J Intell Manuf (2018) 29:753–761
https://doi.org/10.1007/s10845-015-1127-4

Residual life prediction based on dynamic weighted Markov model and particle filtering

Shuai Zhang[1] · Yongxiang Zhang[1] · Jieping Zhu[1]

Received: 16 November 2014 / Accepted: 7 July 2015 / Published online: 17 July 2015

Abstract In order to improve the prediction accuracy of non-Gaussian data and build reasonably the prediction model, a novel residual life prediction method is proposed. A dynamic weighted Markov model is constructed by real time data and historical data, and the residual life is predicted by particle filter. The particles of the state vector are predicted and updated instantaneously using particle filter. The probability distribution of the predicted value is estimated by the updated particles. The residual life can be predicted using the set threshold of the state. This method improves the accuracy of residual life prediction. Finally, the advantage of this proposed method was shown experimentally using the bearings' full cycle data.

Keywords Dynamic weighted Markov model · Particle filtering · Residual life prediction · Probability distribution

Introduction

Rolling element bearings are the most important components used in machinery. Accurate residual life prediction can prevent rolling element bearings from critical damage in order to guarantee the personal safety. Therefore, it deserves much research attention in improving the residual life prediction accuracy for rolling element bearings.

The residual life prediction for rolling element bearings is to value the current or future state and obtain the residual life. Generally, one or more fault features are chosen as the health indicator of machinery. The prognostic model is used to predict the bearings' future state so that the residual life can be obtained. Effective health indicator and robust model can boost the prediction accuracy. However, the full life data of rolling element bearings is non-stationary and non-Gaussian. Hence, the main problem of residual life prediction is how to build reasonably the prediction model and improve the prediction accuracy using non-Gaussian data.

To address this problem, lots of methods have been proposed. Chen et al. (2012), Tian (2012), Chao and Hwang (1997) proposed a new prediction method based on neural network which improve the long-term prediction accuracy slightly. However, the neural network has some limitations of: (1) difficulty of determining the network structure and the number of nodes; (2) slow convergence of the training process. Gebraeel et al. (2005), Kharoufeh and Cox (2005) proposed some prognostic approach based on Bayesian theory, which can predict the probability distribution of the residual life. Meanwhile, Kharoufeh and Mixon (2009) also proposed a new method that can obtain the residual life by computing Phase-Type (PH) distribution. But the shortcoming of this method is also very obvious. It needs a large number of accurate data of prior probability distribution. Unfortunately, it is difficult to meet in actual application. Çaydaş and Ekici (2012), Kimotho et al. (2013), Yaqub et al. (2013), Chen et al. (2013) used support vector machine to predict the residual life of rolling element bearings and obtained high accuracy of short-term prediction. However, the long-term prediction accuracy is low. Wang et al. (2013), Ng et al. (2014), Tobon-Mejia et al. (2012) used Bayesian model for residual life prediction, which made high accuracy valuation for Gaussian data. The above mentioned methods are not good at dealing with non-stationary data and non-Gaussian data. Markov model can deal with these kinds of data very well and exhibit great effectiveness in comparison to the above mentioned methods. Liu et al. (2012)

✉ Yongxiang Zhang
zyx6302@aliyun.com

[1] Naval University of Engineering Power Engineering Marine Engineering, WuHan 430033, China

presented a prediction methodology based on hidden semi-Markov model and sequential Monte Carlo methods, which enhanced the prediction accuracy of non-stationary. Yan et al. (2011) proposed a dynamic multi-scale Markov model based methodology for residual life prediction, which used historical data based model and real-time data based model to obtain the residual life and enhanced long-term prediction accuracy. Particle filtering does well in dealing with non-Gaussian problems. Particle filtering uses non-parameter Monte Carlo method to achieve recurrence Bayesian filtering with great estimation accuracy and easy implementation. What is more, particle filtering is easier implementation. Orchard (2007) used the Paris' law for crack propagation model based on particle filtering. Baraldi et al. (2013) also used particle filtering to predict residual life and obtained good results. However, the previous prediction models were mostly built by historical data, which seldom compromised the information of the real-time data. But the real time data can directly reflect the current development trend of the bearings' residual life while the historical data contains the empirical information of the bearings' residual life. To this point, Using both historical data and real time data can exhibit some advantages theoretically in terms of efficiency and accuracy.

Motivated by the aforementioned problems, a novel residual life prediction method based on dynamic weighted Markov model and particle filtering is proposed. In this method, the real-time data and historical data are used to build dynamic weighted Markov models. The weights of the Markov models are adjusted according to the real-time predicted error. The distribution of predicted value can be obtained through generating the particles by particle filtering with great estimation accuracy under non-Gaussian data. The experimental results show that the proposed method is effective in residual life prediction.

The review of the Markov model and particle filtering

The Markov model

Assuming $\{X_n, n = 1, 2, y\}$ is a random process. $X_n = i$ for state i at moment n and P_{ij} is the probability that the state i transmit to state j. The formula of transition probability can be written by:

$$\begin{aligned} &P\{X_{n+1} = j|X_n = i, X_{n-1} = i_{n-1}, \ldots, X_2 = i_2, X_1 = i_1\} \\ &\quad = P\{X_{n+1} = j|X_n = i\} = P_{ij} \quad (i, j, n > 0) \end{aligned} \tag{1}$$

Markov chain is a sequential random process that satisfy Markov property; The current state X_n is dependent only on the state of one step prior X_{n-1} and the past and the future is irrelevant. This paper selects discrete Markov model, because the experimental data used in this paper is time discreteness and states discreteness.

Particle filtering

In general, particle filtering is a Monte Carlo method that can achieve the recursive Bayesian filtering. Thus we'll introduce the particle filtering based on a quick overview of Bayesian filtering. The state space of dynamic system can be described as follow:

$$x_k = f(x_{k-1}) + u_{k-1} \tag{2}$$

$$y_k = h(x_k) + v_k \tag{3}$$

where $f(\cdot)$ and $h(\cdot)$ are respectively state transition equation and observation equation; x_k is the system's state at time k; y_k is observation at time k; u_k is process noise at time k; v_k is observation noise at time k. Bayesian filtering includes prediction and update. Prediction is to obtain the prior probability of the next moment, and the update is to correct the prior probability using the latest measurement so that the posterior probability can be obtained. Assume that $p(x_{k-1}|y_{k-1})$ is the probability at time $k-1$, and $\overrightarrow{X_k} = x_{0:k}$ and $\overrightarrow{Y_k} = y_{0:k}$ respectively express the states and observations from time 0 to time k. Thus the prediction can be given by:

$$p\left(x_k, x_{k-1}|\,\overrightarrow{Y_{k-1}}\right) = p\left(x_k|x_{k-1}, \overrightarrow{Y_{k-1}}\right) p\left(x_{k-1}|\,\overrightarrow{Y_{k-1}}\right) \tag{4}$$

Since x_k and $\overrightarrow{Y_{k-1}}$ are independent of each other when x_{k-1} is given, we can obtain the formula as follow:

$$p\left(x_k, x_{k-1}|\,\overrightarrow{Y_{k-1}}\right) = p\left(x_k|x_{k-1}\right) p\left(x_{k-1}|\,\overrightarrow{Y_{k-1}}\right) \tag{5}$$

Then we integrate x_{k-1} at the both sides of the above formula, and the Chapman–Kolmogorov equation can be obtained as follow:

$$p\left(x_k|\,\overrightarrow{Y_{k-1}}\right) = \int p(x_k|x_{k-1}) p\left(x_{k-1}|\,\overrightarrow{Y_{k-1}}\right) dx \tag{6}$$

When the prediction is done, the update is launched to get $p(x_k|\,\overrightarrow{Y_k})$ based on $p(x_k|\,\overrightarrow{Y_{k-1}})$ as follow:

$$p\left(x_k|\,\overrightarrow{Y_k}\right) = \frac{p\left(y_k|x_k, \overrightarrow{Y_{k-1}}\right) p\left(x_k|\,\overrightarrow{Y_{k-1}}\right)}{p\left(y_k|\,\overrightarrow{Y_{k-1}}\right)} \tag{7}$$

Assume that y_k is determined by x_k, which is given by:

$$p\left(y_k|x_k,\overrightarrow{Y_{k-1}}\right)=p\left(y_k|x_k\right) \tag{8}$$

Then after normalization, we can obtain:

$$p\left(y_k|\overrightarrow{Y_{k-1}}\right)=\int p(y_k|x_k)p\left(x_k|\overrightarrow{Y_{k-1}}\right)dx \tag{9}$$

Then particle filtering can be used to solve the above integration formula. The key idea of particle filtering is to represent the posterior probability by taking weighted average of random samples. Assume that N random samples $x_k^{(i)}$ can be extracted from $p(x_k|\overrightarrow{Y_k})$, then:

$$p\left(x_k|\overrightarrow{Y_k}\right)\approx\omega_k^i\sum_{i=1}^{N}\delta\left(x_k-x_k^{(i)}\right) \tag{10}$$

where $\delta(x_k-x_k^{(i)})$ is unit impulse function and ω_k^i is the corresponding weight. These particles can be generated from the importance density function $q(x_k|\overrightarrow{Y_k})$. Thus

$$\omega_k^i\propto\frac{p\left(x_k^{(i)}|\overrightarrow{Y_k}\right)}{q\left(x_k^{(i)}|\overrightarrow{Y_k}\right)} \tag{11}$$

Assume that the importance density function $q(x_{0:k}|y_{1:k})$ can be decomposed as follow:

$$q\left(x_{0:k}|y_{1:k}\right)=q\left(x_k|x_{0:k-1},y_{1:k}\right)q\left(x_{0:k-1}|y_{1:k-1}\right) \tag{12}$$

By plugging formula (7) and (12) in formula (11), new formula is obtained as follow:

$$\omega_k^i\propto\omega_{k-1}^i\frac{p\left(y_k|x_k^{(i)}\right)p\left(x_k^{(i)}|x_{k-1}^{(i)}\right)}{q\left(x_k^{(i)}|x_{0:k-1}^{(i)},\overrightarrow{Y_k}\right)} \tag{13}$$

Generally, we choose the state transition function $p(x_k|x_{k-1})$ as the typical choice of importance density in PF. Then

$$\omega_k^i=\omega_{k-1}^i p\left(y_k|x_k^{(i)}\right) \tag{14}$$

Residual life prediction based on dynamic weighted Markov model and particle filtering

The principle of Residual life prediction based on dynamic weighted Markov model and particle filtering is to use K-means clustering algorithm to divide the kurtosis of the historical data, and two Markov models are respectively built by historical data and real-time data. By combining two Markov models with particle filtering, two predicted results can be obtained. The ultimate prediction residual life is obtained by take weighted average of the two predicted results.

State division

The most important step of residual life prediction is state division. K-means clustering algorithm Hartigan and Wong (1979) is used to divide kurtosis of the full life cycle. K-means clustering algorithm can take the most similar data to the same subset so that the different subsets have the least similar. The principle is as follow:

Divide n vectors $x_j(1,2\ldots,n)$ into c clusters and determine the clustering center c_i of the full clusters. The objective function is as follow:

$$J=\sum_{i=1}^{c}Ji=\sum_{i=1}^{c}\left(\sum_{k,x_k\in Gi}||x_k-c_i||^2\right) \tag{15}$$

According to the clustering center, the new division is implemented so that new clusters are obtained. Then, the clustering iteration is implemented and does not stop until the objective function value is less than the set threshold.

The prediction principle of the Markov model

The data in every state of the full life cycle is non-stationary and Markov model can deal with non-station data very well. Using Markov model can achieve forecast by state vectors and state transition matrix. The full states transition probability can be expressed by a transition matrix P, which is shown as follow:

$$P=\begin{bmatrix}P_{11} & P_{12} & \ldots\ldots & P_{1s}\\ P_{21} & P_{22} & \ldots\ldots & P_{2s}\\ \vdots & \vdots & \ldots\ldots & \vdots\\ P_{s1} & P_{s2} & \ldots\ldots & P_{ss}\end{bmatrix}$$

The formula of the transition probability is given by:

$$P_{ij}=\frac{N_{ij}}{N_i} \tag{16}$$

where N_{ij} denotes the transition times from i to j and N_i denotes the number of NLLP indicator values belonging to i.

Assume that the state vector at moment $t-1$ is $\overrightarrow{P_{t-1}}$ and the state vector at moment t is $\overrightarrow{P_t}$. The formula of state transition is shown as follow:

$$\overrightarrow{P_t}=P\cdot\overrightarrow{P_{t-1}} \tag{17}$$

when time t from 1 to K, the formula (17) transforms as follow:

$$\overrightarrow{P_k} = P^k \cdot \overrightarrow{P_0} \tag{18}$$

where $\overrightarrow{P_k}$ is the state vector at moment K. The state vector expresses the probability that each state may occur. Once the initial state vector and state transition matrix are given, the state vector at any time can be predicted. In order to fully express the current state of the machine through the state vector, the weighted average method proposed by Liu et al. (2012) is used to obtain the state value. Assume that state vector at moment t is $\overrightarrow{P_t}$, where $\overrightarrow{P_t} = (p_1, p_2, \ldots, p_n)$, n is the state number. Then, the state value C_t at this moment is calculated as follow:

$$C_t = 1 \times p_1 + 2 \times p_2 + \cdots n \times p_n \tag{19}$$

The state value expresses the current state. If state value has decimal part, it can tell that the machine is in the transition state of the adjacent states. Once the state value reaches the threshold set according to the historical data, the corresponding moment is defined as the fail-point. Then, the residual life can be obtained according to the current moment and the predicted fail-point.

The prediction principle based on particle filtering

Particle filtering with great estimation of non-Gaussian data generates plenty of particles to obtain the distribution of the predicted value that the predicted results can be more effectively expressed in comparison to the single predicted value. The prediction principle based on particle filtering and Markov model contains prediction and update.

The prediction principle is to used Markov model and particles to implement the prediction. Firstly, the state vector's particles with the corresponding weights are generated. Secondly, the predicted results corresponding to every particle is obtained by Markov model. At last, the final predicted result is obtained by taking weighted average of the predicted results corresponding to every particle. Assume that the state vector's particles is expressed as $\overrightarrow{\phi_t^w} = (x_1, x_2, \ldots, x_n)$, where n is state number, t is the moment, w is particle number; ω_w is the weight of the particle; the predicted results corresponding to every particle is expressed as l_w; the total number of the particles is m. The ultimate result L is as follow:

$$L = \sum_{w=1}^{m} \omega_w \cdot l_w \tag{20}$$

The update principle is to update the particles and the weights by obtaining the new observation so that the predicted residential life can be updated. The real-time state vector is obtained according to the updated observation, which can be combined with formula (14) to update the weights and resample so that the new particles and the corresponding weights at the current moment can be got. According to the prediction principle, a new predicted result can be obtained by using the new particles and weights. Therefore, the predicted result can be updated when a new observation is got.

Weight degeneracy is an inevitable problem in particle filtering, which can't be worked out by conventional resampling method. Therefore, this paper uses Metropolis-Hastings algorithm of Pitt et al. (2010) to solve this problem. By using Metropolis-Hastings algorithm, the concrete step that generates the particles from the probability density $q(x)$ is as follow:

(1) Generate random number $\nu \sim U[0, 1]$ from uniform distribution with the range [0,1].
(2) Generate particles $x_k^* \sim p(x_k^*|x_k^{(i)})$ from the importance density function.
(3) If $\nu \leq \min\left\{1, \frac{q(x_k^*)p(x_k^{(i)}|x_k^*)}{q(x_k^{(i)})p(x_k^*|x_k^{(i)})}\right\}$, x_k^* is adopted, otherwise, rejected.

Markov model with dynamic weights

The prediction model based on historical data can capture the overall trend information of the full historical life cycle, but the real time information cannot be compromised. The prediction model based on real time data can only obtain the high accuracy of short-term prediction. In order to improve the long-term prediction accuracy by taking full advantage of historical data and real time data, two Markov models is constructed by historical samples and real time data, respectively. The ultimate predicted result is calculated by taking a weighted average of the prediction results of the two prediction models. Because the prediction accuracy of the two models is different in different fault stages, the dynamic correction coefficient of the weights is set to adjust adaptively the coefficient in accordance with the relative prediction error of the two models. Therefore, the prediction accuracy can be improved according to the real condition. The formula of residual life is given by:

$$Rul = (\alpha_1 - \omega) \cdot l_1 + (\alpha_2 + \omega) \cdot l_2 \tag{21}$$

Here, ω is the dynamic correction coefficient; α_1 is the weight of the Markov model built by historical data and α_2 is the weight of the Markov model built by real-time data, $\alpha_1, \alpha_2 \in [0, 1]$, $\alpha_1 + \alpha_2 = 1$; l_1 stands for the residual life predicted by the historical data based Markov model and l_2 stands for the residual life predicted by the real-time data based Markov model.

α_1 and α_2 can be obtained by predicted error of the historical data. Firstly, the historical data is divided into two groups that one group contains one sample which stands for real-time data and another group stands for historical samples used for prediction. α_1 is valued between 0 and 1 at a step of 0.1 with the initial value 0.1. For each value, calculate the residual life errors between prediction and real values of the historical samples at all moments. Then, the α_1 value corresponding to minimal mean error is the ultimate weight coefficient.

If α_1 and α_2 are directly used for prediction, the real-time performance of prediction is very poor for the reason that α_1 and α_2 is obtained by historical data. Hence, the ω is set to adjust the weights according to the relative predicted error. Assume that $\overrightarrow{\eta_1}$ and $\overrightarrow{\eta_2}$ are the two models' state vectors predicted at moment t and $\overrightarrow{\eta_3}$ is the real state vector at moment t. Through Euclid distance calculation, the dynamic correction coefficient of the two Markov models can be got as follow:

$$\omega = 0.1 * \left(\frac{\left\| \overrightarrow{\eta_1} - \overrightarrow{\eta_3} \right\|}{\left\| \overrightarrow{\eta_2} - \overrightarrow{\eta_3} \right\|} - 1 \right) \tag{22}$$

Here, $\|\cdot\|$ stands for Euclid distance calculation.

In a word, the steps of residual life prediction based on dynamic weighted Markov model and particle filtering are as follow:

(1) Construct dynamic weights Markov model
(2) Calculate the initial state vectors according to the full real-time data.
(3) Predict the state vectors' particles at the next moment
(4) The predicted particles are compared with the set threshold. If the predicted value exceeds the threshold, the prediction should cease and calculate the residual life. If the predicted value doesn't exceed the threshold, the prediction goes on step (3).
(5) The predicted results of the two Markov models are calculated respectively according to formula (20). The ultimate result is obtained according to formula (21).
(6) Once the new observation is got, the update of particles and weights is implemented and the new particles and weights are put back to the model for new prediction.

The detailed process is shown in Fig. 1.

Experimental verification

The full life cycle data of rolling element bearings is from NASA website. The experimental bearing type is Rexnord ZA-2115. The experimental rotate speed is 2000 rm/min and sampling frequency is 20 KHz. Every sampling length of the data segment is 1s and the interval of the data segment is 10 min. Six groups of data are used in this paper. The formula of the average prediction error is given by:

$$e = \frac{1}{n} \sum_{r=1}^{n} \frac{|x_r - d_r|}{d_r} \times 100\,\% \tag{23}$$

Here, x_r stands for the predicted value at moment r, and d_r stands for the real value at moment r.

This paper uses kurtosis as fault indicator. The kurtosis has better sensitive to the fault and its formula is as follow:

$$kurtosis = \frac{\sum_{k=1}^{K} (x(k) - x_m)^4}{(k-1)x_{std}^4} \tag{24}$$

Here, $x(k)$ expresses signal sequence; k stands for the length of the signal sequence; x_{std} stands for standard deviation of $x(k)$; x_m stands for the mean value of the $x(k)$.

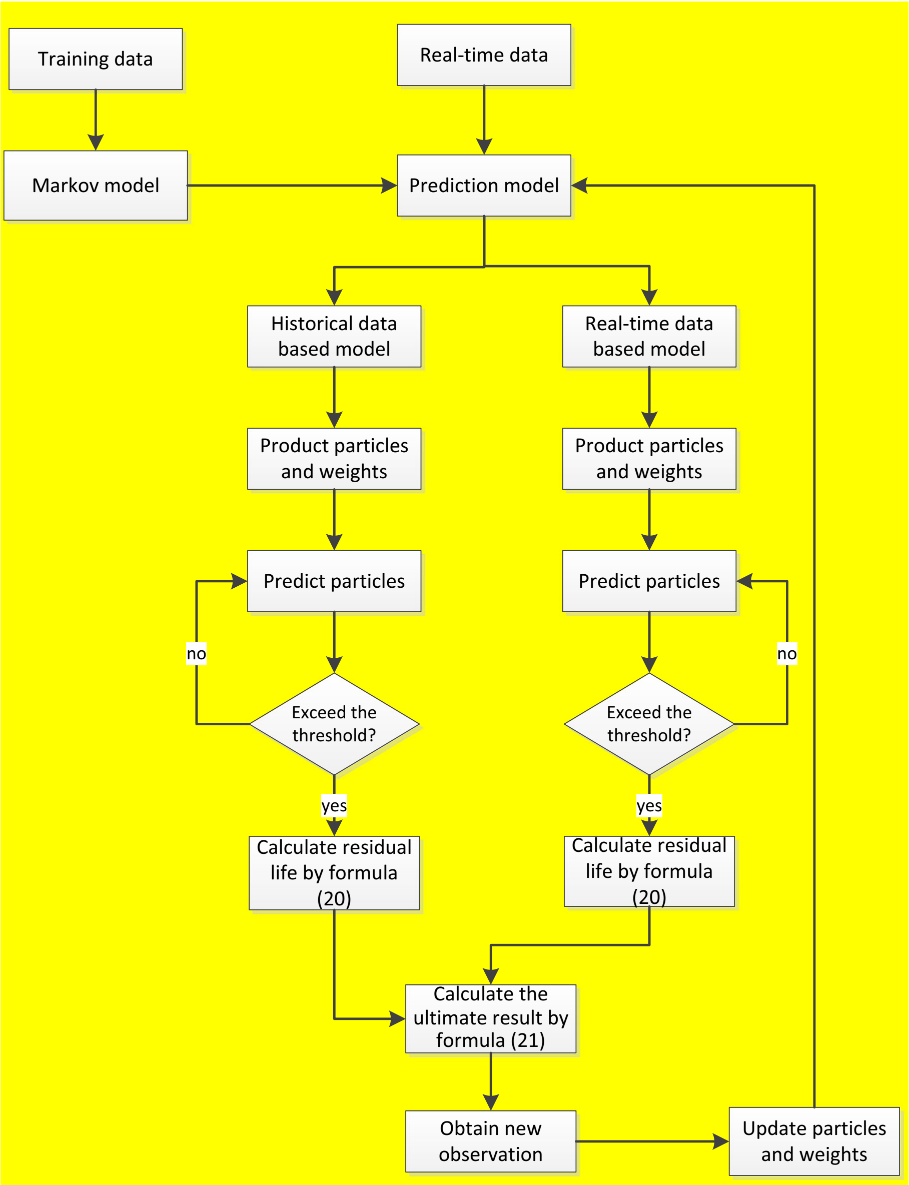
State division

K-means clustering algorithm is used to divide kurtosis of the signal. Figure 2 represents the kurtosis map of the fault trend, which contains six groups of data. As can be seen from Fig. 2, kurtosis becomes larger as the operation time grows longer. The kurtosis is below 20 when operation time doesn't exceed 3000 min; kurtosis is maintaining 50 $\sim$ 100 when operation time is between 3000 and 8000 min; the fault condition reaches fail-point when operation time surpasses 8000 min. According to the above information, the full life cycle data is divided into four states used for initial state of K-means clustering algorithm, where the states is: $(0, 20)$, $(20, 60)$, $(60, 100)$, $(100, \infty)$. Then, the division iteration for six groups of data is implemented. At last, the ultimate divided states is: $(0, 26)$, $(26, 72)$, $(72, 108)$, $(108, \infty)$.

The construction of the dynamic weight Markov model

According to calculation steps of chapter 2.2, α_1 can be calculated. Five samples are chosen to be divided into two groups that one group contains four samples while another one contains one sample. There are five array modes to divide five samples into two groups. Therefore, five weights can be obtained according to the five array modes, and the ultimate weight can be obtained by taking a weighted average of the five weights. Figure 3 is the trend map of prediction error and the weights. At last, the initial weight value is $\alpha_1 = 0.7$.

Fig. 1 The flow chart of residual life prediction


Training data
Real-time data
Markov model
Prediction model
Historical data based model
Real-time data based model
Product particles and weights
Product particles and weights
Predict particles
Predict particles
no
no
Exceed the threshold?
Exceed the threshold?
yes
yes
Calculate residual life by formula (20)
Calculate residual life by formula (20)
Calculate the ultimate result by formula (21)
Obtain new observation
Update particles and weights

Residual life prediction analysis

Five groups of data are used to train the Markov model and one group is used for prediction. The algorithm of reference Liu et al. (2012), on-line monitoring data based Markov model and historical data based Markov model are used for comparison. The total number of particles of reference Liu et al. (2012) is 30 and the initial distribution of the particles is Gaussian distribution. Figure 4 represents the prediction map of the proposed method and Fig. 5 represents the prediction map of the algorithm of reference Liu et al. (2012). Figures 6 and 7 are the prediction map of the Markov model based on on-line monitoring data and the prediction map of the Markov model based on historical data, respectively. The predicted error between every predicted value and the actual life is shown in Figs. 8, 9, 10 and 11, and the average prediction error of the four figures respectively are 0.101, 0.351, 0.372 and 0.394 according to the formula 23. Otherwise we use 3σ rule to compute the mean confidence interval of the four prediction results, which respectively are [11, 550], [123, 1450], [36, 933], and [154, 1157]. As shown from Figs. 4, 5, 6 and 7, the curve in Fig. 4 is closer to the real life curve in comparison to other figures. As can be seen from Figs. 8, 9, 10 and 11, the error of every predicted value in Fig. 8 is all below 600 min, and other figures' predicted errors are very large. We can also see that the predicted error of the proposed method is smaller than other methods from the average prediction error of the four methods. Moreover, the confidence interval of the proposed method is smaller than other methods'.

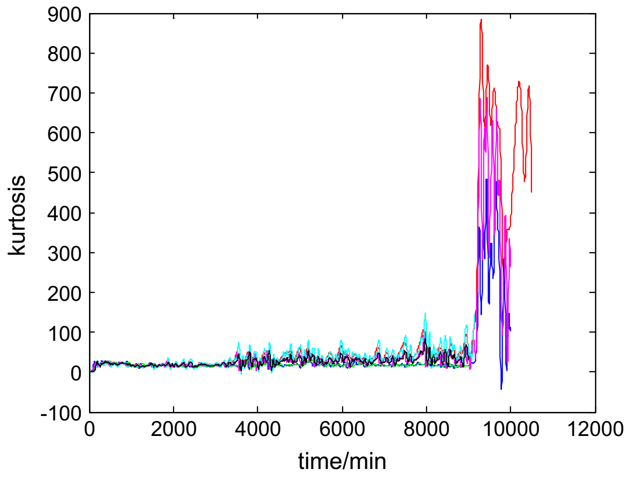


Fig. 2 The full-life kurtosis figure of the rolling bearing

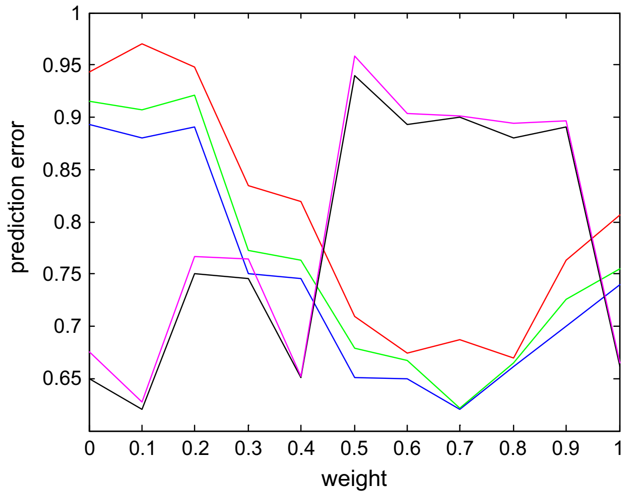


Fig. 3 The trend figure of prediction error and the weights

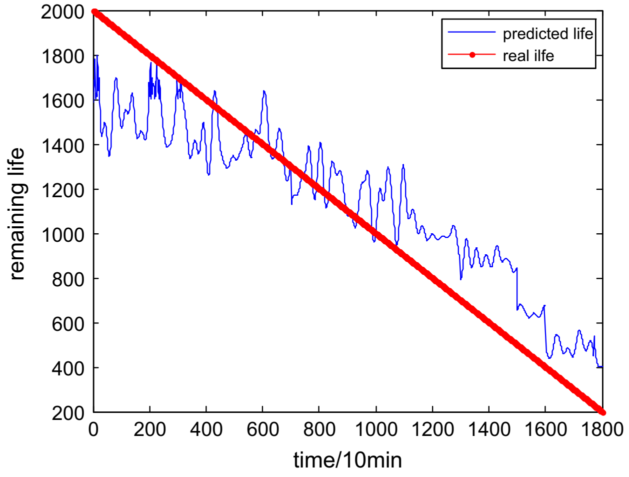


Fig. 4 The life forecast figure of method based on dynamic weighted Markov model and particle filtering

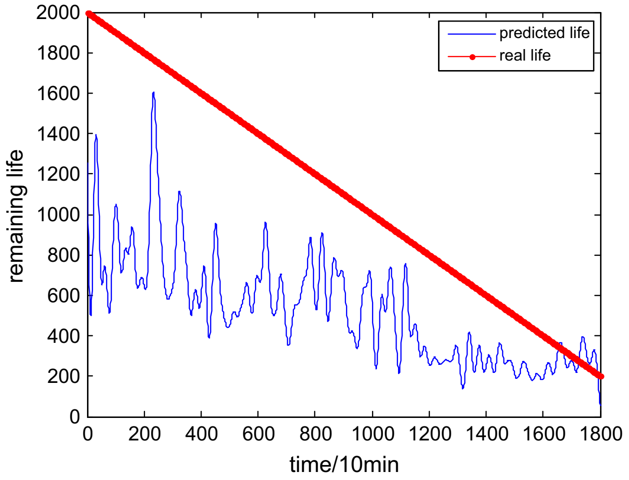


Fig. 5 The figure of life forecast based on reference Liu et al. (2012)

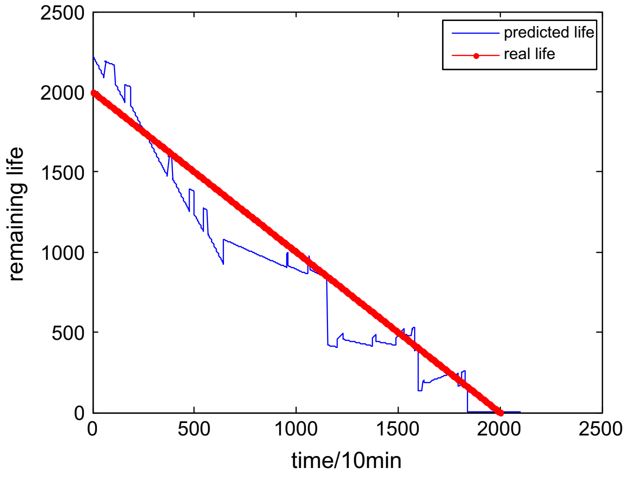


Fig. 6 The prediction map of the Markov model based on on-line monitoring data

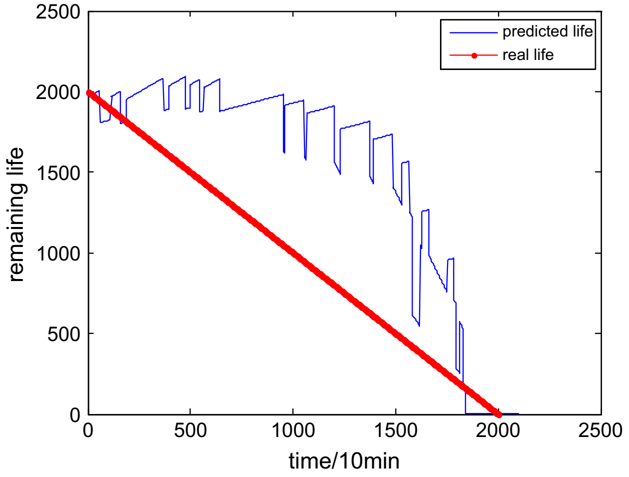


Fig. 7 The prediction map of the Markov model based on historical data

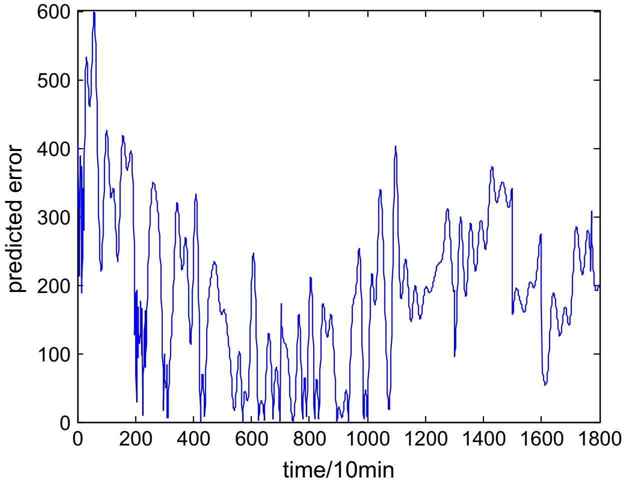


Fig. 8 The predicted error map of method based on dynamic weighted Markov model and particle filtering

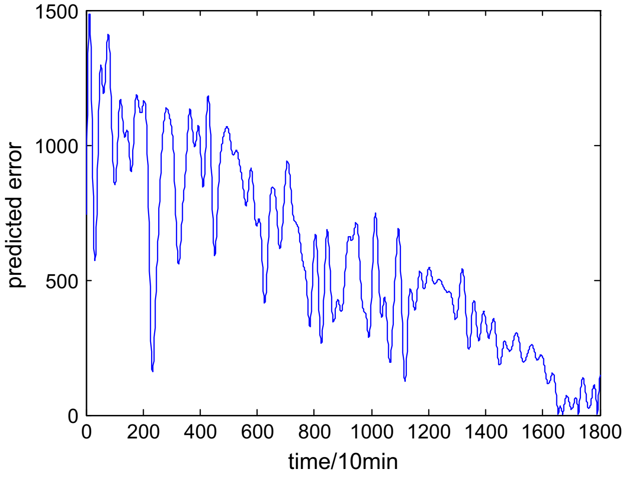


Fig. 9 The predicted error map of method based on reference Liu et al. (2012)

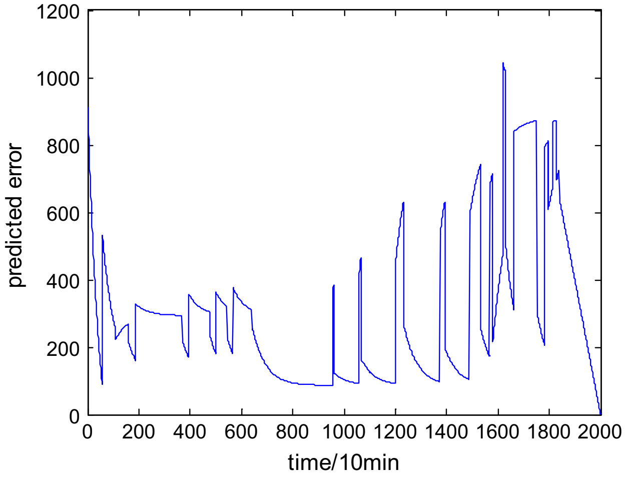


Fig. 10 The predicted error map of the Markov model based on on-line monitoring data

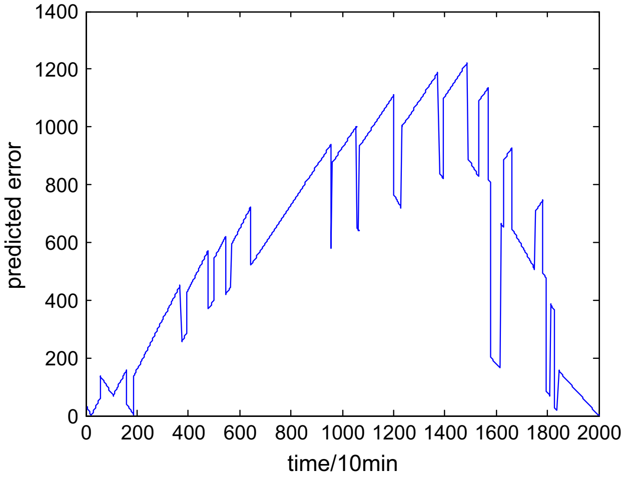


Fig. 11 The predicted error map of the Markov model based on historical data

Therefore, the proposed method has higher prediction accuracy than the other three methods according to the error of every predicted value and the average error.

Conclusion

This paper proposes a novel prediction method based on dynamic weights Markov model and particle filtering. The historical data and real-time data are used to construct dynamic weights Markov models, and then, residual life prediction can be evaluated by particle filter. Through verification of actual bearing data, the results show that the predicted error is small and the prediction accuracy can be improved by making full use of model based on both historical data and real-time data.

Acknowledgements The work described in this paper was supported by a Grant from the national defence researching fund (No. 9140A27020413JB11076).

References

Baraldi, P., Compare, M., Sauco, S., et al. (2013). Ensemble neural network-based particle filtering for prognostics. *Mechanical Systems and Signal Processing*, *41*(1), 288–300.

Çaydaş, U., & Ekici, S. (2012). Support vector machines models for surface roughness prediction in CNC turning of AISI 304 austenitic stainless steel. *Journal of Intelligent Manufacturing*, *23*(3), 639–650.

Chao, P. Y., & Hwang, Y. D. (1997). An improved neural network model for the prediction of cutting tool life. *Journal of Intelligent Manufacturing*, *8*(2), 107–115.

Chen, X., Shen, Z., He, Z., et al. (2013). Residual life prognostics of rolling bearing based on relative features and multivariable support vector machine. *Proceedings of the Institution of Mechanical*

Engineers, Part C: Journal of Mechanical Engineering Science, *227*(12), 2849–2860.

Chen, C., Vachtsevanos, G., & Orchard, M. E. (2012). Machine residual useful life prediction: An integrated adaptive neuro-fuzzy and high-order particle filtering approach. *Mechanical Systems and Signal Processing*, *28*, 597–607.

Gebraeel, N. Z., Lawley, M. A., Li, R., et al. (2005). Residual-life distributions from component degradation signals: A Bayesian approach. *IIE Transactions*, *37*(6), 543–557.

Hartigan, J. A., & Wong, M. A. (1979). Algorithm AS 136: A k-means clustering algorithm. *Applied Statistics*, 100–108.

Kharoufeh, J. P., & Cox, S. M. (2005). Stochastic models for degradation-based reliability. *IIE Transactions*, *37*(6), 533–542.

Kharoufeh, J. P., & Mixon, D. G. (2009). On a Markovc—Modulated shock and wear process. *Naval Research Logistics (NRL)*, *56*(6), 563–576.

Kimotho, J. K., Sondermann-Woelke, C., Meyer, T., et al. (2013). Machinery prognostic method based on multi-class support vector machines and hybrid differential evolution-particle swarm optimization. *Chemical Engineering Transactions*, *33*, 619–624.

Liu, Q., Dong, M., & Peng, Y. (2012). A novel method for online health prognosis of equipment based on hidden semi-Markov model using sequential Monte Carlo methods. *Mechanical Systems and Signal Processing*, *32*, 331–348.

Ng, S. S. Y., Xing, Y., & Tsui, K. L. (2014). A naive Bayes model for robust residual useful life prediction of lithium-ion battery. *Applied Energy*, *118*, 114–123.

Orchard, M. E. (2007). *A particle filtering-based framework for on-line fault diagnosis and failure prognosis*. Atlanta: Georgia Institute of Technology.

Pitt, M., Silva, R., Giordani, P., et al. (2010). Auxiliary particle filtering within adaptive Metropolis–Hastings sampling. arXiv preprint arXiv:1006.1914.

Tian, Z. (2012). An artificial neural network method for remaining useful life prediction of equipment subject to condition monitoring. *Journal of Intelligent Manufacturing*, *23*(2), 227–237.

Tobon-Mejia, D. A., Medjaher, K., & Zerhouni, N. (2012). CNC machine tool's wear diagnostic and prognostic by using dynamic Bayesian networks. *Mechanical Systems and Signal Processing*, *28*, 167–182.

Wang, Y., Deng, C., Wu, J., et al. (2013). Failure time prediction for mechanical device based on the degradation sequence. *Journal of Intelligent Manufacturing*, 13, 1–19.

Yan, J., Guo, C., & Wang, X. (2011). A dynamic multi-scale Markov model based methodology for residual life prediction. *Mechanical Systems and Signal Processing*, *25*(4), 1364–1376.

Yaqub, M. F., Gondal, I., & Kamruzzaman, J. (2013). Multi-step support vector regression and optimally parameterized wavelet packet transform for machine residual life prediction. *Journal of Vibration and Control*, *19*(7), 963–974.