

# Metrics for Offline Evaluation of Prognostic Performance

Abhinav Saxena<sup>1</sup>, Jose Celaya<sup>1</sup>, Bhaskar Saha<sup>2</sup>,  
Sankalita Saha<sup>2</sup>, and Kai Goebel<sup>3</sup>

<sup>1</sup>*SGT Inc., NASA Ames Research Center, Intelligent Systems Division, Moffett Field, CA 94035, USA*  
*abhinav.saxena@nasa.gov*  
*jose.r.celaya@nasa.gov*

<sup>2</sup>*MCT Inc., NASA Ames Research Center, Intelligent Systems Division, MS 269-4, Moffett Field, CA 94035, USA*  
*bhaskar.saha@nasa.gov*  
*sankalita.saha-1@nasa.gov*

<sup>3</sup>*NASA Ames Research Center, Intelligent Systems Division, MS 269-4, Moffett Field, CA 94035, USA*  
*kai.goebel@nasa.gov*

## ABSTRACT

Prognostic performance evaluation has gained significant attention in the past few years. Currently, prognostics concepts lack standard definitions and suffer from ambiguous and inconsistent interpretations. This lack of standards is in part due to the varied end-user requirements for different applications, time scales, available information, domain dynamics, etc. to name a few. The research community has used a variety of metrics largely based on convenience and their respective requirements. Very little attention has been focused on establishing a standardized approach to compare different efforts. This paper presents several new evaluation metrics tailored for prognostics that were recently 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. These metrics have the capability of incorporating probabilistic uncertainty estimates from prognostic algorithms. In addition to quantitative assessment they also offer a comprehensive visual perspective that can be used in designing the prognostic system. Several methods are suggested to customize these metrics for different applications. Guidelines are provided to help choose one method over another based on distribution characteristics. Various issues faced by prognostics and its performance evaluation are discussed followed by a formal notational framework to help standardize subsequent developments.

---

\* 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.

## 1. INTRODUCTION

In the systems health management context, prognostics can be defined as predicting the Remaining Useful Life (RUL) of a system from the inception of a fault based on a continuous health assessment made from direct or indirect observations from the ailing system. By definition prognostics aims to avoid catastrophic eventualities in critical systems through advance warnings. However, it is challenged by inherent uncertainties involved with future operating loads and environment in addition to common sources of errors like model inaccuracies, data noise, and observer faults among others. This imposes a strict validation requirement on prognostics methods to be proven and established though a rigorous performance evaluation before they can be certified for critical applications.

Prognostics can be considered an emerging research field. Prognostic Health Management (PHM) has in most respects been accepted by the engineered systems community in general, and by the aerospace industry in particular, as a promising avenue for managing the safety and cost of complex systems. However, for this engineering field to mature, it must make a convincing business case to the operational decision makers. So far, in the early stages, focus has been on developing prognostic methods themselves and very little has been done to define methods to allow comparison of different algorithms. In two surveys on methods for prognostics, one on data-driven methods (Schwabacher, 2005) and one on artificial-intelligence-based methods (Schwabacher & Goebel, 2007), it can be seen that there is a lack of standardized methodology for performance evaluation and in many cases performance evaluation is not even formally addressed. Even the current ISO standard by International Organization for Standards (ISO, 2004) for prognostics in condition monitoring and diagnostics of machines lacks a firm

definition of any such methods. A dedicated effort to develop methods and metrics to evaluate prognostic algorithms is needed.

Metrics can create a standardized language with which technology developers and users can communicate their findings and compare results. This aids in the dissemination of scientific information as well as decision making. Metrics could also be viewed as a feedback tool to close the loop on research and development by using them as objective functions to be optimized as appropriate by the research effort.

Recently there has been a significant push towards crafting suitable metrics to evaluate prognostic performance. Researchers from government, academia, and industry are working closely to arrive at useful performance measures. With these objectives in mind a set of metrics have been developed and proposed to the PHM community in the past couple years (Saxena, Celaya, Saha, Saha, & Goebel, 2009b). These metrics primarily address algorithmic performance evaluation for prognostics applications but also have provisions to link performance to higher level objectives through performance parameters. Based on experience gained from a variety of prognostic applications these metrics were further refined. The current set of prognostics metrics aim to tackle offline performance evaluation methods for applications where run-to-failure data are available and true End-of-Life (EoL) is known *a priori*. They are particularly useful for the algorithm development phase where feedback from the metrics can be used to fine-tune prognostic algorithms. These metrics are continuously evolving and efforts are underway towards designing on-line performance metrics. This will help associate a sufficient degree of confidence to the algorithms and allow their application in real *in-situ* environments.

### 1.1 Main Goals of the Paper

This paper presents a discussion on prognostics metrics that were developed in NASA's Integrated Vehicle Health Management (IVHM) project under the Aviation Safety program (NASA, 2009). The paper aims to make contribution towards providing the reader with a better understanding of:

- the need for separate class of prognostic performance metrics
- difference in user objectives and corresponding needs from a performance evaluation view point
- what can or cannot be borrowed from other forecasting related disciplines
- issues and challenges in prognostics and prognostic performance evaluation
- key prognostic concepts and a formal definition of a prognostic framework

- new performance evaluation metrics, their application and interpretation of results
- research issues and other practical aspects that need to be addressed for successful deployment of prognostics

### 1.2 Paper Organization

Section 2 motivates the development of prognostic metrics. A comprehensive literature review of performance assessment for prediction/forecasting applications is presented in section 3. This section also categorizes prognostic applications in several classes and identifies the differences from other forecasting disciplines. Key aspects for prognostic performance evaluation are discussed in Section 4. Technical development of new performance metrics and a mathematical framework for the prognostics problem are then presented in detail in Section 5. Section 6 follows with a brief case study as an example for application of these metrics. The paper ends with future work proposals and concluding discussions in Sections 7 and 8 respectively.

## 2. MOTIVATION

This research is motivated by two-fold benefits of establishing standard methods for performance assessment (see Figure 1). One, it will help create a foundation for assessing and comparing performance of various prognostics methods and approaches as far as low level algorithm development is concerned. Two, from a top-down perspective, it will help generate specifications for requirements that are imposed by cost-benefit and risk constraints at different system lifecycle stages in order to ensure safety, availability, and reliability. In this paper we discuss these metrics primarily in the context of the first benefit and only a brief discussion is provided on requirements specification.

### 2.1 Prognostic Performance Evaluation

Most of the published work in the field of prognostics has been exploratory in nature, such as proof-of-concepts or one-off applications. A lack of standardized guidelines has led researchers to use common accuracy and precision based metrics, mostly borrowed from the diagnostics domain. In some cases these are modified on an ad hoc basis to suit specific applications. This makes it rather difficult to compare various efforts and choose a winning candidate from several algorithms, especially for safety critical applications. Research efforts are focusing on developing algorithms that can provide a RUL estimate, generate a confidence bound around the predictions, and be easily integrated with existing diagnostic systems. A key step in successful

deployment of a PHM system is prognosis certification. Since prognostics is still considered relatively immature (as compared to diagnostics), more focus so far has been on developing prognostic methods rather than evaluating and comparing their performances. Consequently, there is a need for dedicated attention towards developing standard methods to evaluate prognostic performance from a viewpoint of how post prognostic reasoning will be integrated into the health management decision making process.

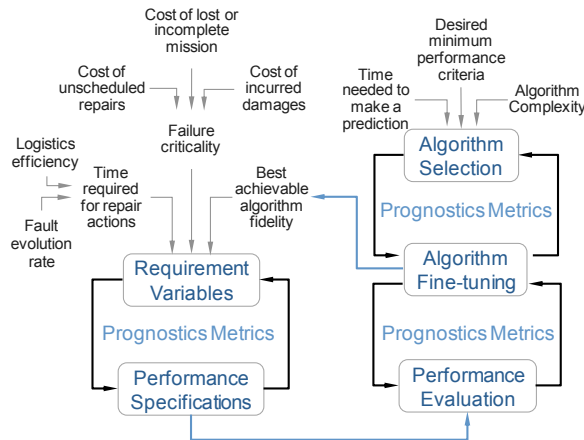


Figure 1: Prognostics metrics facilitate performance evaluation and also help in requirements specification.

## 2.2 Prognostic Requirements Specification

Technology Readiness Level (TRL) for the current prognostics technology is considered low. This can be attributed to several factors lacking today such as

- assessment of *prognosability* of a system,
- concrete Uncertainty Representation and Management (URM) approaches,
- stringent Validation and Verification (V&V) methods for prognostics
- understanding of how to incorporate risk and reliability concepts for prognostics in decision making

Managers of critical systems/applications have consequently struggled while defining concrete prognostic performance specifications. In most cases, performance requirements are either derived from prior experiences like diagnostics in Condition Based Maintenance (CBM) or are very loosely specified. This calls for a set of performance metrics that not only encompass key aspects of predicting into the future but also accommodate notions from practical aspects such as logistics, safety, reliability, mission criticality, economic viability, etc. The key concept that ties all these notions in a prognostic framework is of *performance tracking* as time evolves while various trade-offs continuously arise in a dynamic situation. The prognostics metrics presented in this paper are

designed with intentions to capture these salient features. Methodology for generating requirements specification is beyond the scope of this paper and only a brief discussion explaining these ideas is provided in the subsequent sections.

## 3. LITERATURE REVIEW

As research activities gain momentum in the area of PHM, efforts are underway to standardize prognostics research (Uckun, Goebel, & Lucas, 2008). Several studies provide a detailed overview of prognostics along with its distinction from detection and diagnosis (Engel, 2008; Engel, Gilmartin, Bongort, & Hess, 2000). The importance of uncertainty management and the various other challenges in determining remaining useful life are well presented. Understanding the challenges in prognostics research is an important first step in standardizing the evaluation and performance assessment. Thus, we draw on the existing literature and provide an overview of the important concepts in prognostic performance evaluation before defining the new metrics.

### 3.1 Prediction Performance Evaluation Methods

Prediction or forecasting applications are common in medicine, weather, nuclear, finance and economics, automotive, aerospace, and electronics. Metrics based on accuracy and precision with slight variations are most commonly used in all these fields in addition to a few metrics customized to the domain. In medicine and finance, statistical measures are heavily used exploiting the availability of large datasets. Predictions in medicine are evaluated based on hypothesis testing methodologies while in finance errors calculated based on reference prediction models are used for performance evaluation. Both of them use some form of precision and accuracy metrics such as MSE (mean squared error), SD (standard deviation), MAD (mean absolute deviation), MdAD (median absolute deviation), MAPE (mean absolute percentage error) and similar variants. Other domains like aerospace, electronics, and nuclear are relatively immature as far as fielded prognostics applications are concerned. In addition to conventional accuracy and precision measures, a significant focus has been on metrics that assess business merits such as ROI (return on investment), TV (technical value), and life cycle cost, rather than reliability based metrics like MTBF (mean time between failure) or the ratio MTBF/MTBUR (mean time between unit replacements). Notions of false positives, false negatives and ROC (receiver operator characteristics) curves have also been adapted for prognostics (Goebel & Bonissone, 2005).

### 3.2 Summary of the Review

Active research and the quest to find out what constitutes performance evaluation in forecasting related tasks in other domains painted a wider landscape of requirements and domain specific characteristics than initially anticipated. This naturally translated into identifying the similarities and differences in various prediction applications to determine what can or cannot be borrowed from those domains. As shown in Figure 2, a classification tree was generated that listed key characteristics of various forecasting applications and examples of domains that exhibited those (Saxena et al., 2008).

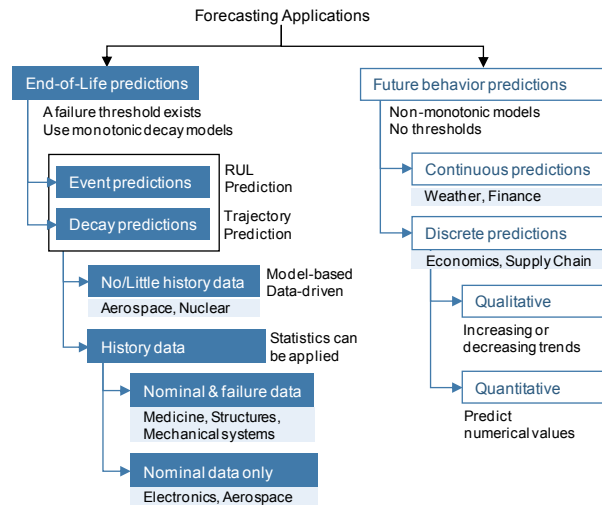


Figure 2: Categories of the forecasting applications (Saxena, et al., 2008).

Coble & Hines (2008) categorized prognostic algorithms into three categories based on type of models/information used for predictions. These types of information about operational and environmental loads are an inherent part of prognostic problems and must be used wherever available. From the survey it was identified that not only did the applications differ in nature, the metrics within domains also varied based on functionality and nature of the end use of performance data. This led to classifying the metrics based on end usage (see Table 1) and their functional characteristics (Figure 3). In a similar effort end users were classified from a health management stakeholder’s point of view (Wheeler, Kurtoglu, & Poll, 2009). Their top-level user groups include Operations, Regulatory, and Engineering. It was observed that it was prognostics algorithm performance that translated into valuable information for these user groups in form or another. For instance, it can be argued that low level algorithmic performance metrics are connected to operational and regulatory branches through a requirement specification process. Therefore, further attention in this effort was focused on algorithmic performance metrics.

Table 1: Classification of prognostic metrics based on end user requirements as adapted from Saxena, et al. (2008) and Wheeler, et al. (2009).

Category	End User	Goals	Metrics
Operations	Program Manager	Assess the economic viability of prognosis technology for specific applications before it can be approved and funded.	Cost-benefit type metrics that translate prognostics performance in terms of tangible and intangible cost savings.
	Plant Manager	Resource allocation and mission planning based on available prognostic information.	Accuracy and precision based metrics that compute RUL estimates for specific Unit Under Test (UUT). Such predictions are based on degradation or damage accumulation models.
	Operator	Take appropriate action and carry out re-planning in the event of contingency during mission.	Accuracy and precision based metrics that compute RUL estimates for specific UUTs. These predictions are based on fault growth models for critical failures.
	Maintainer	Plan maintenance in advance to reduce UUT downtime and maximize availability.	Accuracy and precision based metrics that compute RUL estimates based on damage accumulation models.
Engineering	Designer	Implement the prognostic system within the constraints of user specifications. Improve performance by modifying design.	Reliability based metrics to evaluate a design and identify performance bottlenecks. Computational performance metrics to meet resource constraints.
	Researcher	Develop and Implement robust performance assessment algorithms with desired confidence levels.	Accuracy and Precision based metrics that employ uncertainty management and output probabilistic predictions in presence of uncertain conditions.
Regulatory	Policy Makers	To assess potential hazards (safety, economic, and social) and establish policies to minimize their effects.	Cost-benefit-risk measures, Accuracy and Precision based RUL measures to establish guidelines & timelines for phasing out of aging fleet and/or resource allocation for future projects.

There are different types of outputs from various prognostic algorithms. Some algorithms assess Health Index (HI) or Probability of Failure (PoF) at any given point and others carry out an assessment of RUL based on a predetermined Failure Threshold (FT) (Coble & Hines, 2008; Orsagh, Roemer, Savage, & McClintic, 2001; Saxena, et al., 2008). The ability to generate representations of uncertainty for predictions such as probability distributions, fuzzy membership functions, possibility distribution, etc., further distinguishes some algorithms from others that generate only point estimates of the predictions. This led to the conclusion that a formal prognostic framework must be devised and additional performance metrics needed to be

developed to accommodate most of these scenarios in an intuitive way.

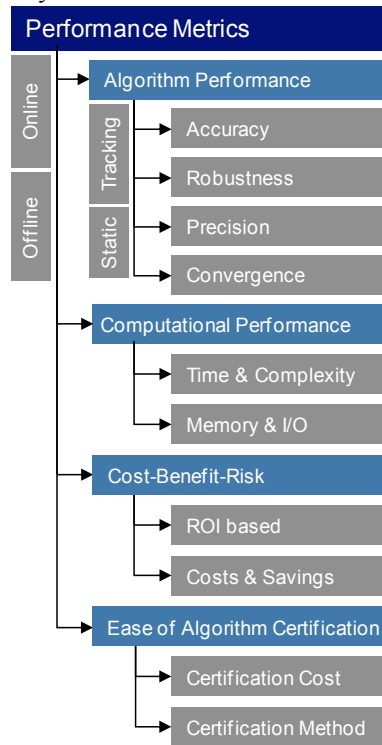


Figure 3: Functional classification of prognostics metrics (adapted from Saxena, et al. (2008)).

For further details on these classifications and examples of different applications the reader is referred to Saxena, et al. (2008).

### 3.3 Recent Developments in the PHM Domain

To update the survey conducted in Saxena, et al. (2008) relevant developments were tracked during the last two years. A significant push has been directed towards developing metrics that measure economic viability of prognostics. In Leao, et al. (2008) authors suggested a variety of metrics for prognostics based on commonly used diagnostic metrics. Metrics like false positives and negatives, prognostics effectiveness, coverage, ROC curve, etc. were suggested with slight modifications to their original definitions. Attention was more focused on integrating these metrics into user requirements and cost-benefit analysis. A simple tool is introduced in Drummond & Yang (2008) to evaluate a prognostic algorithm by estimating the cost savings expected from its deployment. By accounting for variable repair costs and changing failure probabilities this tool is useful for demonstrating the cost savings that prognostics can yield at the operational levels. A commercial tool to calculate the Return on Investment (ROI) for prognostics for electronics systems was developed (Feldman, Sandborn, & Jazouli, 2008). The ‘returns’

that are considered could be the cost savings, profit, or cost avoidance by the use of prognostics in a system. Wheeler, et al. (2009) compiled a comprehensive set of user requirements and mapped them to performance metrics separately for diagnostics and prognostics.

For algorithm performance assessment, Wang & Lee (2009) proposed simple metrics adapted from the classification discipline and also suggested a new metric called “Algorithm Performance Profile” that tracks the performance of an algorithm using a accuracy score each time a prediction is generated. In Yang & Letourneau (2007), authors presented two new metrics for prognostics. They defined a reward function for predicting the correct time-to-failure that also took into account prediction and fault detection coverage. They also proposed a cost-benefit analysis based metric for prognostics. In some other approaches model based techniques are adopted where discrete event simulations are run and results evaluated based on different degrees of prediction error rates (Carrasco & Cassady, 2006; Pipe, 2008). These approaches are beyond the scope of the current discussion.

## 4. CHALLENGES IN PROGNOSTICS

There are several unsolved issues in prognostics that complicate the performance evaluation task. These complications share partial responsibility for the lack of standardized procedures. A good set of metrics should accommodate all or most of these issues but not necessarily require all of them to have been addressed together in any single application. Enumerating these issues briefly here should help understanding the discussions on metrics development later.

**Acausality:** Prognostics is an *acausal* problem that requires an input from future events, for instance the knowledge about operational conditions and load profiles in order to make more accurate predictions. Similarly, to accurately assess the performance (accuracy or precision) one must know the EoL to compare with the predicted EoL estimates. Where the knowledge about these quantities is rarely and completely available, some estimates can be derived based on past usage history, plan for the mission profile, and predictions for future operating and environmental conditions that are not controllable (e.g., weather conditions). This however, adds uncertainty to the overall process and makes it difficult to judiciously evaluate prognostic performance.

**Run-to-Failure Data from Real Applications:** Another aspect that makes this evaluation further complicated is considered the paradox of prognostics – “Not taking an action on a failure prediction involves the risk of failure and an action (e.g. system maintenance and repair), on the contrary, eliminates all chances of validating the correctness of the prediction

itself?. Therefore, it has been a challenging task to assess long term prognostic results. For instance, consider the following scenario where aircraft engines undergo continuous monitoring for fault conditions and scheduled maintenance for system deterioration. In the PHM context a decision about when to perform the maintenance, if not scheduled, is a rather complex one that should be based on current health condition, next flight duration, expected operational (weather) conditions, availability of spares and a maintenance opportunity, options available for alternate planning, costs, risk absorbing capacity, etc. In this situation one could arguably evaluate a prognostic result against statistical (reliability) data about the RULs from similar systems. However, in practice such data are rarely available because there are typically very few faults that were allowed to go all the way to a failure resulting perhaps in an extremely unavoidable in-flight engine shutdown or an aborted takeoff. Furthermore, once the maintenance operation has been performed two problems arise from the perspective of performance evaluation. One, there is no way to verify whether the failure prediction was indeed correct, and two, the useful life of the system has now changed and must have moved the EoL point in time from its previous estimate. Alternatively, allowing the system to fail to evaluate the prognosis would be cost and safety prohibitive.

**Offline Performance Evaluation:** The aforementioned considerations lead to an argument in favor of controlled run-to-failure (RtF) experiments for the algorithm development phase. While this makes it simpler for the offline performance evaluation some issues still remain. First, it is difficult to extend the results of offline setup to a real-time scenario. Second, often in an RtF experiment the setup needs frequent disassemblies to gather ground truth data. This assembly-disassembly process creates variations in the system performance and the EoL point shifts from what it may have been in the beginning of the experiment. Since actual EoL is observed only at the end there is no guarantee that a prediction made based on initial part of data will be very accurate. Whereas, this does not necessarily mean that prognostic algorithm is poorly trained, it is difficult to confirm otherwise. Therefore, one must be careful while interpreting the performance assessment results. Third, even controlled RtF experiments can be very expensive and time consuming, in particular if one seeks to conduct statistically significant number of experiments for all components and fault modes.

There is no simple answer to tackle these issues. However, using reasonable assumptions they can be tackled one step at a time. For instance, most prognostics algorithms make implicit assumptions of

perfect knowledge about the future in a variety of ways such as following:

- operating conditions remain within expected bounds more or less throughout systems life
- any change in these conditions does not affect the life of the system significantly, or
- any controllable change (e.g., operating mode profile) is known (deterministically or probabilistically) and is used as an input to the algorithm

Although these assumptions do not hold true in most real-world situations, the science of prognostics can be advanced and later improved by making adjustments for them as new methodologies develop.

**Uncertainty in Prognostics:** A good prognostics system not only provides accurate and precise estimates for the RUL predictions but also specifies the level of confidence associated with such predictions. Without such information any prognostic estimate is of limited use and cannot be incorporated in mission critical applications (Uckun, et al., 2008). Uncertainties arise from various sources in a PHM system (Coppe, Haftka, Kim, & Yuan, 2009; Hastings & McManus, 2004; Orchard, Kacprzyński, Goebel, Saha, & Vachtsevanos, 2008). Some of these sources include:

- modeling uncertainties (modeling errors in both system model and fault propagation model),
- measurement uncertainties (arise from sensor noise, ability of sensor to detect and disambiguate between various fault modes, loss of information due to data preprocessing, approximations and simplifications),
- operating environment uncertainties,
- future load profile uncertainties (arising from unforeseen future and variability in usage history data),
- input data uncertainties (estimate of initial state of the system, variability in material properties, manufacturing variability), etc.

It is often very difficult to assess the levels and characteristics of uncertainties arising from each of these sources. Further, it is even more difficult to assess how these uncertainties that are introduced at different stages of the prognostic process combine and propagate through the system, which most likely has a complex non-linear dynamics. This problem worsens if the statistical properties do not follow any known parametric distributions allowing analytical solutions.

Owing to all of these challenges Uncertainty Representation and Management (URM) has become an active area of research in the field of PHM. A conscious effort in this direction is clearly evident from recent developments in prognostics (DeNeufville, 2004; Ng & Abramson, 1990; Orchard, et al., 2008; Sankararaman, Ling, Shantz, & Mahadevan, 2009;



Tang, Kacprzyński, Goebel, & Vachtsevanos, 2009). These developments must be adequately supported by suitable methods for performance evaluation that can incorporate various expressions of uncertainties in the prognostic outputs.

Although several approaches for uncertainty representation have been explored by researchers in this area, the most popular approach has been probabilistic representation. A well founded Bayesian framework has led to many analytical approaches that have shown promise (Guan, Liu, Saxena, Celaya, & Goebel, 2009; Orchard, Tang, Goebel, & Vachtsevanos, 2009; Saha & Goebel, 2009). In these cases a prediction is represented by a corresponding Probability Density Function (PDF). When it comes to performance assessment, in many cases a simplifying assumption is made about the form of distribution being Normal or any other known probability distribution. The experience from several applications, however, shows that this is hardly ever the case. Mostly these distributions are non-parametric and are represented by sampled outputs.

This paper presents prognostic performance metrics that incorporate these cases irrespective of their distribution characteristics.

## 5. PROGNOSTIC FRAMEWORK

First, a notational framework is developed to establish relevant context and terminology for further discussions. This section provides a list of terms and definitions that will be used to describe the prognostics problem and related concepts to develop the performance evaluation framework. Similar concepts have been described in the literature. They sometimes use different terms to describe different concepts. This section is intended to resolve ambiguities in interpreting these terms for the purpose of discussions in this paper. It must be noted that in the following discussions  $t_x \in \mathbb{R}^+$  is used to denote time expressed in absolute units e.g., hours, minutes, seconds, etc., and  $x \in I^+$  is a time index to express time in relative units like operating hours, cycles, etc. It follows from the fact that realistic data systems sample from real continuous physical quantities.

Table 2: Frequently used prognostic terms and time indexes to denote important events in a prognostic process.

Prognostic Terms	
UUT	Unit Under Test – an individual system for which prognostics is being developed. Although the same methodology may be applicable for multiple systems in a fleet, life predictions are generated specific to each

	UUT.
PA	Prognostic Algorithm – An algorithm that tracks and predicts the growth of a fault mode with time. PA may be data driven, model-based or a hybrid.
RUL	Remaining Useful Life – amount of time left for which a UUT is usable before some corrective action is required. It can be specified in relative or absolute time units, e.g., load cycles, flight hours, minutes, etc.
FT	Failure Threshold – a limit on damage level beyond which a UUT is not usable. FT does not necessarily indicate complete failure of the system but a conservative estimate beyond which risk of complete failure exceeds tolerance limits.
RtF	Run-to-Failure – refers to a scenario where a system has been allowed to fail and corresponding observation data are collected for later analysis.

### Important Time Index Definitions (Figure 4)

$t_0$	Initial time when health monitoring for a UUT begins.
F	Time index when a fault of interest initiates in the UUT. This is an event that might be unobservable until the fault grows to detectable limits.
D	Time index when a fault is detected by a diagnostic system. It denotes the time instance when a prognostic routine is triggered the first time.
P	Time index when a prognostics routine makes its first prediction. Generally speaking, there is a finite delay before predictions are available once a fault is detected.
EoL	End-of-Life – time instant when a prediction crosses a FT. This is determined through RtF experiments for a specific UUT.
EoP	End-of-Prediction – time index for the last prediction before EoL is reached. this is a conceptual time index that depends on frequency of prediction and assumes predictions are updated until EoL is reached.
EoUP	End-of-Useful Predictions – time index beyond which it is futile to update a RUL prediction because no corrective action is possible in the time available before EoL.

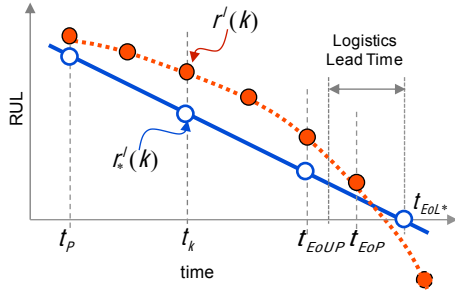


Figure 4: An illustration depicting some important prognostic time indices (definitions and concepts).

### Symbols and Notations

$i$	time index representing time instant $t_i$
$l$	is the index for $l^{\text{th}}$ unit under test (UUT)
$p$	set of all indexes when a prediction is made the first element of $p$ is P and the last is EoP
$t_{\text{EoL}}$	time instant at End-of-Life (EoL)
$t_{\text{EoUP}}$	time for End-of-Useful-Prediction (EoUP)
$t_{\text{repair}}$	time taken by a reparative action for a system
$t_p$	time instant when the first prediction is made
$t_D$	time instant when a fault is detected
$f_n^l(i)$	$n^{\text{th}}$ feature value for the $l^{\text{th}}$ UUT at time $t_i$
$c_m^l(i)$	$m^{\text{th}}$ operational condition for the $l^{\text{th}}$ UUT at $t_i$
$r^l(i)$	predicted RUL for the $l^{\text{th}}$ UUT at time $t_i$ reference to $l$ may be omitted for a single UUT
$r_s^l(i)$	ground truth for RUL at time $t_i$
$\phi^l(i j)$	Prediction for time $t_i$ given data up to time $t_j$ for the $l^{\text{th}}$ UUT. Prediction may be made in any domain, e.g., feature, health, RUL, etc.
$\underline{\Phi}^l(i)$	Trajectory of predictions $\phi^l(i j)$ made for the $l^{\text{th}}$ UUT at time $t_j$ for all times $t_i$ s.t. $i > j$ . E.g., financial and weather forecasts
$h^l(i)$	Health of system for the $l^{\text{th}}$ UUT at time $t_i$
$\alpha$	accuracy modifier such that $\alpha \in [0,1]$
$\alpha^+$	maximum allowable positive error
$\alpha^-$	minimum allowable negative error
$\lambda$	time window modifier s.t. $t_\lambda = t_p + \lambda(t_{\text{EoL}} - t_p)$ where $\lambda \in [0,1]$
$\beta$	minimum desired probability threshold
$\omega$	weight factor for each Gaussian component
$\theta$	parameters of RUL distribution
$\varphi(x)$	non-parameterized probability distribution for any variable $x$
$\varphi_0(x)$	parameterized probability distribution for any variable $x$
$\pi[x]$	probability mass of a distribution of any variable $x$ within $\alpha$ -bounds $[\alpha^-, \alpha^+]$ , i.e. $\pi[x] =$ $\sum_{\alpha^-}^{\alpha^+} \varphi(x); x \in \mathbb{I}^+$ or $\int_{\alpha^-}^{\alpha^+} \varphi_0(x) dx; x \in \mathbb{R}^+$
$M(i)$	a performance metric of interest at time $t_i$

$C_M$  center of mass as a measure of convergence  
for a metric  $M$   
 $x_c, y_c$   $x$  and  $y$  coordinates for center of mass ( $C_M$ )

### Assumptions for the Framework

- Prognostics is condition based health assessment that includes detection of failure precursors from sensor data, prediction of RUL by generating a current state estimate and using expected future operational conditions for a specific system.
- A suitable diagnostic algorithm correctly detects, identifies and isolates the system fault before it triggers a PA to predict evolution for that specific fault mode.
- If the information about future operational conditions is available it may be explicitly used in the predictions. Any prediction, otherwise, implicitly assumes current conditions would remain in the future and/or variations from current operating conditions do not affect the life of a system.
- RUL estimation is a prediction/ forecasting/ extrapolation process.
- Algorithms incorporate uncertainty representation and management methods to produce RUL distributions. Point estimates for RUL may be generated from these distributions through suitable methods when needed.
- RtF data are available that include sensor measurements, operating condition information, and EoL ground truth.
- A definition of failure threshold is available that determines the EoL for a system beyond which the system is not recommended for further use.
- In the absence of true EoL (determined experimentally) statistical (reliability) data such as MTTF (Mean Time to Failure) or MTBF (Mean Time Between Failures) may be used to define EoL with appropriate caution.

In a generic scenario a PA is triggered by an independent diagnostic algorithm whenever it detects a fault in the system with high certainty. PA may take some time to gather more data and tune itself before it starts predicting the growth of that fault. Based on a user defined FT the PA determines where the fault is expected to cross the FT and EoL of the system is reached. An estimate of RUL is generated by computing the difference between estimated EoL and the current time. As time progresses more measurement data become available that are used to make another prediction and the estimates of EoL and RUL are correspondingly updated. This process



continues until one of the following happens:

- the system is taken down for maintenance.
- EoUP is reached and any further predictions may not be useful for failure avoidance operations.
- the system has failed (unexpectedly).
- the case where problem symptoms have disappeared (can occur if there were false alarms, intermittent fault, etc.).

### Definitions

**Time Index:** In a prognostics application time can be discrete or continuous. A time index  $i$  will be used instead of the actual time, e.g.,  $i=10$  means  $t_{10}$ . This takes care of cases where sampling time is not uniform. Furthermore, time indexes are invariant to time-scales.

**Time of Detection of Fault:** Let  $D$  be the time index for time ( $t_D$ ) at which the diagnostic or fault detection algorithm detected the fault. This process will trigger the prognostics algorithm which should start making RUL predictions as soon as enough data has been collected, usually shortly after the fault was detected. For some applications, there may not be an explicit declaration of fault detection, e.g., applications like battery health management, where prognosis is carried out on a decay process. For such applications  $t_D$  can be considered equal to  $t_D$  i.e., prognostics is expected to trigger as soon as enough data has been collected instead of waiting for an explicit diagnostic flag (see Figure 5).

**Time to Start Prediction:** Time indices for times at which a fault is detected ( $t_D$ ) and when the system starts predicting ( $t_P$ ) are differentiated. For certain algorithms  $t_D = t_P$  but in general  $t_P \geq t_D$  as PAs need some time to tune with additional fault progression data before they can start making predictions (Figure 5). Cases where a continuous data collection system is employed even before a fault is detected, sufficient data may already be available to start making predictions and hence  $t_P = t_D$ .

**Prognostics Features:** Let  $f_n^l(i)$  be a feature at time index  $i$ , where  $n = 1, 2, \dots, N$  is the feature index, and  $l = 1, 2, \dots, L$  is the UUT index (an index identifying the different units under test). In prognostics, irrespective of the analysis domain, i.e., time, frequency, wavelet, etc., features take the form of time series and can be physical variables, system parameters or any other quantity that can be computed from observable variables of the system to provide or aid prognosis. Features can be also referred to as a  $1 \times N$  feature vector  $F^l(i)$  of the  $l^{\text{th}}$  UUT at time index  $i$ .

**Operational Conditions:** Let  $c_m^l(i)$  be an operational condition at time index  $i$ , where  $m = 1, 2, \dots, M$  is the condition index, and  $l = 1, 2, \dots, L$  is the UUT index.

Operational conditions describe how the system is being operated and also include the load on the system. The conditions can also be referred to as a  $1 \times M$  vector  $C^l(i)$  of the  $l^{\text{th}}$  UUT at time index  $i$ . The matrix  $C^l$  for all times  $< t_P$  is referred to as load history and for times  $\geq t_P$  as operational (load) profile for the system.

**Health Index:** Let  $h^l(i)$  be a health index at time index  $i$  for UUT  $l = 1, 2, \dots, L$ .  $h$  can be considered a normalized aggregate of health indicators (relevant features) and operational conditions.

**Ground Truth:** Ground truth, denoted by the subscript  $*$ , represents the best belief about the true value of a system variable. In the feature domain  $f_n^l(i)$  may be directly or indirectly calculated from measurements. In the health domain,  $h^l(i)$  is the computed health at time index  $i$  for UUT  $l = 1, 2, \dots, L$  after a run-to-failure test. For an offline study EoL $*$  is the known end-of-life point for the system.

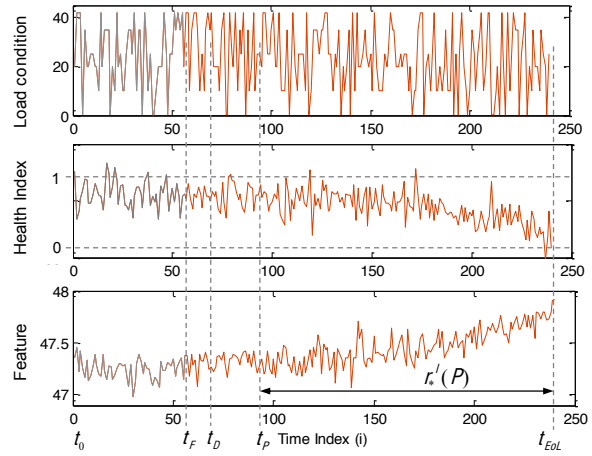


Figure 5: Features and conditions for  $l^{\text{th}}$  UUT (Saxena, et al., 2008).

**History Data:** History data, denoted by the subscript  $\#$ , encapsulates all the a priori information we have about a system. Such information may be of the form of archived measurements or observed EoL data, and can refer to variables in both the feature and health domains represented by  $f_{\#n}^l(i)$  and  $h_{\#}^l(i)$  respectively. For a fleet of systems all reliability estimates such as MTTF or MTBF would be considered history data.

**Point Prediction:** Let  $\phi^l(i|j)$  be a prediction for a variable of interest at a desired point of time  $t_j$  given information up to time  $t_i$ , where  $t_j \leq t_i$  (see Figure 6). Predictions can be made in any domain, features or health. In some cases it is useful to extrapolate features and then aggregate them to compute health and in other cases features are aggregated to a health and then extrapolated to estimate RUL. It must be noted here

that a point prediction may be expressed as probability a distribution or estimated moments derived from the probability distribution.

**Trajectory Prediction:** Let  $\underline{\phi}'(i)$  be a trajectory of predictions formed by point predictions for a variable of interest from time index  $i$  onwards such that  $\underline{\phi}'(i) = \{\phi'(i|i), \phi'(i+1|i), \dots, \phi'(EoP|i)\}$  (see Figure 6). It must be noted that only the last point of this trajectory, i.e.,  $\phi'(EoP|i)$  is used to estimate RUL.

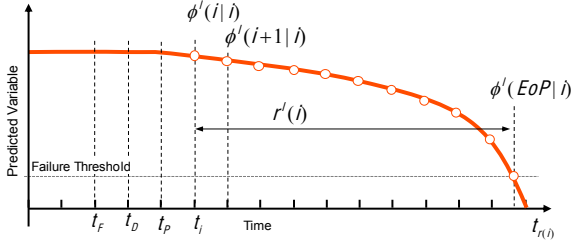


Figure 6: Illustration showing a trajectory prediction. Predictions get updated every time instant.

**RUL:** Let  $r^j(i)$  be the remaining useful life estimate at time index  $i$  given that the information (features and conditions) up to time index  $i$  and an expected operational profile for the future are available. RUL is computed as the difference between the predicted time of failure (where health index approaches zero) and the current time  $t_i$ . RUL is estimated as

$$r^j(i) = t_j - t_i, \text{ where } j = \max_z \{h^j(z) \geq 0, z > i\}. \quad (1)$$

Corresponding ground truth is computed as

$$r_*^j(i) = t_j - t_i, \text{ where } j = \max_z \{h_*^j(z) \geq 0, z > i\}. \quad (2)$$

**RUL vs. Time Plot:** RUL values are plotted against time to compare with RUL ground truth (represented by a straight line). As illustrated in Figure 7, this visually summarizes prediction performance as it evolves through time. This plot is the foundation of prognostic metrics developed in subsequent sections.

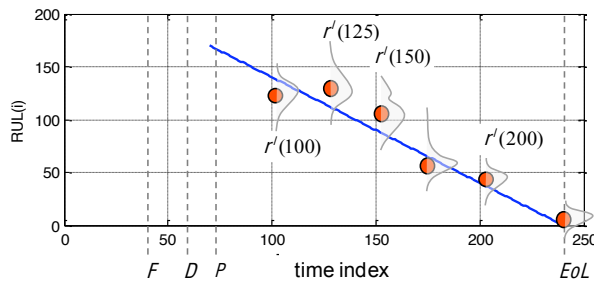


Figure 7: Comparing RUL predictions from ground truth ( $p = \{P | P \in [70, 240]\}$ ,  $t_{EoL} = 240$ ,  $t_{EoP} > 240$ ) (Saxena, et al., 2008).

## 5.1 Incorporating Uncertainty Estimates

As discussed in section 4, prognostics is meaningless unless the uncertainties in the predictions are accounted for. PAs can handle these uncertainties in various ways such as propagating through time the prior probabilities of uncertain inputs and estimating posteriori distributions of EoL and RUL quantities (Orchard & Vachtsevanos, 2009). Therefore, the metrics should be designed such that they can make use of these distributions while assessing the performance. The first step in doing so is to define a reasonable point estimate from these distributions such that no interesting features get ignored in decision making. Computationally the simplest, and hence most widely used, practice has been to compute mean and variance estimates of these distributions (Goebel, Saha, & Saxena, 2008). In reality these distributions are rarely smooth or symmetric, thereby resulting in large errors due to such simplifying assumptions especially while carrying out performance assessment. It is, therefore, suggested that other estimates of central tendency (location) and variance (spread) be used instead of mean and standard deviation, which are appropriate only for Normal cases. For situations where normality of the distribution cannot be established, it is preferable to rely on median as a measure of location and the quartiles or Inter Quartile Range (IQR) as a measure of spread (Hoaglin, Mosteller, & Tukey, 1983). Various types of distributions are categorized into four categories and corresponding methods to compute more appropriate location and spread measures are suggested in Table 3. For the purpose of plotting and visualizing the data use of error bars and box-plots is suggested (Figure 8); more explanation is given in the following sections.

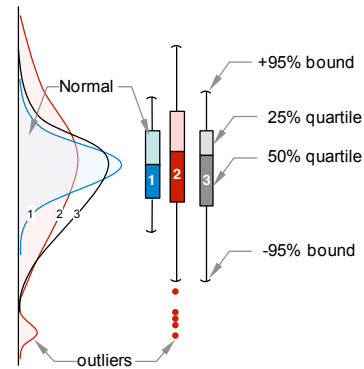


Figure 8: Visual representation for distributions. Distributions shown on the left can be represented by box plots as shown on the right (Saxena, et al., 2009b).

While mean and variance estimates are good for easy understanding they can be less robust when deviations from assumed distribution category are

random and frequent. Furthermore, given the fact that there will be uncertainty in any prediction one must make provisions to account for these deviations. One common way to do so is to specify an allowable error bound around the point of interest and one could use the total probability of failure within that error bound instead of basing a decision on a single point estimate. As shown in Figure 9, this error bound may be asymmetric especially in the case of prognostics, since it is often argued that an early prediction is preferred over a late prediction.

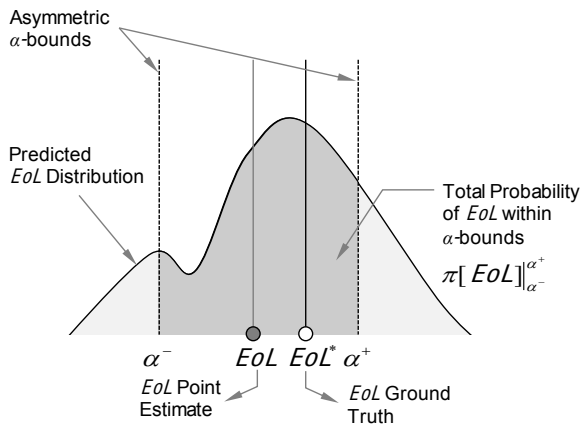


Figure 9: Concepts for incorporating uncertainties.

These ideas can be analytically incorporated into the numerical aspect of the metrics by computing the probability mass of a prediction falling within the specified  $\alpha$ -bounds. As illustrated in the figure, the EoL ground truth may be very different than the estimated EoL and hence the decisions based on probability mass are expectedly more robust. Computing the probability mass requires integrating the probability distribution between the  $\alpha$ -bounds (Figure 10).

The cases where analytical form of the distribution is available, like for Normal distributions, this probability mass can be computed analytically by integrating the area under the prediction PDF between the  $\alpha$ -bounds ( $\alpha^-$  to  $\alpha^+$ ). 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 this probability (see Figure 10). A formal way to include this probability mass into the analytical framework is by introducing a  $\beta$ -criterion, where a prediction is considered inside  $\alpha$ -bounds only if the probability mass of the corresponding distribution within the  $\alpha$ -bounds is more than a predetermined threshold  $\beta$ . This parameter is also linked to the issues of uncertainty management and risk absorbing capacity of the system.

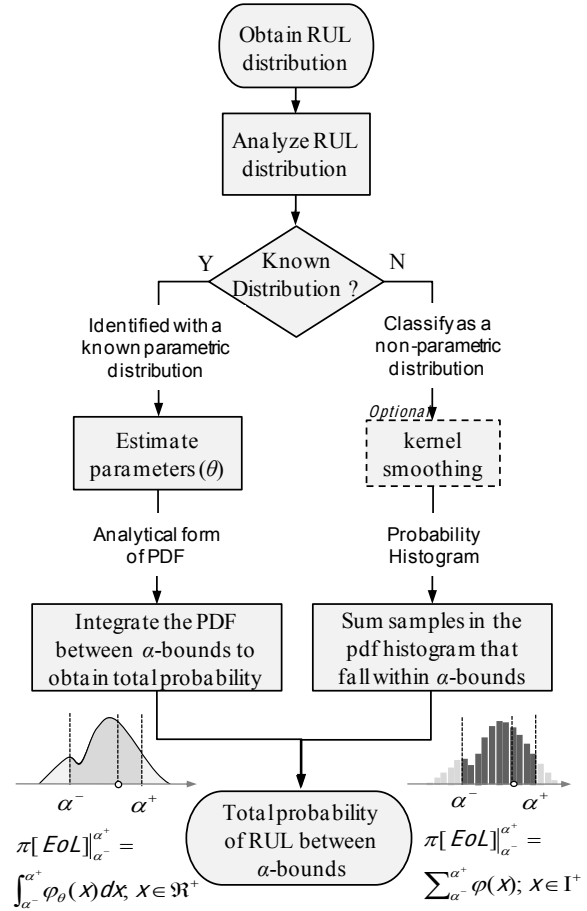


Figure 10: Procedure to compute probability mass of RULs falling within specified  $\alpha$ -bounds.

The categorization shown in Table 3 determines the method of computing the probability of RULs falling between  $\alpha$ -bounds, i.e., area integration or discrete summation, as well as how to represent it visually. For cases that involve a Normal distribution, using a confidence interval represented by a confidence bar around the point prediction is sufficient (Devore, 2004). For situations with non-Normal single mode distributions this can be done with an inter-quartile plot represented by a box plot (Martinez, 2004). Box plots convey how a prediction distribution is skewed and whether this skew should be considered while computing a metric. A box plot also has provisions to represent outliers, which may be useful to keep track of in risk sensitive situations. It is suggested to use box plots superimposed with a dot representing the mean of the distribution. This will allow keeping the visual information in perspective with respect to the conventional plots. For the mixture of Gaussians case, it is recommended that a model with few (preferably  $n \leq 4$ ) Gaussian modes is created and corresponding confidence bars plotted adjacent to each other. The

weights for each Gaussian component can then be represented by the thickness of the error bars. It is not recommended to plot multiple box plots since there is no methodical way to differentiate and isolate the samples associated to individual Gaussian components, and compute the quartile ranges separately for each of them. A linear additive model is assumed here for simplicity while computing the mixture of Gaussians.

$$\varphi(x) \equiv \omega_1 \cdot N(\mu_1, \sigma_1) + \dots + \omega_n \cdot N(\mu_n, \sigma_n); n \in I^+ \quad (3)$$

where:

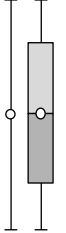
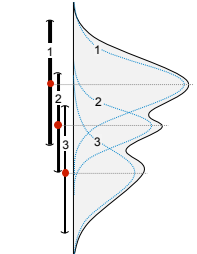


$\varphi(x)$  is a PDF with of multiple Gaussians

$\omega$  is the weight factor for each Gaussian component

$N(\mu, \sigma)$  is a Gaussian distribution with parameters  $\mu$  and  $\sigma$

$n$  is the number of Gaussian modes identified in the distribution.

Table 3: Methodology to select location and spread measures along with visualization methods (Saxena, et al., 2009b).

	Normal Distribution	Mixture of Gaussians	Non-Normal Distribution	Multimodal (non-Normal)
	Parametric		Non-Parametric	
Location (Central tendency)	Mean ( $\mu$ )	Means: $\mu_1, \mu_2, \dots, \mu_n$ weights: $\omega_1, \omega_2, \dots, \omega_n$	Mean, Median, L-estimator, M-estimator	Dominant median, Multiple medians, L-estimator, M-estimator
Spread (variability)	Sample standard deviation ( $\sigma$ ), IQR (inter quartile range)	Sample standard deviations: $\sigma_1, \sigma_2, \dots, \sigma_n$	Mean Absolute Deviation (MAD), Median Absolute Deviation (MdAD), Bootstrap methods, IQR	
Visualization	Confidence Interval (CI), Box plot with mean 	Multiple CIs with varying bar width  Note: here $\omega_1 > \omega_2 > \omega_3$	Box plot with mean 	Box plot with mean 

## 6. PERFORMANCE METRICS

### 6.1 Limitations of Classical Metrics

In Saxena, et al. (2009a) it was reported that the most commonly used metrics in the forecasting applications are accuracy (bias), precision (spread), MSE, and MAPE. Tracking the evolution of these metrics one can see that these metrics were successively developed to incorporate issues not covered by their predecessors. There are more variations and modifications that can be found in literature that measure different aspects of performance. Although these metrics captured important aspects, this paper focuses on enumerating various shortcomings of these metrics from a prognostics viewpoint. Researchers in the PHM community have further adapted these metrics to tackle

these shortcomings in many ways (Saxena, et al., 2008). However, there are some fundamental differences between the performance requirements from general forecasting applications and prognostics applications that did not get adequately addressed. This translates into differences at the design level for the metrics in either case. Some of these differences are discussed here.

These metrics provide a statistical accounting of variations in the distribution of RULs. Whereas this is meaningful information, these metrics are not designed for applications where RULs are continuously updated as more data becomes available. Prognostics prediction performance (e.g., accuracy and precision) tends to be more critical as time passes by and the system nears its end-of-life. Considering EoL as a fixed reference point in time, predictions made at different times create

several conceptual difficulties in computing an aggregate measure using conventional metrics. Predictions made early on have access to less information about the dynamics of fault evolution and are required to predict farther in time. This makes the prediction task more difficult as compared to predicting at a later stage. Each successive prediction utilizes additional data available to it. Therefore, a simple aggregate of performance over multiple predictions made is not a fair representative of overall performance. It may be reasonable to aggregate fixed  $n$ -step ahead (fixed horizon) predictions instead of aggregating EoL predictions (moving horizon). Performance at specific times relative to the EoL can be a reasonable alternative as well. Furthermore, most physical processes describing fault evolution tend to be more or less monotonic in nature. In such cases it becomes easier to learn true parameters of the process as more data become available. Thus, it may be equally important to quantify how well and how quickly an algorithm improves as more data become available.

Following from the previous argument, conventional measures of accuracy and precision tend to account for statistical bias and spread arising from the system. What is missing from the prognostics point of view is a measure that encapsulates the notion of performance improvement with time, since prognostics continuously updates, i.e., successive predictions occur at early stages close to fault detection, middle stages while the fault evolves, and late stages nearing EoL. Depending on application scenarios, criticality of predictions at different stages may be ranked differently. A robust metric should be capable of making an assessment at all stages. This will not only allow ranking various algorithms at different stages but also allow switching prediction models with evolving fault stages instead of using a single prediction algorithm until EoL.

Time scales involved in prognostics applications vary widely (on the order of seconds and minutes for electronic components vs. weeks and years for battery packs). This raises an important question - "how far in advance is enough when predicting with a desired confidence?" Although the earlier the better, a sufficient time to plan and carry out an appropriate corrective action is what is sought. While qualitatively these performance measures remain the same (i.e., accuracy and precision) one needs to incorporate the issues of time criticality.

The new metrics developed and discussed in the following sections attempt to alleviate some of these issues in evaluating prognostic performance.

## 6.2 Prognostic Performance Metrics

In this paper four metrics are discussed that can be used to evaluate prognostic performance while keeping in mind the various issues discussed earlier. These four metrics follow a systematic progression in terms of the information they seek (Figure 11).

The first metric, Prognostic Horizon, identifies whether an algorithm predicts within a specified error margin (specified by the parameter  $\alpha$ , as discussed in the section 5.1) around the actual EoL 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; if not, it may not be meaningful to continue on computing other metrics. If an algorithm passes the PH test, the next metric,  $\alpha$ - $\lambda$  Performance, goes further to identify whether the algorithm performs within desired error margins (specified by the parameter  $\alpha$ ) of the actual RUL at any given time instant (specified by the parameter  $\lambda$ ) that may be of interest to a particular application. This presents a more stringent requirement of staying within a converging cone of the error margin as a system nears EoL. If this criterion is also met, the next step is to quantify the accuracy levels relative to the actual RUL. This is accomplished by the metrics Relative Accuracy and Cumulative Relative Accuracy. These metrics assume that prognostic 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 if it does satisfy all previous metrics. These metrics can be considered as a hierarchical test that provides several levels of comparison among different algorithms in addition to the specific information these metrics individually provide regarding algorithm performance.

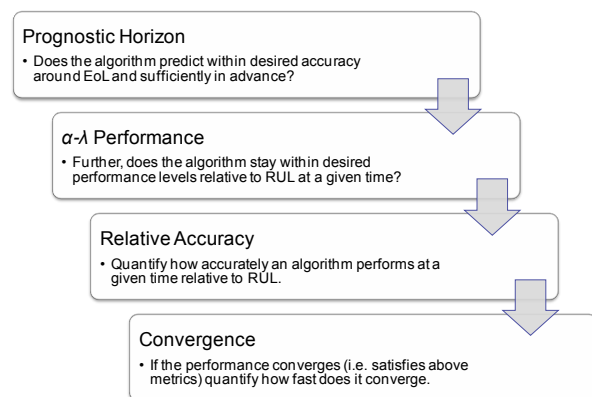


Figure 11: Hierarchical design of the prognostics metrics.



It must be noted that these metrics share the attribute of performance tracking with time unlike the classical metrics. Discussion on detailed definitions and descriptions of these metrics follows henceforth.

**Prognostic Horizon:** Prognostic Horizon (PH) 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 EoL. The performance requirement may be specified in terms of an allowable error bound ( $\alpha$ ) around the true EoL. The choice of  $\alpha$  depends on the estimate of time required to take a corrective action. Depending on the situation this corrective action may correspond to performing maintenance (manufacturing plants) or bringing the system to a safe operating mode (operations in a combat zone).

$$PH = t_{EoL} - t_{i_{\alpha\beta}} \quad (4)$$

where:

$i_{\alpha\beta} = \min \{ j | (j \in p) \wedge (\pi[r(j)]_{-\alpha}^{+\alpha}) \geq \beta \}$  is the first time index when predictions satisfy  $\beta$ -criterion for a given  $\alpha$

$p$  is the set of all time indexes when predictions are made

$i$  is the index for  $i^{\text{th}}$  unit under test (UUT)

$\beta$  is the minimum acceptable probability mass

$r(j)$  is the predicted RUL distribution at time  $t_j$

$t_{EoL}$  is the predicted End-of-Life

$\pi[r(j)]_{-\alpha}^{+\alpha}$  is the probability mass of the prediction PDF within the  $\alpha$ -bounds that are given by  $\alpha^+ = r_* + \alpha \cdot t_{EoL}$  and  $\alpha^- = r_* - \alpha \cdot t_{EoL}$

As shown in Figure 12, the desired level of accuracy with respect to the EoL ground truth is specified as  $\pm\alpha$ -bounds (shaded band). RUL distributions are then plotted against time for all the algorithms that are to be compared. In simple cases the evaluation may be based on point estimates (mean, median, etc.) of the distributions. The PH for an algorithm is declared as soon the corresponding prediction enters the band of desired accuracy. As is evident from the illustration in Figure 12(a), the second algorithm (A2) has a longer PH. However, looking closely at the plots, A1 does not perform much worse than A2, but this method, being less robust due to use of only a point estimate, results in very different PH values for the two algorithms. This can be improved by using the  $\beta$ -criterion, as shown in Figure 12(b).

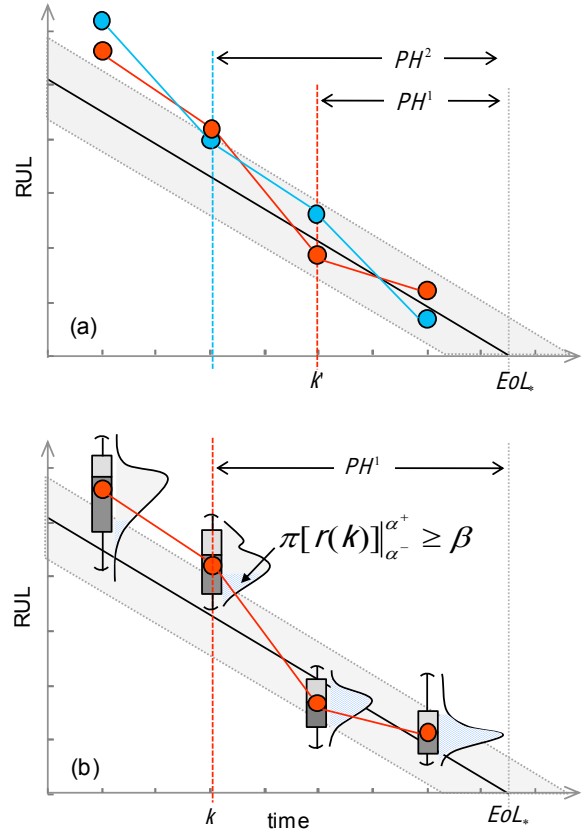


Figure 12: (a) Illustration of Prognostics Horizon while comparing two algorithms based on point estimates (distribution means) (b) PH based on  $\beta$ -criterion results in a more robust metric.

Prognostic 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  $\max\{0, t_{EoL} - t_{EoP}\}$ . The best score for PH is obtained when an algorithm always predicts within desired accuracy zone and the worst score when it never predicts within the accuracy zone. 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 a longer prognostic horizon results in more time available to act based on a prediction that has some desired credibility. Therefore, when comparing algorithms, an algorithm with longer prediction horizon would be preferred.

**$\alpha$ - $\lambda$  Performance:** This metric quantifies prediction quality by determining whether the prediction falls within specified limits at particular times with respect to a performance measure. These time instances may be specified as percentage of total ailing life of the system. The discussion henceforth is presented in the context of



accuracy as a performance measure, hence  $\alpha$ - $\lambda$  accuracy, but any performance measure of interest may fit in this framework.

$\alpha$ - $\lambda$  accuracy is defined as a binary metric that evaluates whether the prediction accuracy at specific time instance  $t_\lambda$  falls within specified  $\alpha$ -bounds (Figure 13). Here  $t_\lambda$  is a fraction of time between  $t_p$  and the actual  $t_{EoL}$ . The  $\alpha$ -bounds here are expressed as a percentage of actual RUL  $r(i_\lambda)$  at  $t_\lambda$ .

$$\alpha - \lambda \text{ Accuracy} = \begin{cases} 1 & \text{if } \pi[r(i_\lambda)]_{-\alpha}^{+\alpha} \geq \beta \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

where:

$\lambda$  is the time window modifier such that  $t_\lambda = t_p + \lambda(t_{EoL} - t_p)$

$\beta$  is the minimum acceptable probability for  $\beta$ -criterion

$r(i_\lambda)$  is the predicted RUL at time index  $i_\lambda$

$\pi[r(i_\lambda)]_{-\alpha}^{+\alpha}$  is the probability mass of the prediction PDF within the  $\alpha$ -bounds that are given by  $\alpha^+ = r_*(i_\lambda) + \alpha \cdot r(i_\lambda)$  and  $\alpha^- = r_*(i_\lambda) - \alpha \cdot r(i_\lambda)$

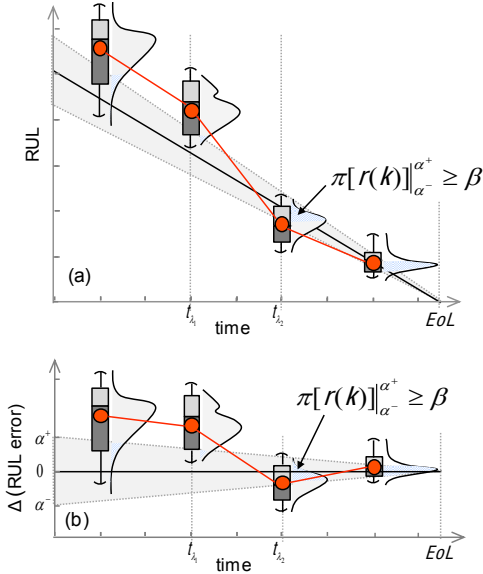


Figure 13: (a)  $\alpha$ - $\lambda$  accuracy with the accuracy cone shrinking with time on RUL vs. time plot. (b) Alternate representation of  $\alpha$ - $\lambda$  accuracy on RUL-error vs. time plot.

As an example, this metric would determine whether a prediction falls within 10% accuracy ( $\alpha = 0.1$ ) of the true RUL halfway to failure from the time the first prediction is made ( $\lambda = 0.5$ ). The output of this metric is binary (1=Yes or 0=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 as shown in Figure 13(a). For easier interpretability  $\alpha$ - $\lambda$  accuracy can also be plotted as shown in Figure 13(b). It must be noted that the set of all time indexes ( $p$ ) where a prediction is made is determined by the frequency of prediction step in a PA. Therefore, it is possible that for a given  $\lambda$  there is no prediction assessed at time  $t_\lambda$  if the corresponding  $i_\lambda \notin p$ . In such cases one can make alternative arrangements such as choosing another  $\lambda'$  closest to  $\lambda$  such that  $i_{\lambda'} \in p$ .

**Relative Accuracy:** Relative Accuracy (RA) is defined as a measure of error in RUL prediction relative to the actual RUL  $r_*(i_\lambda)$  at a specific time index  $i_\lambda$ .

$$RA'_{i_\lambda} = 1 - \frac{|r'_*(i_\lambda) - \langle r'(i_\lambda) \rangle|}{r'_*(i_\lambda)} \quad (6)$$

where:

$\lambda$  is the time window modifier such that  $t_\lambda = t_p + \lambda(t_{EoL} - t_p)$ ,

$i$  is the index for  $i^{\text{th}}$  unit under test (UUT),

$r_*(i_\lambda)$  is the ground truth RUL at time index  $i_\lambda$ ,

$\langle r(i_\lambda) \rangle$  is an appropriate central tendency point estimate of the predicted RUL distribution at time index  $i_\lambda$ .

This is a notion similar to  $\alpha$ - $\lambda$  accuracy where, instead of finding out whether the predictions fall within a given accuracy level at a given time instant, accuracy is measured quantitatively (see Figure 14). First a suitable central tendency point estimate is obtained from the prediction probability distribution using guidelines provided in Table 3 and then using Eq.6.

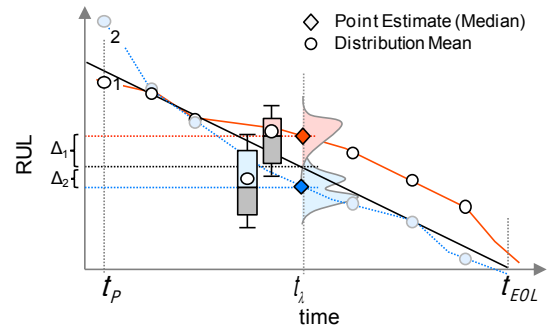


Figure 14: Schematic illustrating Relative Accuracy.

RA may be computed at a desired time  $t_\lambda$ . For cases with mixture of Gaussians a weighted aggregate of the means of individual modes can be used as the point estimate; where the weighting function is the same as the one for the various Gaussian components in the distribution. 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 results in a negative value. Large errors like these, if interpreted in terms of  $\alpha$  parameter for previous metrics, would correspond to values greater than 1. Cases like these need not be considered as it is expected that, under reasonable assumptions, preferred  $\alpha$  values will be less than 1 for PH and  $\alpha$ - $\lambda$  accuracy metrics and that these cases would not have met those criteria anyway.

RA conveys information at a specific time. It can be evaluated at multiple time instances before  $t_\lambda$  to account for general behavior of the algorithm over time. To aggregate these accuracy levels, Cumulative Relative Accuracy (CRA) can be defined as a normalized weighted sum of relative accuracies at specific time instances.

$$CRA'_\lambda = \frac{1}{|\rho_\lambda|} \sum_{i \in \rho_\lambda} w(r'(i)) RA'_\lambda \quad (7)$$

where:

$w(r'(i))$  is a weight factor as a function of RUL at all time indices

$\rho_\lambda$  is the set of all time indexes before  $t_\lambda$  when a prediction is made

$|\rho_\lambda|$  is the cardinality of the set

In most cases it is desirable to weigh those relative accuracies higher that are closer to  $t_{EoL}$ . In general, it is expected that  $t_\lambda$  is chosen such that it holds some physical significance such as a time index that provides a 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  $\lambda$  in order to get an idea on how well a particular algorithm does at significant times. Choice of  $t_\lambda$  should also take into account the uncertainty levels that an algorithm entails by making sure that the distribution spread at  $t_\lambda$  does not cross over expected  $t_{EoL}$  by significant margins especially for critical applications. In other words the probability mass of the RUL

distribution at  $t_\lambda$  extending beyond EoL should not be too large.

**Convergence:** Convergence is a meta-metric defined to quantify the rate at which any metric (M) like accuracy or precision improves with time. It is defined as the distance between the origin and the centroid of the area under the curve for a metric is a measure of convergence rate.

$$C_M = \sqrt{(x_c - t_p)^2 + y_c^2}, \quad (8)$$

where:

$C_M$  is the Euclidean distance between the center of mass  $(x_c, y_c)$  and  $(t_p, 0)$

$M(i)$  is a non-negative prediction accuracy or precision metric with a time varying value

$(x_c, y_c)$  is the center of mass of the area under the curve  $M(i)$  between  $t_p$  and  $t_{EoUP}$ , defined as following

$$x_c = \frac{\frac{1}{2} \sum_{i=P}^{EoUP} (t_{i+1}^2 - t_i^2) M(i)}{\sum_{i=P}^{EoUP} (t_{i+1} - t_i) M(i)}, \text{ and} \quad (9)$$

$$y_c = \frac{\frac{1}{2} \sum_{i=P}^{EoUP} (t_{i+1} - t_i) M(i)^2}{\sum_{i=P}^{EoUP} (t_{i+1} - t_i) M(i)}.$$

As suggested earlier, this discussion assumes that the algorithm performance improves with time. This is easily established if it has passed criteria for previous metrics. For illustration of the concept in Figure 15 three cases are shown that converge at different rates. 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.

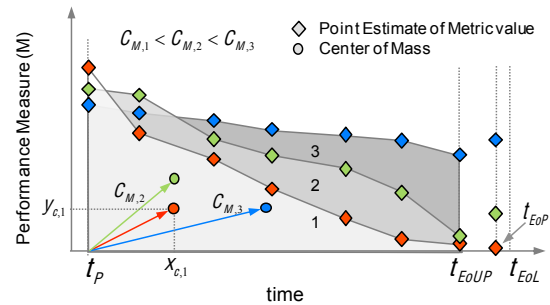


Figure 15: Convergence compares the rates at which different algorithms improve.

### 6.2.1 Applying the Prognostics Metrics

In practice, there can be several situations where the definitions discussed above result in ambiguity. In Saxena, et al. (2009a) several such situations have been discussed in detail with corresponding suggested resolutions. For the sake of completeness such situations are very briefly discussed here.

With regards to PH metric, the most common situation encountered is when the RUL trajectory jumps out of the  $\pm\alpha$  accuracy bounds temporarily. Situations like this result in multiple time indexes where RUL trajectory enters the accuracy zone to satisfy the metric criteria. A simple and conservative approach to deal with this situation is to declare a PH at the latest time instant the predictions enter accuracy zone. Another option is to use the original PH definition and further evaluate other metrics to determine whether the algorithm satisfies all other requirements. Situations like these can occur due to a variety of reasons.

- *Inadequate system model:* Real systems often exhibit inherent transients at different stages during their life cycles. 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. In such cases, one must step back and refine the respective models to incorporate such dynamics.
- *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 models a stochastic process and hence the behavior observed from a particular run (single realization of the stochastic process) may not exhibit the true nature of prediction trajectories. Assuming that all possible measures for uncertainty reduction have been taken during algorithm development, such observations should be treated as isolated realization of the process. In that case these trajectories should be aggregated from multiple runs to achieve statistical significance or more sophisticated stochastic analyses can be carried out.

Plotting the RUL trajectory in the PH plot provides insights for such deficiencies to algorithm developers. It is important to identify the correct reason before computing a metric and interpreting its result. Ideally, an algorithm and a system model should be robust to transients inherent to the system behavior and operational conditions.

The situations discussed above are more common towards the end when a system nears EoL. This is because in most cases the fault evolution dynamics are too fast and complex to model or learn from data as the system nears EoL. Therefore, RUL curve deviates from the error band near  $t_{EoL}$ . To determine whether such deviations are critical for post-prognostic decision making, the concept of  $t_{EoUP}$  or End-of-Useful-Predictions (EoUP) is introduced. This index represents the minimum allowable PH that is required to take a corrective measure. Any predictions made beyond EoUP are of little or no use from a practical viewpoint.

### 6.2.2 Choosing Performance Parameters

From a top-down perspective, the main idea behind these metrics is to help management generate appropriate specifications and requirements for prognostics algorithm in fielded applications. The outcome of these metrics depends directly on the values chosen for input parameters like  $\alpha$ ,  $\lambda$ , and  $\beta$ . Thus, the choice of values for these parameters forms an important aspect of performance evaluation and interpretation. Cost-benefit-risk analyses are generally performed through various methods that model the effects of a variety of constraints (financial costs, safety, criticality of mission completion, reputation, etc.) and derive a range of available slacks in achieving an overall benefit situation (pareto optimal solutions). It is expected that the parameters can be incorporated in these analyses to include the effects of prognostic performance on the cost-benefit of PHM. While this subject is out of the scope of this paper a brief discussion is provided for an overall perspective on how these parameters can be connected to safety, logistics and cost constraints.

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 but no false negatives. In other cases the cost of acting on false positives may be prohibitively high. There are cases where it is more cost effective to tolerate 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 these parameters while carrying out performance evaluation. Some of the most important factors are:

- *Time for problem mitigation:* the amount of time 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 are 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. This factor will have a direct consequence on  $\lambda$  parameter.

- *Cost of mitigation*: cost of the reparative action is an important factor in all management related decisions and hence should be considered. From a decision making point of view this can be associated to the cost due to false positives. This factor influences  $\alpha$ , where there is often a tradeoff between false positives and true positive rates.
- *Criticality of system or cost of failure*: This quantifies the effect of false negatives. Further, while comparing time-critical scenarios, resources should be directed towards more critical and important components in order to efficiently maintain overall health of the system. Likewise, if the health assessment is being performed on multiple units in a system, the parameters for different units should be chosen based on a prioritized list of criticality. Assessment of criticality is usually done based on severity and frequency of occurrence statistics available from Failure Modes, Effects, and Criticality Analysis (FMECA) studies (MIL-STD-1629A, 1980). Another perspective to assess criticality is based on cost-benefit analysis where cost of failures is incorporated to assess the implications of false negatives (Banks & Merenich, 2007; Feldman, et al., 2008).
- *Uncertainty management capability*: Level of confidence on the uncertainty management capability and costs of system failure determine the risk absorbing capacity in a particular scenario. The choice of  $\beta$  is guided by such factors.

Note that these factors mentioned here are not arranged based on any order of importance; users should consider them based on the characteristics of their systems and may skip a few as appropriate.

## 7. FUTURE WORK

A natural extension of this work leads into the development of online prognostic performance metrics. This would require investigations into several issues that were set aside through various assumptions in the present work. For instance, thus far performance evaluation ignores the effect of future loading conditions that alter the rate of remaining life consumption. Performance evaluation without an explicit knowledge about EoL is a challenge for online metrics. These metrics will also need to include provisions for the effects of scheduled maintenance and

self-healing characteristics in some systems. Further, the concepts presented in this paper will be refined and applied to a variety of applications. Developing more metrics like robustness and sensitivity, etc. also remains on the research agenda. Finally, a formal framework for connecting these metrics to top level requirements through development of uncertainty management and representation (URM) methods, incorporation of risk analysis, cost-benefit analysis, and requirements flow down remains a topic of interest in future work.

## 8. CONCLUSION

This paper presents several performance metrics for offline evaluation of prognostics algorithms. A brief overview of different methods employed for performance evaluation is also included. It has been shown that various forecasting related applications differ from prognostics in the systems health management context. This called for developing specialized metrics for prognostics. These metrics were developed keeping in mind various critical aspects that must be included in performance evaluation. A formal prognostic framework was presented to clearly define the concepts and introduce the terminology. Metrics with uncertainty representation capabilities were developed that track the performance of an algorithm with time. Along with detailed discussions and illustrations, it has been shown that these metrics can be successfully applied to evaluate prognostic performance in a standardized manner. Furthermore, it has been discussed that the suggested metrics can be employed to reflect high level requirements in a practical PHM system.

## ACKNOWLEDGMENT

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. This work was funded by NASA Aviation Safety Program-IVHM Project.

## REFERENCES

- Banks, J., & Merenich, J. (2007). *Cost benefit analysis for asset health management technology*. Reliability and Maintainability Symposium (RAMS), Orlando, FL.
- Carrasco, M., & Cassady, C. R. (2006, 23-26 Jan. 2006). *A study of the impact of prognostic errors on system performance*. Annual Reliability and Maintainability Symposium, RAMS06.
- Coble, J. B., & Hines, J. W. (2008). *Prognostic Algorithm Categorization with PHM Challenge*

- Application*. 1st International Conference on Prognostics and Health Management (PHM08), Denver, CO.
- Coppe, A., Haftka, R. T., Kim, N., & Yuan, F. (2009). *Reducing Uncertainty in Damage Growth Properties by Structural Health Monitoring*. Annual Conference of the Prognostics and Health Management Society (PHM09) San Diego, CA.
- DeNeufville, R. (2004). *Uncertainty Management for Engineering Systems Planning and Design*. Engineering Systems Symposium MIT, Cambridge, MA.
- Devore, J. L. (2004). *Probability and Statistics for Engineering and the Sciences* (6th ed.): Thomson.
- Drummond, C., & Yang, C. (2008). *Reverse Engineering Costs: How Much will a Prognostic Algorithm Save?* International Conference on Prognostics and Health Management, Denver, CO.
- Engel, S. J. (2008, October 6-9, 2008). *Prognosis Requirements and V&V: Panel Discussion on PHM Capabilities: Verification, Validation, and Certification Issues*. International Conference on Prognostics and Health Management (PHM08), Denver, CO.
- Engel, S. J., Gilmartin, B. J., Bongort, K., & Hess, A. (2000). *Prognostics, the Real Issues Involved with Predicting Life Remaining*. IEEE Aerospace Conference, Big Sky, MT.
- Feldman, K., Sandborn, P., & Jazouli, T. (2008). *The Analysis of Return on Investment for PHM Applied to Electronic Systems*. International Conference on Prognostics and Health Management (PHM08), Denver, CO.
- Goebel, K., & Bonissone, P. (2005). *Prognostic Information Fusion for Constant Load Systems*. 7th Annual Conference on Information Fusion.
- Goebel, K., Saha, B., & Saxena, A. (2008). *A Comparison of Three Data-Driven Techniques for Prognostics*. 62nd Meeting of the Society For Machinery Failure Prevention Technology (MFPT), Virginia Beach, VA.
- Guan, X., Liu, Y., Saxena, A., Celaya, J., & Goebel, K. (2009). *Entropy-Based Probabilistic Fatigue Damage Prognosis and Algorithmic Performance Comparison*. Annual Conference of the Prognostics and Health Management Society (PHM09), San Diego, CA.
- Hastings, D., & McManus, H. (2004). *A Framework for Understanding Uncertainty and its Mitigation and Exploitation in Complex Systems*. Engineering Systems Symposium MIT, Cambridge MA.
- Hoaglin, D. C., Mosteller, F., & Tukey, J. W. (Eds.). (1983). *Understanding Robust and Exploratory Data Analysis*: John Wiley & Sons.
- Condition Monitoring and Diagnostics of Machines - Prognostics part 1: General Guidelines, ISO/IEC Directives Part 2 C.F.R. (2004).
- 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*. International Conference on Prognostics and Health Management (PHM08), Denver CO.
- Martinez, A. R. (2004). *Exploratory Data Analysis with MATLAB*. In A. R. Martinez (Ed.): CRC Press.
- MIL-STD-1629A. (1980). *Military Standard: Procedures for Performing A Failure Mode, Effects and Criticality Analysis*. Washington DC: Department of Defense.
- NASA. (2009). *NASA Aviation Safety Program* Retrieved December 2009, from [http://www.aeronautics.nasa.gov/programs\\_avsafe.htm](http://www.aeronautics.nasa.gov/programs_avsafe.htm)
- Ng, K.-C., & Abramson, B. (1990). *Uncertainty Management in Expert Systems*. *IEEE Expert Systems*, 5, 20.
- Orchard, M., Kacprzyński, G., Goebel, K., Saha, B., & Vachtsevanos, G. (2008). *Advances in Uncertainty Representation and Management for Particle Filtering Applied to Prognostics*. International Conference on Prognostics and Health Management (PHM08), Denver, CO.
- Orchard, M. E., Tang, L., Goebel, K., & Vachtsevanos, G. (2009). *A Novel RSPF Approach to Prediction of High-Risk, Low-Probability Failure Events*. Annual Conference of the Prognostics and Health Management Society (PHM09), San Diego, CA.
- Orchard, M. E., & Vachtsevanos, G. J. (2009). *A Particle-Filtering Approach for On-line Fault Diagnosis and Failure Prognosis*. *Transactions of the Institute of Measurement and Control*, 31(3-4), 221-246.
- Orsagh, R. F., Roemer, M. J., Savage, C. J., & McClintic, K. (2001). *Development of Effectiveness and Performance Metrics for Mechanical Diagnostic Techniques*. 55th Meeting of the Society for Machinery Failure Prevention Technology, Virginia Beach, VA.
- Pipe, K. (2008). *Practical Prognostics for Condition Based Maintenance*. International Conference on Prognostics and Health Management (PHM08), Denver, CO.
- Saha, B., & Goebel, K. (2009). *Modeling Li-ion Battery Capacity Depletion in a Particle Filtering Framework*. Annual Conference of the Prognostics and Health Management Society (PHM09), San Diego, CA.
- Sankararaman, S., Ling, Y., Shantz, C., & Mahadevan, S. (2009). *Uncertainty Quantification in Fatigue Damage Prognosis*. Annual Conference of the

Prognostics and Health Management Society (PHM09), San Diego, CA.

- Saxena, A., Celaya, J., Balaban, E., Goebel, K., Saha, B., Saha, S., et al. (2008). *Metrics for Evaluating Performance of Prognostics Techniques*. 1st International Conference on Prognostics and Health Management (PHM08), Denver, CO.
- Saxena, A., Celaya, J., Saha, B., Saha, S., & Goebel, K. (2009a). *Evaluating Algorithmic Performance Metrics Tailored for Prognostics*. IEEE Aerospace Conference, Big Sky, MT.
- Saxena, A., Celaya, J., Saha, B., Saha, S., & Goebel, K. (2009b). *On Applying the Prognostics Performance Metrics*. Annual Conference of the Prognostics and Health Management Society (PHM09) San Diego, CA.
- Schwabacher, M. (2005). *A Survey of Data Driven Prognostics*. AIAA Infotech@Aerospace Conference, Arlington, VA.
- Schwabacher, M., & Goebel, K. (2007, 11/09/2007). *A Survey of Artificial Intelligence for Prognostics*. AAAI Fall Symposium, Arlington, VA.
- Tang, L., Kacprzyński, G. J., Goebel, K., & Vachtsevanos, G. (2009). *Methodologies for Uncertainty Management in Prognostics*. IEEE Aerospace Conference, Big Sky, MT.
- Uckun, S., Goebel, K., & Lucas, P. J. F. (2008). *Standardizing Research Methods for Prognostics*. International Conference on Prognostics and Health Management (PHM08), Denver, CO.
- Wang, T., & Lee, J. (2009). *On Performance Evaluation of Prognostics Algorithms*. Machinery Failure Prevention Technology, Dayton, OH.
- Wheeler, K. R., Kurtoglu, T., & Poll, S. (2009). *A Survey of Health Management User Objectives Related to Diagnostic and Prognostic Metrics*. ASME 2009 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference (IDETC/CIE), San Diego, CA.
- Yang, C., & Letourneau, S. (2007). *Model Evaluation for Prognostics: Estimating Cost Saving for the End Users*. Sixth International Conference on Machine Learning and Applications.

**Abhinav Saxena** is a Research Scientist with SGT Inc. at the Prognostics Center of Excellence NASA Ames Research Center, Moffett Field CA. His research focuses on developing and evaluating prognostic algorithms for engineering systems. He received 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 AIAA.

**Jose R. Celaya** is a staff scientist with SGT Inc. 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 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. in Electrical and Computer Engineering from University of Maryland, College Park in 2007. She is currently a Research scientist 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.