Machine Remaining Useful Life Prediction Based on Adaptive Neuro-Fuzzy and High-Order Particle Filtering

Chaochao Chen[1], **George Vachtsevanos**[1], and **Marcos E. Orchard**[2]

[1] *Georgia Institute of Technology, Atlanta, GA 30332 USA*
chaochao.chen@gatech.edu
george.vachtsevanos@ece.gatech.edu

[2] *Electrical Engineering Department, Universidad de Chile, Santiago 8370451, Chile*
morchard@ing.uchile.cl

ABSTRACT

Machine remaining useful life (RUL) prediction is a key part of Condition-Based Maintenance (CBM), which provides the time evolution of the fault indicator so that maintenance can be performed to avoid catastrophic failures. This paper proposes a new RUL prediction method based on adaptive neuro-fuzzy inference systems (ANFIS) and high-order particle filtering, which predicts the time evolution of the fault indicator and computes the probability density function (pdf) of RUL. The ANFIS is trained and integrated in a high-order particle filter to describe the fault propagation process; the high-order particle filter uses real-time data to update the current state estimates so as to improve the prediction accuracy. The performance of the proposed method is evaluated via the real-world data from a seeded fault test for a UH-60 helicopter planetary gear plate. The results show that it outperforms the conventional ANFIS predictor.

1. INTRODUCTION

In machinery Condition-Based Maintenance (CBM), machine prognosis is an important component that possesses the ability to predict accurately and precisely the future condition and remaining useful life (RUL) of a failing component or subsystem (Vachtsevanos et al., 2006). Machine prognosis is considered as 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. Recently, numerous efforts have been reported in the machinery prognosis community (Gebraeel et al., 2004; Tse and Atherton, 1999; Zhao et al., 2009; Wang et al., 2004; Liu et al., 2009; Wang and Vachtsevanos, 2001; Samanta and Nataraj, 2008; Tran et al., 2009; Wang, 2007).

Machine prognostic methods can roughly be categorized into two major classes: model-based (or physics-based) and data-driven methods (Jardine et al., 2006). Model-based methods use mathematical models to predict the fault growth trend. Given a proper model for a specific system, model-based methods can provide accurate prediction estimates. However, it is usually difficult to develop accurate fault growth models in most real-world applications, 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.

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 (Zhao, 2009; Wang et al., 2004; Liu et al., 2009; Samanta and Nataraj, 2008; Tran et al., 2009; Wang, 2007). The superior forecasting performance of these predictors has been exhibited as compared to conventional neural-network-based predictors, i.e. radial-basis-function and recurrent-neural-network based models (Zhao, 2009; Wang et al., 2004). These studies also claimed that the ANFIS is a reliable and robust condition predictor that can capture the system's new dynamics quickly and accurately. However, all of these works only focused on the short-term condition prediction and the RUL prediction study is missed.

[*] This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Recently, particle-filtering-based prognostic approaches have been successfully employed in real-world applications (Orchard, 2009). In these 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 (such as a detailed Finite Element Analysis model) to estimate the values of the parameters of the fault growth model. Moreover, note that the fault propagation model adopted in these works is a first-order hidden Markov model, where the system's current state depends only on the previous state. In many applications, however, the assumption of a first-order Markov model requires to augment considerably the dimension of the state vector to properly describe the trend of the fault growth. In that case, it is more appropriate to consider a higher-order model where the current state not only depends 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 RUL prediction is carried out via a high-order particle filter, where a combination of the ANFIS and the process noise, as a high-order hidden Markov model, is employed to represent the fault growth process.

This paper provides a new approach for machine RUL prediction via the combination of the ANFIS and high-order particle filtering, which integrates the data-driven method (ANFIS) in the state-estimation framework (Particle filtering). Note that real-time data is utilized in the high-order particle filter to update the pdf of RUL, which improves the prediction accuracy.

The remainder of this paper is organized as follows: in the next section, the ANFIS is introduced to perform machine RUL prediction. Section 3 presents the proposed prediction approach. Bayesian estimation is introduced first, a high-order hidden Markov model and its posterior pdf are presented, and the integration of the ANFIS in a high-order particle filter is demonstrated. Then, the overall prediction algorithm is illustrated in detail. Section 4 presents the experimental results of the proposed approach and the performance comparison with the ANFIS predictor is given. Section 5 provides some concluding remarks.

2. THE ANFIS PREDICTOR

Recently, the ANFIS has been applied widely in the field of machinery condition prognosis, since it is able not only to learn highly nonlinear dynamics of machines without the necessity of deriving complex mathematical models but also to adapt itself to machines' new dynamics and capture machines' new dynamical behaviors quickly and accurately. For this predictor, the input variables, $\{x_{t-nr} \cdots x_{t-3r}\ x_{t-2r}\ x_{t-r}\ x_t\}$, and the output/forecasting variable, x_{t+r}, are "monitoring indices" that characterize the machine health condition, where r denotes the prediction step, i.e. when r=1, x_{t+r} 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}.

In general, there are two methods to model the ANFIS predictor. One is direct prediction and the other is recursive prediction. For short-term prediction, direct prediction method is always used, i.e. when r=5 and n=3, the input variables are $\{x_{t-15}\ x_{t-10}\ x_{t-5}\ x_t\}$ and the output is x_{t+5}; for RUL prediction, recursive prediction method is always utilized since we don not know how many step-ahead predicted values can reach the health condition threshold value (or failure indicator) at every prediction time instant. The detailed illustration of recursive prediction method is given below:

$$\hat{x}_{t+1} = f\left(x_t, x_{t-1}, \ldots, x_{t-n}\right) \tag{1}$$

where t denotes current time instant, $\hat{x}_{t+1}$ is the predicted value at next time instant t+1, f represents the ANFIS predictor.
To predict the value at time instant t+2, we use

$$\hat{x}_{t+2} = f\left(\hat{x}_{t+1}, x_t, \ldots, x_{t-n+1}\right) \tag{2}$$

where the previous predicted value $\hat{x}_{t+1}$ instead of the real value x_{t+1} is used to carry out the prediction for time instant t+2.
Similarly, long-term prediction can be performed in this way until the predicted value reaches a predefined failure threshold, like

$$\hat{x}_{t+l} \geq x_T \tag{3}$$

where x_T denotes the monitoring condition threshold value that indicates machine failures.

Therefore, according to Eqs. (1)-(3), the machine RUL at current time instant t can be obtained that equals to l.

Here, a hybrid learning algorithm that combines the gradient descent method and the least squares method is utilized to train the ANFIS. The training process is terminated when the number of training iterations has reached a predefined value or the desired training error has been achieved.

The ANFIS predictor is a fuzzy Sugeno model, whose parameters are optimized via neural network training and structure is determined by expert knowledge (Jang et al., 1997). Like many ANFIS applications in machinery prognosis (Zhao et al., 2009; Wang et al., 2004; Samanta and Nataraj, 2008; Tran et al., 2009), four input variables $\{x_{t-3r}\ x_{t-2r}\ x_{t-r}\ x_t\}$ 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, as shown below:

Rule j:
IF $(x_{t-3r}\ is\ A_1^j)$ AND $(x_{t-2r}\ is\ A_2^j)$ AND $(x_{t-r}\ is\ A_3^j)$ AND $(x_t\ is\ A_4^j)$,
THEN $y^j = c_1^j x_{t-3r} + c_2^j x_{t-2r} + c_3^j x_{t-r} + c_4^j x_t + c_5^j$;
$j = 1,2,\ldots,16$.

where y^j is the prediction result according to the jth fuzzy rule, A_i^j is the fuzzy set associated with the ith input variable in the jth fuzzy rule, and c_k^j is the parameter that is determined by the learning process, here, $i = 1,2,\ldots,4$ and $k = 1,2,\ldots,5$.

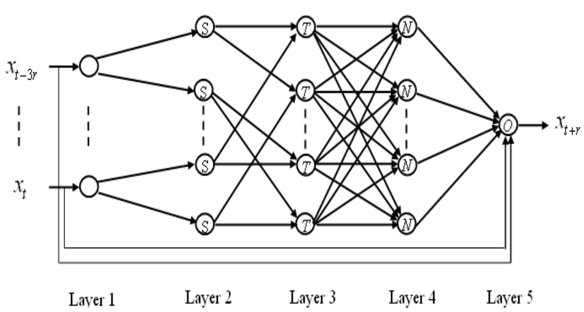

Figure 1: Architecture of the ANFIS predictor; S is a sigmoid function; O is an operator defined in Eq. (8).

The ANFIS predictor consists of five layers. Its architecture is schematically shown in Fig. 1. The signal propagation is illustrated as follows:

In the following description, $x_i^{(k)}$ defines the ith node input in the kth layer, and $y_i^{(k)}$ denotes the ith node output in the kth layer,

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

$$y_i^{(1)} = x_i^{(1)}, \quad i = 1,2,\ldots,4. \tag{4}$$

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

$$u_{A_i^j}^{(2)}\left(x_i^{(1)}\right) = \frac{1}{1+\exp\left(-b_{ij}^{(2)}\left(x_i^{(1)} - m_{ij}^{(2)}\right)\right)}, \quad \begin{array}{l} i = 1,2,\ldots,4, \\ j = 1,2,\ldots,16 \end{array} \tag{5}$$

where $u_{A_i^j}^{(2)}$ is the output signal with respective to the ith input variable $x_i^{(1)}$ in the jth fuzzy rule, $b_{ij}^{(2)}$ and $m_{ij}^{(2)}$ 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

$$y_j^{(3)} = \prod_i u_{A_i^j}^{(2)}\left(x_i^{(1)}\right), \quad i = 1,2,\ldots,4,\ j = 1,2,\ldots,16. \tag{6}$$

where the output $y_j^{(3)}$ represents the firing strength of the jth fuzzy rule.

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

$$y_j^{(4)} = \frac{y_j^{(3)}}{\sum_j y_j^{(3)}}, \quad j = 1,2,\ldots,16. \tag{7}$$

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

$$x_{t+r} = \sum_j y_j^{(4)} (c_1^j x_{t-3r} + c_2^j x_{t-2r} + c_3^j x_{t-r} + c_4^j x_t + c_5^j),\ j = 1,2,\ldots,16. \tag{8}$$

where $\{c_1^j, c_2^j, c_3^j, c_4^j, c_5^j\}$ 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 $\{c_1^j, c_2^j, c_3^j, c_4^j, c_5^j\}$ are optimized by using the least square method, whereas the premise parameters, $b_{ij}^{(2)}$ and $m_{ij}^{(2)}$, are updated via the gradient descent method.

3. PROPOSED PREDICTION ALGORITHM

The proposed prediction approach is discussed in this section. The Bayesian estimation technique using an m^th^-order hidden Markov model 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. Lastly, a thorough description of the algorithm steps is presented.

3.1 Bayesian Estimation Using mth-Order Markov Models

Through the use of noisy observation data, a Bayesian estimation technique is intended to estimate a state vector governed by a dynamic nonlinear state-space model (or a hidden Markov 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 hidden Markov model. In general, a first-order Markov model is used to describe the fault growth process. Here, instead of using a first-order model, an mth-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 mth-order model

$$x_k = f_k\left(x_{k-1}, x_{k-2}, \ldots, x_{k-m}, \omega_{k-1}\right) \tag{9}$$

where x_k is the model state at time *k*, x_{k-m} is the state at time *k*-*m*, ω_{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

$$y_k = h_k\left(x_k, v_k\right) \tag{10}$$

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.

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:

$$p\left(x_{0:k} \middle| y_{1:k-1}\right) = \int p\left(x_k \middle| x_{k-m:k-1}\right) p\left(x_{0:k-1} \middle| y_{1:k-1}\right) dx_{0:k-1} \tag{11}$$

where the probabilistic process model $p\left(x_k \middle| x_{k-m:k-1}\right)$ is defined via Eq. (9), and $p\left(x_{0:k-1} \middle| y_{1:k-1}\right)$ represents the state pdf at time *k*-1. Note that in Eq. (11), the fact that $p\left(x_k \middle| x_{0:k-1}, y_{1:k-1}\right) = p\left(x_k \middle| x_{k-m:k-1}\right)$ is used according to the Markov properties on the moral graph of the mth-order hidden Markov model (Whittaker, 1990).

When a new measurement becomes available, the update step is carried out. By considering the new measurement, the prior state pdf, the likelihood function $p\left(y_k \middle| x_k\right)$, and Bayes' rule, the posterior state pdf can be calculated by

$$\begin{aligned} p\left(x_{0:k} \middle| y_{1:k}\right) &= \frac{p\left(y_k \middle| x_k\right) p\left(x_{0:k} \middle| y_{1:y-1}\right)}{p\left(y_k \middle| y_{1:k-1}\right)} \\ &= \frac{p\left(y_k \middle| x_k\right) p\left(x_k \middle| x_{k-m:k-1}\right) p\left(x_{0:k-1} \middle| y_{1:k-1}\right)}{p\left(y_k \middle| y_{1:k-1}\right)} \\ &\propto p\left(y_k \middle| x_k\right) p\left(x_k \middle| x_{k-m:k-1}\right) p\left(x_{0:k-1} \middle| y_{1:k-1}\right) \end{aligned} \tag{12}$$

where the normalizing constant is

$$p\left(y_k \middle| y_{1:k-1}\right) = \int p\left(y_k \middle| x_k\right) p\left(x_k \middle| y_{1:k-1}\right) dx_k$$

and the likelihood function $p\left(y_k \middle| x_k\right)$ is defined by the measurement model (10).

3.2 Integration of ANFIS in High-Order Particle Filtering

The recursive computation of the posterior state pdf $p\left(x_{0:k} \middle| y_{1:k}\right)$ is more conceptual than practical, since the integrals in Eqs. (11) and (12) do not have an analytical solution in most cases. Therefore, many estimation methods have been developed to solve this problem (Arulampalam et al., 2002). 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 (Arulampalam et al., 2002)

$$p\left(x_{0:k} \middle| y_{1:k}\right) \approx \sum_{i=1}^{N} w_k^i \delta\left(x_{0:k} - x_{0:k}^i\right) \tag{13}$$

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*, $\{x_{0:k}^i, i = 1,2\ldots,N\}$ is a set of particles with associated weights $\{w_k^i, i = 1,2,\ldots,N\}$, 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 $q\left(x_{0:k} \middle| z_{1:k}\right)$, the normalized weights are updated as (Arulampalam et al., 2002)

$$w_k^i \propto \frac{p\left(x_{0:k}^i \middle| y_{1:k}\right)}{q\left(x_{0:k}^i \middle| y_{1:k}\right)} \tag{14}$$

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

$$\begin{aligned} q\left(x_{0:k} \middle| y_{1:k}\right) &= q\left(x_k \middle| x_{0:k-1}, y_{1:k}\right) q\left(x_{0:k-1} \middle| y_{1:k-1}\right) \\ &= q\left(x_k \middle| x_{k-m:k-1}, y_k\right) q\left(x_{0:k-1} \middle| y_{1:k-1}\right) \end{aligned} \tag{15}$$

By substituting Eqs. (12) and (15) into (14), we obtain

$$w_k^i \propto w_{k-1}^i \frac{p\left(y_k \middle| x_k^i\right) p\left(x_k^i \middle| x_{k-m:k-1}^i\right)}{q\left(x_k^i \middle| x_{k-m:k-1}^i, y_k\right)} \tag{16}$$

If we simply choose

$$q\left(x_k^i \middle| x_{k-m:k-1}^i, y_k\right) = p\left(x_k^i \middle| x_{k-m:k-1}^i\right) \tag{17}$$

and substitute Eq. (17) into (16), then yields

$$w_k^i \propto w_{k-1}^i p\left(y_k \middle| x_k^i\right) \tag{18}$$

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 Eqs. (4)-(8), has four previous state values as the inputs. Therefore, the 4th-order hidden Markov model that presents the fault growth process is described as follows:

$$x_k = \hat{x}_k + \omega_{k-1} \tag{19}$$

$$\hat{x}_k = g_k\left(x_{k-1}, x_{k-2}, x_{k-3}, x_{k-4}\right) \tag{20}$$

where $g_k\left(x_{k-1}, x_{k-2}, x_{k-3}, x_{k-4}\right)$ is a nonlinear function denoted by the ANFIS.

The current and three previous states of the hidden Markov model, i.e., $x_{k-1}, x_{k-2}, x_{k-3}, x_{k-4}$ in Eq. (20), are the four inputs of the ANFIS. Accordingly, the output of the ANFIS $\hat{x}_k$ plus its process noise ω_{k-1} is the state of hidden Markov model at the next time, as shown in Eq. (19).

3.3 Algorithm Steps

The detailed algorithm steps for RUL prediction are stated as:

For $i = 1, 2, 3, \ldots, N$

  Draw $x_k^i \sim q\left(x_k \middle| x_{k-m:k-1}^i, y_k\right)$ according to Equation (19)

End for i

Carry out prediction using Equations (21)-(23)

For $i = 1, 2, 3, \ldots, N$

  Calculate weight w_k^i using Equation (18)

End for i

Normalize weights $w_k^i = w_k^i \Big/ \sum_{i=1}^N w_k^i$

Calculate degeneracy measure $\hat{N}_{eff} = 1 \Big/ \sum_{i=1}^N \left(w_k^i\right)^2$

If $\hat{N}_{eff} < N_T$, where N_T is a threshold

  Resample $\left\{x_{k-m+1:k}^i, w_k^i\right\}_{i=1}^N$

End for If

Figure 2: Sequential Importance Sampling and Resampling algorithm, where $k = 1, 2, 3, \ldots, T$

Step 1: The ANFIS is trained via condition data pairs to model the fault propagation process.

Step 2: The fault growth model (19), 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:

$$\tilde{x}_k = \sum_{i=1}^N w_{k-1}^i x_k^i \tag{21}$$

The long-term (p-step-ahead) condition prediction also can be computed by successively taking the expectation of the model update Eq. (19) 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:

$$\tilde{x}_{k+p} = E\left[x_{k+p}^i\right], \quad x_{k+p}^i = \hat{x}_{k+p}^i + \omega_{k+p-1}. \tag{22}$$

When all of the predicted values associated with each particle reach the predefined condition threshold, the expected value of RUL can be obtained from the RUL pdf, as shown below:

$$RUL = \sum_{i=1}^{N} l^i w_{k-1}^i \quad (23)$$

where N is the total number of particles, l^i is the RUL of the ith particle, and w_{k-1}^i is the weight of the ith particle at time instant $k-1$, When a new measurement becomes available, the weights can be calculated according to Eq. (18). If severe degeneracy does exist, resampling is performed.

Step 3: Repeat *Step2* until machine RUL prediction is complete.

Here, *step* 2 can be considered as the execution of a Sequential Importance Sampling and Resampling algorithm that is summarized in Fig. 2. The flowchart of the proposed algorithm is shown in Fig. 3.

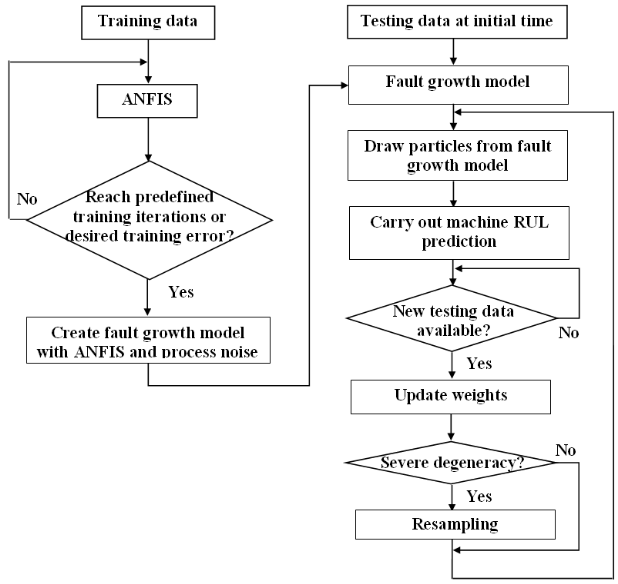

Figure 3: Flowchart of proposed algorithm for machine RUL prediction

4. EXPERIMENTAL STUDIES

The proposed prediction algorithm is validated via vibration data from a seeded fault test for a UH-60 helicopter planetary gear plate.

In order to evaluate the prediction performance, the average prediction error $\bar{e}$ employed in (Tian et al., 2009) is used:

$$\bar{e} = \frac{1}{n} \sum_{k=1}^{n} |y_k - \hat{y}_k| \quad (24)$$

where n is the total number of the time instants at that RUL prediction is carried out, y_k and $\hat{y}_k$ are the actual and predicted RUL values at time instant k, respectively

The smaller value of $\bar{e}$ indicates higher prediction accuracy.

4.1 System Condition Monitoring

The main transmission of UH-60 “Black Hawk” employs a five-planet epicyclic gear system. Recently, a crack in the planetary carrier plate was discovered during regular maintenance, as shown in Fig. 4. 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 RUL 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. 5.

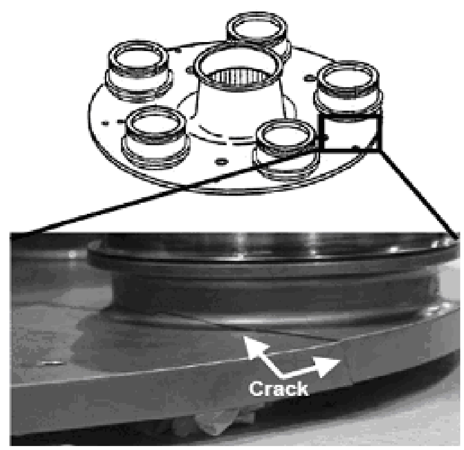

Figure 4: Crack of planetary gear carrier plate

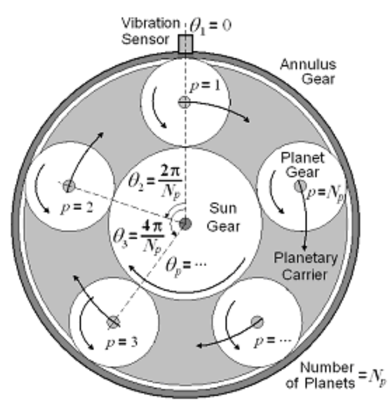

Figure 5: Configuration of an epicyclic gear system

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. 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 (Zhang et al., 2008):

$$SBR(X)=\frac{\sum_{k=1}^{m}\sum_{g=-X}^{X}NonRMC}{\sum_{k=1}^{m}\sum_{g=-X}^{X}(NonRMC+RMC)} \quad (25)$$

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

4.2 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 cycles, the vibration feature (or monitoring index) at 100% torque level is extracted and used for training and testing the ANFIS. Here, failure is defined when the crack reaches 6.21 inches (or at the 714th GAG cycle), i.e., the crack reaches the edge of the carrier plate. Accordingly, the failure threshold values for monitoring indices at different torque levels are obtained. The RUL predictions are carried out from the 366th GAG cycle, using the current estimate for the state pdf as initial condition. The predictions are terminated at the 714th GAG cycle when a failure occurs.

Fig. 6 shows the prognosis results of the proposed algorithm at the 366th GAG cycle. The above subfigure indicates each plausible long-term prediction trajectory associated to every particle with a different color. The horizontal blue line defines the failure threshold value, i.e., 3.46 at 100% torque level. When each predicted trajectory reaches this threshold, the resulting time instants form a Time To Failure (TTF) pdf, as shown in the below subfigure. The vertical black line denotes the TTF expectation. In this example, the expected RUL is 336 GAG cycles that is very close to the real RUL 348 GAG cycles.

Fig. 7 shows the RUL prediction results for the ANFIS and proposed algorithm using the features at 100% torque level. Both methods provide satisfactory

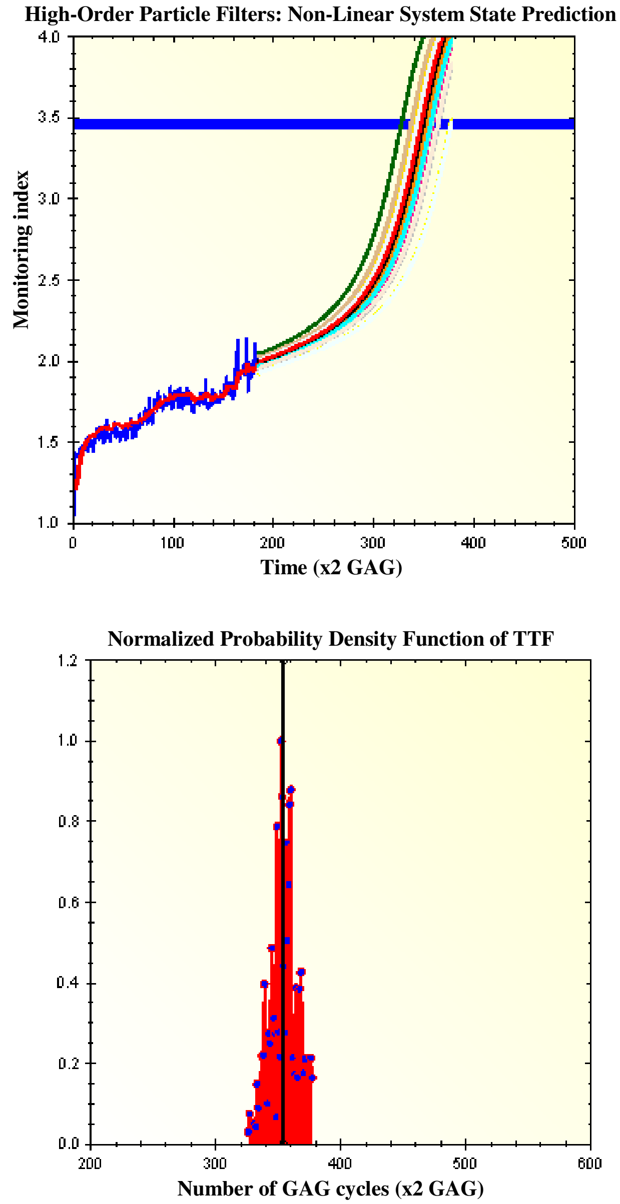

Figure 6: Prognosis results at the 366th GAG cycle using proposed algorithm, TTF means time to failure.

prediction results. But the proposed algorithm apparently outperforms the ANFIS. The average RUL prediction error $\overline{e}$ for the proposed predictor is 20.3448; the error for the ANFIS is 26.3218.

When the high-order particle filter in the proposed predictor is replaced by a conventional one, that is, the fault growth model becomes a first-order Markov model (note that this model is also trained via the ANFIS), the RUL prediction results using the features at 100% torque level are shown in Fig. 8. As can be seen, its prediction accuracy is much lower than that of the proposed predictor (Fig 7(a)), which means that the

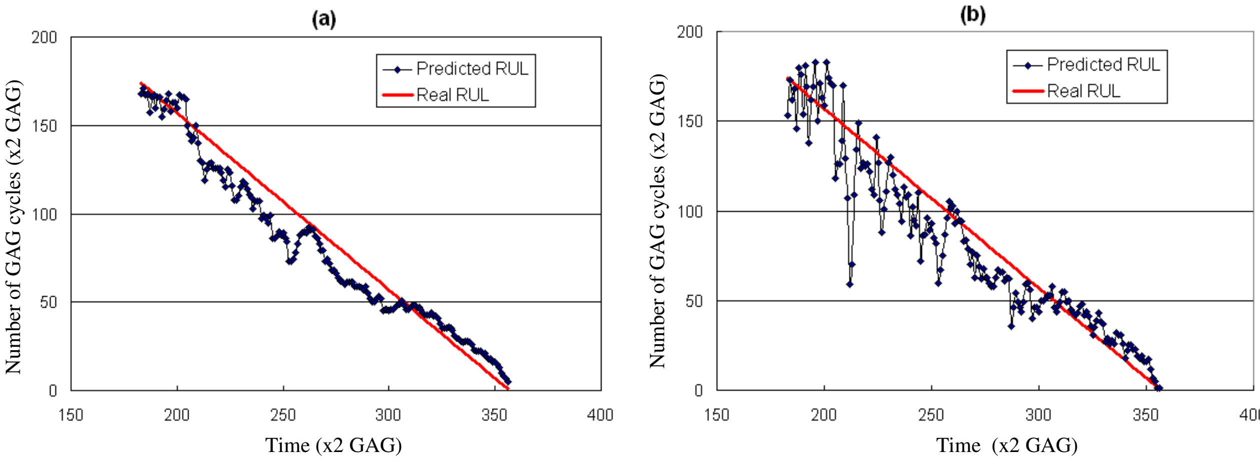

Figure 7: RUL prediction results at 100% torque level. (a): proposed algorithm; (b): ANFIS.

trained first-order model can not accurately capture the long-term prediction dynamics of the fault.

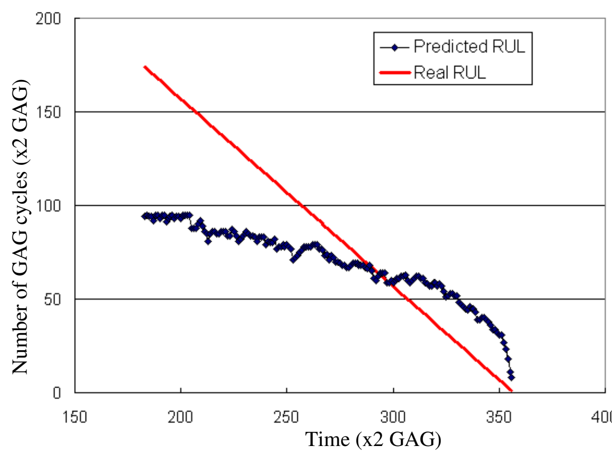

Figure 8: RUL prediction results at 100% torque level using first-order particle filter

5. CONCLUSION

This paper proposes a novel machine remaining useful life (RUL) prediction approach using adaptive neuro-fuzzy inference systems (ANFIS) and high-order particle filtering. Due to the inherent characteristics of capturing new dynamics of machines quickly, the ANFIS is used to model the fault propagation trend. A high-order particle filter is developed to integrate the ANFIS, as an m^{th}-order hidden Markov model, to carry out the predictions. Experimental data from the main gearbox of a UH-60 helicopter subjected to a seeded carrier plate crack fault is used to evaluate the prediction performance of the proposed approach. The results demonstrate that its prediction accuracy is higher than that of the ANFIS predictor and the particle-filter-based predictor where the fault growth model is a first-order model that is trained via the ANFIS.

REFERENCES

Vachtsevanos, G., Lewis, F. L., Roemer, M., Hess, A. & Wu, B. (2006). *Intelligent Fault Diagnosis and Prognosis for Engineering Systems*, 1st ed. Hoboken, New Jersey: John Wiley & Sons, Inc,.

Gebraeel, N., Lawley, M., Liu, R., Parmeshwaran, V. (2004). Residual life predictions from vibration-based degradation signals: a neural network approach, *IEEE Transactions on Industrial Electronics*, vol. 51, pp. 694-700.

Tse, P., Atherton, D. (1999). Prediction of machine deterioration using vibration based fault trends and recurrent neural networks, *Journal of Vibration and Acoustics,* vol. 121, pp.355-362.

Zhao, F., Chen, J., Guo, L., Lin, X. (2009). Neuro-fuzzy based condition prediction of bearing health, *Journal of Vibration and Control*, vol. 15, pp. 1079-1091.

Wang, W., Golnaraghi, F., Ismail, F. (2004). Prognosis of machine health condition using neuro-fuzzy systems, *Mechanical System and Signal Processing*, vol. 18, pp. 813-831.

Liu, J., Wang, W., Golnaraghi, F. (2009). A multi-step predictor with a variable input pattern for system state forecasting, *Mechanical System and Signal Processing,* vol. 23, pp. 1586-1599.

Wang, P., Vachtsevanos, G. (2001). Fault prognostics using dynamic wavelet neural networks, *Artificial Intelligence for Engineering Design*, *Analysis and Manufacturing*, vol. 15, pp. 349-365.

Samanta, B., Nataraj, C. (2008). Prognostics of machine condition using soft computing, *Robotics and Computer-Integrated Manufacturing.,* vol. 24, pp. 816-823.

Tran, V., Yang, B., Tan, A. (2009). Multi-step ahead direct prediction for machine condition prognosis using regression trees and neuro-fuzzy systems, *Expert System with Appication,* vol. 36, pp. 9378-9387.

Wang, W. (2007). An adaptive predictor for dynamic system forecasting, *Mechanical System and Signal Processing,* vol. 21, pp. 809-823.

Jardine, A.K.S., Lin, D., Banjevic, D. (2006). A review on machinery diagnostics and prognostics implementing condition-based maintenance, *Mechanical System and Signal Processing,* vol. 20, pp. 1483-1510.

Arulampalam, M.S., Maskell, S., Gordon, N., Clapp T. (2002). A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking, *IEEE Transactions on Signal Processing.,* vol. 50, pp.174–188.

Orchard, M., and Vachtsevanos, G. (2009). "A Particle Filtering Approach for On-Line Fault Diagnosis and Failure Prognosis," *Transactions of the Institute of Measurement and Control*, vol. 31, no. 3-4, pp. 221-246.

Jang, J. -S. R., Sun, C. -T., Mizutani, E. (1997). *Neuro-Fuzzy and Soft Computing*. NJ: Prentice-Hall PTR.

Whittaker, J. (1990) *Graphical Models in Applied Mathematical Multivariate Statistics*, United Kingdom: John Wiley & Sons.

Tian, Z. Wong, L. Safaei, N. (2009). A neural network approach for remaining useful life prediction utilizing both failure and suspension histories, *Mechanical System and Signal Processing* doi:10.1016/j.ymssp.2009.11.005.

Zhang, B. Khawaja, T. Patrick, R. Vachtsevanos, G. (2008). Blind deconvolution denoising for helicopter vibration signals, *IEEE/ASME Transactions on Mechatronics*, vol. 13, pp. 558-565.

Chaochao 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 ICSL, Georgia Institute of Technology. His research interests include intelligent machine learning, fault diagnosis and failure prognosis, system software architecture development, fault-tolerant control and robotics.

George Vachtsevanos 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 E. Orchard is an Assistant Professor in the Department of Electrical Engineering at the University of Chile and collaborated as a Ph.D. Student in the Intelligent Control Systems Laboratory at The Georgia Institute of Technology. His current research interest is the design, implementation and testing of real-time frameworks for fault diagnosis, failure prognosis, and Automated Contingency Management systems. He received his Ph.D. and MS degrees from The Georgia Institute of Technology, Atlanta, GA, in 2005 and 2007 respectively. He received his B.Sc. degree (1999) and a Civil Industrial Engineering degree with Electrical Major (2001) from Catholic University of Chile.