Qi, H, Shi, Y, Li, S, Tian, Y, Yu, DL and Gomm, JB

Fault tolerant control for nonlinear systems using sliding mode and adaptive neural network estimator

http://researchonline.ljmu.ac.uk/id/eprint/12085/

Article

Fault tolerant control for nonlinear systems using sliding mode and adaptive neural network estimator

Haiying Qi[1] · Yiran Shi[2] · Shoutao Li[2] · Yantao Tian[2] · Ding-Li Yu[3] 🄳 · J. B. Gomm[3]

Abstract

This paper proposes a new fault tolerant control scheme for a class of nonlinear systems including robotic systems and aeronautical systems. In this method, a sliding mode control is applied to maintain system stability under the post-fault dynamics. A neural network is used as on-line estimator to reconstruct the change rate of the fault and compensate for the impact of the fault on the system performance. The control law and the neural network learning algorithms are derived using the Lyapunov method, so that the neural estimator is guaranteed to converge to the fault change rate, while the entire closed-loop system stability and tracking control is guaranteed. Compared with the existing methods, the proposed method achieved fault tolerant control for time-varying fault, rather than just constant fault. This greatly expands the industrial applications of the developed method to enhance system reliability. The main contribution and novelty of the developed method is that the system stability is guaranteed and the fault estimation is also guaranteed for convergence when the system subject to a time-varying fault. A simulation example is used to demonstrate the design procedure and the effectiveness of the method. The simulation results demonstrated that the post-fault is stable and the performance is maintained.

Keywords Fault tolerant control · Intelligent method · Sliding mode control · Adaptive neural estimator · Data analysis

1 Introduction

Fault tolerant control for industrial systems has attracted great attention in the past decades, and different methods have been proposed. A popular method for fault detection of nonlinear dynamic systems is using a nonlinear state observer (Liu et al. 2017; Yang et al. 2015a; Zhang et al. 2016) or a fuzzy observer-based method (Dong et al. 2018; Li et al. 2016). In these methods, observers were used to predict system state. The residual would then be generated as a function of state estimation error. While the fault occurrence can be detected, however, the identification of fault amplitude will normally be difficult. Therefore, these methods do not make a large contribution in passive fault tolerant control design. Another type of fault diagnosis system uses a nonlinear dynamic model to predict system output. The model can be constructed with fuzzy logic or neural networks (Kamal et al. 2014; Yu et al. 2014). In recent years, a nonlinear observer with an on-line estimator method (Zhang et al. 2002; Ferrari et al. 2009; Polycarpou et al. 2004) attracted much attention of researchers. This method uses an on-line estimator in the dynamic equation of the system to estimate a fault or disturbance. While the fault is estimated by the on-line estimator and compensated for in the state dynamics, the observed state under the fault condition converges to the nominal system state. It therefore naturally forms a passive fault tolerant control strategy.

✉ Ding-Li Yu
D.Yu@ljmu.ac.uk

1 School of Electronics and Information, Changchun Architecture and Civil Engineering College, Changchun, China

2 Department of Control Theory and Engineering, College of Communication Engineering, Jilin University, Changchun, China

3 Control Systems Research Group, Liverpool John Moores University, Byrom Street, Liverpool L3 3AF, UK

🄳 Springer

Recently developed methods for fault tolerant control include the use of adaptive neural networks (Chen et al. 2016), back-stepping method (Kwan and Lewis 2000), terminal sliding mode control method (Van et al. 2018), etc. The adaptive observer method for fault tolerant control usually uses an on-line estimator to estimate a fault, where the fault estimator can be implemented using different components and are adapted with different learning algorithms. A radial basis function (RBF) network was used in Trunov and Polycarpou (1999) and Polycarpou and Trunov (2000) as the on-line estimator, and a projection-based learning algorithm was developed to tune the weights of the network. As the reported work was in an early stage, simulations showed that the tuning of the estimator is very difficult and the convergence of the estimation is slow. Rather than directly estimate disturbance, neural networks have also been used to estimate unknown parameters in a nonlinear uncertain system without combining with a nonlinear state observer. In the air-to-fuel ratio (AFR) control of air path in a spark ignition (SI) engine using a sliding mode method (Wang and Yu 2008), an RBF network was used to estimate two unknown parameters, the partial derivative of air passed the throttle with respect to the air manifold pressure and that w.r.t. crankshaft speed. The adaptive law of the network estimator was derived so that the states out of sliding mode will be guaranteed to converge to the sliding mode in finite time. Moreover, a RBF network was used to estimate the optimal sliding gain in Wang and Yu (2007) to achieve an optimal robust performance in AFR control against model uncertainty and measurement noise.

Intelligent methods have been developed for data analysis (Al-Janabi and Alkaim 2019; Al Janabi and Abaid Mahdi 2019; Al Janabi 2018) and applied to networks and communications (Alkaim and Al Janabi 2020; Al Janabi et al. 2020). In fault detection area, fault prediction and fault diagnosis using different analytical methods and data driving method have also investigated (Al-Janabi et al. 2014; Patel et al. 2015; Al-Janabi 2017; Al-Janabi et al. 2015; Ali 2012). The application of these methods greatly enhanced system reliability. Numerical solutions and soft computing have been widely studied Abu Arqub and Abo-Hammour (2014), Abu Arqub et al. (2016) and applied to solve scientific computing and engineering applications (Abu Arqub et al. 2017; Abu Arqub and Maayanh 2018).

2 Related work

Fault tolerant control is defined as a control strategy that guarantees system stability and maintains control performance after a system fault occurs. If an external disturbance is a matched one or the distribution matrix of the system fault is the same as that of the control input, robust control can be achieved easily by using a basic sliding model method. For mismatched disturbance, Chen et al. (2000), Chen and Chen (2010) and Yang et al. (2011) proposed a disturbance observer for a class of nonlinear systems. The developed method can estimate a constant fault but cannot estimate a time-varying fault. A sliding mode method was developed for disturbance estimation in Yang et al. (2013), which used the disturbance observer of Chen et al. (2000) in the sliding surface. While the disturbance estimation converges, the initial state out of the sliding mode would converge to the sliding mode. The method can guarantee the stability of the whole system, but is still robust to constant disturbance only. So, the aforementioned method will have very limited applications in practice as most faults considered in real industrial systems are time-varying. The disturbance observer was also used in Kayacan et al. (2017) to design a robust control for nonlinear systems. In this method, the disturbance estimation was achieved by combining the disturbance observer with a type-2 neural-fuzzy network (T2NFN) in Kayacan et al. (2015), and was tuned by combining three adaptive laws: a conventional estimation law, a robust estimation law and the T2NFN law. While the method can estimate time-varying disturbance without bias, the structure of the estimation system is rather complex and the computing load is high.

A class of nonlinear systems discussed in this paper is described by the state space model in (1). The model represents many industrial systems, including magnetic levitation suspension systems (Michail 2009), the chaotic Duffing oscillators (Shu 2012), a three-dimension of freedom model helicopter system (Chen et al. 2016), near space vehicle systems (Jiang et al. 2014), robotic manipulators (Yang et al. 2015b), etc. This paper proposes a new fault diagnosis and fault tolerant control method for this class of nonlinear systems. The novelty and contributions of the proposed paper are as follows. The method uses a sliding mode approach and includes an on-line fault estimator in the sliding surface.

The major contribution in this paper is that the proposed method is used to design a feedback tracking system tolerant to time-varying faults compared with the existing methods that can be tolerant to constant fault only, or to time-varying faults but with very high computing load. The fault tolerant control is achieved by compensating the impact of the fault on the system performance. Here, the RBF network is used to on-line estimate the first-order derivative of the fault w.r.t. time and is included in the control law, so that the time-varying fault is tolerated. The Lyapunov method is used to derive the adaptation laws for the RBF estimator, so that the convergence of the estimator and the stability of the entire system are guaranteed.

Compared with the three typical existing fault tolerant control methods, the disturbance observer method, the disturbance observer plus neural-fuzzy network and the integral sliding mode method, the advantages of the proposed method are listed in Table 1.

The proposed method and mentioned three major previous existing methods are all methods for control systems design in nature. Therefore, the system stability is the key feature. Also, a fault tolerant control method is mainly featured by its post-fault stabilization ability. Proving stability is most challenging task in developing these new methods. Because of this, the first feature we compared between the proposed method and the existing methods is chosen stability. Secondly, these methods are usually use algorithm on-line adaptation, or self-learning neural networks. Therefore, the convergence of self-learning network or adaptive algorithms would be the second important feature to be considered. Then, if the fault could be tolerated, and if the time-varying fault rather than constant fault only can be tolerated, is also an important feature. Finally, the two features considered are tracking control and computing load, which is also important due to the requirement for real system implementation.

A numerical simulation is used to evaluate the developed method, and the results prove the effectiveness of the method in the passive fault tolerant control by automatic fault compensation. The paper is organized in the following way. The sliding mode control and fault estimators are described in Sect. 3. Convergence of the adaptive laws for the fault estimators and system stability is proved in Sect. 4. Section 5 presents a simulation example, and Sect. 6 draws some conclusions.

3 Fault tolerant control design

In engineering applications, many dynamic systems including robotic systems and aeronautical systems can be modelled with the following class of second-order nonlinear model,

$$
\begin{aligned}
\dot{x}_1 &= x_2 + f(t), \\
\dot{x}_2 &= G(x) + Q(x)u(t), \\
y &= x_1.
\end{aligned}
\tag{1}
$$

where x_1, x_2 are system states, u is the control variable, f denotes the system component fault that is a smooth nonlinear function, y is the output, $G(.)$ and $Q(.)$ are nonlinear functions of state x, and Q is supposed to be invertible. We consider the system component fault which can be presented by the term $f(t)$ in (1). Some model-system parameter mismatch or external disturbance can also be presented by this term, and be considered in robust control system design. Note that in the second term of the right-hand side of the second equation in (1), the input is separable from the function of state, so that a sliding mode control strategy can be applied to the dynamic system to guarantee the system stability and reference tracking property.

For the purpose of fault tolerant control, we propose to use sliding mode control strategy to form a discontinued state feedback control to stabilize the closed-loop system. The sliding mode control is a type of discontinuous state feedback control, in which a sliding surface is designed such that any state of the surface will be driven onto the surface, and converge to zero along the surface. In this system, a neural network of RBF type will be used to estimate the fault change rate, $\dot{f}(t)$. As a class of linearly parameterized neural networks, the RBF network can approximate any smooth nonlinear mapping to any accuracy if provided with enough hidden layer nodes, due to its universal approximation ability. In further, the sliding mode control is such designed that the output is robust to the system fault, to realize the fault tolerant control.

A sliding mode control is developed for the system in (1) with the sliding surface designed as follows:

$$
s = c_1 \dot{x}_1 + \dot{x}_2 + \widehat{\dot{f}} + c_2 x_1,
\tag{2}
$$

where c_1, c_2 are two designing parameters, which will be chosen in the design to give more design freedom and to determine the system response speed, and $\widehat{\dot{f}}$ is an estimate of the change rate of the fault by a RBF network, $\mathrm{NN}(z) : R^q \to R$ as follows:

Table 1 Feature comparison between different methods

Methods	Stability	Convergence	Tolerant fault	Tracking control	Computing load
Disturbance observer (DB)	Stable	Converge	Constant fault	Yes	Low
DB + neural-fuzzy	Not sure	Converge	Time-varying fault	Not sure	High
Integral SMC	Stable	N/A	Constant fault	Yes	Low
Proposed method	Stable	Converge	Time-varying fault	Yes	Low

$$\widehat{f} = \mathrm{NN}(z) = \widehat{W}^{\mathrm{T}}\phi(z), \tag{3}$$

where $z = [z_1, \ldots, z_q]^{\mathrm{T}} \in \Re^q$ is the input vector of the RBF network, $\Phi(z) = [\varphi_1(z), \varphi_2(z), \ldots, \varphi_p(z)]^{\mathrm{T}} \in \Re^p$ is the output of the nonlinear basis function, $\widehat{W} \in \Re^p$ is the weight vector. The Gaussian function is chosen in this research as the basis function,

$$\varphi_i(z) = \exp\left(-\frac{\|z - c_i\|^2}{\sigma_i^2}\right), \quad i = 1, \ldots, p, \tag{4}$$

where c_i is a vector of the same dimension with z and is called the centre of the ith hidden layer node and σ_i is the radius of the ith Gaussian function. There are several nonlinear basis functions which can be used in RBF network, e.g. sigmoid function, hyperbolic tangent, etc. We used Gaussian function because it is commonly used by most researchers for nonlinear mapping.

According to the universal approximation of neural networks, the smooth nonlinear function of the change rate of $f(t)$ can be approximated by an optimal network NN^* with a minimal approximation error, i.e.

$$\dot{f}(t) = \mathrm{NN}^*(z) + \varepsilon, \tag{5}$$

where ε is the approximation error. The weight of the optimal network NN^* is called optimal weight W^*, which gives the minimum approximation error ε and is defined as follows:

$$W^* = \arg\min_{W \in \Omega_f}\left\{\sup\left|\mathrm{NN}(z|W) - \dot{f}\right|\right\}. \tag{6}$$

where $\Omega_f = \left\{\widehat{W} : \|\widehat{W}\| \leq M\right\}$ is a valid field of the estimated weight $\widehat{W}$, $M > 0$ is a design parameter, and $S_z \subset \Re^q$ is an allowable set of the state vectors. The optimal weight will then generate

$$\dot{f}(t) = W^{*\mathrm{T}}\phi(z) + \varepsilon^*, \tag{7}$$

with the minimal approximation error bounded by

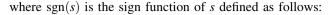$$\varepsilon^* \leq \bar{\varepsilon}, \tag{8}$$

where $\bar{\varepsilon}$ is a positive number.

Now, with the $\widehat{f}(t)$, the estimate of the first-order derivative of the disturbance w.r.t. time given as

$$\widehat{f}(t) = \sum_{i=1}^{p} \widehat{w}_i \exp\left(-\frac{z - c_i^2}{(\sigma_i)^2}\right). \tag{9}$$

We propose the control law as follows:

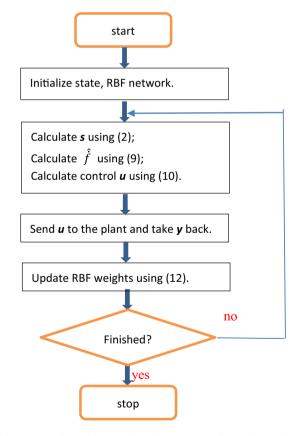$$\begin{cases} u = U^{-1}(x)(u_1 + u_2) \\ u_1 = -G(x) - c_1 x_2 - \widehat{f} - c_2 x_1, \\ \dot{u}_2 = -c_1 \widehat{W}^{\mathrm{T}}\phi(z) - k * \mathrm{sgn}(s) \end{cases} \tag{10}$$

where $\mathrm{sgn}(s)$ is the sign function of s defined as follows:

$$\mathrm{sgn}(s) = \begin{cases} 1, & s > 0 \\ 0, & s = 0 \\ -1, & s < 0 \end{cases}. \tag{11}$$

The estimate of the fault change rate is used to generate fault estimation that is used as fault detection residual. In addition, the estimation of the fault change rate is mainly used in the control system as compensation to form the fault tolerant control. The fault tolerant control configuration including sliding mode control and neural network estimator is demonstrated in the flow chart in Fig. 1.

To outline the algorithm proposed in this paper for fault tolerant control system design, the algorithm is named as the Adaptive Neural Network Estimator based Fault Tolerant Control Scheme (ANNEFTCS). The neural network is used to estimate the fault change rate and is on-line adapted to cope with the post-fault dynamics when the system is subject to fault occurrence. The estimation is then used by the fault compensation mechanism in the fault tolerant control. The stabilization of the post-fault dynamics is operated by the integral sliding mode, which is also used to realize set-point tracking control. In aeronautical engineering and robotic engineering, control systems

Fig. 1 Flow chart of operational procedure of the fault tolerant control system

are often subject to component or actuator faults. The proposed ANNEFTCS will definitely enhance the reliability of these systems.

4 Estimation convergence and stability analysis

As the fault estimator used in this research will estimate the fault change rate on-line, the convergence of the estimation is very important. In addition, when a fault occurs within the system, the post-fault dynamics may cause the system unstable. Therefore, stabilize the post-fault system is also important. For the sliding surface designed in (2) and control law in (10), we now prove system stability and neural estimator convergence. These will be given in the following theorem.

Assumption 1 The system fault $f(t)$ is a smooth time function, i.e. its first-order derivative exists and bounded.

$$\dot{f}(t) \leq \overline{\dot{f}}, \tag{12}$$

where $\overline{\dot{f}}$ is a constant.

Theorem 1 *Consider the nonlinear system in* (1) *with a fault satisfying Assumption 1. If the neural estimator is updated using the adaption law as follows:*

$$\widehat{w}_i(t + \Delta t) = \widehat{w}_i(t) + \int_t^{t+\Delta t} \frac{1}{\gamma} s * \exp\left(-\frac{z - c_i^2}{\sigma_i^2}\right) \mathrm{d}t, \tag{13}$$
$$i = 1, \ldots, p,$$

the system states will be guaranteed to be attracted on the sliding mode (2) *and along it to converge to 0, the neural estimator will converge to the optimal estimators given in* (7).

Proof Proof is given in "Appendix." □

4.1 Tracking control

We consider the case of zero reference. Deriving $x_2 = \dot{x}_1 - f$ from (1) with respect to t, and substitute it into (2), we have

$$s = c_1 \dot{x}_1 + \dot{x}_2 + \widehat{f} + c_2 x_1 = \ddot{x}_1 + c_1 \dot{x}_1 + c_2 x_1 - \widetilde{\dot{f}}, \tag{14}$$

where

$$\widetilde{\dot{f}} = \dot{f} - \widehat{\dot{f}}. \tag{15}$$

when the state converges to the sliding surface, $s = 0$, we have

$$\ddot{x}_1 + c_1 \dot{x}_1 + c_2 x_1 = \widetilde{\dot{f}}. \tag{16}$$

Lemma 1 (Khalil and Grizzle 1996) *If a nonlinear system $F(x, u)$ is input-to-state stable, and the input satisfies $\mathrm{Lim}_{t \to \infty} u = 0$, then the state satisfies $\mathrm{Lim}_{t \to \infty} x = 0$.* □

By choosing appropriate c_1 and c_2 such that the left-hand side polynomial in (16) is Hurwitz, Eq. (16) will be input-to-state stable. Also keep in mind that $\widetilde{\dot{f}}$ converges to 0, we can conclude that according to Lemma 1, $\mathrm{Lim}_{t \to \infty} x_1 = 0$. This proves the output converges to the zero reference, $y = x_1 \to 0$.

4.2 Limitations and hypothesis

To guarantee the stability of the post-fault dynamics of the entire system and the convergence of the adaptive neural network estimator, a limitation to the fault type has to be applied. That is that the change rate of the time-varying fault must be bounded, i.e. the fault change rate should be equal to or smaller than a pre-specified value given in assumption 1. The developed algorithm extends the type of the fault tolerated from the existing methods to time-varying fault. And the bound of fault change rate makes the stabilization of the system and the adaptive algorithm convergence guaranteed. However, this condition limited the applicability of the developed algorithm in practice, as in aeronautical and robotic engineering some system faults with a big change rate and therefore cannot be tolerable with the developed method. More complexed method may be developed to tackle the faults with too big change rate, but the computing load will correspondingly increased. This will also bring difficulty for practical applications.

Besides, to ensure the designed system output tracks setpoint, the system must be input-state stable. This condition is not strict, as most engineering systems possess this feature. If a special system that is not input to state stable, then the stabilization would be first priority in the design.

It is well known the sliding mode control will be subjected to chattering problem that is especially not desirable when the developed method is applied to practical systems. However, we have addressed this problem in the paper by applying an approximating to the sigh function with (24). The simulation example proves that it works very well.

5 Simulation example

5.1 Existing method

To compare the performance of the proposed method with an existing method in Yang et al. (2013), the numerical example used in Yang et al. (2013) is adopted in this paper, and the integral sliding mode method developed in Yang

et al. (2013) is also used for the numerical example that is as follows:

$$\dot{x}_1 = x_2 + f(t),$$
$$\dot{x}_2 = -2x_1 - x_2 + e^{x_1} + u \qquad (17)$$
$$y = x_1.$$

with the initial states $x(0) = [1, -1]^{\mathrm{T}}$. To simulate the control system and compare the performance, a zero reference is used for the output and the fault is simulated as follows:

$$f(t) = \begin{cases} 0, & 0 \le t < 6 \,(\mathrm{s}) \\ 0.5, & 6 \le t < 12 \,(\mathrm{s}) \\ \sin[(t-12)*2\pi/8], & 12 \le t < 20 \,(\mathrm{s}) \end{cases} \qquad (18)$$

Here, the three piece of time function of the fault are used to present: no fault for the first 6 s, then a constant fault of 0.5 for the second 6 s, and finally a sinusoidal function as time-varying function for the last 8 s.

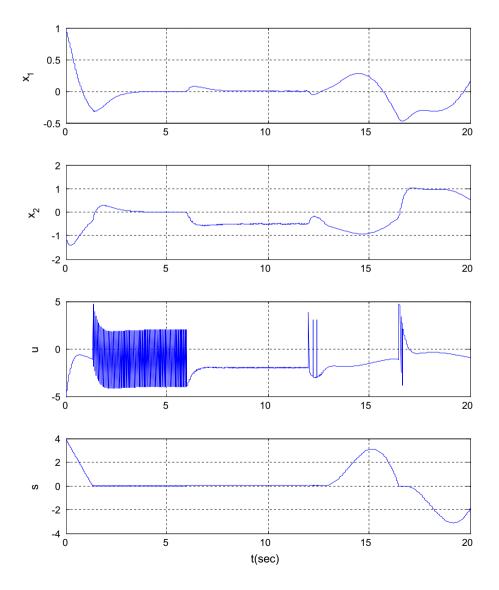The integral sliding mode control in Yang et al. (2013) used the sliding mode:

$$s = x_2 + c_1 x_1 + c_2 \int x_1, \qquad (19)$$

and the control law is:

$$u = -B^{-1}(x)[A(x) + c_1 x_2 + c_2 x_1 + k \, \mathrm{sgn}(s)]. \qquad (20)$$

The design parameters are $c_1 = 5$, $c_2 = 6$, $k = 3$. Computer software of Math Works, Matlab/Simulink, is used for simulation. A Simulink model is developed to represent the plant, while an M-file is developed to implement the control. The simulation result is shown in Fig. 2.

We can see that the output is insensitive to the constant fault for $t = 6$–12 s, but deviates away from the zero reference for the period $t = 12$–20 s.

Fig. 2 Control performance of existing method for comparison

The method developed in Yang et al. (2013) used a disturbance observer to estimate the fault, based on which a siding mode control law is derived. However, the disturbance observer can only estimate a constant disturbance $(\dot{d} = 0)$ and not a time-varying fault. This is why the output is not zero for $t = 12–20$ s. As most faults in real systems are time varying, the method in Yang et al. (2013) cannot be used for fault diagnosis and fault tolerant control of most industrial systems.

5.2 The proposed method

In the simulation of our developed neural estimator based fault tolerant control, we used a similar nonlinear system modified from system (17) used in Yang et al. (2013) to make it more complex. The modified system is as follows:

$$
\begin{aligned}
\dot{x}_1 &= x_2 + f(t), \\
\dot{x}_2 &= -x_1 - 3x_2 + e^{0.8x_1} + u \\
y &= x_1.
\end{aligned}
\tag{21}
$$

To make the simulation results to be illustrated clearer, the fault is also modified as follows. A zero reference is used for the output, and the fault is simulated as follows:

$$
f(t) = \begin{cases}
0, & 0 \leq t < 10 \text{ s} \\
2, & 10 \leq t < 20 \text{ s} \\
\sin\left[(t-20) * \dfrac{2\pi}{10}\right], & 20 \leq t < 30 \text{ s}
\end{cases}
\tag{22}
$$

Here, the three piece of time function of the fault are used to present: no fault for the first 10 s, then a constant fault of 2 for the second 10 s, and finally a sinusoidal function as time-varying function for the last 10 s.

The design follows the following design procedure.

5.2.1 Step 1: Set RBF network parameters

The network input vector is chosen as:

$$
z(k) = [x_1(k), x_2(k), \widehat{d}(k-1)].
\tag{23}
$$

10 centres are chosen to be evenly distributed in the region between $z_{d,\min} = [-1, \ -1, \ -3]^{\mathrm{T}}$ and $z_{d,\max} = [1, \ 1, \ 3]^{\mathrm{T}}$. 10 widths of the Gaussian function are $\sigma = [2, \ldots, 2]^{\mathrm{T}}$. The input data were scaled before feeding into the network by a linear scale: $z_{\text{scale}} = \frac{z - z_{\min}}{z_{\max} - z_{\min}}$.

5.2.2 Step 2: Initialization

The initial value for states is $x_1(0) = 1, x_2(0) = -1$. The initial value for network weights is chosen as a random value between 0 and 0.001.

5.2.3 Step 3: Set control parameters

The design parameters are chosen as $c_1 = 12, c_2 = 36, k = 15$, to set the two poles at $p_{1,2} = -6$. The learning rate for network weight updating is chosen as $\frac{1}{\gamma} = 0.2 (\gamma = 5)$.

In addition, to avoid chattering around the sliding surface, an approximation to the sign function is applied as follows:

$$
\text{sgn}(s) = \frac{s}{|s| + \delta}
\tag{24}
$$

with $\delta = 0.05$.

5.2.4 Step 4: Simulation

The sampling time is chosen as $T_s = 0.01$ s, and the simulation runs for 30 s. The results are displayed in Fig. 3.
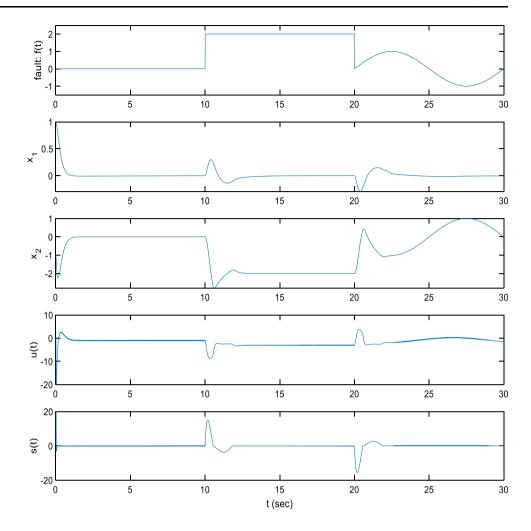
In Fig. 3, the top figure displays the fault that is applied to the system. We can see from the second figure that the output $y = x_1$ converges to the reference value of zero when the fault is zero ($t = 0–10$ s), a nonzero constant ($t = 11–20$ s), and is time varying ($t = 21–30$ s). This confirms that the proposed method is tolerant from the constant fault and time-varying fault. The third figure displays that the other state x_2 is bounded, which implies the entire system including the neural estimator is bounded and the stability of the post-fault system is guaranteed. The fourth figure displays the control signal $u(t)$, which has no chattering due to the damped sign function. At the bottom, the fifth figure displays the sliding mode $s(t)$, which indicates that the state is attracted on to the sliding mode $s = 0$ and stick on it for the whole simulation period. With the compensation of the estimated fault change rate, the system output is not affected by the occurrence of the fault. In addition, the sliding mode control guarantees the entire system stability under the appearance of the fault. Thus, the fault tolerant control is achieved.

6 Conclusions

The paper proposes a new fault diagnosis and fault tolerant control method for a class of nonlinear system. The main contribution of this paper is the tracking control of nonlinear system tolerant to time-varying fault. This is achieved by using a neural network to on-line estimate the change rate of the fault in the sliding mode control scheme. The estimator is on-line adapted to model post-fault dynamics, and the adaptation law is derived using the Lyapunov method. With the estimated fault compensation, fault tolerant control is achieved in terms of maintaining the entire system stability and closed-loop system

Fig. 3 Performance of the proposed fault tolerant control system

performance. A numerical example is simulated to evaluate the proposed method, and the results indicate the effectiveness of the method. As the method is simple and straightforward to implement, it has a great potential to be applied in real fault tolerant control for industrial systems. A limitation to the application of the method is that the fault should be a smooth time function due to the network modelling ability. Otherwise, the modelling accuracy and consequently the fault tolerance will be affected. Further work would be to investigate application to an industrial system with real data experiment.

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

Appendix

Proof of Theorem 1.

Proof

A Lyapunov function as follows is designed:

$$V = \frac{1}{2}s^2 + \frac{1}{2}\gamma c_1 \widetilde{W}^{\mathrm{T}}\widetilde{W} \tag{A1}$$

where $\widetilde{W} = W^* - \widehat{W}$, $W \in \mathcal{R}^q$ is the weight vector of the RBF estimator and $\widetilde{W}$ is the weight optimization error, γ is the learning rate for the network weight. $\qquad\square$

Generally, a Lyapunov function should be chosen such that it represents some variables that must converge to zero in the system, such as the sliding surface in (A1) (for stability) and weight estimation error in (A1) (for network convergence). In addition, the Lyapunov function must be chosen greater than zero, and its first-order derivative with respect to time can be proved to be less than zero.

From (2), we have

$$s = \dot{x}_2 + c_1\dot{x}_1 + \widehat{\dot{f}} + c_2 x_1$$
$$= Z(x) + U(x)u(t) + c_1 x_2 + c_1 f + \widehat{\dot{f}} + c_2 x_1 \tag{A2}$$

Substituting $u(t)$ in (10) into (A2),

$$s = c_1 f + u_2 \tag{A3}$$

Then, using u_2 in (10) and $\dot{f}$ in (7) we have

$$\dot{s} = \dot{u}_2 + c_1\dot{f} = -c_1\widehat{W}^\mathrm{T}\phi(z) - k*\mathrm{sgn}(s) + c_1\left[W^{*\mathrm{T}}\phi(z) + \varepsilon^*\right]$$
$$= c_1\widetilde{W}^\mathrm{T}\phi(z) - k*\mathrm{sgn}(s) + c_1\varepsilon^* \tag{A4}$$

Having prepared $\dot{s}$, we now have

$$\dot{V} = s\dot{s} + \gamma c_1 \widetilde{W}^\mathrm{T}\dot{\widetilde{W}}$$
$$= c_1\widetilde{W}^\mathrm{T}\left[s\phi(z) - \gamma\dot{\widetilde{W}}\right] - k|s| + c_1 s\varepsilon^* \tag{A5}$$

Now, we choose the updating rate of the RBF network weight $\dot{\widehat{W}}$ as,

$$\dot{\widehat{W}} = \frac{1}{\gamma}s\phi(z) \tag{A6}$$

Then, we have

$$\dot{V} = -k|s| + c_1 s\varepsilon^*$$
$$\leq -(k - c_1\overline{\varepsilon})|s| \tag{A7}$$

It can then be seen that in (A7) if k is designed such that $k > c_1\overline{\varepsilon}$, $\dot{V} < 0$ will hold true. According to the Lyapunov stability theory, V will converge to zero, implying that system state will converge to the sliding surface ($s = 0$). At the same time, the adaptive weights of the RBF estimator will converge to the optimal weight existing for the change rate of the fault. On the other hand, weight updating law in (A6) directly leads to Eq. (10) in the theorem.

References

Abu Arqub O, Abo-Hammour Z (2014) Numerical solution of systems of second-order boundary value problems using continuous genetic algorithm. Inf Sci 279:396–415

Abu Arqub O, Maayanh B (2018) Numerical solutions of integral differential equations of Fredholm operator type in the sense of the Atangana-Baleanu fractional operator. Chaos Solitons Fractals 117:117–124

Abu Arqub O, Al-Smadi M, Momani S, Hayat T (2016) Numerical solutions of fuzzy differential equations using reproducing kernel Hilbert space method. Soft Comput 20:3283–3302

Abu Arqub O, Al-Smadi M, Momani S, Hayat T (2017) Application of reproducing kernel algorithm for solving second-order, two-point fuzzy boundary value problems. Soft Comput 21(23):7191–7206

Al Janabi S (2018) Smart system to create optimal higher education environment using IDA and IOTs. Int J Comput Appl. https://doi.org/10.1080/1206212X.2018.1512460

Al Janabi S, Abaid Mahdi M (2019) Evaluation prediction techniques to achievement an optimal biomedical analysis. Int J Grid Utility Comput 45:50. https://doi.org/10.1504/ijguc.2019.10020511

Al Janabi S, Yaqoob A, Mohammad M (2020) Pragmatic Method based on intelligent big data analytics to prediction air pollution. In: Farhaoui Y (ed) Big data and networks technologies. BDNT 2019. Lecture notes in networks and systems, vol 81. Springer, Cham

Ali SH (2012) Miner for OACCR: case of medical data analysis in knowledge discovery. In: 2012 6th international conference on sciences of electronics, technologies of information and telecommunications (SETIT), Sousse, pp 962–975

Al-Janabi S (2017) Pragmatic miner to risk analysis for intrusion detection (PMRA-ID). In: Mohamed A, Berry M, Yap B (eds) Soft computing in data science (SCDS). 2017 communications in computer and information science, vol 788. Springer, Singapore

Al-Janabi S, Alkaim AF (2019) A nifty collaborative analysis to predicting a novel tool (DRFLLS) for missing values estimation. Soft Comput 2019:1–15

Al-Janabi S, Patel A, Fatlawi H, Kalajdzic K, Al Shourbaji I (2014) Empirical rapid and accurate prediction model for data mining tasks in cloud computing environments. In: IEEE 2014 international congress on technology, communication and knowledge (ICTCK), Mashhad, pp 1–8. doi:10.1109/ICTCK.2014.7033495

Al-Janabi S, Rawat S, Patel A, Al-Shourbaji I (2015) Design and evaluation of a hybrid system for detection and prediction of faults in electrical transformers. Int J Electr Power Energy Syst 67:324–335

Alkaim AF, Al Janabi S (2020) Multi objectives optimization to gas flaring reduction from oil production. In: Farhaoui Y (ed) Big data and networks technologies. BDNT 2019. Lecture notes in networks and systems, vol 81. Springer, Cham

Chen M, Chen W-H (2010) Sliding mode control for a class of uncertain nonlinear system based on disturbance observer. Int J Adapt Control Signal Process 24(1):51–64

Chen W-H, Ballance DJ, Gawthrop PJ, O'Reilly J (2000) A nonlinear disturbance observer for robotic manipulators. IEEE Trans Ind Electron 47(4):932–938

Chen M, Shi P, Lim CC (2016) Adaptive neural fault tolerant control of a 3-DOF model helicopter system. IEEE Trans Syst Man Cybern Syst 46(2):260–270

Dong G, Li Y, Sui S (2018) Fault detection and fuzzy tolerant control for complex stochastic multivariable nonlinear systems. Neurocomputing 275(1):2392–2400

Ferrari RMG, Parisini T, Polycarpou MM (2009) Distributed fault diagnosis with overlapping decompositions: an adaptive approximation approach. IEEE Trans on Autom Control 54(4):794–799

Jiang B, Xu D, Shi P, Lim CC (2014) Adaptive neural observer-based backstepping fault tolerant control for near space vehicle under control effector damage. IET Control Theory Appl 8(9):658–666

Kamal MM, Yu DW, Yu DL (2014) Fault detection and isolation for PEM fuel cell stack with independent RBF model. Eng Appl Artif Intell 28:52–63

Kayacan E, Kayakan E, Erdal Kayakan MA (2015) Identification of nonlinear dynamic systems using type-2 fuzzy neural networks: a novel learning algorithm and a comparative study. IEEE Trans Ind Electron 62(3):1716–1724

Kayacan E, Peschel JM, Chowdhary G (2017) A self-learning disturbance observer for nonlinear systems in feedback-error learning scheme. Eng Appl Artif Intell 62:276–285

Khalil HK, Grizzle J (1996) Nonlinear systems, vol 3. Prentice Hall, New Jersey

Kwan C, Lewis LF (2000) Robust backstepping control of nonlinear systems using neural networks. IEEE Trans Syst Man Cybern Part A Syst Hum 30:753–766

Li L, Ding SX, Yang Y, Zhang Y (2016) Robust fuzzy observer-based fault detection for nonlinear systems with disturbances. Neurocomputing 174(1):767–772

Liu M, Zhang L, Zheng WX (2017) Fault reconstruction for stochastic hybrid systems with adaptive discontinuous observer and non-homogeneous differentiator. Automatica 85(11):339–348

Michail K (2009) Optimized configuration of sensing elements for control and fault tolerance applied to an electro-magnetic suspension system. PhD thesis, University of Loughborough, UK

Patel A, Al-Janabi S, AlShourbaji I, Pedersen J (2015) A novel methodology towards a trusted environment in mashup web applications. Comput Secur 49:107–122

Polycarpou MM, Trunov AB (2000) Learning approach to nonlinear fault diagnosis: detectability analysis. IEEE Trans Autom Control 45(4):806–812

Polycarpou M, Zhang X, Xu R, Xu R, Yang YL, Kwan C (2004) A neural network based approach to adaptive fault tolerant flight control. In: Proceedings of the IEEE international symposium on intelligent control, pp 61–66

Shu CF (2012) Adaptive dynamic (CMAC) neural control of nonlinear chaotic systems with tracking performance. Eng Appl Artif Intell 25:997–2008

Trunov AB, Polycarpou MM (1999 June) Robust fault diagnosis of state and sensor faults in nonlinear multivariable systems. In: Proceedings of the American control conference, San Diego, California

Van M, Mavrovouniotis M, Ge SS (2018) An adaptive backstepping nonsingular fast terminal sliding mode control for robust fault tolerant control of robot manipulators. IEEE Trans Syst Man Cybern Syst PP(99):1–11

Wang SW, Yu DL (2007) A new development of IC engine air fuel ratio control. ASME J Dyn Syst Measur Control 129(6):757–766

Wang SW, Yu DL (2008) Adaptive RBF network for parameter estimation and stable air–fuel ratio control. Neural Netw 21(1):102–112

Yang J, Chen W-H, Li S (2011) Nonlinear disturbance observer based robust control for systems with mismatched disturbance/uncertainties. IET Control Theory Appl 5(18):2053–2062

Yang J, Li S, Xinghuo Yu (2013) Sliding mode control for systems with mismatched uncertainties via a disturbance observer. IEEE Trans Ind Electron 60(1):160–169

Yang Y, Ding SX, Li L (2015a) On observer-based fault detection for nonlinear systems. Syst Control Lett 82(8):18–25

Yang Q, Ge SS, Sun Y (2015b) Adaptive actuator fault tolerant control for uncertain nonlinear systems with multiple actuators. Automatica 60:92–99

Yu D, Bohliga A, Gomm B, Singha M (2014) Dynamic fault detection and Isolation for automotive engine air path by independent neural network model. Int J Eng Res 15(1):87–100

Zhang X, Polycarpou MM, Parisini T (2002) A robust detection and isolation scheme for abrupt and incipient faults in nonlinear systems. IEEE Trans Autom Control 47(4):576–593

Zhang K, Jiang B, Yan X-G, Mao Z (2016) Sliding mode observer based incipient sensor fault detection with application to high-speed railway traction device. ISA Trans 63(7):49–59