Reliability Engineering and System Safety 133 (2015) 223–236

Contents lists available at ScienceDirect

Reliability Engineering and System Safety

journal homepage: www.elsevier.com/locate/ress

Review

Practical options for selecting data-driven or physics-based prognostics algorithms with reviews

Dawn An[a,b,1], Nam H. Kim[a,2], Joo-Ho Choi[b,*]

[a] Dept. of Mechanical & Aerospace Engineering, University of Florida, Gainesville, FL 32611, USA
[b] Dept. of Aerospace & Mechanical Engineering, Korea Aerospace University, Goyang-si, Gyeonggi-do 412-791, Republic of Korea

ARTICLE INFO

Article history:
Received 20 January 2014
Received in revised form
31 August 2014
Accepted 2 September 2014
Available online 21 September 2014

Keywords:
Data-driven prognostics
Physics-based prognostics
Neural network
Gaussian process regression
Particle filter
Bayesian inference

ABSTRACT

This paper is to provide practical options for prognostics so that beginners can select appropriate methods for their fields of application. To achieve this goal, several popular algorithms are first reviewed in the data-driven and physics-based prognostics methods. Each algorithm's attributes and pros and cons are analyzed in terms of model definition, model parameter estimation and ability to handle noise and bias in data. Fatigue crack growth examples are then used to illustrate the characteristics of different algorithms. In order to suggest a suitable algorithm, several studies are made based on the number of data sets, the level of noise and bias, availability of loading and physical models, and complexity of the damage growth behavior. Based on the study, it is concluded that the Gaussian process is easy and fast to implement, but works well only when the covariance function is properly defined. The neural network has the advantage in the case of large noise and complex models but only with many training data sets. The particle filter and Bayesian method are superior to the former methods because they are less affected by noise and model complexity, but work only when physical model and loading conditions are available.

Contents

[*] Corresponding author. Tel.: +82 2 300 0117; fax: +82 2 3158 2191.
E-mail addresses: dawnan@ufl.edu (D. An), nkim@ufl.edu (N.H. Kim), jhchoi@kau.ac.kr (J.-H. Choi).
[1] Tel.: +1 352 870 4774; fax: +1 352 392 7303.
[2] Tel.: +1 352 575 0665; fax: +1 352 392 7303.

http://dx.doi.org/10.1016/j.ress.2014.09.014
0951-8320/© 2014 Elsevier Ltd. All rights reserved.

1. Introduction

Condition-based maintenance (CBM), as illustrated in Fig. 1, is a cost-effective maintenance strategy in which maintenance schedules are predicted based on the results provided from diagnostics and prognostics. Diagnostics characterizes the status of damage through detection, isolation and identification using collected data from structural health monitoring. Based on the diagnostics information, prognostics predicts the future behavior of damage and the remaining useful life (RUL), which is the remaining time/cycles before requiring maintenance. The collected data that are important for both diagnostics and prognostics can be classified into two types; (1) event data or run-to-failure data and (2) condition monitoring (CM) data, which provide information on damage, such as wear volume, crack size, vibration signal, oil debris and thermography, to name a few. Some CM data are direct, such as crack size, while others are indirect, such as oil debris. In the case of indirect CM data, a damage quantification process is required for diagnostics and prognostics.

There are several reviews on diagnostics methods [1–3] as well as a large number of publications on diagnostics [4–8], signal processing and damage quantification [9,10]. For example, Jardine et al. [1] provided an overall review on diagnostics and prognostics in terms of implementing CBM and decision making. There are also other references for CBM [11,12]. On the other hand, a relatively small number of reviews on prognostics are available [13–17]. Most of them provide a simple comparison of different methods for a specific application only. In addition, most review papers focus on an algorithm itself, rather than interpreting the algorithms in the context of prognostics. Therefore, the goal of this paper is to provide a practical review of prognostic methods so that beginners can select appropriate methods for their fields of application. To achieve this goal, this paper introduces not only prognostics algorithms, but also their attributes and pros and cons in terms of model definition, model parameter estimation and ability to handle noise and bias in data.

In general, prognostics methods can be categorized into data-driven, physics-based and hybrid approaches, as illustrated in Fig. 2. Data-driven approaches use information from previously collected data (training data) to identify the characteristics of the currently measured damage state and to predict the future trend. Physics-based approaches assume that a physical model describing the behavior of damage is available and combine the physical model with measured data to identify model parameters and to predict future behavior. There are two main differences between data-driven and physics-based approaches; (1) availability of a physical model and (2) use of training data to identify the characteristics of the damage state. Hybrid approaches combine the abovementioned methods to improve the prediction performance [18–20]. The last approach, however, is not mature yet and will not be discussed in this paper.

Data-driven and physics-based approaches have different properties that contribute to the preference of each algorithm. Providing a standard to select the best algorithm for different conditions is important for users. In this paper, therefore, the following conditions will be considered: the number of data sets, the level of noise and bias in data, availability of loading conditions and physical models and model complexity. Typical prognostics algorithms are tested under such conditions to show the influence of each algorithm's attributes as well as the pros and cons of the results. A fatigue crack growth example is used for this purpose because a physical model is available, which enables comparison between different methods in prognostics. However, the conclusions in this paper are not limited to the crack growth problem and can be interpreted in a general context.

The paper is organized as follows: in Sections 2 and 3, reviews on the data-driven and physics-based approaches are presented, respectively. In Section 4, several case studies are presented, in which the attributes of different prognostics methods are analyzed in order to help select an appropriate method, followed by conclusions in Section 5.

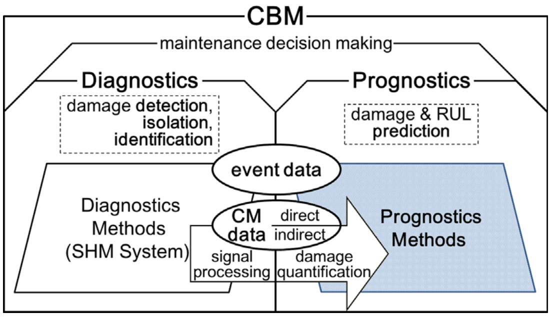


Fig. 1. Building-block of condition-based maintenance.

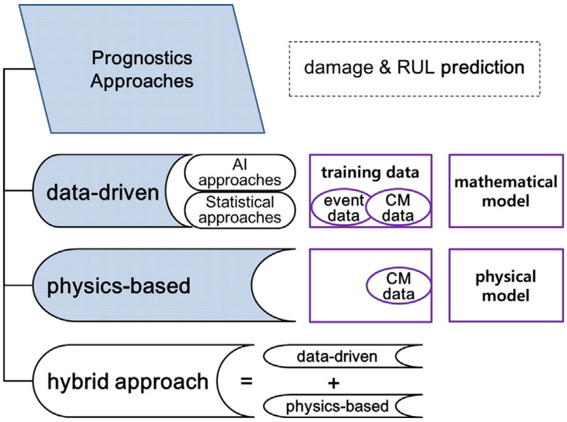


Fig. 2. Categorization and definition of prognostics methods.

2. Reviews on data-driven approaches

Data-driven approaches use information from observed data to identify the characteristics of damage progress and predict the future state without using any particular physical model. Instead, mathematical models and/or weight parameters are employed, which are determined based on the training data that are obtained under various usage conditions. Since the data-driven approaches depend on the trend of data, which often show a distinct characteristic near the end of life, they are powerful in predicting near-future behaviors, especially toward the end of life.

The data-driven approaches are generally divided into two categories: (1) the artificial intelligence approaches that include neural network (NN) [21–23] and fuzzy logic [24,25]; and (2) the statistical approaches that include the Gaussian process (GP) regression [26,27], relevance/support vector machine [28,29], least squares regression [30], the gamma process [31], the Wiener processes [32], hidden Markov model [33], etc. Among these algorithms, NN and GP are most commonly used for prognostics, and thus, will be discussed in this paper.

2.1. Neural network (NN)

The Neural Network algorithm is a representative data-driven method in which a network model learns a way to produce a desired output, such as the level of degradation or lifespan, by reacting to given inputs, such as time and usage conditions. Once the network model learns enough about the relationship between inputs and output, it can be used for the purpose of diagnosis and prognosis. A typical architecture, feed-forward neural network (FFNN) [34], is illustrated in Fig. 3, in which circles represent nodes (also called a neuron or unit) and each set of nodes in the same column is called a layer. The nodes in the input and output layer represent the input and output variables, respectively. The number of nodes in the hidden layer should be determined to properly express the mechanism between the input and output, by receiving signals from the input layer and forwarding them to the output layer. The learning process is equivalent to determining weight parameters such that the network model accurately represents the relationship between inputs and outputs. Once the network model learns enough, the model is functionalized using transfer functions and weight parameters. The transfer functions characterize the relationship between two adjacent layers, where several types of transfer functions are available, such as the sigmoid, inverse and linear functions [35]. The weight parameters include weights and biases [36,37]. The weights, shown as rectangles in Fig. 3, are usually multiplied with the value at the previous nodes, and then the biases shown as ellipses are added to the sum of the results to be an input to the transfer function. The process of finding the optimal weight parameters is called training, or learning, and requires many sets of training data.

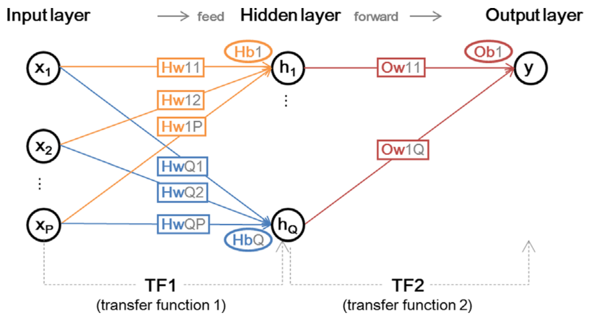


Fig. 3. Illustration of a typical network model: feed-forward neural network.

The FFNN is often called the back-propagation neural network (BPNN) because the weight parameters are determined through a learning/optimization algorithm by the backward propagation of errors between the training data and the output from the network model [38]. In other words, FFNN and BPNN are used, respectively, to calculate the response forward and to update the weight parameters backward, based on the response. Recurrent NN (RNN) [39] is the second most common architecture; its concept is not much different from the FFNN, except that it has local feedback connections between the input and hidden/output nodes. Since the Levenberg–Marquardt back-propagation algorithm [40] is usually employed for the RNN's learning algorithm, RNN and Levenberg–Marquardt neural networks are sometimes used interchangeably. In addition to the FFNN and RNN, there are other neural networks, such as fuzzy-neural [41], wavelet [42], associative-memory [43], modular [44] and hybrid [45,46] neural networks.

In the following, three important issues are discussed for NN-based prognostics. It is important that users are aware of these issues in order to use the algorithm properly.

2.1.1. Issue 1: Network model definition (the number of nodes and layers)

The first issue is the definition of the network model, which includes selecting the number of hidden nodes, hidden layers and input nodes. Although there is no general selection procedure for the number of hidden nodes, Lawrence et al. [47] investigated the usage of mean square errors in order to find the optimal number of hidden nodes. Gómez et al. [48] used the idea of measuring complexity of function to determine the number of nodes and showed that the results were close to the optimum. More studies on number of hidden nodes are summarized in Ref. [49]. While one or two hidden layers are commonly used, Ostafe [50] presented a method using pattern recognition to determine the number of hidden layers.

The problem of determining the number of input nodes always exists in the data-driven approach since all available information, such as time, loading conditions, and damage data can be considered as inputs. It is natural to use damage data as inputs, but Chakraborty et al. [21] concluded that it was unclear how many past values should be used as inputs. Chang and Hsieh [51] explored the optimal number of input nodes using particle swarm optimization. In conclusion, there is no universal procedure to establish a proper NN model, which may be difficult for new users without much experience. In recent years, there have been efforts to find proper model/method for each application rather than a study of selecting number of nodes and layers [52–54].

2.1.2. Issue 2: Finding optimal parameters (weights and biases)

Once a network model is defined, the next issue is to find the weight parameters that are related with the model using a learning/optimization algorithm. In NN, no matter how complex the relationship between the input and output layer is, it is possible to express the relationship by augmenting the number of hidden layers and hidden nodes. However, more complex NN model ends up with more unknown parameters, which requires more training data. When the BPNN algorithm is used especially, the following problems occur: (1) the global optimum is extremely difficult to find in the case of many parameters and (2) the convergence rate is very low and depends on the initial estimates. For these reasons, there have been many efforts to improve the drawbacks of the BPNN algorithm, such as a dynamic self-adaptation algorithm [55] a simulated annealing algorithm [56] combined genetic and differential evolution algorithm [57], and a technique combining the conjugate gradient optimization algorithm with the BPNN algorithm [58]. There are many ensemble techniques to improve the performance of algorithms [59–64], as well as other efforts in Refs. [40,55,58].

While the aforementioned methods are to find the weight parameters in a deterministic sense, there have been probabilistic approaches based on the Bayesian learning technique [65,66], in which the weight parameters are obtained by using a sampling method. Even though probabilistic approaches can resolve the local optimal problem, the sampling error grows as the number of weight parameters increases. Therefore, finding optimal weight parameters is still challenging, and the performance of the NN algorithm quickly deteriorates with local optimal weight parameters.

2.1.3. Issue 3: Uncertainty in data and optimization process

Last but not the least, uncertainty caused by bias and noise in the data is an important issue in the NN, as it has a significant effect on the solution of weight parameters. The bias here is different from the bias in weight parameters in NN; here, the bias in the data is the measurement error caused by sensors, such as a calibration error. Unfortunately, this bias cannot be handled with data-driven approaches because there are no parameters related to it, which is one of the drawbacks of this approach.

In the case of noise, it is common to provide confidence bounds based on nonlinear regression and/or the error between NN outputs and training data [67–70]. Bootstrapping [71,72] is also employed, which can easily be implemented by running Matlab NN toolbox several times because Matlab uses different subsets of the training data to obtain weight parameters. Furthermore, this process can relieve the concern related to selecting initial weight parameters for optimization because the process automatically selects different initial parameters. For example, Liu et al. [36] used this method with 50 attempts to predict a battery's RUL with uncertainty. Actually, a systematic method to handle the uncertainty in NN is the probabilistic neural network (PNN) [73] using the Parzen estimator [74]. However, most papers employ the PNN for classification or risk diagnosis [75,76], and the PNN for prognosis is rarely found, except for the study by Khawaja et al. [77]. They introduced a way to obtain not only confidence bounds but also confidence distributions based on PNN to predict a crack on a planetary gear plate. Khosravi et al. [78] reviewed aforementioned methods, and considered combined intervals from the methods.

In conclusion, although there have been a few attempts in NN algorithms to account for the uncertainty in the prediction process, they are not good enough at dealing with the noise and bias in the data.

2.2. Gaussian process (GP) regression

The GP is a commonly used method among regression-based data-driven approaches for prognostics, whose concept is illustrated in Fig. 4. The GP regression model is composed of a global model and its local departure. An outstanding feature of GP is that the simulated output, denoted by the blue dashed curve in Fig. 4, passes through the set of measured data (training data) $[\mathbf{x}_{1:n}, \mathbf{y}_{1:n}]$. When the prediction point (a new input), x^*, is at a measurement point (e.g., point A), $x_i (i = 1, 2, \ldots, n)$, the magnitude of departure is the same as the difference between measured data, $y_i (i = 1, 2, \ldots, n)$ and the global model; hence, the prediction output y^* becomes measured data, y_i. When the prediction point deviates from a measurement point (e.g., point B), the magnitude of departure is changed based on the correlation between B and the measurement points. As a result, if the prediction point is in between measurement points, the simulated output smoothly interpolates the measured data. If, however, the prediction point moves out of measurement points, which is extrapolation, the influence of this departure is reduced as correlation decreases and the GP becomes closer to the global model. Therefore, the case of extrapolation is not so different from an ordinary least-squares regression [30]. For interpolation, correlation between data points is an important factor to characterize GP's attribute, which is determined by selecting an appropriate type of covariance function and estimating its scale parameters (or hyperparameters). The scale parameters are usually determined using an optimization algorithm based on the measured data (training data).

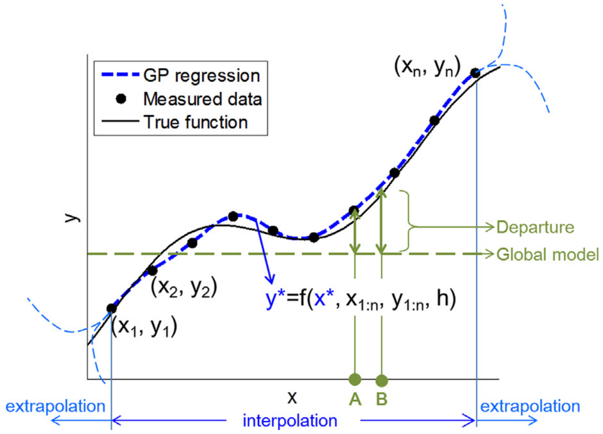


Fig. 4. Illustration of Gaussian process regression. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

GP regression can also be described in stochastic terms to account for the uncertainty in prediction by assuming that the measured output follows multivariate normal distribution with the mean and its covariance being the global model and its departure, respectively. One can obtain the GP model as the mean (composed of a global model and departure) and the variance found in Refs. [79,80] which is used to represent uncertainty in simulated GP outputs. In the following, similar issues to those of the NN-based prognostics are discussed in order to facilitate its appropriate use.

2.2.1. Issue 1: Model definition (global model and covariance function)

In GP, the performance largely depends on the selection of the global model and covariance function. The global model often takes the form of constant or polynomials, while the covariance function includes radial basis (or squared exponential), rational quadratic, neural network, Matern, periodic, constant and linear functions [81,82]. Mohanty et al. [83] compared the prediction results of crack length under variable amplitude loading using the radial basis function (RBF) and NN-based covariance function, and showed that RBF-based GP model outperformed one that was NN-based in their application. A few articles have also introduced non-stationary covariance function, which adapts to variable smoothness by adding or multiplying simple covariance functions. Paciorek and Schervish [84] showed that the results by non-stationary covariance functions are better than those by the stationary GP, but pointed out the loss of simplicity of the algorithm, as the non-stationary GP requires more parameters. Belhouari and Bermak [85] employed non-stationary GP to predict respiration signals and compared them with those from an exponential covariance function. Liu et al. [86] used the combination of three covariance functions to predict lithium-ion battery degradation.

Since GP has usually been employed in interpolation, a constant function has often been employed for the global model, and it has been considered as less important than the covariance function. In prognostics, which is equivalent to extrapolation, the global model in GP is more important than the covariance function. However, there is not much literature on selecting or updating the global model to improve the prediction capability in the extrapolation region.

2.2.2. Issue 2: Finding optimal parameters (scale parameters)

Determining the scale parameters in the covariance function is also important, since they determine the smoothness of the GP model. In general, the parameters are determined via an optimization algorithm by minimizing a likelihood function corresponding to the error between a global model and data [87]. Finding optimal parameters, however, is not guaranteed, and even if they are found, they are not always the best selection [88]. Since the scale parameters are seriously affected by the magnitudes of input and output values, a common practice is to normalize the input and output variables. Mohanty et al. [83] studied the performance of predicting crack growth according to three different types of scaling: logarithmic, normalized and log-normalized scaling. Neal [89] considered the scale parameters as distributions rather than deterministic values, and An and Choi [88] found that the result with distributions outperforms the one by deterministic values.

2.2.3. Issue 3: Number of data and uncertainty in data

Even though a large number of training data is usually good for increasing the accuracy of prediction results, it is not always true for GP because it also increases computational cost to calculate the inverse of the covariance matrix and may cause singularity as well. The direct matrix inversion may become computationally prohibitive when the number of data is greater than 1000 [90]. To relieve this problem, only a portion of the whole data sets can be used [91,92]. Melkumyan and Ramos [93] also suggested a new covariance function based on the cosine function that inherently provides a sparse covariance matrix. Sang and Huang [94] proposed an approximation method of a covariance function for both large and small number of training data based on a reduced rack covariance.

Prediction uncertainty is expressed with Gaussian noise, as mentioned before. Mohanty et al. [18] and Liu et al. [86] calculated the predictive confidence interval of crack length and degradation of a lithium-ion battery, respectively, using standard deviation from the results of stochastic GP. Also, the nugget effect representing the noise in data is sometimes considered by adding a value greater than zero to the diagonal terms of the covariance matrix, so that the simulated outputs do not pass noisy data points [95,96]. Even though the value of the nugget effect is also found via optimization along with the scale parameters, the value cannot be determined uniquely due to the coupling with scale parameters. In addition, in the case of small data, it is inherently difficult to figure out how much noise exists in the data.

3. Reviews on physics-based approaches

Physics-based approaches combine a physical damage model with measured data to predict future behavior of degradation or damage and to predict the RUL, as illustrated in Fig. 5. The behavior of a physical model depends on the model parameters that are obtained from laboratory test, or estimated in real time based on measured data up to the current time. Finally, the RUL is predicted by progressing the damage state until it reaches a threshold as indicated by the dashed curves in Fig. 5. Physics-based approaches have issues similar to data-driven approaches, but the individual algorithms of physics-based approaches are not as different from each other as we observed in the data-driven approaches. Therefore, the following subsections are presented in terms of issues in physics-based approaches, rather than their algorithms.

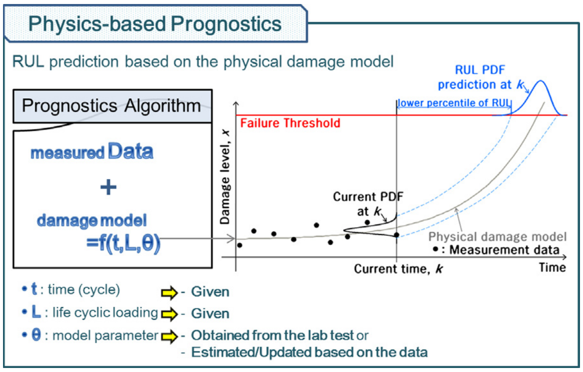


Fig. 5. Illustration of physics-based prognostics.

3.1. Issue 1: Physical model adequacy

Since physics-based approaches employ a physical model describing the behavior of damage, they have advantages in predicting long-term behaviors of damage. However, model validation should be carried out first, since most models contain assumptions and approximations. There has been much literature on model validation using statistical methods, such as the hypothesis test and the Bayesian method [97–100]. In general, as the model complexity increases, the number of model parameters increases, and estimation of parameters becomes more difficult. Recently, Coppe et al. [101] showed that the issue of model accuracy can be relieved by identifying equivalent parameters from a simpler model. A simple Paris model was used with an assumed stress intensity factor to predict crack growth of complex geometries in which the model parameters were adjusted to compensate for the error in the stress intensity factor. Although this is limited to the case of a similar damage behavior between the simple and complex model, additional efforts to validate model accuracy can be avoided.

3.2. Issue 2: Estimating parameters (physical model parameters under noise and bias)

3.2.1. Physics-based algorithms

Once a physical model is available, model parameter identification becomes the most important issue, which is performed by an estimation algorithm based on measured data. There are several algorithms such as the Kalman filter (KF) [102], the extended Kalman filter (EKF) [103], the particle filter (PF) [104], and the Bayesian method (BM) [105]. These algorithms are all based on Bayesian inference [106], in which measured data are used to estimate and update unknown parameters in the form of a probability density function (PDF). The prior distribution, which is prior knowledge or information of the unknown parameters, is multiplied by the likelihood function, which is the PDF value of measured data conditional on the given values of parameters, to obtain the updated PDF, known as the posterior distribution.

Among the aforementioned algorithms, KF, EKF and PF are based on the filtering technique that updates parameters recursively by taking one measurement data at a time. Since PF is readily applicable for a nonlinear system with non-Gaussian noise compared to KF and EKF, PF is mainly used. Orchard and Vachtsevanos [107] estimated the crack closure effect using PF for RUL

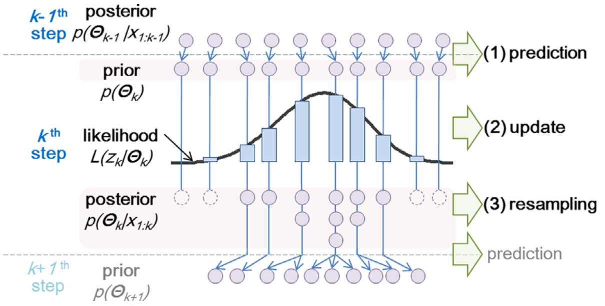


Fig. 6. Illustration of particle filter process.

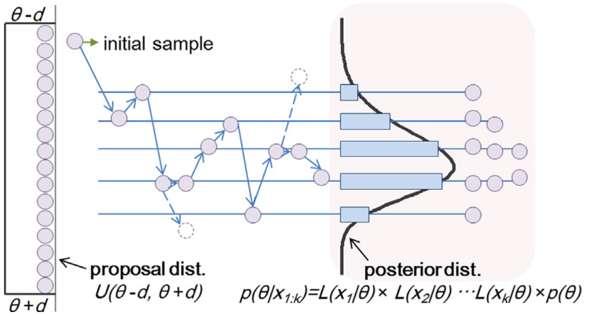


Fig. 7. Illustration of Bayesian Method process with Markov-Chain Monte Carlo.

prediction of a planetary carrier plate based on a vibration-based feature. Daigle and Goebel [108] used PF to estimate wear coefficients by considering multiple damage mechanisms in centrifugal pumps. On the other hand, the Bayesian method is used to implement batch estimation using all the measurement data up to the current time. An et al. [109] estimated wear coefficients to predict the joint wear volume of a slider-crank mechanism using this method. In this study, the PF, which is one of the three filtering methods, and the BM are discussed in the following.

3.2.1.1. Particle filter (PF). PF [104,110] is the most commonly used algorithm in the prognostics field, in which the posterior distribution of model parameters is expressed as a number of particles (or samples) and their weights, as shown in Fig. 6. There are three steps in PF process: (1) prediction—the posterior distribution of model parameters $\mathbf{\Theta}$ at the previous $(k-1)$th step are used as a prior at the current kth step, and the damage state at the current step is obtained by progressing from the previous one based on the physical model; (2) updating—model parameters and damage state are updated (i.e., corrected) based on the likelihood from measured data $\mathbf{x}$ at the current step; and (3) resampling—particles at the updated distribution are resampled based on their weights expressed as vertical bars in Fig. 6 by duplicating or eliminating samples with high or low weights, respectively. The resampled result corresponds to the posterior distribution at the current step and is also used as a prior distribution at the next $(k+1)$th step, which means that the Bayesian update is processed sequentially in PF. More detail is found in Ref. [110].

In PF, several issues need to be addressed in estimating parameter distributions, such as the accuracy of initial distribution and accumulated sampling error. In the latter, a particle depletion problem can occur since those particles with a low weight are eliminated, while those with a high weight are duplicated. A common practice to avoid particle depletion is to add random samples from an arbitrary distribution during the prediction step so that identical particles are not generated [108,111–113]. This method, however, can change probabilistic characteristics of parameters and can increase the variance of parameters. Gilks and Berzuini [114] proposed a resample-move algorithm based on the PF and Markov Chain Monte Carlo (MCMC) method [115]. Kim and Park [116] introduced the maximum entropy particle filter and demonstrated its effectiveness by applying it to highly nonlinear dynamical systems. Other efforts are found in Refs. [117,118].

3.2.1.2. Bayesian method (BM). In BM, Bayesian update is processed simultaneously; that is, the posterior distribution of parameters at the current step is obtained by a single equation, in which all the likelihood functions of measured data up to the current step are multiplied. Once the expression of the posterior distribution is available, a sampling method can be used to draw samples from the posterior distribution. The Markov chain Monte Carlo (MCMC) method [115] is usually employed, which is based on a Markov chain model in random walk as shown in Fig. 7. It starts from an arbitrary initial sample (old sample) and a new sample is drawn from an arbitrary proposal distribution centered at the old sample. The two samples are compared with each other based on an acceptance criterion, from which either one is selected. In Fig. 7, two circles with dashed lines mean that these new samples are not selected according to the criterion, while the old one is selected again. This process is repeated as many times as necessary until a sufficient number of particles are obtained.

In BM, there is no accumulated sampling error as opposed to PF. But there still exists some issues due to random walk. The location of the initial sample, the proposal distribution to draw a new sample and the acceptance ratio to the old sample can have an effect on the sampling results; improper settings or inadequate selection may lead to a convergence failure or show a stationary chain where the old sample is selected repeatedly. In Ref. [119,120], the marginal density function is recommended for the proposal distribution in order to reduce these effects. Gelfand and Sahu [121] presented two distinct adaptive strategies to accelerate the convergence of the MCMC algorithm. More literature can be found in Ref. [115].

3.2.2. Correlation between model parameters

One of the most challenging aspects of model parameter identification is correlation between parameters. Without properly identifying the correlation, the predicted RUL can be significantly different from the reality. An et al. [122] identified the correlation between parameters in a crack growth problem. They showed that even if the accurate value of each parameter was not identified under a high level of noise, the prediction results of damage growth and RUL were reliable since all combinations of the correlated parameters can yield the same prediction results.

3.3. Issue 3: Uncertainty in data (noise and bias)

In order to estimate model parameters of system in service, structural health monitoring (SHM) data are often used for the purpose of prognosis. SHM data could include a high level of noise and bias due to the sensor equipment, and much research addresses the noise and bias in SHM data. Gu et al. [123] presented a prognostics approach that detects the performance degradation of multilayer ceramic capacitors under temperature-humidity-bias conditions. Coppe et al. [124] showed that the uncertainty in structure-specific damage growth parameters can be progressively reduced, in spite of noise and bias in sensor measurements. Guan et al. [125] considered various uncertainties from measurements, modeling and parameter estimations to describe the stochastic process of fatigue damage accumulation based on a maximum entropy-based general framework. It was found that large noise induces slow convergence, and bias shifts RUL distribution.

Noise in sensor signals hinders the detection of degradation features, which adversely affect the prognostic capability in both the physics-based and data-driven approaches. To relieve this, de-noising is usually conducted in signal processing. Zhang et al. [126] proposed a de-noising scheme for improving the signal-to-noise ratio (SNR) and applied it to vibration signals obtained from a helicopter gearbox test-bed. Qiu et al. [127] introduced an enhanced robust method for bearing prognostics, which includes a wavelet filter-based method for extraction of a weak fault feature from vibration signals and a self-organizing map based method for bearing degradation assessment. Abouel-Seoud et al. [128] introduced a robust prognostics concept based on an optimal wavelet filter for fault identification of a gear tooth crack by acoustic emission.

4. Case study to compare different methods

In this section, prognostics algorithms of NN, GP, PF and BM are compared, which the authors believe are the most popular in prognostics applications. The goal is to provide guidelines so that engineers can choose an appropriate algorithm for their field of application. Since there are many variations of each algorithm, as mentioned in the review section, the most common and basic ones are employed to understand each algorithm's attributes. Each algorithm is tested using crack growth problems, including simple and complex damage behaviors. Their attributes, pros and cons are discussed based on the logic of their algorithms. The choice of the crack growth problem is because the damage growth model is available, and hence, it enables comparison of all the algorithms. The discussions are not limited to this problem and can be interpreted in a general context, since the results are caused by the algorithms' inherent attributes.

4.1. Problem definition

In this section, two different physical models, the Paris model [129] and Huang's model [130], are employed to describe the crack growth behavior under ten different constant and variable amplitude loading conditions, respectively, as shown in Fig. 8. The Paris model is defined as

$$\frac{da}{dN}=C(\Delta K)^m$$

where a is the crack size, N the loading cycle, ΔK the stress intensity factor, and m and C are mode parameters. This model assumes that mode-I loading in an infinite plate. On the other hand, Huang's model is defined as

$$\frac{da}{dN}=C\left[\{\Delta K_{eq}(\beta,n,\sigma_Y)\}^m-\{\Delta K_{th}\}^m\right]$$

where m, C, ΔK_{th}, β, n, and σ_Y are model parameters. In Huang's model, the equivalent stress intensity factor, ΔK_{eq}, considers the effect of crack tip plasticity as well as crack closure after overloading. For detailed explanation of Huang's model, readers are referred to Reference [130].

In order to implement prognostics, synthetic measured data are generated by adding random noise and deterministic bias from the model generated using the following true parameters:

$$\text{Paris model parameters}: m=3.8,\quad C=1.5\text{e}-10$$

$$\text{Huang's model parameters}: m=3.1,\quad C=5.5\text{e}-11,$$
$$\Delta K_{th}=5.2,\quad \beta=0.2,\quad n=2.8,\quad \sigma_y=580$$

For the Paris model (simple model), true crack growth data are generated with an interval of 100 cycles under ten different loading cases, of which the stress range is gradually increased from 65 MPa to 83 MPa with 2 MPa increments. Results are given in Fig. 8(a), in which the 26 data points are available in a set until the end of 2500 cycles, and ten sets are available with the set ID denoted by the number in a circle. For Huang's model (complex model), ten data sets are given in Fig. 8(b), which exhibit crack retardation and acceleration due to complex loadings as defined by Fig. 9. In the figure, a loading block is composed of the minimum load fixed at a constant of 5 MPa and two maximum loads (we call these nominal load and overload, respectively) varying in terms of cycles. The loading block is applied with five different scenarios in terms of the cycles as specified in the table, each lasting over the period of 5000 cycles. In each of the scenarios, load ranges are given, all with the width of 10 MPa, which represent the values of ten loading cases with increment of unit MPa. For example, the first loading case consists of a loading block of the nominal load 65 MPa and overload 125 MPa lasting

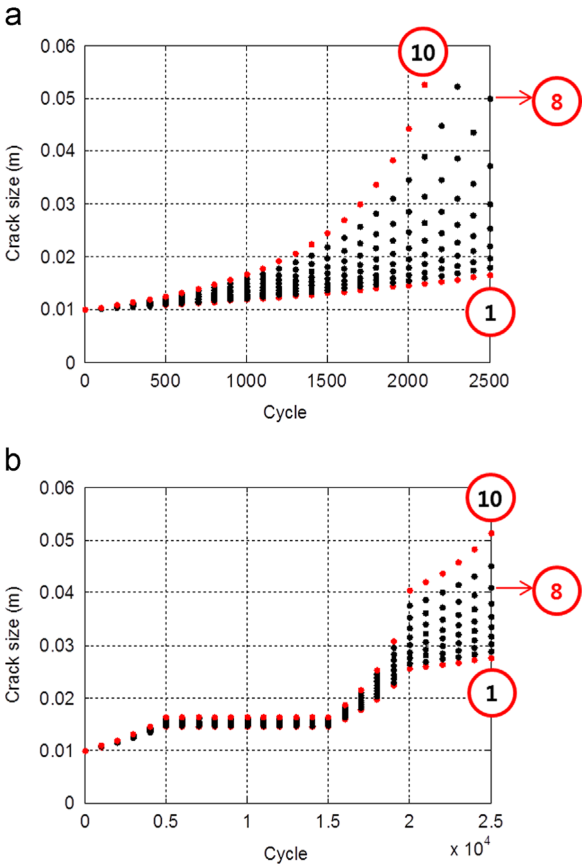
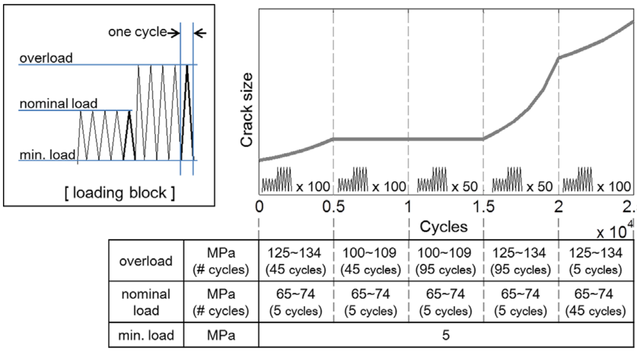


Fig. 8. Ten data sets for crack growth from two different models. (a) simple Paris model with 2 parameters and (b) complex Huang's model with 6 parameters.



overload	MPa (# cycles)	125~134 (45 cycles)	100~109 (45 cycles)	100~109 (95 cycles)	125~134 (95 cycles)	125~134 (5 cycles)
nominal load	MPa (# cycles)	65~74 (5 cycles)	65~74 (5 cycles)	65~74 (5 cycles)	65~74 (5 cycles)	65~74 (45 cycles)
min. load	MPa	5				

Fig. 9. Variable load amplitude spectrum for Huang's model.

5 and 45 cycles, respectively, during the period of 0 to 5000 cycles. This is applied the same way, as indicated in the table, over the rest of the period until it reaches 25,000 cycles. Other loading cases are made by increasing the load magnitude with 1 MPa increments in each of the scenarios, which makes ten loading sets, and yields the crack growth results in Fig. 8(b). In order to simulate measured crack size data, different levels of noise and bias are added to the true crack sizes that are calculated from the model with true parameters. A deterministic bias of -3 mm is added first, after which random noise is uniformly distributed between $-u$mm and $+u$mm, where u of 0, 1 and 5 mm are considered.

4.2. Data-driven approach

4.2.1. Practical considerations

In NN, the network model is constructed based on FFNN with three input nodes and one hidden layer with two nodes (refer to Fig. 3). Then, the total number of weight parameters becomes 11: eight weights ($3\times 2+2\times 1$) and three biases ($2+1$). Since there is one hidden layer, two transfer functions are required, in which the tangent sigmoid and pure linear functions are employed. In GP, the polynomial function is employed for the global model, and the following one-parameter radial basis covariance function is employed:

$$R\left(\mathrm{x}^i,\mathrm{x}^j\right)=exp\left[-(d/h)^2\right],\ d=|\mathrm{x}^i-\mathrm{x}^j|,\quad i,j=1,\ldots,n$$

where h is a scale parameter to be identified, $\mathrm{x}^i,\mathrm{x}^j$ are vectors of input variables, and n is the total number of training data. Regarding uncertainty, GP accounts for noise in training data as a part of the process. In NN, samples of solutions obtained by repeating 30 times with different initial values and different subsets of the training data represent uncertainty in the same way as Liu et al. [36] applied bootstrapping [71] to the battery problem.

There are some typical features in data-driven approaches. The first is that not only training data sets but also the data in the prediction data set up to the current cycle are used for training. For example, if sets #2 and #6 are used for training and set #8 for prediction, the earlier data of set #8 before the current cycle are included for training. The second is the way of assigning inputs and output for training: several recent damage data are used as input variables, while the damage data at the current cycle is used as the output. In the crack growth problem, we have 26 data points in a single data set. If we use past three data points as inputs, for example, we have $26-3=23$ training data, in which x_{k-3}, x_{k-2}, x_{k-1} are used as inputs and x_k as the output. The third feature is about the method of prediction. Depending on how to predict the future damage state, prediction methods are divided into short-term and long-term predictions. Short-term prediction is a one-step-ahead prediction; i.e., the three most recent data x_{k+1}, x_{k+2}, x_{k+3} are used to predict one step ahead x^p_{k+4}. On the other hand, long-term prediction is multi-step prediction; i.e., the predicted data x^p_{k+1}, x^p_{k+2}, x^p_{k+3} are used as inputs to predict the next x^p_{k+4}. This enables prediction into the far future. If loading conditions are available, they can also be included in the input variables.

4.2.2. Prognosis results

Fig. 10 shows the comparison between NN and GP under different levels of noise. In Fig. 10, blue star markers and circles are, respectively, training data to the current cycle (1500 cycles) and simulation results. The thick dotted curve and the thick dashed curve are, respectively, the medians of short-term and long-term predictions; corresponding thin curves represent 90% confidence intervals. Lastly, the black solid curve and green horizontal line are the true damage growth and damage threshold, respectively.

Fig. 10(a) and (b) are the results of NN and GP, respectively, under low noise ($u=1$ mm) with a single data set #5. GP shows very accurate and more precise prediction results than NN. Note that even with only one set of data for prediction, the result is good. In this case, GP prediction is just to extrapolate the model, in which the choice of a global model is important and may affect the accuracy of prediction significantly. It should be noted however that, unlike the conventional GP, which employs the cycles as the input and crack data as the output, the GP here employs crack data for both the input and output. The reason for the lower performance of NN compared to GP is that NN has a local optimum as the network model has 11 parameters.

Fig. 10(c) and (d) are the results under zero noise ($u=0$ mm) with prediction set #8 and training sets #7, 9. Fig. 10(c) and (d) is the case of interpolation, since the prediction set is in-between training sets. Due to this, GP outperforms NN based on correlation with training data.

This, however, is not applicable to a large level of noise ($u=5$ mm), as shown in Fig. 10(e) and (f). Fig. 10(f) shows that long term prediction using GP fails to follow the true curve, since input variables fail to show a consistent behavior due to the large level of noise; thus, the correlation cannot clearly be identified. In addition, a large number of data adversely affect handling the covariance matrix. In this case, NN outperforms GP for both short- and long-term predictions, as shown in Fig. 10(e) because a large number of data provides more information to fit the transfer functions to damage growth. Also, a combination of transfer functions is much less restricted to the level of noise than constructing the covariance matrix. It is important to consider types of input variables and transfer functions/global model in data-driven prognostics.

4.3. Physics-based approach

For physics-based approaches, a normal distribution is employed for the likelihood function, and the prior distributions of model parameters, noise and bias are assumed as listed in Table 1. The distributions of parameters are obtained using 5000 samples in both PF and BM, and future damage growth and RUL are predicted by applying individual samples to the physical model.

4.3.1. Physics-based results

As mentioned before, bias in measured data cannot be handled in data-driven approaches, while physics-based approaches, both PF and BM, can, as shown in Fig. 11(a) and (b), respectively. In the figure, measured data up to the current cycle (blue star makers) are biased by -3 mm, with a low level of noise ($u=1$ mm); measured crack sizes are consistently less than the true sizes. The medians of prediction (dashed curves), however, are close to the true sizes, which means that the bias is well-identified and compensated. Since the physical model describes the behavior of damage accurately, the constant difference between model and data can be identified as a bias. Further study for noise and bias can be found in Ref. [122].

The difference between PF and BM is negligible in terms of prediction results because the two methods have the same theoretical foundation with the same physical model. The only difference is the way of updating distributions and generating samples. When the damage model is given as a closed form like the Paris model, BM is faster than PF because the posterior distribution is given as a single expression and there is no accumulated sampling error. On the other hand, PF accumulates

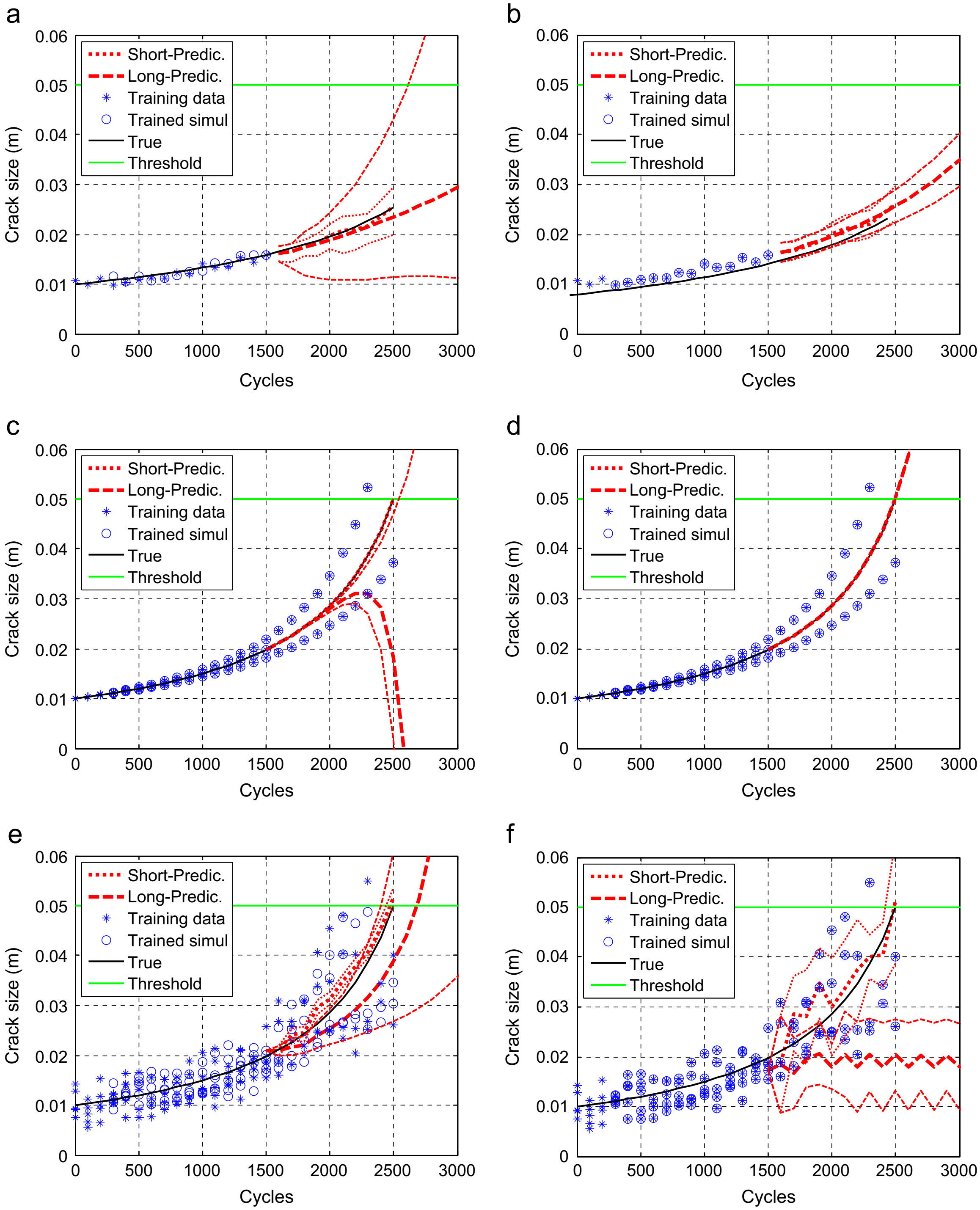


Fig. 10. Prognostics results from data-driven approaches for a simple damage growth model. (a) NN under small noise with prediction set #5. (b) GP under small noise with prediction set #5. (c) NN under perfect data with prediction set #8 and training sets #7, 9. (d) GP under perfect data with prediction set #8 and training sets #7, 9. (e) NN under large noise with prediction set #8 and training sets #6, 7, 9, 10. (f) GP under large noise with prediction set #8 and training sets #6, 7, 9, 10. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

sampling error during the updating process, but is very efficient when damage degradation is given in the incremental form, like Huang's model. From these attributes, PF can handle the case of fixed model parameters obtained from laboratory tests by updating damage size only. On the other hand, BM is not practical for such a case because of tremendous computational costs caused by propagating damage for every cycle with thousands of samples separately [114,131,132]. This is a key difference between the two methods, and more detailed comparison between PF and OBM can be found in An et al. [131].

Fig. 12 shows different results from the two cases using parameters obtained from laboratory tests and updating the parameters. It is assumed that Paris model parameter, *m*, is given between 3.7 and 4.3 from laboratory tests, which is biased from its true value, 3.8, and parameter *C* is the same as the prior distribution given in Table 1. It is clear that error in parameters obtained from laboratory tests produces improper results, as shown in Fig. 12(a): the uncertainty in prediction is too wide and the median of prediction greatly differs from the true curve. For the case of updating parameters, the loading condition from data set #1 is

Table 1
Prior distributions of model parameters, noise and bias.

Paris model	$m=U(3.3,4.3)$	$C=U(log(5\times10^{-11}),\ log(5\times10^{-10}))$	
	(True: 3.8)	(True: 1.5e−10)	
Huang's model	$m=U(2.8,3.4)$	$C=U(log(3\times10^{-11}),\ log(8\times10^{-11}))$	$\Delta K_{th}=U(2,8)$
	(True: 3.1)	(True: 5.5e−11)	(true: 5.2)
	$\beta=U(0.1,0.4)$	$n=U(1.5,4)$	$\sigma_y=U(400,600)$
	(True: 0.2)	(True: 2.8)	(True: 580)
Noise and bias	For small noise	For large noise	Bias $=U(-0.004,0.004)$
	$\sigma=U(0,0.003)$	$\sigma=U(0,0.006)$	

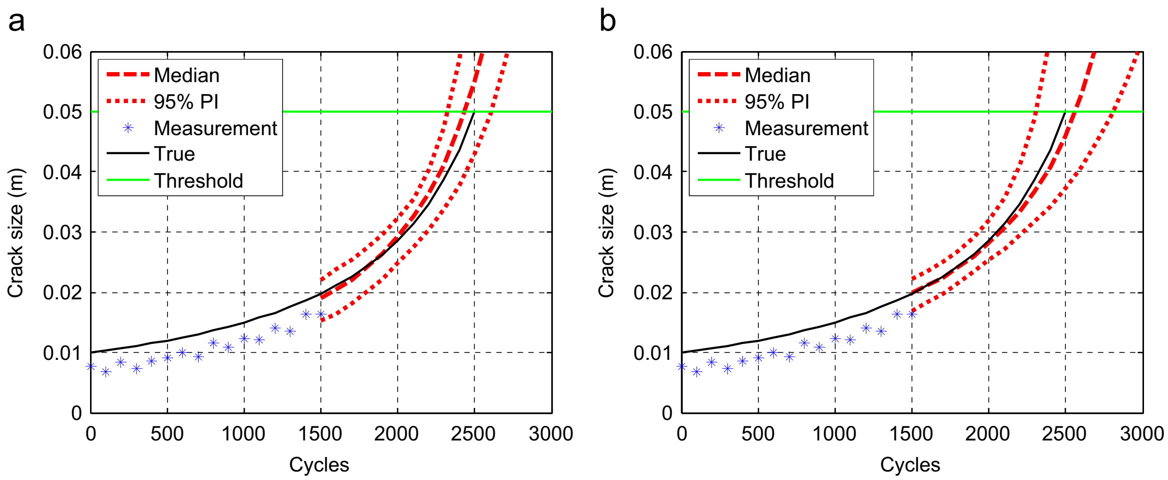


Fig. 11. Prognostics results from physics-based approaches for a simple damage growth under small noise and negative bias. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

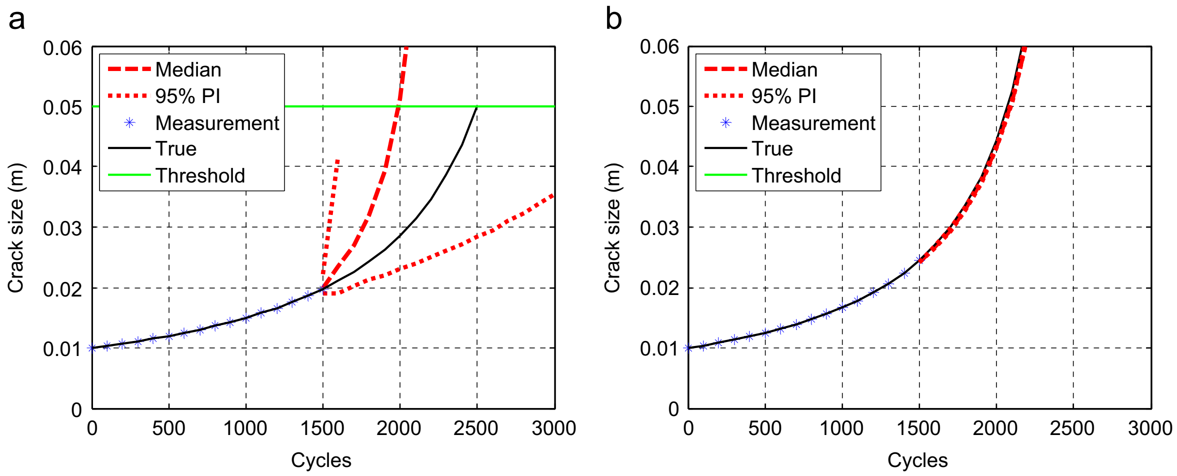


Fig. 12. Comparison of prognostics results using parameters obtained from laboratory tests and parameters updated in the field. (a) Parameters from laboratory tests: prediction set #8 with biased information. (b) Parameter updated: prediction set #10 with loading from data set #1.

given to predict data set #10. The loading difference between two data sets is 18 MPa. Even under incorrect loading conditions, the prediction result from updated parameters shows good results, as indicated in Fig. 12(b). Since the parameters from laboratory tests can be different from those in service due to environmental conditions, model parameters should be updated along with measurement data.

4.4. Comparison between NN and PF

In this section, the data-driven and physics-based approaches are compared in terms of the damage growth by Huang's model (complex model). In the data-driven approaches, NN is chosen because GP is the same as the global model (a polynomial function) at the extrapolation cycles, while different combinations of transfer functions of NN cam predict better than a polynomial function. Since BM is less advisable in a complex model due to its expensive computational costs, only PF is considered in the physics-based approaches. To understand how many data sets for NN are required to yield comparable results with PF, different numbers of training sets are randomly selected among the ten data sets, whose result is presented in Fig. 13(a). Based on three training data sets (for visibility, training data and trained simulation are not shown in the figure), short-term prediction by NN successfully follows the future damage, and long-term prediction by NN are comparable with that of PF in Fig. 13(b).

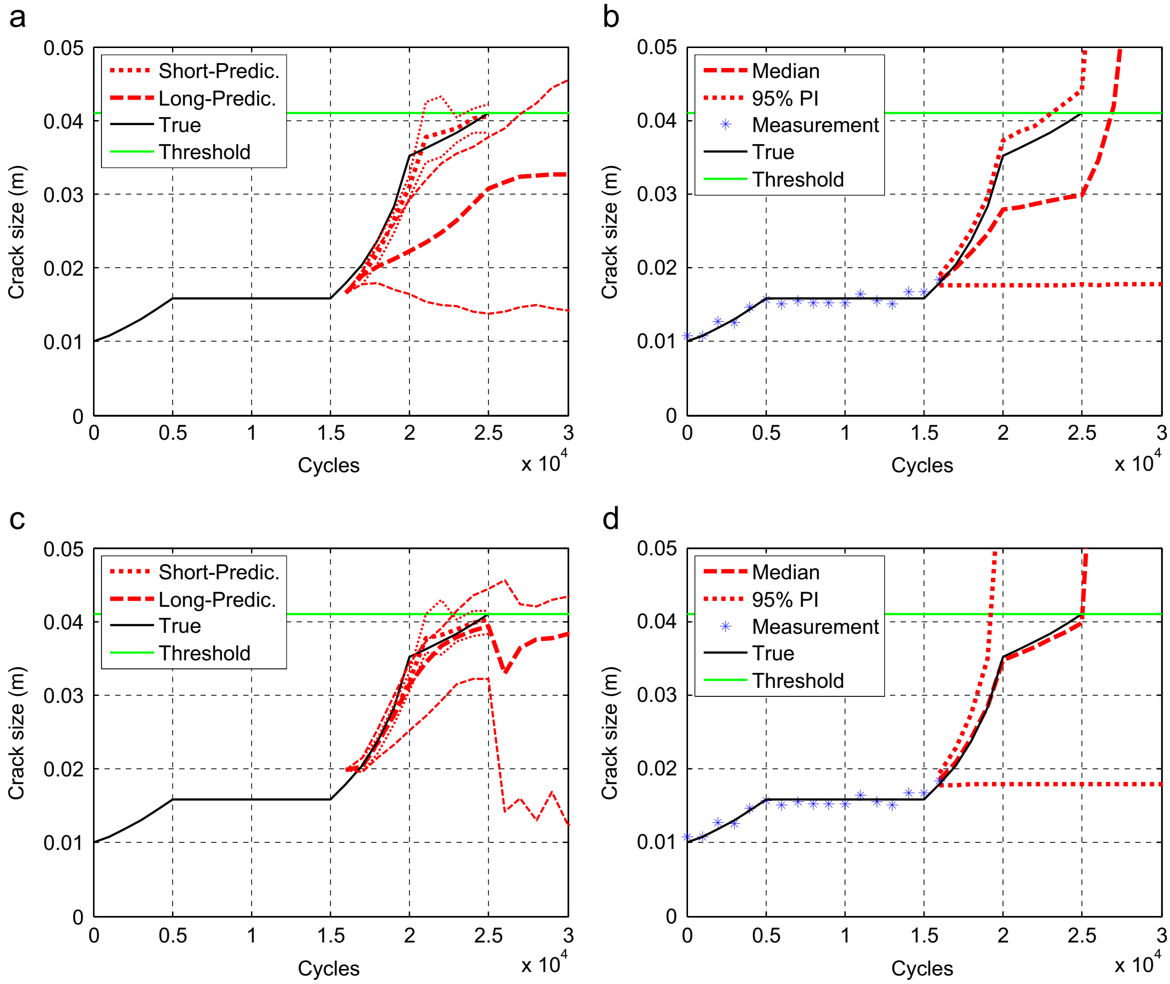


Fig. 13. Comparison of NN and PF for a complex damage growth model under small noise. (a) NN with training sets #2, 7, 9. (b) PF with distributed initial damage size. (c) NN with training sets #6, 7, 10 and loading conditions. (d) PF with true value of initial damage size.

These results can be improved: If the loading conditions are added to training in NN, long-term prediction behaves better than no loading case and closer to the short-term results as shown in Fig. 13(c). For PF, if the distribution of initial damage size in Fig. 13(b) is updated to be close to the true initial damage, the median will be close to the true curve because the physical model largely depends on the initial damage size. It is, however, a time-consuming process to update the initial damage in PF. Fig. 13(d) shows the results using the true value of initial damage instead. Even though physics-based approaches undoubtedly outperform data-driven ones if both a physical model and loading conditions are available, data-driven approaches with more training data (three sets in this case) and other information (loading condition in this case) can outperform physics-based ones, as shown in Fig. 13(c) and (d).

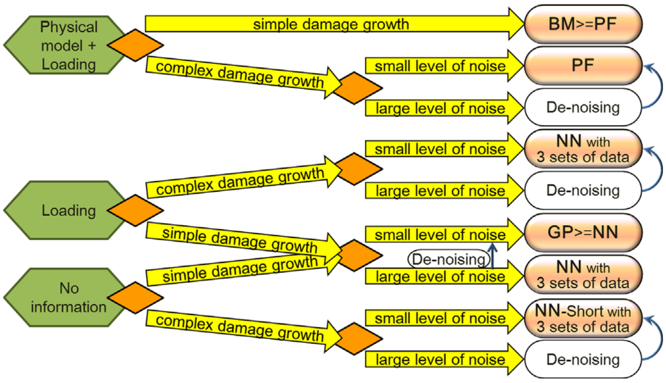


Fig. 14. Selection tree for an appropriate prognostics method.

4.5. Results summary

In terms of algorithms, results from case studies can be summarized as follows. The GP works well when the covariance function is well-defined, such as the case of low noise in data and simple behavior of the damage model. However, all other algorithms work well for such a benign case as well. The GP can outperform other algorithms if interpolation conditions can be considered with the given information. The GP is easy to implement and quick to calculate, which gives prediction results with interval within around 2 s, while NN depends on each training case (a few seconds to minutes for one training) that is usually repeated to account for prediction uncertainty. The NN is advantageous for cases with high levels of noise and complex models with many training data sets. However, it would be challenging to obtain many sets of training data in practice. PF and BM are less affected by the level of noise and model complexity, but they can be employed only when a physical model and loading conditions are available. The results from the two methods are not much different, but PF has a wide range of applications and BM is fast when the expression of posterior distribution is available explicitly. When loading conditions and physical models are not

Table 2
Summary for selective references.

Reference	Algorithm	*Note*
Chakraborty et al. [21]	NN	• To understand usage of NN algorithm • Predicted flour prices using variable network models
Liu et al. [36]	NN	• Applied repeating NN (bootstrapping) • Predicted battery's RUL with uncertainty
Sacks et al. [87]	GP	• To understand GP regression
Mohanty et al. [83]	GP	• Compared the results from different covariance functions • Predicted crack length under variable loading
An et al. [110]	PF	• To understand PF easily (tutorial) • Predicted RUL of battery and crack growth
An et al. [122]	BM	• To understand BM process • Considered different levels of noise and bias • Predicted RUL of crack growth

available, short-term prediction can be done by using data-driven approaches with at least three data sets. For long-term predictions, loading conditions are required.

Fig. 14 shows a selection tree based on the aforementioned summary. The first column is composed of three parts depending on the existence of information. In case of loading information with the physical model, as given in the first diamond, it does not have to be accurate as mentioned in Fig. 12(b). But the loading information in the second diamond as used in the data-driven approach should be accurate. Otherwise, the third diamond can be used. The tree is divided into two depending on (a) the complexity of damage growth behavior and (b) the level of noise. “BM > =PF” and “GP > =NN” mean that BM and GP are better than PF and NN, but PF and NN are still viable. “NN-Short” means that only short-term prediction is possible. However, it is noted that these results are not conclusive. For example, while the path “No information”→“complex damage growth”→“small level of noise” leads to the suggestion of NN-short with 3sets of data, it may not always be the case because there are still chances for achieving long term prediction by combining more transfer functions (adding hidden layers) and adding hidden nodes.

5. Conclusion

This paper provides a practical review of both data-driven and physics-based approaches for prognostics. As common prognostics algorithms, NN, GP, PF and BM are reviewed and employed for case studies to discuss their attributes, pros and cons and applicable conditions. Even if advanced algorithms are available, basic algorithms are employed in this study and the results are analyzed by focusing on their intrinsic characteristics. This will be helpful for beginners in prognostics to choose the best algorithm for their field of application. Furthermore, one can improve each algorithm’s performance by considering three issues discussed in Sections 2 and 3, and can develop hybrid approaches after understanding intrinsic characteristics of each algorithm, which are expected as future works on the topic of prognostics algorithms. Lastly, selective references of the four basic algorithms that are used in this study are listed in Table 2.

Acknowledgments

This work was supported by a grant from the International Collaborative R&D Program (0420-2011-0161) of the Korea Institute of Energy Technology Evaluation and Planning (KETEP), funded by the Korean government’s Ministry of Knowledge Economy.

References

[1] Jardine AKS, Lin D, Banjevic D. A review on machinery diagnostics and prognostics implementing condition-based maintenance. Mech Syst Sig Process 2006;20(7):1483–510.

[2] Martin KF. A review by discussion of condition monitoring and fault diagnosis in machine tools. Int J Mach Tools Manuf 1994;34(4):527–51.

[3] Samuel PD, Pines DJ. A review of vibration-based techniques for helicopter transmission diagnostics. J Sound Vib 2005;282(1–2):475–508.

[4] Tang B, Song T, Li F, Deng L. Fault diagnosis for a wind turbine transmission system based on manifold learning and Shannon wavelet support vector machine. Renewable Energy 2014;62:1–9.

[5] Deguchi Y, Kamimoto T, Wang ZZ, Yan JJ, Liu JP, Watanabe H, et al. Applications of laser diagnostics to thermal power plants and engines. (In press, Available online). Appl Therm Eng 2014;6.

[6] Sugumaran V, Sabareesh GR, Ramachandran KI. Fault diagnostics of roller bearing using kernel based neighborhood score multi-class support vector machine. Expert Syst Appl 2008;34:3090–8.

[7] Yan R, Gao RX. Approximate entropy as a diagnostic tool for machine health monitoring. Mech Syst Sig Process 2007;21:824–39.

[8] Kunli M, Yunxin W. Fault diagnosis of rolling element bearing based on vibration frequency analysis. In: Proceedings of the third international conference on measuring technology and mechatronics automation, Shangshai, China; January 6–7, 2011.

[9] Radzieński M, Krawczuk M, Palacz M. Improvement of damage detection methods based on experimental modal parameters. Mech Syst Sig Process 2011;25:2169–90.

[10] Bediaga I, Mendizabal X, Arnaiz A, Munoa J. Ball bearing damage detection using traditional signal processing algorithms. IEEE Instrum Meas Mag 2013;16(2):20–5.

[11] Grall A, Bérenguer C, Dieulle L. A condition-based maintenance policy for stochastically deteriorating systems. Reliab EngSyst Saf 2002;76(2):167–80.

[12] Liu X, Li J, Al-Khalifa K, Hamouda AS, Coit DW, Elsayed EA. Condition-based maintenance for continuously monitored degrading systems with multiple failure modes. Qual Reliab Eng 2013;45(4):422–35.

[13] Si XS, Wang W, Hu CH, Zhou DH. Remaining useful life estimation—a review on the statistical data driven approaches. Eur J Oper Res 2011;213:1–14.

[14] Lee J, Wu F, Zhao W, Ghaffari M, Liao L, Siegel D. Prognostics and health management design for rotary machinery systems—reviews, methodology and applications. Mech Syst Sig Process 2014;42(1–2):314–34.

[15] Saha B, Goebel K, Christophersen J. Comparison of prognostic algorithms for estimating remaining useful life of batteries. Trans Inst Meas Control 2009;31(3–4):293–308.

[16] Xing, Y, Williard, N, Tsui, K-L, Pecht, M., A comparative review of prognostics-based reliability methods for lithium batteries. In: Prognostics and system health management conference, Shenzhen, China; May 24–25, 2011.

[17] Zhang J, Lee J. A review on prognostics and health monitoring of Li-ion battery. J Power Sources 2011;196:6007–14.

[18] Liao L, Köttig F. Review of hybrid prognostics approaches for remaining useful life prediction of engineered systems, and an application to battery life prediction. IEEE Trans Reliab 2014;63(1):191–207.

[19] Xu J, Xu L. Health management based on fusion prognostics for avionics systems. J Syst Eng Electron 2011;22(3):428–36.

[20] Xing, Y, Miao, Q, Tsui, K-L, Pecht, M., Prognostics and health monitoring for lithium-ion battery. 2011 In: IEEE international conference, pp. 242–247.

[21] Chakraborty K, Mehrotra K, Mohan CK, Ranka S. Forecasting the behavior of multivariate time series using neural networks. Neural Networks 1992;5:961–70.

[22] Ahmadzadeh F, Lundberg J. Remaining useful life prediction of grinding mill liners using an artificial neural network. Miner Eng 2013;53:1–8.

[23] Li D, Wang W, Ismail F. Enhanced fuzzy-filtered neural networks for material fatigue prognosis. Appl Soft Comput 2013;13(1):283–91.

[24] Zio E, Maio FD. A data-driven fuzzy approach for predicting the remaining useful life in dynamic failure scenarios of a nuclear system. Reliab Eng Syst SafReliab EngSyst Saf 2010;95:49–57.

[25] Silva RE, Gouriveau R, Jemeï S, Hissel D, Boulon L, Agbossou K, et al. Proton exchange membrane fuel cell degradation prediction based on adaptive neuro-fuzzy inference systems. Int J Hydrogen Energy 2014;39(21):11128–11144.

[26] Wilson AG, Adams RP. Gaussian process covariance kernels for pattern discovery and extrapolation. arXiv preprint arXiv:1302.4245; 2013. URL: ⟨http://arxiv.org/pdf/1302.4245v3.pdf⟩.

[27] Seeger M. Gaussian processes for machine learning. Int J Neural Syst 2004;14(2):69–106.

[28] Yan J, Liu Y, Han S, Qiu M. Wind power grouping forecasts and its uncertainty analysis using optimized relevance vector machine. Renewable Sustainable Energy Rev 2013;27:613–21.

[29] Benkedjouh T, Medjaher K, Zerhouni N, Rechak S. Health assessment and life prediction of cutting tools based on support vector regression. (article published online). J Intell Manuf 2013. http://dx.doi.org/10.1007/s10845-013-0774-6 (April).

[30] Coppe A, Haftka RT, Kim NH. Uncertainty identification of damage growth parameters using nonlinear regression. AIAA J 2011;49(12):2818–21.

[31] Wang X, Balakrishnan N, Guo B, Jiang P. Residual life estimation based on bivariate non-stationary gamma degradation process. J Stat Comput Simul 2013:1–17 (ahead-of-print).

[32] Si XS, Wang W, Hu CH, Chen MY, Zhou DH. A wiener process-based degradation model with a recursive filter algorithm for remaining useful life estimation. Mech Syst Sig Process 2013;35(1–2):219–37.

[33] Liu A, Dong M, Peng Y. A novel method for online health prognosis of equipment based on hidden semi-Markov model using sequential Monte Carlo methods. Mech Syst Sig Process 2012;32:331–48.

[34] Svozil D, Kvasnička V, Pospíchal J. Introduction to multi-layer feed-forward neural networks. Chemom Intell Lab Syst 1997;39:43–62.

[35] Duch W, Jankowski N. Survey of neural transfer functions. Neural Comput Surv 1999;2:163–212.

[36] Liu, J, Saxena, A, Goebel, K, Saha, B, Wang, W, An adaptive recurrent neural network for remaining useful life prediction of lithium-ion batteries. In: Annual conference of the prognostics and health management society, Portland, Oregon; October 10–16, 2010.

[37] Firth AE, Lahav O, Somerville RS. Estimating photometric redshifts with artificial neural networks. Mon Not R Astron Soc 2003;339:1195. http://dx.doi.org/10.1046/j.1365-8711.2003.06271.

[38] Rumelhart DE, Hinton GE, Williams RJ. Learning internal representations by error propagation. In: Parallel distributed processing: explorations in the microstructure of cognition, vol. 1. Foundations, MIT Press; 1986; 318–362.

[39] Bodén M. A guide to recurrent neural networks and backpropagation. In: The DALLAS project. Report from the NUTEK-supported project AIS-8: application of data analysis with learning systems, 1999–2001. (SICS technical report T2002:03). In: Holst A, editor. Sweden: SICS, Kista; 2002.

[40] Wilamowski, BM, Iplikci, S, Kaynak, O, Efe, MÖ., An algorithm for fast convergence in training neural networks. In: Proceedings of the international joint conference on neural networks, Vol. 3; 2001. pp. 1778–1782.

[41] Liu P, Li H. Fuzzy neural network theory and application. Singapore: World Scientific; 2004.

[42] He Y, Tan Y, Sun Y. Wavelet neural network approach for fault diagnosis of analogue circuits. IEEE Proc Circuits Devices Syst 2004;151(4):379–84.

[43] Bicciato, S, Pandin, M, Didonè, G, Bello, CD., Analysis of an associative memory neural network for pattern identification in gene expression data. In: Biokdd01, workshop on data mining in bioinformatics, pp. 22–30.

[44] Happel BLM, Murre JMJ. The design and evolution of modular neural network architectures. Neural Networks 1994;7:985–1004.

[45] Zhang G, Yuen KKF. Toward a hybrid approach of primitive cognitive network process and particle swarm optimization neural network for forecasting. Procedia Comput Sci 2013;17:441–8.

[46] Rovithakis GA, Maniadakis M, Zervakis M. A hybrid neural network/genetic algorithm approach to optimizing feature extraction for signal classification. IEEE Trans Syst Man Cybern Part B Cybern 2004;34(1):695–702.

[47] Lawrence, S, Giles, CL, Tsoi, AC., What size neural network gives optimal generalization? Convergence properties of backpropagation technical reports, UM computer science department, UMIACS; Octobor 15, 1998.

[48] Gómez I, Franco L, Jérez JM. Neural network architecture selection: can function complexity help? Neural Proc Lett 2009;30:71–87.

[49] Sheela KG, Deepa SN. Review on methods to fix number of hidden neurons in neural networks. Math Prob Eng 2013;2013 (Article ID 425740, 11 pages).

[50] Ostafe, D, Neural network hidden layer number determination using pattern recognition techniques. In: Second Romanian-Hungarian joint symposium on applied computational intelligence, Timisoara, Romania; 2005.

[51] Chang JF, Hsieh PY. Particle swarm optimization based on back propagation network forecasting exchange rates. Int J Innovative Comput Inf Control 2011;7(12):6837–47 (December).

[52] Guo Z, Zhao W, Lu H, Wang J. Multi-step forecasting for wind speed using a modified EMD-based artificial neural network model. Renewable Energy 2012;37(1):241–9.

[53] Hodhod OA, Ahmed HI. Developing an artificial neural network model to evaluate chloride diffusivity in high performance concrete. HBRC J 2013;9(1):15–21.

[54] Kang LW, Zhao X, Ma J. A new neural network model for the state-of-charge estimation in the battery degradation process. Appl Energy 2014;121:20–7.

[55] Salomon R, Hemmen JLV. Accelerating backpropagation through dynamic self-adaptation. Neural Networks 1996;9(4):589–601.

[56] Chen, SC, Lin, SW, Tseng, TY, Lin, HC, Optimization of back-propagation network using simulated annealing approach. In: IEEE international conference on systems, man, and cybernetics, Taipei, Taiwan; October 8–11, 2006.

[57] Subudhi, B, Jena, D, Gupta, MM, Memetic differential evolution trained neural networks for nonlinear system identification. In: IEEE region 10 colloquium and the third international conference on industrial and information systems, Kharagpur, India; December 8–10, 2008.

[58] Nawi NM, Ransing RS, Ransing MR. An improved conjugate gradient based learning algorithm for back propagation neural networks. Int J Comput Intell 2008;4(1):46–55.

[59] Soares S, Antunes CH, Araújo R. Comparison of a genetic algorithm and simulated annealing for automatic neural network ensemble development. Neurocomputing 2013;121:498–511.

[60] Jacobs RA. Methods for combining experts' probability assessments. Neural Comput 1995;7(5):867–88.

[61] Drucker H, Cortes C, Jackel LD, LeCun Y, Vapnik V. Boosting and other ensemble methods. Neural Comput 1994;6(6):1289–301.

[62] Krogh A, Vedelsby J. Neural network ensembles, cross validation, and active learning. In: Tesauro G, Touretzky D, Leen T, editors. Advances in neural information processing systems, vol. 7. The MIT Press; 1995. p. 231–8.

[63] Chao M, Zhi S, Liu X, Min S. Neural network ensembles based on copula methods and distributed multiobjective central force optimization algorithm. Eng Appl Artif Intell 2014;32:203–12.

[64] Naftaly U, Intrator N, Horn D. Optimal ensemble averaging of neural networks. Network Comput Neural Syst 1997;8(3):283–96.

[65] Neal, RM., Bayesian learning for neural networks PhD thesis, University of Toronto, Ontario, Canada, 1995.

[66] Freitas, de JFG., "Bayesian methods for neural networks." PhD thesis. University of Cambridge, UK, 2003.

[67] Chryssoloiuris G, Lee M, Ramsey A. Confidence interval prediction for neural network models. IEEE Tran Neural Networks 1996;7(1):229–32.

[68] Veaux RD, Schumi J, Schweinsberg J, Ungar LH. Prediction intervals for neural networks via nonlinear regression. Technometrics 1998;40(4):273–82.

[69] Rivals I, Personnaz L. Construction of confidence intervals for neural networks based on least squares estimation. Neural Networks 2000;13(4–5):463–84.

[70] Yang, L, Kavli, T, Carlin, M, Clausen, S, Groot, PFM., An evaluation of confidence bound estimation methods for neural networks. In: European symposium on intelligent techniques; 2000, Aachen, Germany, September 14–15, 2000.

[71] Efron B, Tibshirani RJ. An introduction to the bootstrap. FL: Chapman & Hall/CRC; 1994.

[72] Khosravi A, Nahavandi S, Creighton D. Quantifying uncertainties of neural network-based electricity price forecasts. Appl Energy 2013;112:120–9.

[73] Specht DF. Probabilistic neural networks. Neural Networks 1990;3:109–18.

[74] Parzen E. On estimation of a probability density function and mode. Ann Math Stat 1962;33(3):1065–76.

[75] Giurgiutiu V., Current issues in vibration-based fault diagnostics and prognostics. In: SPIE's ninth annual international symposium on smart structures and materials and seventh annual international symposium on NDE for health monitoring and diagnostics, San Diego, CA; March 17–21, 2002.

[76] Mao KZ, Tan K-C, Ser W. Probabilistic neural-network structure determination for pattern classification. IEEE Tran Neural Networks 2000;11(4):1009–16.

[77] Khawaja, T, Vachtsevanos, G, Wu, B., Reasoning about uncertainty in prognosis: a confidence prediction neural network approach, 2005. In: Annual meeting of the north American Fuzzy Information Processing Society, Ann Arbor, MI; June 22–25, 2005.

[78] Khosravi A, Nahavandi S, Creighton D, Atiya AF. Comprehensive review of neural network-based prediction intervals and new advances. IEEE Tran Neural Networks 2011;22(9):1341–56.

[79] Gelman A, Carlin JB, Stern HS, Rubin DB. Bayesian data analysis. NY: Chapman and Hall; 2004.

[80] Santner TJ, Williams BJ, Notz WI. The design and analysis of computer experiments. NY: Springer Verlag; 2003.

[81] Rasmussen CE, Williams CKI. Gaussian processes for machine learning. Cambridge, MA: The MIT Press; 2006.

[82] Williams CKI. Computing with infinite networks. In: Mozer MC, Jordan MI, Petsche T, editors. Advances in neural information processing systems, vol. 9. Cambridge, MA: MIT Press; 1997.

[83] Mohanty S, Das D, Chattopadhyay A, Peralta P. Gaussian process time series model for life prognosis of metallic structures. J Intell Mater Syst Struct 2009;20:887–96.

[84] Paciorek C, Schervish MJ. Nonstationary covariance functions for Gaussian process regression. In: Thrun S, Saul L, Schölkopf B, editors. Advances in neural information processing systems, vol. 16. MIT Press; 2004.

[85] Belhouari SB, Bermak A. Gaussian process for nonstationary time series prediction. Comput Stat Data Anal 2004;47:705–12.

[86] Liu, D, Pang, J, Zhou, J, Peng, Y., Data-driven prognostics for lithium-ion battery based on Gaussian process regression, 2012. In: Prognostics and system health management conference, Beijing, China; May 23–25, 2012.

[87] Sacks J, Welch WJ, Mitchell TJ, Wynn HP. Design and analysis of computer experiments. Stat Sci 1989;4(4):409–23.

[88] An D, Choi JH. Efficient reliability analysis based on Bayesian framework under input variable and metamodel uncertainties. Struct Multi Optim 2012;46:533–47.

[89] Neal RM. Regression and classification using Gaussian process priors. In: Bernardo JM, Berger JO, Dawid AP, Smith AFM, editors. Bayesian statistics, vol. 6. NY: Oxford University Press; 1998. p. 475–501.

[90] MacKay, DJC., Gaussian processes-a replacement for supervised neural networks? Tutorial lecture notes for NIPS, UK, ⟨http://www.inference.phy.cam.ac.uk/mackay/BayesGP.html⟩; 1997.

[91] Lawrence N, Seeger M, Herbrich R. Fast sparse Gaussian process methods: the information vector machine. In: Becker S, Thrun S, Obermayer K, editors. Advances in neural information processing systems, vol. 15. MIT Press; 2003. p. 625–32.

[92] Foster L, Waagen A, Aijaz N, Hurley M, Luis A, Rinsky J, et al. Stable and efficient Gaussian process calculations. J Mach Learn Res 2009;10:857–82.

[93] Melkumyan, A, Ramos, F., "A sparse covariance function for exact Gaussian process inference in large datasets. In: IJCAI 2009, proceedings of the 21st International joint conference on artificial intelligence, Pasadena, CA., July 11–17; 2009, pp. 1936–1942.

[94] Sang H, Huang JZ. A full scale approximation of covariance functions for large spatial data sets. J R Stat Soc Ser B (Stat Methodol) 2012;74(1):111–32.

[95] Gramacy RB, Lee HK. Cases for the nugget in modeling computer experiments. Stat Comput 2012;22(3):713–22.

[96] Andrianakis Ioannis, Challenor Peter G. The effect of the nugget on Gaussian process emulators of computer models. Comput Stat Data Anal 2012;56:4215–28.

[97] Oden JT, Prudencio EE, Bauman PT. Virtual model validation of complex multiscale systems: applications to nonlinear elastostatics. Comput Meth Appl Mech Eng 2013;266:162–84.

[98] Rebba R, Mahadevan S, Huang S. Validation and error estimation of computational models. Reliab Eng Syst Saf 2006;91:1390–7.

[99] Sargent RG. Verification and validation of simulation models. J Simul 2013;7:12–24.

[100] Ling Y, Mahadevan S. Quantitative model validation techniques: new insights. Reliab EngSyst Saf 2013;111:217–31.

[101] Coppe A, Pais MJ, Haftka RT, Kim NH. Using a simple crack growth model in predicting remaining useful life. J Aircr 2012;49(6):1965–73.

[102] Kalman RE. A new approach to linear filtering and prediction problems. Trans ASME-J Basic Eng 1960;82:35–45.

[103] He, W, Williard, N, Osterman, M, Pecht, M., Prognostics of lithium-ion batteries using extended Kalman filtering. In: Proceedings of IMAPS advanced technology workshop on high reliability microelectronics for military applications, Linthicum Heights, MD, USA; May 2011, pp. 17–19.

[104] Doucet A, De Freitas N, Gordon NJ. Sequential Monte Carlo methods in practice. Springer-Verlag; 2001.

[105] Choi JH, An D, Gang J, Joo J, Kim NH. Bayesian approach for parameter estimation in the structural analysis and prognosis. In: Proceedings of the annual conference of the prognostics and health management society, Portland, OR; October 13–16, 2010.

[106] Bayes T. An essay towards solving a problem in the doctrine of chances. Philos Trans R Soc London 1763;53:370–418.

[107] Orchard ME, Vachtsevanos GJ. A particle filtering approach for on-line failure prognosis in a planetary carrier plate. Int J Fuzzy Logic Intell Syst 2007;7(4):221–7.

[108] Daigle, M, Goebel, K., Multiple damage progression paths in model-based prognostics. In: IEEE Aerospace conference, Big Sky, Montana; 2011.

[109] An D, Choi JH, Schmitz TL, Kim NH. In-situ monitoring and prediction of progressive joint wear using Bayesian statistics. Wear 2011;270(11–12):828–38.

[110] An Dawn, Choi Joo-Ho, Kim Nam H. Prognostics 101: a tutorial for particle filter-based prognostics algorithm using Matlab. Reliab EngSyst Saf 2013;115:161–9 (07).

[111] Higuchi T. Monte Carlo filter using the genetic algorithm operators. J Stat Comput Simul 1997;59(1):1–23.

[112] Kitagawa G. Non–Gaussian state space modeling of nonstationary time series (with Discussion),. J Am Stat Assoc 1987;82(400):1032–63.

[113] Wang, WP, Liao, S, Xing, TW., Particle filter for state and parameter estimation in passive ranging. In: IEEE international conference on intelligent computing and intelligent systems. Shanghai, China; 2009.

[114] Gilks WR, Berzuini C. Following a moving target-Monte Carlo inference for dynamic Bayesian models. R Stat Soc B 2001;63(Part 1):127–46.

[115] Andrieu C, Freitas de N, Doucet A, Jordan M. An introduction to MCMC for machine learning. Mach Learn 2003;50(1):5–43.

[116] Kim S, Park JS. Sequential Monte Carlo filters for abruptly changing state estimation. Probab Eng Mech 2011;26:194–201.

[117] Merwe, Rvd, Doucet, A, Freitas, N. de, Wan, E., The unscented particle filter. In: NIPS; 2000, pp. 584–590.

[118] Miao Q, Xie L, Cui H, Liang W, Pecht M. Remaining useful life prediction of lithium-ion battery with unscented particle filter technique. Microelectron Reliab 2013;53:805–10.

[119] Rubin DB. Using the SIR algorithm to simulate posterior distributions. In: Bernardo JM, DeGroot MH, Lindley DV, Smith AFM, editors. Bayesian statistics, vol. 3. Cambridge, MA: Oxford University Press; 1998. p. 395–402.

[120] An D, Choi JH. Improved MCMC method for parameter estimation based on marginal probability density function. J Mech Sci Technol 2013;2(6):1771–9.

[121] Gelfand AE, Sahu SK. On Markov Chain Monte Carlo acceleration. J Comput Graph Stat 1994;3:261–76.

[122] An D, Choi JH, Kim NH. Identification of correlated damage parameters under noise and bias using Bayesian inference. Struct Health Monit 2012;11(3):293–303.

[123] Gu, J, Azarian, MH, Pecht, MG., Failure prognostics of multilayer ceramic capacitors in temperature-humidity-bias conditions 2008. In: International conference on prognostics and health management, Denver, Colorado; October 6–9, 2008.

[124] Coppe, A, Haftka, RT, Kim, NH, Yuan, FG., Reducing uncertainty in damage growth properties by structural health monitoring. In: Annual conference of the prognostics and health management society, San Diego, CA; September 27–October 1, 2009.

[125] Guan, X, Liu, Y, Saxena, A, Celaya, J, Goebel, K., Entropy-based probabilistic fatigue damage prognosis and algorithmic performance comparison. In: Annual conference of the prognostics and health management society, San Diego, CA; September 27–October 1, 2009.

[126] Zhang B, Khawaja T, Patrick R, Vachtsevanos G, Orchard ME, Saxena A. Application of blind deconvolution denoising in failure prognosis. IEEE Tran Instrum Meas 2009;58(2):303–10.

[127] Qiua H, Lee J, Linc J, Yu G. Robust performance degradation assessment methods for enhanced rolling element bearing prognostics. Adv Eng Inf 2003;17:127–40.

[128] Abouel-seoud SA, Elmorsy MS, Dyab ES. Robust prognostics concept for gearbox with artificially induced gear crack utilizing acoustic emission. Energy Environ Res 2011;1(1):81–93.

[129] Paris PC, Erdogan F. A critical analysis of crack propagation laws. Trans ASME, J Basic Eng Ser D 1963;85(3):528–34.

[130] Huang X, Torgeir M, Cui W. An engineering model of fatigue crack growth under variable amplitude loading. Int J Fatigue 2008;30(1):2–10.

[131] An, D, Choi, JH, Kim, NH., A comparison study of methods for parameter estimation in the physics-based prognostics. In: 53rd AIAA/ASME/ASCE/AHS/ASC structures, structural dynamics and materials conference, Honolulu, Hawaii; April 23–26, 2012.

[132] Storvik G. Particle filters in state space models with the presence of unknown static parameters. IEEE Tran Signal Process 2002;50(2):281–9.