Machine Condition Prediction Based on Adaptive Neuro-Fuzzy and High-Order Particle Filtering

Chaochao Chen, Bin Zhang, Senior Member, IEEE, George Vachtsevanos, Senior Member, IEEE, and Marcos Orchard

Abstract—Machine prognosis is a significant part of Condition-Based Maintenance (CBM) and intends to monitor and track the time evolution of the fault so that maintenance can be performed or the task be terminated to avoid a catastrophic failure. A new prognostic method is developed in this paper using adaptive neuro-fuzzy inference systems (ANFIS) and high-order particle filtering. The ANFIS is trained via machine historical failure data. The trained ANFIS and its modeling noise constitute an mth-order hidden Markov model to describe the fault propagation process. The high-order particle filter uses this Markov model to predict the time evolution of the fault indicator in the form of a probability density function (pdf). An on-line update scheme is developed to adapt the Markov model to various machine dynamics quickly. The performance of the proposed method is evaluated by using the testing data from a cracked carrier plate and a faulty bearing. The results show that it outperforms classical condition predictors.

Index Terms—Fuzzy systems, hidden Markov model, high-order particle filter, machinery condition monitoring, neural networks, prognosis.

I. INTRODUCTION

RECENTLY, Condition-Based Maintenance (CBM) is becoming the preferred practice for a variety of engineering systems to maintain their reliability, safety and availability. Instead of traditional scheduled or breakdown maintenance, CBM utilizes run-time data to determine/predict the machinery condition and hence its current/future fault condition, which can be used to schedule required repair and maintenance before malfunctions or even catastrophic failures occur. CBM enabling technologies mainly include sensing and monitoring, information processing, fault diagnosis and failure prognosis algorithms that are capable of detecting accurately and in a timely manner incipient failures and predicting the remaining useful life of failing components [1]. Amongst them prognosis is a key component that possesses the ability to predict accurately and precisely the future condition and remaining useful life of a failing component or subsystem. Prognosis is the Achilles’ heel of CBM and presents major challenges to the CBM system designers primarily because it projects the current condition of the fault indicator in the absence of future observations and necessarily entails large-grain uncertainty. In the last few decades, numerous efforts have been reported in the area of machinery prognosis [2]-[16], [29], [33].

Machine prognostic approaches can generally be categorized into two major classes: model-based (or physics-based) and data-driven methods [17],[18]. Through the understanding of the failure mode progression, model-based methods apply mathematical models to forecast the fault growth trend. Given a proper model for a specific system, model-based methods can offer accurate prediction estimates. However, it is usually difficult to develop accurate models in most practical instances, especially when the process of fault propagation is complex and/or is not fully understood. Data-driven methods, on the other hand, employ the collected condition data to derive the fault propagation models. Since most data-driven methods, such as recurrent neural networks (RNN), adaptive neuro-fuzzy inference systems (ANFIS) and adaptive recurrent neuro-fuzzy inference systems (ARNFIS), can be applied to a variety of systems, they have become a popular prediction tool in machinery prognosis.

In data-driven methods, the integration of neural networks and fuzzy systems, such as ANFIS, has been employed successfully in the prediction of machine condition degradation, where the prediction is carried out via a fuzzy system while its parameters are optimized through an artificial neural network [7]-[9],[13],[15],[16]. The superior forecasting performance of these predictors has been exhibited as compared to conventional neural-network-based predictors such as the radial-basis-function and recurrent-neural-network based models [7],[8]. Since the machine dynamics in real applications change with time, the trained ANFIS may not be able to carry...
out accurate predictions if various dynamics/states are not taken into account during the prediction process. Considering that sequential Monte Carlo methods, such as particle filtering, can update system states in real-time via new data, the ANFIS is integrated with a particle filter so that on-line data can be used to improve the prediction accuracy.

As a powerful methodology for sequential signal processing, particle filtering approximates the state PDF by using point masses (or particles) with associated discrete probability masses (or weights) based on the concept of sequential importance sampling and the use of Bayesian theory [19]-[21]. Recently, particle filtering has been employed in machinery prognosis since the fault degradation is a complex nonlinear problem while particle filtering is particularly useful in dealing with those difficulties [22]-[24], [30]. In most applications, mathematical models have been established to describe the fault propagation process. However, the derivations of these models are complex and require expert knowledge about the degradation process, i.e. a detailed Finite Element Analysis model, for example, to estimate the values of the parameters of the degradation process, i.e. a detailed Finite Element Analysis model, for example, to estimate the values of the parameters of the fault growth model. Moreover, note that the fault growth model, for example, to estimate the values of the parameters of the degradation process, i.e. a detailed Finite Element Analysis model, for example, to estimate the values of the parameters of the fault growth model. Moreover, note that the fault growth model, for example, to estimate the values of the parameters of the degradation process, i.e. a detailed Finite Element Analysis model, for example, to estimate the values of the parameters of the fault growth process.

A high-order model may be more appropriate to describe the fault growth trend, that is, the current system state depends not only on the previous state. In many applications, however, the first-order model is not generally true and a high-order model may be more appropriate to describe the fault growth trend, that is, the current system state depends not only on the previous state but also on multiple p-step-before states, i.e. \( p = 2, 3, 4 \). To overcome these limitations, in this work, machine condition prognosis is carried out via a high-order particle filter, where a combination of the ANFIS and the process noise, as a high-order HMM, is employed to represent the fault growth process.

Note that errors between the actual machine condition and the prediction estimates from the ANFIS model do exist even with a well-trained ANFIS model. Moreover, system dynamics may change in the future. Therefore, an on-line model adaptation scheme for fault propagation is desirable. In this paper, a sliding window with a given length screening the residual signal is utilized. The residual signal screened by the window generates an error PDF that is employed to update the model parameters in real-time.

The integration of the ANFIS and high-order particle filtering in this work forms a new approach for machine health condition prognosis that possesses the merits including non-linear mapping and real-time state estimation. The on-line model update scheme is able to adapt the fault growth model to various machine dynamics quickly. Experimental data from a damaged helicopter transmission component and a faulty bearing are employed to validate the proposed approach. The results demonstrate that it outperforms classical predictors.

The remainder of this paper is organized as follows: in the next section, the RNN, ANFIS and ARNFIS are introduced to perform machine condition prediction. Section III presents the proposed prediction approach. Bayesian estimation is introduced first, a high-order HMM and its posterior PDF are presented, and then the integration of the ANFIS in a high-order particle filter is demonstrated. Next, an on-line model update scheme is given. Finally, the prediction algorithm is illustrated in detail. Section IV presents the experimental results of the proposed approach on two real systems and the performance comparison with classical predictors is given. Section V provides some concluding remarks.

II. THE RNN, ANFIS AND ARNFIS PREDICTORS

The RNN, ANFIS and ARNFIS have been applied successfully in the field of time-series prediction. Recently, these techniques have also been extended to the applications of machinery condition prognosis, since they are able to learn highly nonlinear dynamics of machines without the necessity of deriving complex mathematical models. For these predictors, the input variables, \( \{x_{t-3}, x_{t-2}, x_{t-1}, x_{t}\} \), and the output/forecasting variable, \( x_{t+n} \), are “monitoring indices” that characterize the machine health condition, where \( r \) denotes the prediction step, i.e. when \( r=1 \), \( x_{t} \) means a one-step-ahead prediction value, and \( n \) defines the number of previous time steps, i.e. when \( n=3 \), the values of three previous time steps and the current value are used to carry out the prediction. For example, when \( r=1 \) and \( n=3 \), the input variables are \( \{x_{t-3}, x_{t-2}, x_{t-1}, x_{t}\} \) and the output is \( x_{t+1} \). Historical machine failure data are utilized to train these predictors. The RNN uses the gradient descent approach to tune its parameters while the ANFIS and ARNFIS employ a hybrid learning algorithm that combines the gradient descent method and the least squares method. The training process is terminated when the number of training iterations has reached a predefined value or the desired training error has been achieved.

A. The RNN Predictor

![Fig. 1. Architecture of the RNN predictor; \( Z^{-1} \) is a unit delay operator; \( S \) is a sigmoid function.](image)

The RNN predictor is a commonly used prognostic model in machinery prognosis. Its architecture is similar to that of a feedforward neural network but with additional feedback links. Many studies demonstrate that the closed loop structure can
assist the RNN to capture the temporal behavior of dynamic systems easily [6], [10], [11]. Fig. 1 shows the architecture of the RNN predictor. It possesses three layers, namely input, hidden and output layers, respectively. The corresponding number of nodes in each layer is 4, 20 and 1, respectively. Note that there is no simple way to determine in advance the optimal node number in the hidden layer, here, 20 nodes are adopted. The increase of the nodes in the hidden layer may improve the prediction accuracy, but the computational complexity is increased as well.

The input signals transmit through the nodes in the input layer, and then combine the feedback signals from the output layer to the hidden layer, where the combined signals are processed by the nodes using sigmoid activation functions. Next, the node in the output layer is activated via a sigmoid function while receiving the signals from the hidden layer, and then the output signal is obtained.

**B. The ANFIS Predictor**

![ANFIS diagram](image)

Fig. 2. Architecture of the ANFIS predictor; S is a sigmoid function; O is an operator defined in Equation (5).

The ANFIS predictor is a fuzzy Sugeno model, whose parameters are optimized via neural network training and structure is determined by expert knowledge [25]. Like many ANFIS applications in machinery prognosis, four input variables \{x\_i, x\_i+1, x\_j, x\_\_i\} are chosen and each variable is assigned with two Membership Functions (MFs), namely small and large. Therefore, sixteen fuzzy IF-THEN rules are generated to perform the prediction, which are the same rules \((2^4=16)\) as that in [7],[8],[13],[15], as shown below:

Rule j:
IF \(x\_i \) is \(A_{i1} \) AND \(x\_i+1 \) is \(A_{i2} \) AND \(x\_j \) is \(A_{j1} \) AND \(x\_\_i \) is \(A_{k1} \),
THEN \(y^j = c_{i1}x\_i + c_{i2}x\_i+1 + c_{j1}x\_j + c_{k1}x\_\_i \); \(j = 1,2,\ldots,16\).

where \(y^j\) is the prediction result according to the \(j\)th fuzzy rule, \(A_{i1}\) is the fuzzy set associated with the \(i\)th input variable in the \(j\)th fuzzy rule, and \(c_{ik}\) is the parameter that is determined by the learning process, here, \(i = 1,2,\ldots,4\) and \(k = 1,2,\ldots,5\).

The ANFIS predictor consists of five layers. Its architecture is schematically shown in Fig. 2. The signal propagation is illustrated as follows:

- In the following description, \(x^{(k)}\) defines the \(k\)th node input in the \(k\)th layer, and \(y^{(k)}\) denotes the \(i\)th node output in the \(k\)th layer.

**Layer 1:** The input signals transmit directly to the next layer without any computation. The outputs of this layer can be expressed by

\[ y^{(1)}_i = x^{(1)}_i, \quad i = 1,2,\ldots,4. \]  \(1\)

**Layer 2:** Each node in this layer performs the calculation of a MF, small or large. Sigmoid MFs are used here, as shown below:

\[ u^{(2)}_{ij}(x^{(1)}_i) = \frac{1}{1 + \exp(-b^{(2)}_{ij}(x^{(1)}_i) - m^{(2)}_{ij})} \quad i = 1,2,\ldots,4, \quad j = 1,2,\ldots,16. \]  \(2\)

where \(u^{(2)}_{ij}\) is the output signal with respective to the \(i\)th input variable \(x^{(1)}_i\) in the \(j\)th fuzzy rule, \(b^{(2)}_{ij}\) and \(m^{(2)}_{ij}\) are the parameters of the sigmoid function and referred to as premise parameters.

**Layer 3:** An AND operator is chosen as a fuzzy T-norm operation in this layer, which is described as

\[ y^{(3)}_j = \prod_{i} u^{(2)}_{ij}(x^{(1)}_i), \quad i = 1,2,\ldots,4, \quad j = 1,2,\ldots,16. \]  \(3\)

where the output \(y^{(3)}_j\) represents the firing strength of the \(j\)th fuzzy rule.

**Layer 4:** This layer performs the normalization operation for all the rule firing strengths. The resulting output is given by

\[ y^{(4)}_j = \frac{\sum_{j} y^{(3)}_j}{\sum_{j} y^{(3)}_j}, \quad j = 1,2,\ldots,16. \]  \(4\)

**Layer 5:** After a linear combination of the input signals, the output of the ANFIS is calculated by:

\[ x_{t+r} = \sum_{j} y^{(4)}_j (c^{(4)}_1 x_{t-3} + c^{(4)}_2 x_{t-2} + c^{(4)}_j x_{t} + c^{(4)}_k x_{t+r}), \quad j = 1,2,\ldots,16. \]  \(5\)

where \{\(c^{(4)}_1, c^{(4)}_2, c^{(4)}_j, c^{(4)}_k\}\} are a set of unknown parameters called consequent parameters.

In order to improve the training efficiency and avoid local minima, a hybrid learning algorithm that combines the gradient descent method and the least squares method is used to tune optimally the parameters of the ANFIS. The consequent parameters \{\(c^{(4)}_1, c^{(4)}_2, c^{(4)}_j, c^{(4)}_k\}\} are optimized by using the least square method, whereas the premise parameters, \(b^{(2)}_{ij}\) and \(m^{(2)}_{ij}\), are updated via the gradient descent method.

**C. The ARNFIS Predictor**

Fig. 3 indicates the structure of the ARNFIS predictor, which
variable added in the eight nodes of Layer 2 has five layers and the number of nodes in each layer is 4, 8, 16, 16, and 1, respectively. It has the same layer number and node number as that in ANFIS, except additional feedback links are added in the eight nodes of Layer 2. Each node in Layer 2 functions as a memory unit that performs the following operation:

\[
u^{(2)}_{ij}(x^{(2)}_t) = \frac{1}{1 + \exp[-b^{(2)}_j(x^{(2)}_t - m^{(2)}_j)]}, \quad i = 1,2,\ldots,4, \quad j = 1,2,\ldots,16. \tag{6}\]

where \(u^{(2)}_{ij}\) is the output signal associated with the \(i\)th input variable \(x^{(2)}_t\) in the \(j\)th fuzzy rule, \(b^{(2)}_j\) and \(m^{(2)}_j\) are premise parameters.

Note that the inputs of the nodes in this layer contain the feedback components. They are:

\[x^{(2)}_t(t) = x^{(1)}_t(t) + \theta^{(2)}_{ij}u^{(2)}_{ij}(t-1) \tag{7}\]

where \(\theta^{(2)}_{ij}\) is the feedback link weight and is initially set to zero and then optimized via learning process using the training data set. It is clear that the degree of membership \(u^{(2)}_{ij}(t-1)\) at the previous time step is used as one part of the current input value, which allows the ARNFIS predictor to memorize the past information so that it can deal with temporal issues.

Like the ANFIS predictor, the hybrid training algorithm is used to tune the parameters.

### III. PROPOSED PREDICTION ALGORITHM

The proposed prediction approach is discussed in this section. The Bayesian estimation technique using \(m\)th-order HMM is presented first. Then, the ANFIS predictor described above with the process noise, as the fault growth model, is integrated with a high-order particle filter. Next, an on-line adaptation scheme is given to adapt this model to various machine dynamics quickly. Lastly, a thorough description of the algorithm steps is presented.

#### A. Bayesian Estimation Using \(m\)th-Order Markov Model

Through the use of noisy observation data, a Bayesian estimation technique is intended to estimate a state vector in a mathematical process model. Since the streaming measurement data for prognosis is available at discrete times via digital devices, the present study is focused only on discrete-time systems.

The evolution of the machine condition can be given by a HMM. In general, a first-order Markov model is used to describe the fault growth process. Here, instead of using a first-order model, an \(m\)th-order Markov model is employed since the condition evolution may depend not only on the previous state but also on several \(p\)-step-before states. The following presents the \(m\)th-order model

\[x_k = f_k(x_{k-1}, x_{k-2}, \ldots, x_{k-m}, \omega_{k-1}) \tag{8}\]

where \(x_k\) is the model state at time \(k\), \(x_{k-m}\) is the state at time \(k-m\), \(\omega_{k-1}\) is an i.i.d. process noise at time \(k-1\), and \(f_k\) is a possibly nonlinear function.

The measurement model is expressed by

\[y_k = h_k(x_k, v_k) \tag{9}\]

where \(y_k\) is the measurement, \(v_k\) is an i.i.d. measurement noise, and \(h_k\) is a possibly nonlinear function that denotes the non-linear mapping relationship between the model states and the noisy measurements. Here, Equation (9) can be simply described as \(y_k = x_k + v_k\), since both the model state \(x_k\) and output \(y_k\) represent the machine condition indicator (or monitoring index).

The state estimation is achieved recursively in two steps: prediction and update. The prediction step aims to obtain the prior PDF of the state for the next time instant \(k\) by using the following equation:

\[p(x_{0:k} | y_{1:k-1}) = \int p(x_{0:k} | x_{k-1})p(x_{k-1} | y_{1:k-1})dx_{k-1} \tag{10}\]

where the probabilistic process model \(p(x_{0:k} | x_{k-1})\) is defined via Equation (8), and \(p(x_{0:k} | y_{1:k-1})\) represents the state PDF at time \(k-1\). Note that in Equation (10), the fact that \(p(x_{0:k} | x_{k-1}, y_{1:k-1}) = p(x_{0:k} | x_{k-1})\) is used according to the Markov properties on the moral graph of the \(m\)th-order HMM [26].

When a new measurement becomes available, the update step is carried out. By considering the new measurement, the prior state PDF, the likelihood function \(p(y_k | x_k)\), and Bayes’ rule, the posterior state PDF can be calculated by
The recursive computation of the posterior state PDF

\[ p(x_{0:k}|y_{1:k}) = \frac{p(y_k|x_k)p(x_{0:k}|y_{1:k-1})}{p(y_k|y_{1:k-1})} \]

\[ = \frac{p(y_k|x_k)p(x_{k-1:k}|x_{0:k-1}, y_{1:k-1})}{p(y_k|y_{1:k-1})} \]

\[ \propto p(y_k|x_k)p(x_{k-1:k}|x_{0:k-1}, y_{1:k-1}) \]  

(11)

where the normalizing constant

\[ p(y_k|y_{1:k-1}) = \int p(y_k|x_k)p(x_{k-1:k}|x_{0:k-1}, y_{1:k-1}) dx_k \]

and the likelihood function \( p(y_k|x_k) \) is defined by the measurement model (9).

**B. Integration of ANFIS in High-Order Particle Filtering**

The recursive computation of the posterior state PDF \( p(x_{0:k}|y_{1:k}) \) is more conceptual than practical, since the integrals in Equations (10) and (11) do not have an analytical solution in most cases. Therefore, many estimation methods have been developed to solve this problem [20]. In this paper, a high-order particle filter is employed to approximate the optimal Bayesian solution.

In general, particle filtering is a Monte Carlo method that employs a Sequential Importance Sampling algorithm. The posterior PDF can be approximated by a set of random samples (or particles) with associated weights, as shown below [20]

\[ p(x_{0:k}|y_{1:k}) = \sum_{i=1}^{N} w_i^k \delta(x_{0:k} - x_{0:k}^i) \]  

(12)

where \( N \) is the total number of particles, \( x_{0:k} = \{x_j, j = 0,1,\ldots,k\} \) is the set of all states up to time \( k \), \( \{x_{0:k}, i = 1,\ldots,N\} \) is a set of particles with associated weights \( \{w_i, i = 1,\ldots,N\} \), and \( \delta(\bullet) \) is the Dirac delta measure.

Based on the importance sampling principle, if the particles \( x_{0:k}^i \) are drawn from an importance density \( q(x_{0:k}|y_{1:k}) \), the normalized weights are updated as [20]

\[ w_i^k \propto p(x_{0:k}|y_{1:k}) \]

(13)

Moreover, if the importance density is chosen to factorize such that

\[ q(x_{0:k}|y_{1:k}) = q(x_k|x_{0:k-1}, y_{1:k}) q(x_{0:k-1}|y_{1:k-1}) \]

\[ = q(x_k|x_{k-1:k-1}, y_k) q(x_{0:k-1}|y_{1:k-1}) \]  

(14)

By substituting Equations (11) and (14) into (13), we obtain

\[ w_i^k \propto \frac{p(y_k|x_k)p(x_{k-1:k-1}|x_{0:k-1}, y_{1:k-1})}{q(x_k|x_{k-1:k-1}, y_k)} \]

(15)

If we simply choose

\[ q(x_k'|x_{k-1:k-1}, y_k) = p(x_k'|x_{k-1:k-1}) \]  

(16)

and substitute Equation (16) into (15), then yields

\[ w_i^k \propto w_i^{k-1} p(y_k|x_k') \]

(17)

In order to integrate the ANFIS predictor in the particle filtering framework, we set \( m = 4 \), that is, a 4th-order particle filter is used since the ANFIS, defined by Equations (1)-(5), has four previous state values as the inputs. Therefore, the 4th-order HMM that presents the fault growth process is described as follows:

\[ x_k = \hat{x}_k + \omega_k \]

(18)

\[ \hat{x}_k = g_k(x_{k-1}, x_{k-2}, x_{k-3}, x_{k-4}) \]  

(19)

where \( g_k(x_{k-1}, x_{k-2}, x_{k-3}, x_{k-4}) \) is a nonlinear function denoted by the ANFIS. The current and three previous states of HMM, i.e., \( x_{k-1}, x_{k-2}, x_{k-3}, x_{k-4} \) in Eq. (19), are the four inputs of the ANFIS. Accordingly, the output of the ANFIS \( \hat{x}_k \) plus its process noise \( \omega_k \) is the state of HMM at the next time, as shown in Eq. (18). Therefore, the HMM consists of two components: one is the ANFIS that is trained off-line; the other is the process noise.

Each state in the HMM is set with 100 particles and their corresponding weights are initially the same, i.e., 0.01. The state evolution of the HMM (Eq. (18)) requires the parameters of the ANFIS and the process noise. The premise membership function parameters of the ANFIS \( \{m_i^{(0)} \text{and} b_i^{(0)} \} \) in Eq. (2)) are initially generated with small random number [31]. We also can initially assign \( m_i^{(2)} \) with the value of the mean of the training data set and choose \( b_i^{(2)} \) empirically [9], [32]. The consequent parameters of the ANFIS \( \{c_i', c_i', c_i', c_i', c_i' \} \) in Eq.(5)) are initially assigned with 0 and [8]. After the off-line training, the optimal parameters of the ANFIS are obtained. Note that when different initialization conditions are adopted, the experimental results show that the proposed algorithm still outperforms the three conventional predictors. The process noise is assumed to follow Gaussian distribution and its mean and variance can be initialized directly via the ANFIS’s modeling errors. The likelihood functions \( p(y_k|x_k') \) in Eq. (17) are Gaussian, which are used to update the weights of particles. Here, we empirically set the bias and variance of the likelihood function with 0 and 0.0025, respectively, since we found that the small values of these parameters lead to better prediction accuracy.
C. On-line Model Update

In Equation (18), we note that errors always occur when the ANFIS attempts to simulate the fault growth process, especially when the machine dynamics change during the prediction process due to many factors, e.g., change in operational conditions. The process noise $\omega$ in Equation (18) is a stochastic variable that can be assumed to follow a Gaussian distribution, $\omega \sim N(\mu_\omega, \sigma_\omega^2)$.

Therefore, the mean $\mu_\omega$ and the standard deviation $\sigma_\omega$ of the noise at time $k$ can be calculated by

$$\mu_\omega = \frac{\sum_{i=0}^{n-1} z_{k-i}}{n} \quad (20)$$

$$\sigma_\omega = \sqrt{\frac{\sum_{i=0}^{n-1} (z_{k-i} - \mu_\omega)^2}{n}} \quad (21)$$

where $z_{k-i}$ is the residual between the actual condition data and the prediction estimate via the ANFIS at time $k-i$, and $n$ is the number of the ANFIS’s residuals.

Furthermore, a sliding window that contains $n$ ANFIS’s residuals is utilized in order to estimate the process noise in real-time, as shown in Fig. 4. When a new measurement becomes available, the window moves forward one time step so that it can include the latest model error information.

D. Algorithm Steps

The detailed algorithm steps for condition prognosis are stated as:

1. **Step 1:** The ANFIS is trained with available condition data to model the fault propagation process.

2. **Step 2:** The fault growth model (18), represented by the ANFIS and the process noise, is employed with a 4th-order particle filter to draw a set of particles. According to the values of the particles and current weights, one-step-ahead condition prediction can be carried out via:

$$\overline{x}_k = \sum_{i=1}^{N} w_i^k x_i^k \quad (22)$$

Multi-step-ahead condition prediction also can be computed by successively taking the expectation of the model update Equation (18) for every future time instant, considering the calculated condition value associated to each particle as initial condition value for the next step prediction, as shown in:

$$\overline{x}_{k+r} = E\left[\overline{x}_{k+r}\right]$$

$$x_{k+r}^i = g\left(x_{k+r-1}^i + x_{k+r-2}^i + x_{k+r-3}^i + x_{k+r-4}^i\right) + \omega_{k+r-1} \quad (23)$$

In Eq. (23), $E\left[\right]$ denotes the expectation of $x_{k+r}^i$, i.e., $\overline{x}_{k+r} = \sum_{i=1}^{N} x_{k+r}^i w^i_{k+r-1}$, where $\overline{x}_{k+r}$ is the multi-step-ahead prediction value, $x_{k+r}^i$ is the state value of the ith particle at the (k+r)th time instant, $w^i_{k+r-1}$ is the weight of the ith particle at the (k+r-1)th time instant and $N$ is the total number of particles. Here, $g\left(\right)$ represents the trained ANFIS, and its four inputs can be recursively calculated like below:

$$x_{k+1}^i = g\left(x_{k}^i + x_{k-1}^i + x_{k-2}^i + x_{k-3}^i\right) + \omega_{k-1}$$

$$x_{k+2}^i = g\left(x_{k+1}^i + x_{k}^i + x_{k-1}^i + x_{k-2}^i\right) + \omega_{k}$$

$$\vdots$$
\[ x'_{k+r-1} = g(x'_{k+r-2} + x'_{k+r-3} + x'_{k+r-4} + x'_{k+r-5}) + \omega_{k+r-2} \]

Note that the states \( x'_{k-1}, x'_{k-2}, x'_{k-3}, x'_{k-4} \), the process noise \( \omega_{k-1} \), and the weights \( w'_{k-1} \) have been known.

Also, we set \( w'_{k+r-1} = w'_{k-1} = \omega_{k} = \omega_{k-1} \). Similar multi-step-ahead condition prediction method can be found in [23].

When a new measurement becomes available, the weights can be calculated according to Equation (17). If severe degeneracy does exist, resampling is performed.

Step 3: Update the process noise using Equations (20), (21)

Step 4: Repeat Step 2 and Step 3 until machine prognosis is complete.

Here, step 2 can be considered as the execution of a Sequential Importance Sampling and Resampling algorithm that is summarized in Fig. 5. The flowchart of the proposed algorithm is shown in Fig. 6.

\[
\text{RMSE} = \sqrt{\frac{1}{M} \sum_{i=1}^{M} (y_i - \hat{y}_i)^2}
\]

where \( M \) is the total number of data points, \( y_i \) and \( \hat{y}_i \) are the \( i \)-th actual and predicted values, respectively.

The smaller value of the RMSE means higher prediction accuracy.

A. Cracked Carrier Plate

1) System Condition Monitoring

The main transmission of Blackhawk and Seahawk helicopters employs a five-planet epicyclic gear system. Recently, a crack in the planetary carrier plate was discovered during regular maintenance, as shown in Fig. 7. It apparently endangers the pilot’s life with a possible loss of the aircraft, and thus a condition prognosis scheme is needed to carry out accurate prediction of the asset’s remaining useful life in a real-time manner so that timely maintenance can be implemented before catastrophic events occur.

In order to derive an appropriate condition monitoring index...
(or feature), the gearbox is mounted on a test cell with a seeded crack fault on the planetary gear carrier. An accelerometer is mounted at a fixed point at position $\theta = 0$ on the gearbox to collect the vibration signals, as shown schematically in Fig. 8. Surrounding the sun gear, the planet gears ride on the planetary carrier and also rotate inside the outer ring gear (or annulus gear). Due to the complex operational environment and the large number of noise sources in the system, a blind deconvolution de-noising algorithm has been developed to improve the signal-to-noise ratio [27]. The sideband ratio is set as the condition monitoring index, which is calculated by the ratio between the energy of the NonRMC and all sidebands [28]:

$$SBR(X) = \frac{\sum_{k=-X}^{X} X_{NonRMC}}{\sum_{k=-X}^{X} (X_{NonRMC} + RMC)}$$  \hspace{1cm} (25)$$

where $RMC$ is the Regular Meshing Components or apparent sidebands, and $NonRMC$ represents the Non Regular Meshing Components.

2) Performance Evaluation

The initial length of the seeded crack on the carrier plate is 1.344 inches and it grows with the evolving operation of the gearbox. The gearbox operates for a period of 1000 Ground-Air-Ground (GAG) cycles, and each cycle lasts about 3 minutes at three different torque levels: 20%, 40% and around 100%. Every two GAG circles, the vibration feature (or monitoring index) at 20% torque level is extracted and used for training the RNN, ANFIS and ARNFIS. The features at 40% and 100% torque levels are used for testing, respectively. Therefore, there are 500 data pairs used for training and the time step is two GAG circles (or about 6 minutes).

Fig. 9 shows the comparison results of the two-step-ahead prediction for the monitoring index (or condition indicator) of the damaged gearbox operating at 40% torque level. It is seen that the proposed approach (Fig. 9(d)) captures the system’s dynamics quickly and accurately and the prediction results are quite close to the actual values. High prediction accuracy is also exhibited for the ANFIS and ARNFIS, as shown in Fig. 9(b) and 9(c), respectively. Apparently, the prediction accuracy of the RNN (Fig. 9(a)) is much lower than that of the proposed approach, especially at the beginning of the testing phase.

Fig. 10 indicates the two-step-ahead prediction comparison results at 100% torque level. The RNN fails to capture the system’s new dynamics after the time step of about 350 so that low prediction accuracy is presented, as shown in Fig. 10(a). For the ANFIS, ARNFIS and proposed approach, superior prediction accuracy is exhibited as compared to the RNN. In general, the ANFIS and ARNFIS can track the fault propagation trend. But the prediction accuracy for both is lower than that of the proposed approach, particularly at the end of the testing phase.

Fig. 11 and Fig. 12 demonstrate the four-step-ahead prediction comparison results at 40% and 100% torque levels, respectively. It is observed that the proposed predictor can effectively capture and track the system’s new dynamics and thus outperform the RNN, ANFIS and ARNFIS predictors. Table I gives the prediction performance comparison over several steps in terms of the RMSE metric. Here, $+r$ means $r$-step-ahead prediction. It is clear that the prediction accuracy of the proposed approach is superior to that of the RNN, ANFIS and ARNFIS.

### B. Faulty Helicopter Oil cooler Bearing

A helicopter oil cooler bearing with unknown fault mode is used to evaluate the proposed prediction approach with different time scale and monitoring index. The data is provided with information that the bearing is faulty but without detailed fault information. The predictor is trained by the monitoring index (sum of weighted frequency components related to harmonics of the frequency of interest) acquired from a faulty

---

**Table I**

<table>
<thead>
<tr>
<th>RNN</th>
<th>ANFIS</th>
<th>ARNFIS</th>
<th>Proposed Approach</th>
</tr>
</thead>
<tbody>
<tr>
<td>40% torque level</td>
<td>100% torque level</td>
<td>40% torque level</td>
<td>100% torque level</td>
</tr>
<tr>
<td>1</td>
<td>0.1463</td>
<td>0.4517</td>
<td>0.0617</td>
</tr>
<tr>
<td>2</td>
<td>0.1457</td>
<td>0.4537</td>
<td>0.0654</td>
</tr>
<tr>
<td>3</td>
<td>0.1485</td>
<td>0.4603</td>
<td>0.0700</td>
</tr>
<tr>
<td>4</td>
<td>0.1471</td>
<td>0.4668</td>
<td>0.0738</td>
</tr>
<tr>
<td>5</td>
<td>0.1462</td>
<td>0.4700</td>
<td>0.0751</td>
</tr>
</tbody>
</table>

**Table II**

<table>
<thead>
<tr>
<th>RNN</th>
<th>ANFIS</th>
<th>ARNFIS</th>
<th>Proposed Approach</th>
</tr>
</thead>
<tbody>
<tr>
<td>40% torque level</td>
<td>100% torque level</td>
<td>40% torque level</td>
<td>100% torque level</td>
</tr>
<tr>
<td>1</td>
<td>0.1933</td>
<td>0.0810</td>
<td>0.0883</td>
</tr>
<tr>
<td>2</td>
<td>0.2000</td>
<td>0.0828</td>
<td>0.0792</td>
</tr>
<tr>
<td>3</td>
<td>0.1993</td>
<td>0.1557</td>
<td>0.0843</td>
</tr>
<tr>
<td>4</td>
<td>0.2004</td>
<td>0.1473</td>
<td>0.1234</td>
</tr>
<tr>
<td>5</td>
<td>0.2062</td>
<td>0.1109</td>
<td>0.1406</td>
</tr>
</tbody>
</table>
Fig. 9. Two-step-ahead prediction results for the monitoring index at 40% torque level: (a) RNN; (b) ANFIS; (c) ARNFIS; (d) the proposed approach

Fig. 10. Two-step-ahead prediction results for the monitoring index at 100% torque level: (a) RNN; (b) ANFIS; (c) ARNFIS; (d) the proposed approach
Fig. 11. Four-step-ahead prediction results for the monitoring index at 40% torque level: (a) RNN; (b) ANFIS; (c) ARNFIS; (d) the proposed approach

Fig. 12. Four-step-ahead prediction results for the monitoring index at 100% torque level: (a) RNN; (b) ANFIS; (c) ARNFIS; (d) the proposed approach
bearing with spalling. The testing monitoring index is the Root Mean Square (RMS) value of the vibration signal in a frequency band, which is acquired from an accelerometer on-board a helicopter. Fig. 13 shows the four-step-ahead prediction results. It can be seen that the proposed predictor can track the system response effectively except at the end of the testing phase, which may be caused by the intensive dynamic fluctuations of the system during that period of time. It is clear that the prediction accuracy of the proposed predictor is higher than that of the three classical predictors. Table II gives the comparison result over several steps via the RMSE, which indicates the superior prediction accuracy of the proposed approach.

V. CONCLUSIONS

This paper has addressed a novel machine condition prognosis approach based on adaptive neuro-fuzzy inference systems (ANFIS) and high-order particle filtering. The ANFIS is trained via available condition data to model the fault propagation trend. A high-order particle filter is developed to carry out prediction, based on an m-th-order hidden Markov model that integrates the ANFIS with its modeling noise. Through the estimation of the Probability Density Function (PDF) of the ANFIS’s residuals between the actual and predicted condition data, an on-line update scheme is proposed to adapt this Markov model to various machine dynamics quickly. Experimental data from the main gearbox of a helicopter subjected to a seeded carrier crack fault and a faulty helicopter bearing are used to evaluate the prediction performance of the proposed approach. The results demonstrate that its prediction accuracy is higher than that of three classical predictors: recurrent neural networks (RNN), adaptive neuro-fuzzy inference systems (ANFIS) and adaptive recurrent neuro-fuzzy inference systems (ARNFIS).

REFERENCES


Chaochoa Chen received the B.E. and M.S.E. degrees in Mechanical Engineering from Yanshan University, Qinhuangdao, China, in 2001 and 2004, respectively, and the Ph.D. degree in Intelligent Mechanical Systems Engineering from Kochi University of Technology, Japan, in 2007. From 2008, he was a Research Associate in Electrical and Computer Engineering Department, University of Michigan-Dearborn. Since 2009, he has been a postdoctoral fellow at RSCF, Georgia Institute of Technology. His research interests include intelligent machine learning, fault diagnosis and failure prognosis, integrated system architecture development, fault-tolerant control and robotics.

Bin Zhang (M04-SM08) received the B.E. and M.S.E. degrees in mechanical engineering Nanjing University of Science and Technology, Nanjing, China, in 1993 and 1999, respectively, and the Ph.D. degree in electrical engineering from Nanyang Technological University, Singapore, in 2007. He has been a Postdoctoral Researcher in the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA before he joined Impact Technologies, LLC, Rochester, NY. He is the author and coauthor of more than 70 technical papers. His current research interests include fault diagnosis and failure prognosis, systems and control, digital signal processing, learning control, intelligent systems and their applications to robotics, power electronics, and various mechanical systems.

George Vachtsevanos (S62-M63-SM89) received the B.E.E. degree in electrical engineering from the City College of New York, New York, NY, in 1962, the M.E.E degree in electrical engineering from New York University, New York, in 1963, and the Ph.D. degree in electrical engineering from the City University of New York, New York, in 1970. He is currently a Professor Emeritus of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, where he directs the Intelligent Control Systems Laboratory. His work is funded by government agencies and industry. He is the author or coauthor of more than 240 technical papers.

Marcos Orchard received the B.Sc. degree and the Civil Industrial Engineering degree with Electrical Major from Catholic University of Chile, Santiago, Chile, in 1999 and 2001, respectively, and the M.S. and Ph.D. degree from the Georgia Institute of Technology, Atlanta, GA, in 2005 and 2007, respectively. He is currently an Assistant Professor in the Department of Electrical Engineering at the University of Chile. His current research interests include the design, implementation, and testing of real-time frameworks for fault diagnosis and failure prognosis.