un-tested algorithm.

NOMENCLATURE

k	Number of failure modes
n	Total number of components replaced (failed and not failed)
n_i	Total number of components replaced (failed and not failed) due to failure mode m_i
$P(m_i)$	Relative probability of failure mode m_i
$P(x_i,n_i\|m_i)$	Probability of missing x_i out of n_i failures given the failure mode is m_i
$P(x,n)$	Probability of missing x out of n failures of any combination of failure modes,
x	Total number of components that failed before being replaced
x_i	Total number of components that failed due to failure mode m_i before being replaced

REFERENCES

Azzam, H., Cook, J., & Driver, S. (2004), FUMS™ technologies for verifiable affordable prognostics health management (PHM), *IEEE Aerospace Conference Proceedings* (vol. 6, pp. 3764-3781), March 6-13.

Feldman, A., Kurtoglu, T., Narasimhan, S., Poll, S., Garcia, D., de Kleer, J., Kuhn, L. & van Gemund, A. (2010), Empirical Evaluation of Diagnostic Algorithm Performance Using a Generic Framework, *International Journal of the PHM Society* (vol. 1).

Halligan, G. R., Thumati, B. T., & S. Jagannathan (2011), Artificial immune system-based diagnostics and prognostics scheme and its experimental verification, *IEEE International Conference on Control Applications (CCA)* (pp. 958-963), 28-30 Sept. 2011.

Hao, L., Jinsong, Y., Ping, Z., & Xingshan, L. (2011), Review on Verification and Validation technology in integrated health management system, *10th International Conference on Electronic Measurement & Instruments (ICEMI)* (vol. 2, pp. 125-129), August 16-19.

Kacprzynski, G. J., Liberson, A., Palladino, A., Roemer, M. J., Hess, A. J., & Begin, M. (2004), Metrics and development tools for prognostic algorithms, *IEEE Aerospace Conference Proceedings* (vol. 6, pp. 3809-3815), March 6-13.

Leao B. P., Yoneyama, T., Rocha, G. C., & Fitzgibbon, K. T. (2008), Prognostics Performance Metrics and their Relation to Requirements, Design, Verification and Cost-Benefit, *IEEE International Conference on Prognostics and Health Management*.

Line, J. K., & Clements, N. S. (2006), Prognostics usefulness criteria, *IEEE Aerospace Conference Proceedings*, March.

Luna, J. J. (2010), Towards Defining and Allocating PHM Requirements for Military Systems, *Annual Conference of the Prognostics and Health Management Society*.

Lybeck, N., Marble, S., & Morton, B. (2007), Validating Prognostic Algorithms: A Case Study Using

Comprehensive Bearing Fault Data, *IEEE Aerospace Conference* (pp. 1-9), March 3-10.
Massam, S., & McQuillan, S. (2002), Verification of PHM capabilities: A joint customer/industrial perspective, *IEEE Aerospace Conference Proceedings* (vol. 6, pp. 2799-2813).
Mikat, H., Siddiolo, A. M., & Buderath, M. (2012), Virtual Framework for Validation and Verification of System Design Requirements to enable Condition Based Maintenance, *First European Conference of the Prognostics and Health Management Society*.
Ping, X., Wang, Z., & Vue, L. (2010), Prognostics and Health Management (PHM) System requirements and validation, *Prognostics and Health Management Conference* (pp. 1-4), January 12-14.
Roemer, M. J., Dzakowic, J., Orsagh, R. F., Byington, C. S., & Vachtsevanos, G. (2005), Validation and verification of prognostic and health management technologies, *IEEE Aerospace Conference* (pp. 3941-3947), March 5-12.
Saxena A., Celaya, J., Balaban, E., Goebel, K., Saha, B., Saha, S., & Schwabacher, M. (2008), Metrics for Evaluating Performance of Prognostic Techniques, *IEEE International Conference on Prognostics and Health Management*, October 6-9.
Saxena, A., Celaya, J., Saha, B., Saha, S., & Goebel, K. (2010), Metrics for Offline Evaluation of Prognostic Performance, *International Journal of the PHM Society* (vol. 1).
Tang, L., Orchard, M. E., Goebel, K., & Vachtsevanos, G. (2011), Novel metrics and methodologies for the verification and validation of prognostic algorithms, *IEEE Aerospace Conference* (pp. 1-8), March 5-12.
Usynin, A., Hines, J. W., & Urmanov, A. (2007), Formulation of Prognostics Requirements, *IEEE Aerospace Conference* (pp. 1-8), March 3-10.
Wheeler, K. R., Kurtoglu, T., & Poll, S. D. (2010), A Survey of Health Management User Objectives Related to Diagnostic and Prognostic Metrics, *International Journal of the PHM Society* (vol. 1).
Zhang, B., Tang, L., DeCastro, J., & Goebel, K. (2010), A verification methodology for prognostic algorithms, *IEEE AUTOTESTCON* (pp. 1-8), September 13-16.
Zhou, Y., Bo, J., Jie, Z., & Mingwei, G. (2013), Performance Metrics Assessment Method on Aircraft Prognostics and Health Management, *Fifth International Conference on Measuring Technology and Mechatronics Automation (ICMTMA)* (pp. 799-802), January 16-17.

BIOGRAPHIES

N. Scott Clements is a Systems Engineer at Lockheed Martin Aeronautics Company. He is currently researching physics of failure fault models and associated PHM techniques. He received his bachelor's degree from Mississippi State University in 1996 and his master's and doctoral degrees from Georgia Institute of Technology in 1998 and 2003, respectively. His research interests include PHM, data mining, verification techniques, and fault tolerant control systems.

David S. Bodden is a Senior Fellow in Flight Control/Vehicle Management Systems at Lockheed Martin Aeronautics Company. He received his bachelor's degree in Aerospace Engineering from Texas A&M University in 1976, and his master's degree in Engineering Mechanics from Virginia Tech in 1984. His technical background includes structural dynamics, air vehicle conceptual design, and multi-variable/intelligent control synthesis techniques. His current research area is application of PHM technology to improving system reliability.