A novel method based on least squares support vector regression combing with strong tracking particle filter for machinery condition prognosis

Chengliang Li, Zhongsheng Wang, Shuhui Bu, Hongkai Jiang and Zhenbao Liu
Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science 2014 228: 1048 originally published online 1 August 2013
DOI: 10.1177/0954406213494158

Original Article

Institution of MECHANICAL ENGINEERS

Journal of MECHANICAL ENGINEERING SCIENCE

Proc IMechE Part C:
J Mechanical Engineering Science
2014, Vol. 228(6) 1048–1062

DOI: 10.1177/0954406213494158
pic.sagepub.com
SAGE

A novel method based on least squares support vector regression combing with strong tracking particle filter for machinery condition prognosis

Chengliang Li, Zhongsheng Wang, Shuhui Bu, Hongkai Jiang and Zhenbao Liu

Abstract

A reliable prediction method is very important to avoid a catastrophic failure. This paper presents a novel method for machinery condition prognosis, named least squares support vector regression strong tracking particle filter which is based on least squares support vector regression combing with strong tracking particle filter. There are two main contributions in our work: first, the regression function of least squares support vector regression is extended, which constructs a bridge for the application of combining data-driven method with a recursive filter based on extend Kalman filter; second, an extend Kalman filter-based particle filter is studied by introducing a strong tracking filter into a particle filter. The strong tracking filter is used to update particles and produce importance densities which can improve the performance of the particle filter in tracking saltatory states, and finally strong tracking particle filter improves the prediction performance of least squares support vector regression in predicting saltatory states. In the experiment, it can be concluded that the proposed method is better than classical condition predictors in machinery condition prognosis.

Keywords

Least squares support vector regression, strong tracking particle filter, condition prognosis, saltatory states

Date received: 31 January 2013; accepted: 22 May 2013

Introduction

Remaining useful life (RUL) prediction is one of the key techniques in machinery prognostics and condition-based maintenance (CBM). In order to reduce the occurrence of catastrophic failures, it is necessary to predict the RUL of machinery equipment at early stages which allows people to take the required corrective actions in time. Hybrid prediction methods, with the ability of overcoming limitations of single approach, have become a research hotspot. According to recent literatures,[1–8] failure prognostics methods can be classified into two main categories: model-based and data-driven method for prognostics.

Model-based methods can provide accurate results for prediction only when we could get a proper mathematical model for a specific system. However, it is usually difficult to build accurate fault growth models in most real-world applications, especially when the process of fault propagation is complex or is not fully understood. Data-driven methods, on the other hand, employ the collected condition data to derive the fault propagation models. Least squares support vector regression (LS-SVR)[9] as a data-driven method has been successfully applied to long-term series prediction.[10] LS-SVR based on the structural risk minimization principle has stronger generalization capability than neural network, and also it has less computational complexity than support vector regression (SVR).

Recently, much attention has been paid to the research of prediction algorithms, in which, particle filter (PF) combined with model-based method or data-driven based method is a hot issue, because it shows excellent performance in dealing with the nonlinear and non-Gaussian problems.[11] It has been successfully applied in target tracking,[12] robot localization,[13] fault detection[14–17] and so on. The framework

School of Aeronautics, Northwestern Polytechnical University, Xi'an, China

Corresponding author:
Shuhui Bu, School of Aeronautics, Northwestern Polytechnical University, Xi'an 710072, China.
Email: bushuhui@nwpue.edu.cn

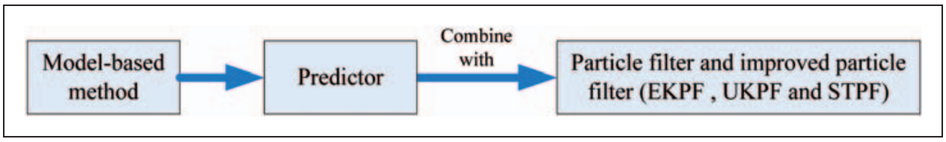

Figure 1. Model-based method combined with particle filter for prognosis.
EKPF: extended Kalman particle filtering; UKPF: unscented particle filtering; STPF: strong tracking particle filter.

for prognosis of model-based method combined with PF is shown in Figure 1. Some works have integrated data-driven method with PF method as a feasible prognostic framework. Zhang et al.[18] introduced a multi-fault modeling approach for fault diagnosis and failure prognosis, where recursive least square algorithm was employed to justify the parameters' values of bearing fault progression models online in the PF framework. Chen et al.[19] proposed an integrated failure prognostic algorithm, which used an adaptive neuro-fuzzy inference system (ANFIS) to model the fault degradation process and incorporated the ANFIS into a high-order PF to carry out multi-step ahead prediction. Saha et al.[20] presented a prognostic method using the Bayesian learning framework, which applied relevance vector machine (RVM) regression to collect parameters of cell failure mechanism model offline and fed the developed model into an online PF diagnosis and prognosis procedure. But these works do not consider an important issue: particle degeneracy in the PF method. Particle degeneration is an inherent defect of the particle filtering method, affecting the PF algorithm for the further development. Resampling technique can solve degradation problem in a certain extent, but it brings particularly prominent sample impoverishment problem in a long-period estimation. Selecting a proper importance density function is another common method to solve degradation problem. Freitas et al.[21] proposed extended Kalman particle filtering (EKPF) method in which PF integrates with extended Kalman filtering (EKF). Merwe et al.[22] proposed unscented particle filtering (UKPF) method in which PF integrates with unscented Kalman filtering (UKF). Rao-Blackwellised particle filter[23] is to divide the state space into deterministic and probabilistic parts. It is also analytically solved for the former while using PF for the latter. Wang and Xie[24] proposed the enhancing particle swarm optimization based particle filter (EPSOPF) to improve the performance of PF in impoverishment phenomenon. In a general situation, these works can solve the degradation issue as well, but there would be a great error when predicting the saltatory state system as shown in Figure 2(a). The time domain feature shows that most of the bearing fatigue time is consumed during the period of material accumulative damage, while the period of crack propagation is relatively short. Correct prognosis of the machine condition is based on tracking the machine condition accurately, and therefore a good predictor not only can predict the gradual state, but also can predict the saltatory state as Figure 2(b) presents. Thus, when the PF is applied to condition prognosis, the main problem is how to avoid particle degeneration and sample impoverishment to improve the ability when predicting the saltatory state. Some researchers focus on that the above important issue. Thrun et al.[25] proposed risk sensitive particle filters (RSPF) which generate particles according to a distribution that combines the posterior probability with a risk function. By incorporating a cost model into particle filtering, states that are more critical to the system performance and more likely to be tracked and the RSPF method were used to rover fault diagnosis,[26] and Orchard et al.[27] presented the proposed risk-sensitive PF (RSPF) framework and analyse the main advantages and disadvantages of its implementation, using actual failure data measuring battery capacity.

In this paper, a new method is proposed by introducing strong tracking filter (STF)[28] algorithm which is also EKF-based to deal with the abovementioned problems. It is sensitive to prediction error by adjusting the prediction variance, and the filter gain is sensitive to the change of the system state. Three kinds of good performance of STF algorithm can be concluded as follows: (1) Better performance in the tracking when system has saltatory state. Even though the system reaches a steady state, the tracking ability in the gradual state and salutatory state continues holding. (2) Better performance in robustness when model is uncertainty. (3) Modest computational complexity. Obviously, items (1) and (2) are benefit to overcome the EKF's defects. Item (3) will be helpful for real-time applications. In this study, the proposed method inherits the first merit of STF algorithm, which is used to improve the PF performance, and then the performance of the predictor (LS-SVR) can be improved when system has saltatory state. Suboptimal fading extended Kalman filtering (SFEKF)[28] is one of the STF algorithms. In the proposed method, we put forward a strong tracking particle filter (STPF) to update particles and to produce importance density. The STPF improves the tracking ability in saltatory state by alleviating particle degeneration and sample impoverishment. Moreover, we also solve another problem which has never been solved in previous works. That is, traditional framework of data-driven method cannot directly combine with the EKF-based PF (Example: STPF) which is illustrated in Figure 3. This is because the STPF is based on EKF, which linearizes about an estimate of the current mean and covariance with the Taylor expansion for first-order linear truncation (state transition matrix is a Jacobian

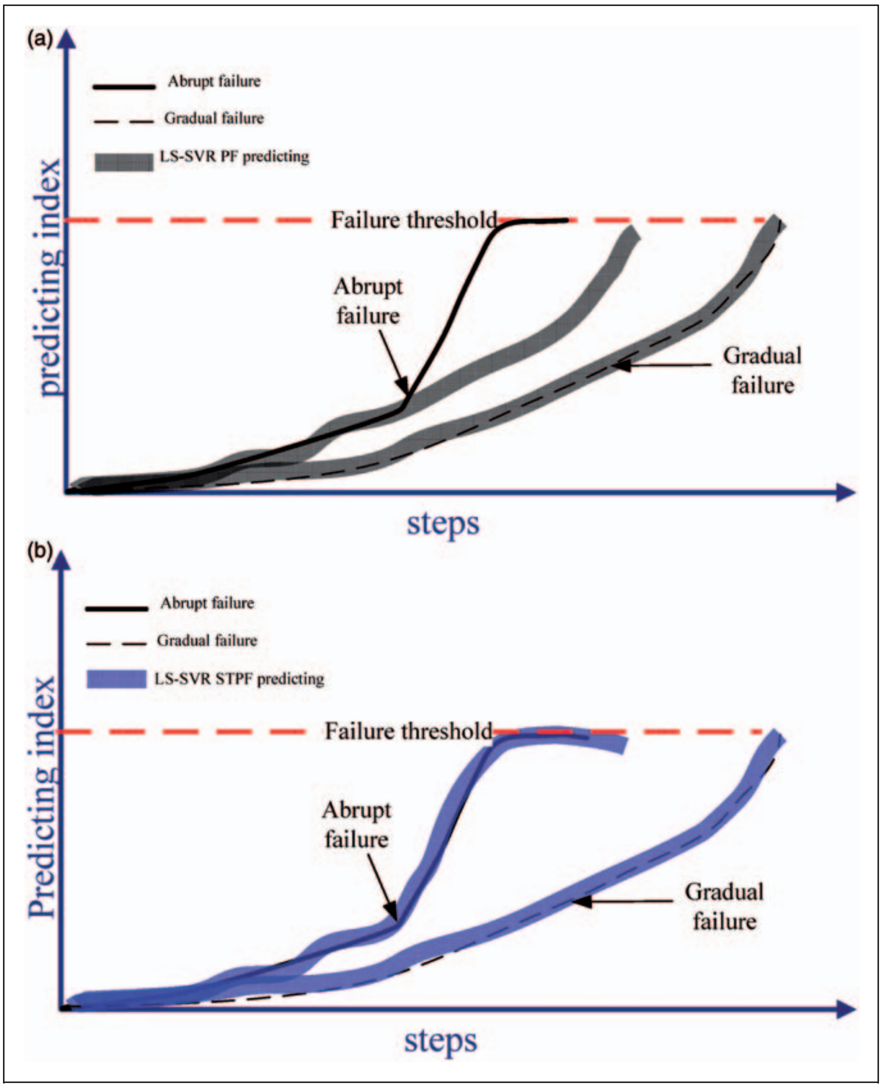

Figure 2. Methods for predicting (a) traditional method and (b) proposed method.
LS-SVR PF: least squares support vector regression particle filter; LS-SVR STPF: least squares support vector regression strong tracking particle filter.

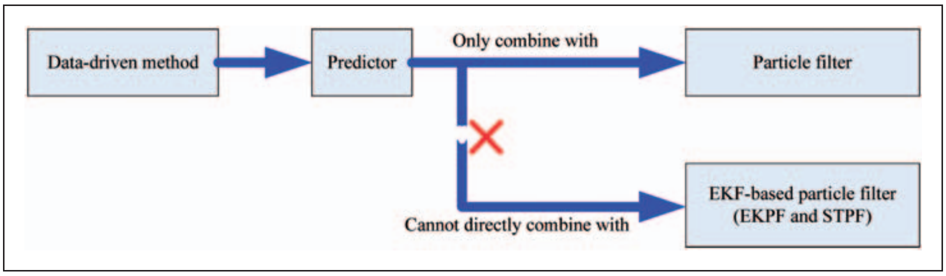

Figure 3. Traditional data-driven particle filter for prognosis.
EKPF: extended Kalman particle filtering; STPF: strong tracking particle filter.

matrix). It ignores the rest of the higher order term, and nonlinear problem is transformed into linear. In this processing we need to derive Jacobian matrix. To overcome the problem, a new part "first partial derive function of data-driven predictor" is added to the traditional data-driven PF framework, which builds a bridge between data driven method and EKF-based particle filter. Finally, a novel integrated prognostic method, named LS-SVR STPF, is proposed in this paper. The proposed method has the capability of prognosing exact failure when saltatory state arises in a system.

The remainder of this paper is organized as follows: in the next section, the proposed method is introduced in detail. The LS-SVR and trend prediction of states based on LS-SVR are introduced. In section "LS-SVR STPF", the regression function of LS-SVR is extended, we derive the first order partial derivative function of LS-SVR regression function for SFEKF, and STPF is also presented. In section "LS-SVR STPF method for failure prognosis", the overall proposed algorithm for prediction is illustrated. Section "Experiments and results" presents the experiment setup and results, and the comparisons of the proposed method with other methods are provided. Finally, we conclude our work in section "Conclusions".

LS-SVR STPF

Model-based method can be combined with EKF-based PF to improve its performance, but it is a hard issue to obtain a proper mathematical model in real applications; therefore, data-driven method cannot directly combine with the EKF-based PF. In this section, the proposed method which can overcome the limitation is introduced in detail. The diagram of our method is shown in Figure 4.

The whole method (LS-SVR STPF) can be separated into four parts: (1) data-driven method (LS-SVR is chosen in this paper), (2) predictor, (3) extend of regression function, and (4) EKF-based particle filter.

Brief review of LS-SVR

Suykens and Vandewalle introduced least squares support vector machines,[9] whose formulation employs equality type constraints. This allows the solution to be found by solving a set of linear equations, instead of the quadratic programming problem that classical SVMs solve.

The training set is depicted as $D = \{(\boldsymbol{P}_i, y_i)| i = 1, 2, \ldots, M\}$, $\boldsymbol{P}_i \in R_n, y_i \in R$, where $\boldsymbol{P}_i$ is the input data and y_i the output data. In the primal space, LS-SVM regression involves solving the following optimization problem.

$$\min J(\boldsymbol{w}, \xi) = \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} + \frac{1}{2}C\xi^T\xi \tag{1}$$

subject to

$$\boldsymbol{w}^T\phi(\boldsymbol{P}_i) + b - y_i + q_i = 0, \quad i = 1, 2, \ldots, , M \tag{2}$$

where $C > 0$ is a regularization parameter, $\phi(\cdot)$ is the nonlinear mapping function in kernel space, ξ is the error variable and b is the bias term. The Lagrangian for the problems (1) and (2) are given by

$$L(\boldsymbol{w}, b, \xi, \alpha) = \frac{1}{2}\left(\boldsymbol{w}^T\boldsymbol{w}\right) + \frac{C}{2}\|\xi\|^2 - \sum_{i=1}^{M}\alpha_i\left(\boldsymbol{w}^T\phi(\boldsymbol{P}_i) + b + \xi_i - y_i\right) \tag{3}$$

where α_i is the Lagrangian multiplier that may be of any sign, as follows from the equality constraints of the Karush–Kuhn–Tucker (K–K–T) conditions. These conditions may be compactly written in the form

$$\begin{bmatrix} \mathbf{K} + \frac{1}{C}\mathbf{I} & \boldsymbol{e} \\ \boldsymbol{e}^T & 0 \end{bmatrix}\begin{bmatrix} \alpha \\ b \end{bmatrix} = \begin{bmatrix} \boldsymbol{Y} \\ 0 \end{bmatrix} \tag{4}$$

where $\boldsymbol{Y} = [y_1, y_2, \ldots, y_M]^T$; $\boldsymbol{e}$ is a vector of ones of appropriate dimension, $\alpha = [\alpha_1, \alpha_2, \ldots, \alpha_M]^T$ are the Lagrange parameters, $\mathbf{I}$ is an identity matrix of appropriate dimension and $\mathbf{K}$ denotes the kernel matrix.

The linear kernel, where

$$\phi\left(\boldsymbol{P}_i, \boldsymbol{P}_j\right) = \left\langle \boldsymbol{P}_i, \boldsymbol{P}_j \right\rangle \tag{5}$$

the polynomial one, where

$$\phi\left(\boldsymbol{P}_i, \boldsymbol{P}_j\right) = \left(1 + \left\langle \boldsymbol{P}_i, \boldsymbol{P}_j \right\rangle\right)^2 \tag{6}$$

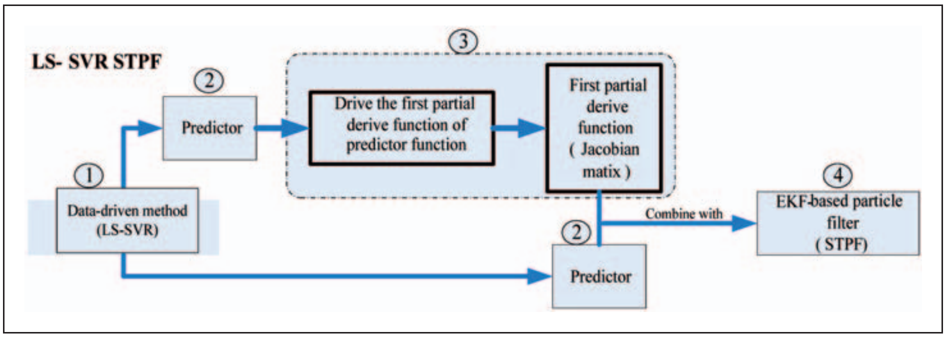

Figure 4. The proposed method.
LS-SVR STPF: least squares support vector regression strong tracking particle filter; LS-SVR: least squares support vector regression; EKF: extend Kalman filter; STPF: strong tracking particle filter.

and the Gaussian kernel, defined by

$$\phi(\boldsymbol{P}_i, \boldsymbol{P}_j) = \exp\left(-\frac{\|\boldsymbol{P}_i - \boldsymbol{P}_j\|^2}{2\delta^2}\right), \quad \delta > 0 \tag{7}$$

are commonly used kernel functions.

The resulting LS-SVM regressor is given by:

$$f(\boldsymbol{P}) = \sum_{i=1}^{M} \alpha_i \phi(\boldsymbol{P}_i, \boldsymbol{P}) + b \tag{8}$$

For a fixed choice of $\mathbf{K}$ and the regularization parameter C, equation (4) constitutes a system of linear equations that can easily be solved for b and α.

LS-SVR predictor

In general, there are two methods to model the LS-SVR predictor. One is direct prediction and the other is recursive prediction. For short-term prediction, the direct prediction method is always used; for RUL prediction, recursive prediction method is always utilized since we do 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:

$$\mathbf{P} = \begin{bmatrix} \boldsymbol{p}_1 & \boldsymbol{p}_2 & \cdots & \boldsymbol{p}_{k-m+1} \end{bmatrix}^T = \begin{bmatrix} x_1 & x_2 & \dots & x_m \\ x_2 & x_3 & \dots & x_{m+1} \\ \vdots & \vdots & \ddots & \vdots \\ x_{k-m+1} & x_{k-m+2} & \dots & x_k \end{bmatrix} \tag{9}$$

and

$$\hat{\boldsymbol{X}} = \begin{bmatrix} \hat{x}_1 \\ \hat{x}_2 \\ \vdots \\ \hat{x}_{k-m+1} \end{bmatrix} = \begin{bmatrix} x_{m+1} \\ x_{m+2} \\ \vdots \\ x_{k+1} \end{bmatrix} \tag{10}$$

where m is the embedding dimension, named mth order Markov model.[19] $\mathbf{P}$ is the input matrix and $\hat{\boldsymbol{X}}$ is the output vector. k denotes current time instant. Then LS-SVR model can be trained with $\mathbf{P}$ and $\hat{\boldsymbol{X}}$. The regressive function (or m-order model) is

$$\hat{x}_s = \sum_{i=1}^{l-m} \alpha_i \phi(\boldsymbol{P}_i, \boldsymbol{P}_s) + b \tag{11}$$

thus, the prediction of the state at t_{k+1} is

$$\hat{x}_{k+1} = \sum_{i=1}^{l-m} \alpha_i \phi(\boldsymbol{P}_i, \boldsymbol{P}_{k-m+1}) + b \tag{12}$$

Let $\boldsymbol{P}_{k-m+2} = [x_{k-m+2} \;\; x_{k-m+3} \dots x_k \;\; \hat{x}_{k+1}]$, then the prediction at t_{k+2} is

$$\hat{x}_{k+2} = \sum_{i=1}^{k-m+1} \alpha_i \phi(\boldsymbol{P}_i, \boldsymbol{P}_{k-m+2}) + b \tag{13}$$

Recursively, the prediction model at t_{k+j} is

$$\hat{x}_{k+j} = \sum_{i=1}^{k-m+1} \alpha_i \phi(\boldsymbol{P}_i, \boldsymbol{P}_{k-m+j}) + b \tag{14}$$

where $\boldsymbol{P}_{k-m+j} = [x_{k-m+j} \dots x_k \;\; \hat{x}_{k+1} \dots \hat{x}_{k+j-1}]$.

LS-SVR regression function extend for SFEKF

The failure evolution of nonlinear systems can be explained by the state evolution model (15) and the observation model (16).

$$\boldsymbol{x}_{k+1} = f(k, \boldsymbol{u}_k, \boldsymbol{x}_k) + \Gamma_k \boldsymbol{v}_k \tag{15}$$

$$\boldsymbol{y}_{k+1} = h(k+1, \boldsymbol{x}_{k+1}) + \boldsymbol{e}_{k+1} \tag{16}$$

where $k \geqslant 0$ is the variable of discrete time; $\boldsymbol{x} \in R^n$ is the variable of state; $\boldsymbol{u} \in R^p$ is the vector of input; $\boldsymbol{y} \in R^m$ is the vector of output, nonlinear state function f: $R^p \times R^n \to R^n$, h: $R^n \to R^m$. $\Gamma \in R^{\mathrm{n}} \times R^q$ is known matrix, $\boldsymbol{v}_k$ is q dimension and $\boldsymbol{e}_k$ is m dimension Gaussian white noise.

$$\begin{aligned} E[\boldsymbol{v}_k] &= E[\boldsymbol{e}_k] = 0 \\ E\left[\boldsymbol{v}_k \boldsymbol{v}_j^T\right] &= Q_k \delta_{k,j} = 0 \\ E\left[\boldsymbol{e}_k \boldsymbol{e}_j^T\right] &= R_k \delta_{k,j} = 0 \\ E\left[\boldsymbol{v}_k \boldsymbol{e}_j^T\right] &= Q_k \delta_{k,j} = 0 \end{aligned}$$

where Q_k is symmetric nonnegative fixed array; R_k is symmetric negative fixed array. Initial state $\boldsymbol{x}_0 \sim N(\boldsymbol{x}_0, \mathrm{P}_0)$, and $\boldsymbol{x}_0$, $\boldsymbol{v}_k$ and $\boldsymbol{e}_k$ are statistical independence.

In the proposed method, the optimal state estimation $\hat{\boldsymbol{x}}_{k+1|k+1}$ and optimal estimate covariance $\hat{\mathrm{P}}_{k+1|k+1}$ are obtained for producing particles in PF framework. The state transition $F(k, \hat{\boldsymbol{x}}_{k|k})$ needs to be computed firstly in SFEKF algorithm. The detailed process of SFEKF algorithm is given in Ref. 28. Here, equation (16) can be simply described as $\boldsymbol{y}_k = \boldsymbol{x}_k + \boldsymbol{e}_k$, since both the model state $\boldsymbol{x}_k$ and output $\boldsymbol{y}_k$ represent the machine condition indicator (or monitoring index).

The state transition $F(k, \hat{\boldsymbol{x}}_{k|k})$ is defined to be the following as equation (17).

$$F(k, \hat{\boldsymbol{x}}_{k|k}) = \left.\frac{\partial f(k, \boldsymbol{u}_k, \boldsymbol{x}_k)}{\partial \boldsymbol{x}}\right|_{\boldsymbol{x}_k = \hat{\boldsymbol{x}}_{k|k}} \tag{17}$$

where $\boldsymbol{u}_k \in R^p$ is zero, if there is no input controlling parameter.

In this subsection, we will describe the derivation of the first-order partial derivative function of LS-SVR regression function for SFEKF. The partial derivative function of LS-SVR put up a bridge for LS-SVR and SFEKF. The trained LS-SVR can get m-order model $f(\cdot)$ (equation (18)) and fault growth model (equation (19)):

$$f_{LS\text{-}SVR}(\boldsymbol{P}_s) = \sum_{i=1}^{k-m+1} \alpha_i \phi(\boldsymbol{P}_i, \boldsymbol{P}_s) + b \tag{18}$$

$$\hat{x}_{s+1} = f_{LS\text{-}SVR}(\boldsymbol{P}_s) + w_s \tag{19}$$

where $\boldsymbol{P}_s = [x_{s-m+1}, \ldots, x_s]$ is m dimension input vector. Here, $f_{LS\text{-}SVR}(\boldsymbol{P}_s)$ is a scalar output of predictor. w_s is the noise.[19]

Computing $f(\boldsymbol{P}_s)$ derivative on element x_s is expressed as $\nabla_{x_s} f_{LS\text{-}SVR}(\boldsymbol{P}_s) \in R$, the derivation process is as follows:

$$\begin{aligned}\nabla_{x_s} f_{LS\text{-}SVR}(\boldsymbol{P}_s) &= \frac{\partial f_{LS\text{-}SVR}(\boldsymbol{P}_s)}{\partial x_s} \\ &= \mathrm{Tr}\left[\left(\frac{\partial f_{LS\text{-}SVR}(\boldsymbol{P}_s)}{\partial \boldsymbol{P}_s}\right)^T \frac{\partial \boldsymbol{P}_s}{\partial x_s}\right]\end{aligned} \tag{20}$$

Equations (21) and (22) must be computed for the result of equation (20), where $\frac{\partial f_{LS\text{-}SVR}(\boldsymbol{P}_s)}{\partial \boldsymbol{P}_s} \in R^{1\times m}$ and $\frac{\partial \boldsymbol{P}_s}{\partial x_s} \in R^{1\times m}$ are as follows:

$$\begin{aligned}\nabla_{\boldsymbol{P}_s} f_{LS\text{-}SVR} &= \frac{\partial f_{LS\text{-}SVR}}{\partial \boldsymbol{P}_s} = \sum_{i=1}^{k-m+1} \alpha_i \frac{\partial \phi(\boldsymbol{P}_i, \boldsymbol{P}_s)}{\partial \boldsymbol{P}_s} \\ &= [\nabla_{s-m+1}, \ldots, \nabla_s]_{1\times m}\end{aligned} \tag{21}$$

$$\frac{\partial \boldsymbol{P}_s}{\partial x_s} = \frac{\partial [x_{s-m+1} \cdots x_s]}{\partial x_s} = [0 \ldots 01]_{1\times m} \tag{22}$$

$\nabla_{x_s} f_{LS\text{-}SVR}(\boldsymbol{P}_s)$ can be derived by equations (20)–(22). The result is as follows:

$$\begin{aligned}\frac{\partial f_{LS\text{-}SVR}(\boldsymbol{P}_s)}{\partial x_s} &= \mathrm{Tr}\left[\left(\frac{\partial f_{LS\text{-}SVR}(\boldsymbol{P}_s)}{\partial \boldsymbol{P}_s}\right)^T \frac{\partial \boldsymbol{P}_s}{\partial x_s}\right] \\ &= \mathrm{Tr}\begin{bmatrix} 0 & \cdots & 0 & \nabla_{s-m+1} \\ 0 & \cdots & 0 & \nabla_{s-m+2} \\ \vdots & \cdots & \vdots & \vdots \\ 0 & \cdots & 0 & \nabla_s \end{bmatrix}_{m\times m} = \nabla_s\end{aligned} \tag{23}$$

In equation (18), $\phi(\cdot)$ is selected as a Gauss kernel function

$$\phi(\boldsymbol{P}_i, \boldsymbol{P}_s) = \exp\left\{\frac{-\|\boldsymbol{P}_s - \boldsymbol{P}_i\|^2}{2\sigma^2}\right\} \quad i = 1, 2, \ldots, k-m+1 \tag{24}$$

$$\begin{aligned}&\nabla_{\boldsymbol{P}_s} f_{LS\text{-}SVR} \\ &= \sum_{i=1}^{k-m+1} \alpha_i \frac{\partial \phi(\boldsymbol{P}_i, \boldsymbol{P}_s)}{\partial \boldsymbol{P}_s} \\ &= \sum_{i=1}^{k-m+1} \alpha_i \exp\left\{\frac{-\|\boldsymbol{P}_s - \boldsymbol{P}_i\|^2}{2\sigma^2}\right\}\left(-\frac{1}{\sigma^2}\right)(\boldsymbol{P}_s - \boldsymbol{P}_i) \\ &i = 1, 2, \ldots, k-m+1\end{aligned} \tag{25}$$

Combing with equations (20) and (22), equation (25) can derive the $\nabla_{x_s} f_{LS\text{-}SVR}(\boldsymbol{P}_s)$ when Gauss kernel function was chosen.

The state transition F of m-order model $f(.)$ can be computed by equation (26) as

$$F(k, \hat{x}_s) = \frac{\partial f_{LS\text{-}SVR}(\boldsymbol{P}_s)}{\partial \hat{x}_s} = \mathrm{Tr}\left[\left(\frac{\partial f_{LS\text{-}SVR}}{\partial \boldsymbol{P}_s}\right)^T \frac{\partial \boldsymbol{P}_s}{\partial \hat{x}_s}\right] \tag{26}$$

Then, equation (26) is as a ''bridge'' to realize the LS-SVR predictor combining with SFEKF.

Strong tracking particle filter

STF as the new importance density function is introduced into PF algorithm for particles update, and finally gets the approximate posterior state pdf as the importance density function, which is equation (27).

$$q(x_k^i | x_{k-1}^i, y_k) = N(\hat{x}_{k+1|k}^i, \mathrm{P}_{k+1|k}^i) \tag{27}$$

The new particles are produced from importance density function, and then the resampling is carried out after updating weight. The STFPF is as follows:

STPF algorithm:

Step 1: Initiate the system. Setting $k = 0$, sampling based on the initial value $\hat{x}_{0|0}$ and $\mathrm{P}_{0|0}$, get initial particle collection $\{\langle x_0^i, 1/n\rangle | i = 1, 2, \ldots, n\}$.

Step 2: Call the **SFEKF**[28] to update the each particle on the particle collection, getting the particle of the importance density.

$$q(x_k^i | x_{k-1}^i, y_k) = N(\hat{x}_{k+1|k}^i, \mathrm{P}_{k+1|k}^i)$$

Step 3: Update weight of each particle

$$\omega_k^i = \omega_{k-1}^i \times \frac{p(y_k | x_k^i)\, p(x_k^i | x_{k-1}^i)}{q(x_k^i | x_{k-1}^i, y_k)}$$

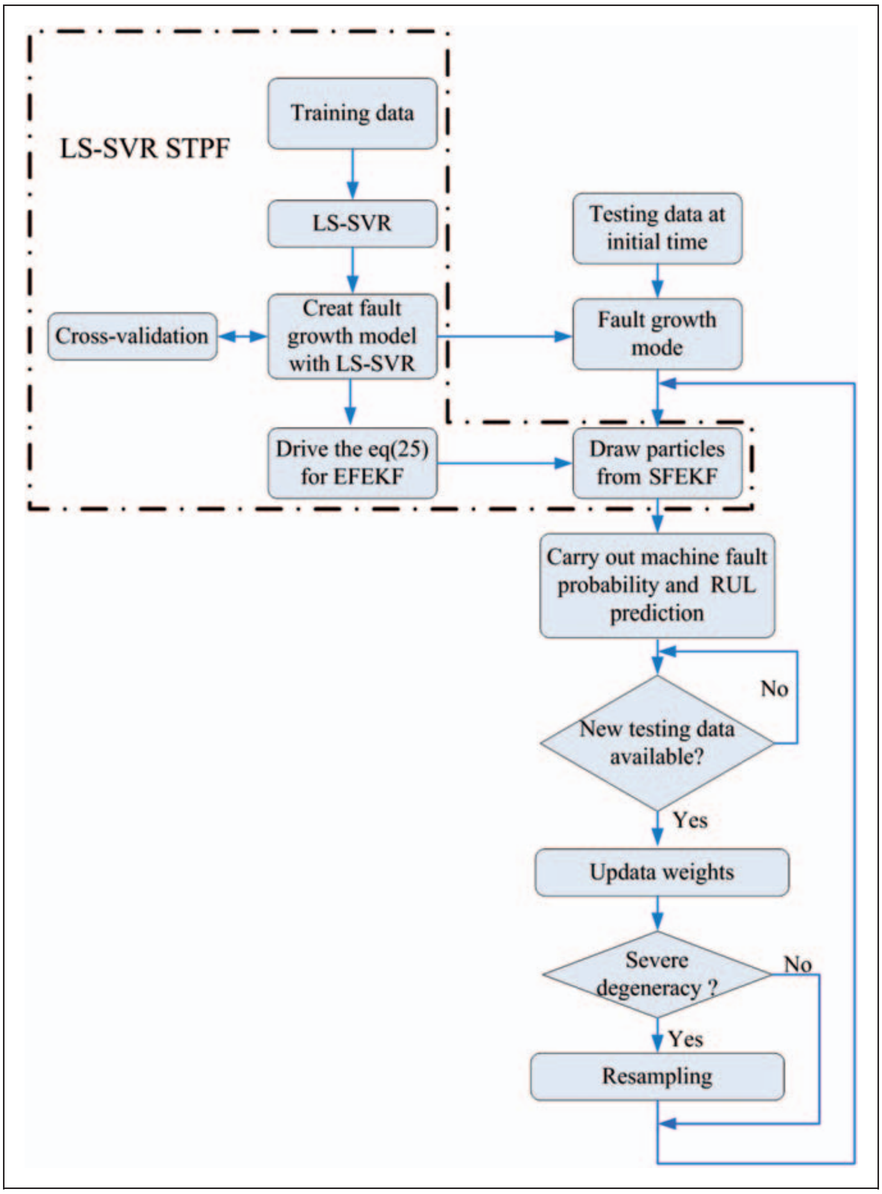

Figure 5. Flowchart of the proposed method for machine failure probability and RUL prediction.
RUL: remaining useful life; LS-SVR: least squares support vector regression; SFEKF: Suboptimal fading extended Kalman filtering.

Step 4: Weight normalization

$$\omega_k^i = \frac{\omega_k^i}{\sum_{i=1}^{n} \omega_k^i}$$

Step 5: State estimate

$$x_k^* = \sum_{i=1}^{n} \omega_k^i \times x_k^i$$

Step 6: If the valid number of the sample $N_{eff} < n/3$,then resampling

$$x_k^i \sim N\left(\hat{x}_{k+1|k+1}^i, \mathrm{P}_{k+1|k+1}^i\right)$$

Step 7: $k = k + 1$, return step 2

LS-SVR STPF method for failure prognosis

The flowchart of the proposed method for machine fault probability and RUL prediction is shown in Figure 5. In the whole flowchart, "LS-SVR STPF" is introduced into the classical framework[14] for failure prognosis.

When LS-SVR STPF method is used for prognosis and the p-step-ahead prediction is operated, values of existing particles are updated, but ω_k remains unchanged, predicting state in the future time point of $k+1$, $k+2, \ldots, k+p$, and resulting in failure prediction probability[29] and RUL. The detail processing flow is listed below.

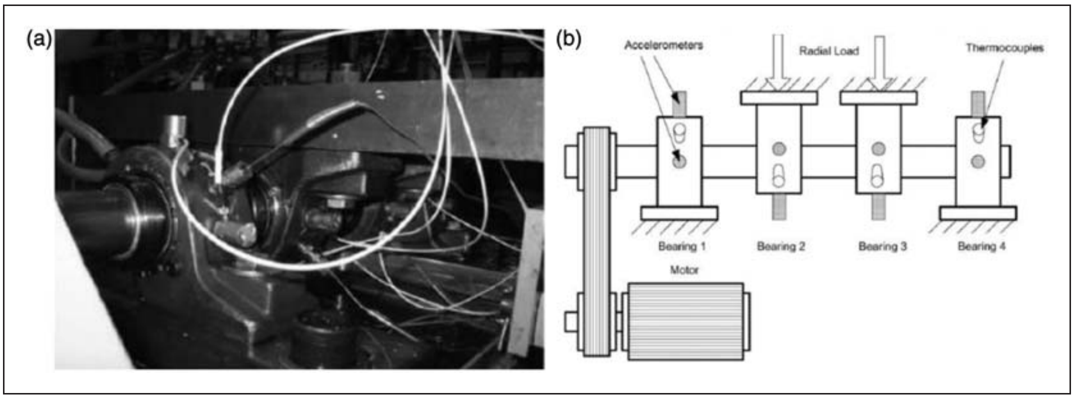

Figure 6. Bearing test rig.[30]

Step 1: The LS-SVR is trained with available condition data to model the fault propagation process, and then equation (25) is derived for connecting STF.

Step 2: The fault growth model (equation (18)), represented by the LS-SVR and the process noise, is employed with an STPF 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: the long-term (p-step-ahead) condition prediction also can be computed by successively taking the expectation of the model update (equation (18)) for every future time instant, considering the calculated condition value associated to each particle as initial condition value for the next step prediction, as shown in

$$x^*_{k+p} = E\left[x^i_{k+p}\right], \quad x^i_{k+p} = x^*_{k+p} + w_{k+p-1}$$

When all of the predicted values associated with each particle reach the predefined condition threshold, fault prediction probability can be computed based on Ref. 29, that is

$$fault(j,k) = \sum_{i=1}^{N} \omega^i_k I\left(x^i_{k+j|k+j-1} \in \omega_0\right), \quad j \in [1,p]$$

where the $p=1$ in the paper. ω_0 is the failure threshold. $I(A)$ is a Boolean function. If the A is true, $I(A)=1$, otherwise $I(A)=0$; and the expected value of RUL can be obtained from the RUL pdf, as shown below:

$$\mathrm{RUL} = \sum_{i=1}^{N} l^i \omega^i_{k-1} \tag{28}$$

where N is the total number of particles, l^i is the RUL of the ith particle, and ω^i_{k-1} 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 equation (18). If the valid number of the sample $N_{eff} < N/3$, then resampling has to be done.

Step 3: Repeat step 2 until failure probability and machine RUL prediction are complete.

Experiments and results

The failure prognostics method presented previously is tested on a condition-monitoring database taken from Ref. 30, and contains several bearings tested until the failure. The test data were extracted from NASA's prognostics data repository.[30] During the experiments, four bearings were tested under constant conditions. The angular velocity was kept constant at 2000 rpm, and a 6000 lb radial load was applied onto the shaft and bearings (Figure 6). On each bearing, two accelerometers were installed for a total of eight accelerometers (one vertical, and one horizontal) to register the accelerations generated by the vibrations, with a sampling frequency equal to 20 kHz. For simulation purposes, the data which is produced by bearing 3 and has inner race defect is chosen, and the data that finally twelve days (in the last 12 days) is used in our experiments, because the state change is not obvious in the first 23 days.[31]

Seven features, e.g. mean, median, standard variance, skewness, root mean square, standard error, and kurtosis are extracted from the original signal. In order to improve the feature information and compute effectively, the local linear embedding (LLE)[32] which is a nonlinear feature dimension reduction method for the high dimension sample space is used for feature extraction. The feature produced by LLE is taken as the monitoring index. In our experiment, 12 (empirical value) is chosen in LLE as the number of nearest neighbours, and the one dimension feature subspace is extracted in original feature space. The flow diagram of new feature producing processing is shown in Figure 7.

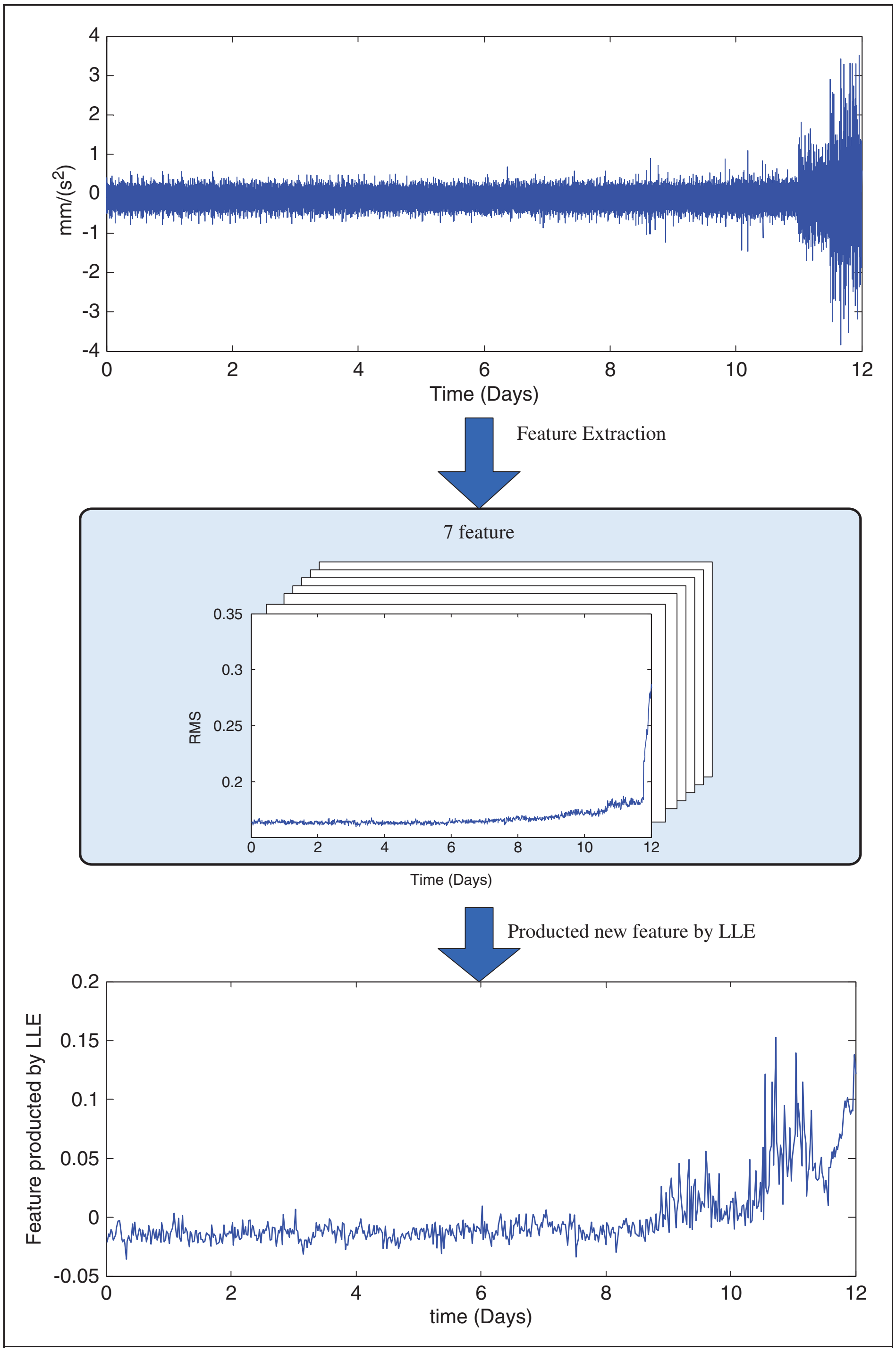

Figure 7. The flow diagram of processing of new feature extraction.
RMS: root-mean square; LLE: local linear embedding.

Firstly, in order to compare the performance in one-step-ahead prediction of other classical predictors with our method (predictor), the results of prediction are presented in Figure 8. The predictor is trained with monitoring index acquired from a bearing with inner race defect in vertical, and the test monitor index is the same bearing in horizontal with noise, the LS-SVR is trained with training set to model the fault growth model. Figure 8 shows the comparison results of the fourth-order and one step-ahead prediction for the monitoring index of inner race defect in the last 12 days. According to Figure 8(a) to (c), it can be concluded that LS-SVR, LS-SVR PF and LS-SVR EKPF fail to capture the system's new dynamics after about 10 days, so that prediction values of low accuracy are output. But the proposed method (Figure 8(d)) can capture the system's dynamics quickly and accurately.

Figure 9(a) shows the absolute error of fourth-order and one-step-ahead prediction by the four

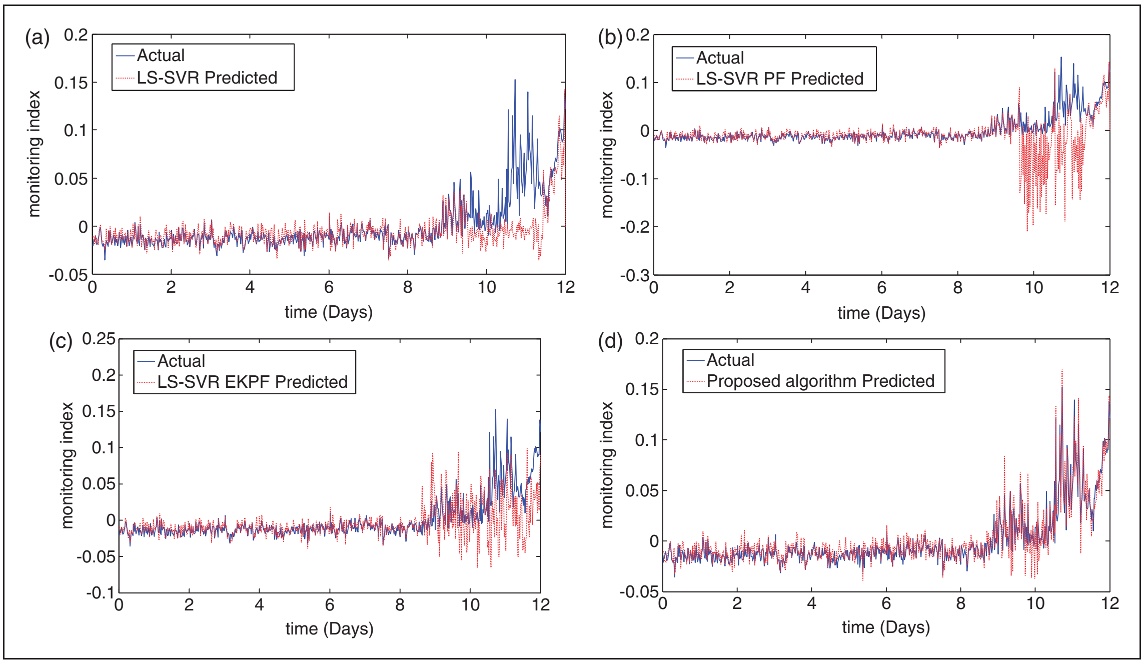

Figure 8. Four-order and one-step-ahead prediction results: (a) LS-SVR; (b) LS-SVR PF; (c) LS-SVR EKFPF and (d) proposed method. LS-SVR: least squares support vector regression; LS-SVR PF: least squares support vector regression particle filter; LS-SVR EKPF: least squares support vector regression extended Kalman particle filtering.

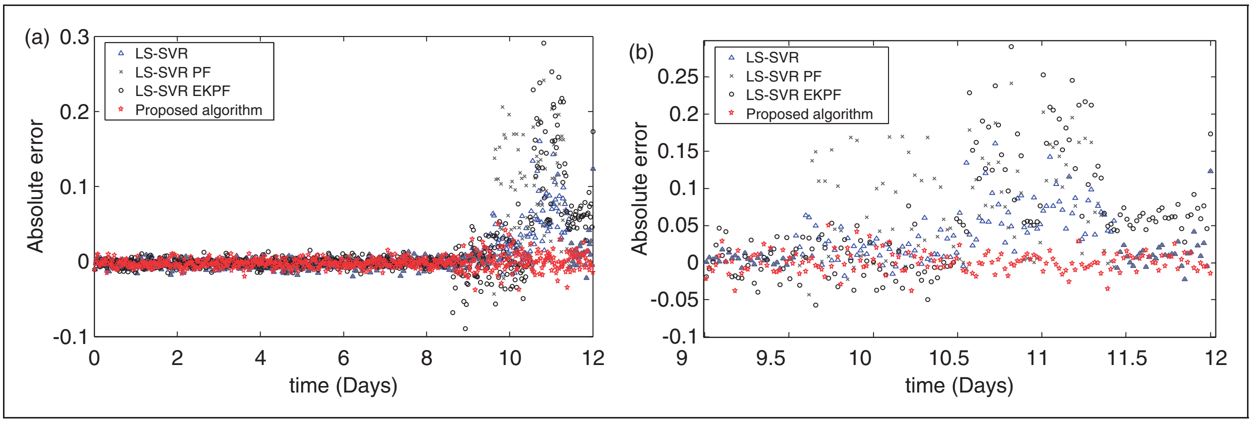

Figure 9. Absolute error (a) 1–12 days absolute error and (b) 10–12 days absolute error.
LS-SVR: least squares support vector regression; LS-SVR PF: least squares support vector regression particle filter; LS-SVR EKPF: least squares support vector regression extended Kalman particle filtering.

Table 1. Prediction of RMSE comparison for faulty bearing.

Four-order and one-step ahead prediction of RMSE	RMSE (1–12 days)	RMSE (10–12 days)
LS-SVR	0.0083	0.4302
LS-SVR PF	0.0127	0.6071
LS-SVR EKFPF	0.0075	0.2546
Proposed algorithm	0.0010	0.0010

RMSE: root-mean square error; LS-SVR: least squares support vector regression; LS-SVR PF: least squares support vector regression particle filter.

algorithms in 12 days. And Figure 9(b) shows absolute error in the last three days. In order to evaluate the prediction performance, the root-mean square error (RMSE)[33] is used as measuring metric. Prediction results are given at different time range in Table 1. It is clear that the prediction accuracy of the proposed approach is superior to the LS-SVR, LS-SVR PF and LS-SVR EKPF.

Then, the ultimate objective of fault prediction is to obtain fault probability and the RUL prediction, which is the remaining time before the system fault indicator crosses its corresponding failure threshold. Due to the indeterminacy of prediction, the result

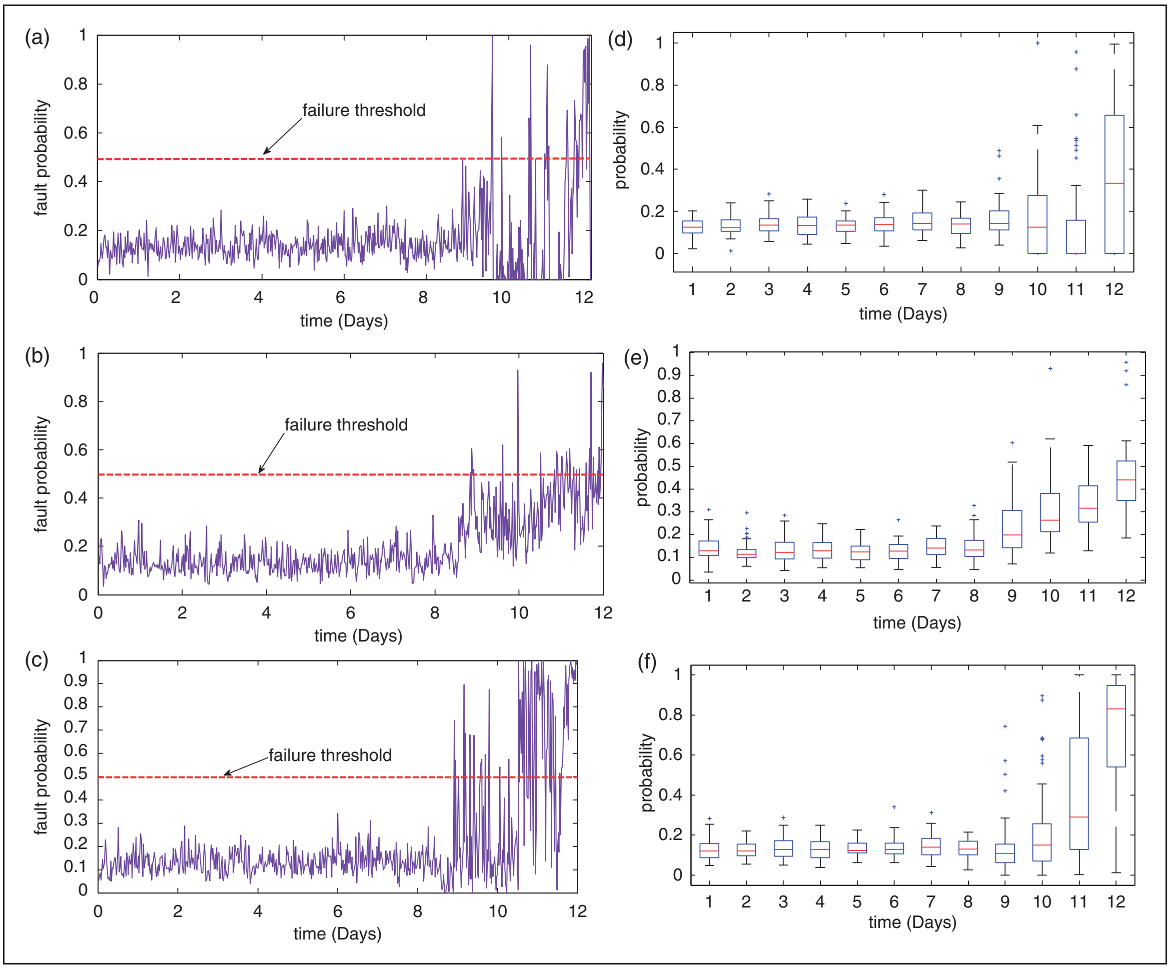

Figure 10. (a)–(c) are the fault probability of three methods in every step and (d)–(f) are the fault probability distribution of three methods in every day.

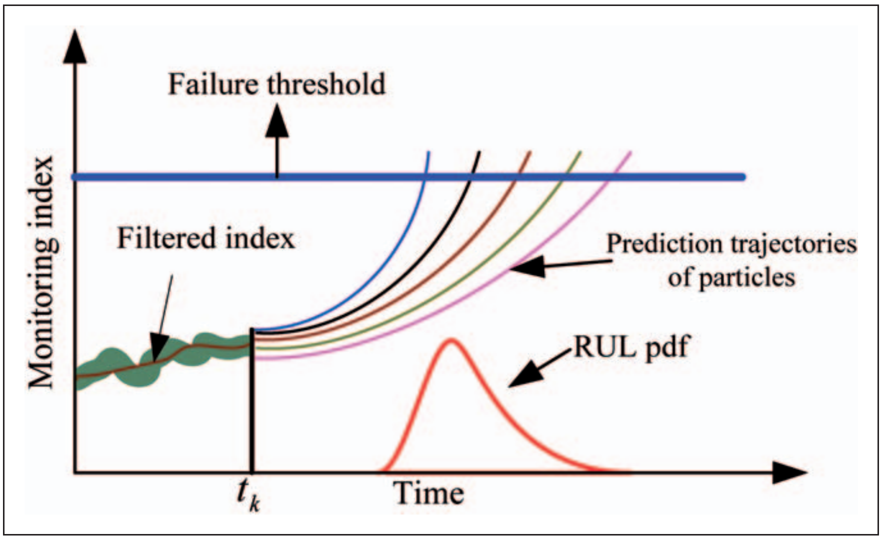

Figure 11. Schematic of prediction of system RUL.
RUL: remaining useful life.

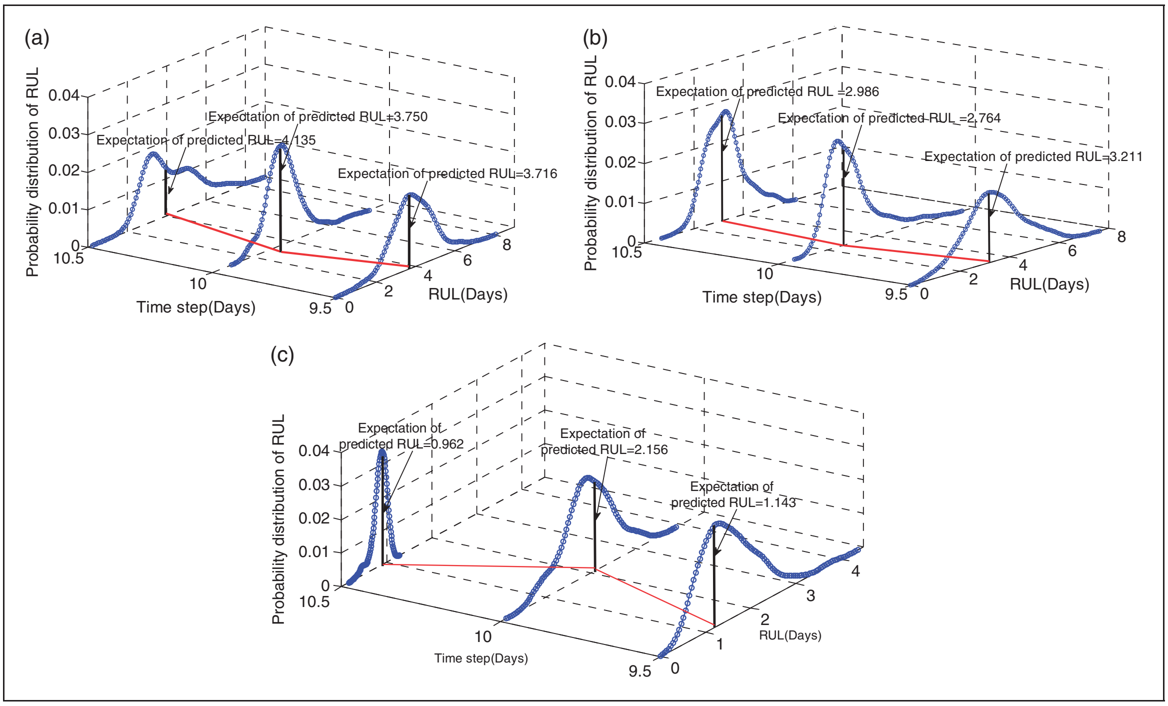

Figure 12. Prediction results of RUL distribution (a) LS-SVR PF (b) LS-SVR EKPF and (c) Proposed method.
RUL: remaining useful life; LS-SVR PF: least squares support vector regression particle filter; LS-SVR EKPF: least squares support vector regression extended Kalman particle filtering.

should be described in the form of probability, and given a predetermined failure threshold of a state. In our experiment, the predetermined failure threshold is set to 0.1, the value which corresponds to about 11th day in the last time.[34] When the processing of one-step-prediction finishes, the fault probability is calculated in every step. The fault probability values of the every step resulting from LS-SVR PF, LS-SVR EKPF and the proposed method are shown in Figure 10(a) to (c). When the fault probability value exceeds the threshold 0.5, the fault probability value is high, and the system can be considered to have a failure at this point of time. We also clearly see that the four methods have similar performance in approximately nine days before, and different performance in the later days.

Fault probability can be seen in Figure 10(a), which is calculated by LS-SVR PF. According to Ref. 23, the fault probability can be calculated. The fault probability values which exceed the threshold are very sparse, especially after the 11th day, the results and actual condition compare unfavorably. In Figure 10(b), most of the fault probability values calculated by LS-SVR EKPF exceed the threshold, but the values are below 0.6, and still do not conform to the actual situation. The proposed method is compared with the former two methods. Most values of fault probability are ranged in 0.7 and 1 after the 10th day. The result is shown in Figure 10(c). Because the prediction performance is good, the predicted results are consistent with the actual values. It correctly captures the system response effectively, especially at the end of the testing phase.

In order to make the experimental results more clear, fault probability distribution of each day is counted with boxplot form. From the fault development analysis of the experiment data, we can conclude that the fault probability should be in range between 0.5 and 0.8 on the 11th day, while the results of the former two methods are not consistent with the actual condition (Figure 10(d) and (e)). But the proposed method can output the correct range result as presented in Figure 10(f).

Figure 11 depicts the implementation process of the prediction algorithm for RUL prediction[14] The long-term predictions start at time $t_{k.}$ According to the fault probability in every day, the RUL predictions are carried out from the ninth day using the current estimate for the state pdf as initial condition. Then the long-term predictions are carried out until the predicted values of all particles reach a predefined failure threshold. Based on the result, the RUL of each particle is determined, and used to form an RUL pdf. The RUL expectation can be obtained from the RUL pdf based on equation (28).

Figure 12 (a) to (c) shows the results of the probability distribution of RUL which was obtained by LS-SVR PF, LS-SVR EKPF, and the proposed method in the important days (9.5th, 10th and 10.5th), because values of the fault probability are

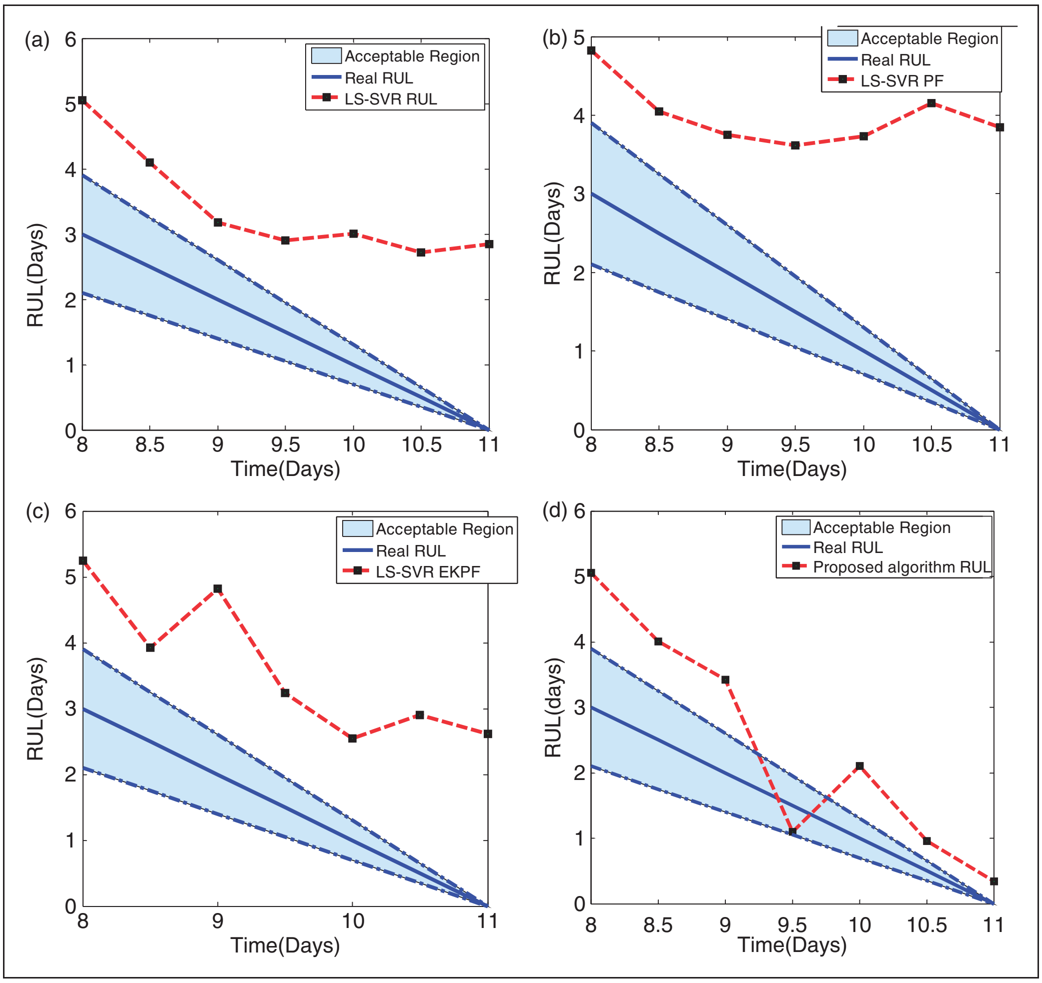

Figure 13. RUL prediction results. (a) LS-SVR; (b) LS-SVR PF; (c) LS-SVR EKFPF and (d) Proposed method.
RUL: remaining useful life; LS-SVR: least squares support vector regression; LS-SVR PF: least squares support vector regression particle filter; LS-SVR EKPF: least squares support vector regression extended Kalman particle filtering

Table 2. Prognostics performance metrics.

	Accuracy	RMSE	$\alpha-\lambda$	CRA
LS-SVR	0.354 2	2.445	False	0.152
LS-SVR PF	0.298 3	3.036	False	0.104
LS-SVR EKPF	0.321	1.980	False	0.213
Proposed method	0.603	1.002	True	0.390

RMSE: root-mean square error; LS-SVR: least squares support vector regression; LS-SVR PF: least squares support vector regression particle filter.

also higher in these days before the failure of the bearing. As the results, the expectation of the predicted RUL is separately 3.716, 3.750 and 4.135 (the unit is days) with LS-SVR PF. 3.211, 2.764 and 2.988 with LS-SVR EKF, and 1.143, 2.156, 0.962 with the proposed method. And when using the proposed method, the probability distribution of RUL has smaller variance and less number of peaks than the results of LS-SVR PF and LS-SVR EKPF in 10.5th day.

In order to better quantify the prognostic performance, the $\alpha-\lambda$ performance metric which is defined in Ref. 35 was used, and RUL prediction result from the LS-SVR, LS-SVR PF, LS-SVR EKPF, and the proposed method using the test data set are shown in Figure 13. The α parameter is set to 30 (acceptable region) for the estimation of $\alpha-\lambda$, the parameter is equal to 30%, and λ is equal to 0.5. However, λ is set 0, 0.25, 0.5, and 0.75 for the estimation of CRA.[35] And another three results of evaluate metric are given in Table 2; the accuracy metric for failure prognostics techniques were proposed in Ref. 36. RMSE as measuring metric was proposed in Ref. 33. The results of the former three methods are far away from the actual RUL values in the most significant time points. The proposed method outputs the RUL values in that the significant time points are very close to the real RUL values. From the results shown in Table 2, we can conclude that the performance of the proposed method is superior to that of the LS-SVR, LS-SVR PF, and LS-SVR EKPF.

Conclusions

Machinery condition prognosis is important for monitoring the degradation conditions. A novel method has been presented, which is based on LS-SVR predictor and STPF. LS-SVR is used to model the fault propagation trend. STF is combined with first-order partial derivative of LS-SVR predictor as a new importance function is introduced into the particle filter for improving the tracking ability in machine system state changes more drastic conditions and then improving the performance of predictor in prediction. Experiment results show that the proposed method has much greater values of performance metrics accuracy and CRA, and has lower RMSE. $\alpha - \lambda$ can illustrate that the proposed method owns more excellent confidence level. Therefore, LS-SVR, LS-SVR PF and LS-SVR EKPF are not suitable for failure prediction of this kind of system which has saltatory states, and proposed method has the much better performance in this respect. In future, other data-driven methods combined with STPF or RSPF will be investigated for prognosis, and more comprehensive experiments will be performed.

Funding

This work was supported in part by the Natural Science Foundation of China (NSFC) (51075330, 61003137, 50975231, 61202185) and the Scientific Research Foundation (SRF) for Returned Overseas Chinese Scholars (ROCS), Northwestern Polytechnical University (NWPU) fundamental fund, and Natural Science Basic Research Plan in Shanxi Province of China (2012JQ8037), Northwestern Polytechnical University (NWPU) fundamental fund (JC201202, JC201220).

References

1. Zhao F, Chen J, Guo L, et al. Neuro-fuzzy based condition prediction of bearing health. *J Vib Control* 2009; 15: 1079–1091.
2. Wang W, Golnaraghi F and Ismail F. Prognosis of machine health condition using neuro-fuzzy systems. *Mech Syst Signal Process* 2004; 18: 813–831.
3. Liu J, Wang W and Golnaraghi F. A multi-step predictor with a variable input pattern for system state forecasting. *Mech Syst Signal Process* 2009; 23: 1586–1599.
4. Wang P and Vachtsevanos G. Fault prognostics using dynamic wavelet neural networks. *Artif Intell Eng Des Anal Manuf* 2001; 15: 349–365.
5. Samanta B and Nataraj C. Prognostics of machine condition using soft computing. *Robot CIM-INT Manuf* 2008; 24: 816–823.
6. Tran V, Yang B and Tan A. Multi-step ahead direct prediction for machine condition prognosis using regression trees and neuro-fuzzy systems. *Expert Syst Appl* 2009; 36: 9378–9387.
7. Wang W. An adaptive predictor for dynamic system forecasting. *Mech Syst Signal Process* 2007; 21: 809–823.
8. Jardine AKS, Lin D and Banjevic D. A review on machinery diagnostics and prognostics implementing condition-based maintenance. *Mech Syst Signal Process* 2006; 20: 1483–1510.
9. Suykens JAK, van Gestel T, De Brabanter J, et al. *Least squares support vector machines*. Singapore: World Scientific Publishing Co. Pte. Ltd, 2002.
10. Xu BH, Li HR and Bai HF. Application of LS-SVM in prediction condition monitoring data. *Sci Technol Eng* 2007; 7: 3924–3926.
11. Sanjeev M and Simon M. A tutorial on particle filter for online nonlinear/no-Gaussian Bayesian tracking. *IEEE Trans Signal Process* 2002; 50: 174–188.
12. Yao A, Lin X, Wang G, et al. A compact association of particle filtering and kernel based object tracking. *Pattern Recognit* 2012; 45: 2584–2597.
13. Zheng F, Guo-Feng T and Xin-He X. A robust and effecient algorithm for mobile robot localization. *Acta Automatic Sinica* 2007; 33: 48–53.
14. Orchard M, Wu B and Vachtsevanos G. A particle filtering framework for failure prognosis. *World Tribology Congress III*, Washington, DC, USA, 12–16 September 2005. American Society of Mechanical Engineers, pp.883–884.
15. Chen MZ and Zhou DH. Particle filtering based fault prediction of nonlinear systems. *IFAC symposium proceedings of safe process*. Washington D.C., USA: Elsevier Science, 2003.
16. Zio E and Peloni G. Particle filtering prognostic estimation of the remaining useful life of nonlinear components. *Reliability Eng Syst Saf* 2011; 96: 403–409.
17. Orchard M and Vachtsevanos G. A particle filtering approach for on-line fault diagnosis and failure prognosis. *Trans Inst Meas Control* 2009; 31: 221–246.
18. Zhang B, Sconyers C, Patrick R and Vachtsevanos G. A multi-fault modeling approach for fault diagnosis and failure prognosis of engineering systems. In: *Annual conference of the prognostics and health management society*, San Diego, CA, 2009, pp.1–10.
19. Chen CC, Zhang B, Vachtsevanos G, et al. Machine condition prediction based on adaptive neuro-fuzzy and high-order particle filtering. *IEEE Trans Ind Electron* 2011; 58: 4353–4364.
20. Saha B, Goebel K, Poll S, et al. Prognostics methods for battery health monitoring using a Bayesian framework. *IEEE Trans Instrum Meas* 2009; 58: 291–296.
21. De Freitas JFG, Niranjan MA, Gee AH, et al. Sequential Monte Carlo methods to train neural network models. *Neural Comput* 2000; 12: 955–993.
22. Vander Merwe R, Doucet A, de Freitas N, et al. The unscented particle filter. Technical Report CUED/FUNFENG/TR380, Oregon Graduate Institute, USA, 2000, pp.1–46.
23. Murphy K and Russell S. *Rao-Blackwellised particle fitlering for dynamix Bayesian networks, sequential Monte Carlo methods in practice*. New York: Springer-Verlag, 2001, pp.500–512.
24. Wang Q, Xue L, Liu J, and Xiang Z. Enhancing particle swarm optimization based particle filter tracker. In: *Lecture Notes in Computer Science*, vol. 4114. Berlin/Heidelberg: Springer, 2006, pp.1216–1221.
25. Thrun S, Langford J and Verma V. *Risk sensitive particle filters*. Cambridge, MA, US: Neural Information Processing Systems (NIPS), 2001.

26. Verma V, Gordon G, Simmons R, et al. Particle filters for rover fault diagnosis. *IEEE Robotics & Automation Magazine*, June 2004, pp.56–64.
27. Orchard M, Tang L, Saha B, et al. Risk-sensitive particle-filtering-based prognosis framework for estimation of remaining useful life in energy storage devices. *studies in informatics and control* 2010; 19: 209–218.
28. Dong-Hua Z and Zhong YY. *Modern fault diagnosis and fault tolerant control*. Beijing: Tsinghua University Press, 2000, pp.256–267.
29. Prognostic data repository: bearing data set NSFI/UCRC center for intelligent maintenance systems, http://ti.arc.nasa.gov/tech/dash/pcoe/prognostic-data-repository/, online in (2010, accessed 22 September 2012).
30. Qiu H, Lee J, Lin J, et al. Wavelet filter-based weak signature detection method and its application on rolling element bearing prognostics. *J Sound Vib* 2006; 289: 1066–1090.
31. Roweis S and Saul L. Nonlinear dimensionality reduction by locally linear embedding. *Science* 2000; 290: 2323–2326.
32. Tian Z, Wong L and Safaei N. A neural network approach for remaining useful life prediction utilizing both failure and suspension histories. *Mech Syst Signal Process* 2010; 24: 1542–1555.
33. Tobon-Mejia DA, Medjaher K, Zerhouni N, et al. A data-driven failure prognostics method based on mixture of Gaussians hidden markov models. *IEEE Trans Reliability* 2012; 61: 491–503.
34. Saxena A, Celaya J, Saha B, et al. On applying the prognostic performance metrics. In: *Annual conference of prognostics and health management society*, PHM09 San Diego, 27 September–1 October 2009.
35. Vachtsevanos G, Lewis FL, Roemer M, et al. *Intelligent fault diagnosis and prognosis for engineering systems*. USA: Wiley, 2006.

Appendix A

Notation

b	bias term
C	regularization parameter
$\boldsymbol{D}$	training set in LS-SVR
$\boldsymbol{e}$	Gaussian white noise in the observation model
$f_{LS\text{-}SVR}(\cdot)$	m-order model trained by LS-SVR
$\boldsymbol{F}(\cdot)$	state transition function
$\mathbf{I}$	identity matrix
k	time instant
$\mathbf{K}$	kernel matrix
l^i	RUL of the i th particle
$L(\cdot)$	Lagrangian function
m	embedding dimension
M	number of training samples
n	number of particles
$N(\cdot)$	normal distribution function
N_{eff}	valid number of the sample
$\boldsymbol{p}$	input vector in predictor
p	number of step in long-term condition prediction
$\mathbf{P}$	input matrix in predictor
$\mathrm{P}_{k\|k}$	optimal estimate covariance in time k
$\mathrm{P}_{k+1\|k}$	predict covariance in time k
$q(\cdot)$	important density function
$\boldsymbol{U}$	controlling parameter
$\boldsymbol{v}$	Gaussian white noise in state evolution model
w	noise in (19)
$\boldsymbol{x}$	variable of state in (15)
$\hat{x}_{k\|k}$	optimal state estimation in time k
$\hat{x}_{k+1\|k}$	prediction in time k
x_k^i	state of the ith particle
x^*	expectation value of particles
$\boldsymbol{Y}$	output vector in LS-SVR
α	a vector of the Lagrange parameters
α_i	Lagrangian multiplier
δ	bandwidth parameter in Gaussian kernel function
∇	differential operator
ξ	error variable
$\phi(\cdot)$	kernel function
ω_{k-1}^i	weight of the ith particle at time $k-1$