

# A Framework for Evaluating Analytic Hyperparameters

Shashvat Prakash<sup>1</sup>, Antoni Brzoska<sup>1</sup>, Sanket Amin<sup>1</sup>

<sup>1</sup>Raytheon Technologies - Collins Aerospace, Windsor Locks, CT, 06096, USA  
shashvat.prakash@collins.com, anotoni.brzoska@collins.com, sanket.amin@collins.com

## ABSTRACT

Prognostic models, when feasible, are favored for avoiding unexpected maintenance. There is a need for a common language when discussing prognostic performance and behavior. The approach presented here considers model behavior in terms of two optimizable sub-problems for better performance assessment. The first evaluation construct considers how well the model tracks degradation over time and a second construct considers how effectively it improves operations. The right set of cost functions can determine the suitability to both objectives. The combined construct enables evaluation of a class of models which augment degradation physics with data-driven heuristics, supporting a more explainable recommendation.

## 1. INTRODUCTION

Unanticipated component failures are often expensive. Considerable costs arise from the loss of operation time, emergency repairs, excess labor, and compensation to aggrieved customers. Understandably, prognostic health management methods are favored when feasible. This requires that failure onset is observable, infrastructure exists to process the data timely, and the mitigation plan actionable.

Prognostic analytic models are essentially degradation models. The model may estimate a remaining useful life (RUL) or, more generally, generate an alert for failure proximity. Recently, more data have become available and cloud computing capabilities continue to grow. There is therefore a growing potential for large scale model deployment. Currently, regulatory bodies are considering prognostic monitoring as a valid substitute for certain routine maintenance inspections (International Maintenance Review Board Policy Board, 2018). Under review is a strategy to use prognostics to increase inspection intervals and apply airworthiness credits for condition based maintenance (Le, Ghoshal & Cuevas, 2011). There is therefore a need to formalize methods which best evaluate the performance of prognostic models.

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

A representative prognostic for condition-based maintenance is shown in Figure 1. A suite of signals is continually providing information about the component health. It falls on the system to decipher the data and develop a decision architecture that drives a discrete maintenance.

Advances in sensors, electronics, and data have enabled an evolution from more traditional reliability-based inspection and maintenance intervals (e.g. Weibull) to some data-augmented hybrid. Generally, more information on failure

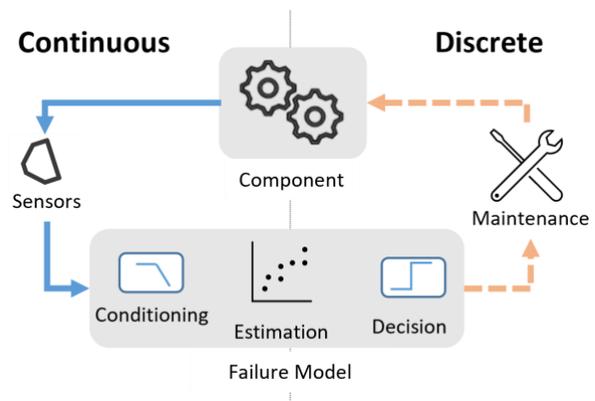


Figure 1. Prognostic Analytic Diagram

modes and usage profiles can drive a more efficient maintenance paradigm. However, with the high availability of data and computation, these approaches can incorporate more data-driven heuristics. Further on the spectrum are the purely data-driven approaches, where the outcome is driven by statistical inference (Luo J. H., Namburu M., Pattipati K., Liu Q., Kawamoto M., and Chigusa S., 2003, Fornlof V., Galar D., Syberfeldt A. and Almgren T., 2016). While this allows for decisions outside the realm of human expertise, they are more difficult to productionize since the outcomes are less explainable.

Combining a physics model with a data-driven one was termed as a 'physics – data hybrid' by Sprong, J. P., Jiang, X., & Polinder, H. (2019). Considering the expense involved in preventative maintenance, it makes sense to maintain the

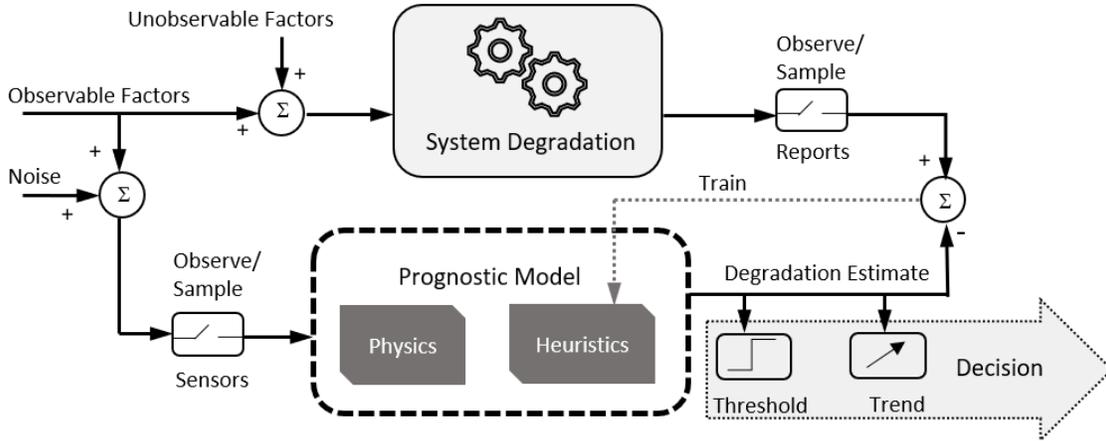


Figure 2. Prognostic model environment and structure with physics and heuristic components

failure mode models grounded in physics but in a way that is enhanced by additional data.

Model based and expert systems use, respectively, a model and a set of rules to infer when the degradation level requires attention (Zhang, Li & Yu, 2006). Models may be trained on a set of failure modes each with its own signature and associated probabilities in a Hidden Markov Model (HMM). Kwan, Zhang, Xu and Haynes (2003) and Zhang, Xu, Kwan, Liang, Xie, and Haynes (2005) developed and implemented an approach where principal components of the input signals were mapped to HMM degradation states. Capturing degradation modes in this manner may not always be scalable across multiple components of a complex system. Further, identifying and training against perceived discrete degradation states might not be needed if a simple signature can be processed out from the available data. The anomaly detection methods like HMM and multivariate Gaussian methods work best when there is some level of cleaned data. In other words, we need to first separate, as best as possible, the signal from the noise in order to produce the best detection outcomes.

One hybrid approach is shown schematically in Figure 2. The system inputs and outputs are rarely known to a complete extent. Rather, only the observables inputs are captured at some time interval and those signals are accompanied with noise. Similarly, real time actual output is not traceable, but a sampled version is available. The truncated input and output information can inform a physics model, with some data-based augmentation to compensate for the modeling and sensing deficiencies. The goal is to reduce estimation error while maintaining the intuition.

Since the ultimate outcome is a discrete decision, there is scope for another level of data-driven heuristics to tune the model. This represents a second set of adjustable hyperparameters. The resulting approach captures the governing physical equations while relying on these

hyperparameters to compensate for incomplete or corrupted inputs not only for the degradation estimate, but also for the final decision-making process.

This paper will deconstruct the development of a prognostic model in two steps, as illustrated in Figure 3. First, it will consider the problem of maximizing a degradation signal and minimizing noise, building upon previous work by the authors (Prakash & Brzoska, 2021). This will involve defining a sensitivity objective, which can be maximized to produce an optimal time-series filter, and used as a proxy for

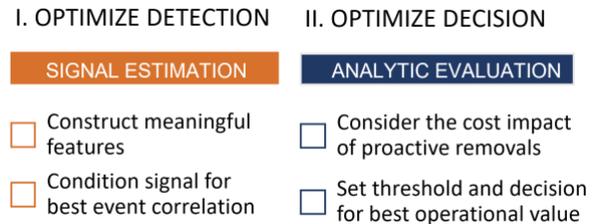


Figure 3. The two sub-problems in developing prognostic models: detection and decision

evaluating signal correlation to events. This is the first optimization objective. Next, the paper will explore the various tradeoffs in implementing a prognostic analytic. Reducing unscheduled events must be balanced with sacrificing useful life. This is the second optimization objective, and also expands previous work by the authors (Prakash, Brzoska, and Ensberg, 2022). Finally, the elements will come together in an implementation framework.

The concepts presented here overlap with previous works, but then go on to develop a few key assertions not previously mentioned. First, the overall evaluation construct is more operationally sensible than not just the binary classifier, but also the common remaining useful life (RUL) error methods. Second, the filter deployed for signal conditioning need not be a linear filter, rather any time series operation can be

evaluated for its ability to create a well correlated signal. Third, there are multiple methods to evaluate the manner in which a signal correlates to an event; more examples are presented in this paper. Fourth, while one of the evaluation methods proposed considers correlation to an RUL estimation, this is fundamentally different from an RUL error. Fifth, the prognostic model approach to avoiding unscheduled failure event costs can be combined with a fixed removal interval to further reduce this cost. Sixth, the potential benefit of a prognostic model can be determined by comparing to the run to failure case. Finally, this paper has a more exhaustive numerical example of the described evaluation construct.

This paper is organized as follows. First, in section 2, we will review the current status quo evaluation, including the binary classifier and RUL evaluation metrics like mean absolute error (MAE), mean absolute percent error (MAPE), and root mean squared error (RMSE). Then, we will propose an alternate method starting with signal sensitivity to the event in section 3. This will constitute the first part of the evaluation construct, examining how well the signal correlates to the event. Then, we will discuss operational impacts that affect the placement of the threshold. Section 4 will discuss the operational value of a fixed threshold while section 5 will delve into the operational impact of the full prognostic model. Taken together, the correlation of the prediction signal to the discrete event and the operational impact of threshold placement, constitute a holistic picture on how to evaluate a prognostic model.

**2. CURRENT METHODS OF EVALUATION**

Common assessments of prognostic models have considered

		Reality		
		No Event	Event	
Predicted	Alert +	False Positive FP	True Positive TP	Precision $\frac{TP}{TP+FP}$
	No Alert -	True Negative TN	False Negative FN	
			Recall $\frac{TP}{TP+FN}$	

Figure 4. Confusion matrix of a binary classifier

the overall goal of maximizing reliability (Fornlof 2016) or the trade-off between part availability versus operational efficiency (Pipe 2008). Such considerations are relevant when the model is already operational and logistic questions remain. This work will consider whether a given model can be tuned to optimize its key performance metrics, and what

those metrics should be. Currently, in the literature and in practice, the commonly accepted evaluation constructs are either the binary classifier or RUL evaluation schemes like MAE and MAEP.

**2.1. Binary Classifiers**

A common assessments of prognostic model performance is the binary classifiers (Fawcett 2006). The capability to detect is quantified with ‘recall’ and the reliability of prediction is similarly quantified with ‘precision’. Recall and precision are based on a confusion matrix, where one axis is detection and the other reality (Figure 4). If both reality and detection agree, the model has scored a ‘true positive’ and if the model alerts without the event it is a ‘false positive’ while a ‘false negative’ means the model failed to adequately detect.

Recall is then the probability of alerting given the event will occur, while precision is the probability of the event occurring given an alert has been announced.

Recall =

$$P(\text{Prognostic Alert} \mid \text{Impending Failure Event}) = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}} \quad (1)$$

Precision =

$$P(\text{Impeding Failure Event} \mid \text{Prognostic Alert}) = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}} \quad (2)$$

The ideal precision or recall values for a given problem can be evaluated by using derived values such as F1 score or receiver operating characteristic (ROC) (Fawcett, 2006). While the former attempts to balance recall and precision, the latter evaluates the tradeoff between the two quantities.

While the above set of parameters present a common metric to evaluate binary classifiers, key elements are lacking for evaluating a prognostic health model. First, the model output and reality are both discretized to, respectively, positive/negative and true/false. Details of the underlying correlation are obscured. This becomes important if we wish to know margin to alert. Second, the time horizon is neglected. The time widow to take action on an alert is limited; alerts without sufficient lead time cannot be acted upon and alerts too early will mean loss of useful life.

**2.2. Remaining Useful Life Evaluation Metrics**

When models are developed for predicting remaining useful life (RUL) specifically, the error between the predicted and actual life remaining are generally evaluated using a suite of three commonly used metrics (Liu & Chen, 2019). These are 1) the mean absolute error (MAE), 2) the mean absolute percent error (MAPE), and 3) the root mean square error

(RMSE) (Liu & Chen, 2019). The equations are defined in  $MAE = \frac{1}{n} \sum_{i=1}^n |y_i - y_i^*|$  (3) –  $RMSE =$

$\sqrt{\frac{\sum_{i=1}^n (y_i - y_i^*)^2}{n}}$  (5), respectively, where  $y_i$  is the true and  $y_i^*$  is the predicted RUL, and  $n$  is the total number of data points.

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - y_i^*| \quad (3)$$

$$MAPE = \frac{1}{n} \sum_{i=1}^n \left| \frac{y_i - y_i^*}{y_i} \right| * 100\% \quad (4)$$

$$RMSE = \sqrt{\frac{\sum_{i=1}^n (y_i - y_i^*)^2}{n}} \quad (5)$$

While the above equations are adequate in evaluating the performance of the prediction across all time intervals, this unnecessarily over-weights the predictions at large RUL values. Indeed, the error tolerance of the prediction is much larger when the component is still healthy and the prediction is likewise not indicating a critical condition. The MAPE algorithm is less affected by this as the evaluation is more sensitive at small values of  $y_i$ , but the sensitivity increases geometrically whereas in practicality, there is an interval around  $RUL=0$  where the prediction is equally critical to overall performance.

The overall problem of a clear evaluation regime can be subdivided into two parts. First, the issue is whether the model is properly capturing the intended degradation leading up to the event. The second part is the fundamental trade-off between too early or unnecessary maintenance against the probability of an unanticipated failure.

### 3. FEATURE SENSITIVITY

A prognostic model is only as effective as its ability to detect a set of given failure modes. This is a correlation problem. Evaluating a model only on the discrete outcomes (like true positives or false positives) misses the nuance of whether the underlying degradation is even well observed.

As depicted in Figure 2, the first problem in estimation is that the available data might not be complete. There are observable as well as unobservable system inputs that contribute to the degradation state. Second, even when the degradation is perfectly observable, the derived features will detect not only the degradation, but other confounding noise. Usually, there are several noise sources, each of which operate with a unique frequency signature.

In the case of aircraft components, often there is a strong seasonality effect, driven by ambient temperature variation. Second, the idiosyncrasies of flight schedules, flight patterns, and daily weather also result in high variability. The goal is then to find the appropriate filter that can best track the real degradation (Figure 5).

### 3.1. Filter Design

A base feature can be derived many ways, usually reflecting some amount of physical modeling. While a feature is sometimes a directly measured attribute like petal length, in PHM applications a feature can itself be an estimated quantity like an effective age or crack length. However, given

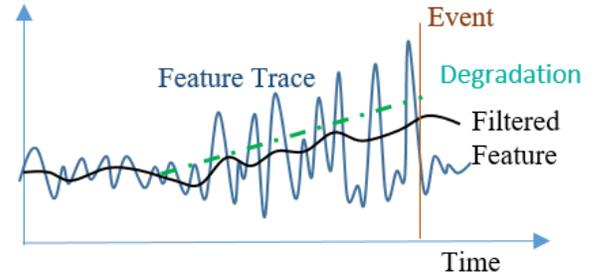


Figure 5. Example noisy degradation feature leading to event

limitations on sensing and observation, the computed feature will propagate these inaccuracies. Therefore, a signal processing step is required to improve the overall estimation.

A dynamic filter modifies the feature in a manner that amplifies certain spectral content and suppresses others. Filters are commonly applied for noise rejection, modeling, estimation, and data fusion. The generic linear discretized filter that produces filtered output  $Y$  from input  $U$  has the form:

$$Y_k = \sum_{i=1}^N a_i * U_{k-i} + \sum_{j=1}^M b_j * Y_{k-j} \quad (6)$$

The filtered output at time instance  $k$  is  $Y_k$  and the output at preceding time samples is  $Y_{k-1}, Y_{k-2}, \dots, Y_{k-M}$ . Similarly, the input at the current  $k$ th instance is  $U_k$  and the preceding values are  $U_{k-1}, U_{k-2}, \dots, U_{k-N}$ . The filter output at the current time instance is therefore a weighted summation of current and previous inputs, and in some cases, previous outputs. The filter coefficients  $a_i$  and  $b_j$  are chosen to achieve a spectral objective: suppression and amplification or a specified frequency range.

The filter described in Eqn. (6) is a specific case of a linear filter. However, the filter structure need not follow this form. In this paper, a filter is simply any operation that acts along the time dimension of the input and prior outputs to produce an output at the current time instance.

In this case, we wish to evaluate the filter and with this evaluation, drive it to an optimal result. In the linear case above, we can assign coefficients  $a_i$  and  $b_j$  if we have the right objective function capturing performance. A description of such a performance evaluation criterion follows.

### 3.2. Lead Time Aggregation

Designing the appropriate filter requires an evaluation construct. Since the main objective is detection, the ideal filter will produce a signal that deviates most from its standard values during time intervals preceding known events and will return to its standard values once events have transpired.

Capturing the filter behavior across all known events requires isolating the filter output for a set lead time interval before each event (Figure 6). Each lead interval data point is averaged on a time or cycle basis with all other lead time traces at the same relative distance from the event.

$$X_i = \sum_{j=1}^N F_{ij} / N \quad (7)$$

The lead interval value  $X$  at the  $i^{\text{th}}$  sample before the event is the average of all trace feature values  $F$  across  $N$  events, at the  $i^{\text{th}}$  value before each  $j$  event. The resulting signal represents typical behavior for the signal ahead of an event.

These aggregated averages  $X$  are then standardized using z-score normalization.

$$Z = (X - E(F)) / STD(F) \quad (8)$$

The normalized value  $Z$  is the aggregated signal value  $X$  subtracted by the original signal mean  $E(F)$  divided by the original signal standard deviation  $STD(F)$ . The z-score normalization has the advantage of allowing comparisons

across all signals with different base units. Further, normalization produces a signal in terms of its standard

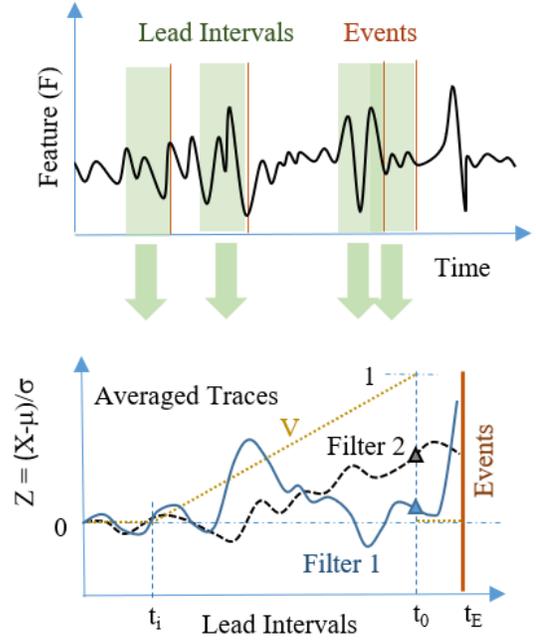


Figure 6. Method of averaging the lead intervals before each event

deviation value so that the larger values, either positive or negative, are more anomalous.

In some cases, there may be events which produce no detectable precursor, as may happen with a false negative. In that case, all filtered outputs  $X$  will be penalized equally, and the event will not play a role in filter selection. Conversely, there may be maxima in a signal that are not associated with an event. These false positives will raise the mean value  $E(F)$  and result in a lower normalized  $Z$  value.

The aggregated and normalized pre-event trace  $Z$  of each filter can be considered at some fixed interval before the event for comparison (Figure 6). In the figure, the normalized and averaged traces of two candidate features are plotted, and the time axis has time of event ( $t_E$ ), minimum lead time to act before the event ( $t_0$ ), and the initial time of the lead interval ( $t_i$ ). Filter 1 in Figure 6 exhibits maxima both well ahead of  $t_0$  as well as in the  $t_0$  to  $t_E$  interval. The filter value between  $t_0$  and  $t_E$  is irrelevant since there isn't enough lead time to mitigate the event. However, too much lead time reduces useful life. Filter 1's behavior is less desirable compared to Filter 2, which has a maximum just before  $t_0$ , providing ample lead time ahead of the anticipated event without sacrificing much useful life.

### 3.3. Evaluation Methods

The Z value at the critical lead interval,  $Z(t_0)$ , can be a useful gauge of the relative performance of a given filter compared to others. This does not require any arbitrary rules or limits, only the process-defined, requisite minimum lead time.

Table 1. Evaluation Methods for Lead Intervals

<i>Method</i>	<i>Formula</i>	<i>Description</i>
Critical Z	$\frac{X(t_0) - E(F)}{STD(F)}$	Aggregated normalized value at min lead time $t_0$
Score S	$\Sigma_i(Z_i V_i) / \Sigma_i(V_i^2)$	Correlation to value function V
Recovery R	$Z(t_0) - Z(t_E + \Delta t)$	Ability of feature to return to nominal
Nominal Stability	$STD(Z(t_i - \Delta t : t_i))$	Variance of nominal signal
Monotonicity	$d/dt Z > 0$	Tendency of feature to increase

In certain cases, the lead trace Z values can be better evaluated with a weighting function V (Figure 5) which rises monotonically from 0 at some  $t_i < t_0$  up to 1 at  $t_0$ , then returning to 0 for the  $t_0$  to  $t_E$  interval. For a given filter, each Z value can be combined with its weight V in a weighted sum:

$$S = \Sigma_i(Z_i V_i) / \Sigma_i(V_i^2) \quad (9)$$

The above equation ascribes a score S to the lead time averaged trace by weighting the  $i$  Z values with the weights  $V_i$ . High Z values near the event but before the critical actionable time ( $t_0$ ) will have high V weights and increase the score while the other Z values will have less bearing on the score.

The V function can be customized to suit the filter objectives. Any monotonically increasing function over  $t_i$  and  $t_0$  will isolate signal components which show greatest anomaly immediately before the critical actionable time ( $t_0$ ). A linearly increasing V value, either over cycles or flight hours, best captures the consumption of useful life on a cost basis. A sigmoid or step would acknowledge that any signal anomaly in the interval has comparable value, and loss of useful life is less important.

Note that correlating with a linear V as a proxy for RUL is fundamentally different from measuring RUL prediction error. In the former case, we are aiming to converge on a degradation estimate that has some increasing trend near the event with enough lead time. In the latter case, there is no consideration for a critical lead time or the beginning of the lead time period, rather the entire history of the RUL estimate is compared to the actual RUL.

### 3.4. Role of Signal Conditioning

This approach to filter tuning offers a more meaningful alternative to the traditional precision, recall, and receiver operating characteristic of binary classifiers as well as a less ambiguous version of the RUL evaluation criteria. Those methods have niche applications which do not translate as well to a continuous signal which can be filtered and correlated.

Table 2. Parameter Summary

<i>Parameter</i>	<i>Definition</i>
$C_u$	Cost of unscheduled action
$C_s$	Cost of scheduled action
$R$	Component operational value
$f$	Time-to-failure probability density
$v$	Proportion of units which fail before replacement
$\mu$	Expected component operational life
$E[f]$	Expected time-to-failure of distribution $f$
$T$	Life at which component is replaced
$J$	Per unit cost of maintenance program
$\tilde{j}$	Maint. program cost per unit life

In this framework, the evaluation methods in Table 1 are both a measure of signal correlation to a discrete event and the mechanism for obtaining the optimal signal. If V is chosen to reflect the value of remaining useful life, the score reflects monetary benefit of the detector.

Conceptually, the filter tuning method is a way to model missing physical elements in a catch-all filter, with the ultimate objective of producing a feature anomaly near failure events but with sufficient lead time. This idea is shown in Figure 2. True degradation is driven by both known and unknown sources. Detection of the degradation is not perfect because confounding factors and sensing limitations respectively introduce noise and limit observability. The resulting signals are arranged into features using the known degradation mechanisms so that the features are physically explainable. The heuristics acknowledge the imperfections in the feature and attempts to compensate for them in order to estimate the degradation level.

## 4. EVALUATION OF MAINTENANCE STRATEGIES

Thus far we have considered the signal estimation problem, informed by physics and operational value, as an approach to improve the overall detection. Ultimately, however, the model must drive a maintenance action. This requires transforming an otherwise continuously varying degradation signal into a discrete decision point. A simple threshold comes with some implementation issues. First, the

underlying estimated degradation signal may not increase monotonically due to the previously explained estimation issues. Second, the placement of the threshold has implications for the performance of the overall model; too conservative and both lead time and recall suffer while in the other extreme precision suffers. Overall, the binary classification system (precision/ recall/ lead time/ receiver operator characteristic) doesn't reduce to a single overall performance metric in a manner that can be interpretable as value to the operator. Rather, operator profitability should be the prime consideration when implementing a prognostic health model framework, and this consideration should be integral to the model implementation.

Failure distributions such as the Weibull are a common approach for reliability modeling (Lei & Sandborn, 2018). However, fitting a distribution model to the failure data has its own challenges. A poor fit can sometimes be attributed to multiple failure modes, each requiring its own model. On occasion the different failure modes are imputed from the aggregated data.

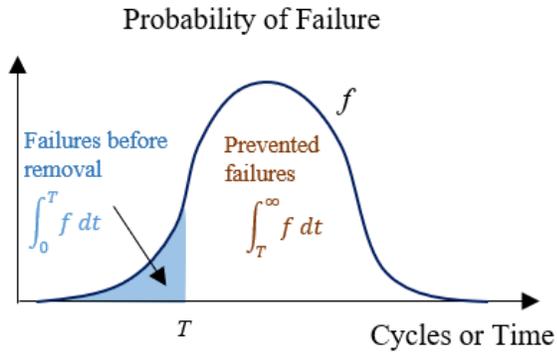


Figure 7. Distribution of part failures

#### 4.1. Objective Cost Models

There have been several cost models developed for specific prognostic maintenance applications. Pattabhiraman, Gogu, Kim, Haftka, and Bes (2012), examined a cost model for structural airframe maintenance, taking into consideration the benefits of prolonging the regular maintenance cycles. In that application, the prognostic model extended the regular repair schedule but provided less advance notice and put certain maintenance schedules out of synch. Lei and Sandborn (2016) similarly developed a cost model for wind turbine maintenance. Key elements of that model were downtime costs and the monetary value of remining useful life of a non-failed component.

Tian et. al. developed a cost model for cases where the failure distribution is known. Their analysis considered the expected cost of repair normalized by expected life of the component.

When the failure probability distribution is available ( $f(t)$ , Figure 7), a service interval  $T$  may be set strategically, without knowing any information about the failure mode or degradation state. This is useful in cases where the part must be replaced quickly in order to limit operational disruption. Diagnosis is a secondary concern. This maintenance approach has been termed 'soft life' as a reference to 'soft reliability' when removals are not prompted by an out-of-spec condition (Geudens, Sonnemans, Petkova, and Brombacher, 2005).

Estimating the ideal replacement interval  $T$  requires developing an understanding of the overall value and costs associated with the strategy. First, we define a set of parameters.  $C_u$  and  $C_s$  denote the costs associated with taking an unscheduled and scheduled action, respectively. Generally,  $C_u \gg C_s$ , and this underscores the benefit of proactive maintenance. Let  $R$  denote the component operational value. The value of  $R$  governs how much value is lost when the functioning component is removed prior to failure. Let  $f(t)$  denote the probability density of the time-to-

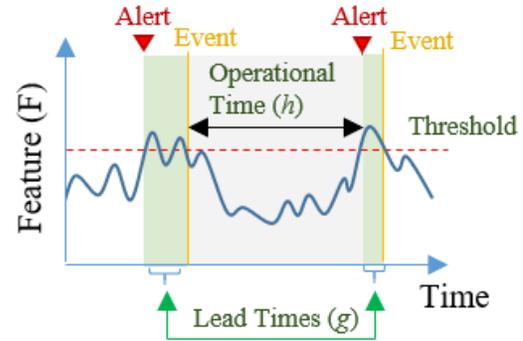


Figure 8. Time series of analytic lead times to failure  $g$  and operational time  $h$

failure of the component (Figure 7). Let  $E[f]$  denote the expected value of  $f(t)$ . The focus of this discussion will be on the effective maintenance cost per unit  $J$  and this cost per unit life  $J' = J/\mu$ . All parameters are summarized in Table 2.

#### 4.2. Run to Failure Cost Model

In the trivial scenario, there is no prognostic maintenance plan, and components are run until failure. The maintenance cost  $J$  is simply the cost of an unscheduled removal  $C_u$ . Normalizing maintenance cost over the expected life  $\mu$  yields the following set of relationships:

$$J = C_u \quad (10)$$

$$\tilde{J} = \frac{J}{\mu} = \frac{C_u}{E(f)} \quad (11)$$

In this case, the expected life until failure  $E(f)$  is equivalent to the expected component life  $\mu$ , which can be expressed in

terms of cycles or operation time. The life measurement units are chosen based on failure mechanism. The maintenance cost per life unit,  $J' = J/\mu$ , captures an overall operational cost objective, as shown in Eqn. ( $\tilde{J} = \frac{J}{\mu} = \frac{C_u}{E(f)}$ ) (11).

### 4.3. Fixed Replacement Interval Cost Model

We can apply insights from the above scenario to the soft life case where all components are removed after a fixed period of time,  $T$ , even if they are still functional. Some fraction of the component population will fail before  $T$ ; this proportion is represented with  $v$  in (12) below.

$$v = \int_0^T f dt = 1 - \int_T^\infty f dt \quad (12)$$

It is possible to use the distribution of the time-to-failure (Figure 7) to determine the expected costs, which are given by Equations (13)-(14) below:

$$J = C_u v + C_s (1 - v) + \frac{R}{\mu} \int_T^\infty (t - T) * f dt \quad (13)$$

$$\mu = \int_0^T t f dt + T * (1 - v) \quad (14)$$

The first and second term in ((13) represent the costs of an unscheduled and scheduled actions multiplied by the probabilities of those actions occurring, respectively. The third term in ((13) represents the residual value of the removed component. Equation ((14) represents the expected

Table 3. Parameter Definitions

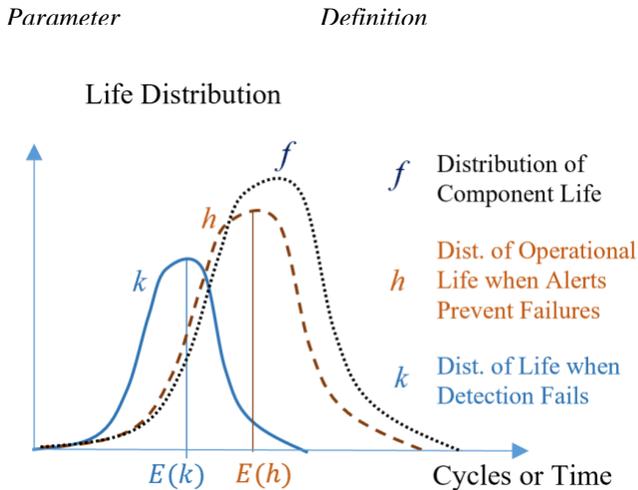


Figure 9. Probability distribution of component operation time for no-analytic case  $f$ , the case when an analytic prevents unscheduled failures  $h$ , and for the case when detection fails  $k$ .

operational life  $\mu$  for a component with a proactive removal program. The first term represents the life of units which fail before the scheduled removal time.

The cost of an unscheduled repair is several times larger than a scheduled repair due to the emergency nature of an operational failure. A larger cost difference favors earlier intervention. However, as  $T$  shrinks,  $R/\mu$  and the integrals in the last two terms grow. An ideal balance (minimum  $J$ ) is achieved when the reduction in unscheduled repairs is not justified by an increase in lost RUL.

### 4.4. Prognostic Analytic Cost Model

A prognostic analytic model aims to remove components before failure based on a precursor alert. The objective is to avoid the costs associated with unanticipated events, however the detection may be imperfect and the removed component is still fully functional. If the analytic alerts, then there is a lead time  $g$  between the alert and the anticipated failure event and an operational life  $h$  between installation and the first alert-driven removal (Figure 8). For this discussion, only the first instance of the alert in the inter-event time period is considered. Although there may be several threshold violations after the first alert, lead and operational times will be measured from the first instance since action is triggered by that indicator.

The framework for the prognostic model builds on the elements of the soft life. First, the expected life of a component must be segregated into two groups. There is a subset where the prognostic analytic detects the degradation and the component is removed before actual failure. These components will have their operational life reduced by the prognostic lead time, and their life distribution is shown as  $h$  in Figure 9. There is also a second group where the detection fails; either the failure mode was not captured, or the detection did not provide enough lead time. The life distribution of this set is shown as  $k$ .

The cost framework is constructed by considering the model configuration and associated costs. Let  $x$  denote the vector of parameters associated with an analytic. For example, this vector will contain the value of the alerting threshold.

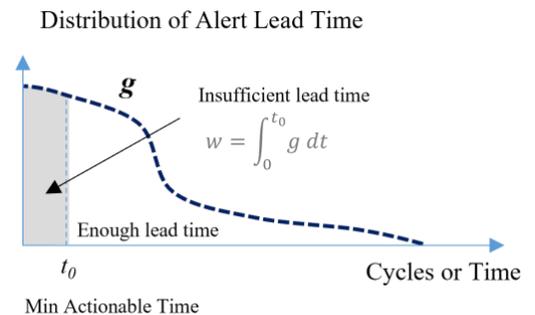


Figure 10. Distribution of analytic lead times to failure  $g$

$$\mu(\mathbf{x}) = p(\mathbf{x})E(h(t, \mathbf{x})) + (1 - p(\mathbf{x}))E(k(t, \mathbf{x})) \quad (15)$$

The probability density of lead times is denoted by  $g(t, \mathbf{x})$  (**Error! Reference source not found.**). The value  $t_0$  denotes the time threshold for acting on an analytic alert. An alert is not actionable if the lead time to failure,  $g(t, \mathbf{x})$  is less than  $t_0$ . These alerts can be represented by  $w(t, \mathbf{x})$  in the equation ((16) below.

$$w(t, \mathbf{x}) = \int_0^{t_0} g(t, \mathbf{x}) dt = 1 - \int_{t_0}^{\infty} g(t, \mathbf{x}) dt \quad (16)$$

Let  $p(\mathbf{x})$  denote the probability of alerting before failure, irrespective of lead time. This quantity is analogous to recall in discrete classifiers. In the trivial case, where  $p(\mathbf{x}) = 0$ ,  $k(\mathbf{x}, t) = f(t)$  from the no-analytic example.

The focus of this discussion will be the cost  $J(\mathbf{x})$  of implementing an analytic with the parameters  $\mathbf{x}$ . Let  $J'(\mathbf{x})$  denote the cost per unit life. Let  $J_a(\mathbf{x})$  denote the cost on the condition that the analytic alerts. All these parameters are summarized in Table 3.

Whether there is an unscheduled or scheduled event depends upon whether the analytic alerted successfully with enough lead time. Thus, it follows that

$$J(\mathbf{x}) = p(\mathbf{x}) * J_a(\mathbf{x}) + (1 - p(\mathbf{x})) * C_u \quad (17)$$

Given the distribution of lead times  $g(t, \mathbf{x})$  (**Error! Reference source not found.**), it is possible to determine the expected cost of acting on an alert. This can be done in a similar fashion as in equation (12).

$$J_a = C_u w + C_s (1 - w) + \frac{R}{\mu} \int_0^{\infty} t * g dt \quad (18)$$

Finally, in order to determine the cost per unit time or unit cycle of implementing the analytic, we normalize the cost per unit by the expected life span

$$\tilde{J}(\mathbf{x}) = \frac{J(\mathbf{x})}{\mu(\mathbf{x})} \quad (19)$$

Equation (19) represents the component expected operational life for an analytic that has probability of alert  $p$ , distribution of time between maintenance and next alert  $h$ , and life distribution in absence of alert  $k$ . As before,  $\tilde{J}(\mathbf{x})$  is a gauge of the per cycle or per time cost of a prognostic analytic. The relative benefit of the analytic is determined when  $(\tilde{J}(\mathbf{x}) = \frac{J(\mathbf{x})}{\mu(\mathbf{x})})$  (19) is compared to  $(\mathcal{J} = \frac{J}{\mu} = \frac{C_u}{E(f)})$  (11).

With this construct, we can look specifically at the unscheduled, scheduled, and RUL components of maintenance cost  $J$  in terms of  $p$ ,  $w$ ,  $R$ , and  $\mu$ .

$$J = C_u(pw + (1 - p)) + C_s p(1 - w) + \frac{R}{\mu} p E(g) \quad (20)$$

$$\mu = p E(h) + (1 - p) E(k) \quad (21)$$

The equations ((20)-(21) are a more intuitive and compact form of (15)-(18). The overall objective is to minimize  $\tilde{J} = J/\mu$ . This means that terms like  $p$ , which determines whether a failure can be detected before onset, must be high, leading to a higher proportion of components in distribution  $h$  instead of  $k$ . This is the same concept as recall. Further, we wish to maximize  $E(h)$ , meaning we want a significant lifespan before the first alert. This is equivalent to precision. Third, we would like a small lead time ( $E(g)$ ) that is greater than  $t_0$  but not too much greater or we will have to contend with the penalty of excess remaining useful life.

#### 4.5. Soft Life and Prognostic Model

Every component in the undetected distribution  $k$  is subject to the high unscheduled maintenance cost. One approach for reducing the  $k$  is to introduce a fixed time interval  $T$  for unit replacement along with the prognostic alerts. This means certain components are removed proactively when the analytic alerts with enough lead time  $t_0$  and the total life is less than  $T$ , others are removed without any prognostic alert at  $T$  because  $T$  either arrived before the alert or the non-alerted failure, and a third category has no prognostic alert and fails before  $T$ .

In this case, the distributions  $h$  and  $k$  are segregated into  $h_v$ , and  $k_v$  for the portions of  $h$  and  $k$  that are less than  $T$ , and  $h_r$ , and  $k_r$  for the portions of  $h$  and  $k$  that are greater than  $T$ .

$$h_v = \int_0^T h dt \quad h_r = \int_T^{\infty} h dt \quad (22, 23)$$

$$k_v = \int_0^T k dt \quad k_r = \int_T^{\infty} k dt \quad (24, 25)$$

The probability of falling into one of these sub-categories, for instance  $h_v$  and  $k_v$ , is as follows:

$$P(h_v) = p(\mathbf{x}) * \frac{h_v}{h_v + h_r} \quad (26)$$

$$P(k_v) = (1 - p(\mathbf{x})) * \frac{k_v}{k_v + k_r} \quad (27)$$

Note that both  $h$  and  $k$  are partial distributions; the entire population is reflected in the sum of  $h$  and  $k$ . We can express the expected life of a component with the combined approach as follows.

$$\mu = P(h_v)E(h_v) + P(k_v)E(k_v) \quad (28)$$

$$+P(h_r)E(h_r) + P(k_r)E(k_r)$$

The lead time distribution  $g$  will now be truncated to  $g'$  with corresponding short lead time segment  $w'$  because it is not possible to have a life span greater than  $T$ . Therefore, there is no scope of alerts with lead time greater than  $T$ .

$$\begin{aligned} J = & C_u(P(h_v)w' + P(k_v)) \\ & + C_s(p(1 - w') + P(k_r)) \\ & + \frac{R}{\mu}(P(h_v)E(g') + P(h_r)(E(f) - E(h_r)) \\ & + P(k_r)(E(f) - E(k_r))) \end{aligned} \quad (29)$$

The terms in equation ((29) are analogous to ((20) in that the three terms reflect the unscheduled, scheduled, and loss of remaining useful life costs.

## 5. IMPLEMENTATION FRAMEWORK

In section 3 we presented a method to amplify the expected degradation element in a noisy signal. The objective was to find a way to filter optimally such that the degradation was evident with only minimal noise. In this manner, we improve the correlation between the feature and the discrete event. In section 4, we considered the relative costs of acting on an analytic alert.

The overall framework is a two-part optimal feature and alert development. The steps are as follows:

1. Model the expected degradation progression with an increasing function (Figure 6)
2. Average the signal traces in the intervals leading up to each event (Figure 6)
3. Optimize a filter (or similar time series model) using an evaluation like score from Table 1 as an objective function
4. Find a threshold and hyperparameter set that minimizes the per unit life cost of the analytic  $\tilde{f}$  from equations ((20)-((21), compare to no-analytic and fixed replacement interval (soft life) cases
5. Consider whether there is an advantage to implementing a fixed replacement interval along with the prognostic analytic

With this framework, the model development has two optimization steps. The first one, in step 3, will reduce noise in the data. Noise is anything in the signal that does not correlate to component degradation. Once noise cancellation methods are applied, any residual noise may be traced to failure cases where there was no detectable degradation (false negatives) or cases where there was a degradation signature without an immediate recorded event (false positive). Given these detection challenges, the filtered signal will produce the best approximation of degradation.

Once the signal is as clean as possible, the second optimization step is for the discrete, decision-making elements (step 4). The signal must be subject to a decision threshold or some other hyperparameter driven heuristics to generate a definitive alert. These must be set in a manner to optimize the costs associated with the analytic: the loss of remaining life and the reduced operational time balanced against the savings from avoiding unscheduled events.

## 6. NUMERICAL EXAMPLE

An example case has been constructed to demonstrate the described method on synthesized data. We will begin by first demonstrating the signal evaluation and processing methods of section 3. We consider a component which has a lifespan distributed normally with mean 1250 cycles and standard deviation 250 cycles. Over the course of 10000 cycles, there are 7 failure events.

A degradation signature is modeled as a linearly increasing signal in the 500 cycles leading up to the event, and zero at all other points. This represents a form of physical process where the degradation is evident only in the final stage of life and progresses at a constant rate until failure. In many practical scenarios, the degradation signal may not always be present before failure. To simulate this case, the degradation element has been removed for failure number 4. This is a false negative example.

A noisy indicator will contain traces of the component degradation and noise from various sources. For this example, the noise is a set of 8 sinusoidal signals with random amplitudes up to .65 and frequencies spanning .01 to .03 Hz. Then, uniform random noise is added with zero mean and amplitude 2.5. The resulting noisy degradation signal is shown in Figure 11. This represents a raw sensor signal.

This raw signal now contains weak signatures prior to each event but one. Where the signatures exist, they are almost indistinguishable against the noise. A plot of lead time traces prior to each of the seven events is shown, mean-normalized, in **Error! Reference source not found.** The individual profiles are the signal value subtracted by the mean and then divided by the standard deviation, in the 500 cycles immediately before each event. Even though Profile 4 has no degradation component, it is similar to all the other profiles which do contain the linear degradation.

Next, an optimal filter was developed in the form of equation (6). Note that we could have also used a nonlinear time series approach like a deep neural net or Gaussian process regression (GPR). The coefficients  $a_i$  and  $b_i$  were chosen to maximize the objective function, which was simply S from Table 4. Running multivariate optimization yielded  $a_0 = -0.8060$ ,  $a_1 = -0.7890$ ,  $b_1 = 1.000$ ,  $b_2 = -0.9295$ ,  $b_3 = 1.1388E-3$ .

The normalized filtered signal traces leading up to each event are shown in **Error! Reference source not found.**, with the resulting time series shown in Figure 14. Now there is a clear

separation between the profiles containing the degradation and profile 4, the one without degradation. This indicates that the filter is amplifying the modeled degradation and suppressing the attendant noise.

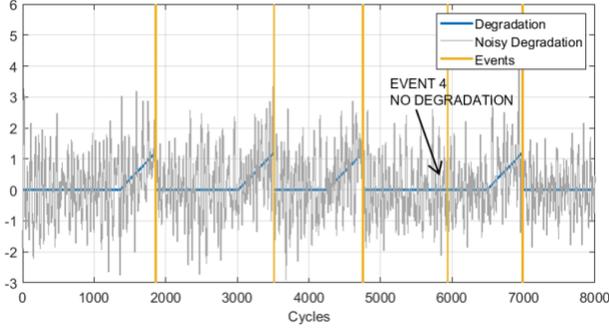


Figure 11. Synthetic data incorporating signal and noise – degradation is omitted for the fourth event

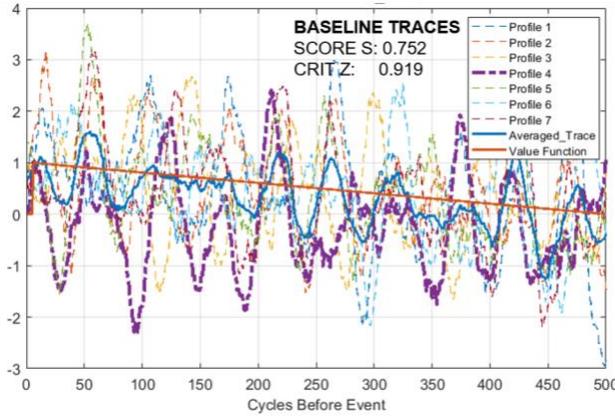


Figure 13. Baseline normalized traces showing behavior before each event, with profile  $n$  signature before event  $n$ .

Note that even with the filter, some residual noise remains. This left over noise is in some part related to the omission of event 4's degradation, in effect pushing the filter to retain some noise in attempt to find the signal.

Table 2 summarizes the performance of the filter across each event and for two of the evaluation metrics from Table 1. Although the filter was optimized for the score metric  $S$ , it also improves on the critical  $Z$  metric for the cases where degradation was present. Event 4 did not have a degradation signal so the metrics decrease after the filter suppresses noise. As shown in Table 2, the overall score value  $S$  improved from .752 to 1.436, and the overall critical  $Z$  also improved from .919 to 1.762.

Next, we will expand our database of failure events and examine the relative tradeoffs between the cost of acting on a prognostic alert against not acting. The cost of a prognostic maintenance strategy is represented by  $\tilde{J}$ , the cost per unit life

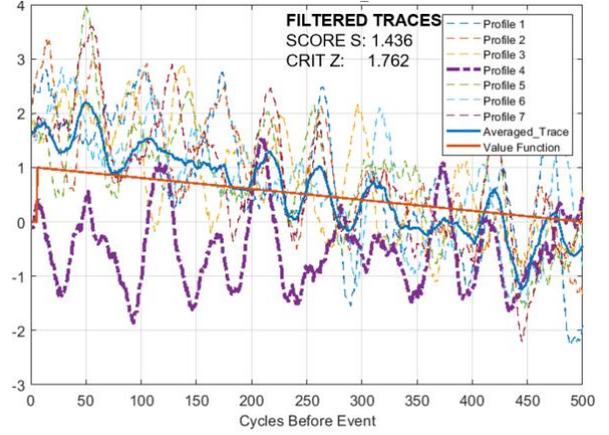


Figure 12. Filtered signal traces with a larger difference between the average and non-degraded fourth profile

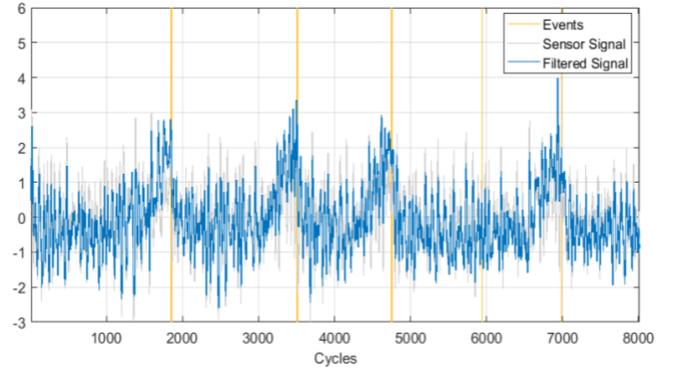


Figure 14. Resulting filtered signal

as represented in equations ((20)-(21). The  $\tilde{J}$  of a run to failure program is represented in equation  $\tilde{J} = \frac{J}{\mu} = \frac{C_u}{E(f)}$

(11). The relative cost savings is the comparison of the two is the ratio of the two expressions.

In this example, we let  $J_{rtf}$  and  $\mu_{rtf}$  represent the run to failure case.

$$\frac{J}{J_{rtf}} = (pw + (1 - p)) + \frac{C_s}{C_u} p(1 - w) + \frac{R}{C_u \mu} pE(g) \quad (30)$$

$$\mu/\mu_{rtf} = pE(h)/E(f) + (1 - p)E(k)/E(f) \quad (31)$$

Equations ((30) and ((31) can be evaluated for the numerical example leaving symbols for the costs. We first institute a threshold of 3.5 and an alerting criterion that generates an alert when the threshold is breached for any 10 cycles in a consecutive 25 cycle window. With these values, the run to failure  $f$ , remove at alert  $h$ , and the missed detection  $k$  distributions are shown in the top chart of Figure 15. The lead time distribution  $g$  is shown in the bottom of Figure 15. There were 1000 total events with alerts preceding 830, so the recall,  $p$ , is 0.83. For simplicity, we have left the precision at 1.0. Accordingly, the ratio of the expected life  $\mu/\mu_{rtf}$  from (29) is 0.8566, so the average life  $\mu=1289.3$  cycles.

If we assume a min lead time requirement ( $t_0$ ) of 115 cycles, the value for  $w$  from Eq. ((16) is 3.91E-3. So the relative advantage of prognostics from Eq. ((28) becomes:

$$\frac{J}{J_{rtf}} = 0.1467 + \frac{C_s}{C_u} 0.8533 + \frac{R}{C_u} 0.1785 \quad (32)$$

Ultimately, we must balance the cost reduction captured in Eqn. ((32) with the loss of life in Eqn. ((31). So, we employ the ratio from Eqn. ( $\tilde{J}(\mathbf{x}) = \frac{J(\mathbf{x})}{\mu(\mathbf{x})}$ ) (19) to provide the complete picture.

$$\frac{\tilde{J}}{\tilde{J}_{rtf}} = \frac{J/J_{rtf}}{\mu/\mu_{rtf}} = 0.1713 + \frac{C_s}{C_u} 0.9961 + \frac{R}{C_u} 0.2084 \quad (33)$$

Equation ((33) captures the overall benefit of the prognostic algorithm, taking into account the factors of performance like lead time to alert,  $E(g)$ , ability to capture the degradation (recall  $p$ ), and how consistently the first alert occurs near the end of life (precision,  $E(h)$ ). In this case, the average lead time  $E(g)$  is relatively small partly since we did not allow for any false positives. Therefore, the main cost component is the ratio between the scheduled and unscheduled maintenance,  $C_s/C_u$ .

Table 4. Evaluation Results from Example

Event	Life	Degraded	Score S		Critical Z	
			Base	Filter	Base	Filter
1	1858	Yes	0.953	1.777	1.169	2.384
2	1656	Yes	1.040	1.946	1.589	2.792
3	1242	Yes	0.895	1.773	0.422	1.519
4	1187	No	-0.311	-0.736	0.988	0.401
5	1045	Yes	0.872	1.790	1.084	1.702
6	1634	Yes	0.866	1.626	0.332	1.677
7	848	Yes	0.951	1.878	0.846	1.858
AVG	1353		0.752	1.436	0.919	1.762

## 7. CONCLUSIONS

The framework presented here is fundamentally a two-step evaluation method for a prognostic analytic. While the analytic may be based on an understanding of physical degradation, unknown effects, confounding factors, and

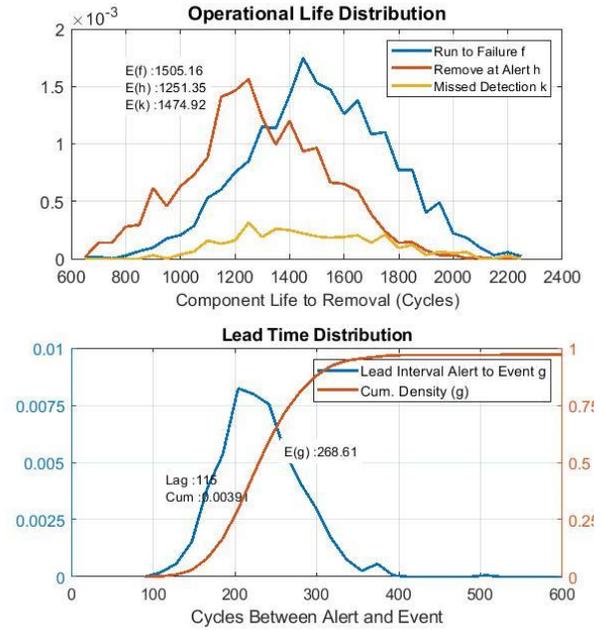


Figure 15. Distributions for (top) run to failure  $f$ , remove at prognostic alert  $h$ , missed detection  $k$ , and (bottom) lead time between alert and failure  $g$

signal limitations will present estimation challenges. The level of signal correlation to an expected degradation progression is one type of evaluation metric.

Second, the decision point of when to act can be separately evaluated. Prognostic alerts are most valuable when the relative benefit of reducing unexpected failures is balanced with the cost of reducing useful life

The discrete classifier evaluation methods (Fawcett, 2006) require arbitrary boundaries between true positives and false positives, and these metrics change with the threshold value. Attempts to disambiguate the threshold, like the receiver operator characteristic (ROC) are not well suited for prognostic applications which use continuous sensor signals. The RUL evaluation approaches are an improvement on the binary classifier, but they unnecessarily consider errors at large RUL (for MAE and RMSE) or the importance of that error grows geometrically (MAPE).

The approaches presented here address those shortcomings by evaluating behavior in the lead intervals ahead of events and considering key behavior patterns like whether the signal is anomalous (critical Z from Table 1) or whether the signal roughly tracks the RUL (score from Table 1).

Once the detection problem is addressed, the decision threshold can be determined based on operational value. The paper has proposed a cost model to represent the overall cost of the prognostic algorithm compared to a baseline run to failure and a fixed interval approach.

These techniques segregate the problem into first, evaluating the signal to event correlation, and second, evaluating the decision thresholds. The nature of communication around prognostics should adopt this continuous-discrete duality mindset and evaluate models at both levels.

## NOMENCLATURE

$C_s$	Cost of scheduled action
$C_u$	Cost of unscheduled action
$E(f)$	Expected time-to-failure of distribution $f$
$F$	Feature trace tracking a measure of degradation
$f$	Time-to-failure probability density
$FN$	False negative
$FP$	False positive
$g$	Distribution of time from first alert to event
$h$	Distribution of life from installation to first alert
$J$	Per unit cost of maintenance program
$J_a$	Per unit cost of maintenance when analytic alerts
$\bar{J}$	Maint. program cost per unit life ( $J/\mu$ )
$k$	Distribution of life when analytic fails to alert
$MAE$	Mean absolute error
$MAPE$	Mean absolute percentage error
$P$	Fraction of failure events detected (Recall)
$R$	Component operational value
$RMSE$	Root mean squared error
$ROC$	Receiver operating characteristic
$RUL$	Remaining useful life
$S$	Score-based correlation of lead interval to event
$STD$	Standard deviation
$T$	Life at which component is replaced
$t$	Unit of life (time or cycles)
$t_0$	Minimum time interval required to avoid event
$t_E$	Time of the event
$t_i$	Time marking the beginning of the lead interval
$TP$	True positive
$TN$	True negative
$X$	Averaged lead interval before events
$Z$	Z-score normalization of the feature

## REFERENCES

- Fawcett T. (2006). An Introduction to ROC Analysis, *Pattern Recognition Letters*. 27 (8): 861–874.
- Fornlof V., Galar D., Syberfeldt A. and Almgren T. (2016). Maintenance, prognostics and diagnostic approaches for aircraft engines, 2016 *IEEE Metrology for Aerospace (MetroAeroSpace)*, pp. 403-407, doi: 10.1109/MetroAeroSpace.2016.7573249
- Geudens, W. H. J. M., Sonnemans, P. J. M., Petkova, V. T., Brombacher, A. C. (2005). Soft Reliability, a new Class of Problems for Innovative Products: “How to Approach Them.” *Annual Reliability and Maintainability Symposium*, Jan 24-77, Alexandria, VA, USA. doi: 10.1109/RAMS.2005.1408391
- International Maintenance Review Board Policy Board, Aircraft Health Monitoring (AHM) Integration in MSG 3, Issue Paper 180. (2018) MSG-3: Operator/Manufacturer scheduled Maintenance Development, Vol. 1- Fixed Wing Aircraft Rev. 2018.1, Air Transport Association of America, <https://www.easa.europa.eu/document-library/imrbpb-issue-papers>.
- Le D.D., Ghoshal A., and Cuevas E. (2011). Condition-based maintenance plus and maintenance credit validation. Army Research Laboratory, Dept. of Defense, Aberdeen Proving Ground, MD 21005. <https://apps.dtic.mil/sti/pdfs/ADA589574.pdf>
- Lei, X., and Sandborn, P. A. (2018). Maintenance Scheduling Based on Remaining Useful Life Predictions for Wind Farms Managed Using Power Purchase Agreements *Renewable Energy*, vol. 116, Part B, pp.188-198 doi: 10.1016/j.renene.2017.03.053
- Lei, X., Sandborn, P. A., Goudarzi, N., & Bruck, M. A. (2015). PHM based predictive maintenance option model for offshore wind farm O&M optimization. In *Annual Conference of the PHM Society (Vol. 7, No. 1)*. doi: 10.1109/ICPHM.2015.7245027
- Liu, J., & Chen, Z. (2019). Remaining useful life prediction of lithium-ion batteries based on health indicator and gaussian process regression model. *IEEE Access*, 7, 39474-39484. doi: 10.1109/ACCESS.2019.2905740
- Luo J. H., Namburu M., Pattipati K., Liu Q., Kawamoto M., Chigusa S., (2003). Model-based prognostic techniques [maintenance applications], *Proc. AUTOTESTCON 2003. IEEE Systems Readiness Technology Conference.*, pp. 330-340, (22-25 Sept.) doi: 10.1109/AUTEST.2003.1243596.
- Mallett, S., Royston, P., Waters, R. Dutton S., and Altman D. (2010). Reporting performance of prognostic models in cancer: A review. *BMC Med* 8, 21. doi: 10.1186/1741-7015-8-21
- Pattabhiraman S., Gogu C., Kim N., Haftka R. T., and Bes C. (2012), Skipping unnecessary maintenance using an on-board structural health monitoring system, *Proc IMechE Part O: J Risk and Reliability* 226(5) 549-560.
- Pipe K. (2008). Practical prognostics for condition based maintenance, *International Conference on Prognostics and Health Management*.
- Prakash, S., & Brzoska, A. (2021). Evaluating and Optimizing Analytic Signals. *Annual Conference of the PHM Society*, Virtual, 13(1). November 11-13, doi: 10.36001/phmconf.2021.v13i1.2968
- Prakash, S., Brzoska A., and Ensberg, J. (2022). A Framework for Evaluating PHM Models. *Proc. IEEE Aerospace Sciences Conference*
- Sprong, J. P., Jiang, X., & Polinder, H. (2019). A Deployment of Prognostics to Optimize Aircraft Maintenance: A Literature Review. *Annual Conference of the PHM Society*, 11(1). doi: 10.36001/phmconf.2019.v11i1.776.
- Tiang Z., Lin, D., and Wu, B. (2012). Condition based maintenance optimization considering multiple

objectives. *J. Intell. Manuf.* 23, 333–340. doi: 10.1007/s10845-009-0358-7

Zhang L., Li X., Yu J. (2006). A review of fault prognostics in condition based maintenance. *Sixth International Symposium on Instrumentation and Control Technology: Signal Analysis, Measurement Theory, Photo-Electronic Technology, and Artificial Intelligence*, (6 November 2006); doi: 10.1117/12.717514

## BIOGRAPHIES



**Shashvat Prakash** received a B.S. in Mechanical Engineering from University of Illinois, Urbana, an M.S. in Mechanical Engineering from Carnegie Mellon University, and a Ph.D. in Mechanical Engineering from the Georgia Institute of Technology. He has worked on satellite attitude and orbit control at NASA Goddard, combustion

and control of aviation turbines at General Electric, and prognostics and health management (PHM) of aircraft components at Raytheon Technologies-Collins Aerospace. Currently a Senior Principal Engineer, he has over 13 years of experience in the commercial aviation industry



**Antoni Brzoska** received B.S., M.S., and Ph.D. degrees in Mathematics from the University of Connecticut. He has presented research on tree topologies and has experience with developing pattern recognition models. He worked at a neuroscience start-up on detecting patterns from brain images. He has been with Collins Aerospace since 2020,

working primarily on prognostic model development for airline customers. He has contributed to papers at previous PHM Society conferences and the IEEE Aerospace Sciences Conference.

**Sanket Amin** received B.S. and M.S. in Mechanical Engineering from The Pennsylvania State University. His 15+ year career is mostly in the area of CBM and PHM and spans from aircraft engine monitoring to various aircraft systems. He has contributed papers in this area related to fault detection of bearing and gear failures, fuzzy logic reasoning, and big data processing for PHM. He has collaborated with various companies and organizations in this area including Impact Technologies, NASA Ames, GE, Pratt and Whitney, NAVAIR, AFRL. He currently leads a group of data scientists and systems engineers in developing predictive analytics for commercial aircraft.