Data-based Fault Tolerant Control for Affine Nonlinear Systems Through Particle Swarm Optimized Neural Networks

Haowei Lin, Bo Zhao, *Member, IEEE*, Derong Liu, *Fellow, IEEE*, and Cesare Alippi, *Fellow, IEEE*

Abstract—In this paper, a data-based fault tolerant control (FTC) scheme is investigated for unknown continuous-time (CT) affine nonlinear systems with actuator faults. First, a neural network (NN) identifier based on particle swarm optimization (PSO) is constructed to model the unknown system dynamics. By utilizing the estimated system states, the particle swarm optimized critic neural network (PSOCNN) is employed to solve the Hamilton-Jacobi-Bellman equation (HJBE) more efficiently. Then, a data-based FTC scheme, which consists of the NN identifier and the fault compensator, is proposed to achieve actuator fault tolerance. The stability of the closed-loop system under actuator faults is guaranteed by the Lyapunov stability theorem. Finally, simulations are provided to demonstrate the effectiveness of the developed method.

Index Terms—Adaptive dynamic programming (ADP), critic neural network, data-based, fault tolerant control (FTC), particle swarm optimization (PSO).

NOMENCLATURE

$\lambda_{\max}(\cdot)$	Maximum eigenvalue of a matrix
$\lambda_{\min}(\cdot)$	Minimum eigenvalue of a matrix
$\mathbb{R}$	Set of all real numbers
$\mathbb{R}^m$	Space of all real m-vectors
$\mathbb{R}^{n \times m}$	Space of all $n \times m$ real matrices
$\nabla J(x)$	Partial derivative of $J(x)$ w.r.t. x
$\triangleq$	Equal by definition
$\|s\|$	2-norm of the vector $s \in \mathbb{R}^n$
$\|w\|$	Frobenius-norm of the matrix $w \in \mathbb{R}^{n \times m}$
I_n	Identity matrix of dimension $n \times n$

I. INTRODUCTION

MODERN complex control systems always require optimal control but ensuring closed-loop stability is a difficult task, as accurate dynamics are hardly modelled. Traditional optimal control theory built upon dynamic programming and Pontryagin's maximum principle finds the optimal objective by optimizing self-defined cost functions. However, those methodologies operate off-line and require availability of equations describing the system in advance [1]. Adaptive dynamic programming (ADP) [2]–[5] is an approximate optimal control approach emerging in the field of intelligent control. Similar to reinforcement learning (RL) [6]–[9], ADP uses two main algorithms named policy iteration (PI) [10], [11] and value iteration (VI) [12] to achieve policy evaluation and policy improvement iteratively. ADP aims at adaptively learning the optimal control strategy by constructing a critic neural network (NN) approximating the solution of the Hamilton-Jacobi-Bellman equation (HJBE) [13]. Based on the NN approximator, the optimal control is obtained forward-in-time and the `curse of dimensionality' is conquered [14]. ADP is a well-known advanced and effective method for optimal control in both the theoretical research and real-world applications [15]–[18]. Extensive efforts have been dedicated to developing ADP approaches for nonlinear systems.

In real systems, it is unavoidable to experience faults in actuators, sensors, or other system parts [19], [20]. In particular, actuator faults would cause severe damages as faults cannot be accommodated by a pre-designed controller. In order to solve this problem and integrate fault tolerance ability at actuator level, robust control strategies should be considered at the control design phase [21].

There are some ADP-based control algorithms that consider both optimization and fault tolerant abilities. In [22], a fault tolerant control (FTC) algorithm based on PI was developed for nonlinear systems. The solution of the HJBE was achieved by using the NN approximation. In order to solve the actuator fault problem, a fault compensator, which did not require fault detection and isolation abilities, was designed, and the closed-loop system with actuator faults was guaranteed to be stable. Zhao *et al.* [23] developed an ADP-based actuator FTC scheme by designing a fault observer for nonlinear systems.

The key idea is that the FTC problem was regarded as an optimization problem by considering the fault estimate in the design of the loss function. Wu *et al.* [24] considered the actuator failure in the tracking control task and developed an optimal adaptive compensation control based on the estimation of actuator failure coefficients. These studies require the availability of system equations. The topic on FTC for discrete-time systems with unknown dynamics has attracted considerable attention [25]–[27]. The RL-based adaptive tracking FTC was studied for MIMO (multi-input and multi-output) discrete-time systems in [25] and [26]. Based on the actor-critic NN structure, systems affected by abrupt faults at actuator level could be maintained stable. The proposed strategy [25] required a lower computational load and fewer learning parameters as it estimated the Euclidean norm of unknown weights of NNs instead of updating the NN weights directly. A model-free FTC strategy was proposed in [27] for single-input single-output systems. The original system is transformed into a model-free data form. By designing an NN approximator to learn the sensor fault, the FTC strategy is reconstructed based on the optimality criterion. For ADP-based FTC of unknown continuous-time (CT) systems, Zhang *et al.* [28] proposed a fuzzy FTC strategy based on RL for systems whose dynamics was partially unknown. They designed a new performance index function which reflects four types of actuator failures. Then, based on the constructed fuzzy-augmented dynamics, the control policy which achieved the tracking goal and stabilized the closed-loop system under actuator failures was obtained. However, this methodology is applicable only to partially unknown fuzzy systems. We finally comment that there are few ADP-based FTC schemes for completely unknown CT nonlinear systems.

As we know, the gradient-based critic NN (GDCNN) methods are widely used to solve HJBEs in order to achieve approximate optima. To train the critic NN with the gradient-based (GD) learning algorithm, one starts with random initial weights and updates them by moving along the direction of gradient descent. It means that the GD algorithm provides a tractable way for local hill climbing on the landscape of the critic NN weight parameter space. However, when initialized at a low hill in the parameter space, the GD algorithm may be trapped by unsatisfactory local optimization, resulting in inefficient HJBE solutions. One may avoid this problem by training the critic NN more than one session or applying specific prior knowledge to choose a good initial parameter. In this paper, we propose a particle swarm optimization (PSO) method to solve this problem.

PSO is a stochastic optimization algorithm where each particle has a virtual position that represents a possible solution to the optimization problem [29]–[31]. In the training phase, a set of particles are initialized and evolve to search the optimal solution associated with the particle characterized by the best fitness value. PSO has multiple initial positions and relies on the global heuristic search principle, which increases the probability to avoid and even jump out of local optimums. Recently, a better performance for NN based methods has been achieved by integrating PSO into NNs. Martin *et al.* [32]

developed the PSO-trained NN to solve the electrical impedance tomography problem. It was shown that the PSO-trained NN converged faster compared to the GD algorithm. Das *et al.* [33] considered PSO-trained NNs in channel equalization problems. The proposed equalizer performs better than other NN-based equalizers in noisy conditions. Chan *et al.* [34] presented a short-term traffic flow forecast algorithm based on PSO and artificial NNs, which required simple NNs and contained memory.

Motivated by the above analysis, this paper develops an ADP approach based on PSO and NNs to achieve actuator fault tolerance of unknown CT affine nonlinear systems. The main contributions are:

1) The proposed data-based FTC algorithm deals with completely unknown CT nonlinear systems, rather than known or partially unknown systems (as in [23] and [28]). Moreover, the dynamics of the unknown systems are approximated by the PSO-trained nonlinear NN identifier based on available measurements, and hence making the method effective in real applications.

2) The HJBE is solved through the particle swarm optimized critic NN (PSOCNN) instead of the general GDCNN; in this way the HJBE is solved with a high successful rate.

3) The presented data-based FTC strategy provides an online fault tolerant control which is shown to be optimal.

The rest of this paper is organized as follows. In Section II, the problem statement for faulty nonlinear CT systems is presented. In Section III, an NN identifier is constructed to estimate the system dynamics. Then, the data-based FTC through the PSONNs is developed based on the adaptive fault estimation. In Section IV, two simulation examples are provided to demonstrate the effectiveness of the proposed method. Finally, Section V concludes the present paper.

II. PROBLEM STATEMENT

Consider the unknown CT affine nonlinear system with actuator faults described by

$$\dot{x}(t) = f(x(t)) + g(x(t))(u(x(t)) + u_f(t)) \qquad (1)$$

where $x(t) \in \mathbb{R}^n$ and $u(x(t)) \in \mathbb{R}^m$ represent the state vector and the control vector, respectively. The unknown terms $f(\cdot)$ and $g(\cdot)$ are Lipschitz and differentiable in their arguments with $f(0) = 0$, and $u_f(t)$ represents the unknown additive actuator fault vector. Let the initial state be $x(0) = x_0$.

Assumption 1: The actuator fault vector and its time-derivative are unknown and norm-bounded as $\|u_f(t)\| \leq \varrho_1 \leq \infty$, $\|\dot{u}_f(t)\| \leq \varrho_2 \leq \infty$, where ϱ_1 and ϱ_2 are two positive constants.

Remark 1: The possible actuator faults are stochastic and may occur simultaneously. Assume that actuator faults do not deteriorate rapidly over a period of time as they happen with exponential behaviours, which destroy the system and prevent any system recovery. In practice, it is reasonable to assume the actuator fault vector u_f and its time-derivative $\dot{u}_f$ to be norm-bounded by constants [23], [35], [36]. For instance, the fault is associated with hardware aging. Therefore, it is possible to change the control law in (1) to achieve the desired system performance under the situation of actuator faults.

The system (1) in fault free case can be described by the corresponding nominal system

$$\dot{x}(t) = f(x(t)) + g(x(t))u(x(t)). \tag{2}$$

Since f and g are Lipschitz and differentiable, $f + gu$ is Lipschitz continuous on a set of $\Omega \subset \mathbb{R}^n$, and the nominal system (2) is controllable.

Define the cost function for system (2) as

$$J(x_0) = \int_0^\infty U(x(\tau), u(\tau))d\tau \tag{3}$$

where $U(x,u) = x^T Q x + u^T R u$ represents the utility function, $U(x,u) \geq 0$ for all x and u with $U(0,0) = 0$, $Q \in \mathbb{R}^{n \times n}$ and $R \in \mathbb{R}^{m \times m}$.

If $u \in \Psi(\Omega)$ is an admissible control [37], [38], where $\Psi(\Omega)$ denotes the set of admissible control actions, and (3) is differentiable, then the infinitesimal version of (3) is the so-called nonlinear Lyapunov equation

$$0 = U(x,u) + (\nabla J(x))^T (f(x) + g(x)u) \tag{4}$$

where $\nabla J(x)$ denotes the partial derivative of $J(x)$ w.r.t. x, i.e., $\nabla J(x) = \dfrac{\partial J(x)}{\partial x}$.

Define the Hamiltonian and the optimal cost function as

$$H(x,u,\nabla J(x)) = U(x,u) + (\nabla J(x))^T (f(x) + g(x)u) \tag{5}$$

and

$$J^*(x_0) = \min_{u \in \Psi(\Omega)} \int_0^\infty U(x(\tau), u(\tau))d\tau \tag{6}$$

respectively. Then

$$0 = \min_{u \in \Psi(\Omega)} H(x,u,\nabla J^*(x)) \tag{7}$$

where $\nabla J^*(x) = \dfrac{\partial J^*(x)}{\partial x}$. If $J^*(x)$ exists and is differentiable, the optimal control law can be expressed as

$$u^*(x) = -\frac{1}{2} R^{-1} g^T(x) \nabla J^*(x). \tag{8}$$

Generally, for the known fault-free system (2), the HJBE (7) can be solved by using the ADP framework. However, the main purpose of this paper is to obtain the ADP-based FTC policy which guarantees the stability of the unknown system under actuator faults, and improves the success rate of solving the HJBEs by employing the PSOCNN method.

III. DATA-BASED FAULT TOLERANT CONTROL METHODOLOGY BASED ON PSONNS

A. Particle Swarm Optimization

In PSO, the jth individual solution is a particle represented as vector $X_j = (X_{j1}, X_{j2}, \ldots, X_{jd})$, where $j = 1, 2, \ldots, k$. Particles have fitness values; velocity $V_j = (V_{j1}, V_{j1}, \ldots, V_{jd})$ directs particles over iteration time. Each particle moves through the problem space with two optimal positions determined by the fitness values. One position is the previous best position $P_j = (P_{j1}, P_{j1}, \ldots, P_{jd})$, and the other is the best of all the best positions among k particles P_g. Based on the former velocity and position P_j and P_g, each particle updates iteratively as

$$V_j = \omega V_j + c_1 r_1 (P_j - X_j) + c_2 r_2 (P_g - X_j) \tag{9}$$

$$X_j = X_j + V_j \tag{10}$$

where r_1, r_2, c_1, c_2 and ω are candidate parameters [29], [30]. Generally, r_1 and r_2 are chosen randomly in the interval $(0,1)$, $c_1, c_2 > 1$, and ω updates as

$$\omega = \omega_{\min} + \frac{\gamma}{\gamma_{\max}} (\omega_{\max} - \omega_{\min}) \tag{11}$$

where $\gamma \leq \gamma_{\max}$ denotes the step of the iteration process and $\gamma_{\max}$ is the maximum step. $\omega_{\min}$ and $\omega_{\max}$ are the minimum and maximum values of ω, respectively.

According to [39], if the parameters ω, c_1, and c_2 satisfy proper conditions, the particle will converge to the best position.

Lemma 1: Given $\omega, c_1, c_2 \geq 0$, if iterative process $\{DX_t\}$ is guaranteed to converge and $f(1) < c_2^2(1+\omega)/6$, then iterative process $\{P_i(t)\}$ will converge to P_g with probability 1 [39].

B. Optimal Control via PSONNs

In this subsection, an NN identifier is constructed by using the measured input-output data to learn the dynamics of the unknown nonlinear CT systems. Then, the PSOCNN is employed to solve the HJBE, and the optimal control strategy is presented.

1) Neural Network Identification: According to [40]–[42], system (2) can be represented accurately by the NN identifier as

$$\dot{x} = Ax + w_2^T \sigma_i(w_1^T z) + \xi_i \tag{12}$$

where $z = [x^T, u^T]^T \in \mathbb{R}^{n+m}$ denotes the input vector of a three-layer NN with l_h neurons in the hidden layer. A is a known Hurwitz matrix. $w_1 \in \mathbb{R}^{(n+m) \times l_h}$ and $w_2 \in \mathbb{R}^{l_h \times n}$ are ideal weight matrices for the input-to-hidden layer, and the hidden-to-output layer, respectively. $\sigma_i(\cdot) = \tanh(\cdot) \in \mathbb{R}^{l_h}$ is the activation function, $\xi_i \in \mathbb{R}^n$ is the functional approximation error.

For simplicity, w_1 is always set to be a constant matrix, and hence only w_2 is required to be learned. Therefore, the output of the NN identifier can be described as

$$\dot{\hat{x}} = A\hat{x} + \hat{w}_2^T \sigma_i(w_1^T \hat{z}) \tag{13}$$

where $\hat{x}$ is the current estimate for system states, $\hat{z} = [\hat{x}^T, u^T]^T$, and $\hat{w}_2$ is the estimate of w_2. Define $s = w_1^T z$ and $\hat{s} = w_1^T \hat{z}$; from (12) and (13), the differential equation of identification error becomes

$$\dot{\tilde{x}}(t) = A\tilde{x}(t) + \tilde{w}_2^T \sigma_i(\hat{s}) + w_2^T (\sigma_i(s) - \sigma_i(\hat{s})) + \xi_i. \tag{14}$$

Remark 2: Initialize k particles for the desired NN identifier weight matrix $\hat{w}_2$ which is updated iteratively by the PSO algorithm. The iterative process $\hat{w}_2$ will converge to the ideal weight matrix w_2 according to Lemma 1, i.e., $Prob(\lim_{t \to \infty} \hat{w}_2 = w_2) = 1$.

Before presenting the uniform ultimate boundedness (UUB) of state identification errors $\tilde{x}$, we give the following reasonable assumptions, which have been used in [43]–[46].

Assumption 2: The ideal weight matrices of the identifier NN are norm-bounded as $\|w_1\| \leq \delta_1$, and $\|w_2\| \leq \delta_2$, where δ_1 and δ_2 are positive constants.

Assumption 3: The functional approximation error ξ_i is

upper-bounded by $\xi^T_i \xi_i \leq \lambda_{\xi_i} \tilde{x}^T \tilde{x}$, and the NN activation function $\sigma_i(\hat{s})$ is upper-bounded by $\|\sigma_i(\hat{s})\| \leq \lambda_{\sigma_i}$, where λ_{ξ_i} and λ_{σ_i} are two positive constants.

Theorem 1: Consider the unknown CT affine nonlinear system in fault free case (2), Assumptions 2 and 3, and Lemma 1, the identification error $\tilde{x}$ is UUB if the NN weights of the identifier are updated by the developed PSO equations (9) and (10).

Proof: Choose the Lyapunov function candidate as

$$L_1(t) = \frac{1}{2} \tilde{x}^T \tilde{x}.$$

By taking the time derivative of $L_1(t)$, we obtain

$$\dot{L}_1(t) = \tilde{x}^T \dot{\tilde{x}}. \tag{15}$$

Based on (14) and (15), we have that

$$\dot{L}_1(t) = \tilde{x}^T A \tilde{x} + \tilde{x}^T w_2^T (\sigma_i(s) - \sigma_i(\hat{s})) + \tilde{x}^T \tilde{w}_2^T \sigma_i(\hat{s}) + \tilde{x}^T \xi_i.$$

Similar to [41], [47], for any $y_1 \geq y_2$ and $y_1, y_2 \in \mathbb{R}$, there exists a constant $\lambda_0 > 0$ such that

$$\sigma_i(y_1) - \sigma_i(y_2) \leq \lambda_0 (y_1 - y_2). \tag{16}$$

According to (16) and Assumption 2, we can obtain

$$\tilde{x}^T w_2^T (\sigma_i(s) - \sigma_i(\hat{s})) \leq \frac{1}{2} \tilde{x}^T w_2^T w_2 \tilde{x} + \frac{1}{2} (\sigma_i(s) - \sigma_i(\hat{s}))^T \\ \times (\sigma_i(s) - \sigma_i(\hat{s})) \\ \leq \frac{1}{2} \tilde{x}^T w_2^T w_2 \tilde{x} + \frac{1}{2} \lambda_0^2 \|s - \hat{s}\|^2 \\ \leq \frac{1}{2} \delta_2^2 \tilde{x}^T \tilde{x} + \frac{1}{2} \lambda_0^2 \delta_1^2 \tilde{x}^T \tilde{x}. \tag{17}$$

According to [48], let $\tilde{w}_2$ be norm-bounded by a positive constant δ_3, such that $\|\tilde{w}_2\| \leq \delta_3$. Based on Assumption 3, we have

$$\tilde{x}^T \tilde{w}_2^T \sigma_i(\hat{s}) \leq \frac{1}{2} \tilde{x}^T \tilde{x} + \frac{1}{2} \tilde{w}_2^T (\sigma_i(\hat{s}))^T \sigma_i(\hat{s}) \tilde{w}_2 \\ \leq \frac{1}{2} \tilde{x}^T \tilde{x} + \frac{1}{2} \lambda_{\sigma_i}^2 \delta_3^2. \tag{18}$$

Based on (17), (18) and Assumption 3, we have

$$\dot{L}_1(t) \leq \tilde{x}^T A \tilde{x} + \frac{1}{2} \delta_2^2 \tilde{x}^T \tilde{x} + \frac{1}{2} \lambda_0^2 \delta_1^2 \tilde{x}^T \tilde{x} + \frac{1}{2} \tilde{x}^T \tilde{x} \\ + \frac{1}{2} \lambda_{\sigma_i}^2 \delta_3^2 + \frac{1}{2} \lambda_{\xi_i} \tilde{x}^T \tilde{x} + \frac{1}{2} \tilde{x}^T \tilde{x} \\ = \tilde{x}^T (A + \frac{1}{2} (2 + \lambda_0^2 \delta_1^2 + \lambda_{\xi_i} + \delta_2^2) I_n) \tilde{x} + \frac{1}{2} \lambda_{\sigma_i}^2 \delta_3^2 \\ = -\tilde{x}^T \Theta \tilde{x} + \Xi$$

where I_n represents the n-dimensional identity matrix and

$$\Theta = -\left(A + \frac{1}{2}(2 + \lambda_0^2 \delta_1^2 + \lambda_{\xi_i} + \delta_2^2 + \lambda_{\sigma_i} \delta_2^2) I_n\right)$$

$$\Xi = \frac{1}{2} \lambda_{\sigma_i}^2 \delta_3^2.$$

Then, we have

$$\dot{L}_1(t) \leq -\lambda_{\min}(\Theta) \|\tilde{x}\|^2 + \Xi \tag{19}$$

where $\lambda_{\min}(\cdot)$ denotes the minimum eigenvalue of a matrix.

Thus, we can observe that $\dot{L}_2(t) \leq 0$ whenever $\tilde{x}(t)$ lies outside the compact set

$$\Omega_{\tilde{x}} = \left\{ \tilde{x} : \tilde{x} \leq \sqrt{\frac{\Xi}{\lambda_{\min}(\Theta)}} \right\}. \tag{20}$$

According to Lyapunov theory, we obtain that the system state estimation error is UUB. ∎

It follows that the NN identifier is described as

$$\dot{\hat{x}} = f(\hat{x}) + g(\hat{x})u = A\hat{x} + \hat{w}_2^T \sigma_i(\hat{s}). \tag{21}$$

By taking the partial derivative of (21) with respect to u, we can obtain the estimated input gain matrix as

$$\hat{g}(\hat{x}) = \frac{\partial (A\hat{x} + \hat{w}_2^T \sigma_i(\hat{s}))}{\partial u} \\ = \hat{w}_2^T \nabla \sigma_i(\hat{s}) w_1^T \begin{bmatrix} O_{n \times m} \\ I_m \end{bmatrix}. \tag{22}$$

Therefore, with a well trained NN identifier, the estimated system state equation $\dot{\hat{x}}$ in (21) and control matrix $\hat{g}(\hat{x})$ in (22) can be derived. It follows that it is feasible to develop a data-based optimal control strategy built upon the ADP method, that can be applied to the FTC of unknown nonlinear CT systems.

2) Implementation Process: As we know, the key point to obtain the optimal control strategy is to solve the HJBE (7). Under the ADP framework, a critic NN is constructed to approximate the solution of (7). In order to obtain the well-trained critic NN which can approximate the solution of the HJBE, in this subsection, we develop the PSOCNN to solve the HJBE instead of the GDCNN. Based on the NN identifier (21), $J^*(x)$ can be reconstructed by a critic NN with a single layer as

$$J^*(\hat{x}) = w_c^T \sigma_c(\hat{x}) + \xi_c(\hat{x}) \tag{23}$$

where $w_c \in \mathbb{R}^l$ is the ideal weight matrix, $\sigma_c(\hat{x}) \in \mathbb{R}^l$ is the activation function, l represents the number of neurons in the hidden layer, and $\xi_c(\hat{x})$ is the approximation error. Taking the partial gradient of both sides of (23), one has

$$\nabla J^*(\hat{x}) = (\nabla \sigma_c(\hat{x}))^T w_c + \nabla \xi_c(\hat{x}). \tag{24}$$

Based on (21) and (24), (4) becomes

$$0 = U(\hat{x}, u) + \left(w_c^T \nabla \sigma_c(\hat{x}) + (\nabla \xi_c(\hat{x}))^T \right) \\ \times (A\hat{x} + \hat{w}_2^T \sigma_i(w_1^T \hat{s})).$$

Similar to [49], w_c, $\sigma_c(\hat{x})$, $\xi_c(\hat{x})$ and its partial derivative $\nabla \xi_c(\hat{x})$ are assumed to be norm-bounded on a compact set Ω. By estimating the ideal NN weight matrix with w_c, the critic NN is approximated as

$$\hat{J}(\hat{x}) = \hat{w}_c^T \sigma_c(\hat{x}). \tag{25}$$

Then, its partial derivative is

$$\nabla \hat{J}(\hat{x}) = (\nabla \sigma_c(\hat{x}))^T \hat{w}_c. \tag{26}$$

Substituting (22) and (24) into (8), we have

$$u^*(\hat{x}) = -\frac{1}{2} R^{-1} \hat{g}^T(\hat{x}) \left((\nabla \sigma_c(\hat{x}))^T w_c + \nabla \xi_c(\hat{x}) \right). \tag{27}$$

Since $\nabla J^*(\hat{x})$ is approximated by (26), the optimal control

can be approximated as

$$\hat{u}(\hat{x}) = -\frac{1}{2}R^{-1}\hat{g}^T(\hat{x})((\nabla\sigma_c(\hat{x}))^T\hat{w}_c. \tag{28}$$

Consider (27) and the NN identifier (21) and (22), the Hamiltonian (5) becomes

$$
\begin{aligned}
H(\hat{x}, w_c) &= \hat{x}^T Q\hat{x} + \frac{1}{4}w_c^T\nabla\sigma_c(\hat{x})\hat{g}(\hat{x})(R^{-1})^T RR^{-1}\hat{g}^T(\hat{x}) \\
&\quad \times (\nabla\sigma_c(\hat{x}))^T w_c + w_c^T\nabla\sigma_c(\hat{x})(A\hat{x} + \hat{w}_2^T\sigma_i(\hat{s}^*)) \\
&\quad - (\nabla\xi_c(\hat{x}))^T(A\hat{x} + \hat{w}_2^T\sigma_i(\hat{s}^*)) \\
&\quad - \frac{1}{4}(\nabla\xi_c(\hat{x}))^T\hat{g}(\hat{x})R^{-1}\hat{g}^T(\hat{x})\nabla\xi_c(\hat{x}) \\
&= \hat{x}^T Q\hat{x} + w_c^T\nabla\sigma_c(\hat{x})(A\hat{x} + \hat{w}_2^T\sigma_i(\hat{s}^*)) \\
&\quad + \frac{1}{4}w_c^T\nabla\sigma_c(\hat{x})\hat{g}(\hat{x})R^{-1}\hat{g}^T(\hat{x})(\nabla\sigma_c(\hat{x}))^T w_c - e_{cH} \\
&= 0 \tag{29}
\end{aligned}
$$

where

$$
\begin{aligned}
e_{cH} &= -(\nabla\xi_c(\hat{x}))^T(A\hat{x} + \hat{w}_2^T\sigma_i(\hat{s}^*)) \\
&\quad - \frac{1}{4}(\nabla\xi_c(\hat{x}))^T\hat{g}(\hat{x})R^{-1}\hat{g}^T(\hat{x})\nabla\xi_c(\hat{x})
\end{aligned}
$$

denotes the residual error with $\hat{s}^* = w_1^T[\hat{x}^T, u^{*T}]$. Assume that there exists a positive constant $\lambda_{e_{cH}}$ such that $\|e_{cH}\| \leq \lambda_{e_{cH}}$. Combining (26) and (28) with (29), the approximate Hamiltonian is derived as

$$H(\hat{x}, \hat{w}_c) = \hat{x}^T Q\hat{x} + \frac{1}{4}\Upsilon^T\Phi\Upsilon + \Upsilon(A\hat{x} + \hat{w}_2^T\sigma_i(\overline{s}^*)) \triangleq e_c \tag{30}$$

where $\Phi = \hat{g}(\hat{x})R^{-1}\hat{g}^T(\hat{x})$, $\Upsilon = \hat{w}_c^T\nabla\sigma_c(\hat{x})$, and $\overline{s}^* = w_1^T[\hat{x}^T, \hat{u}^T]^T$.

From (29) and (30), we can derive

$$
\begin{aligned}
e_c &= e_{cH} - \tilde{w}_c^T\nabla\sigma_c(\hat{x})(A\hat{x} + \hat{w}_2^T\sigma_i(\overline{s}^*)) \\
&\quad - \frac{1}{4}\tilde{w}_c^T\nabla\sigma_c(\hat{x})\Phi(\nabla\sigma_c(\hat{x}))^T\tilde{w}_c.
\end{aligned}
$$

Then, the cost function $E_c = \frac{1}{2}e_c^T e_c$ should be minimized [50]. In order to improve the success rate of solving the HJBE, in this paper, we use the PSO algorithm to train the critic NN.

By using the PSO algorithm, the critic NN weights are regarded as a particle. The fitness function of the particle is defined as [51]

$$fitness_c = \exp\left(-\frac{1}{2}E_c\right). \tag{31}$$

In this sense, with a set of particles update according to (9) and (10) iteratively, the objective is to find the current fitness value which is close to 1. Moreover, according to Lemma 1, $\hat{w}_c$ will converge to w_c with probability 1 after sufficient iterations.

Remark 3: In the proposed PSOCNN, the critic NN is trained by the PSO algorithm. PSO has the probabilistic mechanism. It searches for the ideal weights of the critic NN towards global optimum by means of mutual guidance. PSO has shown to be effective in solving complex optimization tasks. Based on PSO, the critic NN has a high probability to converge to the approximate optimum which can accurately approximate the HJBE solution. It is worth pointing out that

since a large population of initial weight vector is chosen in PSO algorithm, this paper focuses on improving the success rate of solving HJBEs in contrast to using gradient-based learning algorithm. That is to say, the PSO increases the rate to universal optimization, but is still a local optimization. Furthermore, in the learning phase, GDCNN has to compute the partial gradient of the cost function when updating the NN weight vector in each iteration. By contrast, the weight vector of PSOCNN is updated by (9) and (10) which requires simple computation, thereby reducing the computation cost and time.

Remark 4: The particle velocity V and the inertia weight ω of PSOCNN have the main impact on the results as they determine the search area and speed of particles. If V is too large, i.e., particles fly too fast, they may pass good solution of Hamiltonian. In contrast, the search area of particles is limited and it is difficult for particles to jump out of low hills. The inertia weight ω provides a tradeoff between the global and local search. It achieves better performance when choosing a decreasing ω.

C. Data-based Fault Tolerant Control Based on PSONNs and Stability Analysis

Based on the analysis in Subsection III-B, the unknown system with actuator faults is further considered in this subsection. The data-based FTC is established to overcome the impact of faults on the system performance. The stability analysis of the faulty closed-loop system under the proposed strategy is provided as well.

1) Data-based FTC for Actuator Faults: According to [22], the time derivative of the cost function is biased due to unknown actuator faults. It implies that the obtained approximate optimal control $\hat{u}$ (28) derived under the fault-free system cannot guarantee the stability of the closed-loop system. Let $u_n = \hat{u}$, according to (1), (21) and (22), the unknown system with actuator faults should be rewritten as

$$\dot{\hat{x}} = f(\hat{x}) + \hat{g}(\hat{x})(u_n + u_f). \tag{32}$$

Theorem 2: Consider the unknown CT affine nonlinear system with actuator faults (1), the NN identifier (13) and the cost function (3), the closed-loop of the unknown system (1) can be guaranteed to be UUB with the data-based FTC law u as

$$u = u_n - \hat{u}_f \tag{33}$$

where

$$\dot{\hat{u}}_f = \Gamma(2u_n^T R - \hat{x}^T\hat{g}(\hat{x}))^T \tag{34}$$

represents the estimated actuator fault, and Γ is a positive learning rate.

Proof: Choose Lyapunov function candidate as

$$L_2(t) = \frac{1}{2}\hat{x}^T\hat{x} + J(\hat{x}) + \frac{1}{2\Gamma}\tilde{u}_f^T\tilde{u}_f \tag{35}$$

where $\tilde{u}_f = u_f - \hat{u}_f$ is the fault estimation error.

Combining (7), (8), (21) with (22), one can obtain

$$(\nabla J(\hat{x}))^T\hat{g}(\hat{x}) = -2(u_n(\hat{x}))^T R \tag{36}$$

and

$$(\nabla J(\hat{x}))^T(f(\hat{x}) + \hat{g}(\hat{x})u_n) = -\hat{x}^T Q\hat{x} - u_n^T R u_n. \quad (37)$$

Substituting (32), (36) and (37) into the time derivative of (35), we have

$$\begin{aligned}
\dot{L}_2 &= \frac{1}{\Gamma}(\dot{u}_f - \dot{\hat{u}}_f)^T \tilde{u}_f + \hat{x}^T \dot{\hat{x}} + (\nabla J(\hat{x}))^T \dot{\hat{x}} \\
&= \frac{1}{\Gamma}(\dot{u}_f - \dot{\hat{u}}_f)^T \tilde{u}_f + \hat{x}^T(f(\hat{x}) + \hat{g}(\hat{x})(u_n + u_f - \hat{u}_f)) \\
&\quad + (\nabla J(\hat{x}))^T(f(\hat{x}) + \hat{g}(\hat{x})(u_n + u_f - \hat{u}_f)) \\
&= \hat{x}^T f(\hat{x}) + \hat{x}^T \hat{g}(\hat{x})u_n + \left(\hat{x} + (\nabla J(\hat{x}))^T\right)\hat{g}(\hat{x})(\hat{u}_f - u_f) \\
&\quad - \hat{x}^T Q\hat{x} - u_n^T R u_n + \frac{1}{\Gamma}\dot{u}_f^T \tilde{u}_f - \frac{1}{\Gamma}\dot{\hat{u}}_f^T \tilde{u}_f. \quad (38)
\end{aligned}$$

As $f(\hat{x})$ is Lipchitz, there exists a positive constant ρ such that $\|f(\hat{x})\| \le \rho\|\hat{x}\|$. Suppose that the partial derivative of the active function in the identifier is upper-bounded by a constant ϕ, i.e., $\|\nabla\sigma_i(\hat{s})\| \le \phi$. Thus, based on the Assumption 2, we can derive that $\|\hat{g}(\hat{x})\| \le |\phi\delta_1\delta_2|$. Then, without loss of generality, the fault estimation error $\tilde{u}_f$ is supposed to be upper-bounded by a small positive constant ε, i.e., $\|\tilde{u}_f\| \le \varepsilon$ [35], [36].

Based on the above analysis and assumptions, we can derive

$$\begin{aligned}
\dot{L}_2 &\le \rho\|\hat{x}\|^2 + \frac{1}{2}\|\hat{x}\|^2 + \frac{1}{2}\phi^2\delta_1^2\delta_2^2\|u_n\|^2 \\
&\quad - \lambda_{\min}(Q)\|\hat{x}\|^2 - \lambda_{\min}(R)\|u_n\|^2 \\
&\quad + \frac{1}{2\Gamma}\varrho_2^2 + \frac{1}{2\Gamma}\varepsilon^2 - (\hat{x}^T\hat{g}(\hat{x}) - 2u_n^T R + \frac{1}{\Gamma}\dot{\hat{u}}_f^T)\tilde{u}_f. \quad (39)
\end{aligned}$$

Substituting (34) into (39) and by simple transformation, it becomes

$$\begin{aligned}
\dot{L}_2 &\le -(\lambda_{\min}(Q) - \rho - \frac{1}{2})\|\hat{x}\|^2 \\
&\quad - (\lambda_{\min}(R) - \frac{1}{2}\phi^2\delta_1^2\delta_2^2)\|u_n\|^2 + \frac{1}{2\Gamma}(\varrho_2^2 + \varepsilon^2). \quad (40)
\end{aligned}$$

Let $\Lambda_1 = \lambda_{\min}(Q) - \rho - \frac{1}{2}$, $\Lambda_2 = \lambda_{\min}(R) - \frac{1}{2}\phi^2\delta_1^2\delta_2^2$, and $\Lambda_3 = \frac{1}{2\Gamma}(\varrho_2^2 + \varepsilon^2)$ then (40) can be rewritten as

$$\dot{L}_2 \le -\Lambda_1\|\hat{x}\|^2 - \Lambda_2\|u_n\|^2 + \Lambda_3.$$

Thus, we can derive that $\dot{L}_2 < 0$ for any $x \ne 0$ if

$$\begin{cases}
\lambda_{\min}(Q) \ge \rho + \dfrac{1}{2} \\
\lambda_{\min}(R) \ge \dfrac{1}{2}\phi^2\delta_1^2\delta_2^2
\end{cases}$$

holds whenever $\hat{x}$ lies outside the compact set

$$\Omega_{\hat{x}} = \left\{\hat{x}: \|\hat{x}\| \le \sqrt{\frac{\Lambda_3}{\Lambda_1}}\right\}.$$

It implies the state trajectories of the closed-loop system with additive actuator faults can be guaranteed to be UUB through the developed data-based FTC law (33). ∎

2) Data-based FTC Algorithm Based on PSONNs: Based on the NN identifier, PSOCNN and the FTC law (33), the data-based FTC strategy for unknown CT affine nonlinear systems with actuator faults is summarized in Algorithm 1.

The structural diagram of the data-based FTC based on

PSONNs is depicted in Fig. 1.

Algorithm 1 Data-based FTC Algorithm Based on PSONNs

Step 1: Initialization

Initialize k particles and parameters of PSO.

Let the initial iteration index be $i = 0$.

Give a small positive real number ϵ.

Start with N initial iterative value functions $J_1^0, J_2^0, \ldots, J_N^0$, and control laws $u_1^0, u_2^0, \ldots, u_N^0$.

Step 2: Neural network identification

Compute the estimates for system states $\dot{\hat{x}}$ and control matrix $\hat{g}(\hat{x})$ according to (21) and (22). Keep the converged weight matrices of the identifier.

Step 3: Particle update

Update the particles following the standard PSO procedure, equations (9)–(11).

Step 4: Policy evaluation

By using the estimated system states, compute the iterative value functions $J^{i+1}(\hat{x})$, which satisfies the nonlinear Lyapunov equation

$$0 = U(\hat{x}, u_j) + (\nabla J_j^{i+1}(\hat{x}))^T \dot{\hat{x}}$$

with $\nabla J_j^{i+1}(\hat{x}) = 0$, $j = 1, 2, \ldots, N$.

Step 5: Policy implement

Based on the estimated $\hat{g}(\hat{x})$, update the control laws as

$$u_j^{i+1}(\hat{x}) = -\frac{1}{2}R^{-1}\hat{g}^T(\hat{x})\nabla J_j^{i+1}(\hat{x}).$$

Step 6: Fault estimation

Estimate the actuator fault vector by

$$\dot{\hat{u}}_f = \Gamma(2u_n^T R - \hat{x}^T\hat{g}(\hat{x}))^T.$$

Step 7: Fault compensation

Compensate the faulty actuator by the data-based FTC law as

$$u = u_n - \hat{u}_f.$$

Step 8: PSOCNN stopping criterion

If $\|J_j^{i+1}(\hat{x}) - J_j^i(\hat{x})\| \le \epsilon$, the HJBE is solved successfully and the optimal control law $u_j^{i+1}(\hat{x})$ is obtained; else, $i = i + 1$ and go to Step 3.

IV. SIMULATION STUDIES

In this section, simulations of two examples are provided to show the effectiveness of the proposed data-based FTC strategy based on PSONNs.

Example 1: Consider the following CT affine nonlinear system

$$\dot{x} = \begin{bmatrix} -0.5x_1 + x_2(1 + 0.5x_2^2) \\ -0.8(x_1 + x_2) + 0.5x_2(1 - 0.3x_2^2) \end{bmatrix} + \begin{bmatrix} 0 \\ -0.5 \end{bmatrix}(u + u_f) \quad (41)$$

where $x = [x_1, x_2]^T \in \mathbb{R}^2$ and $u \in \mathbb{R}$ are the state and control input variables, respectively. $u_f \in \mathbb{R}$ represents the unknown additive actuator fault. In the simulation, let u_f be

$$u_f = \begin{cases} 0, & 0 \le t \le 5 \\ 8\cos\left(\dfrac{t}{2\pi}\right), & 5 < t \le 20. \end{cases} \quad (42)$$

Since the nonlinear system is unknown, an NN identifier of 3–8–2 structure is built to estimate the unknown system dynamics. In order to simplify the training process of the NN
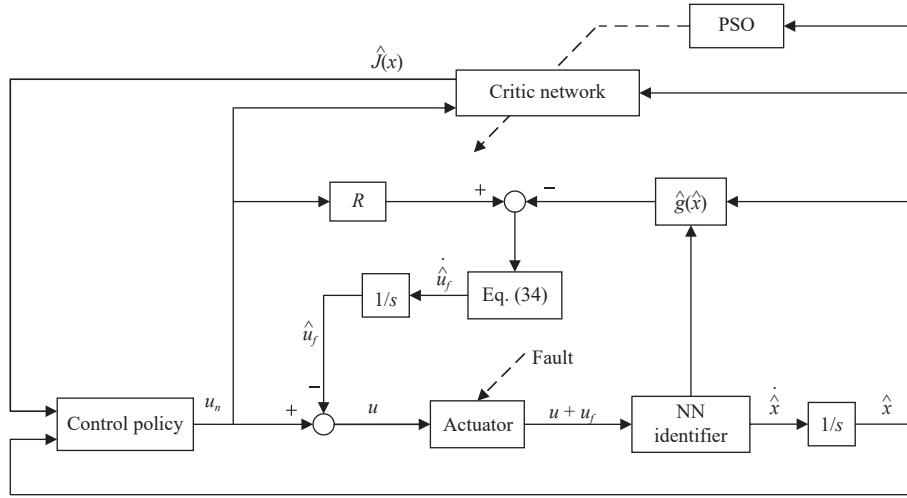
Fig. 1.   The structural diagram of data-based FTC based on PSONNs.

identifier, the NN weight matrix between the input layer and the hidden layer w_1 keeps constant randomly chosen within the interval $[-0.5, 0.5]$. The NN weight matrix w_2 is initialized as particles with population of 20, the fitness function of each particle is defined as $fitness_i = \exp(-\tilde{x}^T \tilde{x})$. Other parameters of the PSO algorithm are chosen the same as in [29]. The identification errors are shown in Fig. 2. It implies the NN identifier learns unknown nonlinear system dynamics successfully.
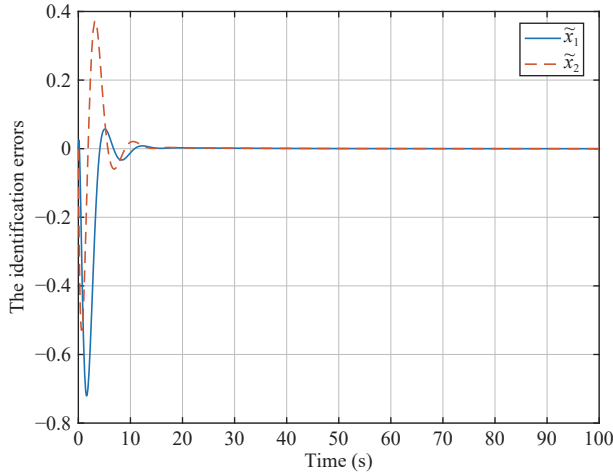

Fig. 2.   Identification errors of Example 1.

Furthermore, the critic NN is trained by the PSO algorithm. The critic NN weights are initialized as particles, whose fitness functions are defined as in (31). The PSO parameters and initial conditions are listed in Table I. The activation function of the critic NN is chosen as $\sigma_c(x) = [x_1^2, x_1 x_2, x_2^2, x_1^4, x_2^3 x_2, x_1^2 x_2^2, x_1 x_2^3, x_2^4]$. Let $Q = 2I_1$, $R = I_2$, $\Gamma = 40$ and the initial state of system (42) be $x_0 = [0.5, -0.5]^T$. Fig. 3 shows the system states under the nominal optimal control law (28) based on PSOCNN. It illustrates that the nominal optimal control law (28) can drive the system states converge to a small region of zero within the first 5 s. However, when the actuator fault occurs at time $t = 5$ s, the system states are away

from equilibriums, which implies that the actuator fault causes the systems unstable. Fig. 4 displays the trajectory of the actuator fault estimated by (34). It can be seen that the estimated fault trajectory can follow the actual one accurately. Fig. 5 illustrates that the converged system trajectories of the faulty system under the data-based FTC law (33). It implies that the data-based FTC law can guarantee the stability of the closed-loop system with the actuator fault (42) since the fault is estimated and compensated successfully. Fig. 6 shows the control input curve. We can observe that the control input presents a change against the actuator fault after the fault occurs, and an acceptable control performance is achieved.

TABLE I
COMPARISON OF THE SUCCESS RATE OF SOLVING THE HJBE
BETWEEN THE PSOCNN AND GDCNN

	Critic NN training algorithm	Parameters	Success rate of 100 trails	
			Example 1	Example 2
PSOCNN	PSO equations (9) and (10)	$c_1 = 2$, $c_2 = 2$ $r1, r2 \in [0, 1]$, $k = 20$ $\omega_{max} = 0.9$ $\omega_{min} = 0.4$ $\gamma_{max} = 1000$	89%	90%
GDCNN	BP $\dot{w}_{c1} = -\alpha_c \nabla \sigma_c(\hat{x}) \dot{\hat{x}}$ (41)	$\alpha_{c1} = 0.01$ $\alpha_{c2} = 0.2$	31%	57%

Example 2: Consider the inverted pendulum system whose dynamics is expressed as

$$\dot{x} = \begin{bmatrix} x_2 \\ \frac{g}{\ell} \sin(x_1) - \kappa \ell x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{m\ell^2} \end{bmatrix} (u + u_f) \qquad (43)$$

where $x = [x_1, x_2]^T \in \mathbb{R}^2$ and $u \in \mathbb{R}$ are the state and control input variables, respectively. Let $\kappa = 0.2$ and g $= 9.8$ m/s^2 be the frictional factor and the gravitation acceleration, respectively. $m = 1/2$ and $\ell = 1/3$ are the mass and the length of the pendulum bar, respectively. u_f is the additive actuator fault chosen as
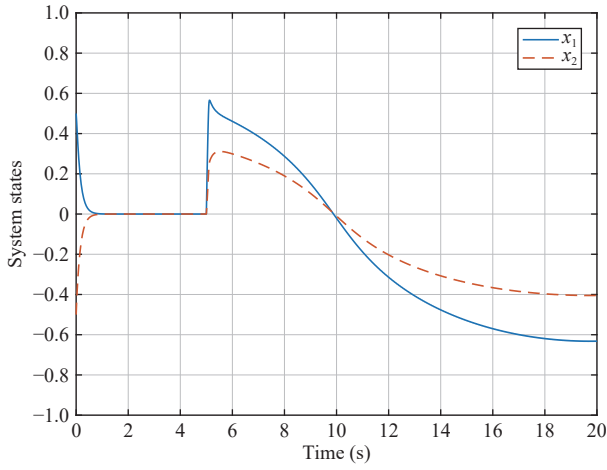
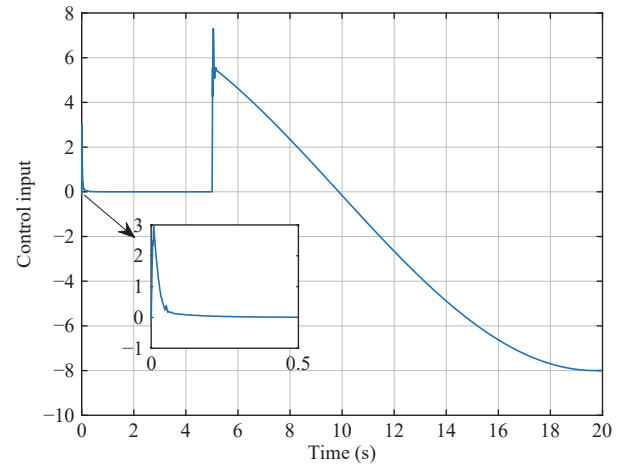Fig. 3.   System states without fault compensation of Example 1.

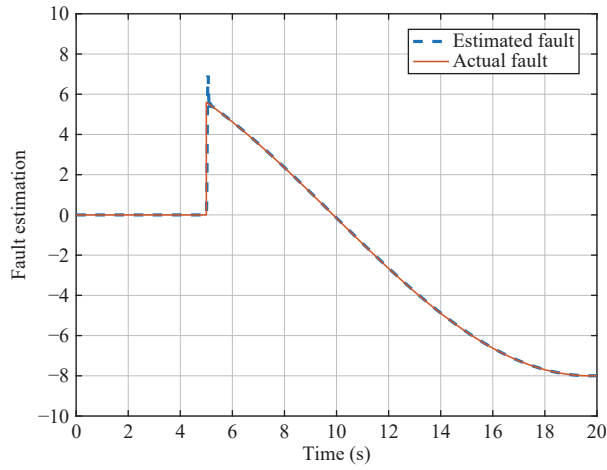Fig. 6.   The control input curve of Example 1.
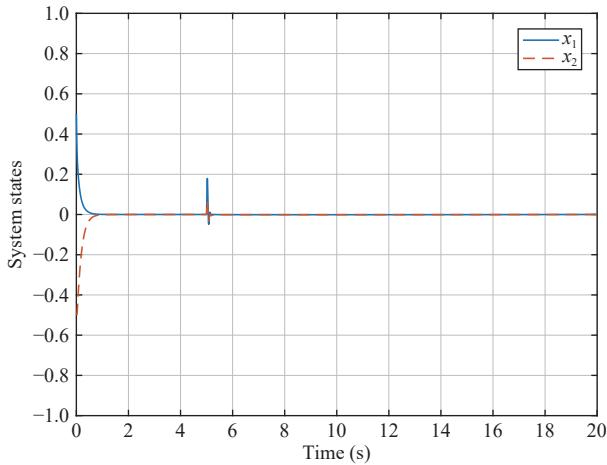

Fig. 4.   Online fault estimation of Example 1.

$\sigma_c(x) = [x_1^2, x_1 x_2, x_2^2]$. Let the initial state vector of the system (44) be $[1, -1]$. The fitness functions and PSO parameters are chosen the same as Example 1. Fig. 7 depicts the converged identification errors. It is obvious that the nonlinear system (44) is well approximated by the NN identifier. Fig. 8 shows that the system states of the system (44) converge to the equilibrium zero but deviate when the actuator fault occurs at $t = 20$ s. It implies that the nominal optimal control law (28) can only guarantee the fault-free system stable. Fig. 9 shows that the actuator fault is well estimated by (34). The system trajectories under the data-based FTC law are given in Fig. 10. It implies that the data-based FTC algorithm can guarantee the stability of the closed-loop system with actuator faults. Fig. 11 shows that the control input changes adaptively against the actuator fault after the fault occurs. These simulation results verify the validity and applicability of the proposed method.
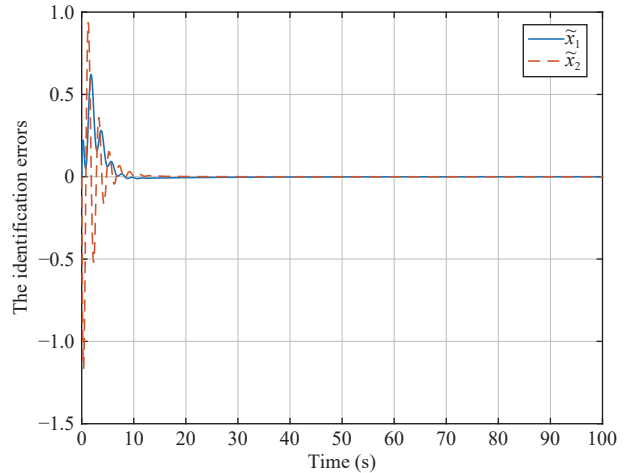

Fig. 5.   The state trajectories under the fault compensation strategy of Example 1.

Fig. 7.   Identification errors of Example 2.

$$u_f = \begin{cases} 0, & 0 \le t \le 20 \\ 2 + 5\sin\left(\dfrac{t}{2\pi}\right), & 20 < t \le 60. \end{cases} \quad (44)$$

The PSO trained identifier NN is built with structure of 5–8–4. The activation function of the critic NN is chosen as

In order to demonstrate the efficiency of the proposed PSOCNN, we have 100 trails and calculate the success rate of solving the HJBEs with both PSOCNN and GDCNN for two examples. For comparison, both PSOCNN and GDCNN employ same critic NNs with same initial values and parameters. With the GDCNN, the critic NN weights are
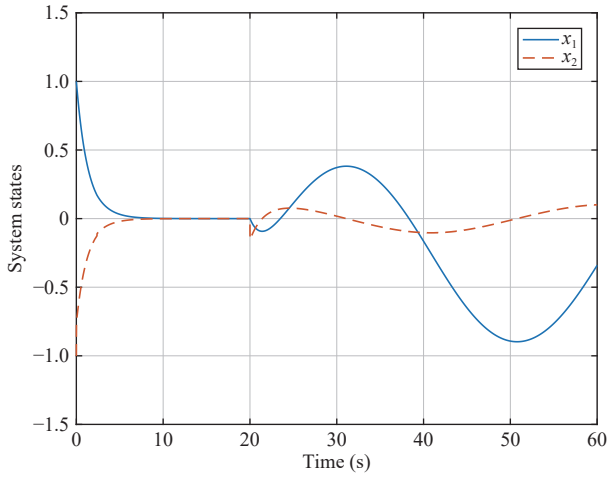
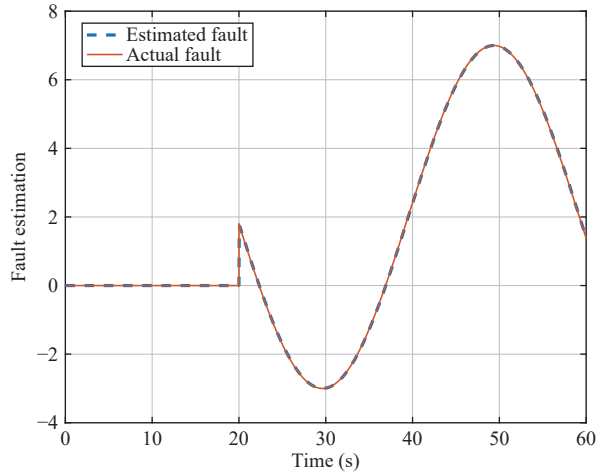Fig. 8.   System states without fault compensation of Example 2.

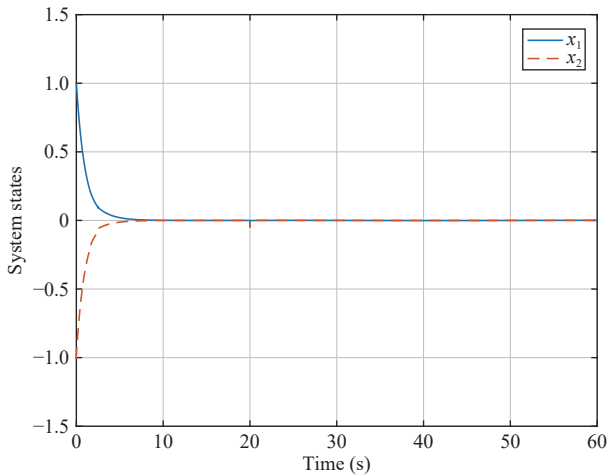Fig. 9.   Online fault estimation of Example 2.

Fig. 10.   The state trajectories under the fault compensation strategy of Example 2.

tuned according to the formula (41) [44]. Through simulations, the learning rates of two examples are chosen as 0.01 and 0.2, respectively. The comparison results are listed in Table I. The statistical success rate shows that the HJBEs can
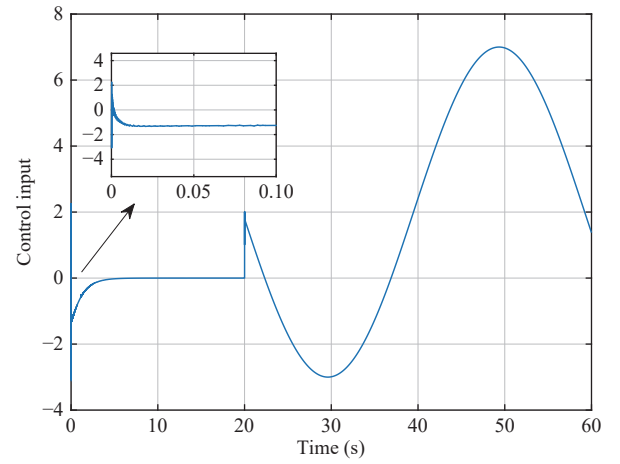

Fig. 11.   The control input curve of Example 2.

be solved with a higher probability with the PSOCNN compared to the general GDCNN, which implies that the PSOCNN is better in producing a good solution for the HJBE than that of GDCNN.

V.   CONCLUSIONS

A data-based FTC algorithm exploiting PSONNs is developed for unknown CT affine nonlinear systems characterized by actuator faults. By constructing a PSO-trained NN identifier, the unknown system dynamics are obtained. Then, the PSOCNN is proposed to approximate the solution of the HJBE for the optimal control. In order to tolerate actuator faults in unknown nonlinear systems, the data-based FTC law is derived by an adaptive compensator. Simulation results show that the proposed data-based FTC algorithm can guarantee the stability of the closed-loop systems with actuator faults. Furthermore, the PSOCNN is better in producing a good solution for the HJBE than that of GDCNN. To the best of our knowledge, the unknown system should be modeled first to estimate the system states and the control matrix before constructing the FTC scheme, which means that the NN identifier is trained off-line and the proposed control scheme cannot be used for non-affine systems. In future work, we will focus on developing an online PSOCNN-based FTC scheme for unknown non-affine nonlinear systems.

REFERENCES

[1] F. L. Lewis, D. Vrabie, and V. L. Syrmos, *Optimal Control*. New York: John Wiley & Sons, 2012.

[2] P. J. Werbos, "Approximate dynamic programming for real-time control and neural modeling," in *Handbook of Intelligent Control: Neural, Fuzzy, and Adaptive Approaches*, D. A. White and D. A. Sofge, Eds. New York, NY, USA: Van Nostrand Reinhold, 1992, ch. 13.

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

[4] D. R. Liu, D. Wang, D. B. Zhao, Q. L. Wei, and N. Jin, "Neural-network-based optimal control for a class of unknown discrete-time nonlinear systems using globalized dual heuristic programming," *IEEE Trans. Autom. Sci. Eng.*, vol. 9, no. 3, pp. 628–634, Jul. 2012.

[5] H. Jiang and H. G. Zhang, "Iterative ADP learning algorithms for discretetime multi-player games," *Artificial Intelligence Review*, vol. 50,

no. 1, pp. 75–91, Jan. 2018.

[6] J. Valasek, J. Doebbler, M. D. Tandale, and A. J. Meade, "Improved adaptivereinforcement learning control for morphing unmanned air vehicles," *IEEE Trans. Syst.*, *Man*, *Cybern.*, *Part B*: *Cybern.*, vol. 38, no. 4, pp. 1014–1020, Aug. 2008.

[7] F. L. Lewis and D. Vrabie, "Reinforcement learning and adaptive dynamic programming for feedback control," *IEEE Circuits and Syst. Mag.*, vol. 9, no. 3, pp. 32–50, Aug. 2009.

[8] F. C. Meng and Y. P. Dai, "Reinforcement learning adaptive control for upper limb rehabilitation robot based on fuzzy neural network," in *Proc. 31st Chinese Control Conf.*, Hefei, China, Jul. 2012, pp. 5157–5161.

[9] D. R. Liu, X. Yang, D. Wang, and Q. L. Wei, "Reinforcement-learning-based robust controller design for continuous-time uncertain nonlinear systems subject to input constraints," *IEEE Trans. Cybern.*, vol. 45, no. 7, pp. 1372–1385, Apr. 2015.

[10] D. Wang, D. R. Liu, and H. L. Li, "Policy iteration algorithm for online design of robust control for a class of continuous-time nonlinear systems," *IEEE Trans. Autom. Sci. Eng.*, vol. 11, no. 2, pp. 627–632, Apr. 2014.

[11] D. R. Liu and Q. L. Wei, "Policy iteration adaptive dynamic programming algorithm for discrete-time nonlinear systems," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 3, pp. 621–634, Sep. 2014.

[12] A. Al-Tamimi, F. L. Lewis, and M. Abu-Khalaf, "Discrete-time nonlinear HJB solution using approximate dynamic programming: Convergence proof," *IEEE Trans. Syst.*, *Man*, *Cybern.*, *Part B*: *Cybern.*, vol. 38, no. 4, pp. 943–949, Jun. 2008.

[13] Y. H. Zhu and D. B. Zhao, "Comprehensive comparison of online ADP algorithms for continuous-time optimal control," *Artificial Intelligence Review*, vol. 49, no. 4, pp. 531–547, Feb. 2018.

[14] D. R. Liu, Q. L. Wei, D. Wang, X. Yang, and H. L. Li, *Adaptive Dynamic Programming with Applications in Optimal Control. Cham*, Switzerland: Springer, 2017.

[15] B. Kiumarsi, K. G. Vamvoudakis, H. Modares, and F. L. Lewis, "Optimal and autonomous control using reinforcement learning: A survey," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 6, pp. 2042–2062, Dec. 2018.

[16] D. R. Liu, Y. C. Xu, Q. L. Wei, and X. L. Liu, "Residential energy scheduling for variable weather solar energy based on adaptive dynamic programming," *IEEE/CAA J. Autom. Sinica*, vol. 5, no. 1, pp. 36–46, 2018.

[17] Q. L. Wei, D. R. Liu, Y. Liu, and R. Z. Song, "Optimal constrained self-learning battery sequential management in microgrid via adaptive dynamic programming," *IEEE/CAA J. Autom. Sinica*, vol. 4, no. 2, pp. 168–176, 2017.

[18] B. Zhao and D. R. Liu, "Event-triggered decentralized tracking control of modular reconfigurable robots through adaptive dynamic programming," *IEEE Trans. Ind. Electron.*, vol. 67, no. 4, pp. 3054–3064, Apr. 2020.

[19] S. C. Tong, B. Y. Huo, and Y. M. Li, "Observer-based adaptive decentralized fuzzy fault-tolerant control of nonlinear large-scale systems with actuator failures," *IEEE Trans. Fuzzy Syst.*, vol. 22, no. 1, pp. 1–15, Jan. 2013.

[20] Y. M. Li, K. K. Sun, and S. C. Tong, "Adaptive fuzzy robust fault-tolerant optimal control for nonlinear large-scale systems," *IEEE Trans. Fuzzy Syst.*, vol. 26, no. 5, pp. 2899–2914, Dec. 2017.

[21] Y. M. Li, K. K. Sun, and S. C. Tong, "Observer-based adaptive fuzzy fault-tolerant optimal control for SISO nonlinear systems," *IEEE Trans. Cybern.*, vol. 49, no. 2, pp. 649–661, Jan. 2018.

[22] B. Zhao, D. R. Liu, and Y. C. Li, "Online fault compensation control based on policy iteration algorithm for a class of affine non-linear systems with actuator failures," *IET Control Theory Appl.*, vol. 10, no. 15, pp. 1816–1823, Oct. 2016.

[23] B. Zhao, D. R. Liu, and Y. C. Li, "Observer based adaptive dynamic programming for fault tolerant control of a class of nonlinear systems," *Inform. Sci.*, vol. 384, no. 1, pp. 21–33, Apr. 2017.

[24] X. Wu, S. J. Zhang, and W. F. Shuang, "Optimal adaptive tracking control for a class of MIMO uncertain nonlinear systems with actuator failures," in *Proc. 36th Chinese Control Conf.*, Dalian, China, Sep. 2017, pp. 6974–6979.

[25] L. Liu, Z. S. Wang, and H. G. Zhang, "Adaptive fault-tolerant tracking control for MIMO discrete-time systems via reinforcement learning algorithm with less learning parameters," *IEEE Trans. Autom. Sci. Eng.*, vol. 14, no. 1, pp. 299–313, Jan. 2016.

[26] Z. S. Wang, L. Liu, H. G. Zhang, and G. Y. Xiao, "Fault-tolerant controller design for a class of nonlinear MIMO discrete-time systems via online reinforcement learning algorithm," *IEEE Trans. Syst.*, *Man*, *Cybern.*, *Syst.*, vol. 46, no. 5, pp. 611–622, May 2016.

[27] Z. S. Wang, L. Liu, and H. G. Zhang, "Neural network-based model-free adaptive fault-tolerant control for discrete-time nonlinear systems with sensor fault," *IEEE Trans. Syst.*, *Man*, *Cybern.*, *Syst.*, vol. 47, no. 8, pp. 2351–2362, Aug. 2017.

[28] H. G. Zhang, K. Zhang, Y. L. Cai, and J. Han, "Adaptive fuzzy fault-tolerant tracking control for partially unknown systems with actuator faults via integral reinforcement learning method," *IEEE Trans. Fuzzy Syst.*, vol. 27, no. 10, pp. 1986–1998, Jan. 2019.

[29] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. IEEE Int. Conf. Neural Network*, Perth, Australia, Dec. 1995, pp. 1942–1948.

[30] Y. H. Shi and R. Eberhart, "A modified particle swarm optimizer," in *Proc. IEEE Int. Conf. Evolutionary Computation*, Anchorage, AK, USA: IEEE, May 1998, pp. 69–73.

[31] M. Clerc and J. Kennedy, "The particle swarm: Explosion, stability and convergence in multi-dimensional complex space," *IEEE Trans. Evolut. Comput.*, vol. 20, no. 1, pp. 1671–1676, Aug. 2002.

[32] S. Martin and M. Choi, "Nonlinear electrical impedance tomography reconstruction using artificial neural networks and particle swarm optimization," *IEEE Trans. Magnetics*, vol. 52, no. 3, pp. 1–4, Oct. 2015.

[33] G. Das, P. K. Pattnaik, and S. K. Padhy, "Artificial neural network trained by particle swarm optimization for non-linear channel equalization," *Expert Systems with Applications*, vol. 41, no. 7, pp. 3491–3496, Jun. 2014.

[34] K. Y. Chan, T. Dillon, E. Chang, and J. Singh, "Prediction of short-term traffic variables using intelligent swarm-based neural networks," *IEEE Trans. Contr. Syst. Technol.*, vol. 21, no. 1, pp. 263–274, Jan. 2013.

[35] Z. W. Zheng, L. Sun, and L. H. Xie, "Error-constrained LOS path following of a surface vessel with actuator saturation and faults," *IEEE Trans. Syst.*, *Man*, *Cybern.*, *Syst.*, vol. 48, no. 10, pp. 1794–1805, Jul. 2017.

[36] B. Jiang, M. Staroswiecki, and V. Cocquempot, "Fault accommodation for nonlinear dynamic systems," *IEEE Trans. Autom. Contr.*, vol. 51, no. 9, pp. 1578–1583, Sep. 2006.

[37] D. R. Liu and Q. L. Wei, "Policy iteration adaptive dynamic programming algorithm for discrete-time nonlinear systems," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 3, pp. 621–634, Sep. 2013.

[38] B. Zhao, D. Wang, G. Shi, D. R. Liu, and Y. C. Li, "Decentralized control for large-scale nonlinear systems with unknown mismatched interconnections via policy iteration," *IEEE Trans. Syst.*, *Man*, *Cybern.*, *Syst.*, vol. 48, no. 10, pp. 1725–1735, Oct. 2017.

[39] M. Jiang, Y. P. Luo, and S. Y. Yang, "Stochastic convergence analysis and parameter selection of the standard particle swarm optimization algorithm," *Information Processing Letters*, vol. 102, no. 1, pp. 8–16, Apr. 2007.

[40] W. Yu, *Recent Advances in Intelligent Control Systems*. London, UK: Springer-Verlag, 2009.

[41] X. Yang, D. R. Liu, and D. Wang, "Reinforcement learning for adaptive optimal control of unknown continuous-time nonlinear systems with input constraints," *Int. J. Control*, vol. 87, no. 3, pp. 553–566, Apr. 2014.

[42] X. N. Zhong and H. B. He, "An event-triggered ADP control approach for continuous-time system with unknown internal states," *IEEE Trans. Cybernetics*, vol. 47, no. 3, pp. 683–694, Apr. 2017.

[43] K. G. Vamvoudakis and F. L. Lewis, "Online actor-critic algorithm to solve the continuous-time infinite horizon optimal control problem," *Automatica*, vol. 46, no. 5, pp. 878–888, May 2010.

[44] D. Wang, D. R. Liu, Q. C. Zhang, and D. B. Zhao, "Data-based adaptive critic designs for nonlinear robust optimal control with uncertain dynamics," *IEEE Trans. Syst.*, *Man*, *Cybern.*, *Syst.*, vol. 46, no. 11, pp. 1–12, Oct. 2015.

[45] X. Yang, D. R. Liu, and Y. Z. Huang, "Neural-network-based online optimal control for uncertain non-linear continuous-time systems with control constraints," *IET Control Theory Appl.*, vol. 7, no. 17,

pp. 2037–2047, Nov. 2013.

[46] B. Zhao, D. R. Liu, and C. Luo, "Reinforcement learning-based optimal stabilization for unknown nonlinear systems subject to inputs with uncertain constraints," *IEEE Trans. Neural Netw. Learn. Syst.*, 2019. doi: 10.1109/TNNLS.2019.2954983.

[47] D. R. Liu, H. L. Li, and D. Wang, "Online synchronous approximate optimal learning algorithm for multi-player non-zero-sum games with unknown dynamics," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 44, no. 8, pp. 1015–1027, Jan. 2014.

[48] Y. Sokolov, R. Kozma, L. D. Werbos, and P. J. Werbos, "Complete stability analysis of a heuristic approximate dynamic programming control design," *Automatica*, vol. 59, pp. 9–18, Sep. 2015.

[49] D. R. Liu, D. Wang, F.-Y. Wang, H. L. Li, and X. Yang, "Neural-networkbased online HJB solution for optimal robust guaranteed cost control of continuous-time uncertain nonlinear systems," *IEEE Trans. Cybern.*, vol. 44, no. 12, pp. 2834–2847, Sep. 2014.

[50] B. Zhao, D. R. Liu, and C. Alippi, "Sliding mode surface-based approximate optimal control for uncertain nonlinear systems with asymptotically stable critic structure," *IEEE Trans. Cybern.*, Jan. 2020. doi: 10.1109/TCYB.2019.2962011.

[51] D. B. Zhao, J. Q. Yi, and D. R. Liu, "Particle swarn optimized adaptive dynamic programming," in *Proc. IEEE Int. Symp. Approx. Dyn. Programm. Reinforcement Learn.*, Honolulu, HI, USA: IEEE, Apr. 2007, pp. 32–37.

Haowei Lin received the B.S. degree at the School of Automation, Guangdong Polytechnic Normal University, in 2017. He is currently pursuing the M.S. degree at the School of Automation, Guangdong University of Technology. His research interests include adaptive dynamic programming and fault tolerant control.

Bo Zhao (M'16) received the B.S. degree in automation, and Ph.D. degree in control science and engineering, all from Jilin University, in 2009 and 2014, respectively. He was a Post-Doctoral Fellow with the State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, from 2014 to 2017. Then, he joined the State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, from 2017 to 2018. He is currently an Associate Professor with the School of Systems Science, Beijing Normal University, Beijing, China. He has authored or coauthored over 80 journal and conference papers. His research interests include adaptive dynamic programming, robot control, fault diagnosis and tolerant control, optimal control, and artificial intelligence-based control. He was the secretary of Adaptive Dynamic Programming and Reinforcement Learning Professional Committee of Chinese Automation Association (CAA), and the secretary of 2017 the 24th International Conference on Neural Information Processing. He is IEEE Member, Asian-Pacific Neural Network Society (APNNS) Member and CAA Member.

Derong Liu (S'91−M'94−SM'96−F'05) received the Ph.D. degree in electrical engineering from the University of Notre Dame, Notre Dame, IN, USA, in 1994. He was a Staff Fellow with General Motors Research and Development Center, Warren, MI, USA, from 1993 to 1995. He was an Assistant Professor with the Department of Electrical and Computer Engineering, Stevens Institute of Technology, Hoboken, NJ, USA, from 1995 to 1999. He joined the University of Illinois at Chicago, Chicago, IL, USA, in 1999, and became a Full Professor of Electrical and Computer Engineering, and of Computer Science, in 2006. He served as the Associate Director of the State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Beijing, China, from 2010 to 2015. He is currently a Full Professor with the School of Automation, Guangdong University of Technology, Guangzhou, China. He has authored or co-authored 19 books. Prof. Liu was a recipient of the Michael J. Birck Fellowship from the University of Notre Dame, in 1990, the Harvey N. Davis Distinguished Teaching Award from the Stevens Institute of Technology, in 1997, the Faculty Early Career Development Award from the National Science Foundation in 1999, the University Scholar Award from the University of Illinois, from 2006 to 2009, and the Overseas Outstanding Young Scholar Award from the National Natural Science Foundation of China in 2008. He was the Editor-in-Chief of the *IEEE Transactions on Neural Networks and Learning Systems* from 2010 to 2015. He is the Editor-in-Chief of *Artificial Intelligence Review* (Springer). He is a Fellow of the International Neural Network Society and the International Association of Pattern Recognition.

Cesare Alippi (F'06) is a Professor with the Politecnico di Milano, Milano, Italy and Università della Svizzera Italiana, Lugano, Switzerland. Alippi is a visiting professor at the Guangdong University of Technology in Guangzhou, China and Consultant Professor at the Northwestern Polytechnical Univeristy in Xi'an, China. He has been a Visiting Researcher/Professor at UCL (UK), MIT (USA), ESPCI (F), CASIA (PRC), A*STAR (SIN), UKobe (JP). Alippi is an IEEE Fellow, Member of the Administrative Committee of the IEEE Computational Intelligence Society, Board of Governors member of the International Neural Network Society, Board of Directors member of the European Neural Network Society, Past Vice-President education of the IEEE Computational Intelligence Society, past associate editor of the *IEEE Transactions on Emerging Topics in Computational Intelligence*, the *IEEE Computational Intelligence Magazine*, the *IEEE Transactions on Instrumentation and Measurements*, the *IEEE Transactions on Neural Networks (and Learning Systems)*. He received the 2018 IEEE CIS Outstanding Computational Intelligence Magazine Award, the 2016 Gabor Award from the International Neural Networks Society and the IEEE Computational Intelligence Society Outstanding Transactions on Neural Networks and Learning Systems Paper Award; in 2013 the IBM Faculty award; in 2004 the IEEE Instrumentation and Measurement Society Young Engineer Award. Current research activity addresses Adaptation and Learning in Non-stationary Environments, Graph Learning and Intelligence for Embedded, IoT and Cyber-physical Systems. He holds 8 patents, has published one monograph book, 7 edited books and about 200 papers in international journals and conference proceedings.