Chapter 4

Prognostics

Kai Goebel, George Vachtsevanos, and Marcos E. Orchard

With silver spears you may conquer the world.—Oracle at Delphi

4.1 Introduction

The methods for performing prognostics have changed considerably since the days when Philip II of Macedon consulted the Oracle at Delphi. Knowledge discovery, statistical learning, and—more specifically—an understanding of the system evolution in time when it undergoes fault conditions (note that we distinguish here between fault and failure, the former an undesirable out-of-spec condition that may worsen toward a failure, the complete loss of function), are critical for an adequate implementation of successful prognostic systems. Prognosis may be understood as the generation of long-term predictions describing the evolution in time of a particular signal of interest or condition indicator, with the purpose of estimating the remaining useful life (RUL) of a failing component/subsystem. Predictions are made using a thorough understanding of the underlying processes and factoring in the anticipated future usage.

In the past, IVHM practitioners have used the terms “trending” and “prognosis” interchangeably. Operators then may have followed the evolution of a single variable, temperature for example, and performed mentally a linear extrapolation as to when the variable might reach a specified threshold requiring equipment maintenance. Failure prognosis, on the other hand, involves fault detection and tracking/prediction of the fault evolution (growth) via process modeling, data analysis, and the utility of estimation algorithms, such as Kalman filtering, particle filtering, and nonlinear regression, among others. We use the term “prognosis” in this chapter in deference to trending, as suggested above.

A fundamental challenge in prognosis stems from the “large-grain” uncertainty inherent in the prediction task. Long-term prediction of the fault evolution, to the point that may result in a failure, requires means to represent and manage the inherent uncertainty. Indeed, data uncertainty, process (system) uncertainty, load uncertainty, and measurement and modeling uncertainties are potential contributors to prognostic uncertainty.

The prognosis scheme should consider critical state variables (such as condition indicators) as random processes in such a way that, once their probability distributions are estimated, other important attributes (such as confidence intervals) may be computed.

An important distinction must be drawn between two major categories of prognostic algorithms: health-based vs. usage-based prognostics. The former refers to prognostic approaches developed and applied online in real time as the system/component at hand is monitored and data are streaming into a processor for fault detection and failure prognosis. In this case, an incipient failure or fault is detected first with specified confidence, and then the prognostic algorithm is initiated to predict the time evolution. The final fault state acts as the initial condition for prognosis.

In contrast, usage-based prognosis considers the past, current, and assumed future usage or stress patterns of the system to estimate the system’s remaining useful life. Such prognostic methods do not presuppose the existence of fault or incipient failure modes, in contrast to health-based prognostics. The remaining useful life may be estimated at any time in the system’s operating history in the absence of a fault. The prediction may be continuously updated as new evidence accumulates. Life-cycle management tools for critical systems take advantage of usage-based prognostic routines to arrive at times needed for maintenance, repair, and overhaul of such systems. The enabling technologies include neuro-fuzzy systems, response surface methodologies, similarity-based methods, and regression analysis techniques, among others.

There are some basic ingredients that are common to all prognostic approaches. These are a model that describes both the system under investigation and damage propagation, a quantification of the damage threshold, an algorithm that handles the propagation of the damage into the future, and a mechanism to deal with uncertainty. These elements will be discussed in more detail in the next few sections.

4.2 System Model

The system model describes the characteristics of the system under nominal conditions. Ideally, such a model should be able to factor in the effects of operational and environmental conditions as well as any other conditions that cause different system response under nominal conditions. The system model could integrate domain expertise and be implemented using either knowledge-based rules, a physical description of the system under investigation, or a structure that could learn the system behavior from examples (for instance, using machine learning techniques).

4.3 Damage Propagation Model

A damage propagation model describes how the damage is expected to grow in the future. It should, similar to the system model, account for operational and environmental conditions as well as any other conditions that have an impact on the damage. While one often thinks of damage as a monotonically increasing phenomenon, it is possible for the domain in which damage is evaluated to have non-monotonic attributes. These could be either intrinsic attributes (for example, recovery effects in batteries' capacity or power semiconductors) or extrinsic effects, such as partial maintenance actions. Depending on the fault mode, damage propagation may exhibit different symptoms, and it may be necessary to consider dedicated damage propagation models for different fault modes.

4.4 Prognostic Algorithm

The role of the prognostic algorithm is applying the damage propagation model into the future. The damage propagation algorithm must properly consider the effects of environmental and operational conditions, and, possibly, healing phenomena.

Depending on the implementation, it is sometimes not completely easy to separate the damage propagation model and the prognostic algorithm. For the general case, we will treat the damage propagation model and the prognostic algorithm as separate.

4.5 Damage Threshold

Damage thresholds define the condition representing the end-of-life, or failure. An end-of-life threshold in prognostics must be a measurable condition, although it should be noted that this condition is not always the complete destruction of a component. In fact, to be effective there should always be some margin between this threshold and complete destruction. Often, thresholds are performance specifications that represent a graceful degradation in performance that constrains the operation of the component, or system. Thus, it is typical that the system may continue to operate beyond the end-of-life threshold conditions (but outside of the specifications for maintenance).

4.6 Prognosis and Uncertainty Characterization

Prognostics are not really useful unless the uncertainties in the predictions are accounted for. Uncertainty management tools seek to improve the signal (fault) to noise (uncertainty) ratio. They begin by determining the uncertainty sources in terms of an uncertainty tree and then exploiting filtering or kernel-based methods for uncertainty management [Orchard et al. 2010]. A remaining life estimate that has no quantification of the uncertainty bounds leaves the user with little practical information. Accounting for the various sources of uncertainty and rigorously combining them will allow decision makers to justify the action taken. Uncertainties need to be managed carefully, because a haphazard stacking of the various uncertainties may lead to wide bounds that wipe out the benefits of estimating remaining life. Sources of uncertainty arise from a wide range of influences including the models (both their structures and their parameters), the current state estimate, the future load and environmental conditions, and sensor noise, just to name a few.

Figure 4.1 summarizes the range of possible prognostic approaches as a function of the applicability to various systems and their relative implementation cost. The pyramid starts at the base with generic, statistical life usage and experience-based prognostic models, migrates to basically data-driven techniques employing evolutionary or trending models, while the top of the pyramid is occupied by physics-based models employed for prognostic purposes. Prognostic technologies typically use measured or inferred features, in combination with data-driven and/or physics-based models, to predict the condition of the system at some future time [Orchard and Vachtsevanos 2009] [Engel, Gilmartin, Bongort, and Hess 2000]. Prognostic techniques combining data-driven and physics-based models are sometimes referred to as ‘hybrid’ prognostics. Model-based prognostics, at the top of the pyramid, are expected to guide the future development of reliable and verifiable prognostic algorithms for complex systems such as aircraft.

Inherently probabilistic or uncertain in nature, prognosis can be applied to failure modes governed by material condition or by functional loss. Prognosis algorithms can be generic in design but specific in terms of application. Prognosis 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 use advanced condition indicator analysis or physics-of-failure models.

Depending on the complexity and criticality of the component/system being monitored, various levels of data, models, and historical information are needed to develop and implement the desired prognostic approach.

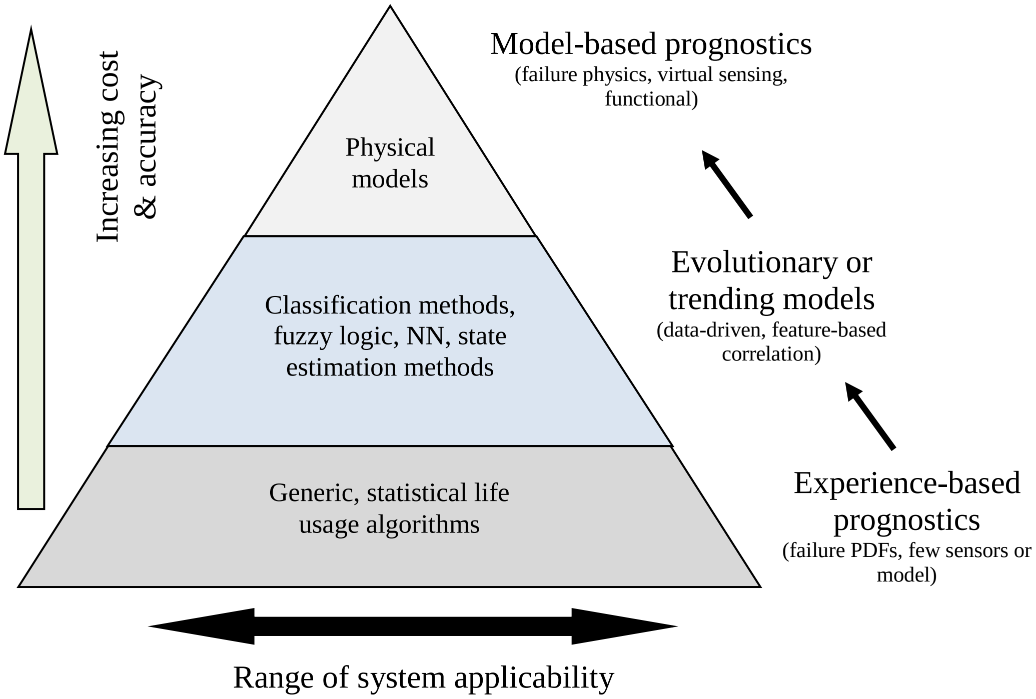
Increasing cost & accuracy
Physical models
Classification methods, fuzzy logic, NN, state estimation methods
Generic, statistical life usage algorithms
Model-based prognostics
(failure physics, virtual sensing, functional)
Evolutionary or trending models
(data-driven, feature-based correlation)
Experience-based prognostics
(failure PDFs, few sensors or model)
Range of system applicability

Fig. 4.1 A taxonomy of prognostic approaches.

In the engineering disciplines, fault 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 multistep adaptive Kalman filtering [Lewis 1986], auto-regressive moving-average models [Pham and Yang 2010], stochastic autoregressive integrated-moving-average models [Jardim-Gonçalves, Martins-Barata, Assis-Lopes, and Steiger-Garcao 1996], Weibull models [Groer 2000], forecasting by pattern and cluster search [Frelicot 1996], and parameter estimation methods [Ljung 1999]. From the artificial intelligence domain, case-based reasoning [Aha 1997], intelligent decision-based models, and min-max graphs have been considered as potential candidates for 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. Physics-based fatigue models [Tangirala 1996] have been employed extensively to represent the initiation and propagation for structural life and failure prediction.

4.7 Prognostic Techniques

4.7.1 Model-Based Techniques

Model-based prognostic schemes include those that employ a dynamic model of the process being predicted. These can include physics-based models, autoregressive moving-average (ARMA) techniques, Bayesian filtering algorithms, and empirical-based methods. Model-based methods provide a technically comprehensive approach that has been used traditionally to understand component failure mode progression.

Figure 4.2 depicts a model-based prognostic scheme. Input from the diagnostic block is combined with stress profiles and feeds into the fault growth model. An estimation method (in this case particle filtering) is called upon to propagate the fault model initially, one step at a time, while model parameters are updated online in real time as new sensor data become available.

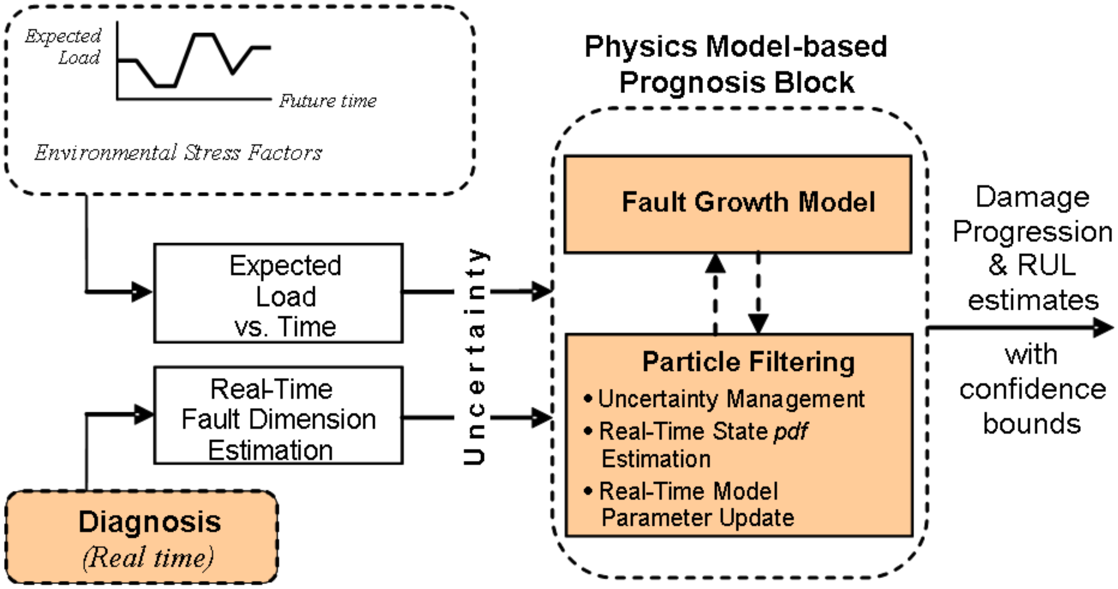
Expected Load
Future time
Environmental Stress Factors
Expected Load vs. Time
Real-Time Fault Dimension Estimation
Diagnosis
(Real time)
Uncertainty
Physics Model-based Prognosis Block
Fault Growth Model
Particle Filtering
• Uncertainty Management
• Real-Time State pdf Estimation
• Real-Time Model Parameter Update
Damage Progression & RUL estimates
with confidence bounds

Fig. 4.2 A scheme for model-based prognosis.

Eventually, the model is allowed to perform long-term prognosis of the remaining useful life of the failing component/system with confidence bounds. The fault model PDF is convolved with the hazard zone PDF when the former reaches the threshold bounds and the resultant PDF is projected along the time axis (which is usually measured in “cycles” of operation) depicting the system’s remaining life statistics.

Implementing physics-based models can provide a means to calculate the damage to critical components as a function of operating conditions and 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 life as a function of uncertainties in component strength/stress properties and loading conditions for a particular fault.

Bayesian estimation techniques can satisfy these requirements. In particular, particle filtering and learning strategies [Orchard and Vachtsevanos 2009; Orchard and Vachtsevanos 2007] employ a state dynamic model and a measurement model to predict the posterior probability density function of the state to predict the time evolution of a fault or fatigue damage. Particle filters avoid the linearity and Gaussian noise assumption of Kalman filtering, and provide a robust framework for long-term prognosis while accounting effectively for uncertainties. Correction terms are estimated in a learning paradigm to improve the accuracy and precision of the algorithm for long-term prediction.

Particle filtering methods assume that the state equations that represent the evolution of the fault mode in time can be modeled as a first order Markov process with additive noise and conditionally independent outputs [Arulampalam, Maskel, Gordon, and Clapp 2002]. Let

$$x_k = f_{k-1}\left(x_{k-1}\right) + \omega_{k-1} \tag{4.1}$$

$$z_k = h_k\left(x_k\right) + \nu_k \tag{4.2}$$

where the state vector x_k includes a set of parameters that characterizes the evolution in time of the fault condition; the process noise ω_{k-1} represents the model uncertainty; z_k is the observation (measurement); and ν_k is the measurement noise (uncertainty associated to sensors specifications and feature computation processes).

While there are several flavors of particle filters, the focus here is on algorithms based on the concept of *Sampling Importance Resampling* (SIR), in which the posterior filtering distribution denoted as $\pi\left(x\right) = p(x_k \mid z_k)$ is approximated by a set of *N* weighted particles $\left\{\left\langle x_k^i, w_k^i \right\rangle; i = 1, \ldots, N\right\}$ sampled from an arbitrarily proposed distribution $q\left(x\right)$ that intends to be "similar" to $\pi\left(x\right)$ (i.e., $\pi\left(x\right) > 0 \Rightarrow q\left(x\right) > 0$ for all $x \in R^{n_x}$), The *importance weights* w_k^i are proportional to the likelihood $p(z_k \mid x_k^i)$ associated to the sample x_k^i, and normalized as in Eq. 4.3:

$$w_k^i = \frac{\pi\left(x_k^i\right)/q\left(x_k^i\right)}{\sum_{j=1}^{N}\pi\left(x_k^j\right)/q\left(x_k^j\right)} \tag{4.3}$$

such that $\sum_{j=1}^{N} w_k^i = 1$, and the posterior distribution (a.k.a the *target* distribution) can be approximated as

$$p\left(x_k | z_k\right) = \sum_{i=1}^{N} w_k^i \delta\left(x_k - x_k^i\right) \tag{4.4}$$

Thus, as in any Bayesian processor, the filtering stage is implemented in two steps: the computation of the *a priori* state density estimate (prediction step), and the update of the estimate according to the information presented by new measurements. Using the model in Eq. 4.1, the prediction step becomes

$$p\left(x_k | z_{k-1}\right) \approx \sum_{i=1}^{N} w_{k-1}^i \delta\left(x_k - f_{k-1}\left(x_{k-1}^i\right) - \omega_{k-1}^i\right) \tag{4.5}$$

The update step, on the other hand, modifies the particle weights according to the relation

$$\overline{w}_k^i = w_{k-1}^i \frac{p(z_k \mid x_k^i)\, p(x_k^i \mid x_{k-1}^i)}{q(x_k^i \mid x_{k-1}^i, z_k)} \tag{4.6}$$

$$w_k^i = \frac{\overline{w}_k^i}{\sum_{j=1}^{N} \overline{w}_k^j} \tag{4.7}$$

It is possible that all but a few of the importance weights degenerate such that they are close to zero. In that case, one has a very poor representation of the system state (and also wastes computing resources on unimportant calculations). To address that, *resampling* of the weights can be used [Arulampalam, Maskel, Gordon, and Clapp 2002]. The basic logical flowchart is shown in Fig. 4.3.

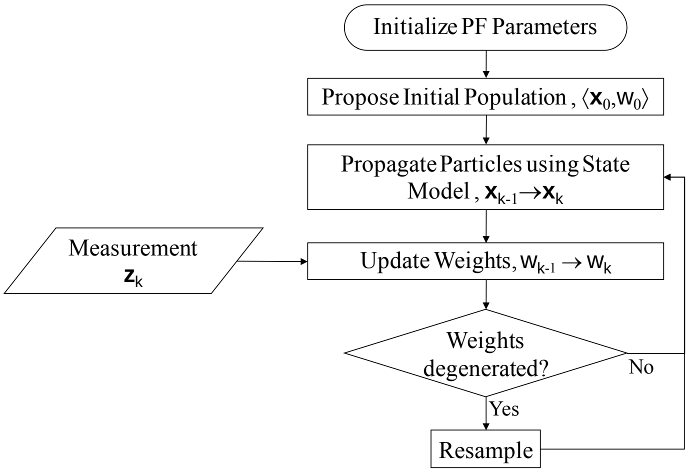
Initialize PF Parameters
Propose Initial Population , ⟨$\mathbf{x}_0$,w_0⟩
Propagate Particles using State Model , $\mathbf{x}_{k-1}\rightarrow\mathbf{x}_k$
Measurement $\mathbf{z}_k$
Update Weights, $w_{k-1} \rightarrow w_k$
Weights degenerated?
No
Yes
Resample

Fig. 4.3 Particle filtering flowchart.

Prognosis, and thus the generation of long-term predictions, is a problem that goes beyond the scope of filtering applications because it involves future time horizons. Hence, a particle-filtering-based prognostic approach requires proposing a procedure to project the current estimate of the state probability density function (PDF) in time. The simplest implementation that can be used to solve this problem uses Eq. 4.1 recursively to propagate the posterior PDF estimate defined by $\left\{\left\langle x_p^i, w_p^i \right\rangle; i = 1,\ldots,N\right\}$ in time, until x_p^i fails to meet the system specifications at time t_{EOL}^i. The RUL PDF - i.e., the distribution $p(t_{EOL}^i - t_p)$ - is given by the distribution of w_p^i. Figure 4.4 shows the flow diagram of the prediction process.

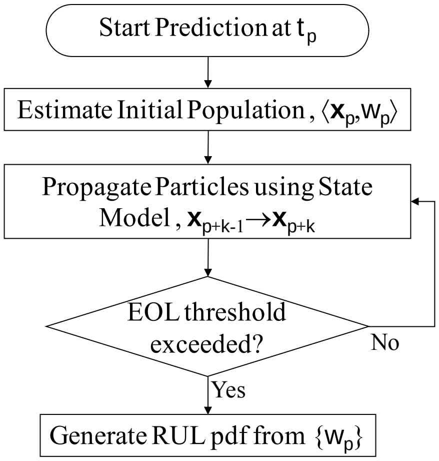
Start Prediction at t_p
Estimate Initial Population , $\langle \mathbf{x}_p, w_p \rangle$
Propagate Particles using State Model , $\mathbf{x}_{p+k-1} \rightarrow \mathbf{x}_{p+k}$
EOL threshold exceeded?
No
Yes
Generate RUL pdf from $\{w_p\}$

Fig. 4.4 Prediction flowchart.

4.7.2 Data-Driven Techniques

In some cases involving complex systems, it may be difficult or impossible to derive dynamic models based on all the physical processes involved. In such cases, it is possible to assume a certain form for the dynamic model and then use observed inputs and outputs of the system to determine the model parameters needed so that the model indeed serves as an accurate surrogate for the system. This is known as model identification.

For fault diagnosis and failure prognosis, a variety of input-output mappings have been employed as surrogate system models. Specifically, one may have historical fault/failure data in terms of time plots of various signals leading up to failure, or statistical data sets. In such cases, it is very difficult to determine any sort of model for prediction purposes. In such situations, one may use nonlinear network approximators that can be tuned using well-established formal algorithms to provide desired outputs directly in terms of the data. They provide structured nonlinear function mappings with very desirable properties between available data and desired outputs.

In prediction, artificial neural networks (ANNs), fuzzy systems, and other computational intelligence methods, based on the linguistic and reasoning abilities of humans, have provided an alternative tool for both forecasting researchers and practitioners [Sharda 1994]. Werbos [1988] reported that ANNs trained with the backpropagation algorithm outperform traditional statistical methods such as regression and Box-Jenkins approaches. In a forecasting competition organized by Weigend and Gerhenfeld [1993] through the Santa Fe Institute, all winners of each set of data used ANNs. Unlike the traditional model-based methods, ANNs are data-driven and self-adaptive, and they make very few assumptions about the models for problems under study. ANNs learn from examples and attempt to capture the subtle functional relationship among the data. Thus, ANNs are well suited for practical problems, for which it is easier to have data than knowledge governing the underlying system's fault behavior. Generally, they can be viewed as one of many multivariate nonlinear and nonparametric statistical methods [Chengand Titerington 1994]. Data-driven approaches to failure prognosis also take advantage of recurrent neural networks, dynamic wavelet neural networks, neuro-fuzzy systems, and a variety of statistical tools. For training and validation, they use the current and past history of input data and feedback outputs via unit delay lines. The main problem of ANNs is that their decisions are not always evident. Nevertheless, they provide a feasible tool for practical prediction problems.

4.7.3 Statistical Techniques

For situations in which sophisticated prognostic models are not or cannot be utilized (perhaps because a high investment in such models was not justified by the business case, low failure-criticality rates, or where there is an insufficient sensor network to assess condition), a statistical reliability or usage-based prognostic approach may be the only alternative. This form of prognostic algorithm is the least complex and requires that a history of component failure or operational usage profile data is available. One typical approach would be to fit a Weibull distribution (or other statistical failure distribution) to such failure or inspection data [Groer 2000; Schömig and Rose 2003]. Despite the obvious loss of information (compared to condition-based approaches), a statistical reliability-based prognostic distribution can still be used to drive interval-based maintenance practices which can then be potentially revised by the information obtained from maintenance. The benefit of a regularly updated maintenance database is critical for this approach.

4.8 Measuring Prognostics Performance

Chapter 9, IVHM Performance Metrics, lists a number of measures that can be used to evaluate prognostic performance. Some metrics of particular interest are listed here.

4.8.1 Prognostic Horizon

Prognostic Horizon (PH) fulfills two roles: first, it identifies whether an algorithm predicts within a specified error margin (specified by α, a statistical confidence parameter) around the actual End-of-Life (EOL, the time index for actual end of life, according to the defined failure threshold); and, second, it indicates how much time the algorithm provides for any corrective action to be taken. In other words, it assesses whether an algorithm yields a sufficient prognostic horizon; if it does not, it may not be useful or meaningful to compute other metrics. PH is defined as the difference between the EoP (End-of-Prediction) and the current time index *i*; it is calculated using data accumulated up to the time index *i* , provided that the prediction meets a minimum set of desired specifications. These specifications may be defined in terms of an allowable error bound (α) around true EOL. It is expected that PHs are determined for an algorithm-application pair offline during the validation phase. These numbers can then be used as guidelines when the algorithm is deployed in test applications when the actual EOL is not known in advance. While comparing algorithms, an algorithm with a longer prediction horizon *H* would be preferred. The prediction horizon is computed as $H = EoP - i$, where $i = min\left\{ j \mid \left(j \in l \right) \wedge \left(r_* \left(1-\alpha\right) \leq r^l \left(j \right) \leq r_* \left(1+\alpha\right) \right) \right\}$, $r_*^l\left(i\right)$ is the true RUL at time *i* given that data is available up to time *i* for the l^{th} UUT (unit under test), and $r^l(i)$ is the RUL estimate for the l^{th} UUT at time *i* as determined from measurement and analysis.

For instance, a PH with error bound of α = 0.05 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.

4.8.2 α-λ Performance

α-λ performance identifies whether the algorithm performs within desired error margins (specified by the parameter α) of the actual RUL at any given time instant (specified by the parameter λ) that may be of interest to a particular application. This presents a more stringent requirement of staying within a converging cone of error margin as a system nears EOL. The time instances may be specified as 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, we define α-λ accuracy as the prediction accuracy to be within $\alpha \cdot 100\%$ 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 Fig. 4.5. An extension of this metric based on other performance measures is straightforward:

$$\left[1-\alpha\right]r_*\left(t\right) \le r^l\left(t_\lambda\right) \le \left[1+\alpha\right]r_*\left(t\right) \tag{4.8}$$

where α is the accuracy modifier, λ is the time window modifier, $t_\lambda = P + \lambda\left(EOL - P\right)$, and P is the time index at which the first prediction is made by the prognostic system.

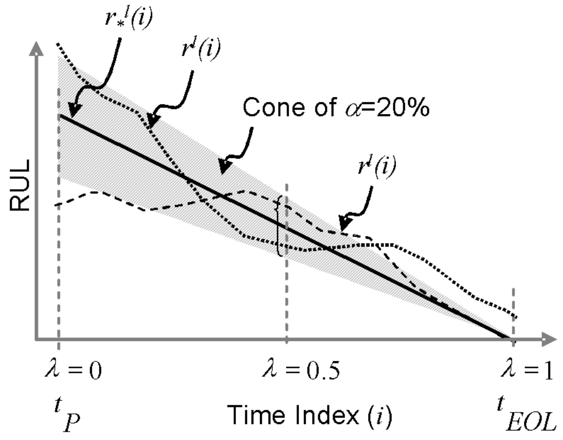

Fig. 4.5 α-λ performance visualization (Saxena et al., 2009)

Note that α-λ performance and prognostic horizon can also be computed as precision metrics.

4.8.3 Prognostic Dynamic Standard Deviation (DSTD)

Other performance measures intend to quantify the volatility of generated predictions, something that can be achieved by computing the standard deviation of the expected EOL over a sliding window:

$$DSTD = \varphi\left(\sqrt{Var\left(E\left\{EOL \mid y_{1:j}\right\}\right)_{j=k_{pred}-\Delta:k_{pred}}} \right)_{\forall k_{pred} \in [1,EOL]} \tag{4.9}$$

where k_{pred} is the cycle in which the prognostic algorithm is executed, Δ is the number of samples considered in the sliding window, φ is the logistic function that aims to scale the results in the range [0,1]. For this measure, the better DSTD, the closer to zero is the measure ($DSTD$ = 0 indicates perfect null volatility and the fact that new measurements do not alter the output of the prognostic algorithm).

4.8.4 Critical-α index

Decision-making support-systems cannot depend solely on information about the expectation of random variables, because the tails of PDFs contain critical information about the risk that is associated to process operation. The *critical-α* index is a measure based on the concept of the JITP (Just in Time Point) that helps to quantify this point. The *critical-α* index is a measure of risk aversion (a significant factor to be considered when using implementations that overestimate the remaining useful life of a system) and is defined as the maximum $\alpha \in [0,100]$ that guarantees that the JITP$_{\alpha\%}(k_{pred})$ value is smaller than the ground truth value of the EOL time instant, for all $k_{pred} \in [1 , \text{EOL}]$:

$$\alpha_{crit} = \arg\max_{\alpha} \left\{ JITP_{\alpha\%}(k_{pred}) \le EOL \right\}_{\forall k_{pred} \in [1, EOL]} \qquad (4.10)$$

Decision-making support systems that consider prognostic algorithms with larger critical-α values in their design are capable of implementing more aggressive strategies. This is based on the fact that these prognostic routines are conservative; thus it is possible to accept the risk of accumulating larger failure probability mass before recommending a corrective action. However, a large critical-α value is also an indicator that the variance of the predicted EOL PDF is large (i.e., less precise estimates of the EOL). For this reason, a good design should try to lessen this problem by selecting prognostic algorithms that allow not only the use of large critical-α values, but also minimize—over time—the difference between the ground truth EOL and the JITP values computed for the corresponding α_{crit}%.

4.9 Electro-Mechanical Actuator Case Study

4.9.1 Introduction

To illustrate the application of the principles outlined in this chapter, a detailed case study of an electro-mechanical actuator is presented. This case study describes the modeling and simulation steps, the experimental evaluation, and the development of fault diagnosis and failure prognosis algorithms for brushless DC motor winding insulation faults in the context of an aircraft EMA application. Electro-Mechanical Actuators (EMAs) are finding extensive utility as drives for modern aircraft systems, in addition to classical hydraulic devices. An EMA is configured as a closed-loop system consisting of a controller, motor(s), and sensing apparatus such as a resolver. Figure 4.6 depicts the main modules of a typical EMA, including the control loops.

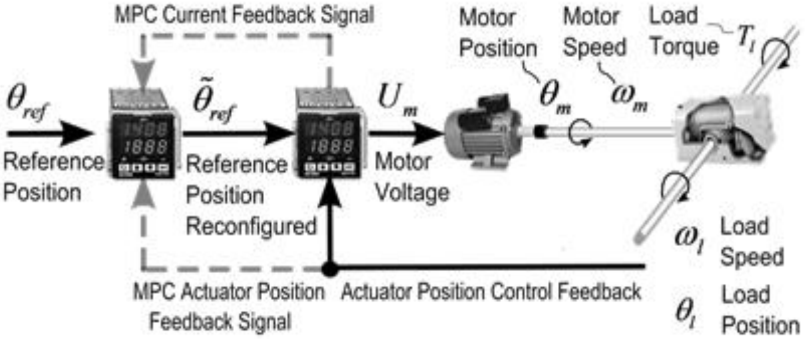

Fig. 4.6 An EMA configuration with associated control loops.

A thorough Failure Modes, Effects, and Criticality Analysis (FMECA) study identified turn-to-turn winding faults as the primary mechanism, or mode, of failure. Physics-of-failure mechanisms were used to develop a model for the identified fault. The model, implemented in Simulink, simulates the dynamics of the motor with a turn-to-turn insulation winding fault.

An experimental test procedure was devised and executed to validate the model. After a fault identification step (not reported here), a condition indicator extraction routine preprocesses monitoring parameters and passes the resulting condition indicators to a particle filter, as a model-based algorithm for fault diagnosis and failure prognosis.

This case study highlights the challenges faced by the IVHM designer in addressing the unique and complex issues posed by such systems. It points toward the necessity for rigorous tools and methods that are based on physics-of-failure mechanisms of such devices: modeling and simulation studies in conjunction with experimental data and a thorough approach to data processing, condition indicator extraction, and novel diagnostic and prognostic routines with guaranteed performance (i.e., customer-specified confidence level with given false alarm rate, detection accuracy, and prescribed RUL prediction). Figure 4.7 shows a flow chart of the motor RUL prediction approach.

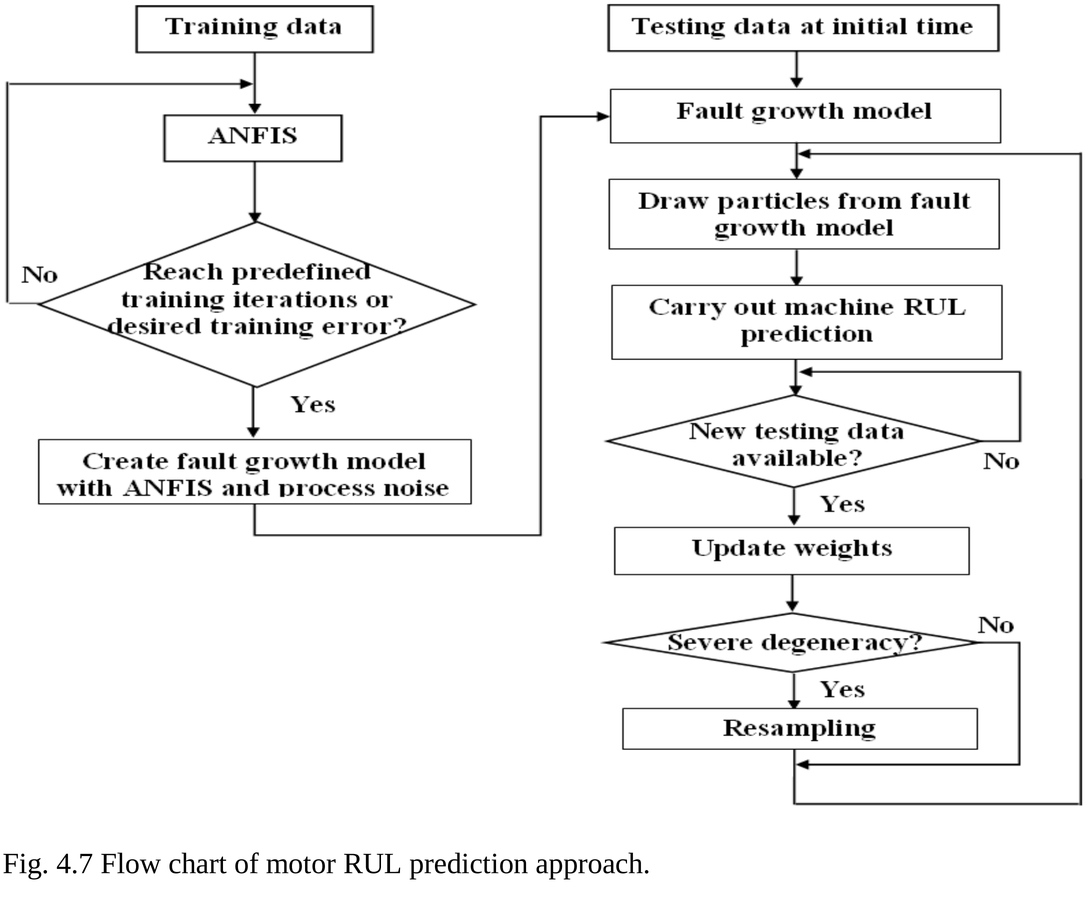
Training data
ANFIS
Reach predefined training iterations or desired training error?
No
Yes
Create fault growth model with ANFIS and process noise
Testing data at initial time
Fault growth model
Draw particles from fault growth model
Carry out machine RUL prediction
New testing data available?
No
Yes
Update weights
Severe degeneracy?
No
Yes
Resampling

Fig.

Fig. 4.7 Flow chart of motor RUL prediction approach.

4.9.2 Actuator Model

An EMA high-fidelity 5th order state-space model was developed to represent higher-order dynamics for a closed-loop actuator position controller [Brown et al. 2009]. The model, which can be expressed by the linear state-space system $\left(A_m, B_m, C_m\right)$ is employed to relate the control inputs and measured outputs of the actuator to the internal system states of the brushless DC motor:

$$\begin{aligned}\dot{\tilde{x}}_m &= A_m\tilde{x}_m + B_m u_m \\ \mathrm{y}_m &= C_m\tilde{x}_m\end{aligned} \tag{4.11}$$

where $\tilde{x}_{m0} = \tilde{x}_m(0)$. The internal state $\tilde{x}_m = \left[\tilde{i}_m\ \tilde{\theta}_m\ \tilde{\omega}_m\ \tilde{\theta}_l\ \tilde{\omega}_l\right]^T \in R^5$ is defined by the motor current, motor position and speed, and load position and speed; the control input $u_m = \left[\tilde{\theta}_{ref}\ T_{load}\right]^T \in R^2$ is defined by the reference position and external load disturbance; and the control output $y_m = \left[\theta_l\ i_m\right]^T \in R^2$ is defined by the load position and motor current.

The derivation of the motor dynamic equations begins by considering the physical layout of the stator windings and, using Kirchoff's second law, one obtains three differential equations, expressing the phase currents as functions of the motor voltages, the motor parameters, the winding inductances, and the mutual fluxes. The final dynamic model is expressed as follows:

$$\begin{bmatrix} \frac{di_{as}}{dt} \\ \frac{di_{bs}}{dt} \\ \frac{di_{cs}}{dt} \end{bmatrix} = \begin{bmatrix} L_{11}^* & L_{12}^* & L_{13}^* \\ L_{21}^* & L_{22}^* & L_{23}^* \\ L_{31}^* & L_{32}^* & L_{33}^* \end{bmatrix} \left\{ \begin{bmatrix} (U_m - \omega_r\psi_{am})\cos(\theta) \\ (U_m - \omega_r\psi_{bm})\cos(\theta - \frac{2\pi}{3}) \\ (U_m - \omega_r\psi_{cm})\cos(\theta + \frac{2\pi}{3}) \end{bmatrix} - \begin{bmatrix} r_{aa}i_{as} \\ r_{bb}i_{bs} \\ r_{cc}i_{cs} \end{bmatrix} \right\} \tag{4.12}$$

4.9.3 Failure Modes, Effects, and Criticality Analysis

An FMECA revealed that gradual deterioration of the insulation due to thermal, electrical, mechanical, and environmental stresses is the leading root cause for motor failures [Lee, Younsi, and Kliman 2005]. Insulation can break down due to many causes, including voltage surge, over temperature, shock, vibration, excessive moisture, contamination, and metallic dust. An FMECA study for the EMA under test was carried out to identify the primary failure mode of interest, turn-to-turn winding insulation faults for a DC brushless motor. The major findings of this study are listed in Table 4.1.

Table 4.1

FMECA for EMA

Item / Component	FAILURE RATE (HR^{-1})	Failure Mode Mechanism
Bearings	1.78E-05	Increased friction due to: - bearing wear - galling
Position Sensor	1.70E-05	Loss/incorrect feedback signal from resolver: - Turn-to-turn short - Turn-to-ground short - Open circuit
Brushless DC Motor	1.03E-05	Breakdown of stator assembly insulation due to: - Turn-to-turn short - Phase-to-phase short - Turn-to-ground short - Open circuit

Power Connector	6.64E-07	Electrical open due to: - Open pin - Open leadwire

To focus attention on the most critical actuator components, the criticality number and severity class of each failure mode and the major contributors to actuation system failure were identified. The severity class of a failure mode was assessed according to the definition of

I. A failure that would cause the actuator to flutter or disconnect and lead to mission loss

II. A failure that would cause the actuator to be held in a fixed position or be driven to a hardover condition

III. A failure that would result in marginal performance degradation

IV. A failure that has no effect on the system performance

According to MIL-STD-1629A *Procedure for Performing a Failure Mode, Effects and Criticality Analysis*, the critical number of a failure mode is defined as $C_m = \beta \alpha \lambda_p t$, where β is conditional probability for failure mode, α is failure mode ratio, λ_p is part failure rate, and t is duration of mission in hours.

Components with a high criticality number for the actuator include motor stator, motor resolver, bearing, and planetary gear head. They account for 95% of the criticality number sum. Furthermore, because motor stator and motor resolver are in severity class II, while bearing and planetary gear head are in severity class III, failures of motor stator and motor resolver will have a more severe impact on vehicle safe operation. Therefore, they were selected for the technology development and demonstration.

4.9.4 Component Phenomenology and System Performance Modeling

To evaluate the phenomenon of a motor winding short and the observed effects, a winding fault model was developed that represents the physics of the winding degradation. The schematic representing a winding fault for any single winding is shown in the left of Fig. 4.8, where *k* out of *N* winding turns experience a short, and the extent is modeled by a shorting resistor R_f. When R_f approaches zero, the *k* winding turns are completely shorted. When R_f approaches infinity, the *k* winding turns do not have any short. By using Thévenin circuit transformation, an equivalent circuit is obtained in which w^f(between 0 and 1) represents the dimension of this turn-to-turn winding fault. That is, $w^f = 1$ represents no winding short at all while $w^f = 0$ represents an entire winding that is completely short. By applying this notion to all three windings, the modified three-phase wye-connected electrical diagram for a brushless DC motor, which is consistent with the test actuator, is shown in the right of Fig. 4.8.

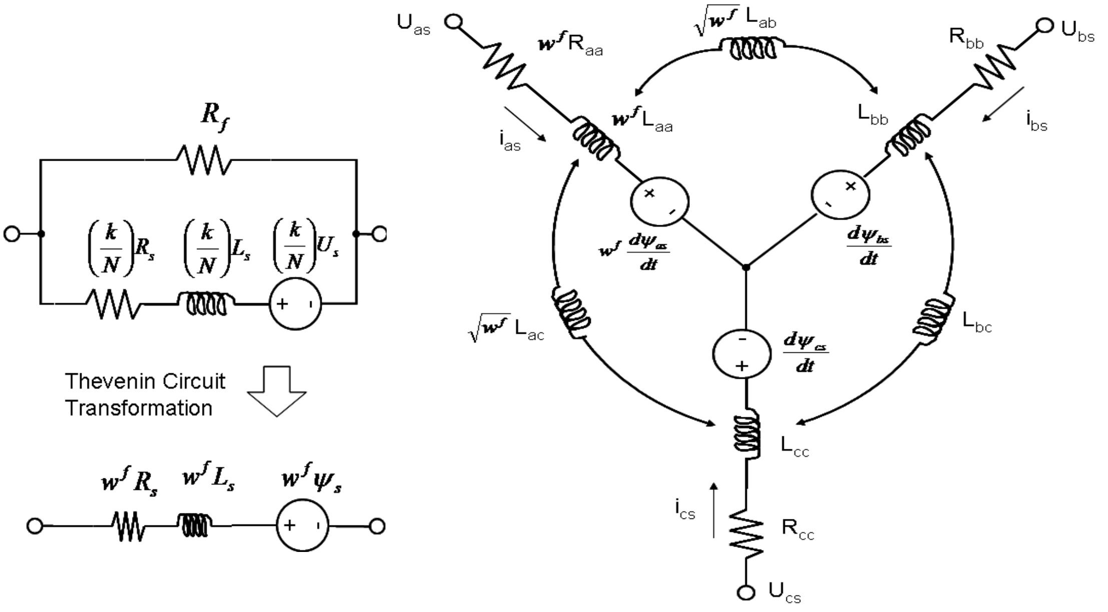

Fig. 4.8 Turn-to-turn winding fault model.

The modeling of a turn-to-turn fault caused by wearing of winding insulation within a single phase of the motor results in a connection between individual windings, as illustrated in the left of Fig. 4.9.

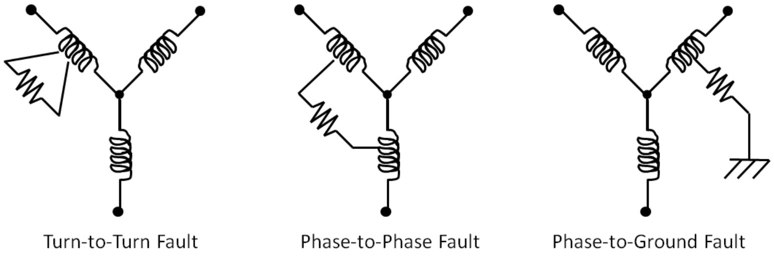

Fig. 4.9 Motor winding fault modes.

A schematic representing a winding fault for any single winding is provided in Fig. 4.10. The symbols L_s, R_s, and U_s represent the total winding inductance, resistance, and back-emf (electromotive force) voltage of the winding during normal conditions (R_f is open). The symbols N, k, and R_f represent the number of total winding turns, number of winding turns between the fault, and the resistance of the insulation fault, respectively.

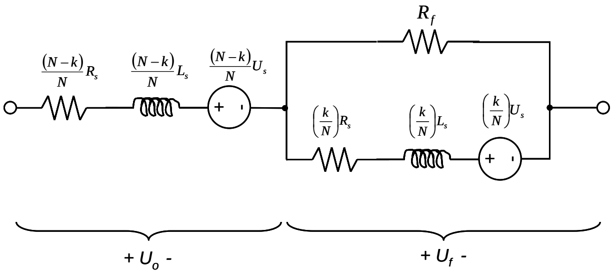
R_f
$\frac{(N-k)}{N}R_s$
$\frac{(N-k)}{N}L_s$
$\frac{(N-k)}{N}U_s$
$\left(\frac{k}{N}\right)R_s$
$\left(\frac{k}{N}\right)L_s$
$\left(\frac{k}{N}\right)U_s$
+
+
+ U_o -
+ U_f -

Fig. 4.10 Schematic of insulation fault model.

The circuit network on the left side can be reduced to a single resistor, inductor, and voltage source, as illustrated in Fig. 4.11, by again applying the Thévenin circuit transformation.

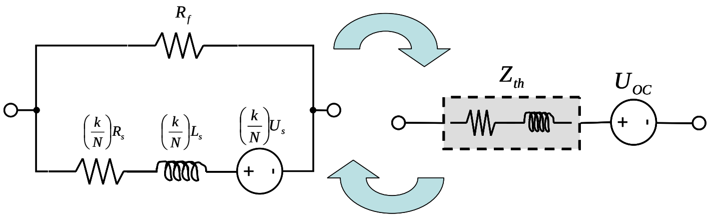
R_f
$\left(\frac{k}{N}\right)R_s$
$\left(\frac{k}{N}\right)L_s$
$\left(\frac{k}{N}\right)U_s$
Z_{th}
U_{OC}
+
+

Fig. 4.11 Thévenin circuit transformation of the winding fault model.

4.9.5 Data Processing/Condition Indicator Extraction and Selection

Condition indicators computed from the preprocessed data are used to develop indications of a fault condition and operating conditions of the DC motor. Four condition indicators are extracted from the filtered voltage and current matrices. The voltage condition indicators *f₁* and *f₂* are related to the operating speed of the motor and are expressed as

$$\begin{gathered} f_1 = \sum_{n=1}^{3} \frac{1}{3} RMS\left(U_{abc}^{1}\left(n\right)\right) \\ f_2 = \sum_{n=1}^{3} \frac{1}{3} RMS\left(I_{abc}^{1}\left(n\right)\right) \end{gathered} \tag{4.13}$$

where U_{abc}^{1} and I_{abc}^{1} are, respectively, obtained by filtering terminal-to-terminal voltage and current matrices around the frequency that defines the motor operating speed. Similarly, the current condition indicators f_3 and f_4 are computed as

$$\begin{gathered} f_3 = \sum_{n=1}^{3} \frac{1}{3} RMS\left(U_{abc}^{2}\left(n\right)\right) \\ f_4 = \sum_{n=1}^{3} \frac{1}{3} RMS\left(I_{abc}^{2}\left(n\right)\right) \end{gathered} \tag{4.14}$$

where U_{abc}^{2} and I_{abc}^{2} are, respectively, obtained by filtering terminal-to-terminal voltage and current matrices around the natural lag frequency of the resolver. The principle current feature is related to the asymmetry of the motor windings.

The final step is to map the computed condition indicators to the known fault values inserted during the seeded fault test. A neural network fitting tool in MATLAB is used to map condition indicators into fault dimensions. The neural network was trained appropriately and used for this mapping.

4.9.6 Motor Winding Short Model

Depending on where the short occurs at the stator's three windings, there are three winding fault modes, as shown in Fig. 4.9. Turn-to-turn winding fault is when the short occurs across one winding. Phase-to-phase winding fault is when the short occurs between two windings. Phase-to-ground winding fault is when the short occurs between the winding and case. Because the insulation between windings, stator core, and case is normally well protected with a polymer and polymer/mica combination, while turn-to-turn insulation can only be applied with a thin film of polyamide on the surface of the magnet wire, turn-to-turn winding fault is the most prevalent winding fault mode and therefore was selected to be the focus of our program.

The schematic representing a turn-to-turn winding fault is shown in the left of Fig. 4.9, where *k* out of *N* winding turns experience a short, and the extent of the short is modeled by a shorting resistor R_f. The symbols L_s, R_s, and U_s represent the total winding inductance, resistance, and back-emf voltage of the winding during normal conditions (i.e., $R_f = \infty$). The equivalent circuit that represents the reduced effectiveness of the winding is

$$w^{f} = 1 - \frac{k}{N}\left(1 - \left[1 + \frac{k}{N}\left(\frac{R_s}{R_f}\right)\right]^{-2}\right) \tag{4.15}$$

Note that w^f always takes a value between 0 and 1. When $k = 0, w^f = 1$ and there is no winding short at all. When $k = N \; and \; R_f = 0, w^f = 0$ and the entire winding is completely shorted.

4.9.7 Prognosis

The nonlinear dynamic state model [Brown et al. 2009; Zhang, Sconyers, Byington, Patrick, Orchard, and Vachtsevanos 2008] for use with the particle filter is

$$\begin{cases} x_d(t+1) = f_b(x_d(t), \mathrm{n}(t)) \\ x_c(t+1) = f_t(x_d(t), x_c(t), w(t)) \\ f_p(t) = h_t(x_d(t), x_c(t), v(t)) \end{cases} \tag{4.16}$$

where f_b, f_t, and h_t are non-linear mappings, x_d is a collection of Boolean states associated with the presence of a particular operating condition in the system (normal operation, fault type #1, #2, etc.), x_c is a set of continuous-valued states that describe the evolution of the system given those operating conditions, f_p is a feature measurement, w and v are non-Gaussian distributions that characterize the process and feature noise signals, respectively. The function h_t is a mapping between the feature value $f_p(t)$ and the fault state $x_c(t)$. In the case of a turn-to-turn winding insulation short circuit fault, the values of $f_p(t)$ and $L(t)$ are related by the operating parameters, motor speed ω_m and motor current i_m. For simplicity, $n(t)$ may be assumed to be zero-mean i.i.d. (independent and identically distributed) uniform white noise. At any given instant of time, this framework provides an estimate of the probability masses associated with each fault mode, as well as a PDF estimate for meaningful physical variables in the system. PDF estimates for the system continuous-valued states (computed at the moment of fault detection) may be used as initial conditions in failure prognostic routines. As a result, a swift transition between the two modules (fault detection and prognosis) may be performed and reliable prognosis can be achieved within a few cycles of operation after the fault is declared. This characteristic is one of the main advantages of the particle-filter-based framework.

The fault model employed for fault diagnosis and failure prognosis is described next.

Historically, the rate of degradation is dependent on temperature:

$$L(k+1) = C_0 \exp\left(-\frac{E_a}{k_b T_{wa}(k)} \right) \text{ where } C_0 > 0 \text{ (Arrhenius law)} \tag{4.17}$$

The relationship between winding current and temperature is expressed as

$$T_{wa}(k+1) = \left[\frac{1}{R_{wa} C_{wa}} \right] \left[\tau^2 R_0 R_{wa} - T_{wa}(k) \right] \tag{4.18}$$

T_{wa} = Difference between winding and ambient temperatures [°K]

R_{wa} = Thermal resistance [W/°K]

C_{wa} = Thermal conductance [°K/W]

I_m = Motor current [A]

R_0= Winding resistance [Ω]

E_a = Activation energy [J]

k_b = Boltzman's constant [J/°K]

C_0 = Constant of proportionality

L = Fault dimension (winding degradation)

τ = Thermal time constant

An adaptation law is used to identify the growth model. The winding resistance depends on the temperature but changes slowly with time, while thermal measurements, T_{wa}, are used to update the model.

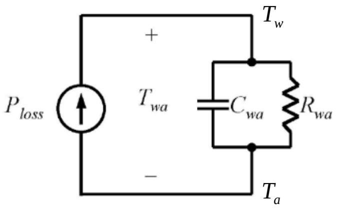

Fig. 4.12 Simplified fault model.

A discrete-time linear approximate model is written as

$$T_{wa}\left(k+1\right)=\hat{a}_m T_{wa}\left(k\right)+\hat{b}_m u\left(k\right)\text{, where } u=\sum_{n=1}^{3} i_n^2\left(k\right)$$

The adaptation law is then

$$\hat{a}_m\left(k+1\right)=\gamma_a T_{wa}\left(k\right)e\left(k\right)$$

$$\hat{b}_m\left(k+1\right)=\begin{cases}\gamma_b\ \mathrm{u}\left(k\right)e\left(k\right) & if\ \hat{b}_m\left(k\right)>\bar{b}\\ \gamma_b u\left(k\right)e\left(k\right)+\dfrac{\bar{b}-\hat{b}_m}{\hat{b}_m-\bar{b}+\varepsilon} & if\ \hat{b}_m\left(k\right)<\bar{b}\end{cases}$$

$$e\left(k\right)=y\left(k\right)-T_{wa}\left(k\right)$$

where $\begin{cases}\bar{b},\gamma_a,\gamma_b,\varepsilon>0\\ a_m\left(0\right)=a_{m0}\\ b_m\left(0\right)=b_{m0}\end{cases}$

As a result, T_{wa} can be computed as

$$\begin{cases}T_{wa}\left(k+1\right)=\hat{a}_m T_{wa}\left(k\right)+\hat{b}_m u\left(k\right)\\ L\left(k+1\right)=L\left(k\right)+\beta\left(k\right)exp\left[-\dfrac{E_a}{k_b T_{wa}}\right]+\omega_1\left(k\right)\\ \beta\left(k+1\right)=\beta\left(k\right)+\omega_2\left(k\right)\end{cases} \tag{4.19}$$

$$Feature(k) = h(L(k)) + v(k)$$

where β is an adaptive parameter, $\omega_{1,2}$ is the process noise, v is the feature noise, and $h(t)$ is the "fault-to-feature" mapping. Condition indicators are extracted in the frequency domain, and a neural network based classifier maps the selected/extracted condition indicators into a fault dimension.

Prognosis is achieved by implementing the particle-filtering framework that was described in Section 4.7.1. Long-term predictions are used to estimate the probability of failure in a system given a hazard zone that is defined via a probability density function with lower and upper bounds for the domain of the random variable, denoted as H_{lb} and H_{up}, respectively. The probability of failure at any future time instant is estimated by combining both the weights $w^{(i)}_{t+k}$ of predicted trajectories and specifications for the hazard zone through the application of the Law of Total Probabilities; see Fig. 4.13**Error! Reference source not found.**.

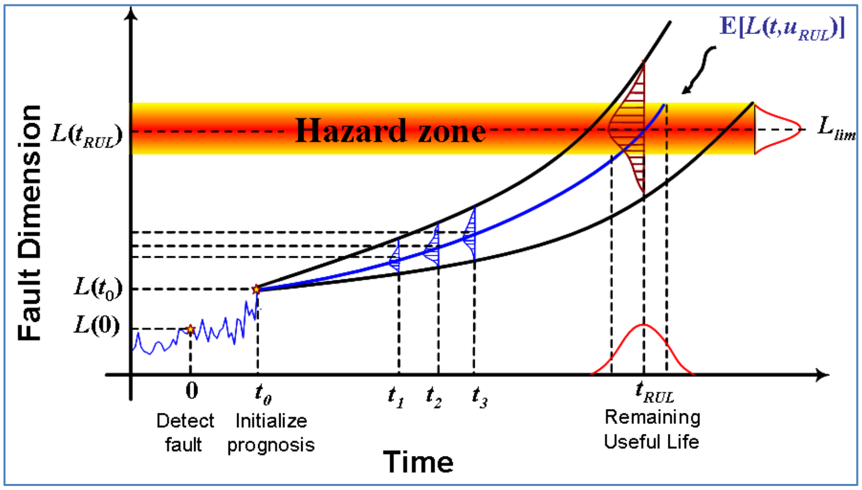

Fig. 4.13 Typical prediction profiles and EOL estimate.

The resulting RUL PDF, in which t_{RUL} refers to RUL, provides the basis for the generation of confidence intervals and expectations for prognosis:

$$\hat{p}_{t_{RUL}} = \sum_{i=i}^{n} p\left(Failure \middle| X = \hat{x}^{(i)}_{t_{RUL}}, H_{lb}, H_{up}\right) \cdot w^{(i)}_{RUL} \tag{4.20}$$

The final step in the motor winding case study involved the development and implementation of a systematic methodology aimed to predict accurately the RUL of the failing motor insulation. The approach takes advantage of a Bayesian estimation technique using an mth-order hidden Markov model, an adaptive neuro-fuzzy inference system (ANFIS) predictor with the process noise, as the fault growth model, integrated with a high-order particle filter. Of course, other techniques may be called upon to address the prediction problem. Figure 4.14 is an illustration of the implemented prediction algorithm showing results of the estimated time to failure at *t*=120 min.

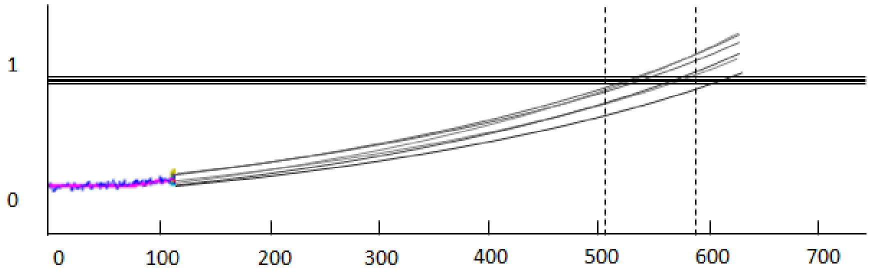

Fig. 4.14 The implementation of the predictive algorithm.

4.9.8 Experimental Evaluation

A scaled version of the EMA was experimentally evaluated in a “proof-of-concept” test setup to evaluate the prognostic methodology. The actuator test setup consists of two commercial off-the-shelf motors; one motor simulates the actuator and the other motor provides dynamic loading.

The actuator and dynamic loader are tied directly back-to-back using metal bellow couplings with a high-precision torque sensor. Both motors are controlled using a digital-space-vector pulse width modulation (PWM) sinusoidal drive. This setup allows the motor under investigation to operate under dynamic loading conditions for both opposing and aiding loads, as shown in Fig. 4.15.

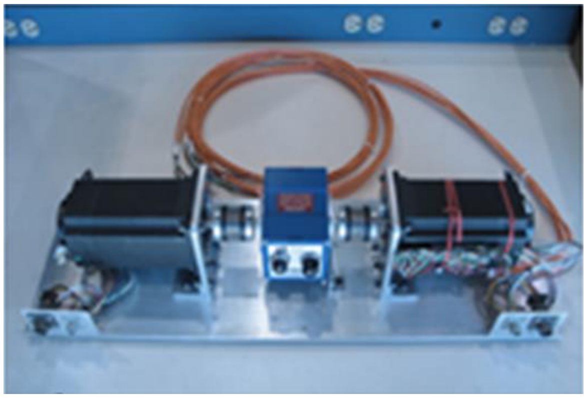

Fig. 4.15 Experimental setup.

Experimental results showed that the proposed failure prognosis architecture enabled early detection of the fault condition under analysis by anticipating failure of the system with a prognostic horizon of more than seven hours, thus providing sufficient time for corrective actions. Conceptually, the performance would be tracked as shown in Fig. 4.15. Here, the first prediction point is shown as a full circle at t=120 min. As can be seen, the prediction is inside the α-cone of α=0.2. Further predictions would be displayed as represented by the empty circles.

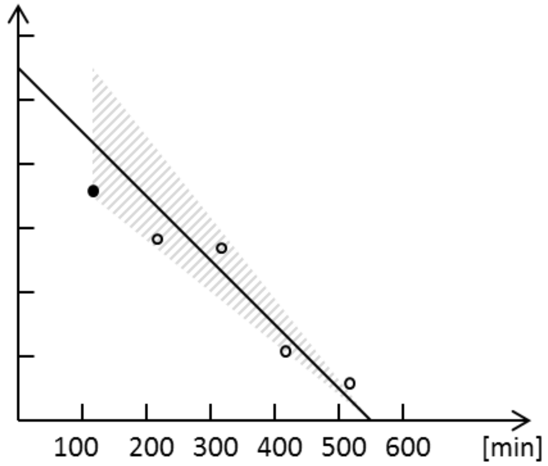

Fig. 4.15 Performance evaluation with α=0.2.

4.10 Conclusions and Recommendations

As the complexity of modern aerospace and industrial systems increases, the need for improved system reliability and autonomy follows an increased trend. Technologists attempting to develop and introduce IVHM methods for such complex systems must deal with the issues exposed in this chapter through new and novel system engineering concepts that integrate facets of modeling, testing, analysis, and algorithm development.

Prognosis is one of the more challenging aspects of the modern IVHM system. It also has the potential to be the most beneficial in terms of reduced operational and support (O&S) cost, and life-cycle total ownership cost (TOC) of many types of critical systems. It also ensures safe operation because the state of health of future (and, as a by-product, current) operation is continuously estimated. The evolution of diagnostic monitoring systems has led to the recognition that predictive prognosis is both desired and technically possible. A coherent and rigorous approach to develop critical health management technologies with emphasis on prognostics was introduced in this chapter, and the application of these innovative technologies was demonstrated in a typical case study. The process starts with the identification of critical components, followed by the physics-based modeling and simulation that were validated using real data, continued with the diagnostic and prognostic algorithm development, and ended with testing and validation using accepted performance metrics.

The case study introduced here exemplifies a formal and theoretically rigorous approach to the development and implementation of IVHM concepts to complex systems. A novel physical modeling and data-driven methodology via experimental testing is required to provide explanations for device behaviors and facilitate the design of fault monitoring instrumentation as well as the development of suitable diagnostic and prognostic algorithms. It is the synergy of model-based and data-driven methodologies based on an understanding of the physics of failure mechanisms for critical assets that will produce accurate, precise, and robust results that may assist the operator/maintainer to implement IVHM practices. It is worth noting the importance of methods and tools that possess the capability to address uncertainty—how to represent it and how to manage it, the Achilles’ heel of IVHM.

References

Aha, D. 1997. Special Issue on Lazy Learning. *Artificial Intelligence Review, vol. 11(1-5)*, 1–6.

Arulampalam, S., L.S. Maskel, N. Gordon, and T. Clapp. 2002. “A Tutorial on Particle Filters for On-line Non-linear/Non-Gaussian Bayesian Tracking,” *IEEE Trans on Signal Processing 50(2)*, 174–188.

Brown, D., G. Georgoulas, H. Bae, R. Chen, Y. Ho, and G. Tannenbaum et al. 2009. “Particle filter based anomaly detection for aircraft actuatoar systems,” *IEEE Aerospace.*

Engel, S., B. Gilmartin, K. Bongort, and A. Hess. 2000. “Prognostics, the Real Issues Involved with Predicting Life Remaining,” *IEEE Aerospace*, pp. 457–469, Big Sky, MT.

Frelicot, C. 1996. “A fuzzy-based prognostic adaptive system,” *RAIRO-APII-JESA, Journal Europeen des Systemes Automatises, Vol.30, No.2–3* , 281–99.

Groer, P. 2000. “Analysis of Time to Failure with a Weibull Mode,” *Maintenance and Reliability Conference, MARCON 2000.*

Jardim-Gonçalves, R., M. Martins-Barata, J. Assis-Lopes, and A. Steiger-Garcao. 1996. “Application of stochastic modelling to support predictive maintenance for industrial environments," *IEEE International Conference on Systems, Man, and Cybernetics*, pp. 117–122.

Kumar, S., N. Vichare, E. Dolev, and M. Pecht. 2012. "A Health Indicator Method for Degradation Detection of Electronic Products," *Microelectronics Reliability, Vol. 52(2)*, 439–445.

Lee, S., K. Younsi, and G. Kliman. 2005. "An Online Technique for Monitoring the Insulation Condition of AC Machine Stator Windings Energy Conversion," *IEEE Transactions on Energy Conversion, Volume 20(4)* , 737–745.

Lewis, F. 1986. *Optimal Estimation: With an Introduction to Stochastic Control Theory.*

Ljung, L. 1999. *System Identification: Theory for the User, 2nd ed.,* New Jersey: Prentice-Hall.

Orchard, M. and G. Vachtsevanos. 2007. "A Particle Filtering Approach for On-Line Failure Prognosis in a Planetary Carrier Plate," *International Journal of Fuzzy Logic and Intelligent Systems, Vol. 7, No. 4* , 221–227.

Orchard, M. and G. Vachtsevanos. 2009. "A particle filtering approach for on-line fault diagnosis and failure prognosis," *Transactions of the Institute of Measurement and Control, Vol. 31,No. 3–4* , 221–246.

Pham, H. and B. Yang. 2010. “Estimation and forecasting of machine health condition using ARMA/GARCH model,” *Mechanical Systems and Signal Processing*, 546–558.

Saxena, A., J. Celaya, B. Saha, S. Saha, and K. Goebel. 2009. “Evaluating Algorithm Performance Metrics Tailored for Prognostics,” *IEEE Aerospace Conference,* Big Sky, MT.

Saxena, A., J. Celaya, B. Saha, S. Saha, and K. Goebel. 2009. “On Applying the Prognostic performance Metrics,” *Annual Conference of the Prognostics and Health Management Society (PHM09),* San Diego, CA.

Schömig, A. and O. Rose. 2003. “On the Suitability of the Weibull Distribution for the Approximation of Machine Failures,” *Proceedings of the 2003 Industrial Engineering Research Conference*, Portland, OR.

Studer, L. and F. Masulli. 1996. “On the structure of a neuro-fuzzy system to forecast chaotic time series,” *International Symposium on Neuro-Fuzzy Systems,* pp. 103–110.

Tangirala, R. 1996. “A nonlinear stochastic model of fatigue crack length for on-line damage sensing,” *Decision and Control Conference.*

Zhang, B., C. Sconyers, C. Byington, R. Patrick, M. Orchard, and G. Vachtsevanos. 2008. “Anomaly Detection: A Robust Approach to Detection of Unanticipated Faults.” *IEEE Conference on Prognostics and Health Management*, Denver, CO.