

Article

A Hybrid LSSVR/HMM-Based Prognostic Approach

Zhijuan Liu^{1,2,*}, Qing Li¹, Xianhui Liu³ and Chundi Mu¹

- ¹ Department of Automation, Tsinghua University, Beijing 100084, China; E-Mails: liqing@tsinghua.edu.cn (Q.L.); muchd@mail.tsinghua.edu.cn (C.M.)
- ² Aerospace System Engineering Shanghai, Shanghai 201109, China
- ³ CAD Research Center, Tongji University, Shanghai 200092, China; E-Mail: liu_xian_hui@163.com
- * Author to whom correspondence should be addressed; E-Mail: liuzhijuan6512@163.com; Tel.: +86-150-0069-2137.

Received: 20 March 2013; in revised form: 11 April 2013 / Accepted: 19 April 2013 / Published: 26 April 2013

Abstract: In a health management system, prognostics, which is an engineering discipline that predicts a system's future health, is an important aspect yet there is currently limited research in this field. In this paper, a hybrid approach for prognostics is proposed. The approach combines the least squares support vector regression (LSSVR) with the hidden Markov model (HMM). Features extracted from sensor signals are used to train HMMs, which represent different health levels. A LSSVR algorithm is used to predict the feature trends. The LSSVR training and prediction algorithms are modified by adding new data and deleting old data and the probabilities of the predicted features for each HMM are calculated based on forward or backward algorithms. Based on these probabilities, one can determine a system's future health state and estimate the remaining useful life (RUL). To evaluate the proposed approach, a test was carried out using bearing vibration signals. Simulation results show that the LSSVR/HMM approach can forecast faults long before they occur and can predict the RUL. Therefore, the LSSVR/HMM approach is very promising in the field of prognostics.

Keywords: prognostics; least squares support vector regression; hidden Markov model; remaining useful life

1. Introduction

5543

Condition-based maintenance (CBM), which can increase system maintenance efficiency and reduce life cycle cost, is gaining popularity. Developing the capability of prognostics and remaining useful life (RUL) prediction can save great costs and improve the logistics support. Therefore, prognostics and RUL prediction can improve the functions of CBM [1,2].

At present, various health management systems are being gradually proposed and applied, such as health and usage monitoring system (HUMS) [3], integrated vehicle health management (IVHM) [4], and prognostics and health management (PHM) [5,6]. PHM, proposed in Joint Strike Fighter (JSF), represents the highest level of CBM. In these health management systems, prognostics is an important but difficult task. Currently, there is limited research in this field due to the fact that small fault symptoms and disturbances cause difficulties in carrying out studies. Therefore, this paper aims to contribute to the growing body of research in the area of prognostics technology.

Prognostic approaches can be placed into three categories: physical model-based prognostics, evolutionary or trending model-based prognostics, and experience-based prognostics [7]. The physical model-based method has the highest accuracy, but it is often complicated to model precisely and the cost is high. Evolutionary or trending-based prognostics are data-driven and feature-based methods. They are considered effective and available prognostics approaches. The third category, experience-based prognostics, relies on statistical reliability. Some popular prognostic methods of the second category are the auto-regressive moving average (ARMA) model [8], the artificial neural network (ANN) [9], the Bayesian network [10], and the support vector regression (SVR) [11,12]. Among these methods, ANN is a widely-used method. The disadvantages of ANN are that the number of hidden layers is difficult to choose and that the calculation may fall into a local minimum. SVR is a machine-learning algorithm based on Vapnik-Chervonenkis (VC) theory and the structural risk minimization principle; therefore it avoids the disadvantages of ANN [11] and is a potential method in prognostics. There are also other extended SVR methods, such as least squares support vector regression (LSSVR) [13,14]. LSSVR proposed by Suykens [15,16] is often applied to predict non-linear time-series signals. In health management, LSSVR can be used to estimate the future health deterioration and can be extended for remaining useful life (RUL) prediction. There are a series of articles about the use of LSSVR in prognosis and monitoring, such as [17,18]. Although the LSSVR algorithm can predict the signal trends, it is difficult to assess the health status of a system for each stage. To solve this problem and make health assessments more clear, in this paper a hidden Markov model (HMM) method is introduced to describe health states during component degradation. The HMM technique, initially introduced by Baum [19], is a stochastic approach based on Markov chains. HMM has strong fault classification and status classification ability, therefore, the HMM method can be used in fault detection and diagnosis [20–22], but it cannot predict the future health status.

In this paper, a hybrid fault prognosis method based on LSSVR and HMM is proposed. In this method, LSSVR is used to predict the fault features, while HMMs are used to describe health states. Fault processes can be divided into several stages and HMMs are built and trained for each stage. Suppose given an observation sequence, the probabilities of the observation for HMMs are used to determine which health state of the given observations the sequence belongs to. Combining these

HMMs with the future features predicted by LSSVR can provide an estimate of the remaining useful life (RUL) of a system.

The remainder of the paper is organized as follows: Section 2 outlines the framework of the LSSVR/HMM prognostics approach. Section 3 presents the training and estimation algorithms of HMM. Section 4 introduces the prediction algorithm based on LSSVR. Section 5 provides an application and simulation results. Section 6 presents our conclusions.

2. LSSVR/HMM Based Prognostics

2.1. The Hidden Markov Model

This aim of this section is to introduce how to use the HMM-based method to describe a fault process. The fault process of a machine has a certain time-span. To analyze the fault process, it can be divided into several health states. However, fault states are unobservable but hidden in some observable signals, such as vibration signals. The HMM-based method provides a way to detect the unobservable fault states, as HMM is a stochastic technique for classifying signals and modeling [23,24]. An HMM describes a double stochastic process. One is the Markov chain, which describes state transitions. Another is the stochastic process, which describes the relations between states and observations. Suppose the number of health states is N, which are $h_1, h_2, ..., h_N$. The last state h_N represents that the system fails. This state sequence is a Markov chain. The structure of an N-state left-to-right HMM is commonly used in fault analysis as shown in Figure 1.

Figure 1. N-state left-to-right hidden Markov model.



As shown in Figure 1, an HMM consists of a finite number of states where a_{ij} (i,j = 1,2,...,N) is the transition probability from state h_i to state h_j . Observation sequences from different conditions can be used to train different HMMs. For instance, observations from normal condition can train an HMM for normal state. An HMM can be described as:

$$\lambda = (N, M, \pi, A, B) \tag{1}$$

where *N* is the number of states, *M* is the number of observations for each state, π is the initial probability vector, $A = (a_{jj})_{N \times N}$ is the probability transition matrix which describes the transition relations between the states, $B = (b_{jk})_{N \times M}$ describes the observation probability distribution and b_{jk} is the probability of the *k*th observation in state *j*. *N* and *M* are known based on history experiences or prior knowledge. In other words, they are assumptions before HMM training. π , *A*, *B* are learned from the data. As *N* and *M* are also included in A and B, the model can be abbreviated as [23]:

As the observation sequence from a health state can train an HMM, N HMMs can be trained, which are written as λ_1 , λ_2 ,..., λ_N . λ_i represents the HMM trained based on the given observation sequence from the ith health state. Given an observation sequence $O = \{o_1, o_2, ..., o_T\}$ where T is the length of the sequence, $P(O|\lambda_i)$ is the likelihood probability of observation sequence O occurring at the condition of model λ_i , which can be used to determine whether O belongs to the health state h_i .

As mentioned above, using HMMs can describe machine fault processes. According to the likelihood probabilities of present observations, HMMs can detect and diagnose faults. However, the HMM method cannot predict future health states and the RUL, because the future observation sequence cannot be obtained. To solve this problem, a hybrid prognostics method based on LSSVR and HMM is proposed in this paper, where LSSVR provides a way to predict the future fault features.

2.2. Framework of LSSVR/HMM Prognostics

The framework of the hybrid fault prognosis scheme is described in Figure 2. In this scheme, the LSSVR algorithm is used to predict the future fault features and the HMMs are used to describe different health states.

Figure 2. Framework of the LSSVR/HMM fault prognostics scheme which contains two processes. One is HMMs training (off line) and another is fault feature and RUL prediction process (on line).



(1) Feature extraction

As most observation signals have much random noise and uncertain interferences, features should be extracted from the signals before fault diagnosis and prognosis. In this paper, the linear predictive (LP) method, as mentioned in article [25], is used to extract the features from vibration signals. In the LP method, the signal S_n can be predicted as a linear combination of the ρ previous signal, which can be written as:

$$\hat{s}_n = l_1 s_{n-1} + l_2 s_{n-2} + \dots + l_\rho s_{n-\rho} = \sum_{i=1}^{\rho} l_i s_{n-i}$$
(3)

where $l_1, l_2, ..., l_{\rho}$ are linear prediction coefficients which can be used as feature vector of the signals.

The residual errors are written as:

$$e_n = s_n - \hat{s}_n = s_n - \sum_{i=1}^{\rho} l_i s_{n-i}$$
(4)

The value of the linear prediction coefficients $l_1, l_2, ..., l_{\rho}$ can be calculated by minimizing the mean-square value of the residual errors over a signal window. The mean square value is written as:

$$\tilde{e}_{n} = \sqrt{\sum_{m} e_{n}^{2}(m)} = \sqrt{\sum_{m} |s_{n}(m) - \sum_{i=1}^{\rho} l_{i} s_{n-i}(m)|^{2}}$$
(5)

where $e_n(m) = e(n + m)$, $S_n(m) = S(n + m)$, *n* is the starting point of the signal window, n + m is the terminal point of the signal window.

Vibration signals are always non-stationary. If signals are segmented into several short-time signals, they can be considered stationary. Therefore, we can obtain short-time signals by segmenting vibration signals into small windows; and each window as a frame. To segment the vibration signals into small windows, the length of a small window should be determined. Each window has equal length. Based on the given length, the vibration signals can be divided into several small windows. And then the linear prediction coefficients can be extracted from these windows. The feature extraction is described in Figure 3.

Each window is coded into a feature vector which is written as:

$$o_t = [l_{t1}, l_{t2}, \cdots , l_{t\rho}]^{\mathrm{T}}, \quad 1 \le t \le T$$
(6)

where o_t is the feature vector of the *i*th signal window, l_{t1} , l_{t2} , ..., l_{tp} are the linear prediction coefficients of the *t*th signal window, *T* is the number of the windows. Then the observation sequence $O=\{o_1, o_2, ..., o_T\}$ is composed of these feature vectors, which will be used for HMM training or LSSVR prediction.



Figure 3. Feature extraction by segmenting the vibration signals into small windows.

(2) HMMs training

HMMs training can now be carried out based on the fault features history. In this paper, the historic fault features are experimental data. The training data used for a specific state come from the corresponding experimental conditions. The essence of this step is to estimate the HMM parameters for each state. Based on observation sequences from *N* different conditions, optimal HMMs $\lambda_1, \lambda_2, ..., \lambda_N$ are trained for corresponding health state. This step is an off-line process.

(3) Prediction of features based on LSSVR

Suppose the present time is *t*, and the feature vector after *k* time units is denoted \hat{o}_{t+k} which can be predicted based on the data before time *t* using the LSSVR algorithm. Based on the feature vector \hat{o}_{t+k} , one can predict the system health state at the future time t + k.

(4) Log-likelihood calculation

Calculating likelihood probabilities $P(\hat{o}_{t+k} | \lambda_i)$ according to forward or backward algorithms [26,27] and comparing all of these probabilities, one can find the maximum probability:

$$i = index(max(P_1, P_2, ..., P_N))$$
 (7)

where $P_1, P_2, ..., P_N$ are the likelihood probabilities for the corresponding HMMs.

The corresponding state h_i is considered the health state of the future time t + k. Therefore, this method can be used to predict future health states. If k = 0, the health state is the present state.

(5) RUL prediction.

As shown in Figure 2, the fault prognosis should run until the probability for the last state model is higher than for all other state models. Then the probability $P(\hat{o}_{t+k} | \lambda_N)$ for the last state indicates that the system will fail at the future time t + k. Therefore, the RUL is *k* time units.

3. HMM Training

HMM training is to obtain the optimal model and the training process needs to estimate the model parameters. An HMM is described in Equation (2). When the observations are discrete, they can be represented by distributed probabilities. However, the observations are always continuous in an actual

system. In this situation, a mixture of several Gaussian distributions is used to describe observation probabilities which are written as [26]:

$$b_{j}(o_{t}) = \sum_{l=1}^{Q} c_{jl} N(o_{t} \mid \mu_{jl}, \Sigma_{jl})$$
(8)

$$N(o_t \mid \mu_{jl}, \Sigma_{jl}) = \frac{1}{(2\pi)^{d/2} \mid \Sigma_{jl} \mid^{1/2}} e^{\frac{1}{2}(x - \mu_{jl})^{\mathrm{T}} \Sigma_{jl}^{-1}(x - \mu_{jl})}$$
(9)

where $b_j(o_t)$ is the probability of observation o_t at the condition of state h_j , o_t is the feature vector of the *t*th window, *Q* is the number of the mixture Gaussian components, c_{jl} is the weight of the *l*th mixture component in state h_j , and μ_{jl} and Σ_{jl} are the mean and covariance of the Gaussian density.

Then an HMM is re-written as:

$$\lambda = (\pi, A, \mu_{il}, \Sigma_{il}, c_{il}) \tag{10}$$

3.1. Forward-Backward Algorithm

Given an observation sequence $O = \{o_1, o_2, \dots, o_T\}$ and a model λ , how to calculate the probability $P(O|\lambda)$ is a basic and important problem. Both forward and backward algorithms can solve this problem.

In a forward algorithm, define the forward variables as $\eta_i(t) = P(o_1, o_2, ..., o_t, q_t = h_i | \lambda), 1 \le t \le T$, which is the probability of generating $o_1, o_2, ..., o_t$ and ending in state h_i . q_t represents the state attime t.

The recursive estimation of the forward is as follows [26,27]:

(a) Initialization:

$$\eta_i(1) = \pi_i b_i(o_1) \tag{11}$$

(b) Forward recursion (For t=1,2,...,T-1, $1 \le j \le N$):

$$\eta_{j}(t+1) = \left[\sum_{i=1}^{N} \eta_{i}(t) a_{ij}\right] b_{j}(o_{t+1})$$
(12)

(c) End:

$$P(O \mid \lambda) = \sum_{i=1}^{N} \eta_i(T)$$
(13)

In a backward algorithm, we define the backward variables as $\beta_i(t) = P(o_{t+1}, o_{t+2}, ..., o_T | q_t = h_i, \lambda)$, $1 \le t \le T$, which is the probability of generating $o_{t+1}, o_{t+2}, ..., o_T$ and ending in state h_i .

The recursive estimation of the backward is as follows [26,27]:

(a) Initialization $(1 \le i \le N)$:

$$\beta_T(i) = 1 \tag{14}$$

(b) Backward recursion (For $t = T-1, T-2, ..., 1, 1 \le i \le N$):

$$\beta_t(i) = \sum_{j=1}^N a_{ij} b_j(o_{t+1}) \beta_{t+1}(j)$$
(15)

(c) End:

$$P(O \mid \lambda) = \sum_{i=1}^{N} \beta_1(i)$$
(16)

3.2. Re-Estimation of the HMM Parameters

The goal of HMM training is to find the model $\lambda^* = \underset{\lambda}{\operatorname{argmax}} P(O|\lambda)$. That means the parameters of each HMM should be re-estimated until the convergence error achieves requirements. The Baum-Welch algorithm [26] provides a solution to this problem.

Given an observation sequence O, define $\gamma_t(i) = P(q_t = h_i | O, \lambda)$, which is the probability of that the state at time *t* is h_i . $\gamma_t(i)$ can be calculated as:

$$\gamma_t(i) = \frac{\eta_i(t)\beta_t(i)}{\sum_{j=1}^N \alpha_j(t)\beta_t(j)}$$
(17)

Define $\gamma_t(i, l)$ as the probability of the lth mixture Gaussian component when the state at time *t* is h_i . $\gamma_t(i, l)$ can be calculated as:

$$\gamma_t(i,l) = \gamma_t(i) \frac{c_{il} N(o_t \mid \mu_{il}, \Sigma_{il})}{\sum_{m=1}^{Q} c_{im} N(o_t \mid \mu_{im}, \Sigma_{im})}$$
(18)

 $\xi_t(i, j) = P(q_t = h_i, q_{t+1} = h_j | O, \lambda)$ is defined as the probability of that the state at time *t* is h_i and the state at time t + 1 is h_j . $\xi_t(i, j)$ can be calculated as:

$$\xi_t(i,j) = \alpha_t(i)\eta_{ij}b_j(o_{t+1})\beta_{t+1}(j)/P(O|\lambda)$$
(19)

If there are K observation sequences and the length of each sequence is T, the parameters of a HMM can be estimated as follows [26,27]:

$$\pi_i = \frac{\sum_{k=1}^{K} \gamma_1^k(i)}{K}$$
(20)

$$a_{ij} = \frac{\sum_{k=1}^{K} \sum_{j=1}^{T} \zeta_{t}^{k}(i,j)}{\sum_{k=1}^{K} \sum_{j=1}^{T} \gamma_{t}^{k}(i)}$$
(21)

$$c_{il} = \frac{\sum_{k=1}^{K} \sum_{t=1}^{T} \gamma_{t}^{k}(i,l)}{\sum_{k=1}^{K} \sum_{t=1}^{T} \gamma_{t}^{k}(i)}$$
(22)

$$\mu_{ll} = \frac{\sum_{k=1}^{K} \sum_{t=1}^{T} \gamma_{t}^{k}(i,l) o_{t}^{k}}{\sum_{k=1}^{K} \sum_{t=1}^{T} \gamma_{t}^{k}(i)}$$
(23)

$$\Sigma_{il} = \frac{\sum_{k=1}^{K} \sum_{t=1}^{T} \gamma_{t}^{k}(i,l) (o_{t}^{k} - \mu_{l}) (o_{t}^{k} - \mu_{l})^{\mathrm{T}}}{\sum_{k=1}^{K} \sum_{t=1}^{T} \gamma_{t}^{k}(i)}$$
(24)

According to Equations (20)–(24), the parameters of the HMMs are re-estimated. Finally, optimal models of the health states can be obtained as shown in Figure 2. Given an observation, the likelihood probabilities for each state can be calculated according to Equations (13) or (16). In the forward-backward

algorithm, $\eta_i(t)$ and $\beta_i(i)$ will tend to be zero as time grows. To avoid this problem, logarithmic values of probabilities are usually used in the actual calculations.

4. Feature Trend Prediction Based on LSSVR

4.1. LSSVR Algorithm

The idea of SVR is to map non-linear problems to linear problems in a high dimension space. LSSVR is an improved algorithm based on SVR, and it uses a least square loss function instead of the ε -insensitive loss function [16,28]. Compared with SVR, the calculation procedure of LSSVR has higher efficiency.

Considering the sample data as $\{(x_1,y_1), ..., (x_b,y_i), ..., (x_D,y_D)\}$, where x_i is the input vector, and y_i is the objective value. The optimization equation of the LSSVR is described as [13,14]:

$$\min_{\boldsymbol{\omega},\boldsymbol{\kappa},\boldsymbol{e}} J(\boldsymbol{\omega},\boldsymbol{e}) = \frac{1}{2} \|\boldsymbol{\omega}\|^2 + \frac{1}{2} C \sum_{i=1}^{D} e_i^2$$

s.t. $y_i = \boldsymbol{\omega}^{\mathrm{T}} \boldsymbol{\phi}(x_i) + \boldsymbol{\kappa} + e_i \quad 1 \le i \le D$ (25)

where ω is weight vector, e_i is error variable, C is a regularization parameter used to control the punishment degree of samples beyond the error, the function of $\Phi(*)$ is to allow LSSVR to work in a linear space, and κ is the bias term which is an unknown and needs to be estimated.

The Lagrangian is written as:

$$L(\boldsymbol{\omega}, \boldsymbol{e}, \boldsymbol{\kappa}, \boldsymbol{\alpha}) = J(\boldsymbol{\omega}, \boldsymbol{e}) - \sum_{i=1}^{D} \alpha_i (\boldsymbol{\omega}^{\mathrm{T}} \boldsymbol{\phi}(\boldsymbol{x}_i) + \boldsymbol{\kappa} + \boldsymbol{e}_i - \boldsymbol{y}_i)$$
(26)

where α_i is Lagrange multipliers.

According to Karush-Kuhn-Tucker (KKT) optimization conditions, the optimality conditions are given as [13,14]:

$$\begin{cases} \frac{\partial L}{\partial \boldsymbol{\omega}} = \boldsymbol{\omega} - \sum_{i=1}^{D} \alpha_{i} \phi(x_{i}) = 0 \\ \frac{\partial L}{\partial \kappa} = -\sum_{i=1}^{D} \alpha_{i} = 0 \\ \frac{\partial L}{\partial e_{i}} = Ce_{i} - \alpha_{i} = 0 \\ \frac{\partial L}{\partial \alpha_{i}} = -(\boldsymbol{\omega}^{\mathrm{T}} \phi(x_{i}) + \kappa + e_{i} - y_{i}) = 0 \end{cases}$$

$$(27)$$

By solving Equation (27), the parameters α_i and κ can be obtained. Then the LSSVR model for estimation is written as:

$$y(x) = \sum_{i=1}^{D} \alpha_i K(\mathbf{x}_i \cdot \mathbf{x}) + \kappa$$
(28)

where $K(x_i, x)$ is the kernel function which satisfies the Mercer condition:

$$K(x_i \cdot x_k) = \phi(x_i)^{\mathrm{T}} \phi(x_k)$$
⁽²⁹⁾

Frequently used kernel functions are the polynomial, sigmoid and radial basis kernel (RBF) functions. This paper applies RBF because it can classify multi-dimensional data. The RBF kernel function is given as:

$$K(x \cdot x_k) = \exp(-\frac{\|x - x_k\|^2}{\sigma^2})$$
(30)

In fault prognosis, time series are the extracted feature data. Suppose the front r data have been obtained, then the object is to predict the future data using the front r data. The input vector and the output value can be constructed as:

$$X = \begin{bmatrix} x_1 & x_2 & \cdots & x_p \\ x_2 & x_3 & \cdots & x_{p+1} \\ \vdots & \vdots & \vdots & \vdots \\ x_{r-p} & x_{r-p+1} & \cdots & x_{r-1} \end{bmatrix}, \quad Y = \begin{bmatrix} x_{p+1} \\ x_{p+2} \\ \vdots \\ x_r \end{bmatrix}$$

where *p* is the number of embedded dimensions. According to above LSSVR method, training this sample data can estimate the parameters α_i and κ . Then the prediction function can be written as:

$$\hat{x}_{r+k} = \sum_{i=1}^{D} \alpha_i K(\boldsymbol{x}_i \cdot \tilde{\boldsymbol{x}}) + \kappa$$
(31)

where k is the prediction steps, x_i is the *i*th support vector, and \tilde{x} is $\tilde{x} = \{x_{r+k-\nu_1}, \dots, \hat{x}_{r+1}, \dots, \hat{x}_{r+k-1}\}$.

As mentioned above, LSSVR can predict future fault features. However, it is difficult to describe and assess health states. Combining LSSVR with HMM, it can realize health state prediction and RUL prediction. The feature vectors predicted based on the LSSVR are used as observations in each HMM to calculate the log-likelihood probabilities as shown in Equation (13) or Equation (16). The model with the highest likelihood probability represents the system health state at the corresponding future time.

4.2. Algorithm Improvement

As analyzed above, two parameters C and σ in the LSSVR algorithm need to be optimized. Selecting superior parameters can improve the training speed and the performance of LSSVR. Genetic algorithms (GA) [13] and particle swarm algorithms [14] are two optimization methods used to determine the LSSVR parameters. In this paper, particle swarm optimization (PSO) is used. PSO realizes optimization by searching the partial and global optimal solution in a particle swarm. Define the fitness function of the *i*th particle as:

$$J_i(C,\sigma^2) = \frac{1}{\sum_{k=1}^{N} (y_k - f(x_k))^2 + eps}$$
(32)

where *eps* is a small value to avoid the denominator to be zero.

The particle positions and velocities are written as $X_i=(x_{i1}, x_{i2}, ..., x_{iD})^T$ and $V_i=(v_{i1}, v_{i2}, ..., v_{iD})^T$ (*i*=1,2, ..., *m*). *m* is the number of the particles. The partial best position is written as P_i and the global best position is written as P_g . Then the velocity and position updating can be written as:

$$v_{id}(k+1) = Iv_{id}(k) + c_1r_1(k)(P_{id}(k) - x_{id}(k)) + c_2r_2(k)(P_{gd}(k) - x_{id}(k))$$
(33)

$$x_{id}(k+1) = x_{id}(k) + v_{id}(k+1)$$
(34)

where $v_{id}(k)$ and $x_{id}(k)$ are the *d*th dimension velocity and position of the *i*th partial in the *k*th iteration, *I* is an inertia factor, c_1 and c_2 are study factor, P_{id} and P_{gd} are the *d*th dimension partial and global best position in the *k*th iteration, r_1 and r_2 are random during [0,1]. The optimization process is described as Figure 4, where the initial positions and velocities are given randomly among the range of the particle swarm.

Figure 4. The parameters optimization process of the LSSVR based the PSO algorithm.



In Figure 4, the partial and global best position searching process is as follows:

if
$$J_i > J_{best}$$
, then $P_i = X_i$, $J_{best} = J_i$
if $J_i > G_{best}$, then $P_g = X_i$, $G_{best} = J_i$ (35)

where J_{best} is the partial best fitness and G_{best} is the global best fitness.

The end condition is meeting the accuracy requirements or arriving at the maximum iterations. When meeting the end condition, the best position is considered as the optimal parameters used in LSSVR.

As time goes by, new observations are added which may include fault information and can improve prediction accuracy. However, the training data will become large with time growth. If the number of sample data is too large, it will affect training efficiency. To solve this problem, some unimportant old data, such as non-support vector samples or far history samples, are selected for deletion in this paper. These data will not affect the training accuracy seriously. By adding new data and deleting some old data, the training samples sequence will be renewed online with time growth. The new training sample can be written as:

$$X(t+1) = X(t) + X_{new}(t+1) - X_{del}(t)$$
(36)

where X(t) is the training sample at time t, $X_{new}(t+1)$ is the added new sample at time t + 1, $X_{del}(t)$ is the deleted sample chosen from X(t).

The RUL prediction process based on the improved LSSVR algorithm is summarized in Figure 5.

Figure 5. RUL prediction process based on the improved LSSVR with data renewed online.



As shown in Figure 5, at each time *t*, RUL equals to the future prediction time whose corresponding state is the last health state. The prediction accuracy of RUL increases as time increases because new observations are added and some unimportant old data are deleted, which makes the feature trend predictions more accurate.

5. Simulations and Results

To evaluate the performance of the LSSVR/HMM prognostic approach, it was tested using experimental bearing vibration data from the Case Western Reserve University Bearing Data Center [29].

The bearing test stand consists of a 2 hp motor, a torque transducer/encoder, a dynamometer, and control electronics. The test bearings support the motor shaft. The details of the test stand are shown in the website [29]. The actual test conditions of the motor as well as the bearing fault status have been carefully documented for each experiment. Different levels of faults in diameter were collected separately for the inner raceway, rolling element (*i.e.*, ball) and outer raceway. Faulted bearings were reinstalled into the test motor and vibration data was recorded.

In our simulation, the inner race vibrations signals are used. Its fault diameter includes the following levels: 0.007 inches, 0.014 inches, and 0.021 inches. The data was sampled at 12 kHz. Added to the normal states, fault process can be divided into four health states. The bearing data center provides vibration data from different health states. Data from different conditions are discrete, which can be used to train HMMs for the corresponding health status.

5.1. Feature Extraction

Vibration data are segmented into several windows and extracted feature vectors as mentioned in Section 2.2 and Figure 3. In the simulation, 256 sample data were chosen as a frame. Based on the linear predictive method, signals in each window can be predicted as a linear combination of its ρ previous signal, which is shown in Equation (3). In the simulation, we take ρ as 8. Then the feature vector of each signal window is $o_t = [l_{t1}, l_{t2}, ..., l_{t8}]^T$ and the number of the feature dimensions is 8. Take the first dimension as an example; the features of the four health states are shown in Figure 6. As shown in Figure 6, the extracted feature for each health state is different. This demonstrates that these features can be used to distinguish different fault states.





5.2. HMMs Training and Fault Diagnosis

Suppose three Gaussian distributions are used to describe observation probabilities for each state. Give the initial HMM parameters as follows:

$$\pi = [1, 0, 0, 0]^{1}$$

$$A = \begin{bmatrix} 0.5 & 0.5 & 0 & 0 \\ 0 & 0.5 & 0.5 & 0 \\ 0 & 0 & 0.5 & 0.5 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Generally speaking, the influence of the initial values of π and A is not large. But the initial values of μ_{jl} , U_{jl} , and c_{jl} may affect the training result. They can be calculated using K-means method based on the given samples, which are not discussed here in detail. The convergence error is given as 0.00001.

To improve the training accuracy, several observation sequences are used to train HMM for a health state. The log-likelihood probabilities of HMMs training are shown in Figure 7.

Figure 7. Log-likelihoods of HMMs during training for the four states.



The simulation results shown in Figure 7 illustrate that the log-likelihoods tend to stabilize with the growth of the number of epochs. All of the training epochs are less than 40. These trained HMMs can be used to diagnose faults. In this paper, two vibration data sequences are specimens for fault diagnosis testing, where one is fault-free data and another is data with a fault diameter of 0.007 inches. The log-likelihood probabilities of the test vibration data with fault-free are shown in Figure 8, while the log-likelihood probabilities of the data with fault diameter of 0.007 inches are shown in Figure 9.



Figure 8. Log-likelihoods for the four state HMMs of the fault-free test data.

As shown in Figure 8, the HMM with the highest probability is health state 1, which represents fault-free state. That is to say the test data are fault-free. The result is consistent with the given data. Figure 9 shows that the HMM with the highest probability is health state 2, which represents the fault diameter of 0.007 inches. Therefore, the diagnosis result is consistent with the given data. According to the above two tests, fault diagnosis based on HMMs is efficient.

Figure 9. Log-likelihoods for the four state HMMs of the test data.



5.3. Features and RUL Prediction

Collecting lifetime data is a difficult work and the bearing data center does not provide this lifetime data. Therefore, we construct a lif time data sequence by taking some vibration data separately from each health states. Then the constructed lifetime data sequence is used to test for features prediction and fault prognostics. After frame segmentation and feature extraction, the length of the feature sequence is 71. We take the front 50 feature vectors as LSSVR training data, and the back 21 as unknown test data. With these parameters, the present time is t = 50 and the future prediction time is k = 21. We take the first dimension of the feature vectors as an example, and the prediction result is shown in Figure 10.

Figure 10. The prediction result of the first dimension of the feature vectors based on the LSSVR.



In Figure 10, the data before t = 50 are regression estimation results and the data after t = 50 are prediction results according to Equation (28). The simulation results indicate that the prediction feature trend is essentially consistent with the actual trend. The simulation results can therefore be used to make a fault prognosis.

Based on the trained four HMMs, the log-likelihood probabilities of the life time data for each state are shown in Figure 11.

The state with the highest log-likelihood is the health state of the corresponding time. As shown in Figure 11, the system goes from health state 1 to health state 4 during the lifetime. The system is in a normal state before t = 38. During the time from t = 39 to t = 55, the system is in health state 2 which means the fault diameter is at 0.007 inches. During the time from t = 56 to t = 65, the system is in health state 3, which means the fault diameter is at 0.014 inches. After t = 66, the system is in health state 4. That indicates that the system fails after t = 66. RUL values can be predicted at each time. The RUL prediction results at different times are shown in Table 1.



Figure 11. Log-likelihoods for the four state HMMs of the test life time data.

Table 1. Remaining useful life prediction results.

Present time	50	51	52	53	54	55	56	57	58	59	60
Actual RUL	16	15	14	13	12	11	10	9	8	7	6
Predicted RUL	9	10	10	11	9	10	9	8	7	6	5
Prediction errors	7	5	4	2	3	1	1	1	1	1	1

In Table 1, the actual value of RUL is the time between the present time and the failure time t = 66. The predicted value of RUL is calculated based on the proposed LSSVR/HMM method. As shown in Table 1, the predicted RUL is less than the actual RUL. Therefore, the LSSVR/HMM hybrid prognosis method can forecast faults long before the actual fault occurs. The RUL prediction errors reduce with increasing time and it tends to be 1 time unit in the simulation, but in real situations, the prediction results may be not so good because the test lifetime data used in the simulation is perfect and shorter than an actual lifetime data sequence.

6. Conclusions

Fault prognosis is an essential and difficult technology in health management and CBM. In this paper, a hybrid prognostics approach which integrates the HMM method and LSSVR is presented. In the proposed LSSVR/HMM approach, the fault process is divided into several health states modeled by HMMs. These HMMs are trained based on history sample data. The lot-likelihood probabilities of the HMMs are used to classify data and determine the health state. The state with the highest log-likelihood is the corresponding health state of given data. LSSVR is used to predict the feature trend. Based on the predicted features and trained HMMs, the future health state can be predicted. LSSVR trains and predicts RULs online until the highest probability of the predicted feature vector is at the last health state. Then the RUL is equal to the prediction time.

Testing of the proposed LSSVR/HMM prognostics approach was carried out using bearing vibration data. The simulation results showed that the novel approach is efficient in fault diagnosis and prognostics and can also predict the RUL. Furthermore, the results showed that the RUL prediction accuracy increases with the passing of time. Future work will focus on how to improve the prediction accuracy and expand the applications.

Conflict of Interest

The authors declare no conflict of interest.

References

- Cheng, S.; Azarian, M.; Pecht, M. Sensor system for prognostics and health management. *Sensors* 2010, 10, 5774–5797.
- 2. Pecht, M. Progn and Health Management of Electron; Wiley-Interscience: New York, NY, USA, 2008.
- Dickson, B.; Cronkhite J.; Bielefeld S.; Killian, L.; Hayden, R. Feasibility Study of a Rotorcraft Health and Usage Monitoring System (HUMS): Usage and Structural Life Monitoring Evaluation; NASA Contractor Report 198447; DOT/FAA/AR-95/9, ARL-CR-290, February 1996.
- Baroth E.; Powers, W. T.; Fox, J.; Prosser, B.; Pallix, J.; Schweikard, K.; Zakrajsek, J. IVHM (Integrated Vehicle Health Management) Techniques for Future Space Vehicles. In Proceedings of the AIAA 37th Joint Propulsion Conference and Exhibit, Salt Lake City, UT, USA, 8–11 July 2001.
- 5. Ferrell, B.L. Air Vehicle Prognostics and Health Management. In Proceedings of the IEEE Conference on Aerospace, Big Sky, MT, USA, 18–25 March 2000; pp. 145–146.
- 6. Hess, A.; Calvello, G.; Dabney, T. *PHM* a key Enabler for the JSF Autonomic Logistics Support Concept. In Proceedings of the IEEE Conference on Aerospace, Big Sky, MT, USA, 6–13 March 2004; pp. 3543–3550.
- Roemer, M.; Dzakowic, J.; Orsagh, R.; Byington, C.; Vachtsevanos, G. An Overview of Selected Prognostic Technologies with Reference to an Integrated PHM Architecture. In Proceedings of the First International Forum on Integrated System Health Engineering and Management in Aerospace, Big Sky, UT, USA, 2005; pp. 3941–3947.
- 8. Box, G.E.P.; Jenkins, G.M. *Time Series Anal: Forecast and Control*, 3rd ed.; Holden-Day: San Francisco, CA, USA, 1994.
- Xue, G.X.; Xiao, L.C.; Bie, M.H.; Lu, S.W. Fault Prediction of Boilers with Fuzzy Mathematics and RBF Neural Network. In Proceeding of the International Conference on Communications, Circuits and Systems, Jiangsu, China, 27–30 May 2005; pp. 1012–1016.
- Ferreiro, S.; Arnaiz, A.; Sierra, B.; Irigoien, I. A bayesian network model integrated in a prognostics and health management system for aircraft line maintenance. J. Aerosp. Eng. 2011, 225, 886–901.
- 11. Smola, A.J.; Schölkopf, B. A tutorial on support vector regression. Stat. Comput. 2004, 14, 199-222.
- 12. Zhang, J.F.; Hu, S.S. Nonlinear time series fault prediction based on clustering and support vector machines (in Chinese). *Control Theory Appl.* **2007**, *24*, 64–68.
- 13. Qu, J.; Zuo, M.J. An LSSVR-based algorithm for online system condition prognostics. *Expert Syst. Appl.* **2012**, *39*, 6089–6102.

- Liu, S.; Jia, S.; Li, B.; Li, G.Y. Investigation of Steering Dynamics Ship Model Identification Based on PSO-LSSVR. In Proceedings of the 2nd International Symposium on Systems and Control in Aerospace and Astronautics, Shenzhen, China, 10–12 December 2008; pp. 1–6.
- 15. Suykens, J.A.K; Vandewalle, J. Least squares support vector machine classifiers. *Neural Process. Lett.* **1999**, *9*, 293–300.
- Suykens, J.A.K; Lukas L.; Vandewalle, J. Spare Least Squares Support Vector Machine Classifiers. In Proceeding of the European Symposium on Artificial Neural Networks, Bruges, Belgium, 26–28 April 2000; pp. 37–42.
- Khawaja, T.; Vachtsevanos, G. A Novel Architecture for On-line Failure Pognosis Using Probabilistic Least Squares Support Vector Regression Mmachines. In Proceeding of Annual Conference of the Prognostics and Health Management Society, San Diego, CA, USA, 27 September–12 October 009; pp. 1–8.
- 18. Ge, Z.G.; Song Z.H. Online monitoring of nonlinear multiple mode processes based on adaptive local model approach. *Control Eng. Pract.* **2008**, *16*, 1427–1437.
- 19. Baum, L.E. Petrie, T. Statistical inference for probabilistic functions of finite state markov chain. *Ann. Math. Statist.* **1966**, *37*, 1554–1563.
- Atlas, L.; Ostendorf, M.; Bernard, G.D. Hidden Markov Models for Monitoring Machining Tool-Wear. In Proceeding of 2000 IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP'00, Istanbul, Turkey, 5–9 June 2000; pp. 3887–3890.
- Lee, J.M.; Kim, S.J.; Hwang, Y.; Song C.S. Diagnosis of mechanical fault signals using continuous hidden markov model. *J. Sound Vib.* 2004, 27, 1065–1080.
- Ocak, H.; Loparo, K.A. A New Bearing Fault Detection and Diagnosis Scheme Based on Hidden Markov Modeling of Vibration Signals. In Proceeding of the International Conference on Acoustics, Speech and Signal Processing, Cleveland, OH, USA, 7–11 May 2001; pp. 3141–3144.
- 23. Ertunc, H.M.; Loparo, K.A. A decision fusion algorithm for tool wear condition monitoring in drilling. *Int. J. Machine Tools Manuf.* **2001**, *41*, 1347–1362.
- 24. Wang, L.; Mehrabi, M.G.; Kannatey-Asibu, E. Hidden markov model-based tool wear monitoring in machining. *ASME J. Manuf. Sci. Eng.* **2002**, *124*, 651–658.
- 25. Ocak, H.; Loparo, K.A. HMM-based fault detection and diagnosis scheme for rolling element bearings. *J. Vib. Acoust.* **2005**, *127*, 299–306.
- 26. Blimes, J.A. A Gentle Tutor. of the EM Algorithm and Its Appl. to Parameter Estim. for Gaussian Mixture and Hidden Markov Models; Report; University of Berkeley: Berkeley, CA, USA, 1998.
- 27. Baruah, P; Chinnam, R.B. HMMs for diagnostics and prognostics in machining processes. *Int. J. Production Res.* **2005**, *43*, 1275–1293.
- Suykens, J.A.K.; Brabanter, J.D.; Lukas, L.; Vandewalle, J. Weighted least squares support vector machines: Robustness and sparse approximation. *Neurocomputing* 2002, 48, 85–105.
- 29. The Bearing Data Center, the Case Western Reverse University. Available online: http://www.eecs.case.edu/laboratory/bearing/welcom_overview.htm (accessed on 20 March 2012).

 \bigcirc 2013 by the authors; licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution license (http://creativecommons.org/licenses/by/3.0/).