17

Prognostics

Michael J. Roemer[1], Carl S. Byington[1], Gregory J. Kacprzynski[1], George Vachtsevanos[1], and Kai Goebel[2]

[1]*Impact Technologies*

[2]*NASA Ames Research Center*

Overview

Prognostics has received considerable attention recently as an emerging sub-discipline within SHM. Prognosis is here strictly defined as “predicting the time at which a component will no longer perform its intended function.” Loss of function is oftentimes the time at which a component fails. The predicted time to that point then becomes the remaining useful life (RUL). For prognostics to be effective, it must be performed well before deviations from normal performance propagate to a critical effect. This enables a failure preclusion or prevention function to repair or replace the offending components, or, if the components cannot be repaired, to retire the system (or vehicle) before the critical failure occurs. Therefore, prognosis has the promise to provide critical information to system operators that will enable safer operation and more cost-efficient use. To that end, the US Department of Defense (DoD), NASA, and industry have been investigating this technology for use in their vehicle health management solutions. Dedicated prognostic algorithms (in conjunction with failure detection and fault isolation algorithms) must be developed that are capable of operating in an autonomous and real-time vehicle health management system software architecture that is possibly distributed in nature. This envisioned prognostic and health management system will be realized in a vehicle-level reasoner that must have visibility and insight into the results of local diagnostic and prognostic technologies implemented at the line replaceable unit (LRU) and subsystem levels. Accomplishing this effectively requires an integrated suite of prognostic technologies that compute failure effect propagation through diverse subsystems and that can capture interactions that occur in these subsystems. In this chapter a generic set of selected prognostic algorithm approaches is presented and an overview of the required vehicle-level reasoning architecture needed to integrate the prognostic information across systems is provided.

System Health Management: with Aerospace Applications, First Edition. Edited by Stephen B. Johnson, Thomas J. Gormley, Charles Mott, Seth S. Kessler, Ann Patterson-Hine, Karl M. Reichard, and Philip A. Scandura, Jr.

17.1 Background

Various health monitoring technologies were developed in the 1990s for aerospace applications that aid in the detection and classification of developing system failures (Marko *et al*.,1996; Schauz, 1996, Shiroishi *et al*.,1997). However, these technologies traditionally focused on failure detection and fault isolation within an individual subsystem or system. Health management system developers have since begun to address the concepts of prognostics and the integration of anomaly[1] detection technologies, diagnostic technologies, and prognostic technologies across subsystems and systems. Prognostics can be performed when the damage (irrespective of whether it is the result of an expected or unexpected anomaly) has not yet reached its maximum threshold and when there is sufficient time to estimate remaining life. The ability to first detect impending failures and isolate their causes[2] and then predict their future progression based on the current diagnostic state and available future operating data is receiving considerable attention at NASA, the DoD, and industry.

One critical element of any prognostic system is the assessment of the prediction uncertainties, which is required to allow the conversion of remaining life estimates into actionable decisions. If a remaining life estimate has very large uncertainty bounds, that information may in the extreme case not be useful if one is already within the bounds of the risk cut-off. In that case, the action would have to be taken immediately and prognostics provides no benefit. The metrics used to calculate the performance of prognostics include accuracy, precision, and robustness (Saxena *et al*., 2008). Increasing the performance of the prediction on a LRU/subsystem's health is therefore often implemented using various algorithmic techniques and information fusion concepts that can optimally combine sensor data, empirical and physics-based models, and historical information. By utilizing a combination of health monitoring data and model-based techniques, a comprehensive prognostic capability can be achieved throughout a component's or LRU's life. For example, model-based estimates can be used initially when no diagnostic indicators are present that at later stages might be supplemented with estimates based on monitored features when failure indications are detectable.

Finally, these technologies must be capable of communicating the root cause of a problem across subsystems and propagating the up/downstream effects across the health management system architecture. This chapter will discuss some generic prognostic system algorithmic approaches that have been demonstrated within various aircraft subsystem components with the ability to predict the time to failure (on a real-time basis). Prognostic systems that can effectively implement the capabilities presented herein offer the potential of reducing the overall life cycle cost (LCC) of operating systems, decreasing the operations/maintenance logistics footprint, and increasing operational safety.

17.2 Prognostic Algorithm Approaches

In the engineering disciplines, failure prognosis has been approached via a variety of techniques ranging from Bayesian estimation and other probabilistic/statistical methods to artificial intelligence tools and methodologies based on notions from the computational intelligence arena. Specific enabling technologies include multi-step adaptive Kalman filtering (Lewis, 1986), autoregressive moving average models (Lewis, 1992), stochastic autoregressive integrated moving average models (Jardim-Goncalves *et al*., 1996), Weibull models (Groer, 2000), forecasting by pattern and cluster search (Frelicot, 1996), parameter estimation methods (Ljung, 1999), and particle filter methods (Orchard *et al*., 2005). From the artificial intelligence domain, case-based reasoning (Aha, 1997), intelligent decision-based models

[1] The term "anomaly" is derived from the Greek word *omalos*, meaning "smooth" or "even." The composite word *anomalos* means the opposite, that is, bumpy, not even, abnormal. In the context of this chapter, the term "anomalous" implies an abnormal or degraded state, whether it is anticipated or not, without knowing its identity or its severity.

[2] It should be noted that there is no consistent taxonomy in the literature about the meaning of fault and failure at this time. Within the SHM taxonomy of this book, failures or anomalies (unacceptable or unexpected performance of system function) are detected and then their causal mechanisms are isolated and identified.

and min–max graphs have been considered as potential candidates from prognostic algorithms. Other methodologies, such as Petri nets, neural networks, fuzzy systems, and neuro-fuzzy systems (Studer and Masulli, 1996), have found ample utility as prognostic tools as well. A comprehensive review of computational intelligence methods for prognostics is given in Schwabacher and Goebel (2007). Physics-based fatigue models (Ray and Tangirala, 1996; Li *et al*., 2000; Muench *et al*., 2004) have been extensively employed to represent the initiation and propagation of structural anomalies (see Chapter 12).

Next, we will provide a brief overview of a representative sample of the multitude of enabling technologies. Prognostic technologies typically utilize measured or inferred features, as well as data-driven and/or physics-based models, to predict the condition of the system at some future time. Inherently probabilistic or uncertain in nature, prognostics can be applied to failure modes governed by material conditions or by functional loss. Prognostic algorithms can be generic in design but are typically rather specific when used in the context of a particular application. Prognostic system developers have implemented various approaches and associated algorithmic libraries for customizing applications that range in fidelity from simple historical/usage models to approaches that utilize advanced feature analysis or physics of failure models.

Various approaches will be needed to develop and to implement the desired prognostic approach depending on (besides resource availability) the criticality of the LRU or subsystem being monitored, but also on the availability of data, models, and historical information. Table 17.1 provides an overview of the recommended models and information necessary for implementing specific approaches. The resolution of this table (somewhat arbitrarily) illustrates only three levels of algorithms, from the simplest experienced-based (reliability) methods to the most advanced physics of failure approaches that are calibrated by sensor data.

17.2.1 Statistical Reliability and Usage-Based Approaches

In situations where high prognostic accuracy and prognostic precision are not warranted due to the lower level of criticality or low failure occurrence rates, and/or there is an insufficient sensor network to assess condition, a statistical reliability or usage-based prognostic approach may be a suitable method. This form of prognostic algorithm is the least complex and requires only the component/LRU failure history data and/or operational usage profile data. Typically, failure and/or inspection data is compiled from legacy systems and a Weibull distribution or other statistical failure distribution can be fitted to the data (Groer, 2000; Schömig and Rose, 2003). Although simplistic, a statistical reliability-based prognostic distribution can be used to drive interval-based maintenance practices that can then be updated on regular intervals. An example may be the maintenance scheduling for an electrical component or airframe component that has few or no sensed parameters and is not critical enough to

Table 17.1 Prognostic accuracy and cost as a function of methods employed

Prognostic accuracy $\longrightarrow$			
	Experience-based	*Evolutionary*	*Physics-based*
Engineering model	Not required	Beneficial	Required
Failures history	Required	Not required	Beneficial
Past operating conditions	Beneficial	Not required	Required
Current conditions	Beneficial	Required	Required
Identified fault patterns	Not required	Required	Required
Maintenance history	Beneficial	Not required	Beneficial
In general	No sensors/no model	Sensors/no model	Sensors and model

warrant a physical model. In this case, the prognosis of when the component will fail or degrade to an unacceptable condition must be based solely on analysis of past experience or reliability. Depending on the maintenance complexity and criticality associated with the component, the prognostics system may be set up for a maintenance interval (i.e., replace every 1000 ± 20 engine flight hours), then updated as more data becomes available. Since the failure rates are typically dependent not just on operating hours but also on operating conditions (e.g., loads) and environmental conditions (e.g., temperature, vibration, etc.), Weibull curves may be adjusted to account for these factors when they are available. However, the estimates provided by this approach suffer from low accuracy and precision. There are still significant benefits to performing maintenance based on available field data.

The logical extension to a reliability-based statistical model is to correlate the failure rate data with specific operational usage profiles that are more directly related to the way a specific vehicle is used. In this manner, statistical damage accumulation models or usage models for specific components/LRUs can be directly tied to the loading profiles inferred from the high-level operations data sets, for example, fatigue cycles that are a function of operating conditions such as speed or maneuvering conditions.

It is important to recognize that this is not another form of reliability-centered maintenance, in which components are replaced based on a conservative safe-life operational time. It is a method to include the operational profile information and up-to-date reliability/inspection data in an automated algorithm that will augment existing failure detection conclusions or provide a prediction of when more accurate means are not justified.

17.2.2 Trend-Based Evolutionary Approaches

A trend-based or evolutionary prognostic approach relies on the ability to track and trend deviations and associated rates of change of these deviations of specific features or measurements from their normal operating condition. Evolutionary prognostics may be implemented on systems or subsystems that experience conditional or slow degradation-type failure effects, and where future load is similar to the load the system experienced in the past. Generally, trend-based prognostics works well for system-level degradation because conditional loss is typically the result of interaction of multiple components functioning improperly as a whole. This approach requires that sufficient sensor information is available to assess the current condition of the system or subsystem and relative level of uncertainty in this measurement. Furthermore, the parametric conditions that signify a known performance-related fault must be identifiable. While a physical or statistical model that can help classify a specific fault is beneficial, it is not an absolute requirement for this technical approach. An alternative to the physical model is knowledge of the fault condition and its manifestation in measured and extracted features. Such knowledge can be acquired by observing past fault characteristics.

This method is predicated on observing detectable features of incipient faults and performance degradations that provide a means to diagnose and predict the future progression of that fault under known operating conditions in electrical and mechanical systems. Similarly, feature-based prognostics can be implemented for electronic systems based on changes in a variety of measurable quantities including temperature, current, and voltage at various locations in the system. Features such as heat generation, electromagnetic interference (EMI), and power consumption that correlate with known faults can be extracted from the sensed data. Once these features are obtained, they can be tracked and trended over the component's life and compared to RUL estimates to provide corroborative evidence of a degrading or failing condition.

17.2.3 Data-Driven Approaches

Data-driven approaches are by some measure the most popular prognostic approaches and are arguably the low-hanging fruit of performing prognostics. This is due to the fact that no fundamental knowledge of the underlying system is necessary to perform prognostics. Instead, where historical time series fault/failure data exists with signals leading up to the failure, or where statistical data sets were recorded, this data is used in pattern matching or regression schemes. A whole host of algorithms have been reported (Schwabacher and Goebel, 2007). Popular techniques include the *neural network* (NN), which is inspired by signal processing techniques in biological nervous systems (but uses very different mathematical algorithms), and *case-based reasoning* algorithms that match the most appropriate set of examples to a given situation to reason about the remaining life (Saxena *et al*., 2008; Xue *et al*, 2008).

Forecasting researchers and practitioners have successfully availed themselves of these techniques (Sharda, 1994). Werbos (1988) reported that NNs trained with the back-propagation algorithm outperform traditional statistical methods such as regression and Box–Jenkins approaches. In a (not so recent) forecasting competition organized by Weigand and Gershenfeld (1993) through the Santa Fe Institute, all winners of each set of data used NNs. Unlike the traditional model-based methods, NNs are data driven and self-adaptive and they make very few assumptions about the underlying problem being studied. NNs learn from examples and attempt to capture the subtle functional relationship between desired output and input data. As such, NNs are well suited for practical problems where it is easier to obtain data than to obtain knowledge governing the underlying system being studied. Generally, they can be viewed as one of many multivariate nonlinear and non-parametric statistical methods (Cheng and Titerington, 1994). The main problems of NNs are that the reasoning behind their decisions is not always evident and that they produce undesired output, the result of overtraining (capturing higher-order effects that are irrelevant to the process) or attempting to use the NN in scenarios that were not bounded by the training data, thus operating it outside its experience. Nevertheless, NNs provide a feasible tool for practical prediction problems (Bonissone and Goebel, 2002; Heimes, 2008). Other commonly used data-driven methods include Gaussian process regression (Goebel *et al*., 2008).

Hence, with an understanding of how the fault/failure signature is related to specific measurable or inferred features from the system being monitored, a data-driven approach can be a popular approach. Based on the selected input features that correlate with the failure progression, a desired output prediction of the time to failure is produced based on a training process in which the network will automatically adjust its weights and thresholds based on the relationships it sees between the time to failure and the correlated feature magnitudes.

17.2.4 Particle Filtering

Particle filtering (PF) is a sequential Monte Carlo (SMC) technique for implementing a recursive Bayesian filter using Monte Carlo simulations. It is primarily used for state estimation and tracking. The mathematical formulation for PF methods has been discussed in Arulampalam *et al*. (2002). The basic idea is to provide a non-parametric representation of the system state probability density function (pdf) in the form of a set of particles with associated importance weights. The particles are sampled values from the unknown state space and the weights are the corresponding discrete probability masses. As the filter iterates, the particles are propagated according to the system state transition model, while their weights are updated based upon the likelihood of the measurement given the particle values. Resampling of the particle distribution is done when needed in order to prevent the degeneracy of the weights. For state prediction purposes the same PF framework can be used by running only the model-based particle propagation step until the predicted state value crosses some predetermined end-of-life threshold. The predicted trajectory of each particle then generates an estimate of RUL, which can be combined with the associated weights to give the RUL pdf. The process is broken down into an offline learning part, and an online tracking and prediction part. During offline analysis, regression is performed to find representative aging curves. Exponential growth models are then fitted on these curves to identify the relevant decay parameters like C and λ:

$$\theta = C\ \exp(-\lambda t)$$

where θ is an internal model parameter of interest. More details of the PF framework used here can be found in Saha and Goebel (2008).

The state and measurement equations that describe the aging model are

$$
\begin{aligned}
\mathbf{z}_k &= \mathbf{z}_{k-1} \cdot \exp[-\boldsymbol{\Lambda}_k (t_k - t_{k-1})] + \omega_k \\
\boldsymbol{\Lambda}_k &= \boldsymbol{\Lambda}_{k-1} + \boldsymbol{\nu}_k \\
\mathbf{x}_k &= [\mathbf{z}_k;\, \boldsymbol{\Lambda}_k] \\
\mathbf{y}_k &= \mathbf{z}_k + \boldsymbol{\upsilon}_k
\end{aligned}
$$

where the vector **z** consists of the exponential time decay constants for a particular component subject to damage, and matrices **C** and **Λ** contain their aging decay parameters, C and λ values, respectively. The **z** and **Λ** vectors are combined to form the state vector **x**. The measurement vector **y** comprises the time decay parameters inferred from measured data. The time index is denoted by k. The values of the **C** and **Λ** vectors learned from regression can be used to initialize the particle filter. The noise samples ω, ν, and υ are picked from zero-mean Gaussian distributions whose standard deviations are derived from the given training data, thus accommodating for the sources of uncertainty in feature extraction, regression modeling, and measurement. System importance resampling of the particles is carried out in each iteration, in order to reduce the degeneracy of particle weights. This helps in maintaining track of the state vector even under the presence of disruptive effects like unmodeled operational conditions.

The system description model developed in the offline process is fed into the online process where the particle filtering prognosis framework is triggered by a diagnostic routine. The algorithm incorporates the model parameter as an additional component of the state vector and thus performs parameter identification in parallel with state estimation. Predicted values of the time decay parameters are compared against end-of-life thresholds to derive time estimates of end of life (EOL) and RUL.

17.2.5 Physics-Based Modeling Approaches

A physics-based model is a technically comprehensive modeling approach that has been traditionally used to understand component failure mode progression. Physics-based models provide a means to calculate the damage to critical components as a function of operating conditions and to assess the cumulative effects in terms of component life usage. By integrating physical and stochastic modeling techniques, the model can be used to evaluate the distribution of remaining useful component life as a function of uncertainties in component strength/stress properties, loading, or lubrication conditions for a particular fault. Statistical representations of historical operational profiles serve as the basis for calculating future damage accumulation. The results from such a model can then be used for real-time failure prognostic predictions with specified confidence bounds.

Model-based approaches to prognostics differ from feature-based approaches in that they can make RUL estimates in the absence of any measurable events, but when related diagnostic information is present, the model can often be updated based on this new information. Therefore, a combination or fusion of the feature-based and model-based approaches provides full prognostic ability over the entire life of the component, thus providing valuable information for planning which components to inspect during specific overhaul periods. While failure modes may be unique from component to component, this combined model-based and feature-based methodology can remain consistent across different types of critical components or LRUs.

To perform prognosis with a physics-based model, information about the future operational profile (and perhaps environmental profile) must be provided. This can be done either through soliciting information from the operator or by developing an operational profile predictor using steady state and transient loads, temperatures, or other online measurements. With this capability, probabilistic critical component models can then be "run into the future" by creating statistical simulations of future operating profiles from the statistics of past operational profiles or expected future operating profiles.

The nonlinear nature associated with many damage mechanisms is dependent on both the inherent characteristics of the damage type (e.g., cracks, spalling, etc.) and operational mix types. Significant component damage resulting from large variability in operating environment and severity of the missions directly affects the vehicle component lifetimes. Very often, component lives driven by fatigue failure modes are dominated by unique operational usage profiles or a few, rare, severe, randomly occurring events, including abnormal operating conditions, random damage occurrences, etc.

17.3 Prognosis RUL Probability Density Function

One element of critical importance in prognostics is uncertainty management. If a prognostics estimate has very large uncertainty (expressing accuracy, precision, and robustness), the estimate may, in the limit, not be more useful than any standard diagnostic information because it forces an action immediately. Ideally, a probabilistic estimate is on the mark, and it stays there as further updates are made. Uncertainty is often expressed with a pdf. Uncertainty management is accomplished by updating the pdf. A comprehensive description on probabilistic techniques for predicting RUL is given in the seminal paper by Engel *et al*. (2000). In this representation, a component or LRU is recommended to be removed from service prior to attaining a high probability of failure, based on the criticality. This concept is depicted in Figure 17.1, in terms of the RUL pdf, where a *just-in-time point (JITP)* is defined for removal from service that corresponds to a particular probability (e.g., 95%) that the component has not yet failed.

A key issue, unfortunately, is that the RUL pdf is actually a *conditional pdf* that changes as time advances. In fact, one must recompute the RUL pdf at each time t based on the new information that *the component has not yet failed at that time*. This concept is shown in Figure 17.2. One starts with an a priori pdf similar to the hazard function. Then, as time passes, one must recompute the a posteriori RUL pdf based on the fact that the failure has not yet occurred. This involves renormalizing the pdf at each time so that its area is equal to one. As time passes, the variance of the RUL pdf decreases; that is, the PDF becomes narrower. This corresponds to the fact that, as time passes and one approaches the failure point, one becomes more and more certain about the time of failure and its predicted value becomes more accurate.

17.4 Adaptive Prognosis

As a direct extension to the concept presented above, the idea of updating the prognosis pdf based on additional state awareness (failure detection and fault diagnostic) information that can become

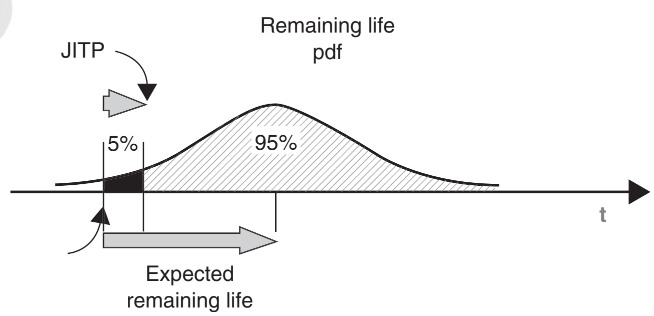

Figure 17.1 A pdf for prognosis (Engel *et al*., 2000). © 2000 IEEE

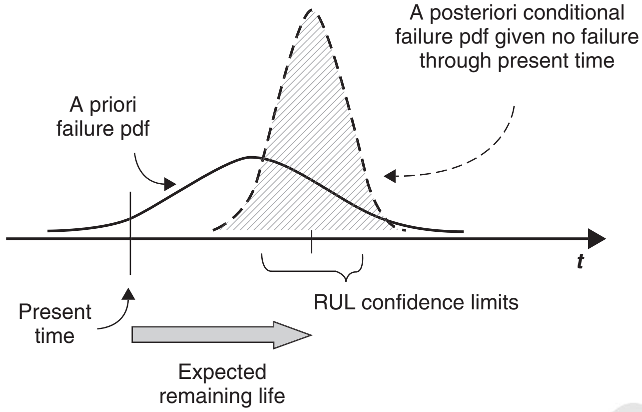

Figure 17.2 Updated prognosis pdf (Engel *et al*., 2000). © 2000 IEEE

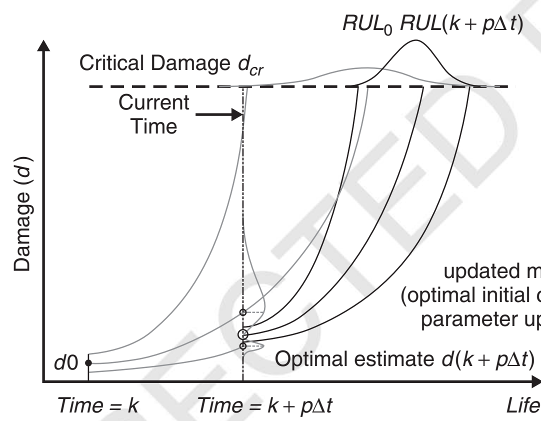

Figure 17.3 Adaptive prognosis concept (Engel *et al*., 2000)

available over time is also desirable. The adaptive prognosis concept entails that information available at the current time (which may or may not be diagnostic in nature) be used to modify future predictions, hence updating the prognosis pdf. This idea is illustrated in Figure 17.3 (Engel *et al*., 2000) and briefly described next.

Consider point $d0$ in **Figure** 17.3 to be the mean initial damage condition for a prognostic model. A prognosis of life, from time k to a predetermined damage level, is found to be represented by RUL_0 or Remaining Useful Life. Suppose that some imperfect measurement $z(k)$ regarding the damage state becomes available after time ΔT has passed, namely, at time $k = k + \Delta T$. The challenge is to find the optimal current damage state to reinitialize the model and/or adjust model parameters so that a calibrated and more accurate prognosis can be established.

Through utilization of a new initial condition, $\tilde{\boldsymbol{d}}(\boldsymbol{k})$, at time $k = k + \Delta T$ as shown in Figure 17.3, it is apparent that the prediction mean has shifted and the confidence bounds on the resulting RUL have less variance than the original. The prediction accuracy improvement would generally mean that a decision to take action based on failure probability will likely reduce lost operational availability over a run-to-failure maintenance plan.

17.5 Performance Metrics

Performance metrics are an important tool in defining requirements and validating how well an algorithm works. Metrics used in other fields of SHM (such as diagnostics) do not work well for prognostics (Saxena *et al*., 2008). The most widely used performance metrics are accuracy, precision, and convergence.

17.5.1 Accuracy

Accuracy, the degree of closeness of a predictive estimate to its actual value, represents one of the most important factors in determining the usefulness of prediction. One of the difficulties in dealing with the criterion of accuracy has been the absence of a single universally accepted measure of accuracy (Makridakis *et al*., 1983). One measure often used is the average bias (shown below). Other traditional metrics include the mean-squared error and the mean absolute percentage error. Newer metrics that are designed specifically with prognostics in mind and that address some of the shortcomings of traditional metrics include the prognostic horizon (PH), α–λ performance, and convergence (Saxena *et al*., 2008). These are described below.

Average bias This is computed by averaging $l = |(i|P \leq i \leq EOP)|$ where EOP (End Of Prediction) is defined as the earliest time index, i, after the prediction crosses the failure threshold. Also, $\Delta^l(i)$ is the error between the predicted and the true RUL at time index i for unit under test (UUT) l. Then,

$$B_l = \frac{1}{\ell} \sum_{i=1}^{\ell} \Delta^l(i)$$

This metric conventionally aggregates prediction errors obtained either from multiple experiments or from a set of similar systems operating under identical conditions. In this form it does not account for the variability in predictions and the presence of outliers.

Prognostic horizon (PH) The longer the PH, the more time is available to act based on a prediction. We define the PH as the difference between the current time index i and the EOP utilizing data accumulated up to the time index i, provided the prediction meets desired specifications. This specification may be defined in terms of allowable error bound (α) around true EOL (which represents the time index for actual EOL defined by the failure threshold). This metric ensures that the predicted estimates are within specified limits around the actual EOL and that the predictions may be considered reliable. It is expected that PHs are determined for an algorithm–application pair offline during the validation phase and then these numbers can be used as guidelines when the algorithm is deployed in test application where actual EOL is not known in advance. While comparing algorithms, an algorithm with a longer prediction horizon would be preferred:

$$H = EOP - i$$

where

$$i = \min\left\{ j|(j \in \ell) \wedge \left(r_*(1-\alpha) \leq r^l(j) \leq r_*(1+\alpha)\right)\right\}$$

$r_*(i)$is the true RUL at time t_i given that data is available up to time t_i, and $r^l(i)$ is the RUL estimate for the lth UUT at time t_i as determined from measurement and analysis.

For instance, a PH with an error bound of $\alpha = 5\%$ identifies when a given algorithm starts predicting estimates that are within 5% of the actual EOL. Other specifications may be used to derive PH as desired.

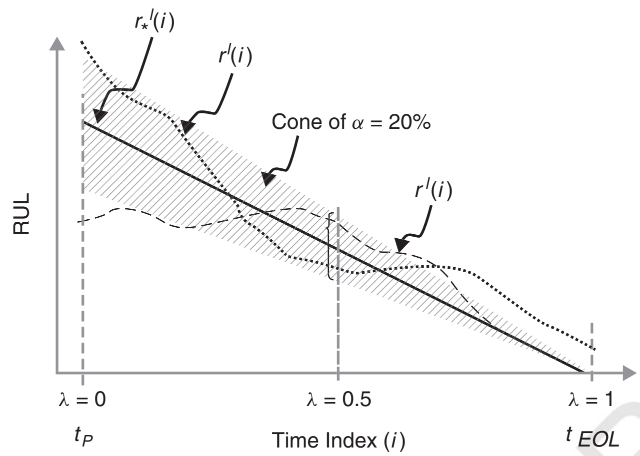

Figure 17.4 Schematic depicting $\alpha - \lambda$ accuracy (Saxena *et al*., 2008). © 2000 IEEE

α–λ performance Another way to quantify prediction quality may be through a metric that determines whether the prediction falls within specified levels of a performance measure at particular times. These time instances may be specified as a percentage of total remaining life from the point the first prediction is made or a given absolute time interval before EOL is reached. For instance, in our implementation we define $\alpha - \lambda$ accuracy as the prediction accuracy to be within α*100% of the actual RUL at specific time instance t_λ expressed as a fraction of time between the point when an algorithm starts predicting and the actual failure. For example, this metric determines whether a prediction falls within 20% accuracy (i.e., $\alpha = 0.2$) halfway to failure from the time the first prediction is made (i.e., $\lambda = 0.5$). The metric is visualized in Figure 17.4. An extension of this metric based on other performance measures is straightforward:

$$[1 - \alpha] \cdot r_*(t) \leq r^l(t_\lambda) \leq [1 + \alpha] \cdot r_*(t)$$

with α the accuracy modifier, λ the time window modifier,

$$t_\lambda = P + \lambda(EOL - P)$$

and P the time index at which the first prediction is made by the prognostic system.

17.5.2 Precision

Precision-based metrics are designed to quantify variability in predictions. Variability in predictions arises from different raw data series, the extent of data preprocessing, prognostic algorithms, different prediction horizons, different time scales involved, etc. Sample standard deviation measures the dispersion/spread of the error with respect to the sample mean of the error:

$$S = \sqrt{\frac{\sum_{i=1}^{l} (\Delta(i) - m)^2}{l - 1}}$$

where m is the sample mean of the error.

This particular metric instantiation is restricted to the assumption of normal distribution of the error. It is, therefore, recommended to carry out an inspection (visual or otherwise) of the error plots to determine the distribution characteristics before interpreting this metric.

Note that $\alpha - \lambda$ performance and PH can also be computed as precision metrics.

17.5.3 Convergence

Convergence is defined to quantify the manner in which any metric like accuracy or precision improves with time to reach its final estimate. As illustrated below, three cases converge at different rates. It can be shown that the distance between the origin and the centroid of the area under the curve for a metric quantifies convergence. A smaller distance implies faster convergence. A prognostic algorithm is expected to converge to the true value as more information becomes available over time. Fast convergence is desired to achieve a high confidence in keeping the prediction horizon as large as possible.

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

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

$$x_c = \frac{\frac{1}{2}\sum_{i=P}^{EOP}(t_{i+1}^2 - t_i^2)M(i)}{\sum_{i=P}^{EOP}(t_{i+1} - t_i)M(i)}$$

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

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

The metric is visualized in Figure 17.5.

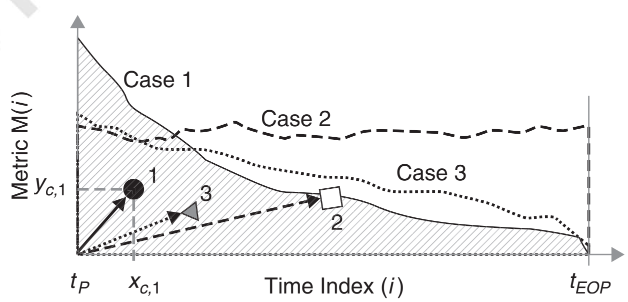

Figure 17.5 Schematic for the convergence of a metric (Saxena *et al.*, 2008). © 2000 IEEE

17.6 Distributed Prognosis System Architecture

The cornerstone of an effective SHM system is the information/data architecture and the ability for understanding and managing the anomaly, diagnostic, and prognostic (A/D/P) information from the LRU level all the way up through to the subsystem and vehicle-level reasoners. In general, the A/D/P technologies implemented at the lower levels (LRUs) are used to detect and predict off-nominal conditions or damage accumulating at an accelerated rate. In the distributed health management (HM) architecture, this information is analyzed through the hierarchy of reasoners to make informed decisions on the health of the vehicle subsystems/systems and how they affect total vehicle capability. This integration across LRUs, subsystems, and systems is vital to correctly isolating the root cause of failures and understanding the propagation of up/downstream effects of the faults. Integration of the individual subsystem HM results is eventually accomplished with the vehicle-level reasoner, which will assess the intra-system A/D/P results in order to prioritize the recommended actions to perform in order to correct the problem. These actions include maintenance action, but they can also be changes in operational behavior to extend system life or the action can be an on-board reconfiguration to compensate for the shortcomings detected.[3] Some initial studies on autonomous reconfiguration that specifically incorporate prognostic information have recently been carried out (Tang *et al*., 2008). Other studies describe how to perform post-prognostic decision support (Iyer *et al*., 2006) as a multi-objective optimization problem.

Challenges in system-level reasoning may arise from the large amount of different information pieces which an integrated architecture has to process. Conflicting information from different subsystems of the system with different levels of uncertainty and criticality, correlation between different components and subsystems, varying flight loads, dynamically changing requirements from operations and fleet management, and the need to provide an accurate health assessment within strict time constraints make system-level reasoning a difficult undertaking. Optimal health reasoning needs to be based on rigorous analysis of the information from different HM modules. At the same time, the uncertainty associated with each information piece needs to be quantified and included in the decision-making process to yield an outcome with the highest probability to provide the optimal information for a decision-maker.

A distributed HM architecture has many benefits including: (1) optimal computational resource management (i.e., placing high-bandwidth processing at the lowest level and only passing up critical features); (2) supporting the concept of "Smart LRU/Subsystem," where the most detailed "intelligence" about the system exists (i.e., supplier/designer responsibility); (3) providing the ability to isolate and assess the extent of multiple faults and battle damage, hence improving survivability of the vehicle; (4) hierarchical reasoners which have a "built-in" data management capability for containing erroneous information and utilizing multiple data and information sources; and (5) the ability to capture and localize system degradations (as opposed to only hard failures), based on increased health awareness of the lowest-level LRUs, hence providing a more accurate vehicle availability assessment.

17.7 Conclusions

This chapter reviewed generic prognosis algorithmic approaches and introduced some of the basics associated with probabilistic predictions and a required architecture for performing prognostics on critical aerospace systems.

Prognosis is a critical element of a HM system and has the promise to realize major benefits for cost avoidance and safety improvement for fielded systems. It also presents a number of challenges to the HM system designer, primarily due to the need to properly model damage progression and to deal with large-grain uncertainty. Long-term prediction of a fault's evolution to the point that

[3] In the terminology of this book, these actions are mechanisms to perform failure preclusion or prevention functions.

may result in a failure requires means to represent and manage the inherent uncertainty. Moreover, accurate and precise prognosis demands good models of the fault growth and statistically sufficient samples of failure data to assist in training, validating, and fine tuning prognostic algorithms. Prognosis performance metrics, robust algorithms, and test platforms that may provide needed data have been the target of HM researchers in the recent past. Many accomplishments have been reported but major challenges still remain to be addressed.

To address the issue of inherent uncertainties that are the aggregate of many unknowns and can result in considerable prediction variability, the concept of adaptive prognosis was introduced. In that case, available, albeit imperfect, information is used to update elements of the prognostic model. Only one of many approaches for accomplishing this was briefly introduced, namely, the particle filter. Other statistical update techniques include Bayesian updating, constrained optimization, and Kalman filtering.

The design process is not a trivial process by which features and models are chosen for integration such that the best possible prediction on RUL still is obtained. It takes substantial effort to design systems so that measured data can be fused and used in conjunction with physics-based models to estimate current and future damage states. This is exacerbated when multiple models are employed that may use different feature inputs. The prognosis system must also be capable of intelligently calibrating a priori initial conditions (e.g., humidity, strain, and temperature) and random variable characteristics in an automated yet lucid process.

Bibliography

Aha, D.W. (1997) Special issue on lazy learning. *Artificial Intelligence Review*, **11**, 1-5–6.

Arulampalam, S., Maskell, S., Gordon, N.J., and Clapp, T. (2002) A tutorial on particle filters for on-line non-linear/non-Gaussian Bayesian tracking. *IEEE Transactions on Signal Processing*, **50**, 174–88.

Q1 Atkeson, C.G., Moore, A.W., and Schaal, S. (1997) locally weighted learning. *Artificial Intelligence Review*, **11**, 1–5, 11–73.

Bonissone, P. and Goebel, K. (2002) When will it break? A hybrid soft computing model to predict time-to-break margins in paper machines. *Proceedings of SPIE 47th Annual Meeting, International Symposium on Optical Science and Technology*, Vol. 4787, 53–64.

Q2 Chelidze, D. and Cusumano, J.P. (2004) A dynamical systems approach to failure prognosis. *Journal of Vibration and Acoustics*, **126**, 2–8.

Cheng, J. and Titerington, D.M. (1994) Neural networks: a review from a statistical perspective. *Statistical Science*, **9** (1), 2–54.

Engel, S.J., Gilmartin, B.J., Bongort, K., and Hess, A. (2000) Prognostics, the real issues involved with predicting life remaining. *Aerospace Conference Proceedings*, **6**, 457–69.

Frelicot, C. (1996) A fuzzy-based prognostic adaptive system. RAIRO-APII-JESA, *Journal Européen des Systemes Automatises*, **30** (2–3), 281–99.

Goebel, K., Saha, B., and Saxena, A. (2008) A comparison of three data-driven techniques for prognostics. Proceedings of MFPT.

Groer, P.G. (2000) Analysis of time-to-failure with a Weibull model. Proceedings of the Maintenance and Reliability Conference, MARCON.

Q3 Hagan, M.T. and Menhaj, M. (1994) Training feedforward networks with the Marquard algorithm. *IEEE Transactions on Neural Networks*, **5**, 989–93.

Heimes, F. (2008) Recurrent neural networks for remaining useful life estimation. Proceedings of International Conference on Prognostics and Health Management.

Iyer, N., Goebel, K., and Bonissone, P. (2006) Framework for post-prognostic decision support. Proceedings of the IEEE Aerospace Conference, 11.0903.

Q4 Jardim-Concalves, R., Martins-Barata, M., Assis-Lopes, J. A., and Steiger-Garcao, A. (1996) Applications of stochastic modeling to support predictive maintenance for industrial environments. Proceedings of the 1996 IEEE International Conference on Systems, Man, and Cybernetics, Information Intelligence and Systems, Vol. 1, pp. 117–22.

Q5 Khiripet, N. (2001) An architecture for intelligent time series prediction with causal information. PhD thesis. Georgia Institute of Technology.

Q6 Leonard, J.A., Kramer, M.A., and Ungar, L.H. (1992) A neural network architecture that computes its own reliability. *Computers in Chemical Engineering*, **16** (9), 819–35.

Lewis, F.L. (1986) *Optimal Estimation: With an Introduction to Stochastic Control Theory*, John Wiley & Sons, Inc., New York.

Lewis, F.L. (1992) *Applied Optimal Control and Estimation: Digital Design and Implementation*, TI Series, Prentice Hall, Englewood Cliffs, NJ.

Li, Y., Kurfess, T.R., and Liang, S.Y. (2000) Stochastic prognostics for rolling element bearings. *Mechanical Systems and Signal Processing*, **14**, 747–62.

Q7 Liu, J.S. and Chen, R. (1998) Sequential Monte Carlo methods for dynamical systems. *Journal of the American Statistical Association*, **93**, 1032–44.

Ljung, L. (1999) *System Identification: Theory for the User* (2nd ed.) Prentice Hall: Englewood Cliffs, NJ.

Makridakis, S., Wheelwright, S.C., and McGee, V.E. (1983) *Forecasting: Methods and Applications* (2nd ed.), John Wiley & Sons, Inc., New York.

Marko, K.A., James, J.V., Feldkamp, T.M. *et al*. (1996) Applications of neural networks to the construction of "virtual" sensors and model-based diagnostics. Proceedings of ISATA 29th International Symposium on Automotive Technology and Automation, pp. 133–8.

Q8 Minsky, M.L. (1961) Step toward artificial intelligence. *Proceedings of the IRE*, **49**, 8–30.

Muench, D., Kacprzynski, G., Liberson, A. *et al*. (2004) Model and sensor fusion for prognosis, example: Kalman filtering as applied to corrosion-fatigue and FE models. SIPS Quarterly Review presentation.

Orchard, M., Wu, B., and Vachtsevanos, G. (2005) A particle filtering framework for failure prognosis. Proceedings of the World Tribology Congress.

Ray, J. and Tangirala, S. (1996) Stochastic modeling of fatigue crack dynamics for on-line failure prognostics. *IEEE Transactions on Control Systems Technology*, **4** (4), 443–51.

Saha, B. and Goebel, K. (2008) Uncertainty management for diagnostics and prognostics of batteries using Bayesian techniques. Proceedings of the IEEE Aerospace Conference, 10.1109.

Q9 Saxena, A., Wu, B., and Vachtsevanos, G. (2005) Integrated diagnosis and prognosis architecture for fleet vehicles using dynamic case-based reasoning. Proceedings of Autotestcon, pp. 26–9.

Saxena, A., Celaya, J., Balaban, E. *et al*. (2008) A survey of metrics for performance evaluation of prognostics. Proceedings of International Conference on Prognostics and Health Management.

Schauz, J.R. (1996) Wavelet neural networks for EEG modeling and classification. PhD thesis. Georgia Institute of Technology.

Q10 Schömig A.K. and Rose, O. (2003) On the suitability of the Weibull distribution for the approximation of machine failures. Proceedings of the 2003 Industrial Engineering Research Conference.

Schwabacher, M. and Goebel, K. (2007) A survey of artificial intelligence for prognostics. Working Notes of 2007 AAAI Fall Symposium: AI for Prognostics.

Sharda, R. (1994) Neural network for the MS/OR analyst: an application bibliography. *Interfaces*, **24** (2), 116–30.

Shiroishi, J., Li, Y., Liang, S. *et al*. (1997) Bearing condition diagnostics via vibration and acoustic emission measurements. *Mechanical Systems and Signal Processing*, **11** (5), 693–705.

Q11 Specht, D.F. (1991) A general regression neural network. *IEEE Transactions on Neural Networks*, **2** (6), 568–76.

Studer, L. and Masulli, F. (1996) On the structure of a neuro-fuzzy system to forecast chaotic time series. International Symposium on Neuro-Fuzzy Systems, pp. 103–10.

Q12 Sutton, R.S. (1992) Introduction: the challenge of reinforcement learning. Machine Learning, **8**, 225–7.

Tang, L., Kacprzynski, G., Orchard, M. *et al*. (2008) Prognostics-enhanced automated contingency management for advanced autonomous systems. Proceedings of International Conference on Prognostics and Health Management.

Q13 Veaux, D.S.J., Schweinsberg, J., and Ungar, J. (1998) Prediction intervals for neural networks via nonlinear regression. *Technometrics*, **40** (4), 273–82.

Q14 Vichare, N. and Pecht, M. (2006) Prognostics and health management of electronics. *IEEE Transactions on Components and Packaging Technologies*, **29** (1), 222–9.

Q15 Wang , P. and Vachtsevanos, G. (2001) Fault prognostics using dynamic wavelet neural networks. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, **15** (4), 349–65.

Weibull, W. (1951) A statistical distribution function of wide applicability. *Journal of Applied Mechanics*, **18**, 293–7.

Weigand, A.S. and Gershenfeld, N.A. (1993) *Time Series Prediction: Forecasting the Future and Understanding the Past*, Addison-Wesley, Reading, MA.

Werbos, P.J. (1988) Generalization of back propagation with application to recurrent gas market model. *Neural Networks*, **1**, 339–56.

Xue, F., Goebel, K., Bonissone, P., and Yan, W. (2008) An instance-based method for remaining useful life estimation for aircraft engines. Proceedings of MFPT.

Queries in Chapter 17

Q1. Atkeson, C.G., Moore, A.W., and Schaal, S. (1997) has not been cited in the text. Please clarify that where it has to be cited.

Q2. Chelidze, D. and Cusumano, J.P. (2004) has not been cited in the text. Please clarify that where it has to be cited.

Q3. Hagan, M.T. and Menhaj, M. (1994) has not been cited in the text. Please clarify that where it has to be cited.

Q4. Jardim-Concalves, R., Martins-Barata, M., Assis-Lopes, J. A., and Steiger-Garcao, A. (1996) has not been cited in the text. Please clarify that where it has to be cited.

Q5. Khiripet, N. (2001) has not been cited in the text. Please clarify that where it has to be cited.

Q6. Leonard, J.A., Kramer, M.A., and Ungar, L.H. (1992) has not been cited in the text. Please clarify that where it has to be cited.

Q7. Liu, J.S. and Chen, R. (1998) has not been cited in the text. Please clarify that where it has to be cited.

Q8. Minsky, M.L. (1961) has not been cited in the text. Please clarify that where it has to be cited.

Q9. Saxena, A., Wu, B., and Vachtsevanos, G. (2005) has not been cited in the text. Please clarify that where it has to be cited.

Q10. Schömig A.K. and Rose, O. (2003) has not been cited in the text. Please clarify that where it has to be cited.

Q11. Specht, D.F. (1991) has not been cited in the text. Please clarify that where it has to be cited.

Q12. Sutton, R.S. (1992) has not been cited in the text. Please clarify that where it has to be cited.

Q13. Veaux, D.S.J., Schweinsberg, J., and Ungar, J. (1998) has not been cited in the text. Please clarify that where it has to be cited.

Q14. Vichare, N. and Pecht, M. (2006) has not been cited in the text. Please clarify that where it has to be cited.

Q15. Wang, P. and Vachtsevanos, G. (2001) has not been cited in the text. Please clarify that where it has to be cited.