Artificial Intelligence for Engineering Design, Analysis and Manufacturing (2001), *15*, 349–365. Printed in the USA.
Copyright © 2001 Cambridge University Press 0890-0604/01 $12.50

Fault prognostics using dynamic wavelet neural networks

PENG WANG AND GEORGE VACHTSEVANOS
School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332-0250, USA

(RECEIVED October 27, 2000; ACCEPTED November 27, 2000)

Abstract

Modern industry is concerned about extending the lifetime of its critical processes and maintaining them only when required. Significant aspects of these trends include the ability to diagnose impending failures, prognosticate the remaining useful lifetime of the process and schedule maintenance operations so that uptime is maximized. Prognosis is probably the most difficult of the three issues leading to condition-based maintenance (CBM). This paper attempts to address this challenging problem with intelligence-oriented techniques, specifically dynamic wavelet neural networks (DWNNs). DWNNs incorporate temporal information and storage capacity into their functionality so that they can predict into the future, carrying out fault prognostic tasks. Such fundamental issues as the network structure, learning algorithms, stability analysis, uncertainty management, and performance assessment are studied in a theoretical framework. An example is presented in which a trained DWNN successfully prognoses a defective bearing with a crack in its inner race.

Keywords: Condition-Based Maintenance; Fault Diagnosis; Fault Prognosis; Neural Networks; Wavelets

1. INTRODUCTION

The manufacturing and industrial sectors of our economy are increasingly called to produce at higher throughput and better quality while operating their processes at maximum yield. As manufacturing facilities become more complex and highly sophisticated, the quality of the production phase has become more crucial. The manufacture of such typical products as aircraft, automobiles, appliances, medical equipment, and so forth, involves a large number of complex processes, most of which are characterized by highly nonlinear dynamics coupling a variety of physical phenomena in the temporal and spatial domains. It is not surprising, therefore, that these processes are not well understood and their operation is "tuned" by experience rather than through the application of scientific principles. Machine breakdowns are common, limiting uptime in critical situations. Failure conditions are difficult and, in certain cases, almost impossible to identify and localize in a timely manner. Scheduled maintenance practices tend to reduce machine lifetime and increase downtime, resulting in loss of productivity. Recent advances in instrumentation, telecommunications, and computing are making available to manufacturing companies new sensors and sensing strategies, plant-wide networking and information technologies that are assisting in improving substantially the production cycle. Machine diagnostics/prognostics for conditional-based maintenance (CBM) involves an integrated system architecture with a diagnostic module—the diagnostician—which assesses through on-line sensor measurements the current state of critical machine components, a prognostics module—the prognosticator—which takes into account input from the diagnostician and decides upon the need to maintain certain machine components on the basis of historical failure rate data and appropriate fault models, and a maintenance scheduler whose task is to schedule maintenance operations without affecting adversely the overall system functionalities of which the machine in question is only one of its constituent elements.

This paper addresses issues relating to the prognostic module—the Achilles heel of the CBM architecture. Fault diagnosis is a mature field with contributions ranging from model-based techniques to data-driven configurations that capitalize upon soft computing and other "intelligent" tools (Konrad & Isermann, 1996; Mylaraswamy & Venkatasubramanian, 1997). CBM scheduling is a complex task that involves finding the "optimum" time to perform maintenance within the window prescribed by the Prognosticator while

Reprint requests to: George Vachtsevanos, School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332-0250, USA. E-mail: gjv@ece.gatech.edu

https://doi.org/10.1017/S0890060401154089 Published online by Cambridge University Press

meeting a host of constraints. This scheduling problem may be formulated as a multiobjective optimization problem where the main objective is to maximize process uptime while satisfying a set of constraints that relate to resource and maintenance personnel availability, production and scheduling requirements, redundant or relocatable machines, timing constraints, and so forth (Barbera et al., 1996; Makis et al., 1998; Prickett & Eavery, 1991). The word "prognosis" implies the foretelling of the probable course of a disease (Taylor, 1953), a term widely used in medical practice. In the industrial and manufacturing arenas, prognosis is interpreted to answer the question: What is the remaining useful lifetime of a machine or a component once an impending failure condition is detected and identified? Stochastic Auto-Regressive Integrated Moving Average (ARIMA) models (Jardim-Goncalves et al., 1996), fuzzy pattern recognition principles (Frelicot, 1996), knowledge-intensive expert systems (Lembessis et al., 1989), nonlinear stochastic models of fatigue crack dynamics (Ray & Tangirala, 1994), polynomial neural networks (Parker et al., 1993), Weibull models (Groer, 2000), and other techniques have been introduced over the past years to address the diagnostic/prognostic problem. This paper attempts to address this issue by introducing a novel combination of a "virtual" sensor as a mapping tool between known measurements and "difficult-to-access" quantities and a dynamic wavelet neural network as the "predictor," that is, the construct that projects into the future the temporal behavior of a faulted component.

2. PROGNOSTICATION

Prognosticators perform the vital function of linking the diagnostic information with the maintenance scheduler. They are probably the least understood but most crucial component of the diagnostic/prognostic/CBM hierarchical architecture. Furthermore, they entail ambiguity and large-grain uncertainty, since the historical evolution of a failure event—the growth of a structural fault, for example—is difficult if not impossible to model accurately, historical data is not readily available, and the particular growth phenomenon may be strongly dependent on the system structure, operating conditions, environmental effects, and so forth. They are viewed as dynamic predictors that receive fault data from the diagnostic module and determine the allowable time window during which machine maintenance must be performed if the integrity of the process is to be kept as high as possible. The term "dynamic predictor" implies also the functional requirement that the target output, that is, remaining useful lifetime or time-to-failure, is dynamically updated as more information becomes available from the diagnostician. Thus, this scheme should reduce the uncertainty and improve the prediction accuracy as the accumulated evidence grows.

Figure 1 depicts the overall architecture of the proposed prognostic system. The diagnostician monitors continuously critical sensor data and decides upon the existence of impending or incipient failure conditions. The detection and identification of an impending failure triggers the prognosticator. The latter reports to the CBM module primarily the remaining useful lifetime of the failing machine or component. The CBM module schedules the maintenance so that uptime is maximized while certain constraints are satisfied. The schematic of Figure 1 focuses on the functionalities of the prognosticator. The diagnostician alerts the prognostic module and provides failure and other pertinent sensor data to it. The prognostic architecture is based on two constructs: a static "virtual sensor" that relates known measurements to fault data and a predictor which attempts to project the current state of the faulted component into the future, thus revealing the time evolution of the failure mode and allowing the estimation of the component's remaining useful lifetime. Both constructs rely upon a Wavelet Neural Network (WNN) model acting as the mapping tool.

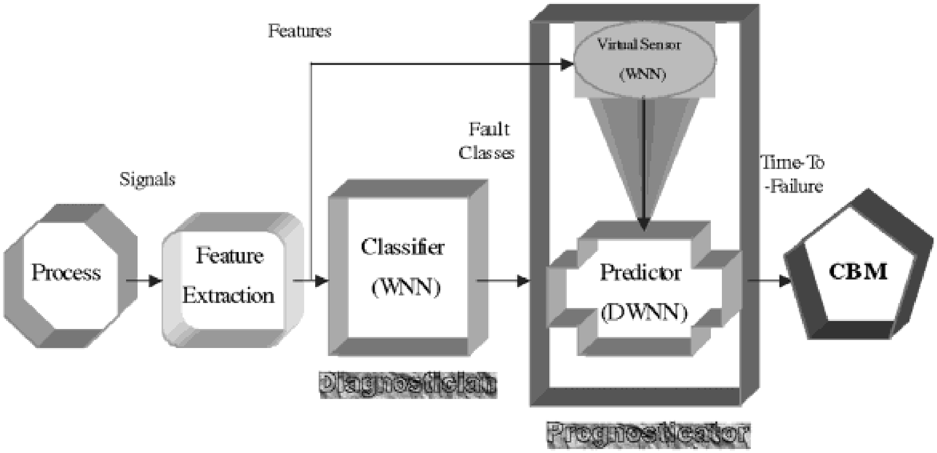

Fig. 1. The overall architecture of the prognostic system.

https://doi.org/10.1017/S0890060401154089 Published online by Cambridge University Press

2.1. Wavelet neural networks

WNNs belong to a new class of neural networks with unique capabilities in addressing identification and classification problems. Wavelets are a class of basic elements with oscillations of effectively finite duration that makes them like "little waves." The self-similar, multiple resolution nature of wavelets offers a natural framework for the analysis of physical signals and images. On the other hand, artificial neural networks constitute a powerful class of nonlinear function approximants for model-free estimation. A common ground between these two technologies may be coherently exploited by introducing a WNN. Indeed, the implementation of a neural network is closely related to a truncated version of the wavelet series.

A multi-input multi-output (MIMO) WNN is illustrated in Figure 2, which has only one hidden layer. This WNN can be formulated, in a vector format, as (Schauz, 1996):

$$y = [\psi_{A_1,b_1}(x)\psi_{A_2,b_2}(x)\cdots\psi_{A_M,b_M}(x)]C + [x1]C_{lin}, \qquad (1)$$

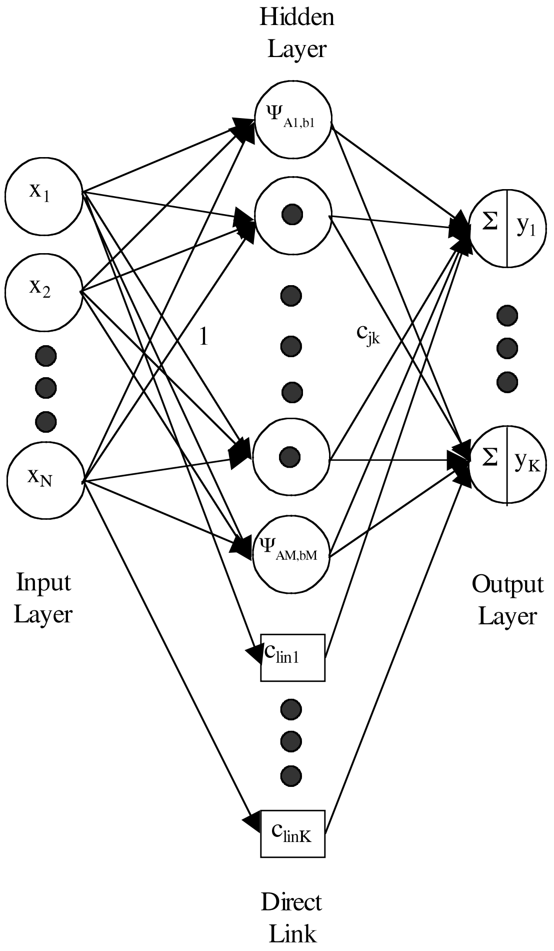

Fig. 2. A WNN.

where x is the $1 \times N$ input row-vector; y is the $1 \times K$ output row-vector, and K is the number of outputs; A_j is the $N \times N$ squashing matrix for the jth node; b_j is the $1 \times N$ translation vector for the jth node; C is the $M \times K$ matrix of output coefficients, where M is the number of wavelet nodes; C_{lin} is the $(N+1) \times K$ matrix of output coefficients for the linear direct link; and ψ is the wavelet function that can take the form:

$$\psi_{A,b}(x) = |A|^{1/4}\psi(\sqrt{(x-b)A(x-b)^T}), \qquad (2)$$

where x is the input row-vector, A the squashing matrix for the wavelet, b the translation vector, and T the transpose operator. Composed of localized basis functions, the WNNs are suitable for capturing the local nature of the data patterns and thus are efficient tools for both classification and approximation problems.

Equation (1) is a static model in the sense that it establishes a static relation between its inputs and outputs. All signals flow in a forward direction only with this configuration. Dynamic or recurrent neural networks, on the other hand, are required to model the time evolution of dynamic systems. Signals in such a network configuration can flow not only in the forward direction but also can propagate backwards, in a feedback sense, from the output to the input nodes. Dynamic wavelet neural nets have recently been proposed to address the prediction/classification issues. A multiresolution dynamic predictor that utilizes the discrete wavelet transform and recurrent neural networks forming nonlinear models for prediction was designed and employed for multistep prediction of the intracranial pressure signal (Tsui et al., 1995). A recurrent wavelet neural network was developed for the blind equalization of nonlinear communication channels (He & He, 1997); recurrent wavelet neural networks were also derived by Rao and Kumthekar (1994) using the real-time Back-Propagation (BP) algorithm.

The basic structure of a DWNN is shown in Figure 3. Delayed versions of the input and output augment now the

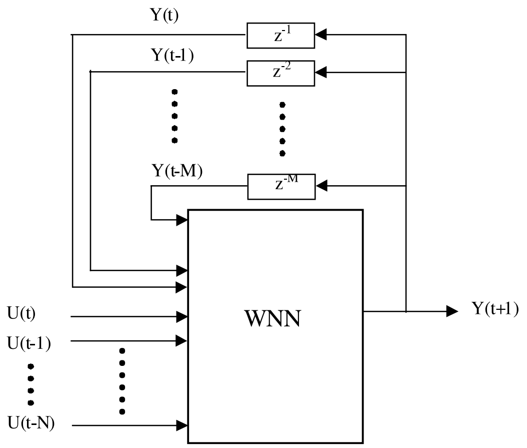

Fig. 3. A DWNN.

input feature vector and the resulting construct can be formulated as

$$Y(t+1) = WNN(Y(t),\ldots,Y(t-M),U(t),\ldots,U(t-N)), \quad (3)$$

where U is the external input, Y is the output, M is the number of outputs minus 1, N is the number of external inputs minus 1, and WNN stands for a static WNN. When measuring the angular velocity of a servo motor, for example, $Y(t)$ could be the velocity to be measured and $U(t)$ be the regulating voltage that controls the motor's rotation at time t. Equation (3) forms an evolving or prediction model that dynamically maps the historical and current data into the future. In essence, this DWNN is a kind of partially recurrent WNNs of simple but well-defined structures that are more convenient to design and apply in practical situations than the fully recurrent WNNs. Compared to traditional prediction techniques such as ARIMA, DWNNs offer, in a systematic manner, more flexibility in terms of nonlinear mapping, parallel processing, heuristics-based learning, and hardware implementation.

2.2. Learning algorithms

The DWNN described by Eq. (3) can be trained in a time-dependent way, using either a gradient-descent technique like the Levenberg–Marquardt (LM) algorithm or an evolutionary one such as the Genetic Algorithm (GA). Here, a learning algorithm that evolves over time is derived using the traditional gradient-descent technique. For simplicity, assume that in Eq. (3), Y is 1×1 in dimension, U ignored, $M = N = 0$, and WNN composed of Eqs. (1) and (2) but without C_{lin}, which results in a simplified DWNN to be formulated, in a summation format, as

$$y(t+1) = \sum_{j=1}^{M} c_j \psi_j(\|q_j y(t) - b_j\|_{a_j}), \quad (4)$$

where y is a single output; a_j, b_j, q_j, c_j are all real numbers and a_j must be nonnegative. The radial distance is designated as

$$\|q_j y(t) - b_j\|_{a_j} = \sqrt{a_j}\,|q_j y(t) - b_j|. \quad (5)$$

A cost function at time $t+1$ is defined as

$$E(t+1) = \tfrac{1}{2}[\bar{y}(t+1) - y(t+1)]^2. \quad (6)$$

With respect to a_j, taking derivatives on both sides of Eq. (6) produces

$$\frac{\partial E(t+1)}{\partial a_j} = \frac{\partial}{\partial a_j}\left\{\frac{1}{2}[\bar{y}(t+1) - y(t+1)]^2\right\} = -\frac{\partial y(t+1)}{\partial a_j}. \quad (7)$$

From Eq. (4), it follows that

$$\begin{aligned}\frac{\partial y(t+1)}{\partial a_j} &= \frac{\partial}{\partial a_j}\left[\sum_{j=1}^{M} c_j \psi_j(\|q_j y(t) - b_j\|_{a_j})\right]\\ &= \sum_{j=1}^{M} c_j \psi_j'(\|q_j y(t) - b_j\|_{a_j}) \frac{\partial}{\partial a_j}(\|q_j y(t) - b_j\|_{a_j})\\ &= \sum_{j=1}^{M} c_j \psi_j'(\|q_j y(t) - b_j\|_{a_j})\\ &\quad \times \left[\frac{|q_j y(t) - b_j|}{2\sqrt{a_j}} + \sqrt{a_j}\,\mathrm{Sign}(q_j(t) - b_j) q_j \frac{\partial y(t)}{\partial a_j}\right].\end{aligned} \quad (8)$$

From the above equation, it can be observed that $[\partial y(t+1)]/\partial a_j$ is a function of $y(t)$ and $\partial y(t)/\partial a_j$. Thus, the equation can be used iteratively over time in order to generate new gradients for minimizing the cost function (6). Similarly, the other needed gradients can be obtained as follows:

$$\begin{aligned}\frac{\partial E(t+1)}{\partial b_j} &= \sum_{j=1}^{M} c_j \psi_j'(\|q_j y(t) - b_j\|_{a_j})\sqrt{a_j}\,\mathrm{Sign}(q_j y(t) - b_j)\\ &\quad \times \left(q_j \frac{\partial y(t)}{\partial b_j} - 1\right),\end{aligned} \quad (9)$$

$$\begin{aligned}\frac{\partial E(t+1)}{\partial c_j} &= \sum_{j=1}^{M} \Bigg[c_j \psi_j'(\|q_j y(t) - b_j\|_{a_j})\sqrt{a_j}\,\mathrm{Sign}(q_j y(t) - b_j) q_j\\ &\quad \times \frac{\partial y(t)}{\partial c_j} + \psi_j(\|q_j y(t) - b_j\|_{a_j}\Bigg],\end{aligned} \quad (10)$$

$$\begin{aligned}\frac{\partial E(t+1)}{\partial q_j} &= \sum_{j=1}^{M} c_j \psi_j'(\|q_j y(t) - b_j\|_{a_j})\sqrt{a_j}\,\mathrm{Sign}(q_j y(t) - b_j)\\ &\quad \times \left(y(t) + q_j \frac{\partial y(t)}{\partial q_j}\right).\end{aligned} \quad (11)$$

It is assumed that the initial condition is independent of the network parameters, that is,

$$\frac{\partial y(t)}{\partial a_j} = \frac{\partial y(t)}{\partial b_j} = \frac{\partial y(t)}{\partial c_j} = \frac{\partial y(t)}{\partial q_j} = 0. \quad (12)$$

Then, there is no substantial difficulty in running these gradient equations so as to derive the required gradients, with which a gradient-based adaptation scheme, such as a δ-learning rule or the LM algorithm, can readily be implemented.

2.3. Stability analysis

Stability is a critical concept in system theory that describes the ability of a system to stay at a point or in a region in its state space. Equivalently, this concept is of considerable importance for system identification and modeling because it has great impact on the accuracy and sensitivity of the learning algorithms involved. With reference to its characteristics, stability may be categorized as the Lyapunov stability or the bounded-input bounded-output (BIBO) stability. With respect to the excitation sources, stability can be divided into structural stability and total stability. Here, the BIBO stability and Lyapunov stability are examined for the DWNN.

Equation (4) can be written with an additional term as

$$y(t+1) = \sum_{j=1}^{M} c_j \psi_j(\|Q_j x(t) - b_j\|_{A_j}) + p(x(t)), \tag{13}$$

where a single output y is considered and x is the input vector to the DWNN, which, for simplicity, is expressed as:

$$x(t) = \begin{bmatrix} y(t) \\ u(t) \end{bmatrix}. \tag{14}$$

THEOREM 1. *System (13) is BIBO stable if and only if $p(x(t))$ is bounded.* ■

Proof: It is straightforward since the normalized wavelets $\psi_j(\|Q_j x(t) - b_j\|_{A_j}) \le 1$ for all t. ■

In situations such as modeling an unstable process, it is helpful to let $p(x(t))$ carry as much instability as possible so that the wavelet network can be trained without difficulty. Then, $p(x(t))$ can be designed using standard techniques for polynomial approximation.

In the Lyapunov sense, a system is considered to be stable if its total energy decreases monotonically towards an equilibrium point or region. Without losing generality, Eq. (13) can be written into a continuous form:

$$\dot{y}(t) = \sum_{j=1}^{M} c_j \psi_j(\|Q_j x(t) - b_j\|_{A_j}) + p(x(t)), \tag{15}$$

where, for simplicity, $p(x(t))$ will be neglected. An energy function $y^2(t)$ is selected for system (15). Clearly $y^2(t) \ge 0$. The derivative of the Lyapunov function is

$$\frac{d(y^2(t))}{dt} = 2y(t)\dot{y}(t) = 2y(t) \sum_{j=1}^{M} c_j \psi_j(\|Q_j x(t) - b_j\|_{A_j}). \tag{16}$$

Due to Theorem 1, $y(t)$ is bounded. From Eq. (16), it can be seen that $[d(y^2(t))]/dt \le 0$ if $\text{Sign}(c_j) = \text{Sign}(y(t))$, which can be considered as a design rule leading to an asymptotically stable DWNN.

2.4. Virtual sensors

It is often true that machine or component faults are not directly accessible for monitoring of their growth behavioral patterns. Consider, for example, the case of a bearing fault. No direct measurement of the crack dimensions is possible when the bearing is in an operational state. That is, there is no such device as a "fault meter" capable of providing direct measurements of the fault evolution. Examples of a similar nature abound. Marko et al. (1996) developed a neural net-based virtual or ideal sensor used to diagnose engine combustion failures, known as misfire detection. Their technique employs a recurrent neural net as the classifier that takes such inputs as crankshaft acceleration, engine speed, engine load, and engine ID and produces a misfire diagnostic evaluation as the output. In the present study, the same concept is exploited to design a virtual sensor which takes as inputs measurable quantities or features and outputs the time evolution of the fault pattern. A schematic representation of such a WNN as a virtual sensor is illustrated in Figure 4.

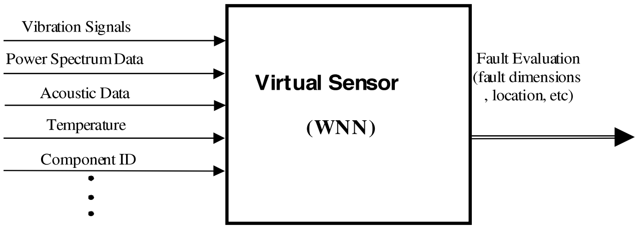

Fig. 4. A schematic representation of the WNN as a virtual sensor.

2.5. Predictors

Prediction of the course in which a fault could develop can be looked into from two different viewpoints: The first one is to locate the fault value at a certain time moment while the other is to find the time moment when the fault reaches a given value, that is, the fault dimensions reach a prespecified threshold. The latter appears to be more meaningful because it concentrates on revealing the critical time without requiring estimation of the whole time interval, thus resulting in a more efficient algorithm. The notion of Time-To-Failure (TTF) is the most important measure in prognosis. In fact, prognosis can be accomplished in either the time or frequency or even the event domain, since all of these domains are made up of ordered points.

A fault predictor based on the DWNN is shown in Figure 5. The process is monitored in real-time using appro-

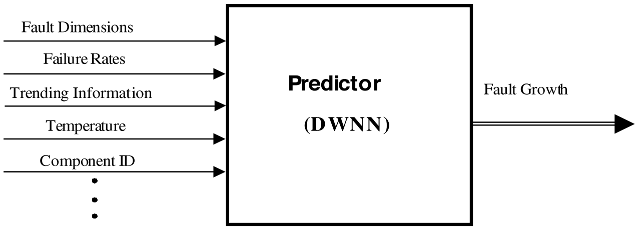

Fig. 5. A schematic representation of the DWNN as the predictor.

https://doi.org/10.1017/S0890060401154089 Published online by Cambridge University Press

priate sensors. Here, virtual sensors can also be employed to measure signals or their derivatives that are difficult to record on-line and on-site. Data obtained from measurements are continuously processed and features extracted on a time scale. The features are organized into a time-stamped feature vector that serves as the input to the DWNN. Consequently, the DWNN performs as a dynamic classifier or identifier. The data used to train the predictor must be recorded with time information, which is the basis for the prognosis-oriented prediction task. In the case of a bearing fault, the predictor could take the fault dimensions, failure rates, trending information, temperature, component ID, and so on as its inputs and generate the fault growth as the output. Feature extraction can be performed periodically for the processes under prognosis. It should be noted that features are extracted in temporal series and are dynamic in the sense that the DWNN processes them in a dynamic fashion. Then, the obtained features are fused into the time-dependent feature vector that characterizes the process at the designated time instants. Feature selection is based on criteria that distinguish a fault signature from normal operating conditions and one particular fault mode from another. Such other criteria as computational cost may be included.

The DWNN must be trained and validated before any on-line implementation and use. Such algorithms as the Back-Propagation or GA can be used to train the network. Once trained, the DWNN, along with the TTF calculation mechanism, can act as an on-line prognostic operator. It is worth reiterating that the results from the diagnosis serve as the input to the prognosis. Thus, the fidelity and accuracy of the diagnostician bears a direct impact on the reliability of the prognosticator. Predictions can be substantially improved as more fault data become available. The diagnostic/prognostic operation is viewed, therefore, as a dynamic, “evolving” mechanism with adaptive observation and prediction windows. More accurate predictions can result from the utility of additional historical information. The DWNN is, indeed, updated on-line in a real-time fashion.

3. UNCERTAINTY MANAGEMENT

Uncertainty and ambiguity are the rule rather that the exception in the diagnosis and prognosis of failure modes in practical systems. They manifest themselves at various levels of abstraction: at the data level, the feature level, the decision level, and classification level. As the prediction window increases, so does the uncertainty resulting from the levels of the data processing hierarchy. There are many potential root causes of uncertainty associated with fault conditions: Faults exhibit varying signatures depending upon the location, cause, prevailing operating conditions, and the state of the component materials. Detection and identification at an early stage of an incipient failure mode requires reliable and robust techniques for accurate declaration without false alarms. Prediction of the future behavior of a fault is much more demanding—essentially taxing severely the available means to quantify uncertainty. Prediction algorithms, therefore, must incorporate possibilistic (or probabilistic) quantifiers that inform the user of the expected time-to-failure as well as its anticipated variance (in terms of the earliest and latest time estimates). Fuzzy notions, such as fuzzy membership functions, are known to capture well uncertainty estimates and Dempster–Shafer theory may prove useful in combining conflicting evidence and supporting upper and lower bounds (plausibility and belief metrics) in these estimates.

Uncertainty representation and management for fault prognosis are difficult tasks, since prognosis involves both subjective and objective uncertainties and operates over the time horizon from the past, through the present, and to the future. Uncertainty sources must be identified and modeled. Uncertainty management schemes, that is, methods to reduce the uncertainty bounds as more data becomes available, must be derived. Probability and possibility theories are two candidates of mathematical tools to deal with these issues.

For simplicity, this paper deals only with data uncertainties and uses uncertainty boundaries for reporting prognostic results. This results in the so-called interval predictions, compared to point predictions. An uncertainty interval can be generated through the estimation of a lower and an upper bound of the prediction window. As shown in Figure 6, a fault indicated by the feature $F(t)$ would evolve along its mean $F_M(t)$ and within its lower bound $F_L(t)$ and upper bound $F_U(t)$. Hence, the fault prognosis problem can be stated as: using historic data of $F(t)$ to predict its mean $F_M(t)$ and boundaries $[F_L(t)F_U(t)]$ until the remaining useful lifetime or the time-to-failure of the targeted component is found with its mean T_M and confidence interval $[T_L T_U]$, under a certain failure criterion F_F. For example, a faulty servomotor could be prognosticated as having a remaining useful lifetime of around 15 hours, probably between 10 and 20 hours, under the criterion that the temperature of the

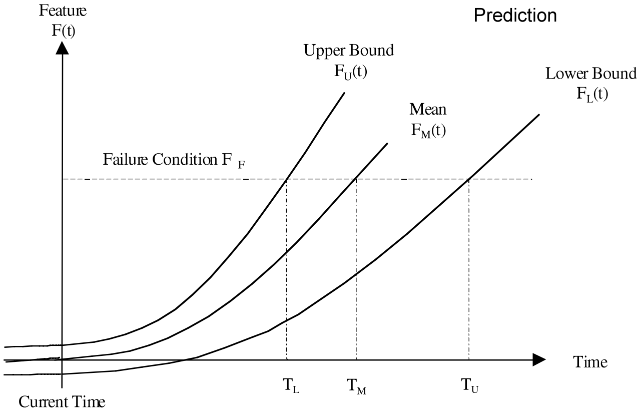

Fig. 6. Uncertainty boundaries in a prognostic task.

motor should not exceed 70°C. In this case, $T_M = 15$ days, $T_L = 10$ days, $T_U = 20$ days, and $F_F = 70$°C.

Generally, $F_M(t)$, $F_L(t)$, and $F_U(t)$ can be obtained by applying statistical or fuzzy clustering techniques to the given data. The mean $F_M(t)$ is the center of the data points at the time instant t. The lower bound $F_L(t)$ is the smallest data point at the time instant t. The upper bound $F_U(t)$ is the largest data point at the time instant t. The upper bound $F_U(t)$ and the lower bound $F_L(t)$ harness the development of the mean $F_M(t)$ so that the instantaneous feature $F(t)$ should appear to be moving in a band. However, it is not very applicable to allow $F_U(t)$ and $F_L(t)$ to be the extreme data points that are much less populated. A better way is to choose $F_U(t)$ and $F_L(t)$ using a confidence level, say, α to trim the probability density function (PDF) of $F(t)$, as shown in Figure 7.

For the cases where subjective uncertainties are involved, fuzzy autoregression (Lu & Xu, 1993) can be employed to establish possibilistic boundaries. A fuzzy linear predictor may be formulated as

$$\tilde{F}(t) = \tilde{\phi}_1^* \tilde{F}(t-1) + \tilde{\phi}_2^* \tilde{F}(t-2) + \cdots + \tilde{\phi}_p^* \tilde{F}(t-p) + \tilde{a}(t), \tag{17}$$

whose task is to minimize a threshold criterion in terms of quasi-conjunction degrees of estimated and observed values, where $\tilde{F}(t), \tilde{F}(t-1), \cdots, \tilde{F}(t-p)$ are the historic data, $\tilde{\phi}_1^*, \tilde{\phi}_2^*, \cdots, \tilde{\phi}_p^*$ are the estimated coefficients, $\tilde{a}(t)$ is the estimation error, and $\tilde{F}(t)$ is the predicted value. All variables in Eq. (17) can be expressed in fuzzy notions, for example, fuzzy L-R numbers. A fuzzy number is defined by a fuzzy membership function that has a peak value of 1 and monotonically descending spreads, that is, something like a PDF function in Figure 7. A fuzzy L-R number can be described by a linear function of the corresponding fuzzy variable. Once $\tilde{F}(t)$ is calculated, possibilistic boundaries of $\tilde{F}(t)$ can be derived using a number of truncation methods, the simplest of which is the α-cut technique similar to setting the $1-\alpha$ level confidence level in Figure 7.

For more effective fault prognosis, it is essential to know not only how uncertainties would propagate but also how they could be reduced or managed along the prognostic horizon. In the case of unsupervised prognosis, uncertainty reduction might only be achieved using higher prediction models that are expected to capture more reliable uncertainty tendencies from the available historic data. For supervised prognosis, there is, however, an opportunity to implement various uncertainty management mechanisms, due to the fact that new data will be available while the prognosis task proceeds. Fresh information on the process under prognosis can be utilized to update the structure as well as the parameters of the prognosticator so that it operates with shrinking or bounded uncertainty levels. Thus, on-line learning or adaptation is considered a major tool for harnessing uncertainties associated with fault prognostics.

Among many learning algorithms available in the literature, Reinforcement Learning (RL; Kaelbling et al., 1996) appears to be quite applicable in the fault prognosis arena. Its applicability lies in the fact that the algorithm itself interacts with its environment such that it can improve its performance gradually. Three primary approaches for implementing RL are Dynamic Programming (DP), Monte Carlo (MC) methods, and Temporal Difference (TD) Learning. For fault prognostics, TD is sometimes preferred because it is more relevant to time series prediction upon which most algorithms for fault prognostics depend. The simplest TD, known as TD(0), is given by Sutton (1988), which employs an intelligent agent responsible for updating the learning policies through exploration in its surroundings.

However, conventional RL methods appear to be cumbersome when used for on-line implementation of fault prognostic algorithms, since those algorithms usually prohibit too long exploration in the environment because of strict real-time requirements and low availability of new data. As shown in Figure 8, a new adaptation scheme is suggested, which is actually a combination of RL and GA. While accommodating evolutionary-type policy optimization, such adaptation mechanism embodies the main idea of RL, that is, to reward based on the behavior of the algorithm and update only when necessary. Figure 8, in fact, illustrates an

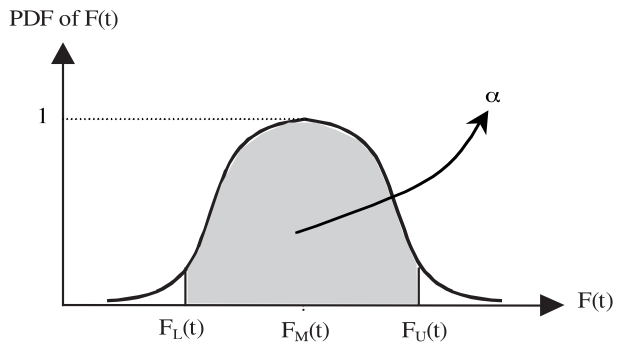

Fig. 7. An uncertainty interval with α-level confidence.

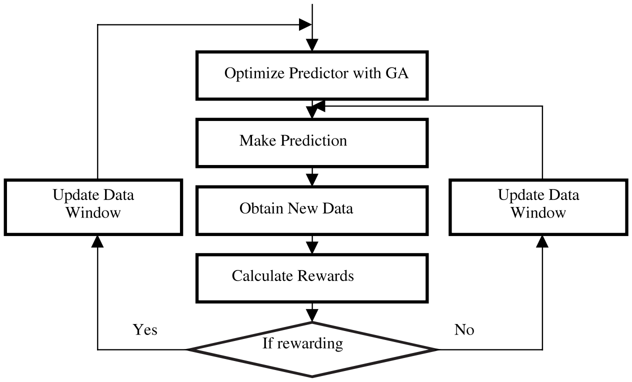

Fig. 8. An adaptation scheme for updating fault prognostic algorithms.

https://doi.org/10.1017/S0890060401154089 Published online by Cambridge University Press

on-line implementation procedure as follows: (1) Initialize the prognosticator; (2) Design a reward that should be related to the prognosticator's behavior; (3) Establish a rewarding criterion, for example, a threshold; (4) Select a training window for the prognosticator; (5) Optimize the prognosticator using GA within the current training window; (6) Use the trained prognosticator to predict the next step; (7) Calculate the reward when the new data is available; (8) Update the training window with the new data; (9) If rewarding, go to Step (5); if not, go to Step (6).

This procedure does not update the prognostic algorithm in every step. Smartly enough, it does that whenever necessary according to the current status of the algorithm and the condition of the environment. Thus, it avoids unnecessary algorithm alterations and becomes more efficient than those step-by-step learning algorithms.

4. PERFORMANCE ASSESSMENT

When a number of prognostic algorithms are available for a certain prognostic task, it is essential to compare these algorithms and select the best one for implementation so that the prognosis can be accomplished in a more efficient and effective manner, for example, how to rate a DWNN-based prognosticator against a traditional autoregression (AR) based one for the targeted application. It is essential, therefore, that means are devised to assess the performance of various prognostic algorithms. In general, an assessment methodology should consider both the technical and economic feasibility of the algorithms and their associated implementation platforms. Consequently, performance measures (PMs) should include the cost of equipment and maintenance, personnel expenses, accuracy of detection and prediction, and so on, which are usually grouped into two categories: those associated with economic factors and the ones relating to the technical (or algorithmic) concerns. In the second, accuracy, speed, complexity, and scalability are typical measures, whereas the first includes purchase and implementation costs, maintainability, computing resources, reliability, user-friendliness, among others.

Very limited information is currently available on performance assessment of prognostic algorithms. Recently, Essawy and Zein-Sabatto (1999) presented a number of performance metrics to evaluate diagnostic algorithms on system-level qualities as well as component- or subsystem-level behaviors. Several measures, such as failure rate, time delay, reliability, and so on, are defined and universal measures of effectiveness and performance are suggested for fault diagnostics but without detailing how these metrics can be applied to fault prognostics. Vachtsevanos et al. (1999) proposed a number of PMs specifically for fault prognostic algorithms. Technically, two fundamental classes are distinguished to address the prognostic assessment problem: The first one is concerned only with the final outcome (or target point) of an algorithm. It responds to the question of how close the output of the algorithm is to the target value. From a practical standpoint, it provides a measure of the deviation of the predicted time-to-failure from its measured value. The second, though, gauges if an algorithm can approach the target within specified bounds. Depending on the complexity of prognostic problems, more PM classes may have to be identified.

Measurable factors influencing the prediction performance can be exploited to carry out the comparison of various prognostic algorithms. Since historical data is essential for evaluating the performance of diagnostic and prognostic algorithms, it is assumed that a sufficient database is available that characterizes significant process variables and failure modes. With reference to Figure 9, where the graphical comparison of two output curves is depicted, a number of PMs can be proposed for fault prognosis as follows:

Prediction Target Error. This measure calculates how close to the target the output of an algorithm can arrive, which may be defined as

$$PM = \alpha \| y_r(n_f) - y_p(n_f) \| + \beta \| y_r(n_s) - y_p(n_s) \|, \quad (18)$$

where n_f indicates the time at the target and n_s stands for the start time; α and β are weighting factors accounting for any difference in initial conditions; when $\alpha = 1$ and $\beta = 0$, the same initial conditions are provided.

Prediction Behavior Error. This measure is employed to gauge if an algorithm can approach the target in an appropriate manner and may be defined as follows:

$$PM = \sum_{i=n_s}^{n_f} w(i) \| y_r(i) - y_p(i) \|, \quad (19)$$

where $w(i)$ is a weighting coefficient that takes into account the fact that the closer to the target, the more important the accuracy requirement becomes.

Prediction Error Rate. If a failure occurs at time T with an acceptable error range $\pm r$, then the Prediction Error

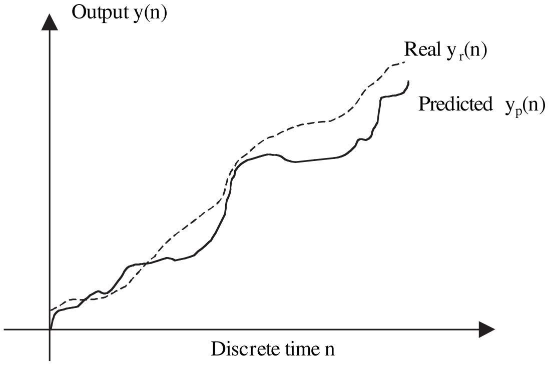

Fig. 9. Comparison of the predicted output with the real one.

https://doi.org/10.1017/S0890060401154089 Published online by Cambridge University Press

Rate or Wrong Output Rate (WOR), that is, the number of predictions outside the acceptable range $[T - r, T + r]$ over the total number of predicted outcomes may be expressed as

$$WOR = 1 - \frac{\sum_{i=1}^{N} DC(t_i)}{N}, \tag{20}$$

with a distribution of confidence values denoted by DC.

Prediction Similarity Error. The following measure is used when comparing more than one predicted time series against the real series from a starting point to a certain future point in time. This measurement can be calculated as:

$$SIMILARITY(x, y) = \sum_{i=1}^{N} \left(1 - \frac{|x_i - y_i|}{\max_i \ - \min_i}\right), \tag{21}$$

where x_i and y_i are two ith elements of two different time series; and are the maximum and minimum of all ith elements.

Overall Prediction Error. An overall performance metric can be formulated by aggregating appropriately key PMs in a linear scaled and weighted sum:

$$\begin{aligned} \textbf{Overall Performance} = {} & \mathbf{w_1 s_1}\textbf{Target-Error} \\ & + \mathbf{w_2 s_2}\textbf{Similarity-Error} \\ & + \dots, \end{aligned} \tag{22}$$

where $s_1, s_2, \dots,$ are scaling factors and $w_1, w_2, \dots,$ are weighting factors.

5. AN ILLUSTRATIVE EXAMPLE

Industrial chillers are typical processes found in many critical applications. These devices support electronics, communications, and so forth on a navy ship, computing and communication in commercial enterprises, refrigeration and other functions in food processing, and so on. Of special interest is the fact that their design incorporates a diverse assemblage of common and vital components, that is, pumps, motors, compressors, and so forth. A rich variety of failure modes are observed on such equipment ranging from vibration-induced faults to electrical failures and a multitude of process-related failure events. Most chillers are well instrumented, monitoring vibrations, temperature, pressure, flow, and so on, and many mechanical faults exhibit symptoms that are sensed via vibration measurements. For example, a water pump will vibrate if its motor bearing is defective, if its shaft is misaligned, or if its mounting is somewhat loose. A rolling-element bearing fault is used in this study to demonstrate the feasibility of the prognostic algorithms.

Defective bearings or loose mounting bolts would cause pumps to vibrate abnormally. The vibrations are typically monitored by accelerometers with the measured signals transferred to data acquisition units via coaxial cables. Shiroishi et al. (1997) collected triaxial vibration signals originating from a bearing with a crack in its inner race. An initial crack was seeded in the bearing and the experiment was run for a period of time and vibration data were recorded during that period. The setup was then stopped and the crack size was increased followed by a second run. This procedure was repeated until the bearing failed. The crack sizes were organized in an ascending order while time information was assumed uniformly distributed among the crack sizes. A training data set relating to the crack growth was thus obtained. Time segments of vibration signals from a good bearing and a defective one are shown in Figure 10. Their corresponding power spectral densities (PSDs) are shown in Figure 11. The original signals were windowed with each window containing 1000 time points. The maximum values of the vibration signals in each window were also recorded as shown in Figure 12 where x, y, and z represent the three Cartesian axes along which the accelerometer measures the vibrations. The PSDs of the windowed vibration signals were calculated and their peak values extracted as depicted in Figure 13. Figure 14 shows the corresponding crack sizes. Crack size information at intermediate points was generated via interpolation to avoid a large number of repeated experiments. There are 100 data points for each curve in the figures. The features chosen for prognosis are the maximum signal values and the maximum signal PSDs for all three axes, that is, (MaxSx MaxSy MaxSz) and (MaxPSDx MaxPSDy MaxPSDz).

Figure 14 demonstrates the crack growth as a function of time. The model was first trained using the fault data up to the 100th time window; from then on, it predicted the crack

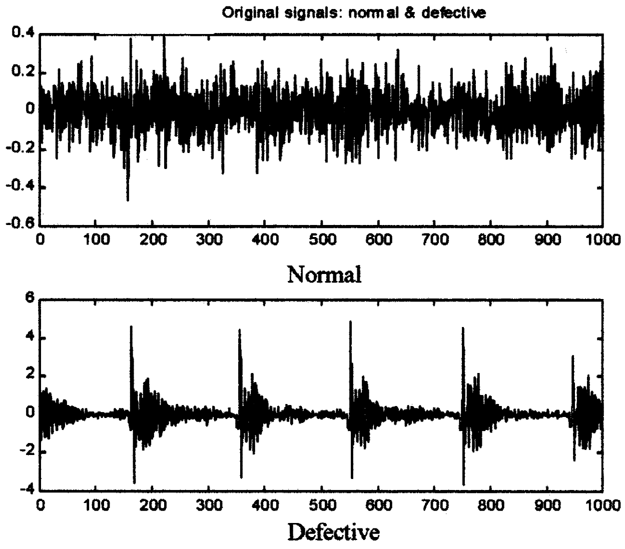

Fig. 10. Vibration signals from a normal and a defective bearing.

https://doi.org/10.1017/S0890060401154089 Published online by Cambridge University Press

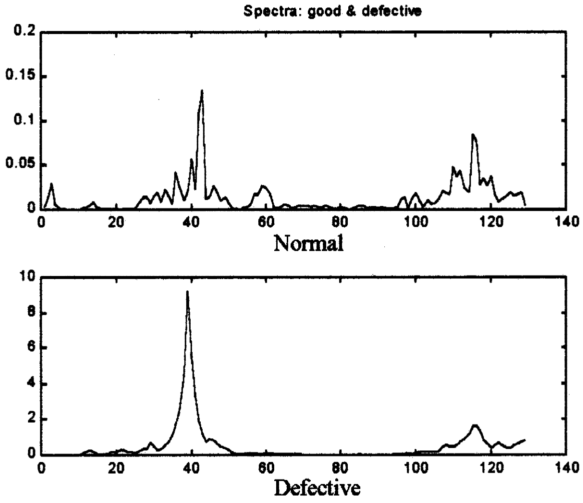

Fig. 11. PSDs of the vibration signals in Figure 10.

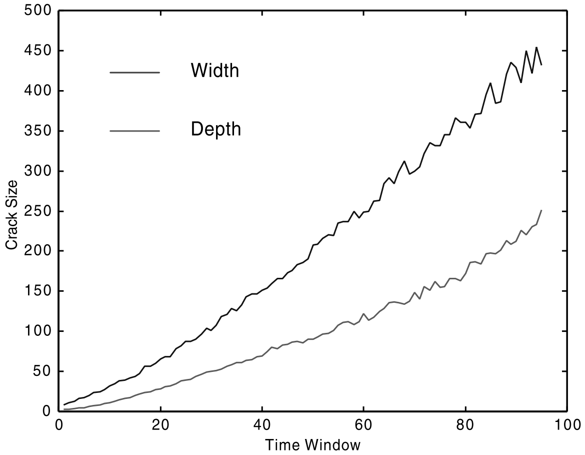

Fig. 14. The original crack sizes.

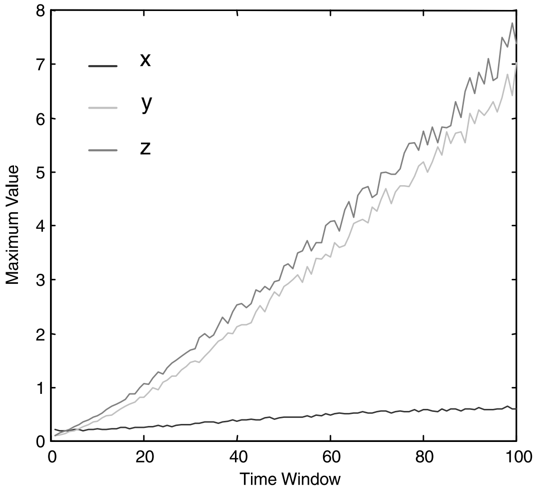

Fig. 12. The peak values of the original signals.

evolution until the final bearing failure. Mexican hats were used as mother wavelets throughout all the experiments. The virtual sensor, implemented as a WNN with seven hidden nodes or neurons, was trained through the process of Figure 15. This virtual sensor "measured" the crack size on the basis of the maximum signal amplitudes and the maximum signal PSDs as inputs. The training results are depicted in Figure 16. It is observed that 100 data points employed for training led to very satisfactory results. The DWNN, acting as the predictor, was trained next, as shown in Figure 17. The optimized training procedure resulted in a DWNN of six input (i.e., the model order is 6), eight hidden and two output neurons. The training took several hours to finish due to a large number of network parameters to be trained, which totaled 368 composed of $6 \times 6 \times 8$ for A, 6×8 for B, 2×8 for C and 2×8 for C_{lin}. The training results are shown in Figure 18. Training was deemed satisfactory when 100 data points were used. The trained predictor was

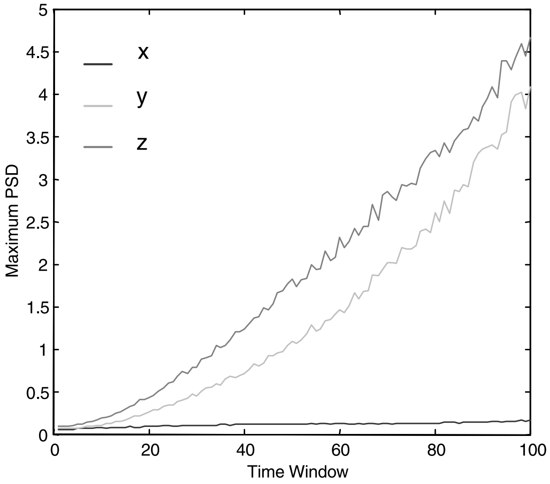

Fig. 13. The maximum PSDs of the original signals.

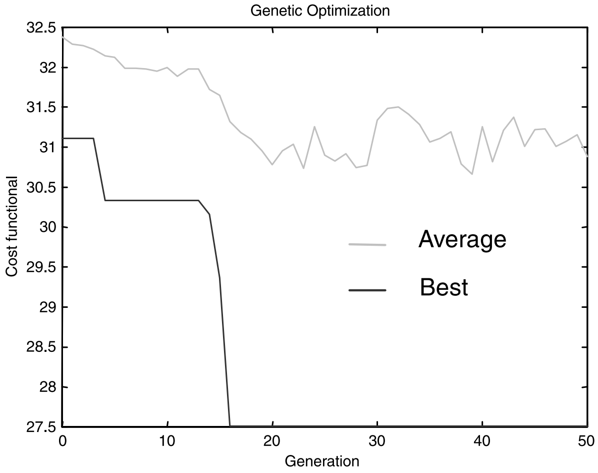

Fig. 15. The training of the virtual sensor.

https://doi.org/10.1017/S0890060401154089 Published online by Cambridge University Press

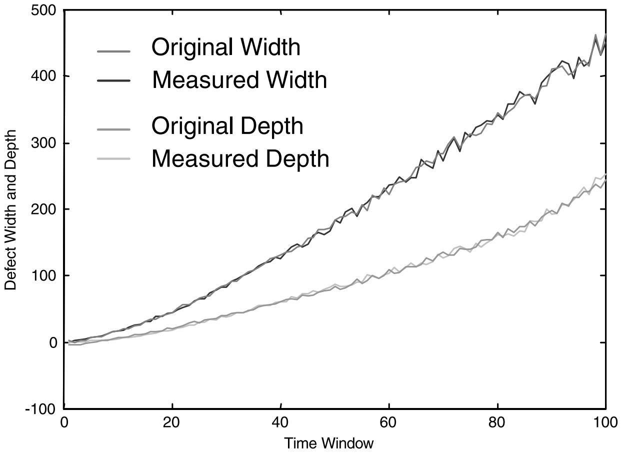

Fig. 16. The crack sizes measured by the trained virtual sensor.

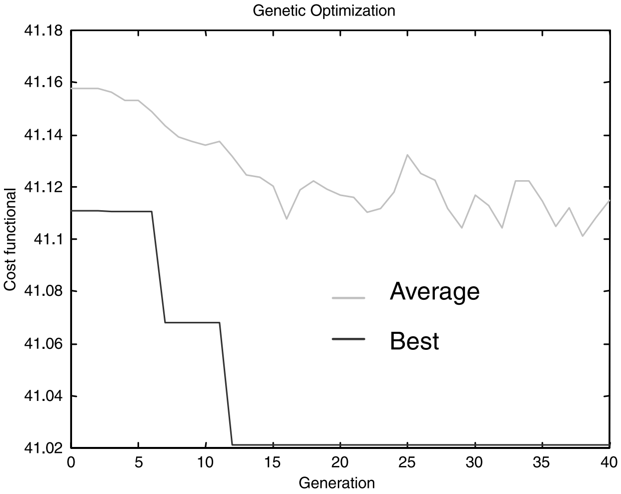

Fig. 17. The training of the predictor.

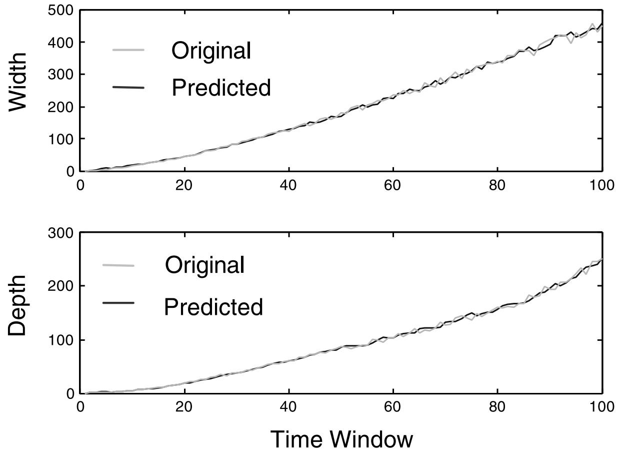

Fig. 18. The crack growth predicted by the trained predictor within 100th time window.

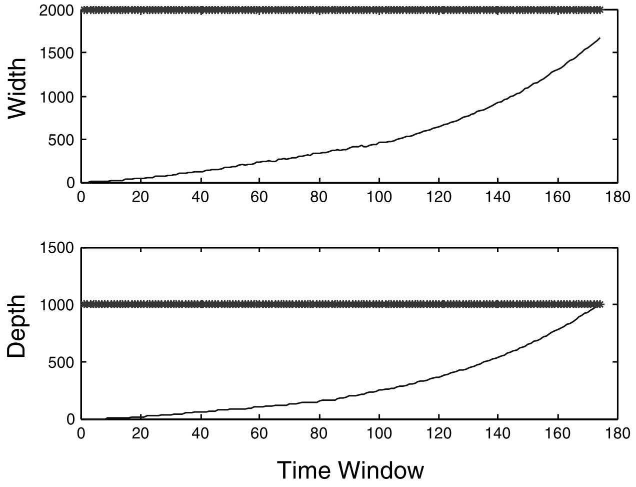

Fig. 19. The crack growth predicted by the trained predictor beyond 100th time window.

employed finally to predict the future crack development, as shown in Figure 19. A failure hazard threshold was established on the basis of empirical evidence corresponding to Crack_Width = 2000 microns or Crack_Depth = 1000 microns. The crack reached this hazard condition at the 174th time window. The Crack_Width criterion was reached first. It should be noted that these results are preliminary and intended only to illustrate the proposed prognostic architecture. In practices, a substantially large database may be required for feature extraction, training, validation, and optimization. Such a database will permit a series of sensitivity studies that may lead to more conclusive results as to the capabilities and effectiveness of the proposed method.

Suppose that the depth of the bearing crack had historically demonstrated its growth along its mean and within its lower and upper boundaries, as shown in Figure 20. A DWNN was then trained using this historical data in order to predict the development of the crack depth in terms of a mean value as well as an uncertainty value indicated by lower and upper boundaries. The training result is shown in Figure 20 as well. The trained DWNN was then employed to prognosticate the bearing crack. A failure threshold was set at 1000 μm. As shown in Figure 21, the mean value of the crack depth would reach the failure condition at the 176th time window. However, the lower and upper boundaries would arrive at the failure condition at the 181st and 169th time windows, respectively. Therefore, the remaining useful lifetime of the bearing is estimated to be about 76 time windows and in the range [69 81] time windows, counting from the current 100th time window.

In the vicinity of a growth curve for bearing vibration, nine more curves were generated by adding small random variations. The resulting 10 growth curves are shown in Figure 22. Initially, the prognosticator was trained using

https://doi.org/10.1017/S0890060401154089 Published online by Cambridge University Press

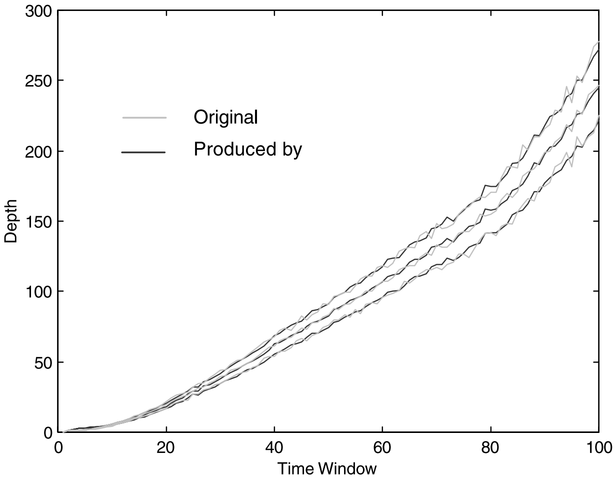

Fig. 20. The mean, lower, and upper boundaries of the crack depth.

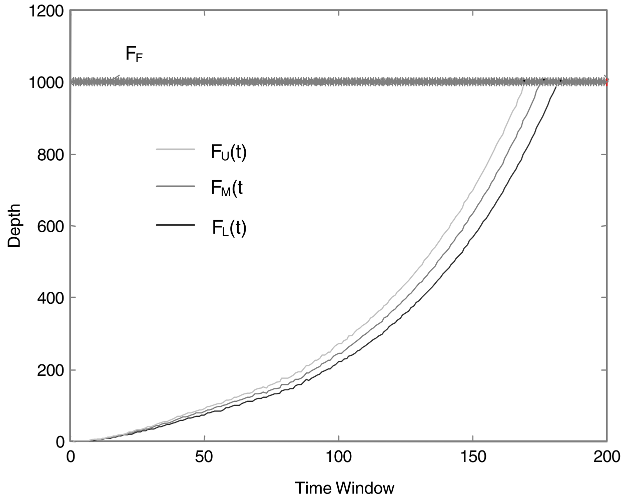

Fig. 21. Prognosis of the crack depth in terms of uncertainty intervals.

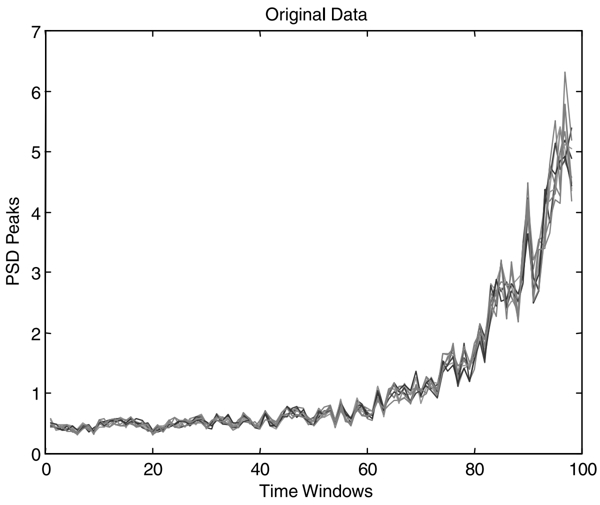

Fig. 22. Ten curves indicating bearing crack growth.

supervised learning with the data points from the time window #1 to #59. The prognosticator was designed to use five past data points to predict the next one and trained point by point (one-step prediction) for demonstration purposes. Starting from the time window #60, the prognosticator was either not retrained or retrained in a supervised fashion using the proposed RL + GA learning algorithm. The retraining was carried out either 100% or 67% by altering the reward

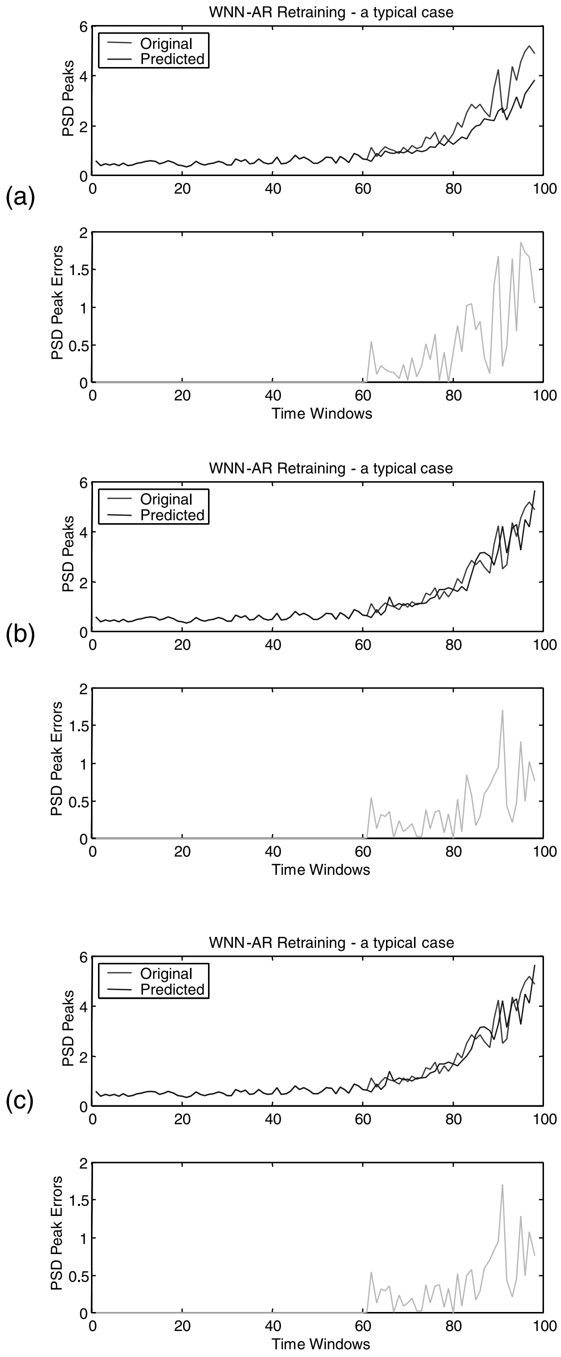

Fig. 23. Prognosis after (a) not retrained, (b) 100% retrained, and (c) 67% retrained for a typical bearing fault case.

https://doi.org/10.1017/S0890060401154089 Published online by Cambridge University Press

threshold in the learning algorithm. The prognostication results using these three retraining strategies (not retrained, 100% trained, and 67% trained, respectively) are shown in Figure 23 for a typical fault case, in Figure 24 for a statistical view of the prognosis, and in Table 1 for a statistical summary of the prognostic errors. From Table 1, it can be observed that after the retraining, almost all error metrics were reduced. It is also interesting to note that the 67% retraining generated results as good as those in the 100% case, which supports the proposed learning strategy. The rare spikes shown on a few curves in Figure 24 were due to the sensitivity of step-by-step retraining and could easily be eliminated with a limiting mechanism embedded in the learning procedure.

For performance comparison purposes, an AR and a WNN prognosticator were employed to predict the fault growth beyond the 87th time window for the curves shown in Figure 25 that were generated using the small perturbation

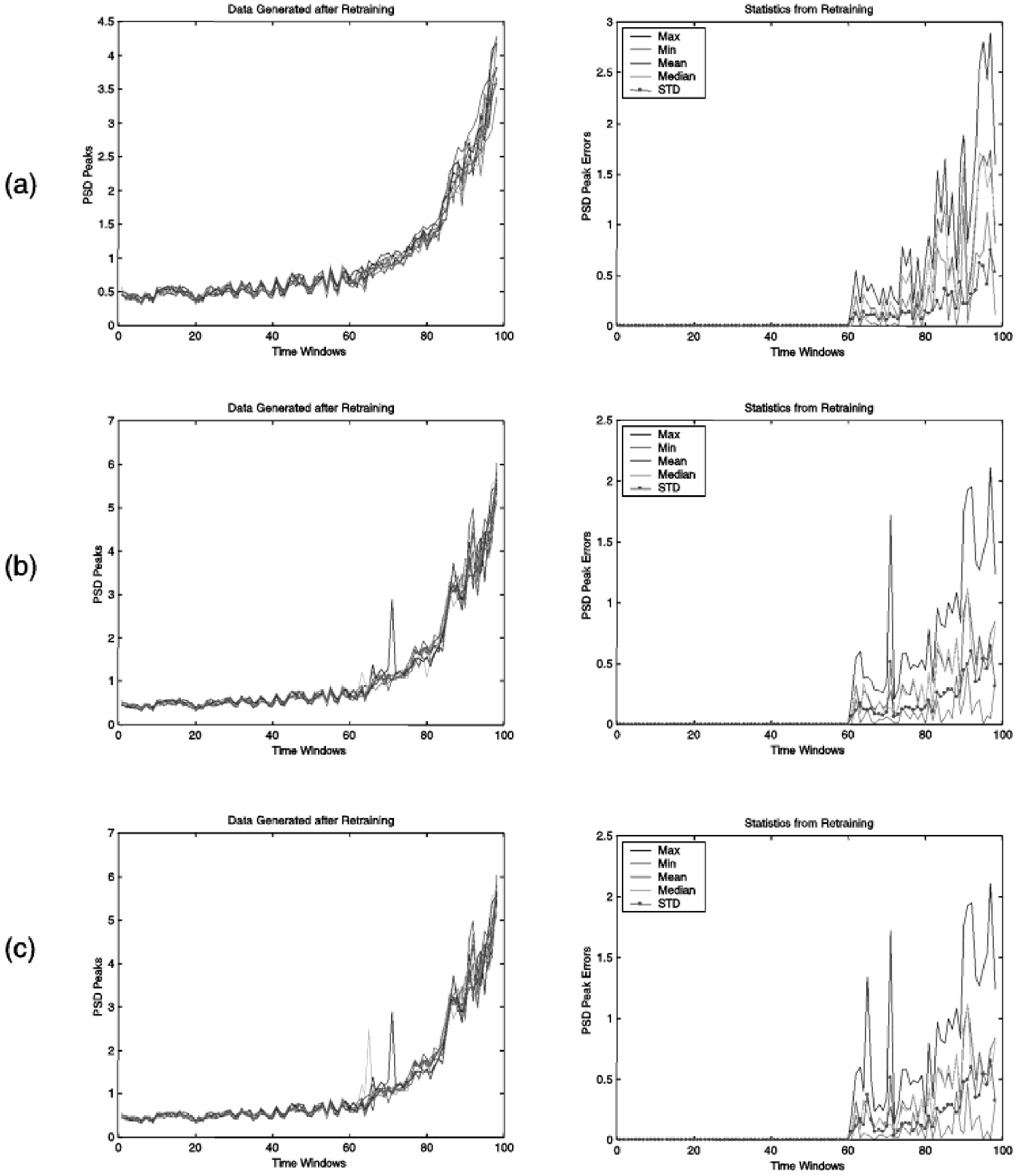

Fig. 24. Prognosis after (a) not retrained, (b) 100% retrained, and (c) 67% retrained for 10 bearing fault cases.

https://doi.org/10.1017/S0890060401154089 Published online by Cambridge University Press

Table 1. *Statistical summary of the prognosis for the three retraining cases*

Retraining	Max-sum	Min-sum	Mean-sum	Median-sum	STD-sum
0%	36.3343	10.6000	22.2808	21.8376	8.4170
100%	31.5005	3.8809	15.0437	13.8941	8.9190
67%	32.2762	3.6422	15.0822	13.7477	9.1208

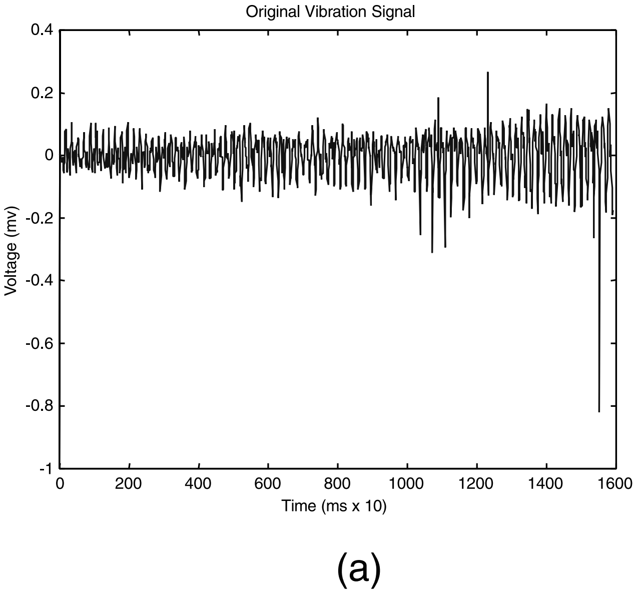

Fig. 25. A growing vibration signal (a) and its PSD peak curve (b).

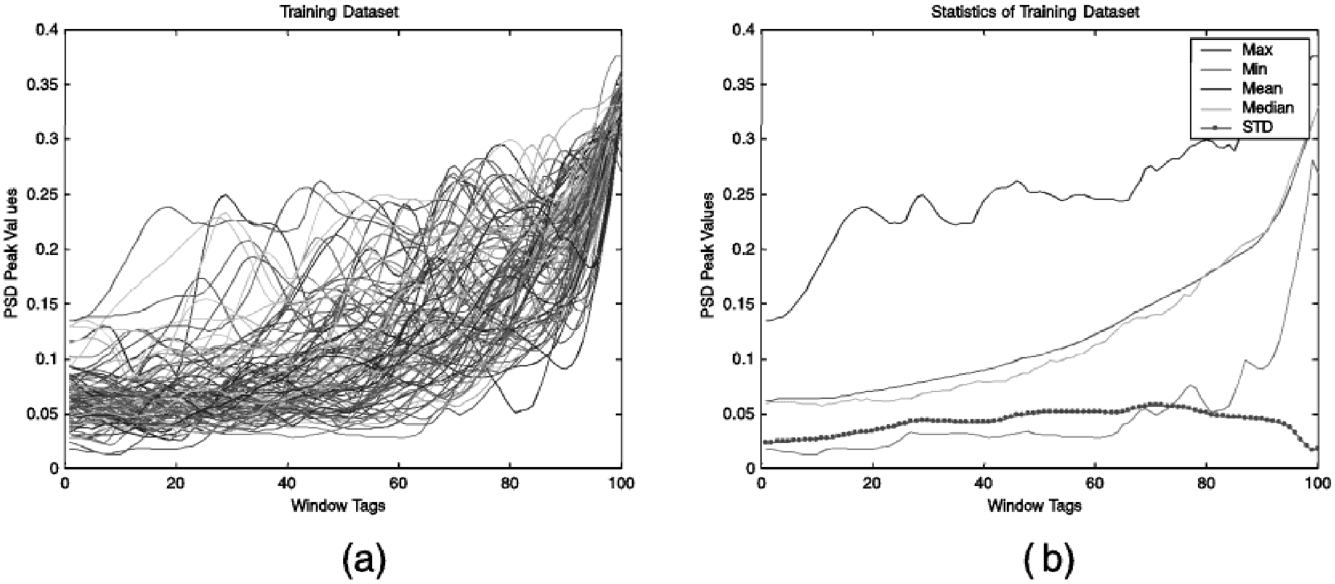

Fig. 26. The training data set (a) and its statistics (b).

https://doi.org/10.1017/S0890060401154089 Published online by Cambridge University Press

Table 2. *Performances of the AR and the WNN prognosticator*

Performance measures	TTF error rate	Dynamic error	Time × dynamic error	Similarity error	Output error	Total error
Scaling factor	1.0	100.00	1.0	0.1	1.0	N/A
Weighting coefficients	0.20	0.20	0.20	0.20	0.20	1.0
AR performance	0.4275	0.5200	0.4074	0.3448	0.3200	2.0197
WNN [erformance	0.1855	0.5500	0.2684	0.2857	0.3200	1.6096

technique. A hazard level or threshold was set at a value of 5 for the PSD and five historical points were used to predict the next data point. The prognostication results are shown in Figure 26. The five performance measures described previously were estimated. The assessment results were summarized in Table 2. It can be recognized that the total error for the AR prognosticator is 2.0197 compared to a total error of 1.6096 for the WNN prognosticator. Thus, in this case, the WNN algorithm performs better than its AR counterpart. It is also interesting to notice that the low-order AR predictor, like an average performer, had a tendency to average out sharp curves instead of approximating them precisely.

To further verify the effectiveness of the proposed prognostic methodology, a mixer system was adopted as another testing platform. The mixer consists of a motor mounted with a mechanical fixture on the wall of a water tank and a long metal rod with a fan at its end driven by the motor. If the mounting fixture became loose, the mixer system would vibrate drastically since its fan was submerged in the water. The looser the fixture was, the stronger the vibration would be. An experiment was designed in which the fixture was loosened gradually and the growing vibration was recorded. Such an experiment was performed about 400 times, resulting in a database of 400 growth data curves, each of which has 5000 data points. From this database, 100 curves were selected as a training data set and another 100 curves as a validation data set for a WNN prognosticator that used five historic values to predict a new value. For simplicity, these vibration signals were shortened to be of 1500 data points and then transformed into PSD peak curves of 100 data points. This was done by selecting the maximum value of the PSD of the windowed signal as the only feature, as shown in Figure 25. Statistics including maximum, minimum, mean, median, and standard deviation were calculated, which are shown in Figure 26 for training and Figure 27 for validation. It is important to see that the training and the validation data sets exhibit similar statistics. The prognostication was carried out using five-step prediction with the results shown in Figure 28 and Figure 29 for training and validation, respectively. It can be observed that the WNN prognosticator works as expected even with such diverse training and validation data sets in which one data curve looks quite different from another.

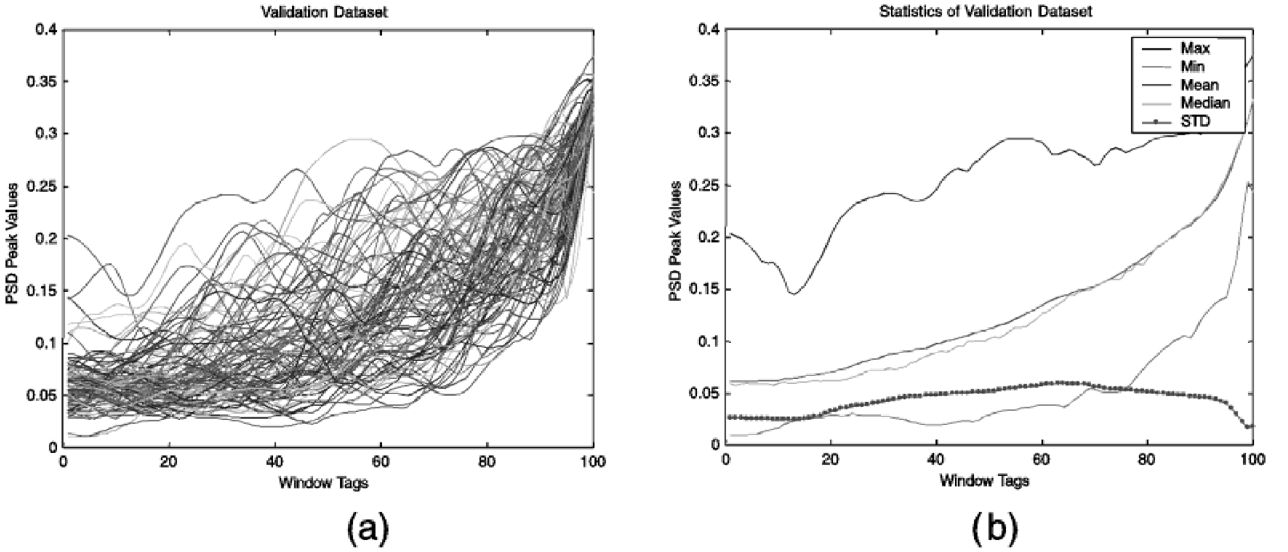

Fig. 27. The validation data set (a) and its statistics (b).

https://doi.org/10.1017/S0890060401154089 Published online by Cambridge University Press

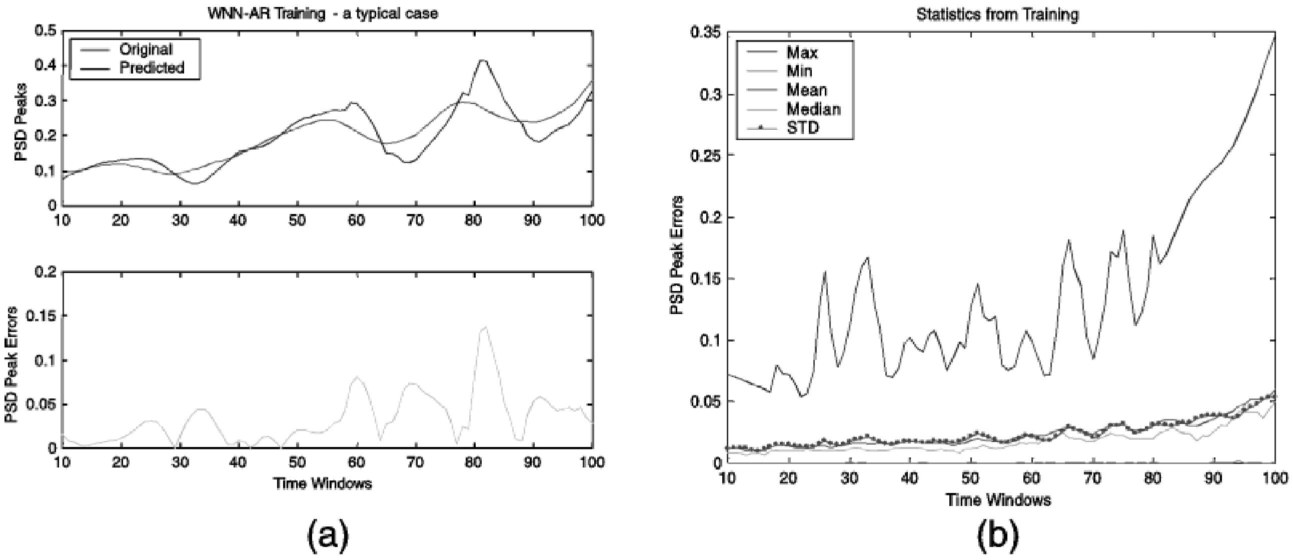

Fig. 28. Five-step prognostication of (a) 1 case and (b) 100 cases—training.

6. CONCLUSIONS

A fault prognostication architecture consisting of a virtual sensor and a dynamic wavelet neural network is proposed. The proposed model addresses two challenging issues relating to prognosis of machine or component failures: How do we "measure" the growth of a fault and how do we predict the remaining useful lifetime of such a failing component or machine? Reliable answers to these questions are bound to assist maintenance personnel in the conduct of condition-based maintenance so that uptime is maximized and the useful life of critical assets is prolonged. Simulation studies of the virtual sensor—predictor configuration, based on a limited experimental data set, show promise. More extensive failure data—difficult to obtain in critical processes—are required to draw firm and comparative conclusions. The proposed architecture provides a generic and open platform that can be easily modified and augmented as new failure evidence becomes available. The WNN construct (in both the static and dynamic versions) is amenable

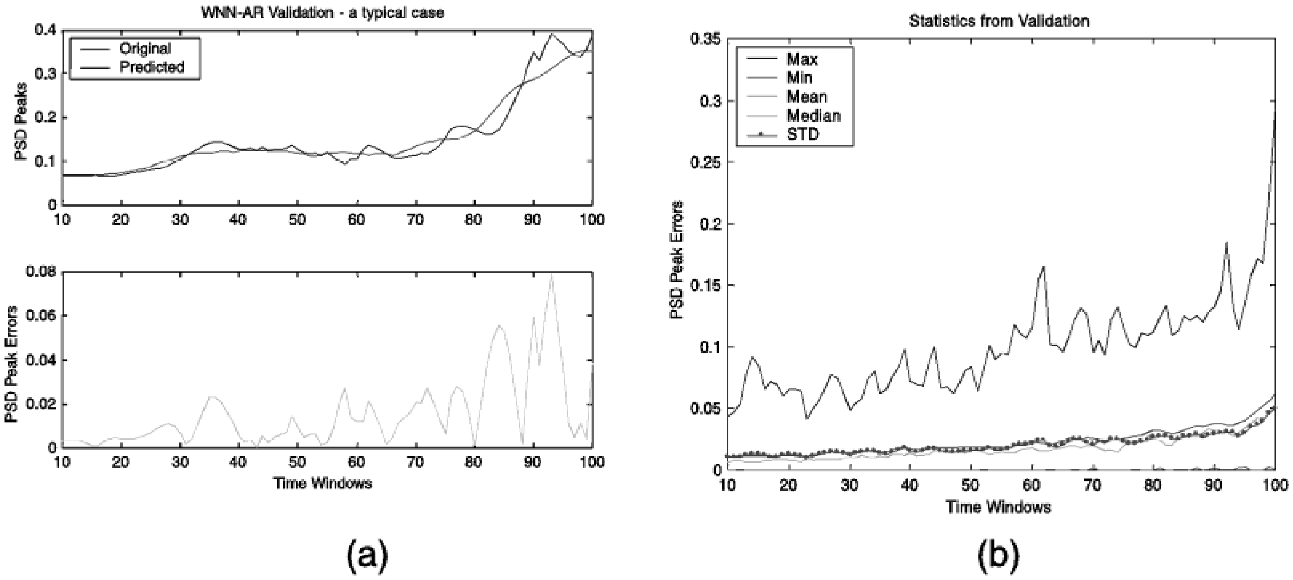

Fig. 29. Five-step prognostication of (a) 1 case and (b) 100 cases—validation.

https://doi.org/10.1017/S0890060401154089 Published online by Cambridge University Press

to accommodating learning routines (on-line and off-line) so that the algorithm can be improved with time. Uncertainty, a dominant influence in diagnostics and prognostics, must be accommodated and managed. A neuro-fuzzy version of the basic WNN and DWNN can assist in this direction when coupled with notions from Dempster–Shafer theory. This paper, therefore, serves as a motivation to encourage further research in those challenging areas of data collection and management, modeling, validation and verification, implementation, and assessment that are crucial to a successful penetration of these technologies in the industrial and manufacturing sectors of our economy.

ACKNOWLEDGMENTS

This research was partially supported by Honeywell Inc. and the Office of Naval Research under contract No. E-21-K30. Their support and assistance in the conduct of this work is greatly appreciated. The reviewers' comments are greatly appreciated as well.

REFERENCES

Barbera, F., Schneider, H. & Kelle, P. (1996). A condition based maintenance model with exponential failures and fixed inspection intervals. *Journal of the Operational Research Society 47(8)*, 1037–1045.

Essawy, M.A., & Zein-Sabatto, S. (1999). Measures of effectiveness and measures of performance for machine monitoring and diagnosis systems. *Proc. MARCON 99—Maintenance and Reliability Conference*, 42.101–42.109.

Frelicot, C. (1996). A fuzzy-based prognostic adaptive system. *Journal Europeen des Systemes Automatises 30(2–3)*, 281–299.

Groer, P.G. (2000). Analysis of time-to-failure with a Weibull model. *Proc. of MARCON 2000 Conference*, 59.01–59.04.

He, S.C., & He, Z.Y. (1997). Blind equalization of nonlinear communication channels using recurrent wavelet neural networks. *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, 3305–3308.

Jardim-Goncalves, R., Martins-Barata, M., Alvaro Assis-Lopes, A., & Steiger-Garcao. (1996). Application of stochastic modeling to support predictive maintenance for industrial environments. *Proc. IEEE International Conference on Systems, Man and Cybernetics, Information Intelligence and Systems*, 117–122.

Kaelbling, L.P., Littman, M.L., & Moore, A.W. (1996). Reinforcement Learning: A Survey. *Journal of Artificial Intelligence Research 4*, 237–285.

Konrad, H., & Isermann, R. (1996). Diagnosis of different faults in milling using drive signals and process models. *Proc. the 13th IFAC World Congress*, Vol. B, 91–96.

Lembessis, E., Antonopoulos, G., King, R.E., Halatsis, C., & Torres, J. (1989). CASSANDRA: an on-line expert system for fault prognosis. *Proc. the 5th CIM Europe Conference on Computer Integrated Manufacturing*, 371–377.

Lu, F., & Xu, X.G. (1993). A forecasting model of fuzzy self-regression. *Fuzzy Sets and Systems 58*, 239–242.

Makis, V., Jiang, X., & Jardine, A.K.S. (1998). A condition-based maintenance model. *IMA Journal of Mathematics Applied in Business and Industry 9(2)*, 201–210.

Marko, K.A., James, J.V., Feldkamp, T.M., Puskorius, C.V., Feldkamp, J.A., & Roller, D. (1996). Applications of neural networks to the construction of "virtual" sensors and model-based diagnostics. *Proc. ISATA 29th International Symposium on Automotive Technology and Automation*, 133–138.

Mylaraswamy, D., & Venkatasubramanian, V. (1997). A hybrid framework for large scale process fault diagnosis. *Computers & Chemical Engineering 21*(Suppl. Issue), S935–940.

Parker, B.E., Jr., Nigro, T.M., Carley, M.P., Barron, R.L., Ward, D.G., Poor, H.V., Rock, D., and DuBois, T.A. (1993). Helicopter gearbox diagnostics and prognostics using vibration signature analysis. *Proc. SPIE—The International Society for Optical Engineering*, Vol.1965, 531–542.

Prickett, P., & Eavery, S. (1991). The case for condition based maintenance. *Integrated Manufacturing Systems 2(3)*, 19–24.

Rao, S.S., & Kumthekar, B. (1994). Recurrent wavelet networks. *Proc. IEEE International Conference on Neural Networks*, 3143–3147.

Ray, A., & Tangirala, S. (1994). Stochastic modeling of fatigue crack dynamics for on-line failure prognostics. *IEEE Transactions on Control Systems Technology 4(4)*, 443–451.

Schauz, J.R. (1996). Wavelet neural networks for eeg modeling and classification. PhD Thesis, Atlanta, GA: Georgia Institute of Technology.

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

Sutton, R.S. (1988). Learning to predict by the method of temporal differences. *Machine Learning 3(1)*, 9–44.

Taylor, N.B. (1953). *Stedman's Medical Dictionary*, 18th ed., Baltimore, MD: The Williams & Wilkins Company.

Tsui, F.C., Sun, M.G., Li, C.C., & Sclabassi, R.J. (1995). A wavelet based neural network for prediction of ICP signal. *Proc. IEEE 17th Annual Conference on Engineering in Medicine and Biology and 21st Canadian Conference on Medical and Biological Engineering*, 1045–1046.

Vachtsevanos, G., Wang, P., & Khiripet, N. (1999). Prognostication: Algorithms and performance assessment methodologies. *Proc. ATP Fall National Meeting Condition-Based Maintenance Workshop*, San Jose, California, 1–12.

Peng Wang is a Ph.D. candidate at the School of Electrical and Computer Engineering of Georgia Institute of Technology, Atlanta, Georgia. He holds a B.S. degree in Electrical Engineering from Tsinghua University, Beijing, China and a Ph.D. degree from Hong Kong Polytechnic University, Kowloon, Hong Kong. He has authored/co-authored over 35 conference and journal publications. His research interests are in integrated system design, fault diagnostics, digital signal processing, and optical communication. He is a member of IEEE.

George J. Vachtsevanos received the B.E.E. degree in 1962 from the City College of New York and the M.E.E. degree from New York University in 1963. He received the Ph.D. degree in 1970 from the City University of New York. His doctoral dissertation was in the area of adaptive optimal control of dynamical systems using functional analysis techniques. Dr. Vachtsevanos had served at several universities in Europe and the U.S. before joining the faculty of the School of Electrical Engineering at Georgia Institute of Technology in 1984. He served as an advisor to the European Economic Commission and is a technical consultant for industry and government agencies. He has published several books and over 85 technical papers in scientific journals.

https://doi.org/10.1017/S0890060401154089 Published online by Cambridge University Press