

A New Approach to Robust Fault Detection of Model-unknown Nonlinear Systems

Shengchao Su^{1,2} and Wei zhang¹

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

²*College of Information Science and Technology, Donghua University, Shanghai, 201620, China*
sytussc@163.com

Abstract

Most of actual engineering systems have the characteristics such as nonlinearity, difficult modeling, and high cost to experiment with fault. So fault detection in these systems is difficult. In this paper, a new approach to robust fault detection is proposed for model-unknown nonlinear systems. Firstly, a general framework for fault detection is discussed. This framework is capable of associating each detection result with a confidence level. An online learning least squares support vector regression algorithm is then used to design the framework. Meanwhile, robust strategy and forgetting strategy are added to the algorithm. Accordingly, a robust online fault detection method is developed with less amount of calculation. Finally, the proposed method is applied to fault detection of model-unknown fighter F-16. The simulation results show that the proposed method can detect fault quickly without the model of system or any prior knowledge of fault.

Keywords: *robust; online fault detection; model-unknown; nonlinear systems; least squares support vector regression; fighter*

1. Introduction

With the enhancing requests on system reliability and security, fault detection technique are always the hot issues of research field in weapons, aircrafts, and almost all the industrial applications [1]. Now, a little headway has been made in fault detection methods. In general, there are two main categories, one is based on model and the other is based on knowledge.

The methods based on the system model [2-5] can use the in-depth knowledge of system fully, and can combine closely with control system. This is advantageous to detect the fault promptly and accurately, which is the premise of monitoring, fault-tolerant control, system repair and restructuring. However, these methods need the precise mathematical model, and they are also affected by the error of modeling, parameter perturbation and outside disturbance. Generally they are only suitable for the linear object single fault analysis.

Because most of actual industrial system's dynamics has the complex nonlinearity, the precise mathematical model is not easy to establish. So the researcher pay more attention to the methods based on knowledge [6-11], which includes neural networks, wavelet technique, statistical method and rough set, etc. These methods can avoid the dependence on the model of system, so that they have robustness, adaptation, self-study and disposal of complex nonlinearity. It is worth to mention that many knowledge-based methods need fault training data or prior knowledge. For example, the neural networks need many data of all kinds of

faults and its performance is limited by the number and distributed situation of the chosen sample.

Many complex engineering systems such as fighters have multiform of faults which are dangerous and have a short latent period. Experiment with fault is dangerous and expensive. So the data or prior knowledge of fault, which needed for fault detection, could hardly be acquired. In other words, the data of normal situation can be acquired only. Therefore fault detection only with data of normal condition is provided with realistic significance.

In this paper, we propose a new approach to detect fault in model-unknown nonlinear system only with the data of normal situation. As with other detection algorithms, it is impossible for our 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. The remainder of this paper is organized as follows. Firstly, the fault detection problem above is formulated in Section 2. Then, a general online fault detection framework is proposed in Section 3. Based on the framework, a robust online fault detection algorithm using incremental learning Least Squares Support Vector Regression (LS-SVR), is devised in Section 4. In Section 5, the simulation example of fighter F-16 is given to demonstrate the effectiveness of the proposed method. Finally, conclusions and future research directions are discussed in Section 6.

2. Problem Description

Consider the fault detection problem as follows.

For a non-linear system whose mathematics model is unknown, suppose that the system work in normal state at the beginning and the sample data of system outputs $\mathbf{x}(t)$, since the system ran is known. We want to detect whether or not a latent fault appear at present only based on the normal data of system outputs.

According to the knowledge of pattern recognition and time series analysis, the problem above can be formulated as follows:

Let $e_n(t)$ denote an event in the time sequences of system outputs, which can be calculated by the sample data. $e_n(t)$ represents the working state of systems at time t . It is known that $e_n(t)$ ($t = 1, 2, \dots, N$) is the normal event and we want to judge whether $e_n(N+1)$ is an abnormal or fault event.

3. A Framework for Online Fault Detection

We first reflect on the concept of fault, as well as that of fault detection. First of all, fault is always a relative concept with regard to our current knowledge. Therefore, fault should be defined in the context of a representation of our current knowledge. Such representation can be a database or a model. In this paper, we prefer representing our knowledge with a model, simply because of its mathematical neatness. Moreover, in online applications, it is desirable that the representation of our knowledge can also be updated with the acquisition of new data. Second, there is generally no clear-cut line between fault state and normal state in real world application. So it is attractive to associate each fault event with a value to characterize how confident our judgment is. Finally, in a real world environment with uncertain factors the fault events in a time sequence are generally associated with segments, instead of individual time points. These understandings lay the foundation of a formal formulation of an online fault detection framework [12], which is introduced in the remaining part of this section.

In this paper, a temporal sequence is represented as $\mathbf{X}(t)$, where $t = 1 \dots N$. They can be time series, or some other objects that can be indexed by time t . $\mathbf{X}(t)$ is a stochastic process, and $\mathbf{x}(t)$ is employed to represent one of its realizations.

As mentioned previously, a model $M_x(t_0)$ is used to represent our knowledge about the underlying temporal sequence up to t_0 . This model can be a physics-based model provided by domain experts, or a model constructed from available data $\mathbf{x}(t)$, where $t = 1 \dots t_0$.

Definition 1 Matching Function and Matching Value.

The matching function, denoted as $F(M_x(t_0 - 1), x(t_0))$, is a function that can quantify how well the model $M_x(t_0)$ matches the temporal sequence. The match value $V(t_0)$, where $V(t_0) \in R$, is defined as $V(t_0) = F(M_x(t_0 - 1), x(t_0))$. In other words, the matching value $V(t_0)$ is used to measure how well our knowledge about the time series at $t_0 - 1$, represented by $M_x(t_0 - 1)$, can describe the instance $x(t_0)$.

Definition 2 Occurrence.

Denoted by $O(t_0)$, occurrence at t_0 is defined as

$$O(t_0) = I\{V(t_0) \notin (-\varepsilon(t_0), \varepsilon(t_0))\} \quad (1)$$

Where $I\{\cdot\}$ is the indicator function, and $2\varepsilon(t_0) > 0$ is a predefined tolerance width. $\varepsilon(t_0)$ can be determined based on the noise level, as well as the precision requirement of the underlying problem. Note that $O(t_0)$ is a random variable.

Definition 3 Wonder.

A wonder is observed if $O(t_0) = 1$. That is, a wonder happens when a new instance $x(t_0)$ falls outside of the tolerance range $(-\varepsilon(t_0), \varepsilon(t_0))$.

Definition 4 Event and Event Duration.

Denoted by $E_n(t_0)$, an event at time t_0 is defined as

$$E_n(t_0) = [O(t_0) O(t_0 + 1) \dots O(t_0 + n - 1)] \quad (2a)$$

where n is called the event duration. The 1-norm of $E_n(t_0)$ is denoted as $|E_n(t_0)|$. That is

$$|E_n(t_0)| = \sum_{i=0}^{n-1} O(t_0 + i) \quad (2b)$$

$|E_n(t_0)|$ is the number of wonders happened in the event $E_n(t_0)$. Event duration n is a predetermined algorithmic parameter. We omit the absolute operation on $O(t_0 + i)$ in formula (2b) because $O(t_0 + i)$ is nonnegative according to formula (1). $E_n(t_0)$ is a random binary vector with at most $2n$ different realizations, while $|E_n(t_0)|$ is a random variable with at most $n+1$ different realizations. A realization of $E_n(t_0)$ (or $|E_n(t_0)|$) is denoted as $e_n(t_0)$ (or $|e_n(t_0)|$). The discrete density function of $|E_n(t_0)|$ is represented as $p_{E_n}(|e_n(t_0)|)$, where $|e_n(t_0)| = 0, 1, \dots, n$. The formulation of $p_{E_n}(|e_n(t_0)|)$ can be determined by the

occurrence $O(t_0 + i), i = 0, 1, \dots, n - 1$. For example, if the occurrences in $\{O(t_0 + i), i = 0, 1, \dots, n - 1\}$ are identical independent Bernoulli variables, $E_n(t_0)$ becomes a binomial random variable. $P_{E_n}(|e_n(t_0)|)$ will be different if the occurrences in $\{O(t_0 + i), i = 0, 1, \dots, n - 1\}$ are interdependent, such as following a Markov chain distribution.

Definition 5 Fault Event with Confidence.

Given a confidence level $c(t_0)$, where $c(t_0) \in (0, 1)$, Event $e_n(t_0)$ is defined as a fault event with confidence $c(t_0)$, if $e_n(t_0)$ satisfies:

$$(a) |e_n(t_0)| > h, \text{ where } h \text{ is a fixed lower bound of } |E_n(t_0)| \text{ with } h \in N, \quad (3a)$$

$$(b) P_{E_n}(|e_n(t_0)|) < 1 - c(t_0). \quad (3b)$$

The condition (3a) makes sure that at least certain number of wonders happen in the event $e_n(t_0)$, while the condition (3b) ensures that the probability for the number of wonders to happen in the event $e_n(t_0)$ is small enough to satisfy our confidence level $c(t_0)$. The h is an algorithmic parameter to define the lower bound of the number of wonders.

Four items in this framework need to be instantiated before it becomes a concrete algorithm.

- (a) The model $M_x(t_0)$ to represent the temporal sequence $X(t)$;
- (b) The matching function $F(M_x(t_0 - 1), x(t_0))$ to quantify the disagreement between the model output and the temporal sequence $X(t)$ observation at t_0 ;
- (c) The tolerance width $2\varepsilon(t_0)$ at t_0 ;
- (d) The discrete density function of $|E_n(t_0)|$, $P_{E_n}(|e_n(t_0)|)$, where $|e_n(t_0)| = 0, 1, \dots, n$.

In Section 4, an online fault detection algorithm is derived by instantiating the four items discussed above.

4. Robust Online Fault Detection based on Incremental Learning LS-SVR

4.1. Incremental Learning LS-SVR

LS-SVR is an extension of Support Vector Regression (SVR) which is a machine learning method based on statistical learning theory. In this section, we briefly introduce the basic concepts of LS-SVR [13], and propose an incremental learning 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 $y \in R$, construct a linear regression function with regard to W and $\Phi(x)$

$$f(x) = W^T \Phi(x) + b \quad (4)$$

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 (4) are obtained by solving an optimization problem:

$$\begin{cases} \min_{w,b,e} J(W, e) = \frac{1}{2}W^T W + \frac{1}{2}\gamma \sum_{i=1}^l e_i^2 \\ s.t.: y_i = W^T \phi(x_i) + b + e_i, i = 1, 2, \dots, l \end{cases} \quad (5)$$

In formula (5), the regularization parameter γ determines the fitting error and smoothness, and the nonnegative error variable e_i is used to construct a soft margin hyper plane. This optimization problem including the constraints can be solved by the Lagrange function as following:

$$L(W, b, e, a) = J(W, e) - \sum_{i=1}^l a_i \{W^T \phi(x_i) + b + e_i - y_i\} \quad (6)$$

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 .

$$\begin{bmatrix} 0 & e1^T \\ e1 & Q + I/\gamma \end{bmatrix} \begin{bmatrix} b \\ a \end{bmatrix} = \begin{bmatrix} 0 \\ y \end{bmatrix} \quad (7)$$

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 [7]. Given the solution of formula (7), the LS-SVR model can be expressed as following:

$$y(x) = \sum_{i=1}^l a_i K(x, x_i) + b \quad (8)$$

The formula (8) 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 (8).

However, it is worth mentioning that the LS-SVR training algorithm introduced above 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 detection algorithm. Fortunately, we have recently derived an incremental LS-SVR training algorithm as follows.

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

Hence 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 as $Q_{ij}(k) = K(x_{i+k-1}, x_{j+k-1}), i, j = 1, \dots, l, a(k) = [a_k, a_{k+1}, \dots, a_{k+l-1}]^T, b(k) = b_k$, so the output of LS-SVR (8) is transformed to be equation (9).

$$y(x) = \sum_{i=k}^{k+l-1} a_i(k) K(x, x_i) + b(k) \quad (9)$$

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

$$\begin{bmatrix} 0 & e1^T \\ e1 & U(k) \end{bmatrix} \begin{bmatrix} b(k) \\ a(k) \end{bmatrix} = \begin{bmatrix} 0 \\ Y(k) \end{bmatrix} \quad (10)$$

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

$$P(k) = [Q(k) + I/\gamma]^{-1} = \begin{bmatrix} h(k) & H(k)^T \\ H(k) & D(k) \end{bmatrix}^{-1} = \begin{bmatrix} 0 & 0 \\ 0 & D(k)^{-1} \end{bmatrix} + s_h(k) s_h(k)^T c_h(k) \quad (11)$$

Where $D(k) = \begin{bmatrix} K(x_{k+1}, x_{k+1}) + 1/\gamma & \dots & K(x_{k+l-1}, x_{k+1}) + 1/\gamma \\ \vdots & \ddots & \vdots \\ K(x_{k+1}, x_{k+l-1}) + 1/\gamma & \dots & K(x_{k+l-1}, x_{k+l-1}) + 1/\gamma \end{bmatrix}$, $s_h(k) = [-1, H(k)^T D(k)^{-1}]^T$,

$$c_h(k) = 1/(h(k) - H(k)^T D(k)^{-1} H(k)).$$

Put formula (11) into formula (10), then

$$b(k) = \frac{e1^T P(k) Y_s(k)}{e1^T P(k) e1}, a(k) = P(k) \left[Y(k) - \frac{e1 e1^T P(k) Y_s(k)}{e1^T P(k) e1} \right] = P(k) (Y_s(k) - e1 b(k)) \quad (12)$$

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

$$Q_{ij}(k+1) = K(x_{i+k}, x_{j+k}), i, j = 1, \dots, l, l+1, P(k+1) = U(k+1)^{-1} = [Q(k+1) + 1/\gamma]^{-1}.$$

4.2. Online Fault Detection Algorithm Devised by Incremental Learning LS-SVR

In this section, the four items mentioned in Section 3 are instantiated to design a fault detection algorithm.

First, LS-SVR is used to model a temporal sequence $\mathbf{X}(t)$. LS-SVR has quite a few attractive features. For example, (a) LS-SVR is established on the structural risk minimization principle, so it has the advantages of simple structure, the global optimum, good generalization ability (b) LS-SVR converts solving the problem of convex quadratic programming into solving linear equation group, which greatly reduces the computational complexity, and (c) LS-SVR can handle high dimensional data efficiently.

More specifically, given a realization of a temporal sequence, or $\mathbf{x}(t)$, where $t = 1, 2, \dots, t_0$, we can construct a set of training samples $T_D(t_0)$ from $\mathbf{x}(t)$:

$$T_D(t_0) = \{(x_D^t, y^t), t = D \dots t_0 - 1\} \quad (13)$$

Where $x_D^t = [x(t-D+1) \dots x(t)]^T$, $y^t = x(t+1)$, and D is called the embedding dimension of the training set $T_D(t_0)$. According to Section 4.1, from the training set $T_D(t_0)$ LS-SVR training algorithm can construct a regression function $f(x_D^{t_0})$

$$\hat{y}^{t_0} = \hat{x}(t_0 + 1) = f(x_D^{t_0}) = \sum_{i=1}^{t_0-1} a_i K(x_D^i, x_D^{t_0}) + b \quad (14)$$

Naturally, the model $M_x(t_0)$ representing the temporal sequence $\mathbf{X}(t)$ can be defined as

$$M_x(t_0) = f(x_D^{t_0}) = \sum_{i=1}^{t_0-1} a_i K(x_D^t, x_D^{t_0}) + b \quad (15)$$

Equation (15) suggests that model $M_x(t_0)$ essentially incorporates all the points of the temporal sequence $\mathbf{x}(t)$, where $t = 1, 2, \dots, t_0$, and thus is a good candidate for representing our acquired knowledge of $\mathbf{X}(t)$.

Second, the matching function and the matching value are defined as

$$V(t_0) = F(M_x(t_0 - 1), x(t_0)) = x(t_0) - M_x(t_0 - 1) = x(t_0) - \hat{x}(t_0) \quad (16)$$

Equation (11) indeed suggests that the matching value $v(t_0)$ is the residual of the regression function (14) at t_0 .

Third, the tolerance width $2\varepsilon(t_0)$ at t_0 is needed to define. Fortunately, LS-SVR formulation generally adopts a nonnegative error variable e [13], which possesses exactly the same flavor as our tolerant range defined in Definition 2. Thus, we just simply merge the concepts of two tolerant ranges together, and define the tolerant width in Definition 2 as $2e$ for any t_0 . A direct consequence of this definition is that any sample $(x_D^{t_0}, y^{t_0})$ in $T_D(t_0)$ that turns to be a wonder to model $M_x(t_0)$ will be a support vector in the updated model $M_x(t_0 + 1)$ [13].

Finally, in order to make the algorithm theoretically tractable, we simply define $E_n(t_0)$ as a sequence of independent Bernoulli random variables with the same parameter, and the discrete density function of $|E_n(t_0)|$, $p_{E_n}(|e_n(t_0)|)$, can thus be formulated as [14]

$$p_{E_n}(|e_n(t_0)|) = \begin{cases} \binom{n}{|e_n(t_0)|} q^{t_0^{|e_n(t_0)|}} (1 - q)^{n - |e_n(t_0)|}, & \text{where } |e_n(t_0)| = 0 \dots n \\ 0, & \text{otherwise} \end{cases} \quad (17)$$

Where $q(t_0)$ is the probability of an occurrence in the even $E_n(t_0)$ to be a wonder. The $q(t_0)$ can be approximately estimated as

$$\hat{q}(t_0) = \frac{N_{sv}(t_0)}{D(t_0)} \quad (18)$$

Where $N_{sv}(t_0)$ is the number of support vectors in model $M_x(t_0)$, and $D(t_0)$ is the number of training samples in $T_D(t_0)$ according to (18).

This definition is valid under the following two assumptions [12]: (a) The occurrences in an event are independent; (b) All occurrences in an event have approximately the same probability of being a wonder. Intuitively, the first assumption is reasonable if the regression function (14) can sufficiently capture the dependent relationship in a temporal sequence. This can be achieved by an adequate training stage to fully build the model $M_x(t_0)$ before it is utilized for detection. The second assumption is sensible if the event duration n is not too large. Failure of meeting either assumption deteriorates the accuracy of the confidence level associated with the detection output.

Now, a complete online fault detection algorithm has been proposed based on the framework in Section 3. This algorithm requires a set of algorithmic parameters, which include (a) Embedding dimension D ; (b) Event duration n ; (c) Tolerant width 2ε ; (d) Kernel function $K(x_i, x_j)$; (e) Confidence level c ; and (f) Fixed lower bound of the number of wonders h . Finally, the online algorithmic procedures can be summarized as follows:

Step1: Initialization, establish the training set $T_D(t)$ and the initialized model $M_x(t)$.

Step2: When a new point $x(t_0)$ becomes available, calculate the matching value $v(t_0)$ following (16).

Step3: Based on $v(t_0)$ and the tolerant width 2ε , determine the value of occurrence $O(t_0)$ following (1).

Step4: Based on the values of occurrence $O(t_0 - i)$, where $i = 0 \dots n - 1$, determine the value of event $E_n(t_0 - n + 1)$ following (2). If $E_n(t_0 - n + 1)$ meets the conditions in condition (3), a fault event, is detected with a confidence of c .

Step5: The training set $T_D(t_0 - 1)$ is updated to $T_D(t_0)$ by considering the new point $x(t_0)$ following (13). Accordingly, the model $M_x(t_0 - 1)$ is then updated to $M_x(t_0)$ using the incremental training algorithm of LS-SVR in Section 4.1.

Step6: If the detection stage does not finish, go to step 1. Otherwise, quit.

4.3. Robust Online Fault Detection

LS-SVR is computationally more efficient than the standard SVR, but unfortunately the robustness of standard SVR is lost. So the robustness of the fault detection algorithm must be considered.

The event duration n is a critical parameter in our algorithm. In some cases where the novel events are outstanding, the original algorithm is not sensitive to n . However, in the other cases, an improper choice of this parameter does lead to the deterioration of the algorithmic performance. Meanwhile, it is hard to know in advance what kind of fault events will be detected in a temporal sequence, which is indeed the nature of fault detection. Although, given a particular problem, how to select an optimal event duration n for the original detection algorithm is still an open topic, in some applications it is possible for us to know the range in which an optimal event duration n may fall in. Based on this assumption, a robust version of the original detection algorithm can be intuitively devised according to [12]. In this variant, we evenly pick up z different event duration n 's from the available range, and apply each of them to the original detection algorithm, and thus obtain z detection outputs. The final robust detection output is obtained by a voting procedure among all the generated outputs. When implementing this idea, it is not necessary to literally repeat the original detection algorithm for each event duration n . Repeating merely the step 4 for each event duration n is adequate.

4.4. Methods to Reduce the Amount of Calculation

One problem with the detection algorithm is the longer the prediction goes on, the bigger the training set $T_D(t_0)$ will become, and the more support vectors will be involved in the LS-SVR regression function (14). 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 (14) growing in this way. One way to deal with this problem is to adopt a “forgetting”

strategy. When training set $T_D(t_0)$ grows to this maximum M , then the LS-SVR model (14) will first be trained to remove the oldest sample before the next new sample is used to update the model. Accordingly, the $q(t_0)$ in (18) becomes $\hat{q}(t_0) = \frac{N_{sv}(t_0)}{M}$, where $N_{sv}(t_0)$ is still the number of support vectors in model $M_x(t_0)$.

This variant is intrinsically suitable for non-stationary temporal sequences, as it can be updated in real-time to fit the most recent behavior of a temporal sequence.

5. Application Example

The fighter F-16 of USAF [15] is employed as the example for simulation. Suppose that the fighter flies straight-away at a height of 500 m and at a speed of 0.45 Mach at the beginning. The original trimming conditions are angle of attack $\alpha = 2.4686^\circ$, elevator deflection $\delta_e = -1.9815^\circ$, and composite linear velocity $V = 152.4 \text{ m/s}$. In this paper, we consider the faults of fighter's structure only. As knows, the position saturation of F-16's left/right elevator is $\pm 25^\circ$, the velocity saturation is $60^\circ/\text{s}$; the position saturation of F-16's left/right aileron is $\pm 21.5^\circ$, the velocity saturation is $80^\circ/\text{s}$; the position saturation of F-16's rudder is $\pm 30^\circ$, the velocity saturation is $120^\circ/\text{s}$. The sampling period of flight system is 12.5 ms .

We suppose that left elevator locked into $+0.5^\circ$ at 1.5s. According to the features of faults in fighter's structure, the vertical speed w , pitching rate q and yawing rate r are chosen as the criterion to check the situation of fault. Our models are then trained to realize the online fault detection of F-16 by detecting the novelty on the time series of w, q, r . The algorithmic parameters are arbitrarily set as (a) Embedding dimension $D = 8$; (b) Event duration $n = 6$; (c) Tolerant width $2\varepsilon = 0.2$; (d) Kernel function $K(X_i, X_j) = \exp\{-|X_i - X_j|^2\}$; (e) Confidence level $C = 96\%$; (f) Fixed lower bound of number of wonders $h=n/2$. The results of simulation show in Figure 1 to Figure 3.

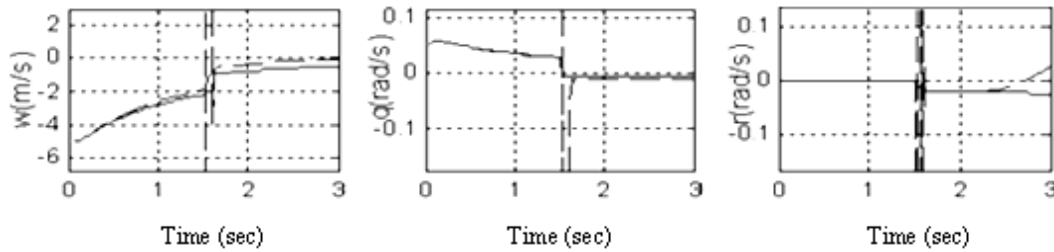


Figure 1. The One-step Prediction Results of System Output Variable w, q, r

The results in Figure 1 are the one-step prediction results of w, q, r using online SVR. Where the solid line means the real sequences of w, q, r and the dashed line means the results of one-step prediction on w, q, r . The plots in Figure 1 show that, our algorithm has good approximation and generalization properties, which is the foundation of our method to detect fault successfully.

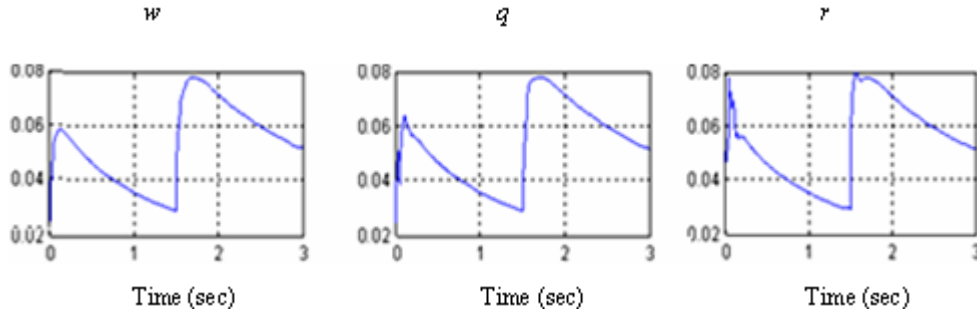


Figure 2. The Corresponding Discrete Density Function Values of $|E_n(t_0)|$

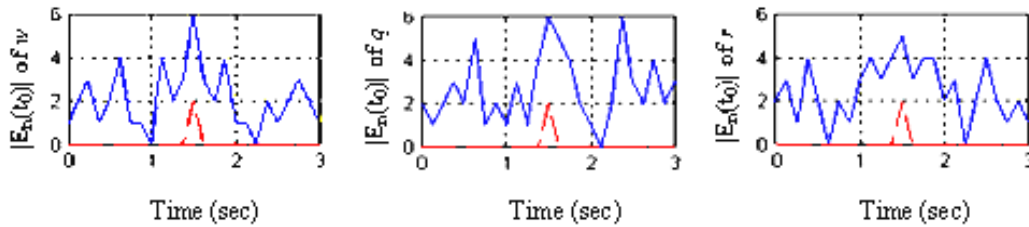


Figure 3. The Corresponding Values of $|E_n(t_0)|$ and the Results of Detection

The Figure 2 gives the discrete density function results of $|E_n(t_0)|$ of w, q, r , where all of ordinates mean the discrete density values without unit. Meanwhile, the corresponding values of $|E_n(t_0)|$ are also shown by the solid curve in the Figure 3, where all of ordinates reflect the number of wonders happened in the $|E_n(t_0)|$ of w, q, r without unit. Based on definition 5, the fault events in the time series of w, q and r are successfully detected. The dashed line in the Figure 3 shows the results of detection. It doesn't equal to zero when we detects fault, on the contrary, it equals to zero when we doesn't detect fault.

The above results show that our method can detect the latent fault in time without advance fault data. According to the features of faults in fighter's structure [10], if the lateral vertical speed w , pitching rate q and yawing rate r are detected anomaly at the same time, we can make a conclusion that the position of fault is elevator. This means that our method realize the online fault detection of elevator successfully.

In addition, for F-16, generally it is considered that the fighter can be repaired in 8 periods since the fault take place, so the method can also do the job of abrupt fault forecast.

6. Conclusion

This paper presents a robust online fault detection scheme for a class of model-unknown nonlinear systems. The scheme includes a mechanism for associating each detection result from an online LS-SVR with a confidence value, and the robust strategy and forgetting strategy are adapted to guarantee the robustness and celerity of it. So the method can detect fault quickly with good robust ability only using normal data of system. The result of simulation on fighter F-16 proved the efficiency of our method. It is a new approach for online fault detection of actual engineering systems. However, how to utilize the existing information and then detail the situation and degree of fault are still the problems that need study in the future.

Acknowledgements

This work is supported by Innovation Program of Shanghai Municipal Education Commission under Grant 12YZ156 and Shanghai Municipal Natural Science Foundation under Grant 12ZR1412200.

References

- [1] G. Vachtsevanos, F. Lewis, M. Roemer, A. Hess and B. Wu, "Intelligent Fault Diagnosis and Prognosis for Engineer Systems", New Jersey: John Wiley & Sons, (2006), pp. 1-18.
- [2] A. Medvedev, "Disturbance Attenuation Enhancement in Continuous Parity Space Methods", European Control Conference (ECC 97), Brussels, Belgium, (1997) July 1-4.
- [3] V. Venkatasubramanian, Rengaswamy, K. Yin and S. N. Kavuri, "A review of process fault detection and diagnosis part I: Quantitative model-based methods", Computers and Chemical Engineering, vol. 27, no. 3, (2003), pp. 293-311.
- [4] C. Qu and H. Zhang, "The fault diagnosis of momentum close-loop system used in satellite based on the UIO double observers", Aerospace Control, vol. 23, no. 6, (2005), pp. 66-71. (in Chinese)
- [5] N. Tudoroiu, K. Khorasani, "Satellite fault diagnosis using a bank of interacting Kalman filters", IEEE Transactions on Aerospace and Electronic Systems, vol. 43, no. 4, (2007), pp. 1334-1350.
- [6] M. Catelani and A. Fort, "Fault diagnosis of electronic analog circuits using a radical basis function network classifier", Journal of the International Measurement Confederation, vol. 28, no. 3, (2000), pp. 147-158.
- [7] A. Quteishat and C. P. Lim, "A modified fuzzy min-max neural network with rule extraction and its application to fault detection and classification", Applied Soft Computing, vol. 8, no. 2, (2008), pp. 985-995.
- [8] L. X. Dong, D. M. Xiao, Y. S. Liang and Y. L. Liu, "Rough set and fuzzy wavelet neural network integrated with least square weighted fusion algorithm based fault diagnosis research for power transformers", Electric Power Systems Research, vol. 78, no. 1, (2008), pp. 129-136.
- [9] J. P. Dai, Z. S. Wei, C. P. Wang and G. Y. Zhang, "Expert system for radar equipment faults diagnosis based on rough set theory", Proceedings of the 8th International Conference on Electronic Measurement and Instruments, Xian, China, (2007) August 16-18.
- [10] M. Kano, S. Hasebe, I. Hashimoto and H. Ohno, "A New Multivariate Statistical Process Monitoring Method Using Principal Component Analysis", Computers and Chemical Engineering, vol. 25, (2011), pp. 1103-1113.
- [11] C. Li, Z. H. Song and P. Li, "Bearing fault detection via wavelet packet transform and rough set theory", Proceedings of the 5th World Congress on Intelligent Control and Automation, Hangzhou, China, vol. 2, (2004) June 15-19, pp. 1663-1666.
- [12] J. Ma and S. Perkins, "Online novelty detection on temporal sequences", Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining (SIGKDD '03), Washington, DC, USA, (2003) August 24-27.
- [13] 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, (2002), pp. 85-105.
- [14] A. M. Mood, F. A. Graybill and D. C. Boes, "Introduction to The Theory of Statistics, 3rd Edition", McGraw-Hill, Inc, New York, (1974).
- [15] Y. Liu, "Intelligent adaptive reconfigurable control for complex nonlinear system", Nanjing university of aeronautics and astronautics, Nanjing, China, (2003).

Authors



Shengchao Su, he received the B.Sc. degree in Electrical Engineering and Automation and M.Sc. degree in Testing Technology and Automation Devices from Jiangnan University, China, in 2004 and 2007, respectively. He joined to Shanghai University of Engineering Science, China, as lecture in 2007. Currently, he is a Ph.D. student of Control Theory and Control Engineering, Donghua University, Shanghai, China. His research interests include fault detection and diagnosis, intelligent control and networked control systems.



Wei Zhang, he received his B.Sc. degree in applied chemistry and M.Sc. degree in computational mathematics from the University of Electronic Science and Technology of China, in 1999 and 2005, respectively. He got a PhD degree in control theory and control engineering from Shanghai Jiao Tong University, China, in 2010. He is currently an associate professor of Shanghai University of Engineering Science. His research interests include time-delay systems, multi-agent systems and nonlinear control.