AN ADAPTIVE NF PREDICTOR FOR GEAR SYSTEM STATE FORECASTING

[1]ANDREW KADIK, [2]WILSON WANG

[1]Dept. of Electrical & Computer Engineering, Lakehead University, Thunder Bay, ON, Canada, P7B 5E1
[2]Dept. of Mechanical Engineering, Lakehead University, Thunder Bay, ON, Canada
E-mail: [1]akadik@Lakeheadu.ca, [2]wwang3@lakeheadu.ca

Abstract - A reliable predictor is very useful in industries to forecast the future states of a dynamic system. In this paper, an adaptive predictor is developed based on a weighted neuro-fuzzy (NF) system to forecast properties of dynamic gear systems. An online training technique is adopted to improve forecasting convergence and adaptive capability. The effectiveness of the developed predictor is evaluated based on some benchmark data set; then it is implemented for gear system monitoring. The monitoring index is derived from measurement based on a beta kurtosis reference function. The investigation results show that the developed adaptive NF predictor is a reliable forecasting tool. It can capture the system's dynamic behavior quickly and track the system's characteristics accurately.

Index Terms - Neuro-fuzzy (NF) system, system state prediction, machine learning, gear system condition monitoring.

I. INTRODUCTION

A reliable forecasting tool is very useful to a wide array of industries to forecast the future states of a dynamic system. For example, in machinery condition monitoring, it can provide an early alarm before a fault reaches critical levels so as to prevent machinery performance degradation, or even catastrophic failures to the machinery or human beings. The diagnostic information is also very useful in scheduling repairs and maintenance operations for faulty manufacturing facilities.

System state forecasting is a process to utilize available observations to predict the future states of a dynamic system. The representative observations can be derived by using proper signal processing techniques. Several techniques have been proposed in the literature for time series prediction, such as based on the classical stochastic dynamic models and data-driven models such as neural networks (NNs) and neural fuzzy paradigms [1,2]. The authors' research group also developed several flexible model-based predictors for machinery applications [3,4]; their investigation indicated that if a neuro-fuzzy (NF) predictor is properly trained, it outperforms both the feedforward and the recurrent network forecasting schemes.

Even though the data-driven NF predictors have demonstrated their superior potential to other classical forecasting tools, advanced research needs to be done in improving the convergence and adaptive capability to accommodate different system operating conditions, especially for machinery health condition monitoring [5,6]. As gearboxes are commonly used in rotating machinery, the objective of this work is to develop an adaptive neuro-fuzzy (ANF) predictor for gear system state forecasting. The forecasting information can be used not only for gear system health state estimation, but also used for repair and maintenance applications. The monitoring indices are derived from advanced signal processing techniques. The forecasting reasoning is performed by a weighted ANF scheme, whereas the fuzzy system parameters are optimized by an online machine learning algorithm. Its effectiveness is verified by both simulation, and real-time tests corresponding to gear system monitoring.

II. THE ADAPTIVE NEURO-FUZZY (ANF) PREDICTOR

The ANF reasoning applies a set of fuzzy IF-THEN rules to describe the input/output data mapping. Consider (n+1) input state variables $\{x_0\ x_{-s}\ x_{-2s}\ \cdots\ x_{-ns}\}$, which are normalized in [0, 1]; x_0 represents the current state of a dynamic system and s denotes the time steps. To simplify implementation, two membership functions (MFs), small and large are used for fuzzification. The s-step-ahead prediction of a dynamic system x_{+s} can be formulated by:

$$\Re_j:\ \text{IF}\ (x_0\ \text{is}\ A_0^j)\ \text{AND}\ (x_{-s}\ \text{is}\ A_1^j)\ \text{AND}\ \cdots\ \text{AND}\ (x_{-ns}\ \text{is}\ A_n^j)\quad \text{THEN}\ x_{+s}=f_j \tag{1}$$

where $\Re_j$ denotes the jth fuzzy rule, j = 1, 2, $\cdots$, M, M is the total number of fuzzy rules; A_i^j is the jth fuzzy set for x_{-is}, i = 0, 1, $\cdots$, n.

A first-order TS model is used for this analysis: $f_j = c_0^j x_0 + c_1^j x_{-s} + \cdots + c_n^j x_{-ns} + c_{n+1}^j$, where $c_\bullet^j$ are constants. The network architecture of the proposed

ANF predictor is schematically shown in Fig. 1. It is a 6-layer network in which each node performs a particular activation function on the incoming signals. The links have unity weights unless specified. The nodes in layer 1 transmit input signals to the next layer. Each node in layer 2 acts as an MF. This ANF has a weighted feedback link to each node in layer 2. The feedback links can deal with time explicitly. The context units copy the activations of output nodes from the previous time step, and allow the network to remember cues from the past, which forms a context for current processing. This function of recurrent networks is even more valuable for predictors with limited inputs so as to improve forecasting accuracy.

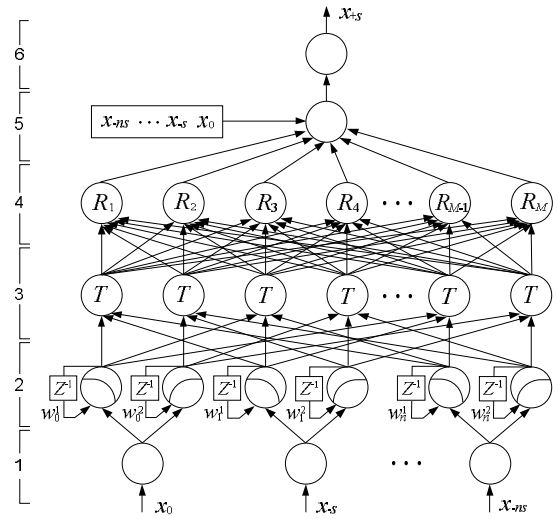


Fig. 1. Network architecture of the ANF predictor.

If sigmoid MFs with parameters $\{a_i^j \;\; b_i^j\}$ are utilized in this case, in which $a_i^j > 0$ and $a_i^j < 0$ correspond to a large and a small MF, respectively, then at the kth process step, the node output is

$$\mu_{A_i^j}(X_{-is}) = \frac{1}{1+\exp[-a_i^j(X_{-is}-b_i^j)]} \quad , \qquad (2)$$

$$X_{-is} = x_{-is}^{(k)} + w_i^m \mu_{A_i^j}(x_{-is}^{(k-1)})$$

$$= x_{-is}^{(k)} + \frac{w_i^m}{1+\exp[-a_i^j(x_{-is}^{(k-1)}-b_i^j)]} \qquad (3)$$

where m = 1, 2; i = 0, 1, . . ., n. $x_{-is}^{(k)}$ and $x_{-is}^{(k-1)}$ are the input x_{-is} at the k-th and (k-1)-th time steps, respectively. k = 1, 2, . . ., K, K is the total number of training data sets.

Each node in layer 3 performs a fuzzy T-norm operation. If a max-product operator is applied in this case, the rule firing strength is

$$\mu_j = \prod_{i=0}^{n} \mu_{A_i^j}(X_{-is}) \;\; , \;\; \text{j = 1, 2, . . ., M.} \qquad (4)$$

The rule firing strengths are normalized in layer 4. After combination of the input variables in layer 5, the predicted output x_{+s} is predicted by centroid defuzzification:

$$x_{+s} = \sum_{j=1}^{M} \overline{\mu}_j (c_0^j x_0 + c_1^j x_{-s} + \cdots + c_n^j x_{-ns} + c_{n+1}^j) \quad (5)$$

where $\overline{\mu}_j = \dfrac{\mu_j}{\sum_{j=1}^{M} \mu_j}$ is the normalized firing strength of the jth rule.

The fuzzy system parameters A_i^j , w_i^m and c_i^j are optimized by using an adaptive machine learning algorithm as discussed in the next section.

III. THE ADAPTIVE TRAINING PROCESS

Once the ANF predictor structure is created, the forecasting paradigm should be properly trained to make a better input/output mapping. For offline training, representative training data should cover all the possible application conditions [5,6]. Such a requirement is difficult to achieve in real-world applications because most machinery operates in noisy environment. Machinery dynamic characteristics may change suddenly, for example, just after repair or regular maintenance. In this work, an online machine learning algorithm is adopted for this multi-step-ahead ANF predictor, which has the following merits: 1) it possesses randomness that may help escape some local minima; 2) online training is necessary for time-varying systems.

A. An adaptive gain function

The training data sets are represented as

$$\boldsymbol{T}_d = \begin{bmatrix} \vdots & & & \\ x_0^{(k-1)} & x_{-s}^{(k-1)} & \cdots & x_{-ns}^{(k-1)} \\ x_0^{(k)} & x_{-s}^{(k)} & \cdots & x_{-ns}^{(k)} \\ x_0^{(k+1)} & x_{-s}^{(k+1)} & \cdots & x_{-ns}^{(k+1)} \\ \vdots & & & \end{bmatrix}. \qquad (6)$$

Consider the kth input data pair $\{x_0^{(k)} \;\; x_{-s}^{(k)} \cdots x_{-ns}^{(k)}\}$, k = 0, 1, …, K-1, K is the total number of training data pairs. The forecast state after s steps $x_{+s}^{(k)}$ is computed by Eq. (5)

$$x_{+s}^{(k)} = \sum_{j=1}^{M} \overline{\mu}_j (c_0^j x_0^{(k)} + c_1^j x_{-s}^{(k)} + \cdots + c_3^j x_{-ns}^{(k)} + c_{n+1}^j) \,. \qquad (7)$$

Different from general online training algorithms that use a constant forgetting factor [7], this adaptive training method will use an exponential gain function to highlight the contribution of the recent data sets from a time-variant series

$$g_k = \frac{1}{1+\exp[u(v-k)]}\ , \qquad (8)$$

where $g_k \in [0,1]$; $u = 10/K$ and $v = K/3$ in this case. Fig. 2 shows the comparison of this proposed gain function versus the commonly used forgetting factors. It is seen that the proposed gain function is more reasonable for data processing.

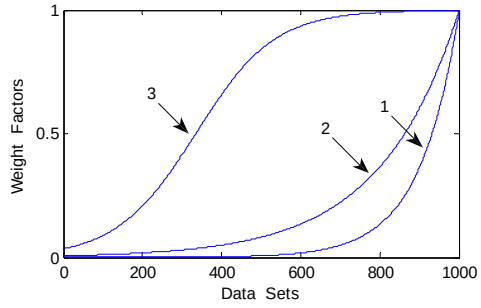


Fig. 2. Weight functions: 1) based on a forgetting factor 0.99; 2) based on a forgetting factor 0.995; 3) based on the proposed gain function.

Based on Eq. (8), the recent data sets are given more gain factors, whereas the old data pairs contribute less to training. All the gain factors are recognized in a gain matrix G:

$$\boldsymbol{G} = \begin{bmatrix} g_1 & 0 & 0 & \cdots & 0 \\ 0 & g_2 & 0 & \cdots & 0 \\ & & & \ddots & \\ 0 & 0 & 0 & \cdots & g_K \end{bmatrix}, \qquad (9)$$

where $g_K = 1$. In real applications, usually data pairs over a limited window period are used.

If $d^{(k)}$ denotes the actually observed output value for the kth training data set, or $d^{(k)} = x_0^{(k+1)}$, the forecasting error will be

$$\boldsymbol{E} = \sum_{k=1}^{K} E_k = \frac{1}{2}\sum_{k=1}^{K} g_k (x_{+s}^{(k)} - d^{(k)})^2\ . \qquad (10)$$

B. Training of premise parameters

The premise fuzzy system parameters are trained by using the gradient algorithm. That is, the MF parameters $\{a_i^j \ \ b_i^j\}$ and recurrent weight factors w_i^m are updated by:

$$a_i^j(p) = a_i^j(p-1) - \eta_a \sum_{k=1}^{K} \frac{\partial E_k}{\partial a_i^j} \qquad (11)$$

$$b_i^j(p) = b_i^j(p-1) - \eta_b \sum_{k=1}^{K} \frac{\partial E_k}{\partial b_i^j} \qquad (12)$$

$$w_i^m(p) = w_i^m(p-1) - \eta_w \sum_{k=1}^{K} \frac{\partial E_k}{\partial w_i^m} \qquad (13)$$

where p denotes the pth training epoch, p = 1, 2, . . ., P; P is the total number of training epochs; η_a, η_b and η_w are the corresponding step sizes.

C. Recursive training of consequent parameters

Given the values of the MF parameters and K training data pairs to the adaptive predictor, $\{x_0^{(k)} \ x_{-s}^{(k)} \cdots x_{-ns}^{(k)} \ d^{(k)}\}$, $d^{(k)} = x_0^{(k+1)}$, k = 0, 1, 2, . . ., K-1, K linear equations in terms of the consequent parameters $\boldsymbol{\vartheta}$ can be formed as:

$$\boldsymbol{A\theta} = \boldsymbol{d}\ , \qquad (14)$$

where R is the resulting matrix from the inference operation of the adaptive predictor:

$$\boldsymbol{R} = \begin{bmatrix} \bar{\mu}_1^{(1)}x_0^{(1)} & \bar{\mu}_1^{(1)}x_{-s}^{(1)} & \bar{\mu}_1^{(1)}x_{-2s}^{(1)} & \cdots & \bar{\mu}_M^{(1)}x_{-(n-1)s}^{(1)} & \bar{\mu}_M^{(1)}x_{-ns}^{(1)} & \bar{\mu}_M^{(1)} \\ \bar{\mu}_1^{(2)}x_0^{(2)} & \bar{\mu}_1^{(2)}x_{-s}^{(2)} & \bar{\mu}_1^{(2)}x_{-2s}^{(2)} & \cdots & \bar{\mu}_M^{(2)}x_{-(n-1)s}^{(2)} & \bar{\mu}_M^{(2)}x_{-ns}^{(2)} & \bar{\mu}_M^{(2)} \\ & & & \vdots & & & \\ \bar{\mu}_1^{(K)}x_0^{(K)} & \bar{\mu}_1^{(K)}x_{-s}^{(K)} & \bar{\mu}_1^{(K)}x_{-2s}^{(K)} & \cdots & \bar{\mu}_M^{(K)}x_{-(n-1)s}^{(K)} & \bar{\mu}_M^{(K)}x_{-ns}^{(K)} & \bar{\mu}_M^{(K)} \end{bmatrix} \qquad (15)$$

$\boldsymbol{\vartheta}$ is the consequent parameter set whose elements are to be updated:

$$\boldsymbol{\vartheta} = \begin{bmatrix} c_0^1 & c_1^1 & c_2^1 & \cdots & c_n^M & c_{n+1}^M \end{bmatrix}^T; \qquad (16)$$

d represents the vector of the actually observed system states:

$$\boldsymbol{d} = \begin{bmatrix} d^{(1)} & d^{(2)} & \cdots & d^{(K)} \end{bmatrix}^T. \qquad (17)$$

Because the row vectors in R and the associated elements in R are obtained sequentially, $\boldsymbol{\vartheta}$ in Eq. (16) can be computed recursively. Let $\boldsymbol{r}_k$ denote the kth row vector in matrix R, $\boldsymbol{R}_k$ be a submatrix of R composed by the first kth rows of R, $\boldsymbol{d}_k$ be the subvector of d composed by the first kth elements of d, $d^{(k)}$ be the kth element of d, $\boldsymbol{G}_k$ be a submatrix of G composed by the first kth rows of G. Then the current output of the predictor is

$$x_{+s}^{(k)} = \boldsymbol{r}_k \boldsymbol{\vartheta}_k \qquad (18)$$

where k = 0, 1, . . ., K-1. Correspondingly, Eq. (10) can be rewritten as

$$\boldsymbol{E} = \frac{1}{2}\sum_{k=0}^{K-1} g_k (\boldsymbol{r}_k \boldsymbol{\vartheta}_k - d^{(k)})^2 = \frac{1}{2}(\boldsymbol{R\vartheta} - \boldsymbol{d})^T \boldsymbol{G}(\boldsymbol{R\vartheta} - \boldsymbol{d})$$
$$= \frac{1}{2}(\boldsymbol{\vartheta}^T \boldsymbol{R}^T \boldsymbol{GR\vartheta} - 2\boldsymbol{d}^T \boldsymbol{GR\vartheta} + \boldsymbol{d}^T \boldsymbol{Gd}) \qquad (19)$$

To minimize the forecasting error, set $\frac{\partial \boldsymbol{E}}{\partial \boldsymbol{\vartheta}} = 0$, then

$$\boldsymbol{\vartheta} = (\boldsymbol{R}^T\boldsymbol{G}\boldsymbol{R})^{-1}\boldsymbol{R}^T\boldsymbol{G}\boldsymbol{d}\,. \tag{20}$$

To derive the formula for a recursive estimate, corresponding to the kth input data pair, $\boldsymbol{\vartheta}_k$ can be computed by Eq. (20), that is,

$$\boldsymbol{\vartheta}_k = (\boldsymbol{R}_k^T\boldsymbol{G}_k\boldsymbol{R}_k)^{-1}\boldsymbol{R}_k^T\boldsymbol{G}_k\boldsymbol{d}_k\,. \tag{21}$$

Corresponding to the (k+1)th data pair $\{\boldsymbol{a}_{k+1}^T,\ d^{(k+1)}\}$, Eq. (21) becomes

$$\boldsymbol{\vartheta}_{k+1} = \left(\begin{bmatrix}\boldsymbol{R}_k\\ \boldsymbol{r}_{k+1}^T\end{bmatrix}^T\begin{bmatrix}\boldsymbol{G}_k & 0\\ 0 & 1\end{bmatrix}\begin{bmatrix}\boldsymbol{R}_k\\ \boldsymbol{r}_{k+1}^T\end{bmatrix}\right)^{-1}\begin{bmatrix}\boldsymbol{R}_k\\ \boldsymbol{r}_{k+1}^T\end{bmatrix}^T\begin{bmatrix}\boldsymbol{G}_k & 0\\ 0 & 1\end{bmatrix}\begin{bmatrix}\boldsymbol{d}_k\\ d^{(k+1)}\end{bmatrix}$$

$$= (\boldsymbol{R}_k^T\boldsymbol{G}_k\boldsymbol{R}_k + \boldsymbol{r}_{k+1}\boldsymbol{a}_{k+1}^T)^{-1}(\boldsymbol{R}_k^T\boldsymbol{G}_k\boldsymbol{d}_k + \boldsymbol{r}_{k+1}d^{(k+1)}) \tag{22}$$

Let $\boldsymbol{S}_k = (\boldsymbol{R}_k^T\boldsymbol{G}_k\boldsymbol{R}_k)^{-1}$, that is, $\boldsymbol{S}_k^{-1} = \boldsymbol{R}_k^T\boldsymbol{G}_k\boldsymbol{R}_k$, then

$$\begin{aligned}\boldsymbol{\vartheta}_{k+1} &= \boldsymbol{S}_{k+1}\boldsymbol{R}_k^T\boldsymbol{G}_k\boldsymbol{d}_k + \boldsymbol{S}_{k+1}\boldsymbol{r}_{k+1}d^{(k+1)}\\ &= \boldsymbol{\theta}_k + \boldsymbol{S}_{k+1}\boldsymbol{r}_{k+1}(d^{(k+1)} - \boldsymbol{r}_{k+1}^T\boldsymbol{\theta}_k)\,,\end{aligned} \tag{23}$$

$$\begin{aligned}\boldsymbol{S}_{k+1} &= \left(\begin{bmatrix}\boldsymbol{R}_k\\ \boldsymbol{r}_{k+1}^T\end{bmatrix}^T\begin{bmatrix}\boldsymbol{G}_k & 0\\ 0 & 1\end{bmatrix}\begin{bmatrix}\boldsymbol{R}_k\\ \boldsymbol{r}_{k+1}^T\end{bmatrix}\right)^{-1}\\ &= \boldsymbol{S}_k - \boldsymbol{S}_k\boldsymbol{r}_{k+1}(\boldsymbol{I} + \boldsymbol{r}_{k+1}^T\boldsymbol{S}_k\boldsymbol{r}_{k+1})^{-1}\boldsymbol{r}_{k+1}^T\boldsymbol{S}_k\,.\end{aligned} \tag{24}$$

where $k = 0, 1, \ldots, K\text{-}1$. The least squares estimate of $\boldsymbol{\vartheta}$ is $\boldsymbol{\vartheta}_K$. The initial conditions to Eq. (23) and (24) are $\boldsymbol{\vartheta}_1 = 0$ and $\boldsymbol{S}_1 = \alpha\boldsymbol{I}$, respectively, where $\alpha \in [10^2\ 10^6]$, and I is an identity matrix.

IV. PERFORMANCE EVALUATION

In this section, firstly, the performance of the proposed ANF predictor is evaluated using a benchmark data model. Next, it is implemented for gear health condition forecasting. The performance of the ANF predictor is compared with a commonly used ANFIS predictor, which has a similar architecture as shown in Fig. 1, but without feedback links in the second layer. In addition, the gradient algorithm and least squares estimate (LSE) are applied for ANFIS training.

A. Sunspot activity forecasting

The sunspot activity series is one of the commonly used benchmark data sets in forecasting research, which has natures of nonlinear, non-Gaussian, and non-stationary. The available annual sunspot activity record is from years 1700 to 2017 [8]. The first 230 samples (years 1700-1929) are used for training, whereas the remaining data pairs are for testing. Fig. 3 denonstrates the forecasting results. It is seen from Fig. 3(a) that the ANFIS cannot capture the system dynamic behavior (E_T = 176.51) even for a one-step-ahead prediction, mainly due to its poor training and lack of adaptive capability. Generally, to properly train the ANFIS predictor requires more than five times the number of parameters (96 in this case) to be updated [9]. Fig. 3(b) shows the three-step-ahead forecasting result using the developed ANF predictor (E_T = 10.76). It is clear that the ANF predictor can capture and track the characteristics of this sunspot activity series, due to its more flexible reasoning architecture and adaptive training.

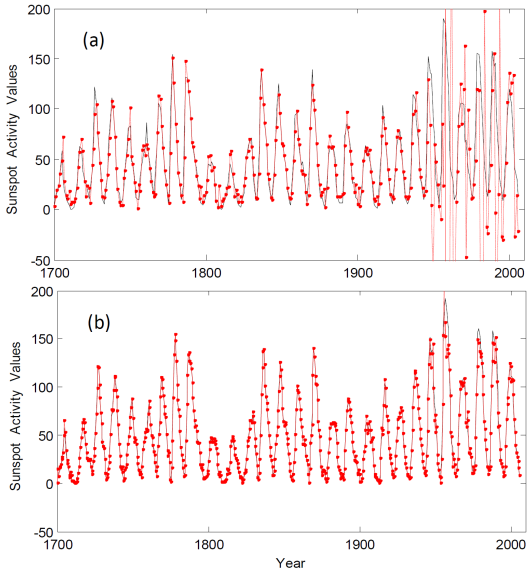


Fig. 3. Forecasting results of a sunspot activity record (solid curves): (a) by using the ANFIS for one-step-ahead prediction (dotted curve); (b) by using the ANF for three-steps-ahead prediction (dotted curve).

B. Machinery condition monitoring applications

Gearboxes are commonly used in rotating machines. A gear fault can not only deteriorate transmission accuracy but may also cause sudden failures in applications such as airplanes, power turbines, and chemical engineering facilities. A few examples related to gear condition monitoring are employed in this work to demonstrate how to apply the developed ANF predictor for real-time multi-step machinery forecasting applications. Fig. 4 schematically shows the experimental setup used for this online gear system monitoring tests. The system is driven by a 3-HP AC motor. The load is provided by a heavy-duty magnetic break system. A two-stage gearbox is being tested in this work. Gears #1 to #3 are identical but with different health conditions. The shafts are mounted to the gearbox housing by rolling element bearings. A magnetic pick-up sensor is mounted in the radial direction of gear #6 to provide a reference signal (one pulse per each tooth period) for the time synchronous average filtering. The gap between the magnetic pick-up sensor and the gear top land is in the range of 0.5 to 1.5 mm. The motor rotation is controlled by a speed controller, which allows tested

gear system to operate in the range of 200 to 4200 rpm. The speed of the drive motor and the load of the magnetic break system are adjusted automatically to simulate different speed/load operating conditions. The vibration signals are measured by using two accelerometers mounted at both ends of the gearbox housing. The signals from different sources are collected using an intelligent data acquisition device made by the authors' research team, which consists of micro-controllers and the circuits for the functionalities such as sampling, amplifiers, anti-aliasing filters, and power source for ICP accelerometers. The preconditioned signals are then fed into a computer for further processing.

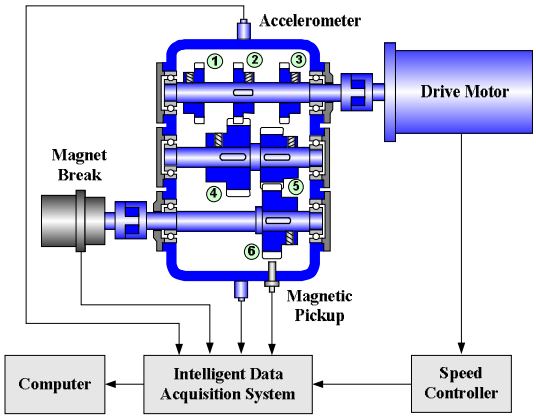


Fig. 4. Experimental setup for gear system monitoring.

In this work, the fault detection of the gear system is conducted gear by gear. The measured vibration from the experimental setup is an overall signal associated with different vibratory sources such as shafts, bearings and gear mesh. Because gear signal is periodic in nature, it can be differentiated by applying a time synchronous filtering process [9], based on the reference signature provided by the magnetic pickup sensor. In this case, the rotation reference for each gear is computed by the corresponding transmission ratio with respect to gear #6. As a result, the signals nonsynchronous to the rotation of the gear of interest can be removed, and each gear signal is derived and represented in one full revolution (i.e., the signal average). In machinery condition monitoring, a monitoring index should be sensitive to pattern modulation due to machinery faults but insensitive to noise [10]. Several signal processing techniques have been proposed in the authors' research group for gear fault diagnosis. In this case, a beta kurtosis monitoring index is adopted as an example for gear fault diagnosis/prognosis [9]. The beta kurtosis is the normalized fourth moment of a signal, in terms of a beta function instead of a generally used Gaussian function. Suppose m_z and σ_z represent, respectively, the mean and standard deviation of the data over the zth tooth period of a gear signal, z = 1, 2, . . ., Z, Z is the number of teeth of the gear of interest. The beta kurtosis reference function is defined as

$$f_B(z) = \frac{\alpha\beta(\alpha+\beta+2)(\alpha+\beta+3)}{3(\alpha+\beta+1)(2\alpha^2-2\alpha\beta+\alpha^2\beta+\alpha\beta^2+2\beta^2)} \quad (25)$$

where $\alpha = \frac{m_z}{\sigma_z^2}(m_z - m_z^2 - \sigma_z^2)$,

$$\beta = \frac{1-m_z}{\sigma_z^2}(m_z - m_z^2 - \sigma_z^2)$$

Two gear cases are tested in this study as represented in Fig. 5: (a) healthy gears (gear #2 in Fig. 4) and (b) cracked gears (gear #1). Many tests have been conducted corresponding to each gear condition, and two typical examples are illustrated in the following sections.

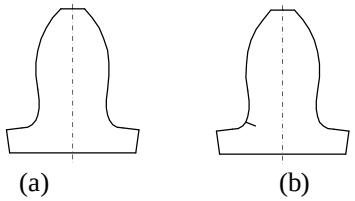
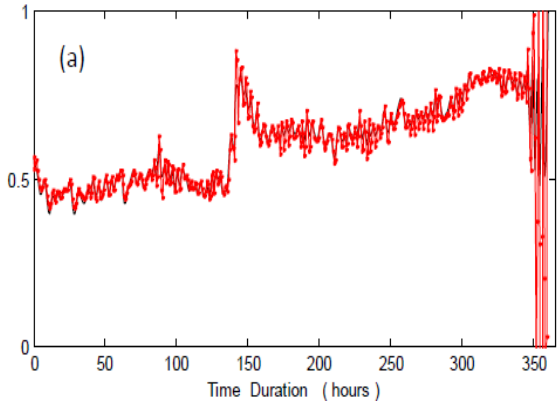


Fig. 5. Gear conditions tested: (a) healthy gear, (b) cracked gear.

C. Cracked gear monitoring

At first, all the gears in the gearbox are in healthy condition. The tests are conducted under load levels from 0.5 to 3 hp and motor speeds from 200 to 3600 rpm. During testing, motor speed and load levels are randomly changed to simulate time-varying operating conditions. Before application, the predictors are initially trained by a data set recorded from a previous test. The monitoring time-step is set at s = 30 minutes, that is, both predictors are applied automatically in every 30 minutes to forecast the upcoming values of the beta kurtosis index after three steps, or 1.5 hours ahead. After about 140 hours, a cracked gear #1 with a transverse cut with 20% of the tooth root thickness is moved to mesh with the mating gear. Then the test is resumed and continues until the damaged tooth is broken off about 220 hours later. Fig. 6 shows the forecasting results of the monitoring index for the tested gears.



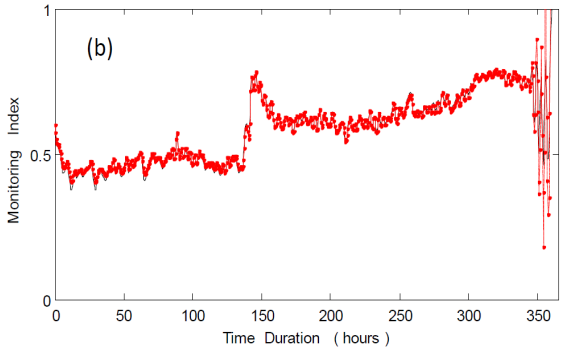


Fig. 6 Three-steps-ahead forecasting results for the cracked gear (solid curves): (a) by using the ANF (dotted curve); (b) by using ANFIS (dotted curve).

It is seen that both predictors can capture system's dynamic behaviors, but the ANF performs better. After the fault is introduced, the gear mesh dynamics change correspondingly. The ANFIS (Fig. 6b) takes about 8 steps to re-capture the system's dynamic characteristics, whereas the ANF predictor (Fig. 6a) takes only a couple of steps. During the last monitoring section, big fluctuations appear because the gear mesh dynamics change dramatically just before and after the tooth breakage. The ANF performs better to track these dramatic changes in system characteristics due to superior convergence.

CONCLUSION

An ANF (adaptive neuro-fuzzy) predictor has been developed in this paper for multi-step-ahead prediction of machinery systems. The forecasting reasoning is based on a weighted recurrent NF paradigm, while the ANF parameters are adaptively trained by a recursive hybrid algorithm. From the comparison study, it has shown that the proposed ANF predictor has an efficient forecasting accuracy. The ANF predictor is also implemented for gear system monitoring corresponding to healthy and localized fault conditions. Online test results have shown that the developed ANF predictor is a reliable forecasting tool. It can capture the system's dynamic behavior quickly and track the system's characteristics accurately. It is also a robust forecasting tool in terms of its capabilities to accommodate different system conditions.

REFERENCES

[1] J. Korbicz, "Fault Diagnosis: Models, Artificial Intelligence, Applications", Springer, 2004.
[2] D. Chelidze and J. Cusumano, "A dynamical systems approach to failure prognosis," Journal of Vibration and Acioustics, Vol. 126, pp. 1-7, 2004.
[3] D. Li, W. Wang, and F. Ismail, "A regulated boosting technique for material fatigue property prognostics", Engineering Applications of Artificial Intelligence, vol. 37, no. 1, pp. 153-159, 2017.
[4] D. Li, W. Wang and F. Ismail. A mutated particle filter technique for system state estimation and battery life prediction. IEEE Transactions on Instrumentation and Measurement, vol., 63, pp. 2034-2043, 2014.
[5] Y. Liang and X. Liang, "Improving signal prediction performance of neural networks through multiresoltion learning approach," IEEE Trans. Syst. Man Cyber. - Part B, vol. 36, pp. 341-352, 2006.
[6] Y. Xu, X. Li, D. Chen, H. Li, "Learning rates of regularized regression with multiple Gaussian kernels for multi-task learning", vol. 29, no. 11, pp. 5408 - 5418, 2018.
[7] H. Ishibuchi and T. Yamamoto, "Rule weight specification in fuzzy rule-based classification systems," IEEE Trans. Fuzzy Syst. vol. 13, pp. 428-435, 2005.
[8] SIDC, RWC Belgium World Data Center Online Sunspot Data Archive (http://sidc.oma.be/index .php3), 2018.
[9] W. Wang, F. Ismail F, and F. Golnaraghi. "A neuro-fuzzy approach for gear system monitoring," IEEE Trans. Fuzzy Syst. vol. 12, pp. 710-723, 2004.
[10] A. Mariello and R. Battiti, "Feature selection based on the neighborhood entropy," vol. 29, no. 12, pp: 6313-6322, 2018.

★★★