2019 12th Asian Control Conference (ASCC)
Kitakyusyu International Conference Center
Japan, June 9-12, 2019

Parameter Self-Tuning of SISO Compact-Form Model-Free Adaptive Controller based on Neural Network with System Error Set as Input

Chen CHEN, Xueyuan LI, Ye YANG, Jiarong XU, Zuwei LIAO, Xinggao LIU, Jinshui CHEN, Jiangang LU*

Abstract— **Model-Free Adaptive Control (MFAC) is a new data-driven control method, which depends only on the input/output (I/O) measurement data rather than the mathematical model information of the actual controlled system. The SISO MFAC based on the compact-form dynamic linearization (SISO-CFMFAC) is a promising approach to control the SISO nonlinear systems. However, the parameters in the SISO-CFMFAC should be tuned carefully before being put into use. Unfortunately, so far the parameter tuning of SISO-CFMFAC is still a laborious, time-consuming and cost-consuming work. In this paper, a novel parameter self-tuning approach of SISO-CFMFAC based on back propagation Neural Network with System Error set as input (SISO-CFMFAC-NNSE) is proposed, and then verified by using a typical time-varying nonlinear SISO system. Results show that the proposed controller named SISO-CFMFAC-NNSE can achieve better control stability and accuracy than the existing controller of SISO-CFMFAC.**

Keywords— SISO Compact-Form MFAC; Parameter Self-tuning; BP Neural Network

I. INTRODUCTION

In many practical situations, the structure of the plant is often difficult to identify, sometimes, it is impossible. This motivates the research on the model-free control or data-driven control approaches, which does not require any model information of the controlled system. Since the input and output measurement data of the system can precisely reflect the valuable state information of process operations and equipment, it becomes meaningful to design the controller by directly using the I/O data when the accurate model of the controlled plant is unavailable [1-2].

Up to now, data-driven control (DDC) methods can be divided into three categories: (a) DDC based on online data, such as Model-Free Adaptive Control (MFAC) [3], Unfalsified Control (UC) [4]. (b) DDC based on offline data, such as Iterative Feedback Tuning (IFT) [5], Proportional Integral Derivative (PID) [6]. (c) DDC based on a combination of online and offline data, such as Iterative Learning Control (ILC) [7], Lazy Learning (LL) [8].

Model free adaptive control (MFAC), based on the online I/O measurement data, is originally proposed by Hou [3, 9] for a class of unknown nonlinear discrete-time systems, and has been successfully applied to many control systems [10-14]. Due to the increasing complexity of many plants investigated, the dynamics of these plants become more and more difficult to identify, which makes MFAC outstanding among many control methods. Besides, a large number of simulations and applications show that the MFAC is suitable for many practical systems with a small computational burden and strong robustness to the system uncertainties.

The basis of MFAC is the novel dynamic linearization(DL) method [15], including the compact form dynamic linearization (CFDL), the partial form dynamic linearization (PFDL) and the full form dynamic linearization (FFDL), which builds an equivalent linearization data model of the original nonlinear systems at each operation point of the closed-loop system by introducing the new concept, including the pseudo partial derivative (PPD) or the pseudo gradient(PG) vector for SISO nonlinear system, and pseudo Jaccobian matrix (PJM) for MIMO nonlinear system [9].

The SISO MFAC [16] based on the compact form dynamic linearization (SISO-CFMFAC) is a promising approach to control a class of SISO nonlinear discrete-time systems. However, choosing appropriate and robust controller parameters is also of great importance for an actual system. Up to now, there is no systematic method for the parameters tuning of SISO-CFMFAC. In [17], an improved particle swarm optimization (PSO) was used to optimize the parameters in MFAC, which wasted a lot of time due to the iteration of the algorithm and was difficult to apply in practical systems. In [18], the parameter tuning of an improved MFAC was discussed, but it only target at the second order universal model. In [19], a parameter tuning method based on minimum entropy optimization is proposed, facing the same problem with [17]. In [20], an offline parameters tuning method was proposed by using a nonlinear Virtual Reference Feedback Tuning (VRFT) algorithm.

In this paper, a novel parameter self-tuning approach of SISO-CFMFAC based on back propagation neural network with system error set as input (SISO-CFMFAC-NNSE) is

*Research supported by the National Natural Science Foundation of China (No. 61590925, No. U1609212, No. U1509211), the National Key Research and Development Plan of China (No. 2017YFC0210403), and the Robotics Institute of Zhejiang University under Grant K11901.

Chen CHEN, Xueyuan LI, Ye YANG, Jiarong XU, Xinggao LIU, Jinshui CHEN are with the State Key Laboratory of Industrial Control Technology, College of Control Science and Engineering, Zhejiang University, Hangzhou 310027, Zhejiang, China.
Zuwei LIAO is with the State Key Laboratory of Chemical Engineering, College of Chemical and Biological Engineering, Zhejiang University, Hangzhou 310027, Zhejiang, China.
Jiangang LU is with the State Key Laboratory of Industrial Control Technology, College of Control Science and Engineering, Zhejiang University, Hangzhou 310027, Zhejiang, China; he is also with the Robotics Institute of Zhejiang University, Yuyao Technology Innovation Center, Ningbo 315400, Zhejiang, China (corresponding author; e-mail: lujg@zju.edu.cn).

proposed, which achieves better control stability and accuracy than the existing controller of SISO-CFMFAC. In addition, the innovation of this paper is also reflected in the online tuning of parameters, which has an excellent application prospect compared with the offline parameter tuning method that existed.

The rest of this paper is organized as follows. Section II introduces the dynamic linearization technique and an ideal controller of SISO-CFMFAC. In section III, parameter self-tuning approach of SISO-CFMFAC based on back propagation Neural Network with System Error set as input (SISO-CFMFAC-NNSE) is present. Simulation comparison results are shown in Section IV. In section V, some conclusions are given and possible future work is also discussed.

II. SISO Compact-Form Dynamic Linearization based Model Free Adaptive Control

In this section, the equivalent dynamic linearization data model of SISO-CFMFAC is given. The descriptions are then used in the following section to design the parameter self-tuning in SISO-CFMFAC using a neural network. In this section, the equivalent dynamic linearization data model of SISO-CFMFAC is given. The descriptions are then used in the following section to design the parameter self-tuning in SISO-CFMFAC using a neural network.

Consider the following SISO nonlinear discrete-time system:

$$y(k+1)=f(y(k),\cdots,y(k-n_y),u(k),\cdots,u(k-n_u)) \quad (1)$$

where $y(k)\in\mathbf{R}$ is the system outputs and $u(k)\in\mathbf{R}$ is the system inputs, n_y , n_u are the orders of $y(k)$ and $u(k)$, respectively, $f(\cdots)$ is a general nonlinear function.

In discrete-time system, minimizing the one-step-head error may lead to a large control input and thus harmful to the system. Therefore, the weighted one-step-head control input criterion function is considered:

$$J(\Delta u(k))=\left|y^*(k+1)-y(k+1)\right|^2+\lambda\left|u(k)-u(k-1)\right|^2 \quad (2)$$

where $\lambda>0$ is a weighting factor which is introduced to restrict the variance of the control input $u(k)$, and $y^*(k)$ is the expecting output of controlled plant.

By setting the derivative of (2) in respect to $u(k)$ equals to zero, an iterative function is obtained

$$u(k)=u(k\text{-}1)+\frac{\rho\varphi(k)}{\lambda+\left|\varphi(k)\right|^2}(y^*(k+1)-y(k)) \quad (3)$$

where $\rho\in(0,1]$ is introduced as a penalty factor for a more general and flexible controlling rule. And φ(k) is the pseudo partial derivative (PPD). Besides, we have noticed that λ is not only a penalty factor on $\Delta u(k)$, but also is a part of denominator in (3), thus it is an important parameter for the SISO-CFMFAC scheme.

The estimation of the time-varying value of pseudo partial derivative (PPD) in SISO CFDL data model is considered here:

$$\hat{\varphi}(k)=\hat{\varphi}(k-1)+\frac{\eta\Delta u(k-1)}{\mu+\Delta u(k-1)^2}(\Delta y(k)-\hat{\varphi}(k-1)\Delta u(k-1)) \quad (4)$$

where $\eta\in(0,1]$ is a step size constant added in order to make the Eq. (4) general and will be used in the analytical stability proofs. $\hat{\varphi}(k)$ is the approximation of $\varphi(k)$, and μ is a weighting factor which is introduced to punish the sudden change of $\varphi(k)$.

Additionally, in order to make the condition $\Delta u(k)\neq 0$, and to ensure the adaptive law to possess strong tracking capability on time-varying parameters, a reset algorithm is considered in the following structure:

$$\hat{\varphi}(k)=\hat{\varphi}(1)\text{ , if }\left|\hat{\varphi}(k)\right|\le\varepsilon\text{ or }\left|\Delta u(k-1)\right|\le\varepsilon$$
$$\text{or } sign(\hat{\varphi}(k))\neq sign(\hat{\varphi}(1)) \quad (5)$$

where ε is a sufficiently small positive constant which is often selected as 10^{-5} or 10^{-6}, and $\hat{\varphi}(1)$ is the initial estimation value of $\hat{\varphi}(k)$.

In the SISO-CFMFAC scheme above, the number of controller parameters needed to be adjusted online for SISO nonlinear system is only one, that is, the PPD.

III. Parameter Self-tuning based on Neural Network

So far, the parameter tuning of SISO-CFMFAC is still a laborious, time-consuming and cost-consuming work, which is restricting the development of MFAC. As can be seen from the analysis in Section II, the suitable choice of weighting constant λ and step-size ρ can improve the performance of the control system significantly. Therefore, the research about parameter self-tuning of SISO-CFMFAC, including λ and ρ , is of great significance to the applications in practice engineering.

Although there are already some methods for offline parameter tuning in literature as mentioned in Section I, it is still difficult to apply to the actual system due to their limitations. In the case of control plants subject to continuous parameter changes or external disturbances, online parameter adjustment is required. In this section, an online parameter self-tuning method of SISO-CFMFAC based on BP neural network with system error set as input (SISO-CFMFAC-NNSE) is presented, and then introduce the design and analysis of the novel model later.

A. Model Presented

The novel controller based on back propagation neural network which is designed combining traditional SISO-CFMFAC strategy with neural network has created a new concept and a new tool for control. The blocked diagram of the proposed controller called SISO-CFMFAC-NNSE could be set up as depicted in Figure 1.

According to the principle of minimum $e^2(k)$, BP neural network can auto-adjust its weights to vary λ and ρ in SISO-CFMFAC with the system error set, including $e(k)$, $\sum_{t=0}^{k} e(t)$ and $e(k)-e(k-1)$, as input. Then a specific algorithm will be introduced later.

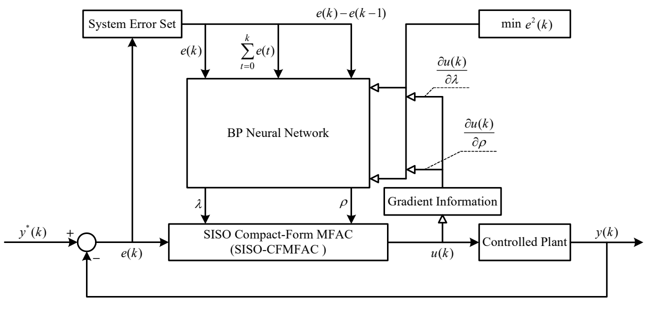


Figure 1. Block diagram of an auto-tuned SISO-CFMFAC with BPNN.

B. Design and Analysis of SISO-CFMFAC-NNSE

Back Propagation is commonly used by the gradient descent optimization algorithm to adjust the weight of neurons by calculating the gradient of the loss function. Figure 2 shows the structure of the BPNN used to auto-tuned the SISO-CFMFAC parameters λ and ρ.

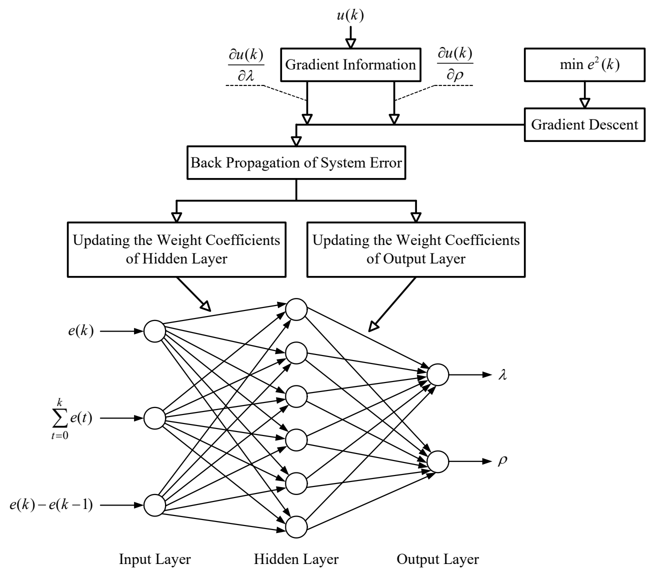


Figure 2. Structure of the BPNN used to auto-tuned the SISO-CFMFAC parameters.

Learning algorithm of BPNN is described as follows:

- Initialize the structure of BP neural network by setting three neurons on the input layer, six neurons on the hidden layer and two neurons on the output layer.
- The network input is $x_j\,(j=1,2,3)$

$$x_1 = e(k)$$
$$x_2 = \sum_{t=0}^{k} e(t) \tag{6}$$
$$x_3 = e(k) - e(k-1)$$

- The input and output of the hidden layer is shown in formulas (7) and (8).

$$hnet_i(k) = \sum_{j=1}^{3} \omega_{ij} x_j(k) \tag{7}$$

$$hide_i(k) = f(hnet_i(k)), i = 1,2,...,6 \tag{8}$$

where ω_{ij} are the connection weights parameters between input layer and hidden layer. The activation function of hidden layer is as follows:

$$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{9}$$

- Inputs and outputs of the output layer is shown in formulas (10).

$$onet_l(k) = \sum_{i=1}^{6} \omega_{li} hide_i(k)$$
$$out_l(k) = g(onet_1(k)) \qquad l = 1,2 \tag{10}$$
$$\lambda = out_1(k)$$
$$\rho = out_2(k)$$

where ω_{li} are the connection weights parameters between hidden layer and output layer. The activation function of output layer is as follows:

$$\mathrm{g}(x) = \frac{e^x}{e^x + e^{-x}} \tag{11}$$

- The performance index J is described as follows:

$$J = \frac{1}{2}[y^*(k+1) - y(k+1)]^2 = \frac{1}{2}e^2(k+1) \tag{12}$$

The partial derivative of J with respect to the weighting coefficients ω_{li} and ω_{ij}, can be obtained by applying the chain rule:

$$\Delta\omega_{li}(k+1) = -\beta\frac{\partial J}{\partial \omega_{li}} + \alpha\Delta\omega_{li}(k)$$
$$\frac{\partial J}{\partial \omega_{li}} = \frac{\partial J}{\partial y(k+1)}\frac{\partial y(k+1)}{\partial u(k)}\frac{\partial u(k)}{\partial out_l(k)}\frac{\partial out_l(k)}{\partial onet_l(k)}\frac{\partial onet_l(k)}{\partial \omega_{li}}$$
$$\Delta\omega_{ij}(k+1) = -\beta\frac{\partial J}{\partial \omega_{ij}} + \alpha\Delta\omega_{ij}(k)$$
$$\frac{\partial J}{\partial \omega_{ij}} = \frac{\partial J}{\partial y(k+1)}\frac{\partial y(k+1)}{\partial u(k)}\frac{\partial u(k)}{\partial out_l(k)}\frac{\partial out_l(k)}{\partial onet_l(k)}\frac{\partial onet_l(k)}{\partial hide_i(\mathrm{k})}\frac{\partial hide_i(\mathrm{k})}{\partial \omega_{ij}} \tag{13}$$

where α and β are inertia coefficient and learning rate. The partial derivative of u with respect to λ and ρ:

$$\frac{\partial u(k)}{\partial \lambda}=\frac{\rho\hat{\varphi}(k)(y^*(k+1)-y(k))}{(\lambda+|\hat{\varphi}(k)|^2)^2}$$
$$\frac{\partial u(k)}{\partial \rho}=\frac{\hat{\varphi}(k)(y^*(k+1)-y(k))}{\lambda+|\hat{\varphi}(k)|^2} \quad (14)$$

- The performance index J is described as follows:

$$\Delta w_{li}(k+1)=\beta o\delta_l hide_i(k)+\alpha\Delta w_{li}(k)$$
$$o\delta_l=e(k+1)sign\left[\frac{\partial y(k+1)}{\partial u(k)}\right]\frac{\partial u(k)}{\partial out_l}g'[onet_l(k)]$$
$$\Delta w_{ij}(k+1)=\beta h\delta_i x_j(k)+\alpha\Delta w_{ij}(k) \quad (15)$$
$$h\delta_i=f'[hnet_i(k)]\sum_{l=1}^{2}o\delta_l w_{li}(k)$$
$$i=1,2,3,4,5,6$$

The above is a specific algorithm of the BPNN used to auto-tuned the SISO-CFMFAC parameters with the system error set as input, which is called SISO-CFMFAC-NNSE.

IV. Simulation Study

Here, a typical time-varying nonlinear SISO system is given to show the effectiveness and advantages of the SISO-CFMFAC-NNSE. In the following simulations, all mathematical model just serves as I/O data generator for the systems to be controlled, no any information of them will be included in the controller design.

Consider the nonlinear system [9]:

$$y(k+1)=\begin{cases}\dfrac{5y(k)y(k-1)}{1+y^2(k)+y^2(k-1)+y^2(k-2)}+u(k)+1.1u(k-1), & k\le 500\\ \dfrac{2.5y(k)y(k-1)}{1+y^2(k)+y^2(k-1)}+1.2u(k)+1.4u(k-1) & \\ \quad +0.7\sin(0.5(y(k)+y(k-1)))\cos(0.5(y(k)+y(k-1))), & k>500\end{cases}$$

The desired trajectory is:

$$y^*(k+1)=\begin{cases}5\sin(k\pi/50)+2\cos(k\pi/100), & k\le 300\\ 5(-1)^{\mathrm{round}(k/100)}, & 300<k\le 700\\ 5\sin(k\pi/50)+2\cos(k\pi/100), & k>700\end{cases}$$

The initial values are: $u(1)=u(2)=0, y(1)=-1, y(2)=1$, $\varphi(1)=2$. These values are cited from [9]. The controller parameters are: $\eta=\mu=1, \varepsilon=10^{-5}$. The structure of BP neural network is 3-6-2, inertia coefficient α=0.01, learning rate β=0.01, initial weight value ω_{ij}, ω_{li} are random between $[-0.05, 0.05]$.

The weighting constant λ and step-size ρ are two important design parameters for the SISO-CFMFAC control system as mentioned above. To demonstrate the impact of λ and ρ on the efficiency of SISO-CFMFAC, a comparison between fixed and auto-tuned parameter value is presented. Two special parameters of SISO-CFMFAC scheme are set to be $\lambda=2, \rho=0.5$ [9]. The simulation results are shown in Figure 3, where Figure 3 (a)-(b) shows the tracking performance of the output y and the control input u respectively. Figure 4 shows the parameter self-tuning results of λ, ρ.

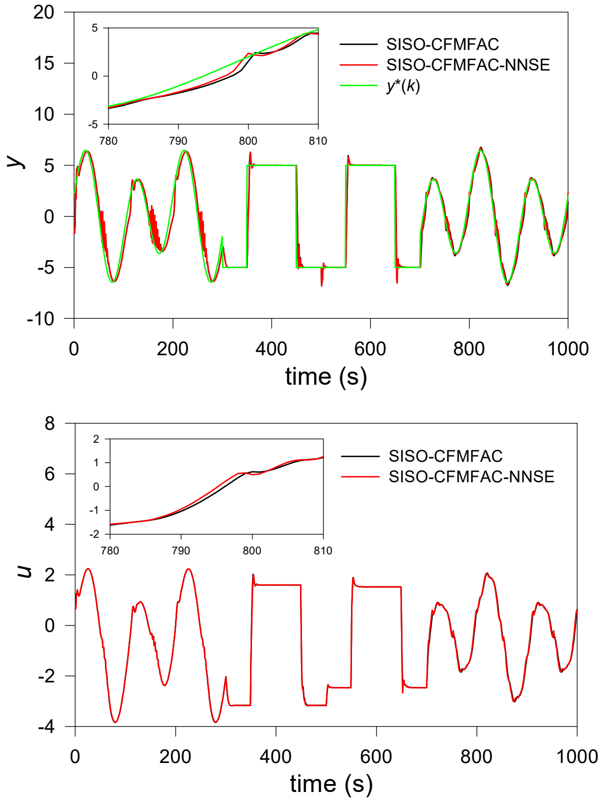
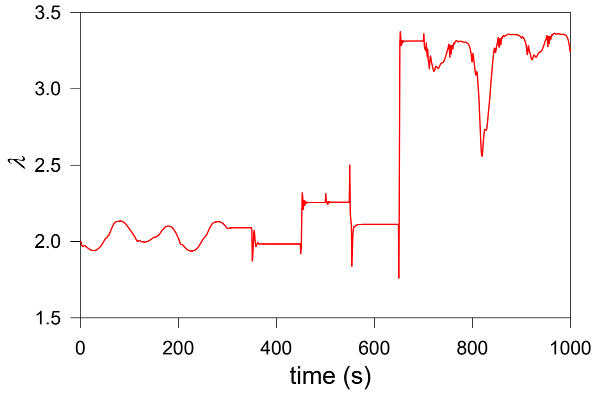


Figure 3. Simulation results. (a) Tracking performance y(k). (b) Control input u(k).



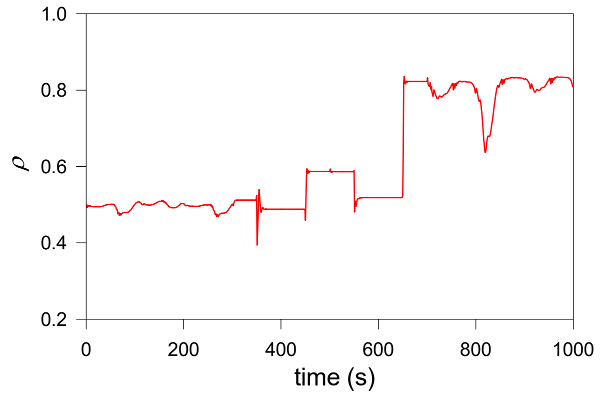


Figure 4. Parameter self-tuning results for λ and ρ

Two performance indexes of the controlled variable are given for comparison.

- Root Mean Square Error (RMSE)

$$RMSE = \sqrt{\frac{1}{N}\sum_{k=1}^{N} e(k)^2} \quad (16)$$

- Poor Control Ratio (PCR)

$$PCR(\zeta) = \frac{1}{N}\sum_{k=1}^{N} p(k,\zeta) \quad (17)$$

$$p(k,\zeta) = \begin{cases} 1, & \text{if } |y(k) - y^*(k)| > \zeta \\ 0, & \text{else} \end{cases} \quad (18)$$

where ζ is a small positive constant.

The quantitative comparison indexes are presented in TABLE I.

TABLE I. COMPARISON OF PERFORMANCE INDEXES

Index	SISO-CFMFAC	SISO-CFMFAC-NNSE	Improvement
RMSE	1.16	1.14	1.72%
PCR(1)	0.185	0.149	19.5%

Root Mean Square Root (RMSE) and Poor Control Ratio (PCR) offers a better notion of the results, leading to the conclusion that the auto-tuned SISO-CFMFAC is better than the conventional SISO-CFMFAC control, as can be seen in Figure 3. Thus, it demonstrates the effectiveness of the proposed control algorithm. In addition, it can also be seen that the curve of actual output under SISO-CFMFAC-NNSE tracks precisely with the reference with smaller overshoot.

V. Conclusion

In this paper, a novel parameter self-tuning approach of SISO-CFMFAC based on back propagation neural network with system error set as input (SISO-CFMFAC-NNSE) is proposed. The optimal parameter obtained from the BPNN can assure that the system has better dynamic performance. From the perspective of algorithm design, the proposed control scheme is simpler than SISO-CFMFAC because of its fewer adjustable parameters. Future researches will be focused on the application of the control strategy to more advanced systems with higher efficiency.

References

[1] Z. S. Hou, and Z. Wang, "From model-based control to data-driven control: Survey, classification and perspective," *Information Sciences,* vol. 235, pp. 3-35, Jun. 2013.

[2] Z. S. Hou, R. Chi and H. Gao, "An Overview of Dynamic-Linearization-Based Data-Driven Control and Applications," *IEEE Transactions on Industrial Electronics,* vol. 64, no. 5, pp. 4076-4090, May. 2017.

[3] Z. S. Hou, "The parameter identification, adaptive control and model free learning adaptive control for nonlinear systems," Shenyang: North-eastern University, 1994.

[4] M. G. Safonov and T. C. Tsao, "The unfalsified control concept: A direct path from experiment to controller," *Feedback Control, Nonlinear Systems, and Complexity,* Springer, Berlin, Heidelberg, 1995, pp. 196-214.

[5] H. Hjalmarsson, M. Gevers, S. Gunnarsson and O. Lequin, "Iterative feedback tuning: theory and applications," *IEEE Control Systems Magazine*, vol. 18, no. 4, pp. 26-41, Aug. 1998.

[6] K. J. Åström, and T. Hägglund, *PID controllers: theory, design, and tuning*, 2nd ed. Instrument society of America, 1995.

[7] Y. Q. Chen, K. L. Moore, and H. S. Ahn, "Iterative learning control." *Encyclopedia of the Sciences of Learning*, Springer, Boston, MA, 2012, pp. 1648-1652.

[8] D. Wettschereck, D. W. Aha, and T. Mohri, "A review and empirical evaluation of feature weighting methods for a class of lazy learning algorithms," *Artificial Intelligence Review,* vol. 11, issue. 1-5, pp. 273-314, Feb. 1997

[9] Z. S. Hou, and S. T. Jin, "*Model free adaptive control: theory and applications*, " CRC press, 2013.

[10] J. Qi, R. Cao, Z. Hou, and H. Zhou, "The model-free adaptive control for complex connected systems in the H-type motion platform," *2017 6th Data Driven Control and Learning Systems (DDCLS)*, Chongqing, 2017, pp. 129-134.

[11] X. Yu, Z. S. Hou, and X. Zhang, "Model-free Adaptive Control for a Vapour-Compression Refrigeration Benchmark Process," *IFAC* [PapersOnLine], vol. 51, no. 4, pp. 527-532, June. 2018.

[12] X. Bu, Z. S. Hou and H. Zhang, "Data-Driven Multiagent Systems Consensus Tracking Using Model Free Adaptive Control," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 5, pp. 1514-1524, May. 2018.

[13] W. Zhang, D. Xu, X. Lou, W. Yan and W. Yang, "Power Management of Battery Energy Storage System Using Model Free Adaptive Control," *2018 IEEE 7th Data Driven Control and Learning Systems Conference (DDCLS)*, Enshi, 2018, pp. 798-801.

[14] T. Lei and Z. S. Hou, "Model Free Adaptive Perimeter Control for Two-Region Urban Traffic System with Input and Output Constraints," *2018 IEEE 7th Data Driven Control and Learning Systems Conference (DDCLS)*, Enshi, 2018, pp. 757-762.

[15] Z. S. Hou, R. H. Chi and H. Gao, "An Overview of Dynamic-Linearization-Based Data-Driven Control and Applications," *IEEE Transactions on Industrial Electronics*, vol. 64, no. 5, pp. 4076-4090, May 2017.

[16] Z. S. Hou and W. H. Huang, "The model-free learning adaptive control of a class of SISO nonlinear systems," *Proceedings of the 1997 American Control Conference (Cat. No.97CH36041)*, Albuquerque, NM, USA, 1997, pp. 343-344 vol.1.

[17] L. dos Santos Coelho, and A. A. R. Coelho, "Model-free adaptive control optimization using a chaotic particle swarm approach," *Chaos, Solitons & Fractals*, vol. 41, no. 4, pp. 2001-2009, Aug. 2009.

[18] J. Wang, C. Ji, L. L. Cao, and Q. B. Jin, "Model free adaptive control and parameter tuning based on second order universal model, " *Journal of Central South University (Science and Technology)*, vol. 43, no.5, pp. 1795-1802, May. 2012.

[19] C. Ji, J. Wang, L. Cao and Q. Jin, "Parameters tuning of model free adaptive control based on minimum entropy," *IEEE/CAA Journal of Automatica Sinica*, vol. 1, no. 4, pp. 361-371, Oct. 2014.
[20] R. C. Roman, et al. "Data-driven model-free adaptive control tuned by virtual reference feedback tuning." *Acta Polytechnica Hungarica*, vol. 13, no. 1, pp. 83-96, Jan. 2016.