

Research Article

Fault Prediction for Nonlinear System Using Sliding ARMA Combined with Online LS-SVR

Shengchao Su,^{1,2} Wei Zhang,¹ and Shuguang Zhao²

¹Laboratory of Intelligent Control and Robotics, Shanghai University of Engineering Science, Shanghai 201620, China

²College of Information Science and Technology, Donghua University, Shanghai 201620, China

Correspondence should be addressed to Shengchao Su; shengchao.su@outlook.com

Received 3 April 2014; Accepted 7 June 2014; Published 16 July 2014

Academic Editor: Jingjing Zhou

Copyright © 2014 Shengchao Su et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

A robust online fault prediction method which combines sliding autoregressive moving average (ARMA) modeling with online least squares support vector regression (LS-SVR) compensation is presented for unknown nonlinear system. At first, we design an online LS-SVR algorithm for nonlinear time series prediction. Based on this, a combined time series prediction method is developed for nonlinear system prediction. The sliding ARMA model is used to approximate the nonlinear time series; meanwhile, the online LS-SVR is added to compensate for the nonlinear modeling error with external disturbance. As a result, the one-step-ahead prediction of the nonlinear time series is achieved and it can be extended to n -step-ahead prediction. The result of the n -step-ahead prediction is then used to judge the fault based on an abnormality estimation algorithm only using normal data of system. Accordingly, the online fault prediction is implemented with less amount of calculation. Finally, the proposed method is applied to fault prediction of model-unknown fighter F-16. The experimental results show that the method can predict the fault of nonlinear system not only accurately but also quickly.

1. Introduction

Fault prediction mainly deals with the fault that will happen according to the past and current states of the systems, which can avoid large calamity. With the further requirements of system reliability and security, fault prediction has attracted considerable attention in aircrafts, weapons, and almost all the industrial applications [1].

Fault prediction is regarded as a challenging task of fault diagnosis. Now little headway has been made in fault prediction and there are only a few results about it. The main methods in this field can be classified into two categories, namely, model-based methods and data-driven methods. The methods based on system model require the establishment of exact system models, filtering of the measure data, and estimation of the future state. The basic idea is to judge the fault according to the prediction output of the model and make a decision about the future state of system. A variety of methods such as the extended Kalman filtering (EKF) studied in [2–4], strong tracking filtering studied in [5, 6], and particle filtering studied in [7, 8] are the typical representative in this

category. As a premise, the system model must be known beforehand and accurate for these methods to be highly effective. Unfortunately, nonlinearity and uncertainty widely exist in the system model. For some complex nonlinear systems, accurate models cannot be obtained. This can easily degrade the estimation of output and cause either missed detections or false alarms [9].

Owing to increased automation, faster sampling rate, and advances in computing power, large amount of data is available online. Consequently, the researchers pay much attention to the data-driven methods [10]. This category of methods based on the sensor or historical data can avoid the dependence on the model of system, so that more and more efforts are devoted to the research on the methods. At present, most of data-driven fault prediction methods are associated with time series [11]. One of the common methods is to apply classical time series analysis and prediction theory to the field of fault prediction. Ho and Xie [12] established autoregressive integrated moving average (ARIMA) model to predict the time of next fault occurrence. Li and Kang [13] forecasted failure rate with autoregressive moving average (ARMA)

model. Zhao et al. [14] developed an ARMA model to forecast the equipment fault. The methods based on classical time series prediction theory are mature and have achieved certain success in practical application, especially in linear system. Nevertheless, this kind of methods uses a linear statistical model to fit the data sequence. Therefore, in essence, they are not suitable for the prediction of nonlinear system. Moreover, these methods are also affected by the error of modeling, parameter perturbation, and outside disturbance. So the robustness is unsatisfactory [15]. As another kind of data-driven methods, intelligent fault prediction approaches such as neural networks, support vector regressions (SVRs), and other computational intelligence methods have elicited considerable research interests in the last decade. Neural network-based methods have increasingly attracted attention because of their robust implementation, good performance in learning arbitrarily, and excellent ability for nonlinear mappings. The methods combining time series forecasting and neural network have been especially studied and applied recently. A large number of publications are noticeable in fault prediction [16–22]. However, the application of neural network suffers from a number of drawbacks. One of the drawbacks is that the structure of neural networks is confirmed difficultly and empirically. Different people may design different network and choose different parameters for training and testing. Therefore, they may get different results on the same subject. Another significant drawback is that the performance of network is limited by the number and distributed situation of the chosen sample. In addition, neural network is usually a network without any constraint conditions. If there are any constraints on inputs and/or outputs, it will be very hard for the neural network to be trained to meet such constraints [9].

As a novel machine learning method, SVR which is proposed by Vapnik based on statistical learning theory adopts structural risk minimization principle instead of empirical risk minimization principle used in neural network, so that it has simpler structure and better generalization ability with small sample [23]. Besides, SVR has the advantages of the global optimum, handling high dimensional nonlinear data efficiently, and so on. For these reasons, it has been an effective method for modeling time series and widely applied in nonlinear time series prediction [24, 25]. In recent years, SVR has been also used successfully in the field of fault prediction [26–30]. However, some problems arise from the application of SVR techniques. The training of SVR must solve the problem of convex quadratic programming. Although the obtained solution is an optimal solution, the more the sample data is, the slower the computing speed will be and the bigger the spaces it takes, because the complexity of the algorithm relies on the number of the sample data.

Least squares support vector regression (LS-SVR) is the development of standard SVR. It can convert solving the problem of convex quadratic programming into solving linear equation group [31, 32], which greatly reduces the computational complexity of SVR. In other words, LS-SVR is computationally more efficient than standard SVR. Nevertheless, LS-SVR is more sensitive to outliers than standard SVR, which can result in less robustness. For some complex

nonlinear and nonstationary time series, it is hard to achieve expected prediction result with single LS-SVR. Furthermore, the training of traditional LS-SVR is a batch learning process that will affect the real-time fault prediction because of time-consuming and heavy computation.

The purpose of this paper is to develop an underlying approach for real-time fault prediction of unknown nonlinear systems. Firstly, an online LS-SVR algorithm is derived for nonlinear time series prediction. On this basis, a combined online prediction method is developed for nonlinear system prediction. The sliding ARMA model is used to approximate the nonlinear time series and the LS-SVR is added to online compensate for the nonlinear modelling error with external disturbance. The combined method is then used to predict the nonlinear system. Based on the forecasting results of outputs, the fault prediction can be achieved using a density function estimation method only with normal data of system.

The remainder of this paper is organized as follows. An online LS-SVR algorithm is designed for nonlinear time series prediction in Section 2. In Section 3, the sliding ARMA modeling combined with online LS-SVR compensation is discussed and an effective prediction method is developed for nonlinear system prediction. In Section 4, an abnormality estimation method is used to predict the fault based on the prediction results of system. In Section 5, a simulation example is provided to illustrate the implementation procedures and prediction performance of the proposed method. Finally, conclusions and future work are given in Section 6.

2. Time Series Prediction Based on Online LS-SVR

Compared with standard SVR, LS-SVR can greatly reduce the computational complexity, so that it has drawn more attention in practical application. However, traditional LS-SVR is off-line learning, so the real time is poor. In this section, we briefly introduce the basic concepts of LS-SVR [33] and design an online LS-SVR algorithm for nonlinear time series prediction algorithm.

2.1. A Brief Introduction to LS-SVR. Given a time series training set $T = \{(x_i, y_i), i = 1, 2, \dots, l\}$, where $x_i \in R^D$ and $y \in R$, a linear regression function with regard to W and $\Phi(x)$ is constructed as

$$f(x) = W^T \Phi(x) + b, \quad (1)$$

where W is a vector in a huge dimensional feature space F ; it determines the margin of support vectors. $\Phi(x)$ is a nonlinear function which maps $x \in R^D$ to a vector in F . The W and b in formula (1) are obtained by solving an optimization problem:

$$\min_{w,b,e} J(W, e) = \frac{1}{2} W^T W + \frac{1}{2} \gamma \sum_{i=1}^l e_i^2 \quad (2)$$

$$\text{s.t.: } y_i = W^T \phi(x_i) + b + e_i, \quad i = 1, 2, \dots, l.$$

In formula (2), the regularization parameter γ determines the fitting error and smoothness and the nonnegative error

variable e_i is used to construct a soft margin hyperplane. This optimization problem including the constraints can be solved by the Lagrange function as follows:

$$L(W, b, e, a) = J(W, e) - \sum_{i=1}^l a_i \{W^T \phi(x_i) + b + e_i - y_i\}, \quad (3)$$

where a_i is the Lagrange coefficient. According to Karush-Kuhn-Tucker (KKT) optimization condition, a linear equation group below can be obtained by expurgating W, e :

$$\begin{bmatrix} 0 & e1^T \\ e1 & Q + \frac{I}{\gamma} \end{bmatrix} \begin{bmatrix} b \\ a \end{bmatrix} = \begin{bmatrix} 0 \\ y \end{bmatrix}, \quad (4)$$

where $y = [y_1, \dots, y_l]^T$, $e1 = [1, \dots, 1]^T$, $a = [a_1, \dots, a_l]^T$, $Q_{ij} = \phi(x_i) \cdot \phi(x_j) = K(x_i, x_j)$, $i, j = 1, \dots, l$, and $K(x_i, x_j)$ is a kernel function. Different kernel functions present different mappings from the input space to the feature space. So the LS-SVR model changes with the different kernel function. In most cases, the radial basis function (RBF) $K(x_i, x_j) = \exp(-\|x_i - x_j\|^2/2\sigma^2)$ is employed as the kernel function because of its robustness.

Given the solution of formula (4), the LS-SVR model can be expressed as follows:

$$y(x) = \sum_{i=1}^l a_i K(x, x_i) + b. \quad (5)$$

Formula (5) is a nonlinear function with regard to $x \in R^D$. Normally, those samples x_i with nonzero θ_i are called support vectors of the regression function, because it is these critical samples in the training set T that solely determine the formulation of (5).

2.2. The Design of Online Learning LS-SVR Algorithm. The LS-SVR training algorithm introduced in Section 2.1 is a batch algorithm. That is, whenever a new sample is added into the training set, the existing regression function can only be updated by retraining the whole training set, which is not an efficient way to implement our prediction algorithm. Fortunately, we have recently derived an incremental LS-SVR training algorithm [34] as follows.

At the moment $k + l$, let the sample be $\{(x_i, y_i), i = k, \dots, k + l - 1, x_i \in R^D, y_i \in R\}$. The learning sample sets can be represented as $\{X_S(k), Y_S(k)\}$, where $X_S(k) = [x_k, x_{k+1}, \dots, x_{k+l-1}]^T$, $Y_S(k) = [y_k, y_{k+1}, \dots, y_{k+l-1}]^T$, and $x_k \in R^D, y_k \in R$.

So the kernel function matrix Q , the to-be-computed Lagrange coefficient a , and constant warp b are all functions of k . That is to say, at the moment k , they can be denoted separately by $Q_{ij}(k) = K(x_{i+k-1}, x_{j+k-1})$, $i, j = 1, \dots, l$, $a(k) = [a_k, a_{k+1}, \dots, a_{k+l-1}]^T$, and $b(k) = b_k$, so the output of LS-SVR (8) is transformed to be

$$y(x) = \sum_{i=k}^{k+l-1} a_i(k) K(x, x_i) + b(k). \quad (6)$$

Let $U(k) = Q(k) + I/\gamma$, where I is a unit matrix, so formula (7) can be rewritten as follows:

$$\begin{bmatrix} 0 & e1^T \\ e1 & U(k) \end{bmatrix} \begin{bmatrix} b(k) \\ a(k) \end{bmatrix} = \begin{bmatrix} 0 \\ Y(k) \end{bmatrix}. \quad (7)$$

Let $P(k) = U(k)^{-1}$, $h(k) = K(x_k, x_k) + 1/\gamma$, $H(k) = [K(x_{k+1}, x_k) + 1/\gamma, \dots, K(x_{k+l-1}, x_k)]^T$; then

$$\begin{aligned} P(k) &= \left[Q(k) + \frac{I}{\gamma} \right]^{-1} = \begin{bmatrix} h(k) & H(k)^T \\ H(k) & D(k) \end{bmatrix}^{-1} \\ &= \begin{bmatrix} 0 & 0 \\ 0 & D(k)^{-1} \end{bmatrix} + s_h(k) s_h(k)^T c_h(k), \end{aligned} \quad (8)$$

where

$$\begin{aligned} D(k) &= \begin{bmatrix} K(x_{k+1}, x_{k+1}) + \frac{1}{\gamma} & \cdots & K(x_{k+l-1}, x_{k+1}) \\ \vdots & \ddots & \vdots \\ K(x_{k+1}, x_{k+l-1}) & \cdots & K(x_{k+l-1}, x_{k+l-1}) + \frac{1}{\gamma} \end{bmatrix}, \\ s_h(k) &= [-1, H(k)^T D(k)^{-1}]^T, \\ c_h(k) &= \frac{1}{(h(k) - H(k)^T D(k)^{-1} H(k))}. \end{aligned} \quad (9)$$

Substituting formula (8) into (7) yields

$$\begin{aligned} b(k) &= \frac{e1^T P(k) Y_S(k)}{e1^T P(k) e1}, \\ a(k) &= P(k) \left[Y(k) - \frac{e1 e1^T P(k) Y_S(k)}{e1^T P(k) e1} \right] \\ &= P(k) (Y_S(k) - e1 b(k)). \end{aligned} \quad (10)$$

At the moment $k + l + 1$ a new sample (x_{k+l}, y_{k+l}) is added and the kernel matrix

$$\begin{aligned} Q_{ij}(k+1) &= K(x_{i+k}, x_{j+k}), \\ i, j &= 1, \dots, l, l+1, \end{aligned} \quad (11)$$

$$P(k+1) = U(k+1)^{-1} = \left[Q(k+1) + \frac{1}{\gamma} \right]^{-1}.$$

One problem with the algorithm is that the longer the prediction goes on, the bigger the training set will become and the more support vectors will be involved in the LS-SVR regression function (6). In some environments with limited memory and computational power, it is possible to stress out the system resources with the complexity of the LS-SVR model (6) growing in this way. One way to deal with this problem is to adopt a "forgetting" strategy. When training set grows to this maximum, then the LS-SVR model (6) will first be trained to remove the oldest sample before the next new sample is used to update the model.

To sum up, online learning LS-SVR is an optimization process along with time rolling; the whole process is described as follows.

Step 1. Initialization, $k = 1$.

Step 2. Select new data, while removing the oldest ones.

Step 3. Compute kernel matrices $Q(k)$ and $P(k)$.

Step 4. Compute $b(k)$ and $a(k)$ and predict $y(x_{k+l})$.

Step 5. Consider $k \leftarrow k + 1$; return to Step 2.

2.3. Online LS-SVR for Time Series Prediction. Let the dimension D of the reconstruction space F be equal to p ; the corresponding input and output samples for online learning are then constructed as follows:

$$X_S = \begin{bmatrix} x_k \\ x_{k+1} \\ \vdots \\ x_{k+l-1} \end{bmatrix} = \begin{bmatrix} X(k+p-1) \\ X(k+p) \\ \vdots \\ X(k+m-2) \end{bmatrix}$$

$$= \begin{bmatrix} x(k+p-1) & x(k+p-2) & \cdots & x(k) \\ x(k+p) & x(k+p-1) & \cdots & x(k+1) \\ \vdots & \vdots & \cdots & \vdots \\ x(k+m-2) & x(k+m-3) & \cdots & x(k+m-p-1) \end{bmatrix},$$

$$Y_S = \begin{bmatrix} y_k \\ y_{k+1} \\ \vdots \\ y_{k+l-1} \end{bmatrix} = \begin{bmatrix} x(k+p) \\ x(k+p+1) \\ \vdots \\ x(k+m-1) \end{bmatrix}, \quad l = m - p. \quad (12)$$

Along with the progression of the time k , we can get the following prediction function through the continuous learning samples:

$$\begin{aligned} x(t+1) &= f(X(t)) \\ &= \sum_{i=k+p-1}^{k+m-2} a_i(k) K(X(i), X(t)) \\ &\quad + b(k), \quad k = t - m + 2. \end{aligned} \quad (13)$$

From formula (13), the one-step-ahead prediction value can be obtained at current time T :

$$\begin{aligned} \hat{x}(T+1) &= f(X(T)) \\ &= \sum_{i=k+p-1}^{k+m-2} a_i(k) K(X(i), X(T)) \\ &\quad + b(k), \quad k = T - m + 2. \end{aligned} \quad (14)$$

Considering moving horizon prediction, we take the prediction value $\hat{x}(T+1)$ as the known condition of the next step prediction and construct the next input value $X(T+1) = [\hat{x}(T+1), x(T), \dots, x(T-p+2)]$. So we can obtain the two-step-ahead prediction value $\hat{x}(T+2)$ according to formula (6).

By analogy, the n -step prediction is then achieved as follows:

$$\begin{aligned} \hat{x}(T+2) &= f(\hat{x}(T+1), x(T), \dots, x(T-p+2)) \\ &\quad \vdots \\ \hat{x}(T+n) &= f(\hat{x}(T+n-1), \hat{x}(T+n-2), \dots, \\ &\quad x(T), \dots, x(T+n-p)). \end{aligned} \quad (15)$$

3. Nonlinear Time Series Prediction Based on Sliding ARMA Combined with Online LS-SVR

3.1. Review of Sliding Time Window ARMA. For the stationary time series, the ARMA model is defined as

$$\begin{aligned} x(t) &= \eta_1 x(t-1) + \cdots + \eta_p x(t-p) \\ &\quad + \varepsilon_t + \lambda_1 \varepsilon_{t-1} + \cdots + \lambda_q \varepsilon_{t-q}, \end{aligned} \quad (16)$$

where $\varepsilon_t \sim WN(0, \sigma^2)$, $p, q \geq 0$, and (p, q) is called the order of the model $\{x(t)\} \sim ARMA(p, q)$. In ARMA model, the current state $x(t)$ can be represented using its p past value $x(t-1), x(t-2), \dots, x(t-p)$, current white noise ε_t , and q past white noise value $\varepsilon_{t-1}, \varepsilon_{t-2}, \dots, \varepsilon_{t-q}$ in a linear regression form. The basic procedures of time series prediction utilizing ARMA can be described as follows.

3.1.1. Data Processing. If the data exhibit a clear nonstationary and periodic feature, we should take some preprocessing methods such as p -order difference and subtracting means of the series. The time series are much more stationary to fit the ARMA model after data processing.

3.1.2. Autocorrelation Analysis. The autocorrelation feature of the existing time series is then analyzed. Here, the characteristics of the autocovariance function (ACVF) and the autocorrelation function (ACF) are adopted to assure the stationary of the time series.

3.1.3. Modeling and Parameter Estimation. The parameters (p, q) are estimated according to the ACF and partial ACF. Besides, Bayesian information criterion (BIC) and Akaike's information criterion (AIC) are usually employed to compute the least order of the ARMA model. A more detailed introduction about ARMA modeling can be found in [35].

3.1.4. Time Series Prediction. According to the ARMA model defined by (16), we can obtain the following prediction function:

$$\begin{aligned} x(t+1) &= \eta_1 x(t) + \cdots + \eta_p x(t-p+1) \\ &\quad + \varepsilon_t + \lambda_1 \varepsilon_{t-1} + \cdots + \lambda_q \varepsilon_{t-q}. \end{aligned} \quad (17)$$

So the one-step-ahead prediction value at current time T can be described as

$$\begin{aligned} \hat{x}(T+1) &= \eta_T^1 x(T) + \cdots + \eta_T^p x(T-p+1) \\ &+ \varepsilon_T^1 + \lambda_T^1 \varepsilon_{T-1}^1 + \cdots + \lambda_T^q \varepsilon_{T-q}^1. \end{aligned} \quad (18)$$

Correspondingly, the two-step-ahead prediction value at current time T can be obtained as

$$\begin{aligned} \hat{x}(T+2) &= \eta_T^1 \hat{x}(T+1) + \eta_T^2 x(T) + \cdots + \eta_T^p x(T-p+2) \\ &+ \varepsilon_T^2 + \lambda_T^1 \varepsilon_{T-1}^2 + \cdots + \lambda_T^q \varepsilon_{T-q}^2. \end{aligned} \quad (19)$$

Substituting formula (18) into (19), formula (19) is then represented as

$$\begin{aligned} \hat{x}(T+2) &= (\eta_T^1 \eta_T^1 + \eta_T^2) x(T) + (\eta_T^1 \eta_T^2 + \eta_T^3) x(T-1) \\ &+ \cdots + (\eta_T^1 \eta_T^{p-1} + \eta_T^p) x(T-p+2) \\ &+ \eta_T^1 \eta_T^p x(T-p+1) + \eta_T^1 \varepsilon_T^1 + \varepsilon_T^2 + \eta_T^1 \lambda_T^1 \varepsilon_{T-1}^1 \\ &+ \lambda_T^1 \varepsilon_{T-1}^2 + \cdots + \eta_T^1 \lambda_T^q \varepsilon_{T-q}^1 + \lambda_T^q \varepsilon_{T-q}^2. \end{aligned} \quad (20)$$

In a similar manner, the n -step-ahead prediction value can be derived as

$$\begin{aligned} \hat{x}(T+n) &= \eta_T^1 \hat{x}(T+n-1) + \eta_T^2 \hat{x}(T+n-2) \\ &+ \cdots + \eta_T^n x(T) + \cdots + \eta_T^p x(T+n-p) \\ &+ \varepsilon_T^n + \lambda_T^1 \varepsilon_{T-1}^n + \cdots + \lambda_T^q \varepsilon_{T-q}^n \\ &= \eta_1(T) x(T) + \cdots \\ &+ \eta_p(T) x(T-p+1) + \varepsilon_q(T). \end{aligned} \quad (21)$$

As is discussed above, the ARMA model can be employed to forecast the further n -step value of time series. However, when the time series updates for online application, the ARMA model should vary with the data set. To ensure the efficiency and decrease the computation burden, the model is built in the neighborhood through sliding time window.

3.2. Nonlinear Prediction Using Sliding ARMA Revised by Online LS-SVR. The sliding ARMA above can be used to approximate time series online and achieve the prediction of time series. However, the ARMA model is essentially linear. Using it to fit nonlinear time series online, we can only obtain the local linear model of nonlinear systems and the unknown modeling errors caused by the nonlinearity and outside disturbance of system can be ignored on the whole. In order to compensate for the ARMA modeling errors and the unknown disturbance, the online LS-SVR model is adopted to revise the nonlinear prediction errors of the sliding ARMA. The flow of combined online prediction algorithm is described in Figure 1.

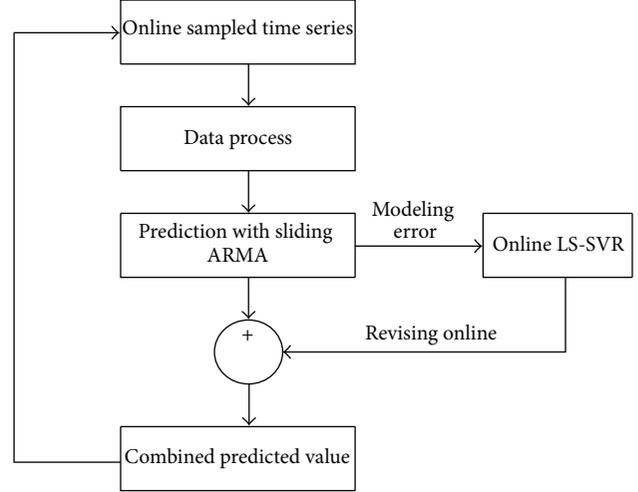


FIGURE 1: Process of combined online prediction algorithm.

The initial data is preprocessed firstly to train for the ARMA model. In the local neighborhood, the sliding ARMA model is built for the future n -step prediction of nonlinear time series. At the same time, the modeling errors of the sliding ARMA are used to compose new time series. The online LS-SVR algorithm is then trained for the n -step prediction of the new time series. The final predicted value is yielded by both of the two model outputs. More detailed steps are as follows.

Step 1. Data processing.

Step 2. Initial training of the ARMA model.

Step 3. At time T , the new sample $x(T)$ is added to online train the sliding ARMA model (in the neighborhood of $x(T)$) and $\hat{x}(T+n)$ is then predicted.

Step 4. The modeling errors $\tilde{x}(T) = x(T) - \hat{x}(T)$ are added to online train the LS-SVR and $\hat{\tilde{x}}(T+n)$ is then predicted.

Step 5. Final predicted value $\hat{x}(T+n) = \hat{x}(T+n) + \hat{\tilde{x}}(T+n)$.

Step 6. When data updates, repeat Step 1.

4. Fault Prediction Based on Time Series Abnormality Estimation

For many complex engineering systems, the acquisition of fault data generally is at the cost of equipment damage and loss of economy; it is unpractical to fully obtain the fault training data or prior knowledge [15]. Even though the training data of faults can be gained, the obtained data will gradually be out of use because of time variance and uncertainty of the actual system. Therefore fault prediction only with data of normal condition is provided with realistic significance.

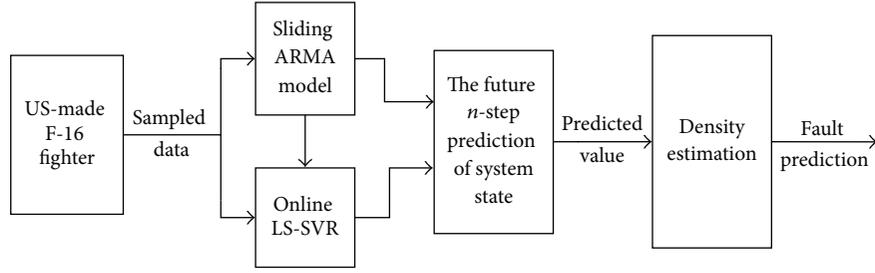


FIGURE 2: Online fault prediction process of F-16.

Based on the combined prediction algorithm proposed above, the n -step-ahead prediction value at time T , $\hat{x}(T+n)$, $T = p, p+1, \dots$, can be obtained. This result includes the future information corresponding to fault. So we can use it to predict the fault.

Let ω_1 denote the normal state and ω_2 denote abnormal or fault state. It is assumed that system is normal at the beginning and all of time series data $x(i)$, $i = 1, 2, \dots, N$ in the past are known. The probability density functions of the time series sample $x(k)$, $k = 1, 2, \dots$ can be estimated by [15]

$$p(x(k) | \omega_1) = \frac{1}{NV_N(x(k))} \sum_{i=1}^{k-1} \exp\left(-\frac{\|x(k) - x(i)\|^2}{2V_N^2(x(k))}\right). \quad (22)$$

For a given sample $\hat{x}(T+n)$ at the time T , we want to judge whether $\hat{x}(T+n) \in \omega_2$ or not. We construct a test set M including all of normal data till now, namely, $M = \{x(i) | i \in \{1, 2, \dots, T\}\}$. For an arbitrary sample $y \in M$, let its log-likelihood be denoted by $L(y) = \log p(y | \omega_1)$. Similarly, the log-likelihood of $\hat{x}(T+n)$ is denoted by $L(\hat{x}(T+n)) = \log p(\hat{x}(T+n) | \omega_1)$. We test the hypothesis with the following rule:

$$\begin{aligned} &\text{If } P(L(y) \leq L(\hat{x}(T+n))) > \theta, \\ &\text{then } \hat{x}(T+n) \in \omega_2, \end{aligned} \quad (23)$$

where θ is a contrived threshold and $0 < \theta < 1$. It means the confidence limit of hypothesis tests and influences the performance of fault predictor. A larger θ makes missing alarm rate become greater and makes false alarm rate become lesser; on the contrary, a smaller θ makes missing alarm rate become lesser and makes false alarm rate become greater. So the value of θ can be adjusted to obtain the best predictor with the error as low as possible.

5. Experiments and Results

In order to demonstrate the potential of the proposed method in engineering practice, a practical case study for the US-made F-16 fighter from [36] is provided in this section. Suppose that the fighter flies straight away at a height of 500 m with the speed of 0.45 Mach at the beginning. The original trimming angle of attack $\alpha = 2.4686^\circ$; the elevator deflection angle $\delta_e = -1.9815^\circ$; the composite linear velocity

$V = 152.4$ m/s and the sampling period is 12.5 ms. In this paper, we consider the faults of fighter's structure only. According to the features of faults in fighter's structure, the pitching angular velocity q is chosen as the criterion to check the situation of fault.

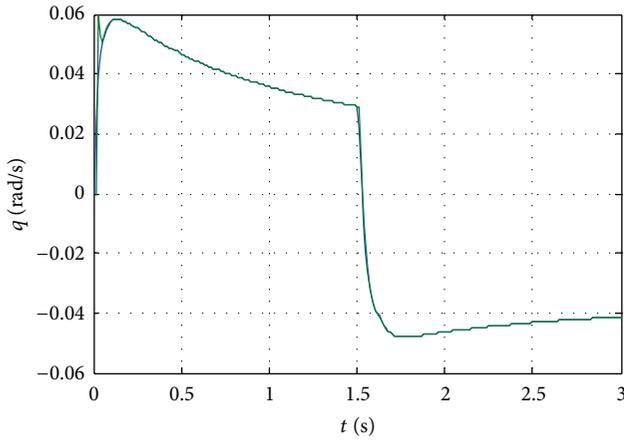
Suppose that left elevator is locked into $+0.5^\circ$ at 1.5 s. Using the method presented in this paper, the structure of simulation process is shown in Figure 2.

At the k th sampling time point, a sliding ARMA model can be obtained using the approach in Section 3. According to the error between the actual system output and the ARMA model output in (16), the online LS-SVR algorithm can be designed to revise the nonlinear prediction errors of the sliding ARMA. To simplify the calculating process, the order of ARMA model is set as 2. The other parameters selected are the embedding dimension $p = 6$ and the parameters of LS-SVR with RBF kernel $\gamma = 1000$, $e = 0.075$, $\sigma = 1$.

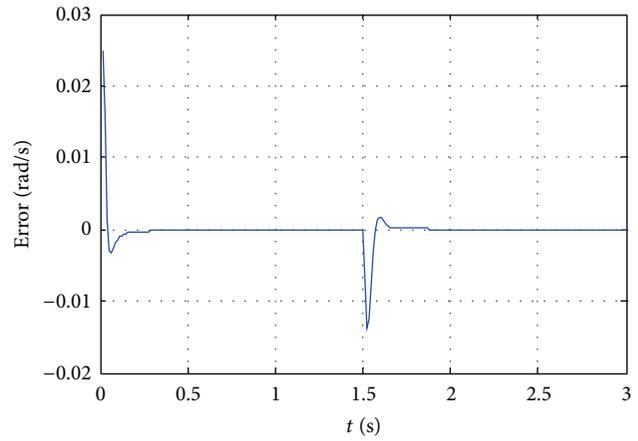
Figure 3 shows the one-step prediction results and the prediction errors when the system has no external disturbance. The solid line means the actual output of system and the dash line means the prediction output of our approach. According to Figure 3, the future dynamics of the system can be predicted accurately.

To further prove the effectiveness of the proposed method, the external disturbance of system is then considered. In Figure 4, the prediction errors of our approach are compared with the results only with online LS-SVR. The solid line represents the errors without online LS-SVR, while the dash line represents the error of our approach. Figure 4(a) shows the result when the variance of external disturbance is 0.02 and Figure 4(b) shows the result when the variance of external disturbance is 0.1. With the variance of external disturbance increasing, the errors of only using online LS-SVR are increasing rapidly, while the prediction errors of our approach are changeless. Obviously, the proposed approach can compensate for the external disturbance efficiently.

At last, the fault judgment result through hypothesis test is provided in Figure 5, when $V_N(x) = 0.0000005$. The result shows that the fault is predicted at the time 1.5125 s when the variance of external disturbance is 0.1, which means that the fault can be predicted in one period. As is known, the aircraft has a flexible element in input channels to avoid the abrupt changes of control. Generally, it can be regarded as an inertial construction; the time to add a control completely into the

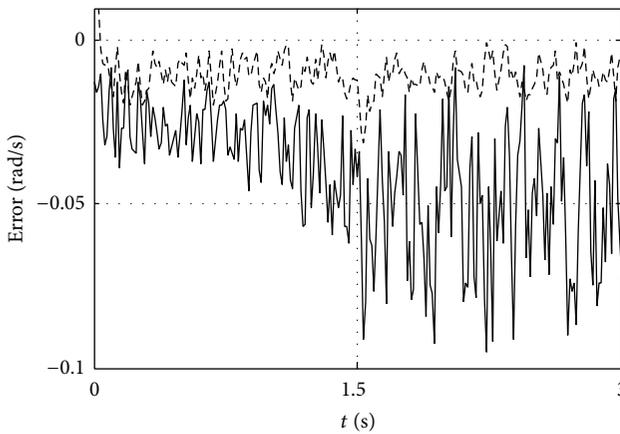


(a) The predicted output and the actual output of the system

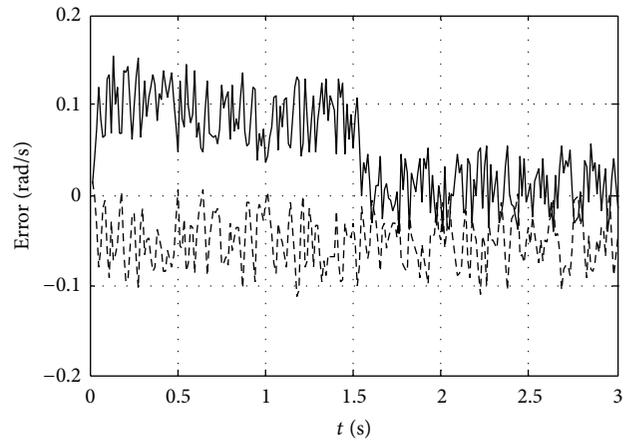


(b) The prediction error

FIGURE 3: The simulation results without external disturbance.



(a) The comparison of the prediction error when the disturbance variance is 0.02



(b) The comparison of the prediction error when the disturbance variance is 0.1

FIGURE 4: The comparison of the prediction error.

input side of system is 0.05 seconds. Same argument is that the time for the fault signal added to the input side of system is also considered as 0.05 seconds. This means that the time of the period when the systems deviate from normal condition until the fault happened is 0.05 s, namely, four sampling periods. Under such circumstances, the proposed approach can predict the fault in advance with three sampling periods. So the approach can do the job of online fault forecasting successfully. It is worth mentioning that the fault can be predicted only with data of normal condition.

6. Conclusion

This paper presents a robust online fault prediction approach for model-unknown nonlinear system. The sliding ARMA model is used to approximate the nonlinear systems and the online LS-SVR is designed to compensate for the nonlinear modeling error with external disturbance. The prediction

results from the two models are used to judge the fault based on a density function estimation method. The proposed method can predict the fault of nonlinear system accurately only using normal data of system. And it can study and predict with fewer amounts of calculation while system is running. So, it is believed with good real-time capability and applicability. The results of simulation on fighter F-16 show the efficiency of the proposed method.

The purpose of the paper is to supply an effective approach for online fault prediction of some actual systems with complex nonlinearity. As with other approaches, it is impossible for the proposed approach to succeed in all scenarios. However, it can at least provide an alternative and complementary solution to some problems in which other available techniques may fail. In the future, we should further improve the celerity of our method through optimizing the sliding ARMA in the aspect of model updating. Besides, the parameter selection and the interaction between two models will be also a further research work.

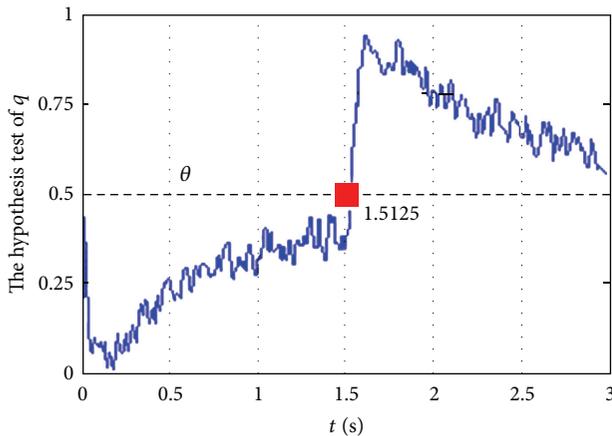


FIGURE 5: The hypothesis test and the result of fault prediction when the variance of external disturbance is 0.1.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work is supported by Innovation Program of Shanghai Municipal Education Commission under Grant no. 12YZ156, the Fund of SUES under Grant no. 2012gp45, Shanghai Municipal Natural Science Foundation under Grant no. 12ZR1412200, and National Natural Science Foundation of China under Grant no. 61271114.

References

- [1] G. Vachtsevanos, F. Lewis, M. Roemer, A. Hess, and B. Wu, *Intelligent Fault Diagnosis and Prognosis for Engineer Systems*, John Wiley & Sons, 2006.
- [2] S. K. Yang, "An experiment of state estimation for predictive maintenance using Kalman filter on a DC motor," *Reliability Engineering and System Safety*, vol. 75, no. 1, pp. 103–111, 2002.
- [3] X. Hu, D. V. Prokhorov, and D. C. Wunsch II, "Time series prediction with a weighted bidirectional multi-stream extended Kalman filter," *Neurocomputing*, vol. 70, no. 13–15, pp. 2392–2399, 2007.
- [4] Z. J. Zhou, C. H. Hu, H. D. Fan, and J. Li, "Fault prediction of the nonlinear systems with uncertainty," *Simulation Modelling Practice and Theory*, vol. 16, no. 6, pp. 690–703, 2008.
- [5] D. H. Zhou and P. M. Frank, "Strong tracking filtering of nonlinear time-varying stochastic systems with coloured noise: application to parameter estimation and empirical robustness analysis," *International Journal of Control*, vol. 65, no. 2, pp. 295–307, 1996.
- [6] D. Wang, D. H. Zhou, Y. H. Jin, and S. J. Qin, "A strong tracking predictor for nonlinear processes with input time delay," *Computers & Chemical Engineering*, vol. 28, no. 12, pp. 2523–2540, 2004.
- [7] M. Z. Chen, D. H. Zhou, and G. P. Liu, "A new particle predictor for fault prediction of nonlinear time-varying systems," *Developments in Chemical Engineering and Mineral Processing*, vol. 13, no. 3–4, pp. 379–388, 2005.
- [8] M. E. Orchard and G. J. Vachtsevanos, "A particle filtering approach for on-line failure prognosis in a planetary carrier plate," *International Journal of Fuzzy Logic and Intelligent Systems*, vol. 7, no. 4, pp. 221–227, 2007.
- [9] X. Si, C. H. Hu, J. B. Yang, and Q. Zhang, "On the dynamic evidential reasoning algorithm for fault prediction," *Expert Systems with Applications*, vol. 38, no. 5, pp. 5061–5080, 2011.
- [10] M. A. Schwabacher, "A survey of data-driven prognostics," in *Proceedings of the InfoTech at Aerospace: Advancing Contemporary Aerospace Technologies and Their Integration*, pp. 887–891, Arlington, Va, USA, September 2005.
- [11] G. Niu and B. S. Yang, "Dempster-Shafer regression for multi-step-ahead time-series prediction towards data-driven machinery prognosis," *Mechanical Systems and Signal Processing*, vol. 23, no. 3, pp. 740–751, 2009.
- [12] S. L. Ho and M. Xie, "The use of ARIMA models for reliability forecasting and analysis," *Computers & Industrial Engineering*, vol. 35, no. 1–4, pp. 213–216, 1998.
- [13] R. Y. Li and R. Kang, "Research on failure rate forecasting method based on ARMA model," *Systems Engineering and Electronics*, vol. 30, no. 8, pp. 1588–1591, 2008.
- [14] J. Zhao, L. M. Xu, and L. Lin, "Equipment fault forecasting based on ARMA model," in *Proceedings of the IEEE International Conference on Mechatronics and Automation (ICMA '07)*, pp. 3514–3518, Harbin, China, August 2007.
- [15] Z. D. Zhang and S. S. Hu, "Fault prediction of fighter based on nonparametric density estimation," *Journal of Systems Engineering and Electronics*, vol. 16, no. 4, pp. 831–836, 2005.
- [16] J. T. Connor, R. D. Martin, and L. E. Atlas, "Recurrent neural networks and robust time series prediction," *IEEE Transactions on Neural Networks*, vol. 5, no. 2, pp. 240–254, 1994.
- [17] S. L. Ho, M. Xie, and T. N. Goh, "A comparative study of neural network and box-jenkins ARIMA modeling in time series prediction," *Computers & Industrial Engineering*, vol. 42, no. 2–4, pp. 371–375, 2002.
- [18] G. Xue, L. Xiao, M. Bie, and S. Lu, "Fault prediction of boilers with fuzzy mathematics and RBF neural network," in *Proceedings of the International Conference on Communications, Circuits and Systems*, vol. 2, pp. 1012–1016, Hong Kong, May 2005.
- [19] M. Luo, D. Wang, M. Pham et al., "Model-based fault diagnosis/prognosis for wheeled mobile robots: a review," in *Proceedings of the 31st Annual Conference of IEEE Industrial Electronics Society (IECON '05)*, pp. 2267–2272, Raleigh, NC, USA, November 2005.
- [20] C. H. Hu, X. P. Cao, and W. Zhang, "Fault prediction based on Bayesian MLP neural networks," *GESTS Transaction Journal*, vol. 16, no. 1, pp. 17–25, 2005.
- [21] G. Yang and X. Wu, "Fault prediction of ship machinery based on gray neural network model," in *Proceedings of the IEEE International Conference on Control and Automation (ICCA '07)*, pp. 1063–1066, Guangzhou, China, June 2007.
- [22] H. G. Han and J. F. Qiao, "Prediction of activated sludge bulking based on a self-organizing RBF neural network," *Journal of Process Control*, vol. 22, no. 6, pp. 1103–1112, 2012.
- [23] D. Basllk, S. Pal, and D. C. Patranabis, "Support vector regression," *Neural Information Processing—Letters and Reviews*, vol. 11, no. 10, pp. 203–224, 2003.
- [24] U. Thissena and R. Brakela, "Using support vector machines for time series prediction," *Chemometrics & Intelligent Laboratory Systems*, vol. 6, pp. 35–49, 2003.

- [25] A. J. Smola and B. Schölkopf, "A tutorial on support vector regression," *Statistics and Computing*, vol. 14, no. 3, pp. 199–222, 2004.
- [26] X. X. Ma, X. Y. Huang, and Y. Chai, "Fault process trend prediction based on support vector machines," *Acta Simulata Systematica Sinica*, vol. 14, pp. 1548–1551, 2002.
- [27] K. Y. Chen, "Forecasting systems reliability based on support vector regression with genetic algorithms," *Reliability Engineering and System Safety*, vol. 92, no. 4, pp. 423–432, 2007.
- [28] S. Hou and Y. Li, "Short-term fault prediction based on support vector machines with parameter optimization by evolution strategy," *Expert Systems with Applications*, vol. 36, no. 10, pp. 12383–12391, 2009.
- [29] D. T. Liu, Y. Peng, and X. Y. Peng, "Online adaptive status prediction strategy for data-driven fault prognostics of complex systems," in *Proceedings of the Prognostics & System Health Management Conference (PHM-Shenzhen '11)*, pp. 1–6, Shenzhen, China, May 2011.
- [30] L. Ye, D. H. You, X. G. Yin, K. Wang, and J. C. Wu, "An improved fault-location method for distribution system using wavelets and support vector regression," *Electrical Power and Energy Systems*, vol. 55, pp. 467–472, 2014.
- [31] A. Baylar, D. Hanbay, and M. Batan, "Application of least square support vector machines in the prediction of aeration performance of plunging overfall jets from weirs," *Expert Systems with Applications*, vol. 36, no. 4, pp. 8368–8374, 2009.
- [32] Y. H. Gao and Y. B. Li, "Fault prediction model based on phase space reconstruction and least squares support vector machines," in *Proceeding of the 9th International Conference on Hybrid Intelligent Systems (HIS '09)*, pp. 464–467, Shenyang, China, August 2009.
- [33] J. A. K. Suykens, J. De Brabanter, L. Lukas, and J. Vandewalle, "Weighted least squares support vector machines: robustness and sparse approximation," *Neurocomputing*, vol. 48, pp. 85–105, 2002.
- [34] S. Su and W. Zhang, "Fault prediction of nonlinear system using time series novelty estimation," *Journal of Digital Information Management*, vol. 11, no. 3, pp. 207–212, 2013.
- [35] J. Q. Fan and Q. W. Yao, *Nonlinear Time Series: Nonparametric and Parametric Methods*, pp. 12–123, Springer, New York, NY, USA, 2003.
- [36] Y. Liu, *Intelligent Adaptive Reconfigurable Control for Complex Nonlinear System*, Nanjing University of Aeronautics and Astronautics, Nanjing, China, 2003.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

