This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

Adaptive Fault-Tolerant Tracking Control for MIMO Discrete-Time Systems via Reinforcement Learning Algorithm With Less Learning Parameters

Lei Liu, Zhanshan Wang, *Member, IEEE*, and Huaguang Zhang, *Fellow, IEEE*

***Abstract*—This paper is concerned with a reinforcement learning-based adaptive tracking control technique to tolerate faults for a class of unknown multiple-input multiple-output nonlinear discrete-time systems with less learning parameters. Not only abrupt faults are considered, but also incipient faults are taken into account. Based on the approximation ability of neural networks, action network and critic network are proposed to approximate the optimal signal and to generate the novel cost function, respectively. The remarkable feature of the proposed method is that it can reduce the cost in the procedure of tolerating fault and can decrease the number of learning parameters and thus reduce the computational burden. Stability analysis is given to ensure the uniform boundedness of adaptive control signals and tracking errors. Finally, three simulations are used to show the effectiveness of the present strategy.**

***Note to Practitioners*—As the practical engineering systems are becoming more and more complex, it is more likely to suffer from faults which can lead to unpredictable behaviors and serious damages. Therefore, adaptive fault tolerant control technique plays an important role in modern engineering systems. As a matter of fact, it is always desired to minimize the maintenance cost even when some faults occur in the systems, which can reduce the energy consumption in the industrial production process. One way is to integrate the adaptive reinforcement learning algorithm into the fault tolerant controller. Furthermore, a major restriction of the learning algorithm is that a large number of parameters should be tuned online, which directly increase the computational burden. In order to tackle this difficulty, by adjusting the estimated values of the network weight vectors instead of their weights, the number of adaptive learning parameters is decreased, and the computational burden is reduced dramatically.**

***Index Terms*—Adaptive critic design, fault tolerant control, multiple-input multiple-output discrete-time systems, neural networks, reinforcement learning algorithm.**

Manuscript received August 27, 2015; revised November 09, 2015; accepted December 17, 2015. This paper was recommended for publication by Associate Editor H. Wang and Editor M. Wang upon evaluation of the reviewers' comments. This work was supported in part by the National Natural Science Foundation of China under Grant 61473070 and Grant 61433004, in part by the Fundamental Research Funds for the Central Universities under Grant N130504002, Grant N140406001, and Grant N130104001, and in part by the State Key Laboratory of Synthetical Automation for Process Industries under Grant 2013ZCX01.

The authors are with the College of Information Science and Engineering, Northeastern University, and the State Key Laboratory of Synthetical Automation for Process Industries, Shenyang, Liaoning 110819, China (e-mail: liuleill@live.cn, zhanshan_wang@163.com, hgzhang@ieee.org).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TASE.2016.2517155

I. Introduction

With the scale of practical engineering applications is becoming larger and larger, the industrial control system is becoming more and more complex. Therefore, it is necessary to improve the reliability and security of the industrial system. How to eliminate or decrease the potential dangers and risks has attracted considerable attention all over the world. Therefore, to handle the problem of actuator, sensor or other components faults in the industrial control system, fault tolerant control (FTC) and fault detection and isolation (FDI) have motivated much research (see, for instance, [1]–[14], and the references therein).

In the past two decades, the analysis and design of FTC have received significant attention. Specifically speaking, two different kinds of adaptive FTCs for linear systems with actuator faults were presented [1], [2], in which both loss of effectiveness and lock-in-place (two typical actuator fault types) were considered. Besides, an information-based fault detection, isolation and accommodation methodology was developed for a general multivariable nonlinear dynamic system in [3] by using online health monitoring module and the controller module. In a word, some recent results on the FTC or FDI have been proposed for continuous-time nonlinear systems such as Markovian jump systems [4], [5], complex networks control systems [6], large-scale interconnected nonlinear systems [7] and linear/nonlinear multi-agent systems [8].

As is known to all, a lot of practical continuous-time control systems are implemented digitally based on computer by using sampling techniques, which is a process of discretization actually. Therefore, the discrete-time systems may be more suitable to describe practical problem in control systems than the continuous-time ones. The topic on FTC for discrete-time systems has attracted considerable attention [9]–[14], [58]. A real-time fault estimation scheme was proposed in [9] to reconfigure a compensation controller such that the loss of effectiveness in the multiple-input multiple-output (MIMO) discrete-time systems could be reimbursed. Meanwhile, fault diagnosis and detection (FDD) methods were also widely established for discrete-event systems [12], [13]. Although many FTC results have been proposed, e.g., [9]–[14], and [58], in which just stabilization policy is required, these researches ignore the optimization problem of fault tolerant policy.

In the FTC field, in order to keep the production rate at acceptable levels as well as to avoid undesired hazardous situations, the investment in these systems is huge inevitably. Thus, how to achieve an optimal or near optimal FTC is of great significance. Some methods combining optimization algorithms with FTC were proposed [15]–[17]. Either maximizing a H_- performance index [15], [16] or optimizing an observer-free parameters [17] were needed in these schemes. However, the results in [15]–[17] ignored the revised actions and reciprocities of the FTC from its environment. In [18], it was pointed out that reinforcement learning (RL) algorithm had huge potential for relaxing the restriction. This motivates us to extend RL to FTC scheme.

The RL-based adaptive critic design has emerged as a powerful approach to generate an optimal or near-optimal signal. Many applications of RL in engineering have been vastly pursued in many literatures because RL does not depend on the knowledge of system dynamics (e.g., see [19] and [29]). Different from traditional neural networks (NNs) learning [55]–[57] and supervised learning (SL), in the situation of RL, the critic neural networks (CNNs) are presented to monitor the state behaviors, which are used to estimate the long-term cost function (LTCF) and to provide a satisfatory reinforcement signals to the action neural networks (ANNs). Several synonyms have been used for RL such as "adaptive critic designs" (ACDs) [20]–[22], "adaptive/approximate dynamic programming" (ADP) [23]–[25], "neuro-/neural dynamic programming" (NDP) [26], [27]. What's more, two different versions of RL-based control schemes such as output feedback control policy [29]–[31] and state feedback control policy [28], [32] were proposed in the literature. Even though the adaptive optimal controller is a key element in those literatures [18]–[32], it displays a major limitation, i.e., all those researches neglect the unknown fault dynamics in the controlled systems.

In addition, a common shortage of those RL-based algorithm is that the number of tuning parameters is very large.In many references, less learning parameter-based control policy had been developed to decrease the computation burden such as for SISO systems [34], [35] and MIMO systems [37], [38], [41]. Nevertheless, research on the less learning parameter-based method restriction in optimal FTC field should not be ignored because it is always desired to minimize the total cost and to avoid the additional or superfluous control action. Therefore, how to design an optimal or near optimal FTC strategy with less learning parameters is a matter of great significance.

Note that in the faulty case of a control system, the classical FTC scheme is usually to keep the system stable, in which the states of the systems remain bounded. In this case, a decreased performance index or cost function can be accepted. However, how to quickly respond to the fault change and try to maintain the desired cost function is an urgent task for FTC research. Based on above discussion, in this paper, we will present an RL-based FTC scheme for a class of MIMO nonlinear discrete-time systems with fault. The considered faults are both incipient faults and abrupt faults. First, in virtue of the effective capability of NNs in function approximation, ANNs and CNNs are developed to yield the optimal control strategy and the novel LTCF, respectively. These two networks may keep the cost function of system optimal in the normal operation and near optimal in the abnormal case. Second, in order to increase the learning process of NNs, it is necessary to reduce the number of adjusting parameters. To do this, instead of updating the weights of NNs, the Euclidean norm of unknown weights is estimated to update the ANNs and CNNs, respectively. Therefore, when fault occurs, the control system can keep the cost function near optimal by the revised action of ANNs online as quickly as possible.

Compared with the previous results, the main advantages of the presented policy are summarized as follows.

1) An adaptive FTC approach for MIMO nonlinear discrete-time systems is proposed by exploiting the RL algorithm. Unlike the traditional FTC schemes in [3],[4], [10], [12], and [14], the cost function of systems in this paper is considered. Meanwhile, an optimal or minimum LTCF is achieved by using the RL algorithm.
2) A quick updating law is used to reduce the number of learning parameters of the RL-based FTC policy. Different from the methods in [19] and [28]–[30] in which the estimated values of the weight vectors are directly updated, the norms of the weight vectors are estimated to update the NNs in this paper. Thus, the computation burden is further reduced.

The rest of this paper is organized as follows. In Section II, the problem statement and preliminary knowledge are introduced. By means of RL algorithm, a novel FTC policy is presented to minimize the redefined LTCF in Section III. To demonstrate the uniform boundedness, the performance analysis is presented in Section IV. In Section V, three examples are provided to demonstrate the validity of the obtained results, while the conclusions of the whole paper are given in Section VI.

II. Problem Statement and Preliminaries

A. Problem Statement

Consider a class of MIMO discrete-time systems with fault as follows:

$$\begin{cases} X_1(k+1) = X_2(k), \\ \qquad \vdots \\ X_{n-1}(k+1) = X_n(k), \\ X_n(k+1) = F(X(k)) + G(X(k))u(k) \\ \qquad\qquad +\Phi(k-k_f)H(X(k),u(k)), \\ y(k) = X_1(k), \end{cases} \tag{1}$$

where $X(k) = [X_1^T(k), X_2^T(k), \ldots, X_n^T(k)]^T \in R^{mn}$ represents the system state vector, $u(k) \in R^m$ stands for the control input, $F(X(k))$ and $G(X(k))$ are the unknown nonlinear internal dynamic and positive definite gain matrix (i.e., $G(X(k)) > 0$), respectively. $y(k) \in R^m$ is the system output. $\Phi(k - k_f)H(X(k), u(k))$ represents the deviation in the system dynamics due to a fault, and $H(X(k), u(k))$ denotes the nonlinear fault function. $\Phi(k - k_f)$ is the time profiles that is defined as follows:

$$\Phi(k-k_f) = \begin{cases} 0, & \text{if } k < k_f \\ 1 - e^{-\bar{k}_i \Delta T(k-k_f)}, & \text{if } k \geq k_f \end{cases} \tag{2}$$

where k_f means the moment that the unknown fault occurs in the systems (1), ΔT stands for the sampling period and $\bar{k}_i$ is the attenuation index of fault which usually is a constant. In general,

$\Phi(k-k_f)$ can be considered as both the incipient faults (for example, $\Phi(k-k_f)=0$ for $k<k_f$ and $0<\Phi(k-k_f)<1$ increasing monotonically for $k\geq k_f$) and the abrupt faults (for example, $\Phi(k-k_f)=0$ for $k<k_f$ and $\Phi(k-k_f)=1$ for $k\geq k_f$) [45], [46], [51]. Obviously, incipient or abrupt fault is described by the time profile $\Phi(k-k_f)$, which is entirely unrelated to the fault function $H(X(k))$. Therefore, a typical incipient fault in a control system, which is used to capture the characteristics of fault itself, is different from the incipient fault defined by (2). Basically, they are two ways to deal with the fault information.

It is worth noting that the fault time profile given by (2) only reflects the attenuation rate of the fault, while all its other basic features are captured by the function $H(X(k),u(k))$, which describes the changes in the dynamics due to the fault. Moreover, system (1) can describe many physical systems such as chemical reactors and robotic manipulator [43], [44], [52].

B. Preliminaries and Main Control Objective

Before establishing the main results, it is necessary to provide the following assumptions.

Assumption 1: The output vector $y(k)$ and the state vector $X(k)$ are all measurable and available at the kth step in the whole design procedure.

Note that Assumption 1 guarantees the availability of input and output data in system (1), which will be used to estimate the optimal control strategy and the cost function via NNs in Sections III and IV, respectively. In fact, many FTC methods have been achieved based on the same assumption (see [44]–[46]). The main work in [44]–[46] is to design some different FTC policies to stabilize the nonlinear systems under the case of faults. In contrast, the proposed RL-based FTC policy not only can stabilize the systems under the case of faults, but also can meet the requirement of optimal control performance. If $y(k)$ and $X(k)$ are not measurable, observer or filter techniques would be used to estimate the system states. The purpose here is to mainly demonstrate how the RL-based FTC controller is constructed. Therefore, we do not put much emphasis on the case of unmeasurable states.

Assumption 2: There exists a known positive constant $L>0$ such that the unknown fault function $H(X(k),u(k))$ satisfies the following condition:

$$\|H(X(k),u(k))\|\leq L\|X(k)\| \tag{3}$$

where $\|\cdot\|$ denotes the Euclidean norm.

Remark 1: The constrained condition as shown in (3) with respect to fault function satisfies the unboundedness if the state is unbounded. If the fault function is bounded by a constant, the FTC scheme can be changed into a direct robust control. As stated in [45] (see Assumption 1) and [52] (see Assumption 3 and Remark 5), in order to realize the adaptive-approximation-based fault accommodation controller, such assumption as $\|H(X(k),u(k))\|\leq\sum_{i=1}^{n}\|\psi_i(X_i(k))\|$ is required, where $\psi_i(\cdot)$ is a class-$\mathcal{K}$ functions. While this assumption is relaxed to $\|H(X_1(k),u(k))-H(X_2(k),u(k))\|\leq L_1\|X_1(k)-X_2(k)\|$ in [53], [54] for fault estimation, $L_1>0$ is a constant. Similarly, in the present paper, in order to derive the optimal FTC policy from the prescribed performance index, Assumption 2 is required. In fact, the fault function in (1) is used to describe the changes of system dynamics under the fault. Similar to the Assumptions in [53], [54], Assumption 2 is an alternative in the present paper.

Assumption 3: The reference output signal $y_d(k)\in\Omega_y$ where Ω_y is a bounded compact set, and $y_d(k)$ is a known smooth function over the bounded compact set Ω_y.

Assumptions 3 is feasible because it is determined by the designers in practical engineering and one can always choose an appropriate signal for the desired output satisfying this condition.

According to Assumption 3, in this paper, the desired state trajectory is selected as $X_d(k)=[X_{1d}(k),\ldots,X_{nd}(k)]^T$, where $X_{id}(k+1)=X_{(i+1)d}(k)$ and $y_d(k)=X_{1d}(k)$.

From Assumptions 3, one has $X_{id}(k)=X_{(i-1)d}(k+1)=X_{(i-2)d}(k+2)=\cdots=X_{1d}(k+i-1)$. Define the tracking error $z_i(k)=X_i(k)-X_{id}(k)$, and it can be rewritten as follows:

$$z_i(k)=X_i(k)-y_d(k+i-1), i=1,\ldots,n. \tag{4}$$

Lemma 1 (Young's Inequality [39], [40]): For $\forall(x,y)\in R^2$, the following inequality holds:

$$xy\leq\frac{\epsilon^p}{p}|x|^p+\frac{1}{q\epsilon^q}|y|^q$$

where $p>0, q>0, \epsilon>0$ and $(p-1)(q-1)=1$.

Our control objective is to design an online RL-based adaptive tracking FTC strategy with less learning parameters to tolerate the unknown fault dynamic for the MIMO discrete-time systems (1) as well as to satisfy the following objectives: i) the cost function, which will be defined in (10), is as small as possible such that it can generate a near optimal FTC strategy and ii) the system output $y(k)$ can quickly track a desired trajectory $y_d(k)\in R^m$ and all the variables in the closed-loop MIMO systems are uniformly bounded.

III. Reinforcement Learning-Based Adaptive Tracking FTC Design

The objective of this section is to propose an RL-based online adaptive tracking FTC policy with less learning parameters. In the design procedure, a novel LTCF will be introduced to evaluate the cost of systems. It is optimal if the value of LTCF is minimum.

Three subsections are included in this section. First, design of ANNs is introduced. Second, design of CNNs is given and the definition of LTCF will be given. Finally, the weight updating laws for ANNs and CNNs are presented, respectively.

A. Design of ANNs

In this subsection, with the occurrence of fault at step k_f, ANNs are used to approximate the near optimal control policy.

According to (4), the future values of the tracking error $z_n(k)$ is given as

$$\begin{aligned}z_n(k+1)=&F(X(k))+G(X(k))u(X(k))\\&+\Phi(k-k_f)H(X(k),u(k))-y_d(k+n).\end{aligned} \tag{5}$$

The positive constants $\lambda_{\min}(G)$ and $\lambda_{\max}(G)$ denote the minimum eigenvalue and maximum eigenvalue of gain $G(X(k))$, respectively. Hence, it satisfies $0 < \lambda_{\min}(G) \leq \lambda_{\max}(G)$. Consider the system (1) under healthy operation, the control policy for nominal system is designed as

$$u^*(k) = G^{-1}(X(k))[-F(X(k)) + y_d(k+n) + \Gamma z_n(k)] \quad (6)$$

where $\Gamma \in R^{m\times m}$ is a designed positive definite matrix, and its maximum eigenvalue is $\lambda_{\max}(\Gamma)$.

The tracking error of the closed-loop system driven by $u^*(k)$ will not converge to zero due to the occurrence of faults. Therefore, a fault compensation is required to ensure the tracking performance.

The proposed fault tolerant controller is designed as

$$u(k) = u_N(k) + u_F(k)$$

where $u_N(k) = G^{-1}(X(k))[-F(X(k)) + y_d(k+n) + \Gamma z_n(k)]$, and $u_F(k) = -G^{-1}(X(k))L\|z_n(k)\|$ is an augmented fault tolerant control law. Note that the augmented fault tolerant control component $u_F(k)$ is inspired by the previous results [45], [46]. The augmented control law $u_F(k)$ is provided here to compensate the fault satisfying Assumption 2.

In fact, the control policy $u(k)$ cannot be implemented directly in practical engineering because both nonlinear internal dynamic $F(X(k))$ and gain $G(X(k))$ are unknown. Instead, ANNs are presented to approximate the desired control policy. From Assumption 1, the desired control policy is estimated as

$$u_i(k) = \theta_{ai}^T \varphi_{ai}(S_a(k)) + \varepsilon_{ai}(S_a(k)) \quad (7)$$

where $u_i(k) (i = 1, \ldots, m)$ is the ith element of $u(k)$, $\theta_{ai} \in R^{l_a}$ is the weight value vector, $l_a > 1$ is the node number, $\varphi_{ai}(S_a(k)) \in R^{l_a}$ is the basis function of ANNs, the input of ANNs is $S_a(k) = \left[X^T(k), y_d^T(k), y_d^T(k+n)\right]^T$, and $\varepsilon_{ai}(S_a(k))$ represents the ANNs approximation error which satisfies $\varepsilon_{ai}(S_a(k)) \leq \bar{\varepsilon}_{ai}$, where $\bar{\varepsilon}_{ai}$ is a positive constant.

Hence, the RL-based optimal FTC signals are expressed as

$$u(k) = \theta_a^T \varphi_a(S_a(k)) + \varepsilon_a(S_a(k)) \quad (8)$$

where $\varphi_a(S_a(k))$ represents the basis function of ANNs and $\varphi_a(S_a(k)) = [\varphi_{a1}(S_a(k)), \ldots, \varphi_{am}(S_a(k))]^T \in R^{ml_a}$ is a column vector. In this paper, $\varphi_{ai}(S_a(k))$ is chosen as a Gaussian function

$$\varphi_{ai}(S_a(k)) = \exp\left(-\frac{(S_a(k)-\pi_i)^T(S_a(k)-\pi_i)}{v_i^2}\right)$$

$\theta_a^T \in R^{m\times ml_a}$ is a block diagonal matrix which is called the weight vector of ANNs. The detailed structure of θ_a^T is

$$\theta_a^T = \begin{bmatrix} \theta_{a1}^T & 0 & \cdots & 0 \\ 0 & \theta_{a2}^T & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & \theta_{am}^T \end{bmatrix} \in R^{m\times ml_a}$$

$\varepsilon_a(S_a(k)) = [\varepsilon_{a1}(S_a(k)), \ldots, \varepsilon_{am}(S_a(k))]^T$ is a column vector, and $\bar{\varepsilon}_a = \max\{\bar{\varepsilon}_{ai}\}, i = 1, \ldots, m$, is the upper bound of $\varepsilon_a(S_a(k))$.

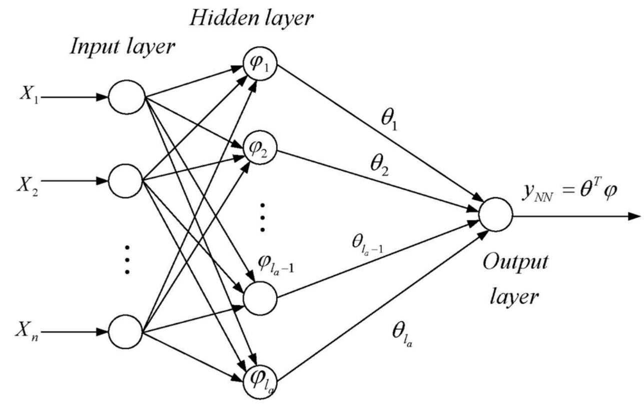

Fig. 1. The traditional neural networks.

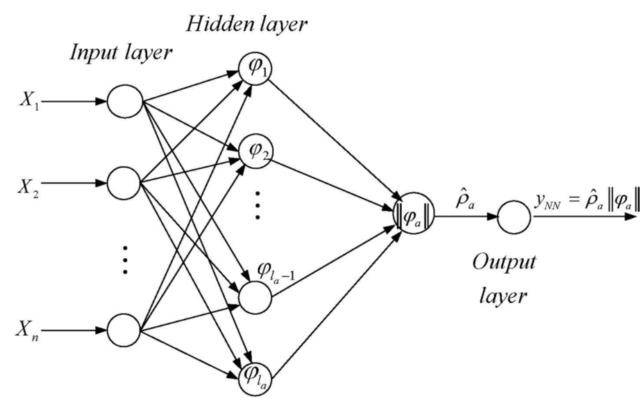

Fig. 2. The neural networks used in this paper.

In order to decrease the adjusting time in FTC scheme, the learning time of ANNs and CNNs should be reduced, which will reduce the online computation burden. In this aspect, an effective method is to adjust the estimation of the weights' Euclidean norm instead of updating the weights. Figs. 1 and 2 are provided to describe the traditional NNs and the improved NNs with less learning parameters, respectively.

Thus, the control input is designed as

$$u(k) = \hat{\rho}_a \bar{\varphi}_a(S_a(k)) \quad (9)$$

where $\hat{\rho}_a$ is the estimation of ρ_a with $\rho_a = \|\theta_a\|$ and $\bar{\varphi}_a(S_a(k)) = [\|\varphi_{a1}(S_a(k))\|, \ldots, \|\varphi_{am}(S_a(k))\|]^T$.

Remark 2: In several previous results, adaptive control theory has been paid much attention for nonlinear systems such as updating the estimation values of the fuzzy parameters [14] or the NNs' weights [19], [23], [25]. Nevertheless, the number of the hidden layer nodes of the NNs or the fuzzy rules is required to be very large in order to enhance the approximation accuracy [37]. Hence, the online learning time will tend to be very large in those schemes. However, in the faulty case, in order to realize the quick response to the fault, the FTC scheme must act as quickly as possible to tolerate the fault. In this case, FTC can guarantee the stable and reliable operation of the systems at the expense of decreased performance index. Therefore, how to reduce the learning time of the adaptive FTC is an important problem in the fault diagnosis fields. From the above design in (9), only one parameter is updated in this paper while $l_a * m$ parameters were adjusted in [19], [23], and [25] for the optimal policy. Thus, compared with the existing results, the online computation time is alleviated in this paper, which decreases the response time to tolerate the fault as quickly as possible.

B. Design of CNNs

In the research of FTC, it is an important topic to optimize the cost function or performance index of the control systems. In the previous FTC, the optimal fault tolerant controller is always designed under the prespecified performance index. In fact, the performance index is also changed in different operating mode. Therefore, how to evaluate the dynamic performance index and design the corresponding optimal fault tolerant controller simultaneously is an important problem. In the model-based FTC methods, this problem is not well done due to the complexity of the operating conditions. With the development of data-based methods, such as RL, ADP and so on, this problem can be tackled using the concept of critic and action principle. However, in the research of RL and ADP, the proposed methods are always designed for the normal control systems. Few results are for the system with fault. In general, RL or ADP methods designed for normal system are not suitable for the faulty system, this point will be verified by the numerical example in the sequel.

In this section, to meet the need of optimizing the performance index, CNNs are exploited to approximate the LTCF $J(k)$ defined in (10). Because $J(k)$ is unavailable in an online learning framework at the kth step, the CNNs are initiated online to guarantee the output closely converging to $J(k)$. The LTCF $J(k)$ is defined by

$$\begin{aligned} J(k) &= J(X(k), u(k)) \\ &= \sum_{j=k_0}^{\infty} \gamma^j [W(X(k+j)) + P^2(X(k+j)) \\ &\quad + u^T(X(k+j))Qu(X(k+j))] \end{aligned} \tag{10}$$

where $\gamma (0 \leq \gamma \leq 1)$ is the discount factor for the infinite-horizon problem, $W(X(k+j))$ is a positive semi-definite function in regard to the tracking error $z(k)$, $P(X(k+j))$ is a positive definite function satisfying $P(X(k+j)) = L\|X(k+j)\| \geq \|H(X(k+j))\|$ (see Assumption 2), and Q is a symmetric positive definite matrix.

Remark 3: It is necessary to add the discount factor γ in the LTCF (10). A requirement for minimizing the cost function (10) is that the output error is close to zero as time approaches to infinity. However, this is not easy to carry out in practical engineering, for instance, sine-cosine signals, sigmoid curve and unit step function. It implies that the meaning of minimality is dropped under this circumstance. By introducing the discount factor γ, this confined condition is relaxed. Similar approach could be found in [19] and [30].

Generally, an optimal control law can be expressed as $u^*(k) = u^*(X(k))$. Thus, the optimal LTCF can be written as $J^*(k) = J^*(X(k))$, which is a function of the current states [33]. The immediate cost in the jth step is $[W(X(k+j)) + P^2(X(k+j)) + u^T(X(k+j))Qu(X(k+j))]$, which can be regarded as a performance index in that step of system (1) with fault dynamic $\Phi(k-k_f)H(X(k), u(k))$. In fact, we are looking forward to generating an optimal control policy to minimize the total cost in the procedure of tolerating fault.

Therefore, the CNNs are exploited to estimate the cost function as follows:

$$J^*(k) = \theta_c^T \varphi_c(X(k)) + \varepsilon_c(X(k)) \tag{11}$$

where $\theta_c \in R^{l_c}, l_c > 1, \varphi_c(X(k)) \in R^{l_c}$ and $\varepsilon_c(X(k))$ are the weight vector, the node number, the basis function and the approximation error of the CNNs, respectively.

Define $\hat{J}(k)$ as the estimation of $J(k)$. Using the thought of less learning parameters, as illustrated in Fig. 2, it can be approximated by

$$\hat{J}(k) = \hat{\rho}_c^T(k) \bar{\varphi}_c(X(k)) \tag{12}$$

where $\hat{\rho}_c$ is the estimation of ρ_c with $\rho_c = \|\theta_c\|$ and $\bar{\varphi}_c(X(k)) = \|\varphi_c(X(k))\|$.

Remark 4: The complete approximation property of NNs could be found in [18], [22], and [36]. From the discussion in these references, the NNs as shown in Fig. 1 can make sure $\sup_{X(k)} \left|\theta_c^T \varphi_c(X(k)) - J^*(k)\right| < \varepsilon_c(X(k))$. According to the description in [34], [38], and [41], the NNs as given in Fig. 2 also have the approximation ability to estimate the optimal LTCF $J^*(k)$. NNs in Fig. 1 and NNs in Fig. 2 have their own advantages and disadvantages, especially the NNs in Fig. 2 is more suitable for the case where the fast learning is required. Considering the demand of FTC, quick response to the fault and fast action for the control law are the fundamental requirements. Therefore, NNs in Fig. 2 can deal with the complex cases such as variable working conditions and faulty operations. Later a numerical example will be used to show this point.

C. Updating Laws for ANNs and CNNs

In this section, the weight updating laws for ANNs and CNNs are realized based on the chain rule.

According to the discussion in Section III-A, the objective function to be minimized by the CNNs can be rewritten in a quadratic function forms [19]

$$Z_c(k) = \frac{1}{2} z_c^T(k) z_c(k) \tag{13}$$

where $z_c(k)$ denotes the prediction error for the CNNs which is described as

$$z_c(k) = \delta \hat{J}(k) - \hat{J}(k-1) + \xi(k) \tag{14}$$

and

$$\begin{aligned} \xi(k) = [&P^2(X(k)) + z_n^T(k) \Lambda z_n(k) \\ &+ u^T(X(k))Qu(X(k))] \end{aligned} \tag{15}$$

Λ is a symmetric positive definite matrix, $\delta > 0$ is a known positive constant.

In accordance with the standard gradient descent approach, the updating laws of weight vector for the CNNs are developed as

$$\hat{\rho}_c(k+1) = \hat{\rho}_c(k) + \Delta \hat{\rho}_c(k) \tag{16}$$

where

$$\Delta\hat{\rho}_c(k) = \eta_c \left[-\frac{\partial Z_c(k)}{\partial \hat{\rho}_c(k)}\right] \tag{17}$$

$\eta_c > 0$ is the adaptive gain constant.

By taking advantage of the chain rule mentioned in [19], [29] and combining (13), (16), and (17), the updating laws for CNNs are further given as

$$\hat{\rho}_c(k+1) = \hat{\rho}_c(k) - \eta_c \delta \bar{\varphi}_c(X(k)) \times \left[\delta \hat{J}(k) - \hat{J}(k-1) + \xi(k)\right]. \tag{18}$$

Next, the updating laws for ANNs will be presented by using the algorithm similar to that of CNNs. Hence, the following function will be minimized by the ANNs

$$Z_a(k) = \frac{1}{2} z_a^T(k) z_a(k) \tag{19}$$

where $z_a(k) = [z_{a1}(k), z_{a2}(k)]$. Now, the following definition are presented:

$$z_{a1}(k) = \sqrt{G(X(k))}\vartheta_a(k) \tag{20}$$

$$z_{a2}(k) = \sqrt{G(X(k))}(\overline{J}(k) - BJ_d(k)) \tag{21}$$

where $\sqrt{G(X(k))} \in R^{m\times m}$ is the mean square root of $G(X(k))$, $B = [1, 1, \ldots, 1] \in R^m$, the vector $\overline{J}(k)$ is defined as $\overline{J}(k) = [\hat{J}(k), \hat{J}(k), \ldots, \hat{J}(k)]^T \in R^m$ and $\hat{J}(k)$ is the estimation of $J(k)$ [see (12)].

By making full use of the chain rules differential equation, the following property holds:

$$\Delta\hat{\rho}_a(k) = -\eta_a \frac{\partial Z_a(k)}{\partial z_a(k)} \frac{\partial z_a(k)}{\partial \vartheta_a(k)} \frac{\partial \vartheta_a(k)}{\partial \hat{\rho}_a(k)} \tag{22}$$

where $\eta_a \in R$ is a designed positive parameter which stands for the adaptation gain.

The updating laws for ANNs, which are derived by gradient-based adaption rule [19], [29], [43], are developed as

$$\hat{\rho}_a(k+1) = \hat{\rho}_a(k) - \eta_a \bar{\varphi}_a^T(S_a(k)) \times \left[z_n(k+1) - \Gamma z_n(k) + B\hat{J}(k)\right] \tag{23}$$

where $z_n(k+1)$ is defined in (24).

Now, substituting (6) and (8)–(9) into (5), the tracking errors (5) are rewritten as

$$\begin{aligned} z_n(k+1) &= G(X(k))[\hat{\rho}_a \bar{\varphi}_a(S_a(k)) \\ &\quad - \theta_a^T \varphi_a(S_a(k))] - G(X(k))\varepsilon_a(S_a(k)) \\ &\quad + \Phi(k-k_f)H(X(k),u(k)) + \Gamma Z_n(k) \\ &= G(X(k))\vartheta_a(k) + \Gamma Z_n(k) - G(X(k)) \\ &\quad \times \varepsilon_a(S_a(k)) + \Phi(k-k_f)H(X(k),u(k)) \\ &\quad + G(X(k))[\rho_a \bar{\varphi}_a(S_a(k)) - \theta_a^T \varphi_a(S_a(k))] \end{aligned} \tag{24}$$

where $\vartheta_a(k) = \tilde{\rho}_a(k)\overline{\varphi}_a(S_a(k))$ and $\tilde{\rho}_a(k) = \hat{\rho}_a(k) - \rho_a$.

Remark 5: In contrast with traditional FTC schemes [4], [9], [10] where no optimal cost function is developed, the present paper makes it possible to reduce or minimize the cost function in the procedure of tolerating fault. As a matter of fact, a designed control method should not only guarantee the stability of nonlinear dynamic systems but also make sure the cost function to be small enough. By using the RL algorithm, a novel LTCF [see (10)], in which an additional term $P^2(\xi(k+j))$ is added, is minimized when fault occurs in the MIMO systems.

Remark 6: A lot of optimal algorithms in FTC fields were successfully studied, such as maximizing a finite horizon $H\infty$ performance index in [15], time domain approach-based necessary and sufficient condition of the parameterized performance index [16] and optimization of observer-free parameters [17]. Those methods mainly emphasized the effects of FTC in the faulty case, while the performance optimization problems were not considered. Therefore, the methods in [15]–[17] cannot apply to the changeable working conditions perfectly. In fact, in the faulty case or in the changeable working conditions, it is possible to consider the FTC and performance optimization simultaneously. This motivates us to propose the RL-based FTC with less learning parameters.

Remark 7: In [42], an advanced reconfigurable FTC policy was proposed for nonlinear systems utilizing the globalized dual heuristic programming (GDHP). However, the policy was established under the precondition that *a priori* data and knowledge stored in a dynamic model bank (DMB) was required. Because of the unavailability of the exact knowledge in some nonlinear systems, it is natural to induce great difficulties to realize the optimal objective. To cope with the challenges, the RL algorithm is presented to overcome the optimal FTC issue in this paper.

IV. Performance Analysis of RL-Based FTC

The purpose of this section is to establish the performance analysis for the proposed RL-based FTC with less learning parameters.

Theorem 1: Suppose that Assumptions 1–3 hold. Consider the discrete-time MIMO systems (1), in which the fault may occur at step k_f. The optimal FTC strategy is given in (9). Besides, the weight updating laws are chosen as (18) and (23), respectively. If the following conditions hold:

$$\begin{aligned} \lambda_{\max}^2(\Gamma) \le \frac{1}{5} - 2L^2 - 4\frac{\tau_3}{\tau_1}\lambda_{\max}(\Lambda) - 6\frac{\tau_2^2}{\tau_1^2}\lambda_{\max}^{-2}(G)L^2 \\ - 24\frac{\tau_2}{\tau_1}\eta_a l_a L^2 - 8\frac{\tau_3}{\tau_1}L^2 \end{aligned} \tag{25}$$

$$\begin{aligned} A_2 &= \tau_2\lambda_{\min}(G) - 4\tau_3\lambda_{\max}(Q) \\ &\quad - 2(\tau_1 + \tau_2^2\eta_a l_a)\lambda_{\max}^2(G) > 0 \end{aligned} \tag{26}$$

$$A_3 = \tau_3\delta^2 - \frac{m\tau_2^2}{\tau_1}\lambda_{\max}^{-2}(G) - 4m\tau_2\eta_a l_a - \tau_4 > 0 \tag{27}$$

$$\tau_4 \ge 4\tau_3, \eta_c \le \frac{1}{\delta^2 l_c} \tag{28}$$

the tracking error $z(k)$ and the updating parameters $\hat{\rho}_a(k)$ and $\hat{\rho}_c(k)$ are uniformly bounded, where τ_1, τ_2, τ_3 and τ_4 are known positive constants, $\lambda_{\max}(\Lambda)$ and $\lambda_{\max}(Q)$ are the maximum eigenvalues of Λ and Q, respectively.

Proof: Choose the following Lyapunov function:

$$V(k) = \sum_{i=1}^{4} V_i(k) \tag{29}$$

where

$$\begin{aligned}V_1(k) &= \frac{\tau_1}{5} z_n^T(k) z_n(k)\\ V_2(k) &= \frac{\tau_2}{\eta_a}\tilde{\rho}_a^2(k)\\ V_3(k) &= \frac{\tau_3}{\eta_c}\tilde{\rho}_c^2(k)\\ V_4(k) &= \tau_4 \vartheta_c^2(k-1)\end{aligned}$$

where $\vartheta_c(k) = \tilde{\rho}_c(k)\bar{\varphi}_c(x(k))$ and $\tilde{\rho}_c(k) = \hat{\rho}_c(k) - \rho_c$.

According to (2), we can know that $0 < \Phi(k - k_f) < 1$. From Assumption 2, it can be obtained that

$$\begin{aligned}&\|\Phi(k-k_f)H(X(k),u(k))\|^2\\ &\le L^2\|\Phi(k-k_f)\|^2\|X(k)\|^2\\ &\le 2L^2\|z_n(k)\|^2 + 2L^2\|y_d\|^2. \end{aligned} \tag{30}$$

For the convenience of analysis, we define

$$\begin{aligned}R(k) = &\,G(X(k))[\rho_a\bar{\varphi}_a(S_a(k)) - \theta_a^T\varphi_a(S_a(k))]\\ &- G(X(k))\varepsilon_a(S_a(k))\\ &+ \Phi(k-k_f)H(X(k),u(k)). \end{aligned} \tag{31}$$

Using the following fact [37]:

$$\left(\sum_{i=1}^n a_i\right)^2 \le n\sum_{i=1}^n a_i^2$$

one gets

$$\begin{aligned}&\|z_n(k+1)\|^2\\ &= \|G(X(k))\vartheta_a(k) + \Gamma z_n(k) + R(k)\|^2\\ &\le 5\lambda_{\max}^2(G)\|\vartheta_a(k)\|^2 + 5\lambda_{\max}^2(\Gamma)\|z_n(k)\|^2\\ &\quad + 5\lambda_{\max}^2(G)\bar{\varepsilon}_a^2 + 10L^2\|z_n(k)\|^2 + 10L^2\|y_d\|^2\\ &\quad + 10\lambda_{\max}^2(G)l_a\rho_a^2. \end{aligned} \tag{32}$$

From (24), the difference of $V_1(k)$ can be obtained as

$$\begin{aligned}\Delta V_1(k) &= \frac{\tau_1}{5}\|z_n(k+1)\|^2 - \frac{\tau_1}{5}\|z_n(k)\|^2\\ &\le -\frac{\tau_1}{5}\left(1 - 5\lambda_{\max}^2(\Gamma)\right)\|z_n(k)\|^2\\ &\quad + \tau_1\lambda_{\max}^2(G)\|\vartheta_a(k)\|^2 + \tau_1\lambda_{\max}^2(G)\bar{\varepsilon}_a^2\\ &\quad + 2\tau_1L^2\|z_n(k)\|^2 + 2\tau_1L^2\|y_d\|^2. \end{aligned} \tag{33}$$

According to (23), we can get $\Delta V_2(k)$ as follows:

$$\begin{aligned}\Delta V_2(k) = &\,\frac{\tau_2}{\eta_a}\tilde{\rho}_a^2(k+1) - \tilde{\rho}_a^2(k)\\ = &-2\tau_2\vartheta_a^T(k)G(X(k))\vartheta_a(k) - 2\tau_2\vartheta_a^T(k)R(k)\\ &- 2\tau_2\vartheta_a^T(k)B\hat{J}(k) + \tau_2\eta_a\|\varphi_a(S_a(k))\|^2\\ &\times\|G(X(k))\vartheta_a(k) + R(k) + B\hat{J}(k)\|^2. \end{aligned} \tag{34}$$

According to the definition of $\vartheta_c(k)$, $\hat{J}(k)$ can be rewritten as

$$\hat{J}(k) = \vartheta_c(k) + \rho_c\bar{\varphi}_c(X(k)). \tag{35}$$

Using Lemma 1, it has

$$\begin{aligned}&-2\tau_2\vartheta_a^T(k)G(X(k))\vartheta_a(k)\\ &\qquad\le -2\tau_2\lambda_{\min}(G)\|\vartheta_a(k)\|^2,\\ &-2\tau_2\vartheta_a^T(k)R(k) \le \frac{\tau_2^2}{\tau_1}\lambda_{\max}^{-2}(G)(3\lambda_{\max}^2(G)\bar{\varepsilon}_a^2\\ &\qquad + 6L^2\|z_n(k)\|^2 + 3\lambda_{\max}^2(G)l_a\rho_a^2\\ &\qquad + 6L^2\|y_d\|^2 + \tau_1\lambda_{\max}^2(G)\|\vartheta_a(k)\|^2,\\ &-2\tau_2\vartheta_a^T(k)B\vartheta_c(k)\\ &\qquad\le \frac{m\tau_2^2}{\tau_1}\lambda_{\max}^{-2}(G)\vartheta_c^2(k) + \tau_1\lambda_{\max}^2(G)\|\vartheta_a(k)\|^2, \end{aligned} \tag{36}$$

$$\begin{aligned}&-2\tau_2\vartheta_a^T(k)B\rho_c\varphi_c(X(k))\\ &\qquad\le \frac{m\tau_2^2}{\tau_1}\lambda_{\max}^{-2}(G)l_c\rho_c^2 + \tau_1\lambda_{\max}^2(G)\|\vartheta_a(k)\|^2. \end{aligned} \tag{37}$$

Then, $\Delta V_2(k)$ becomes

$$\begin{aligned}\Delta V_2(k) \le &-\tau_2\left[2\lambda_{\min}(G) - \left(\frac{3\tau_1}{\tau_2} + 4\eta_a l_a\right)\lambda_{\max}^2(G)\right]\\ &\times\|\vartheta_a(k)\|^2 + \left(\frac{m\tau_2^2}{\tau_1}\lambda_{\max}^{-2}(G) + 4m\tau_2\eta_a l_a\right)\\ &\times\|\vartheta_c(k)\|^2 + 3\left(\frac{\tau_2^2}{\tau_1}\lambda_{\max}^{-2}(G) + 4\tau_2\eta_a l_a\right)\\ &\times(\lambda_{\max}^2(G)\bar{\varepsilon}_a^2 + 2L^2\|z_n(k)\|^2 + 2L^2\|y_d\|^2)\\ &+ \frac{m\tau_2^2}{\tau_1}l_c{\rho_c}^2\lambda_{\max}^{-2}(G) + 4m\tau_2\eta_a l_a l_c{\rho_c}^2\\ &+ 3\lambda_{\max}^2(G)l_a\rho_a^2(1 + 4\tau_2\eta_a l_a). \end{aligned} \tag{38}$$

According to (18), the difference of $V_3(k)$ is

$$\begin{aligned}\Delta V_3(k) &= \frac{\tau_3}{\eta_c}\tilde{\rho}_c^2(k+1) - \frac{\tau_3}{\eta_c}\tilde{\rho}_c^2(k)\\ &= -2\tau_3\delta\vartheta_c(k)z_c(k)\\ &\quad + \tau_3\eta_c\delta^2\|z_c(k)\|^2\|\varphi_c(X(k))\|^2\\ &\le -\tau_3(1 - \eta_c\delta^2 l_c)z_c^2(k)\\ &\quad - \tau_3\delta^2\|\vartheta_c(k)\|^2 + \tau_3[z_c(k) - \delta\vartheta_c(k)]^2. \end{aligned} \tag{39}$$

Based on the definition of $\xi(k)$, one gets

$$\begin{aligned}\xi^2(k) &\le P^2(X(k)) + \lambda_{\max}(\Lambda)\|z_n(k)\|^2\\ &\quad + \lambda_{\max}(Q)\|u(k)\|^2\\ &\le \lambda_{\max}(\Lambda)\|z_n(k)\|^2 + P^2(X(k))\\ &\quad + 2\lambda_{\max}(Q)(\|\vartheta_a(k)\|^2 + \rho_a^2 l_a). \end{aligned} \tag{40}$$

Substituting (35) into (14), one has

$$\begin{aligned}z_c(k) = &\,\delta\vartheta_c(k) - \vartheta_c(k-1) + \delta\rho_c\varphi_c(X(k))\\ &- \rho_c\varphi_c(X(k-1)) + \xi(k) \end{aligned} \tag{41}$$

where

$$\begin{aligned}&|\delta\rho_c\varphi_c(X(k)) - \rho_c\varphi_c(X(k-1))|\\ &\qquad\le \delta\rho_c\sqrt{l_c} + \rho_c\sqrt{l_c} \triangleq \zeta. \end{aligned} \tag{42}$$

Considering (41), it leads to

$$\begin{aligned}&\tau_3[z_c(k)-\delta\vartheta_c(k)]^2\\&=\tau_3[\delta\rho_c\varphi_c(X(k))-\rho_c\varphi_c(X(k-1))-\vartheta_c(k-1)+\xi(k)]^2\\&\leq 4\tau_3\zeta^2+4\tau_3\vartheta_c^2(k-1)+4\tau_3\xi^2(k).\end{aligned}$$

Then, $\Delta V_3(k)$ becomes

$$\begin{aligned}\Delta V_3(k)\leq&-\tau_3(1-\eta_c\delta^2 l_c)\|z_c(k)\|^2+8\tau_3L^2\|y_d\|^2\\&+8\tau_3L^2\|z_n(k)\|^2-\tau_3\delta^2\vartheta_c^2(k)+4\tau_3\zeta^2\\&+4\tau_3\vartheta_c^2(k-1)+4\tau_3\lambda_{\max}(\Lambda)\|z_n(k)\|^2\\&+8\tau_3\lambda_{\max}(Q)(\|\vartheta_a(k)\|^2+\rho_a^2 l_a).\end{aligned}\tag{43}$$

$\Delta V_4(k)$ can be obtained as

$$\Delta V_4=\tau_4(\vartheta_c^2(k)-\vartheta_c^2(k-1)).\tag{44}$$

Combining (33), (38), (43), and (44), it has

$$\begin{aligned}\Delta V(k)\leq&-A_1\|z_n(k)\|^2-2A_2\|\vartheta_a(k)\|^2\\&-A_3\vartheta_c^2(k)-A_4\vartheta_c^2(k-1)\\&-A_5\|z_c(k)\|^2+A_6\end{aligned}\tag{45}$$

where

$$\begin{aligned}A_1=&\frac{\tau_1}{5}-\tau_1\lambda_{\max}^2(\Gamma)-2\tau_1L^2-4\tau_3\lambda_{\max}(\Lambda)\\&-\frac{6\tau_2^2}{\tau_1}\lambda_{\max}^{-2}(G)L^2-24\tau_2\eta_a l_a L^2-8\tau_3L^2\\A_4=&\tau_4-4\tau_3,\ A_5=\tau_3(1-\eta_c\delta^2 l_c)\\A_6=&\frac{3\tau_2^2}{\tau_1}\bar{\varepsilon}_a^2+\frac{m\tau_2^2}{\tau_1}l_c\lambda_{\max}^{-2}(G){\rho_c}^2+12\tau_2\eta_a l_a\lambda_{\max}^2(G)\bar{\varepsilon}_a^2\\&+4\tau_3\zeta^2+4m\tau_2\eta_a l_a l_c{\rho_c}^2+8\tau_3L^2\|y_d\|^2\\&+24\tau_2\eta_a l_a L^2\|y_d\|^2+\tau_1\lambda_{\max}^2(G)\bar{\varepsilon}_a^2+2\tau_1L^2\|y_d\|^2\\&+\frac{6\tau_2^2}{\tau_1}\lambda_{\max}^{-2}(G)L^2\|y_d\|^2+8\tau_3\lambda_{\max}(Q)\rho_a^2 l_a\\&+3\lambda_{\max}^2(G)l_a\rho_a^2(1+4\tau_2\eta_a l_a)\end{aligned}$$

and A_2, A_3 are defined in (26) and (27).

By choosing $\tau_4\geq 4\tau_3$ and $\eta_c\leq 1/\delta^2 l_c$ [see (29)], it can result in $A_4>0$ and $A_5>0$. This further follows that:

$$\begin{aligned}\Delta V(k)\leq-A_1\|z_n(k)\|^2-2A_2\|\vartheta_a(k)\|^2&\\-A_3\vartheta_c^2(k)+A_6.&\end{aligned}\tag{46}$$

Substituting (25) into A_1, it yields $A_1>0$. Considering (25)–(27), it can be concluded that $A_2>0$ and $A_3>0$. Based on the Lyapunov stability theory, one can deduce that $\Delta V(k)<0$ if at least one of the following inequalities hold:

$$\|z_n(k)\|>\sqrt{\frac{A_1}{A_6}}$$

$$\|\vartheta_a(k)\|>\sqrt{\frac{2A_2}{A_6}}$$

or

$$\vartheta_c(k)>\sqrt{\frac{A_3}{A_6}}.$$

Therefore, the tracking error $\|z_n(k)\|$ and the weight estimate errors $\vartheta_a(k)$ and $\vartheta_c(k)$ are bounded. Consider $z_i(k)=X_i(k)-y_d(k+i-1), i=1,\ldots,n$, we can draw that $z_i(k)$ is also bounded. Besides, due to the boundedness of $\vartheta_a(k)$ and $\vartheta_c(k)$, the weight estimations $\hat{\rho}_a(k)$ and $\hat{\rho}_c(k)$ are also bounded. This completes the proof of **Theorem 1**.

Note that, because two neural networks are used in this paper, how to choose those parameters in neural networks plays an important role in the RL-based FTC policy. Therefore, based on Lyapunov stability analysis approach, we have presented Theorem 1 to show how to choose those parameters. The emphasis in this paper is not placed on the stability analysis of the control systems.

Remark 8: In this paper, the abrupt fault and incipient fault are defined by the time profiles (2). A small value of $\bar{k}_i$ in (2) implies that the considered fault is an incipient fault. Specifically, an incipient fault model can be written as $\left(1-e^{-\bar{k}_i\Delta T(k-k_f)}\right)H(X(k),u(k))$, where $\bar{k}_i$ is a sufficiently small constant. It can be seen that the value of the first term in the incipient fault model is restricted in the interval $(0,1)$. In the designed fault tolerant tracking control scheme, this case is dealt with in (30). Therefore, the fault tolerant tracking control method proposed in this paper can be applied to the case of incipient fault, too.

V. Simulation Results

1) Example 1: To demonstrate the feasibility of the presented RL-based FTC strategy with less learning parameters, we consider the following MIMO systems:

$$\begin{cases}X_1(k+1)=X_2(k)\\X_2(k+1)=F(X(k))+G(X(k))u(k)\\\qquad+\Phi(k-k_f)H(X(k),u(k))+d(k)\\y(k)=X_1(k)\end{cases}\tag{47}$$

where $X_1(k)=[X_{11}(k),X_{12}(k)]$ and $X_2(k)=[X_{21}(k),X_{22}(k)]$ are the system states, $u(k)\in R^2$ is the system input, $y(k)=[y_1(k),y_2(k)]\in R^2$ is the system output, $d(k)=[d_1(k),d_2(k)]$ is the random disturbance which is generated in the interval $[0,1]$. The nonlinear internal dynamic $F(X(k))$ is defined as

$$F(X(k))=\begin{bmatrix}\frac{0.4X_{11}(k)}{(1+X_{21}^2(k))}\\(0.1+0.05\cos(X_{12}(k)))X_{22}(k)\end{bmatrix}^T$$

and the gain matrix $G(X(k))=[2\quad 0;\quad 0\quad 2]$. In this simulation, the tracking error is defined as $e_{1i}(k)=y_i(k)-y_{di}(k)$ with $i=1,2$.

The main objective is to develop an online RL-based FTC policy for the system (47) with less learning parameters such that: 1) the output $y(k)$ tracks the reference signal defined as

$$y_d(k)=\begin{bmatrix}y_{d1}(k)\\y_{d2}(k)\end{bmatrix}=\begin{bmatrix}0.3\sin(\frac{0.5k\pi}{50}+\frac{\pi}{4})\\0.2\cos(\frac{0.5k\pi}{50}-\frac{\pi}{4})\end{bmatrix}$$

to a small bounded compact set and 2) all the signals in the MIMO systems are bounded.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

We assume that the faults occur in the MIMO systems at step $k_f = 500$. The faults dynamics are given as

$$\Phi_1(k-k_f)H_1(X(k),u(k)) = \begin{cases} 0, & \text{if } k < k_f \\ 0.78(1-e^{-0.15(k-k_f)}) \times X_{11}(k), & \text{if } k \geq k_f \end{cases} \quad (48)$$

$$\Phi_2(k-k_f)H_2(X(k),u(k)) = \begin{cases} 0, & \text{if } k < k_f \\ 0.42(1-e^{-0.11(k-k_f)}) \times X_{12}(k), & \text{if } k \geq k_f \end{cases}. \quad (49)$$

In the simulation, the widths and nodes of ANNs are 3 and 15, while those of CNNs are 3 and 25, respectively. The input variables of ANNs are $S_a(k) = [X_1^T(k), X_2^T(k), y_d^T(k), y_d^T(k+n)]^T$, and the input variables of CNNs are $X(k) = [X_1^T(k), X_2^T(k)]^T$. The initial values of the states $X(k) = [X_1^T(k)X_2^T(k)] \in R^{2\times 2}$ are chosen as $X_1(0) = [-1, 1]$ and $X_2(0) = [-0.5, 1]$. The initial conditions of the adaptive laws are chosen as $\hat{\rho}_a(1) = [0.2, 0.25]^T$ and $\hat{\rho}_c(1) = 0.1$. In addition, the parameters decided by the designers are selected as $\eta_a = 0.05$, $\eta_c = 0.02$, $\delta = 0.3$, and the diagonal matrix $\Gamma = diag\{0.2, 0.2\}$, $\Lambda = diag\{0.02, 0.02\}$, $Q = diag\{0.5, 0.5\}$.

In order to investigate the effectiveness of the proposed approach, three cases are discussed in this example.

Case (a) The traditional direct adaptive control methods (backstepping technique) as mentioned in [36] are used in this example.

Case (b) Standard reinforcement learning algorithms in [19], in which the ANNs and CNNs are introduced without reducing learning parameters, are used in this example.

Case (c) In contrast to Case (b), our proposed RL-based FTC policies are taken into account. The estimation of the norm of the NNs' weight vector is updated.

The simulation results for Case (a) are depicted in Figs. 3– 6, respectively. Note that only a single neural network is needed in this case. The tracking trajectories of $y_1(k)$ and $y_{d1}(k)$ as well as $y_2(k)$ and $y_{d2}(k)$ are all unsatisfactory by viewing the Figs. 3 and 4, respectively. Meanwhile, the corresponding controllers $u_1(k)$ and $u_2(k)$ are shown in Figs. 5 and 6, respectively. Notice that the NN approximators in Case (a) are identical to the ANNs mentioned in Cases (b) and (c) and the same initial values are chosen.

For Case (b), the simulations are presented in Figs. 7–10, respectively. Fig. 7 shows the tracking effect for $y_1(k)$ and $y_{d1}(k)$ as well as $e_{11}(k)$. Fig. 8 illustrates the system tracking trajectories of $y_2(k)$ and $y_{d2}(k)$ as well as $e_{12}(k)$. The traditional RL-based control signals are given in Figs. 9 and 10. It can be seen that the tracking performance is unsatisfactory.

For Case (c), the simulation results are presented in Figs. 11–14, respectively. The system tracking trajectories of $y_1(k)$ and $y_{d1}(k)$ are depicted in Fig. 11. Fig. 12 presents the system tracking trajectories of $y_2(k)$ and $y_{d2}(k)$. The wave phenomenon between 500th and 550th is normal, because the toleration procedure needs a certain time. The trajectories of the RL-based FTC policies $u_1(k)$ and $u_2(k)$ with less learning parameters are described in Figs. 13 and 14. It can be seen that a good tracking

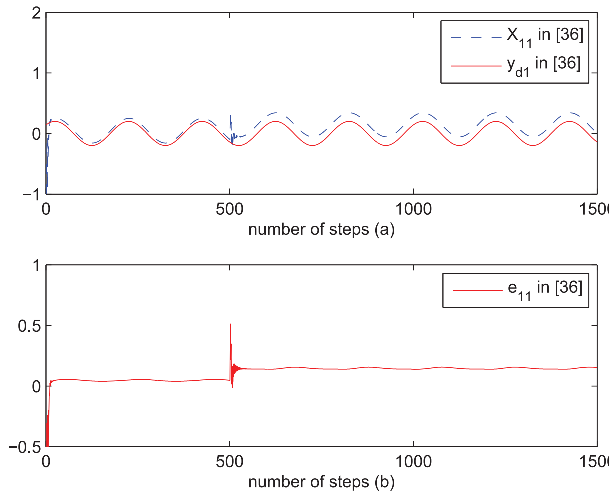

Fig. 3. The curves of $y_{d1}(k)$ and $X_{11}(k)$ using the adaptive control in [36].

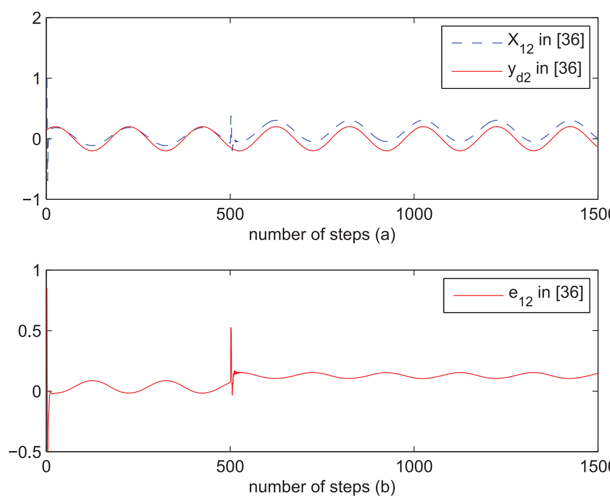

Fig. 4. The curves of $y_{d2}(k)$ and $X_{12}(k)$ using the adaptive control in [36].

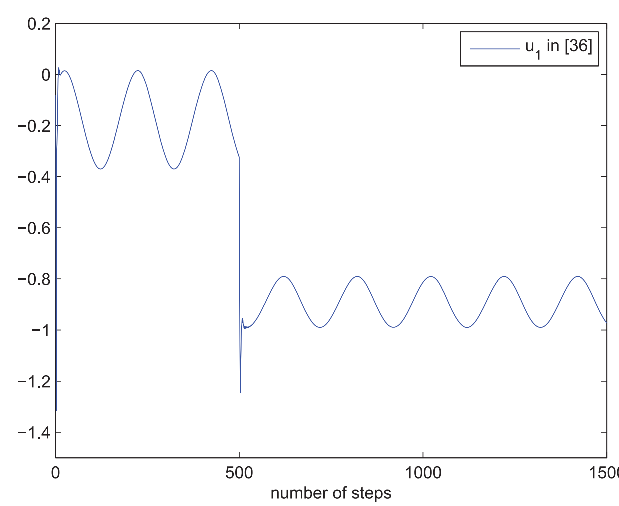

Fig. 5. The control signals $u_1(k)$ using the adaptive control in [36].

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

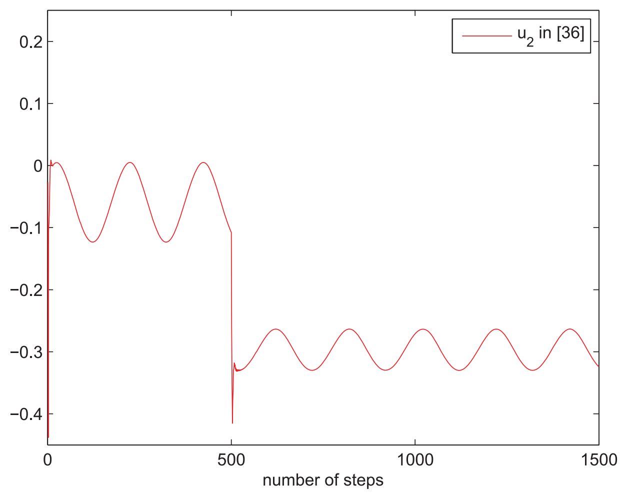

Fig. 6. The control signals $u_2(k)$ using the adaptive control in [36].

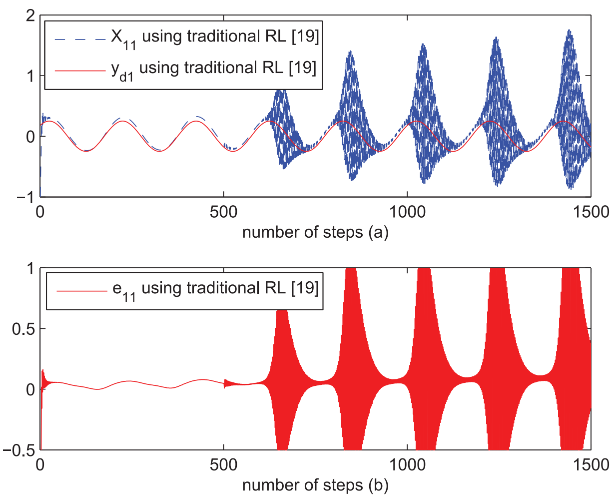

Fig. 7. The curves of $y_{d1}(k)$, $X_{11}(k)$ and $e_{11}(k)$ using the traditional RL in [19].

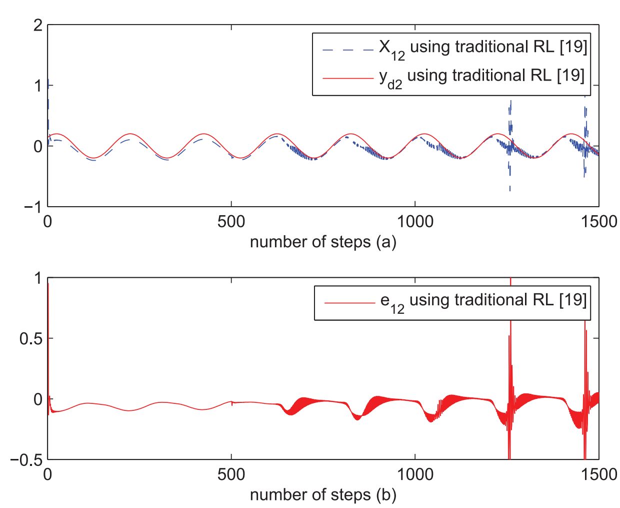

Fig. 8. The trajectories of $y_{d2}(k)$, $X_{12}(k)$ and $e_{21}(k)$ using the traditional RL in [19].

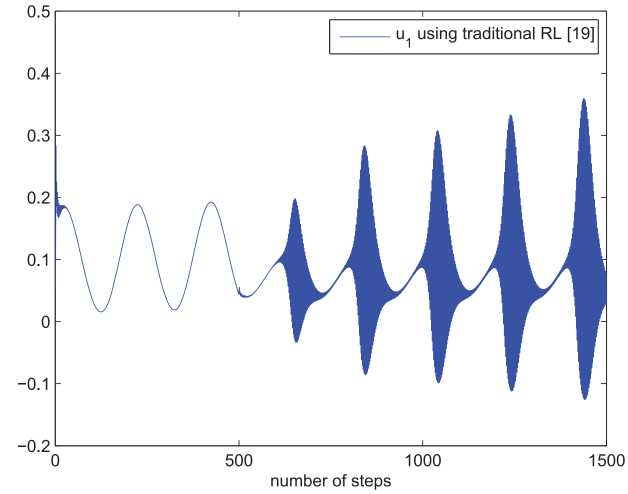

Fig. 9. The control policy $u_1(k)$ using the traditional RL in [19].

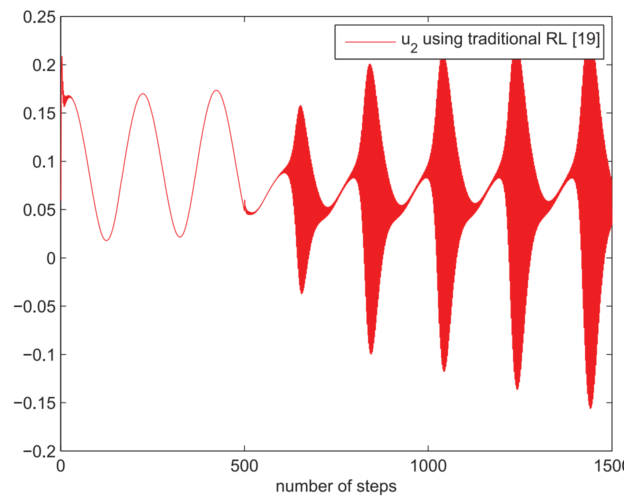

Fig. 10. The control policy $u_2(k)$ using the traditional RL in [19].

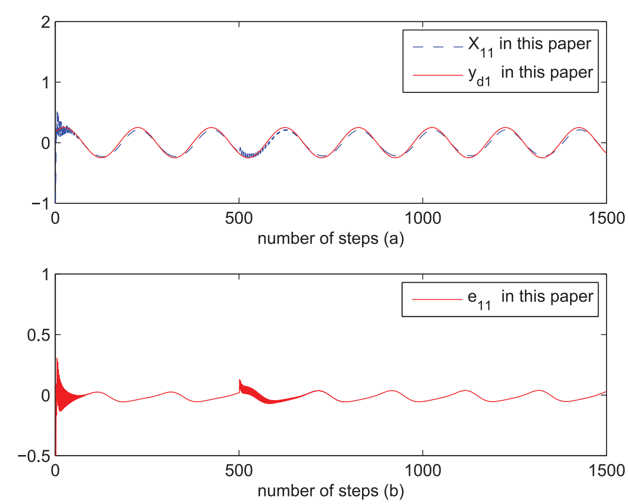

Fig. 11. The trajectories of $y_{d1}(k)$ and $X_{11}(k)$ as well as $e_{11}(k)$ using the proposed RL-based FTC.

performance is obtained for the output $y_1(k)$ and $y_2(k)$ since the tracking errors $e_{11}(k)$ and $e_{12}(k)$ are almost equal to zero.

Comparing these three cases, the tracking performance in Case (c) is the best. By looking into the curves in Figs. 5–6,

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

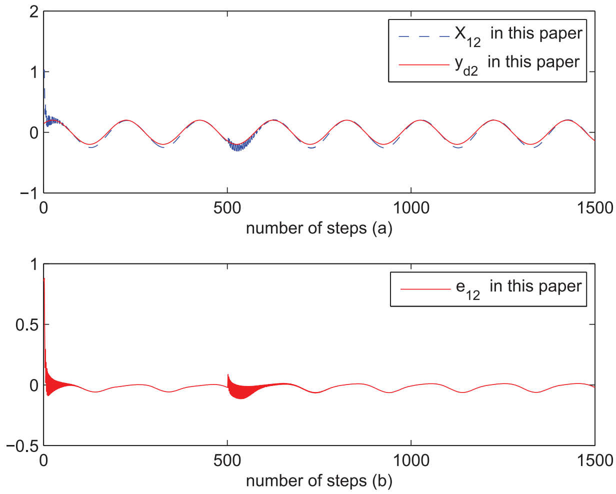

Fig. 12. The trajectories of $y_{d2}(k)$ and $X_{12}(k)$ as well as $e_{12}(k)$ using the proposed RL-based FTC.

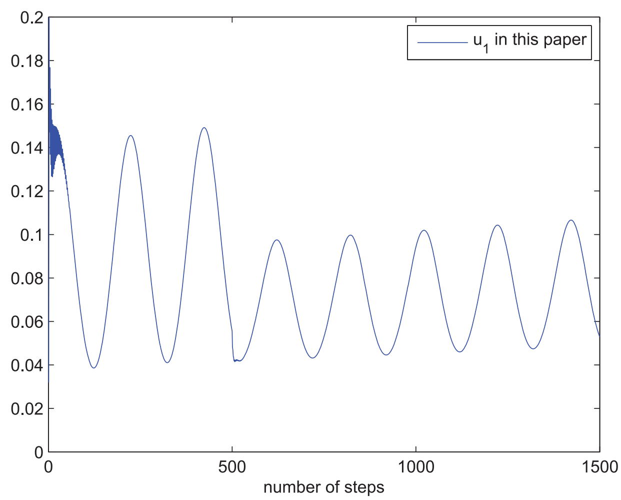

Fig. 13. The trajectory of $u_1(k)$ using the proposed RL-based FTC.

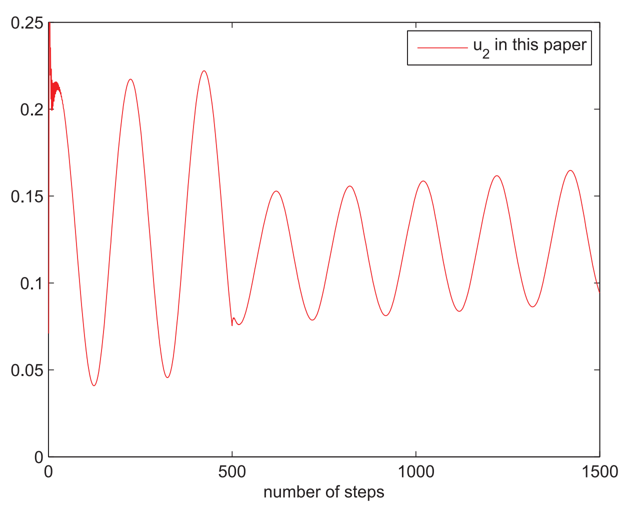

Fig. 14. The trajectory of $u_2(k)$ using the proposed RL-based FTC.

9–10, and 13–14, the magnitude of control signals $u_1(k)$ and $u_2(k)$ in Case (c) is smaller than those in Cases (a) and (b).

TABLE I
PERFORMANCE COMPARISONS FOR THE THREE CASES

	Case (a) (Ref. [36])	Case (b) (Ref. [19])	Case (c) (This paper)
The values of total cost	46.5304	81.8627	37.0362
The number of the required NNs	1	2	2
Learning parameters in each step	15	15+25	1+1
The operation time of simulation program	0.771s	1.548s	0.354s

In addition, we calculate the values of total cost $P^2(X(k)) + {z_n}^T(k)\Lambda z_n(k) + u^T(k)Qu(k)$ in the first 1500 steps. It is 46.5304 and 37.0362 corresponding to Cases (a) and (c) but 81.8627 in Case (b). Thus, the cost function is decreased based on the approaches in Case (c). Moreover, the operation time of the proposed simulation program for the three cases is also calculated (see Table I). According to this Table, it can be concluded that the online computation time in Case (c) is less than those in Cases (a) and (b).

2) Example 2: To further validate the proposed RL-based FTC, we consider the two-link planar robot manipulator system [43]

$$\begin{bmatrix} D_{11}(q_1) & D_{12}(q_1) \\ D_{12}(q_1) & D_{22}(q_1) \end{bmatrix} \begin{bmatrix} \ddot{q}_2 \\ \ddot{q}_1 \end{bmatrix} = \begin{bmatrix} F_{12}(q_1)\dot{q}_1^2 + 2F_{12}(q_1)\dot{q}_2\dot{q}_1 \\ -F_{12}(q_1)\dot{q}_2^2 \end{bmatrix} + \begin{bmatrix} b_1(q_2,q_1)g \\ b_2(q_2,q_1)g \end{bmatrix} + \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \tag{50}$$

where $D_{11}(q_1) = (m_1+m_2)(r_1)^2 + 2m_2r_1r_2\cos(q_1) + J_1 + m_2(r_2)^2$, $D_{12}(q_1) = m_2(r_2)^2 + m_2r_1r_2\cos(q_1)$, $D_{22}(q_1) = J_2+m_2(r_2)^2$, $b_1(q_2,q_1) = -\cos(q_1+q_2)m_2r_2-(m_1+m_2)\times r_1\cos(q_2)$ and $b_2(q_2,q_1) = -m_2r_2\cos(q_1+q_2)$. The two-link planar robot arm system is shown in Fig. 15. In practical manipulator systems, the acceleration of gravity is $g = 9.8\ \mathrm{m/s}^2$, $m_1 = 0.8$ kg and $m_2 = 6$ kg stand for the point masses at the end, $r_1 = 0.9$ m and $r_2 = 1.2$ m represent the arm length. The rotational angles are q_1 and q_2, which are the states here. The control inputs are u_1 and u_2, while they are also the torques in practical manipulator. The discretization technique is same as that in [43] where the time step is $\Delta T = 0.05$ s.

Since the spring pressure may be too large or too small, or the nut may be loose, or the arm may wear out or be deformed, fault (the change of rotational angles) will occur in the two-link planar robot arm system. This just verifies that the fault is the function of states. We assume that the faults occur in the system at step $k_f = 500$. The faults dynamics are given as

$$\Phi_1 H_1 = \begin{cases} 0, & \text{if } k < k_f \\ 1.4(1-e^{-0.2(k-k_f)})q_1(k), & \text{if } k \geq k_f \end{cases} \tag{51}$$

$$\Phi_2 H_2 = \begin{cases} 0, & \text{if } k < k_f \\ 0.18(1-e^{-0.11(k-k_f)})q_2(k), & \text{if } k \geq k_f. \end{cases} \tag{52}$$

The desired trajectory of rotational angles are $q_{1d}(k) = 2.04 + 0.08(\cos(k/100\pi) + 0.05)$ and $q_{2d}(k) = 1.32 + 0.012(\sin(k/100\pi))$. The initial values are chosen as $q_1(0) = q_2(0) = 0.1$ and the parameters are chosen as

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

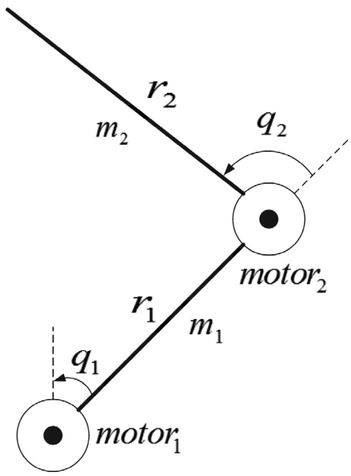

Fig. 15. Two-link planar robot arm.

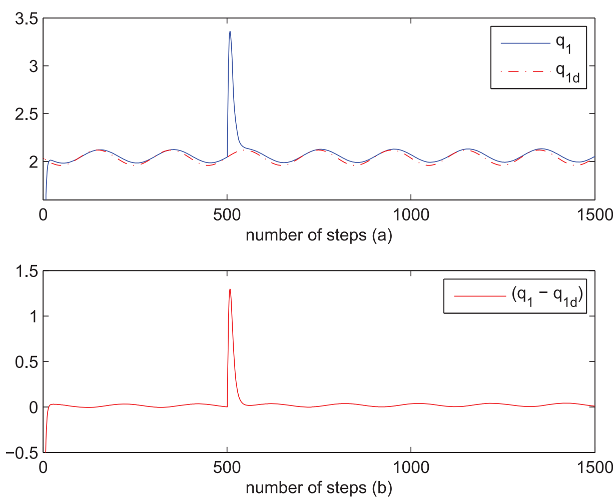

Fig. 16. The trajectory of q_1, q_{1d} and tracking error in Example 2.

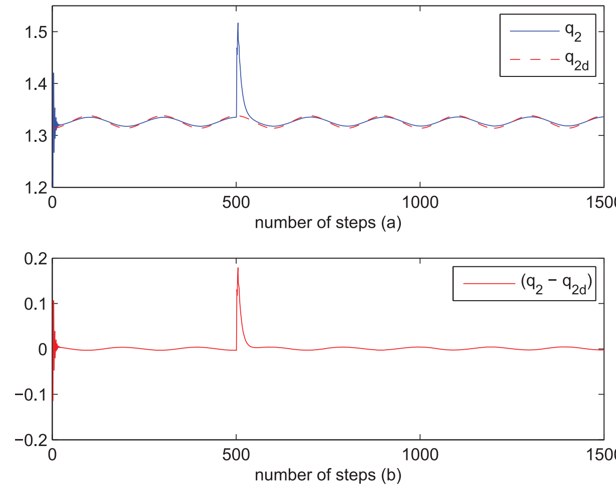

Fig. 17. The trajectory of q_2, q_{2d} and tracking error in Example 2.

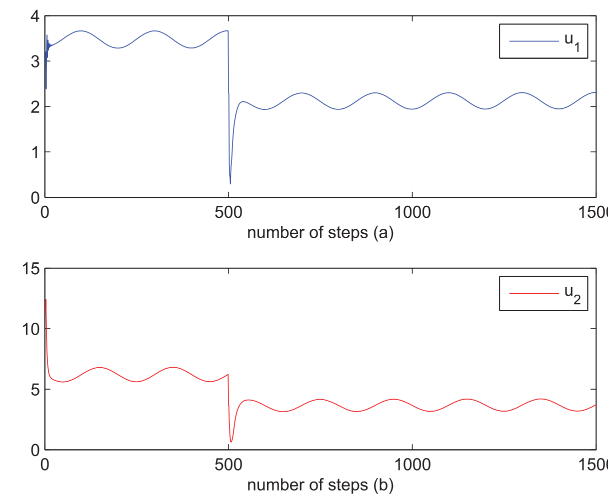

Fig. 18. The trajectory of u_1 and u_2 in Example 2.

$J_1 = 4, J_2 = 5$. Using the same design method in Example 1 Case C, the parameters of NNs are also the same.

The simulation results are shown in Figs. 16–18, respectively. As seen in Fig. 16, the tracking error converges to a small neighborhood of the origin. Fig. 17 shows the tracking trajectories of q_2, q_{2d} and tracking errors, respectively. It can be concluded that good tracking performances are achieved. The trajectories of the controllers u_1 and u_2 are given in Fig. 18. It can be seen that they are bounded.

3) Example 3: In order to highlight the effectiveness of the developed RL-based FTC method for incipient fault, which is changed gradually, let us recall Example 1. We consider the MIMO systems as shown in (47). Suppose that the nonlinear internal dynamic $F(X(k))$, $G(X(k))$ and the reference signal $y_d(k)$ are the same as in Example 1.

We assume that the incipient faults occur in the MIMO system at step $k_f = 480$. The time profiles are defined as

$$\Phi_1(k-k_f) = \begin{cases} 0, & \text{if } k < k_f \\ 1 - e^{-0.0075(k-k_f)}, & \text{if } k \geq k_f \end{cases} \quad (53)$$

$$\Phi_2(k-k_f) = \begin{cases} 0, & \text{if } k < k_f \\ 1 - e^{-0.006(k-k_f)}, & \text{if } k \geq k_f \end{cases}. \quad (54)$$

The fault functions are expressed as

$$H_1(\cdot) = \begin{cases} 0, & \text{if } k < k_f \\ X_{11}(k)X_{12}(k) + |X_{11}(k)|, & \text{if } k \geq k_f \end{cases} \quad (55)$$

$$H_2(\cdot) = \begin{cases} 0, & \text{if } k < k_f \\ X_{21}^2(k)X_{22}(k) + X_{22}(k), & \text{if } k \geq k_f \end{cases}. \quad (56)$$

In this simulation, the initial values, ANNs, CNNs, and designed parameters are chosen as the same in Case C of Example 1. Because the disturbance always exists in real-time systems, the disturbance is also taken into account here. Therefore, the disturbance $d(k) = [d_1(k), d_2(k)]$ is generated randomly in the interval $[0, 0.1]$.

The simulation results are depicted in Figs. 19–20. As observed from Fig. 19, the tracking error converges to a small neighborhood of the origin. Fig. 20 shows the tracking trajectories of y_{d2}, X_{12} and tracking errors, respectively. It can be seen that good tracking performances are obtained. The wave phenomenon between 480th step and about 650th step is normal,

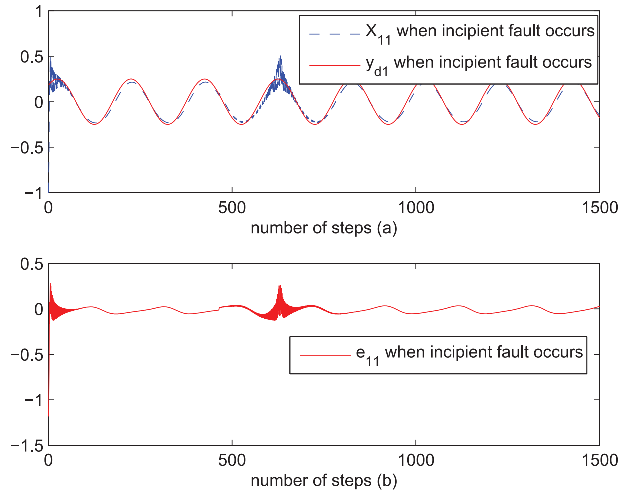

Fig. 19. The curves of $y_{d1}(k)$, $X_{11}(k)$ and $e_{11}(k)$ in Example 3 when incipient fault occurs.

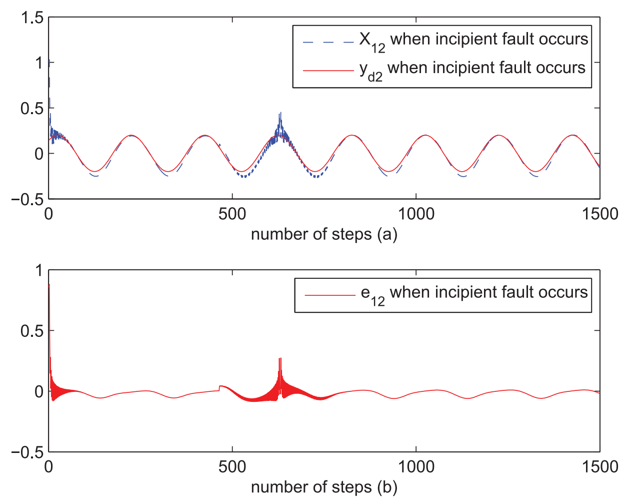

Fig. 20. The curves of $y_{d2}(k)$, $X_{12}(k)$ and $e_{12}(k)$ in Example 3 when incipient fault occurs.

because the toleration procedure of incipient fault requires a certain time.

VI. Conclusion

In this paper, an RL-based adaptive tracking FTC problem is investigated for a class of MIMO nonlinear discrete-time systems with less learning parameters. Both incipient faults and abrupt faults are considered. Based on the approximation capability of NNs, ANNs, and CNNs have been designed to estimate the near optimal control signals and the long-term cost function, respectively. Notice that the number of the learning parameters has been reduced by updating the estimation values of the unknown weights' Euclidean norm, which may further reduce the online computation time. Finally, three simulations have been used to verify the effectiveness of the proposed FTC policy. However, the proposed RL-based FTC method is designed for a class of MIMO nonlinear discrete-time systems (1) with $G(X(k))$ being a positive definite matrix. This constraint condition may limit the application range of the proposed method. In fact, in order to quickly respond to the fault and maintain the suboptimal cost function, how to reduce the learning time of NNs and optimize the cost function simultaneously is a difficult problem. To the best of the authors' knowledge, there are few results to be published. In this paper, we have done some preliminary researches. It is a further topic to design RL-based FTC for MIMO nonlinear discrete-time systems with $G(X(k))$ being indefinite or unknown. In addition, for the case that the states are not measurable, it is a way to utilize the observer technique to estimate the states as a future research work.

References

[1] G. H. Yang and H. Wang, "Fault detection for a class of uncertain state feedback control systems," *IEEE Trans. Control Syst. Technol.*, vol. 18, no. 1, pp. 201–212, Jan. 2010.
[2] D. Ye and G. H. Yang, "Adaptive fault-tolerant tracking control against actuator faults with application to flight control," *IEEE Trans. Control Syst. Technol.*, vol. 14, no. 6, pp. 1088–1096, Nov. 2006.
[3] X. Zhang, T. Parisini, and M. M. Polycarpou, "Adaptive fault-tolerant control of nonlinear uncertain systems: An information-based diagnostic approach," *IEEE Trans. Autom. Control*, vol. 49, no. 8, pp. 1259–1274, Aug. 2004.
[4] H. Y. Li, H. J. Gao, P. Shi, and X. D. Zhao, "Fault-tolerant control of Markovian jump stochastic systems via the augmented sliding mode observer approach," *Automatica*, vol. 50, no. 7, pp. 1825–1834, 2014.
[5] M. Liu, P. Shi, L. X. Zhang, and X. D. Zhao, "Fault-tolerant control for nonlinear Markovian jump systems via proportional and derivative sliding mode observer technique," *IEEE Trans. Circuits Syst. I: Regular Papers*, vol. 58, no. 11, pp. 2755–2764, Nov. 2011.
[6] Z. S. Wang, T. S. Li, and H. G. Zhang, "Fault tolerant synchronization for a class of complex interconnected neural networks with delay," *Int. J. Adaptive Control Signal Process.*, vol. 28, no. 10, pp. 859–881, 2014.
[7] V. Reppa, M. M. Polycarpou, and C. G. Panayiotou, "Decentralized isolation of multiple sensor faults in large-scale interconnected nonlinear systems," *IEEE Trans. Autom. Control*, vol. 60, no. 6, pp. 1582–1596, Jun. 2015, DOI: 10.1109/TAC.2014.2384371.
[8] Z. Zuo, J. Zhang, and Y. Wang, "Adaptive fault tolerant tracking control for linear and Lipschitz nonlinear multi-agent systems," *IEEE Trans. Ind. Electron.*, vol. 62, no. 6, pp. 3923–3931, Jun. 2015, DOI: 10.1109/TIE.2014.2367034.
[9] B. Jiang and F. N. Chowdhury, "Fault estimation and accommodation for linear MIMO discrete-time systems," *IEEE Trans. Control Syst. Technol.*, vol. 13, no. 3, pp. 493–499, May 2005.
[10] C. Tan, G. Tao, and R. Qi, "A discrete-time parameter estimation based adaptive actuator failure compensation control scheme," *Int. J. Control*, vol. 86, no. 2, pp. 276–289, 2013.
[11] R. M. G. Ferrari, T. Parisini, and M. M. Polycarpou, "Distributed fault detection and isolation of large-scale discrete-time nonlinear systems: An adaptive approximation approach," *IEEE Trans. Autom. Control*, vol. 57, no. 2, pp. 275–290, Feb. 2012.
[12] S. L. Shu and F. Lin, "Fault-tolerant control for safety of discrete-event systems," *IEEE Trans. Autom. Sci. Eng.*, vol. 11, no. 1, pp. 78–89, Jan. 2014.
[13] A. Ramirez-Trevino, E. Ruiz-Beltran, I. Rivera-Rangel, and E. Lopez-Mellado, "Online fault diagnosis of discrete event systems. A Petri net-based approach," *IEEE Trans. Autom. Sci. Eng.*, vol. 4, no. 1, pp. 31–39, Jan. 2007.
[14] B. Jiang, K. Zhang, and P. Shi, "Integrated fault estimation and accommodation design for discrete-time Takagi-Sugeno fuzzy systems with actuator faults," *IEEE Trans. Fuzzy Syst.*, vol. 19, no. 2, pp. 291–304, Apr. 2011.
[15] M. Y. Zhong, S. X. Ding, and E. L. Ding, "Optimal fault detection for linear discrete time-varying systems," *Automatica*, vol. 46, no. 8, pp. 1395–1400, 2010.
[16] X. Li and H. Liu, "Characterization of H_∞ index for linear time-varying systems," *Automatica*, vol. 49, no. 5, pp. 1449–1457, 2013.
[17] D. Efimov and A. Zolghadri, "Optimization of fault detection performance for a class of nonlinear systems," *Int. J. Robust Nonlinear Control*, vol. 22, no. 17, pp. 1969–1982, 2012.
[18] X. Yang, D. R. Liu, D. Wang, and Q. L. Wei, "Discrete-time online learning control for a class of unknown nonaffine nonlinear systems using reinforcement learning," *Neural Netw.*, vol. 55, pp. 30–41, 2014.

[19] Q. Yang, J. Vance, and S. Jagannathan, "Control of nonaffine nonlinear discrete-time systems using reinforcement-learning-based linearly parameterized neural networks," *IEEE Trans. Syst., Man, Cybern., Part B: Cybern.*, vol. 38, no. 4, pp. 994–1001, Aug. 2008.

[20] A. Al-Tamimi, M. Abu-Khalaf, and F. L. Lewis, "Adaptive critic designs for discrete-time zero-sum games with application to H_∞ control," *IEEE Trans. Syst., Man, Cybern., Part B: Cybern.*, vol. 37, no. 1, pp. 240–247, 2007.

[21] S. Ray, G. K. Venayagamoorthy, B. Chaudhuri, and R. Majumder, "Comparison of adaptive critic-based and classical wide-area controllers for power systems," *IEEE Trans. Syst., Man, Cybern., Part B: Cybern.*, vol. 38, no. 4, pp. 1002–1007, Aug. 2008.

[22] D. R. Liu, H. Javaherian, O. Kovalenko, and T. Huang, "Adaptive critic learning techniques for engine torque and air-fuel ratio control," *IEEE Trans. Syst., Man, Cybern., Part B: Cybern.*, vol. 38, no. 4, pp. 988–993, Aug. 2008.

[23] Q. Wei and D. Liu, "Adaptive dynamic programming for optimal tracking control of unknown nonlinear systems with application to coal gasification," *IEEE Trans. Autom. Sci. Eng.*, vol. 11, no. 4, pp. 1020–1036, 2014.

[24] J. J. Murray, C. J. Cox, G. G. Lendaris, and R. Saeks, "Adaptive dynamic programming," *IEEE Trans. Syst., Man, Cybern., Part C: Appl.*, vol. 32, no. 2, pp. 140–153, 2002.

[25] H. G. Zhang, C. B. Qin, and Y. H. Luo, "Neural-network-based constrained optimal control scheme for discrete-time switched nonlinear system using dual heuristic programming," *IEEE Trans. Autom. Sci. Eng.*, vol. 11, no. 3, pp. 839–849, Jul. 2014.

[26] D. P. Bertsekas and J. N. Tsitsiklis, "Neuro-dynamic programming: An overview," in *Proc. 34th IEEE Conf. Decision Control*, 1995, pp. 560–564.

[27] H. Xu and S. Jagannathan, "Stochastic optimal controller design for uncertain nonlinear networked control system via neuro dynamic programming," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 24, no. 3, pp. 471–484, Mar. 2013.

[28] F. L. Lewis and K. G. Vamvoudakis, "Reinforcement learning for partially observable dynamic processes: Adaptive dynamic programming using measured output data," *IEEE Trans. Syst., Man, Cybern., Part B: Cybern.*, vol. 41, no. 1, pp. 14–25, Feb. 2011.

[29] B. Xu, C. Yang, and Z. Shi, "Reinforcement learning output feedback NN control using deterministic learning technique," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 3, pp. 635–641, Feb. 2014.

[30] P. He and S. Jagannathan, "Reinforcement learning-based output feedback control of nonlinear systems with input constraints," *IEEE Trans. Syst., Man, Cybern., Part B: Cybern.*, vol. 35, no. 1, pp. 150–154, Feb. 2005.

[31] F. L. Lewis and K. G. Vamvoudakis, "Optimal adaptive control for unknown systems using output feedback by reinforcement learning methods," in *Proc. 8th IEEE Int. Conf. Control Autom. (ICCA)*, 2010, pp. 2138–2145.

[32] Q. L. Wei and D. R. Liu, "A novel iterative θ-adaptive dynamic programming for discrete-time nonlinear systems," *IEEE Trans. Autom. Sci. Eng.*, vol. 11, no. 4, pp. 1176–1190, Oct. 2014.

[33] D. P. Bertsekas, *Dynamic Programming and Optimal Control*. Belmont, MA, USA: Athena Scientific, 1995.

[34] Q. Zhou, P. Shi, J. J. Lu, and S. Y. Xu, "Adaptive output-feedback fuzzy tracking control for a class of nonlinear systems," *IEEE Trans. Fuzzy Syst.*, vol. 19, no. 5, pp. 972–982, Oct. 2011.

[35] Y. Yang and J. Ren, "Adaptive fuzzy robust tracking controller design via small gain approach and its application," *IEEE Trans. Fuzzy Syst.*, vol. 11, no. 6, pp. 783–795, Dec. 2003.

[36] S. S. Ge, J. Zhang, and T. H. Lee, "Adaptive neural network control for a class of MIMO nonlinear systems with disturbances in discrete-time," *IEEE Trans. Syst., Man, Cybern., Part B: Cybern.*, vol. 34, no. 4, pp. 1630–1645, Aug. 2004.

[37] Y. J. Liu, L. Tang, S. C. Tong, C. L. Philip Chen, and D. J. Li, "Reinforcement learning design-based adaptive tracking control with less learning parameters for nonlinear discrete-time MIMO systems," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 1, pp. 165–176, Jan. 2015.

[38] Q. Zhou, P. Shi, H. H. Liu, and S. Y. Xu, "Neural-network-based decentralized adaptive output-feedback control for large-scale stochastic nonlinear systems," *IEEE Trans. Syst., Man, Cybern., Part B: Cybern.*, vol. 42, no. 6, pp. 1608–1619, Dec. 2012.

[39] H. Deng and M. Krstic, "Stochastic nonlinear stabilization-I: A backstepping design," *Syst. Control Lett.*, vol. 32, no. 3, pp. 143–150, 1997.

[40] H. Q. Wang, B. Chen, K. F. Liu, X. P. Liu, and C. Lin, "Adaptive neural tracking control for a class of nonstrict-feedback stochastic nonlinear systems with unknown backlash-like hysteresis," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 5, pp. 947–958, May 2014.

[41] T. S. Li, S. C. Tong, and G. G. Feng, "A novel robust adaptive-fuzzy-tracking control for a class of nonlinearmulti-input/multi-output systems," *IEEE Trans. Fuzzy Syst.*, vol. 18, no. 1, pp. 150–160, Feb. 2010.

[42] G. G. Yen and P. G. DeLima, "Improving the performance of globalized dual heuristic programming for fault tolerant control through an online learning supervisor," *IEEE Trans. Autom. Sci. Eng.*, vol. 2, no. 2, pp. 121–131, Apr. 2005.

[43] Q. M. Yang and S. Jagannathan, "Reinforcement learning controller design for affine nonlinear discrete-time systems using online approximators," *IEEE Trans. Syst., Man, Cybern., Part B: Cybern.*, vol. 42, no. 2, pp. 377–390, Apr. 2012.

[44] X. D. Zhang, M. M. Polycarpou, and T. Parisini, "Adaptive fault diagnosis and fault-tolerant control of MIMO nonlinear uncertain systems," *Int. J. Control*, vol. 83, no. 5, pp. 1054–1080, 2010.

[45] P. Panagi and M. M. Polycarpou, "Decentralized fault tolerant control of a class of interconnected nonlinear systems," *IEEE Trans. Autom. Control*, vol. 56, no. 1, pp. 178–184, Jan. 2011.

[46] P. Panagi and M. M. Polycarpou, "A coordinated communication scheme for distributed fault tolerant control," *IEEE Trans. Ind. Informat.*, vol. 9, no. 1, pp. 386–393, Feb. 2013.

[47] Q. Shen, B. Jiang, P. Shi, and C. C. Lim, "Novel neural networks-based fault tolerant control scheme with fault alarm," *IEEE Trans. Cybern.*, vol. 44, no. 11, pp. 2190–2201, Nov. 2014.

[48] H. Y. Li, Y. Gao, L. Wu, and H. K. Lam, "Fault detection for T-S fuzzy time-delay systems: Delta operator and input-output methods," *IEEE Trans. Cybern.*, vol. 45, no. 2, pp. 229–241, Feb. 2015.

[49] Q. Liu, S. J. Qin, and T. Y. Chai, "Decentralized fault diagnosis of continuous annealing processes based on multilevel PCA," *IEEE Trans. Autom. Sci. Eng.*, vol. 10, no. 3, pp. 687–698, Jul. 2013.

[50] Q. Shen, B. Jiang, and V. Cocquempot, "Adaptive fuzzy observer-based active fault-tolerant dynamic surface control for a class of nonlinear systems with actuator faults," *IEEE Trans. Fuzzy Syst.*, vol. 22, no. 2, pp. 338–349, Apr. 2014.

[51] M. M. Polycarpou and A. T. Vemuri, "Learning methodology for failure detection and accommodation," *IEEE Control Syst.*, vol. 15, no. 3, pp. 16–24, Jun. 1995.

[52] S. J. Yoo, "Fault detection and accommodation of a class of nonlinear systems with unknown multiple time-delayed faults," *Automatica*, vol. 50, pp. 255–261, 2014.

[53] Y. Y. Xu, S. C. Tong, and Y. M. Li, "Prescribed performance fuzzy adaptive fault-tolerant control of non-linear systems with actuator faults," *IET Control Theory Appl.*, vol. 8, no. 6, pp. 420–431, 2014.

[54] T. Chen, C. Wang, and D. J. Hill, "Small oscillation fault detection for a class of nonlinear systems with output measurements using deterministic learning," *Syst. Control Lett.*, vol. 79, pp. 39–46, 2015.

[55] H. F. Min and N. Duan, "Neural network-based adaptive state-feedback control for high-order stochastic nonlinear systems," *Acta Automatica Sinica*, vol. 40, no. 12, pp. 2968–2972, 2014.

[56] X. J. Xie and C. R. Zhao, "State feedback stabilization of stochastic feedforward nonlinear systems with input time-delay," *Acta Automatica Sinica*, vol. 40, no. 12, pp. 2972–2976, 2014.

[57] Z. H. Peng, D. Wang, H. Wang, and W. Wang, "Cooperative iterative learning control of linear multi-agent systems with a dynamic leader under directed topologies," *Acta Automatica Sinica*, vol. 40, no. 11, pp. 2595–2601, 2014.

[58] Z. H. Wang, M. Rodrigues, D. Theilliol, and Y. Shen, "Sensor fault estimation filter design for discrete-time linear time-varying systems," *Acta Automatica Sinica*, vol. 40, no. 10, pp. 2364–2369, 2014.

Lei Liu received the B.S. degree in information and computing science and the M.S. degree in applied mathematics from the Liaoning University of Technology, Jinzhou, China, in 2010 and 2013, respectively. He is currently pursuing the Ph.D. degree in Northeastern University, Shenyang, China.

His current research interests include fault-tolerant control, fault detection and diagnosis, optimal control for nonlinear systems, neural network control and their industrial applications.

Zhanshan Wang (M'13) was born in Liaoning, China, in 1971. He received the M.S. degree in control theory and control engineering from Fushun Petroleum Institute (now Liaoning Shihua University), Fushun, China, in 2001, and the Ph.D. degree in control theory and control engineering from Northeastern University, Shenyang, China, in 2006.

Since 2010, he has been a Professor with Northeastern University. He has authored and coauthored over 110 journal and conference papers and five monographs. He is the holder of ten patents. His research interests include the stability analysis of recurrent neural networks, fault diagnosis, fault tolerant control, nonlinear control theory and their applications in smart grid.

Dr. Wang was an Associate Editor of the IEEE TRANSACTION ON NEURAL NETWORKS AND LEARNING SYSTEMS. Now, he is a member of the Editorial Board of *Acta Automatica Sinica*.

Huaguang Zhang (SM'4–F'15) received the B.S. and M.S. degrees in control engineering from Northeastern Dianli University, Jilin, China, in 1982 and 1985, respectively, and the Ph.D. degree in thermal power engineering and automation from Southeast University, Nanjing, China, in 1991.

He joined the Department of Automatic Control, Northeastern University, Shenyang, China, in 1992, as a Post-Doctoral Fellow for two years. Since 1994, he has been a Professor and the Head of the Institute of Electric Automation, School of Information Science and Engineering, Northeastern University. He has authored and coauthored over 300 journal and conference papers, six monographs, and co-invented 90 patents. His current research interests include fuzzy control, stochastic control, neural networks-based control, nonlinear control, and their industrial applications.

Dr. Zhang was a recipient of the Outstanding Youth Science Foundation Award from the National Natural Science Foundation of China in 2003 and the IEEE Transactions on Neural Networks Outstanding Paper Award in 2012. He was named the Cheung Kong Scholar by the Education Ministry of China in 2005. He was the Chair of the Adaptive Dynamic Programming and Reinforcement Learning Technical Committee of the IEEE Computational Intelligence Society. He is an Associate Editor of *Automatica*, the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, the IEEE TRANSACTIONS ON CYBERNETICS, and *Neurocomputing*. He was an Associate Editor of the IEEE TRANSACTIONS ON FUZZY SYSTEMS from 2008 to 2013.