On Applying the Prognostic Performance Metrics

Abhinav Saxena¹, Jose Celaya², Bhaskar Saha³, Sankalita Saha⁴ and, Kai Goebel⁵

^{1,2}SGT Inc., NASA Ames Research Center, Intelligent Systems Division, MS 269-4, Moffett Field, CA 94035, USA <u>abhinav.saxena@nasa.gov</u> jose.r.celaya@nasa.gov

^{3,4}MCT Inc., NASA Ames Research Center, Intelligent Systems Division, MS 269-4, Moffett Field, CA 94035, USA <u>bhaskar.saha@nasa.gov</u> sankalita.saha@nasa.gov

⁵NASA Ames Research Center, Intelligent Systems Division, MS 269-4, Moffett Field, CA 94035, USA <u>kai.goebel@nasa.gov</u>

ABSTRACT

Prognostics performance evaluation has gained significant attention in the past few years. As prognostics technology matures and more sophisticated methods for prognostic uncertainty management are developed, a standardized methodology for performance evaluation becomes extremely important to guide improvement efforts in a constructive manner. This paper is in continuation of previous efforts where several new evaluation metrics tailored for prognostics were introduced and were shown to effectively evaluate various algorithms as compared to other conventional metrics. Specifically, this paper presents a detailed discussion on how these metrics should be interpreted and used. Several shortcomings identified while applying these metrics to a variety of real applications are also summarized along with discussions that attempt to alleviate these problems. Further, these metrics have been enhanced to include the capability of incorporating probability distribution information from prediction algorithms as opposed to evaluation based on point estimates only. Several methods have been suggested and guidelines have been provided to help choose one method over another based

on probability distribution characteristics. These approaches also offer a convenient and intuitive visualization of algorithm performance with respect to metrics like prediction horizon and α - λ performance, and also quantify the corresponding performance while incorporating the uncertainty information.

1. INTRODUCTION

Prognostics being an emerging research field, most of the published work has naturally been exploratory in nature, consisting mainly of proof-of-concepts and oneoff applications. Prognostic Health Management (PHM) has by-and-large been accepted by the engineered systems community in general, and the aerospace industry in particular, as the direction of the future. However, for this field to mature, it must make a convincing case in numbers to the decision makers in research and development as well as fielded applications. It is as Prof. Thomas Malone, an eminent management guru, said, "If you don't keep score, you are only practicing."

In research, metrics are not simply a means to evaluate the quality of an approach, they can be useful in a variety of different ways. One of the most direct uses is reporting performance both internally and externally with respect to the research organization. Metrics can create a standardized language with which technology developers and users can communicate their findings with each other and compare results. This aids in the dissemination of scientific information as well as decision making. Metrics may also be viewed as a feedback tool to close the loop on research and

^{*}Abhinav Saxena 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.

development by using them as objective functions to be minimized or maximized, as appropriate, by the research effort.

The multifarious uses notwithstanding, metrics can be a double-edged sword. An off repeated quote in the management world goes, "Be careful what you measure—you might just get it." What this saying means is that we mostly set goals based on what we can measure, and then we set about achieving those goals without heed to the inherent value of the work. It is usually much harder to ascertain value than it is to evaluate metrics. Thus, the careful choice and design of a metric to reflect the intended value is of paramount importance, especially in a nascent research area like prognostics.

Recently there has been a significant push towards crafting suitable metrics to evaluate prognostics performance. Researchers from academia and industry are working closely to arrive at useful performance measures. For instance, in [1] authors propose some prognostics metrics and compare them with diagnostics metrics. However, these metrics are mostly derived from metrics used for prediction tools in finance as opposed to being specifically tailored for prognostics. In [2] we categorized various forecasting applications based on their different characteristics and pointed out that there are notable differences other domains and the task of remaining life prediction. It is therefore, important to develop metrics that directly address the problem at hand. On the flipside it becomes quite challenging to reshape the mindset around using metrics from other forecasting domains. Wang and Lee [3] propose simple metrics in their paper from the classification discipline and also suggest a new metric called "Algorithm Performance Profile" that tracks the performance of an algorithm using the accuracy score each time an estimated RUL is estimated. In [4], authors present two new metrics for prognostics; in particular they define a reward function for predicting the correct time-to-failure that also takes into account prediction and problem detection coverage. They also propose a cost-benefit analysis based metric to quantify how much an organization could save by deploying a given prognostic model. Thus in general efforts are being made to evaluate prognostics from different enduser point of views.

This paper is a thematic continuation of previous works that surveyed metrics in use for prognostics in a variety of domains [2] in order to come up with a list of metrics to assess critical aspects of RUL predictions and showed how such metrics can be used to effectively assess the performance of prognostic algorithms [5]. This paper will focus on the design and choice of parameters for metrics that are specifically designed for prognostics beyond the conventional ones being used for diagnostics and other forecasting applications. These metrics have been introduced in [2] and their implementation discussed in [5]. Here we discuss the ways in which these metrics may be interpreted and used, and even misused or abused depending upon specific application scenarios. Furthermore, some enhancements over the original definitions have been presented to incorporate issues observed while applying these metrics to real applications.

The next section motivates our detailed analysis on prognostics metrics. This analysis is first presented in the context of prediction horizon and then extended to other metrics in section 3. Section 3 also presents enhancements on these metrics to include prediction distributions and methods to implement them. Finally, the paper discusses future directions in section 4 followed by conclusions in section 5.

2. MOTIVATION

In this paper we discuss the two fold benefits of performance metrics (see Figure 1).



Figure 1. Prognostics metrics are used for performance evaluation and also help in requirement specification

Given current TRL level for the prognostics technology, lack of assessment about prognosability of a system and concrete uncertaintv management approaches, managers of critical systems/applications have struggled to define concrete performance specifications. In most cases performance requirements are either derived from previous diagnostics experience or are very loosely specified. Prognostics metrics as proposed in [2] depend on various parameters that must be specified by the customer as requirements that an algorithm should attempt to meet as specifications. Process of coming up with reasonable values for these parameters must consider a complex interplay between several other factors as discussed in this paper. We

show, here, how these metrics can help users to come up with these specifications by taking such factors into account in a systematic framework. It is anticipated that in this manner these metrics will be useful for decision making in practical implementations of prognostics. On the other hand, providing feedback to algorithm developers and helping them improve their algorithms while trying to meet such specifications is yet another role these metrics are expected to play in a more conventional sense.

The new prognostics metrics developed in previous work require a change in thinking about what constitutes a good performance. More importantly the time varying aspect of performance, each time the estimates are updated, differentiates these metrics from other related domains. These metrics offer visual as well as quantitative assessment of performance as it evolves over time. The visual representation allows making several observations about the performance and it is necessary for us, now, to understand the capabilities and the limits of information these new metrics can provide. Therefore, we try to draw a scope where these metrics may be applicable, useful and also describe where the limitations may be.

With some initial experience we found ourselves under a dilemma between creating a comprehensive but complicated metric and a simple but less generic metric. The trade-off originates from the interplay between the ease of use, interpretability, and comprehensiveness. We determined that a more complicated metric has fewer chances of being adopted and more chances of breaking in many special cases that may not have been envisioned while formulating these ideas. Therefore, suggesting enhancements to the extent where these metrics are still simple enough to use and at the same time pointing out cases where these metrics are not expected to break is another objective of this paper. A significant enhancement presented in this paper is the ability of these metrics to incorporate uncertainty estimates available in the form of RUL distributions.

3. PROGNOSTIC PERFORMANCE METRICS

In this paper we discuss the four prognostics metrics namely; Prediction Horizon, α - λ Performance, Relative Accuracy, and Convergence. These four metrics follow a systematic progression in terms of the information they seek (Figure 2).



Figure 2. Hierarchical design of the prognostics metrics.

First, the prediction horizon identifies whether an algorithm predicts within a specified error margin (specified by the parameter α) around the actual end-oflife and if it does how much time it allows for any corrective action to be taken. In other words it assesses whether an algorithm yields a sufficient prognostic horizon and if not it may not even be meaningful to compute other metrics. Thus if an algorithm passes the PH test the α - λ Performance goes further to identify whether the algorithm performs within desired error margins (specified by the parameter α) of the actual RUL at any given time instant (specified by the parameter λ) that may be of interest to a particular application. This presents a more stringent requirement of staying within a converging cone of error margin as a system nears endof-life (EoL). If this criterion is also met, the next step is to quantify the accuracy levels relative to actual remaining useful life. These notions assume that prognostics performance improves as more information becomes available with time and hence by design an algorithm will satisfy these metrics criteria if it converges to true RULs. Therefore, the fourth metric Convergence quantifies how fast the algorithm converges provided it satisfies all the previous metrics. The group of these metrics can be considered a hierarchical test that yields several levels for comparison among different algorithms in addition to the specific information these metrics provide individually regarding algorithm performance.

Since, these metrics share the attribute of performance tracking with time; we first develop our discussions using Prediction Horizon as an example. These discussions are then extended to the rest three with additional details specific to individual metrics.

3.1 Prediction Horizon

Prognostic Horizon is defined as the difference between the time index i when the predictions first meet the specified performance criteria (based on data accumulated until time index i) and the time index for End-of-Life (EoL). The performance specification may be specified in terms of allowable error bound (α) around true EoL.

$$PH = EoL - i, \qquad (1)$$

where:

 $i = \min\{j \mid (j \in \ell) \land (r_*(1-\alpha) \le r^l(j) \le r_*(1+\alpha))\}$ is the first time index when predictions satisfy α -bounds

 ℓ is the set of all time indexes when a prediction is made

l is the index for l^{th} unit under test (UUT)

 r_* is the ground truth RUL

Prediction horizon produces a score that depends on length of ailing life of a system and the time scales in the problem at hand. The range of PH is between $(t_{EoL}-t_P)$ and $max[0, t_{EoL}-t_{EoP}]$, where the best score can be obtained when the algorithm always predicts within desired accuracy zone and the worst score when it never predicts within the accuracy zone.

3.1.1 What can be inferred from the metric

The notion for Prediction Horizon has been long discussed in the literature from a conceptual point of view. This metric indicates whether the predicted estimates are within specified limits around the actual EoL so that the predictions are considered trustworthy. It is clear that longer the prognostics horizon more time is available to act based on a prediction that has some desired credibility. Therefore, while comparing algorithms, an algorithm with longer prediction horizon would be preferred.

As shown in Figure 3, the desired level of accuracy with respect to the EoL ground truth is specified as $\pm a$ -bounds. RUL values are then plotted against time for various algorithms that are being compared. The PH for an algorithm is declared as soon the corresponding predictions enter the band of desired accuracy. As clearly evident from the illustration, the first algorithm has a longer PH.



Figure 3. Prognostic Horizon.

3.1.2 Issues

There are several cases where standard definition for PH breaks from a practical point of view and declaring a PH may not be straight forward. We discuss some such cases next and suggest possible ways to deal with them.

RUL trajectory jumps out of the accuracy zone: Based on our experience and feedback received from fellow researchers while applying these metrics to several applications there are often cases where RULs jump in and out of the $\pm \alpha$ accuracy zone. In such cases it may not be appropriate to declare the PH at a time instant where RUL enters within $\pm \alpha$ accuracy zone for the very first time and then jumps out again. As illustrated in Figure 4, for both examples predictions at time instant c jump out of the accuracy zone and get back in at a later prediction step. In absence of such an anomaly the PHs for these algorithms would have been declared at times a and b. A situation like this results in multiple time indexes when RUL trajectory enters the accuracy zone. A simple approach to deal with this situation can be being more conservative and declaring PH at the latest time instant the predictions enter accuracy zone. Another option is to use the original PH definition and evaluate other metrics to determine if the algorithm satisfies other requirements. To avoid confusions we recommend using the original definition. This will encourage practitioners to go back to the algorithm development stage and improve their prediction process to incorporate capabilities to deal with such anomalies in their algorithms.



Figure 4. Cases where RUL predictions do not stay consistently within the accuracy zone.

Situations like these can occur due to various reasons as listed below and it is important to identify the correct one before computing a PH.

- Inadequate system model: Real systems often exhibit inherent transients at different stages of their lives. These transients get reflected as deviations in computed RUL estimates from the true value if the underlying model assumed for the system does not account for these behaviors. For example, in [5] authors describe an application of Li-ion battery health management where the capacity decay shows such transient behaviors in the beginning and the end phases of the battery life. Their examples show cases where RUL trajectories jump away from ground truth whenever such transient phases occur. Therefore, for situations as depicted in Figure 4 one must go back and refine their models to incorporate such anomalies.
- Operational transients: Another source of such behaviors can be due to sudden changes in operational profiles under which a system is operating. Prognostic algorithms may show a time lag in adapting to such changes and hence resulting in temporary deviation from the real values.
- Uncertainties in prognostic environments: Prognostics is inevitably surrounded bv uncertainties arising from a variety of sources. This makes prognostics inherently a stochastic process and hence the behavior observed from a particular run may not exhibit the true nature of prediction trajectories. This discussion assumes that all measures for uncertainty reduction have already been taken during algorithm development and that such observations are an isolated realization of the process. In that case these trajectories should be obtained based on multiple runs to achieve

statistical significance or more sophisticated stochastic analyses can be carried out.

Before one arrives at the final assessment for PH metric, a situation like the one discussed above helps pinpoint the exact reason for such behaviors. Whenever such behavior is observed one must go back and identify the most probable cause and try to improve the models, fine tune algorithms, or better the experimental design as the situation may demand. A robust algorithm and a system model should be capable of taking care of transients inherent to the system behavior and operational conditions. Plotting the RUL trajectory in prediction horizon plot provides clues regarding such deficiencies to algorithm developers. Once these deficiencies are taken care of there is a good chance that such behavior disappears and a PH can be easily determined, otherwise the simple but conservative approach, discussed earlier, may be used.

RUL trajectory jumps out close to EoL: other situations that were reported included cases where one observes a well behaved converging behavior for the RUL trajectory for most of the ailing life except at the very end when they jump out of the accuracy zone (see Figure 5).



Figure 5. Prediction behavior after EoUP is practically inconsequential and hence need not affect the PH.

In [5] authors attribute such behavior to system transients that were not modeled well by some of the data-driven algorithms that were used. To deal with situations we introduce a new concept of "*useful predictions*". All engineered systems undergo non-linear dynamics during fault progression, leading to a system failure at t_{EoL} . More often than not these dynamics are difficult to model or learn from data as the system nears the failure point. Thus, it is possible when evaluating the PH metric for a particular algorithm in a given application that the RUL curve deviates away from the error band near t_{EoL} , having entered it earlier during its trajectory. In such a case, it may be counterproductive to bias ourselves against an algorithm which has a very

small or no PH, since we would be ignoring the algorithm's performance elsewhere on the RUL curve. Consequently it may be prudent to evaluate PH on a error band that is limited in extent on the time x-axis by the time instant t_{EoUP} , which denotes the End-of-Useful-Predictions (EoUP), such that we ignore the region near t_{EoL} , within which it is impossible to take any corrective action based on the RUL prediction and these predictions are of little or no use practically. The value of t_{EoUP} chosen is dependent upon the application, the time and cost for possible redress actions in that domain. In other words EoUP determines the lower limit on acceptable range for PH in a given application.

As an example, Figure 6 shows the results for 4 different algorithms in predicting battery life. All algorithms except Relevance Vector Machines (RVM) deviate away from the true RUL whereas RVM reaches very close to true RUL near EoL. But at the same time RVM results in a shorter PH. In cases depending on how much time may be needed to repair/replace the battery would determine the EoUP and hence a definition of a better PH.



Figure 6. An example showing different cases in a real application [5].

3.1.3 Guidelines for using the metrics

The main idea behind these metrics is not only to compare different algorithms for performance evaluation but also to help management decide on specifications and requirements on prognostics algorithm in the fielded applications. The outcome of the metric depends directly on the values chosen for input parameters like α . To that end, we describe how α can be chosen for a specific application.

Prognostic horizon emphasizes on the time critical aspects of prognostics. A prediction for catastrophic event ahead of time is meaningful only if a corrective action can be completed before the system fails. Keeping the description generic, there are systems that involve different levels of criticality when they fail. In a mission critical scenario a failure may be catastrophic and hence a limited number of false positives may be tolerable whereas in other cases cost of acting on false positives may be prohibitively high. There are even cases where it is more cost effective if there are several false negatives as opposed to reacting to a false positive and hence it is acceptable even if the system runs to failure once in a while. There are several factors that determine how critical it may be to make a correct prediction. These factors combined together should dictate the choice of α while implementing PH for performance evaluation. We list here some of the most important such factors.

- Time for problem mitigation: the amount of time that takes to mitigate a problem or start a corrective action when critical health deterioration of a component/system has been detected is a very important factor. As mentioned earlier, very accurate predictions at a time when no recovery action can be made is not useful. Hence, a tradeoff between error tolerance and time for recovery from fault should be considered. The time for problem mitigation will vary from system to system and involves multiple factors.
- 2) Cost of mitigation: cost of the reparative action is an important factor in all management related decisions and hence should be considered while determining α .
- 3) Criticality of system or cost of failure (false positive): In time-critical applications, resources should be directed towards more critical and important components in order to efficiently maintain overall health of the system. Hence, if health assessment is being performed on multiple units in a system, α for the different units should be chosen based on a prioritized list of criticality.

Note that the factors mentioned above are not arranged based on any order of importance and users should utilize them based on characteristics of their systems and may skip a few of them. We denote the combination of factors used in determining α as "recovery cost" or t_{recovery} .

3.1.4 Recipe to choose a

A suitable value for α can be chosen following few simple steps.

Step 0: Initialize α based on past experience (or arbitrarily if no past information available) Step 1: Plot RULs and compute PH as per the definition.

- Step 2: If the PH is much larger than $t_{\rm repair}$ repeat steps 1 2 by reducing α until PH is close to $t_{\rm repair}$.
- Step 3: If in step 1, PH is already smaller than $t_{\rm repair}$, increase α until a desired length of PH is obtained.

This procedure provides a range for suitable values of α and one can choose to operate at a particular value based on amount of risk that may be acceptable by being closer to EoL if a smaller α is desirable.

This procedure accounts for the requirements from logistics point of view of the health management and does not take into account the factor of *prognosability* of the system. There may be limitations on how well a prognostics algorithm may be able to manage and reduce uncertainty and hence is limited by a lower bound on the best achievable precision. Although the preferred approach should be to improve the algorithm to meet specified α requirement wherever possible, but if due to the nature of the problem no algorithm can meet the specifications one needs to relax the specifications by choosing a larger α from the range obtained above.

3.1.5 Incorporating probability distributions of predictions

In previous work presented in [2] and [5] we presented the definitions and examples of performance metrics considering that the prognostics algorithm provides a RUL prediction r(k) represented by a single point. This assumed that such prediction is deterministic or that an algorithm includes additional reasoning to compute a single point estimate of the prediction distribution. Given that there are multiple sources of uncertainties inherent to the prognostics problem, it is expected/required that a prognostics algorithm provides information about the confidence around the prediction. This confidence can be represented in several ways. There are algorithms that provide an approximation of the probability distribution of the RUL, r(k) at any point k by providing a set of discrete samples of r(k) with their corresponding probabilities [6]. Other algorithms that rely on Guassian assumptions describe the uncertainty by providing the mean and variance of a normally distributed r(k) prediction [7]. In some cases where multimodal distributions are obtained an approximation with mixture of Gaussians has been considered to derive at the distribution characteristics [8].

Generally, a common way to describe a distribution is based on the first two moments. The mean is an indication of central tendency or location

and the variance is an indication of the spread of the distribution. These quantities completely summarize Gaussian distributions. For cases were normality cannot be claimed, one can rely on median as a measure of location and the quartiles or inter quartile range as a measure of spread [9].

Prognostics metrics like prediction horizon and α - λ performance provide a great deal of visual information in addition to answers that one seeks at specific time instances. Therefore, incorporating enhanced visual representations for prediction distributions improves the efficacy of these metrics for performance comparison. For cases involving Normal distribution, including a confidence interval represented by an error bar around the point prediction is useful [10]. For cases with non-Normal single mode distributions this can be done with an inter-quartile plot represented by a box plot [11]. This conveys how a prediction distribution is skewed and whether these skew should be considered while declaring prediction horizon. Box plot also has provisions to represent outliers that may be useful to keep track of in critical situations (Figure 7).



Figure 7. Representations for distributions.

In addition to visual enhancements, distribution information can be better utilized by computing total probability of a prediction falling within the specified α bounds as against using a point estimate to compute a metric. This concept has been depicted in Figure 8 with original point prediction superimposed on box plots.



Figure 8. Enhanced representation for prognostic horizon incorporating distribution information.

The original PH metric assumed a single point prediction for an output of the prognostics algorithm. This ignores uncertainty information even if algorithms provide this information as distributions and does not allow a fair comparison for situations where a prediction is very close to the alpha bound but not quite inside it. Clearly as shown in Figure 9, a larger PH can be obtained for the same case if the RUL distribution satisfies β criterion. Therefore, it is desirable to take advantage of the uncertainty information and use it to declare PH even if the point prediction does not fall within the bound but is very close.



Figure 9. Integrating the probabilities of RUL falling in α -bound can be used to decide declaration of PH

Method to compute the metric: as mentioned above, computing the metric now requires integrating the probability distribution that overlaps with the desired region to compute the total probability. For cases where analytical form of the distribution is available, like for Normal distributions, it can be computed analytically by integrating the area under the prediction pdf between the α -bounds (α to α^+). However, for cases where there is no analytical form available, a summation based on histogram obtained from the process/algorithm can be used to compute total probability. This procedure has been pictorially depicted in Figure 10.

An important question still remains to be answered is what should one use as the representations for location and spread. For simple cases like Normal distributions this is straightforward, however, for other cases this may not be very clear. As outlined in Table 1, there are four main categories a distribution may be assigned to, which can be further classified under parametric and nonparametric subclasses. This sub classification mainly determines the method of computing the total probability, i.e. continuous integration or discrete summation. It is suggested to use box plots along with a dot representing the mean of the distribution, which will allow keeping the visual information in perspective with respect to original plots. For mixture of Gaussians case, it is suggested that a model with few (preferably $n \leq 4$) Gaussians is created and corresponding error bars plotted adjacent to each other.

$$\phi(x) \cong \lambda_1 \cdot N(\mu_1, \sigma_1) + \dots + \lambda_n \cdot N(\mu_n, \sigma_n); n \in I^+ \quad (2)$$

where:

 λ is the weight factor for each Gaussian component

 $N(\mu, \sigma)$ is a Gaussian distribution with mean μ and standard deviation σ .

The weights for each Gaussian component can then be represented by the thickness of the error bars. We do not recommend multiple box plots in this case as there is no methodical way to differentiate between samples, assign them to particular Gaussian components and compute the quartile ranges for each of them. Also, to keep things simple we assume a linear additive model while computing the mixture of Gaussians.



Figure 10. Procedure to compute total probability of RULs being within specified α-bounds.

	Normal Distribution	Mixture of Gaussians	Non-Normal distribution	Multimodal (non normal)
	Parametric		Non-Parametric	
Location (central tendency)	Mean (µ)	Means: $\mu_1, \mu_2,, \mu_n$	Mean, Median,	Dominant median, Multiple medians,
		weights: $\lambda_1, \lambda_2, \dots, \lambda_n$	L-estimator [9], M-estimator [9]	L-estimator, M-estimator
Spread (variability)	Sample standard deviation (σ), IQR (inter quartile range)	Sample standard deviations: $\sigma_1, \sigma_2,, \sigma_n$	Mean Absolute Deviation (MAD) $\left(\frac{1}{n}\sum_{i=1:n} x_i - median \right)$, Median Absolute Deviation (MdAD) $median(x_i - median)$, Bootstrap methods , IQR	

Table 1 – Recipe to select location and spread measures along with visualization methods



In this section we have described in details various aspects of prognostics metrics in the context of prognostic horizon. Many of these concepts naturally transfer to the other metrics and therefore for the sake of conciseness and minimum repeatability, we will very briefly point out other salient features of the rest of the three metrics and frequently refer to the discussion above.

3.2 $\alpha - \lambda$ Performance

Performance quantifies prediction quality α-λ hv determining whether the prediction falls within specified limits at particular times. These time instances may be specified as percentage of total ailing life of the system. Any performance measure of interest may fit in this framework. In general, so far we have used accuracy as the main performance measure. In our implementation we define α - λ accuracy as the prediction accuracy to be within α *100% of the actual RUL at specific time instance t_i expressed as a fraction of time between the point when an algorithm starts predicting and the actual failure. For example, this metric determines whether a prediction falls within 10% accuracy (i.e., $\alpha=0.1$) halfway to failure from the time the first prediction is made (i.e., $\lambda = 0.5$). Therefore, one needs to evaluate whether the following condition is met.

$$(1-\alpha) \cdot r_*(t) \le r^l(t_\lambda) \le (1+\alpha) \cdot r_*(t) \tag{3}$$

where:

 α is the accuracy modifier

 λ is a time window modifier such that

$$t_{\lambda} = t_{P} + \lambda (EoL - t_{P}).$$

The output of this metric is binary (Yes or No) stating whether the desired condition is met at a particular time. This is a more stringent requirement as compared to prediction horizon as it requires predictions to stay within a cone of accuracy i.e. the bounds that shrink as time passes by. With the new enhancements, it is also possible to compute total probability overlapping with this cone to determine whether the criteria are met (Figure 11a). For easier interpretability α - λ accuracy can also be plotted as shown in Figure 11b. The concept of α - λ precision is further illustrated in Figure 12. The choice of precision measure may be application specific or based on the type of distribution. This plot shows how precision evolves with time and whether satisfies a given level at a specified time instant t_i .



Figure 11. (a) α - λ accuracy with error bars (b) alternative representation of α - λ accuracy.



Figure 12. Concept of α - λ precision

3.3 Relative Accuracy

Relative prediction accuracy is a notion similar to α - λ accuracy where, instead of finding out whether the predictions fall within given accuracy levels at a given time instant, we also quantitatively measure the accuracy. The time instant is again described as a fraction of the ailing life. An algorithm with higher relative accuracy is desirable. The range of values for RA is [0,1], where the perfect score is 1. It must be noted that if the prediction error magnitude grows beyond 100% RA gives a negative value. We do not consider such cases since these cases would not have qualified the first two tests in the first place.

$$RA_{\lambda} = 1 - \frac{\left|r_{*}(t_{\lambda}) - r^{I}(t_{\lambda})\right|}{r_{*}(t_{\lambda})} \tag{4}$$



Figure 13. Schematic showing Relative Accuracy concept.

RA conveys information at a specific time. However, to account for general behavior of the algorithm over time Cumulative Relative Accuracy (CRA) can be used. Relative accuracy can be evaluated at multiple time instances before t_{λ} . To aggregate these accuracy levels, we define Cumulative Relative Accuracy as a normalized weighted sum of relative prediction accuracies at specific time instances.

$$CRA_{\lambda} = \frac{1}{\left|\ell_{\lambda}\right|} \sum_{i=1}^{\ell_{\lambda}} w(r^{i}) RA_{\lambda}$$
(5)

where:

 $w(r^{l})$ is a weight factor as a function of RUL at all time indices

 ℓ_{λ} is the set of all time indexes before t_{λ} when a prediction is made

 $|\ell_{2}|$ is the cardinality of the set.

In most cases it is desirable to weigh the relative accuracies higher closer to the EoL. In general it is expected that t_{λ} is chosen such that it holds some physical significance such as a time index that provides required prediction horizon, or time required to apply a corrective action, etc. For instance RA evaluated at $t_{0.5}$ signifies the time when a system is expected to have consumed half of its ailing life, or in terms of damage index the time index when damage magnitude has reached 50% of the failure threshold. This metric is useful in comparing different algorithms for a given λ to get an idea on how well a particular algorithm does when it is critical. Choice of t_{λ} should also take into account the uncertainty levels that an algorithm entails by making sure that distribution spread at t_{λ} does not cross over expected EoL by significant margins especially for critical applications.

On the issue of incorporating distribution information for RA, one can make an informed decision on choosing a righteous measure of accuracy as against to choosing only the mean value. As pointed out in Table 1, the shape of RUL distribution should guide the selection of location indicator. This choice should also consider the nature of the application. For instance a critical application where risk tolerance level may be low one should choose an indicator that weighs the tails importantly and even outliers in some cases.

3.4 Convergence

Convergence is defined to quantify the manner in which any metric like accuracy or precision improves with time to reach its perfect score. As suggested earlier, our discussion assumes that the algorithm performance improves with time, i.e. has passed all previous tests. For illustration of the concept we show three cases that converge at different rates. It can be shown that the distance between the origin and the centroid of the area under the curve for a metric quantifies convergence. Lower distance means a faster convergence. Convergence is a useful metric since we expect a prognostics algorithm to converge to the true value as more information accumulates over time. Further, a faster convergence is desired to achieve a high confidence in keeping the prediction horizon as large as possible.

Let (x_c, y_c) be the center of mass of the area under the curve M(i). Then, the convergence C_M can be represented by the Euclidean distance between the center of mass and $(t_p, 0)$, where

$$C_M = \sqrt{(x_c - t_P)^2 + {y_c}^2},$$

$$\mathbf{x}_{c} = \frac{\frac{1}{2} \sum_{i=P}^{EoP} (t_{i+1}^{2} - t_{i}^{2}) M(i)}{\sum_{i=P}^{EoP} (t_{i+1} - t_{i}) M(i)}, \text{ and}$$

$$\mathbf{y}_{c} = \frac{\frac{1}{2} \sum_{i=P}^{EoP} (t_{i+1} - t_{i}) M(i)^{2}}{\sum_{i=P}^{EoP} (t_{i+1} - t_{i}) M(i)}.$$
(6)

M(i) is a non-negative prediction error accuracy or precision metric.



Figure 14. Schematic for the convergence of a metric.

4. FUTURE WORK

We would like to conduct a quantitative analysis for comparison of various approaches suggested in this paper using a real application scenario and identify some key characteristics that may dictate the choice of one over another. We will investigate how performance estimates get affected by choosing different options of integrating the uncertainty estimates. This will allow us to identify the advantages and limitations of these techniques and their applicability towards a standardized performance evaluation method.

So far, the performance evaluation assumes that future loading conditions do not change or at least do not change the rate of fault growth. For offline studies this may be reasonable as we know the actual EoL index and can linearly extrapolate true RUL for all previous time indices to draw a straight line. However, for real-time applications this would not hold true as changes in operating conditions do affect the rate of fault evolution. Hence, we would also like investigate about how to incorporate effects of changes in the loading conditions that alter the RUL slope by changing the rate of remaining life consumption.

We will continue to refine the concepts presented in this paper and apply them to a variety of applications in addition to developing more metrics. Developing more metrics like robustness and sensitivity, etc. remains on our research agenda.

5. CONCLUSION

In this paper we have presented a detailed analysis on how prognostics metrics should be used and interpreted. Based on feedback available from fellow researchers, who applied these metrics to a variety of applications, several refinements were carried out. Various cases were pointed out and discussed where these metrics may present an ambiguous situation, while making decisions. A detailed recipe was presented on how to select various parameters for these metrics on which the evaluation outcome depends. Furthermore, it was shown that these metrics are not only useful for algorithmic performance evaluation but also for coming up with performance specifications while keeping several critical constraints in mind. A detailed discussion on ways to include prediction distribution information for visual enhancements and more robust performance evaluation was presented. It is expected that this paper will greatly enhance the understanding of these performance metrics and encourage a wider community to use these metrics and help standardize the prognostics performance evaluation.

ACKNOWLEDGMENTS

The authors would like to express their gratitude to colleagues at the Prognostic Center of Excellence (NASA Ames Research Center) and external partners at Impact Technologies, and Clarkson University, for participating in research discussions, evaluating metrics in their respective applications, and providing a valuable feedback.

REFERENCES

- B. P. Leao, T. Yoneyama, G. C. Rocha *et al.*, "Prognostics performance metrics and their relation to requirements, design, verification and cost-benefit," in Prognostics and Health Management (PHM08), International Conference on, Denver, CO, 2008, pp. 1-8.
- [2] A. Saxena, J. Celaya, E. Balaban *et al.*, "Metrics for evaluating performance of prognostic techniques," in Prognostics and Health Management (PHM08), International Conference on, Denver CO, 2008, pp. 1-17.
- [3] T. Wang, and J. Lee, "On Performance Evaluation of Prognostics Algorithms," in Machinery Failure Prevention Technology, Dayton OH, 2009.
- [4] C. Yang, and S. Letourneau, "Model evaluation for prognostics: estimating cost saving for the end users," in Sixth International Conference on Machine Learning and Applications, 2007, pp. 304-309.
- [5] A. Saxena, J. Celaya, B. Saha *et al.*, "Evaluating Algorithmic Performance Metrics Tailored for Prognostics."
- [6] M. E. Orchard, and G. J. Vachtsevanos, "A particlefiltering approach for on-line fault diagnosis and failure prognosis," *Transactions of the Institute of Measurement and Control*, vol. 31, no. 3-4, pp. 221-246, 2009.
- [7] K. Goebel, B. Saha, and A. Saxena, "A Comparison of Three Data-Driven Techniques for Prognostics," in 62nd Meeting of the Society For Machinery Failure Prevention Technology (MFPT), Virginia Beach, VA, 2008, pp. 191-131.

- [8] B. Saha, K. Goebel, S. Poll *et al.*, "Prognostics Methods for Battery Health Monitoring Using a Bayesian Framework," *IEEE Transactions on Instrumentation and Measurement*, vol. 58, no. 2, 2009.
- [9] D. C. Hoaglin, F. Mosteller, and J. W. Tukey, Understanding Robust and Exploratory Data Analysis: Wiley, 1983.
- [10] J. L. Devore, Probability and Statistics for Engineering and the Sciences, 6th ed.: Thomson, 2004.
- [11] A. R. Martinez, "Exploratory data analysis with MATLAB," A. R. Martinez, ed., CRC Press, 2004.

BIOGRAPHIES

Abhinav Saxena is a Research Scientist with SGT Inc. at the Prognostics Center of Excellence NASA Ames Research Center, Moffet Field CA. His research focus lies in developing and evaluating prognostic algorithms for engineering systems using soft computing techniques. He is a PhD in Electrical and Computer Engineering from Georgia Institute of Technology, Atlanta. He earned his B.Tech in 2001 from Indian Institute of Technology (IIT) Delhi, and Masters Degree in 2003 from Georgia Tech. Abhinav has been a GM manufacturing scholar and is also a member of IEEE, AAAI and ASME.

Jose R. Celaya is a staff scientist with Stinger Ghaffarian Technologies at the Prognostics Center of Excellence, NASA Ames Research Center. He received a Ph.D. degree in Decision Sciences and Engineering Systems in 2008, a M. E. degree in Operations Research and Statistics in 2008, a M. S. degree in Electrical Engineering in 2003, all from Rensselaer Polytechnic Institute, Troy New York; and a B.S. in Cybernetics Engineering in 2001 from CETYS University, Mexico.

Bhaskar Saha is a Research Scientist with Mission Critical Technologies at the Prognostics Center of Excellence NASA Ames Research Center. His research is focused on applying various classification, regression and state estimation techniques for predicting remaining useful life of systems and their components. He has also published a fair number of papers on these topics. Bhaskar completed his PhD from the School of Electrical and Computer Engineering at Georgia Institute of Technology in 2008. He received his MS from the same school and his B. Tech. (Bachelor of Technology) degree from the Department of Electrical Engineering, Indian Institute of Technology, Kharagpur.

Sankalita Saha received her B.Tech (Bachelor of Technology) degree in Electronics and Electrical Communication Engineering from Indian Institute of Technology, Kharagpur, India in 2002 and Ph.D. degree in Electrical and Computer Engineering from University of Maryland, College Park in 2007. She is currently a Research scientist working with Mission Critical Technologies at NASA Ames Research Center, Moffett Field, CA. Her research interests are in prognostics algorithms and architectures, distributed systems, and system synthesis.

Kai Goebel received the degree of Diplom-Ingenieur from the Technische Universität München, Germany in 1990. He received the M.S. and Ph.D. from the University of California at Berkeley in 1993 and 1996, respectively. Dr. Goebel is a senior scientist at NASA Ames Research Center where he leads the Diagnostics & Prognostics groups in the Intelligent Systems division. In addition, he directs the Prognostics Center of Excellence and he is the Associate Principal Investigator for Prognostics of NASA's Integrated Vehicle Health Management Program. He worked at General Electric's Corporate Research Center in Niskayuna, NY from 1997 to 2006 as a senior research scientist. He has carried out applied research in the areas of artificial intelligence, soft computing, and information fusion. His research interest lies in advancing these techniques for real time monitoring, diagnostics, and prognostics. He holds eleven patents and has published more than 100 papers in the area of systems health management.