

基于强化学习的多智能体系统一致性跟踪控制算法

刘人志

广东工业大学自动化学院, 广东 广州

收稿日期: 2025年3月22日; 录用日期: 2025年4月22日; 发布日期: 2025年4月29日

摘要

本文提出了一种新颖的基于强化学习的无模型自适应控制算法, 适用于具有未知动态的离散时间非线性多智能体系统。采用等效动态线性化算法来设计最优控制器。针对Q学习策略和演员-评论家(actor-critic)神经网络进行了重构, 以促进一致性控制。所提出的强化学习方法能够仅基于输入-输出数据实时动态调整线性化参数。通过数值仿真验证了该方法的有效性。

关键词

强化学习, 非线性多智能体系统, 跟踪控制, 动态线性化

Reinforcement Learning-Based Consensus Tracking Control Algorithm for Multi-Agent Systems

Renzhi Liu

School of Automation, Guangdong University of Technology, Guangzhou Guangdong

Received: Mar. 22nd, 2025; accepted: Apr. 22nd, 2025; published: Apr. 29th, 2025

Abstract

This paper presents a novel reinforcement learning-based model-free adaptive control algorithm for discrete-time nonlinear multi-agent systems with unknown dynamics. The equivalent dynamic linearization algorithm is employed to design an optimal controller. The Q-Learning strategy and actor-critic neural network are restructured to facilitate consensus control. The proposed reinforcement learning approach dynamically adjusts linearization parameters in real-time using only input-output data. Numerical simulations validate the method's effectiveness.

文章引用: 刘人志. 基于强化学习的多智能体系统一致性跟踪控制算法[J]. 计算机科学与应用, 2025, 15(4): 374-381.
DOI: 10.12677/csa.2025.154110

Keywords

Reinforcement Learning, Nonlinear Multi-Agent System, Tracking Control, Dynamic Linearization

Copyright © 2025 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

1. 引言

近年来,随着计算能力的提升,神经网络在解决复杂非线性系统的跟踪控制问题方面展现出了巨大潜力。强化学习(Reinforcement Learning, RL)是一种机器学习的分支,其中智能体(agent)通过与环境的交互进行决策训练。在强化学习中,智能体通过试错学习,并根据获得的奖励或惩罚来优化决策策略。深度学习的引入使强化学习能够处理复杂的高维输入空间[1],从而大幅扩展了其适用范围。随着强化学习的发展及其有效性的提升,越来越多的控制方法基于强化学习算法被设计。在[2]中,强化学习方法通过跟踪误差变换实现了数据驱动的最优控制器的演化;在[3]中,强化学习被集成到迭代学习控制中。

无模型自适应控制(Model-Free Adaptive Control, MFAC)是一种无需精准数学模型即可运行的控制策略,特别适用于动态特性复杂或未知的系统。MFAC方法能够根据当前系统性能调整控制策略,其自适应性使其非常适用于处理许多实际应用中的非线性和不确定性问题。在[4]中,事件触发的MFAC方法被用于在传感器故障和拒绝服务(DoS)攻击情况下保证跟踪性能。在[5]中,MFAC方法成功解决了无人水面艇的航向控制问题,并有效应对了系统的不确定性。此外,在[6]中,MFAC与神经网络的结合实现了更精确的控制效果。

在多智能体系统(Multi-Agent Systems, MASs)领域,控制问题的复杂性显著增加。MAS由多个智能体组成,这些智能体在决策过程中协同工作,并不断变化,表现出复杂的非线性行为[7][8]。一致性控制[9][10]是MAS研究的核心内容,旨在使所有智能体在个体差异和外部不确定性影响下仍能实现共同目标。确保系统协调行为是一项重大挑战,且受制于系统的非线性特性。[11]和[12][13]分别采用了非周期采样数据方法和事件触发方法,实现了对复杂非线性系统的有效一致性控制。

结合强化学习与无模型自适应控制以控制MASs提供了一种新颖而有效的方法。该组合充分利用了两种方法的优势,为离散非线性MAS的一致性控制提供了全面的解决方案。基于上述讨论,本文的主要贡献可总结如下:本文研究了强化学习和无模型自适应控制在MASs中的集成,并重新设计了Q学习策略和演员-评论家(actor-critic)神经网络,使其更适用于MASs。通过重新设计,这些方法能够更精确地逼近最优动态线性化参数。结合MFAC方法,所提出的算法在MASs的一致性控制方面表现出较强的能力。此外,该算法显著增强了对环境变化的适应性和鲁棒性,并加快了收敛速度。通过仿真实验进一步验证了其实用性。

2. 控制器设计

在多智能体系统中,网络结构由 $\mathcal{G}=(\mathcal{V},\mathcal{E},\mathcal{A})$ 表示,其中 $\mathcal{V}$ 表示智能体集合 $\{1,2,\dots,N\}$ 。集合 $\mathcal{E}$ 是 $\mathcal{V}\times\mathcal{V}$ 的子集,描述了智能体之间可能的通信连接。邻接矩阵 $\mathcal{A}$ 的元素为 a_{ij} ,其中 $a_{ij}=1$ 表示智能体 j 到智能体 i 存在直接通信链接,而 $a_{ij}=0$ 表示没有直接通信。此外,当智能体 i 接收来自领导者的数据时,设定 $d_i=1$;否则, $d_i=0$ 。对角矩阵 $\mathcal{D}$ 定义为 $\text{diag}(d_1,d_2,\dots,d_N)$ 。每个智能体 i 的邻居集合表示为 $N_i=\{j\in\mathcal{V}|\{(j,i)\in\mathcal{E}\}\}$ 。智能体 i 的入度(in-degree)定义为 $d_i^{\text{in}}=\sum_{j=1}^N a_{ij}$ 。最终,拉普拉斯矩阵(Laplacian

Matrix) $\mathcal{L}$ 由 $\mathcal{L} = L - \mathcal{A}$ 给出, 其中 L 是 $\text{diag}(d_1^{\text{in}}, d_2^{\text{in}}, \dots, d_N^{\text{in}})$ 。

由 N 个智能体组成的 MAS 的动力学可以表示为:

$$y_i(t+1) = f_i(y_i(t), y_i(t-1), \dots, y_i(t-n_y), u_i(t), u_i(t-1), \dots, u_i(t-n_u)) \quad (1)$$

其中, $i \in \mathcal{V}$, $f_i(\cdot)$ 代表一个未知的非线性函数。变量 $y_i(t) \in \mathbb{R}$ 表示智能体 i 在离散时间 $t \in \mathbb{Z}^+$ 时刻的输出, $u_i(t) \in \mathbb{R}$ 表示对应的输入。常数 n_y 和 n_u 分别表示系统的未知阶数。

根据文献[14], 如果存在理想的非线性控制器, 则系统(1)可以转换为如下形式:

$$u_i(t) = C_i(e_i(t+1), \dots, e_i(t-n_e), u_i(t-1), \dots, u_i(t-n_c)) \quad (2)$$

其中, $C_i(\cdot)$ 是智能体 i 的未知非线性函数, $e_i(t) = y_d(t) - y_i(t)$ 代表智能体 i 的跟踪误差, $y_d(t)$ 是理想系统输出, $n_e, n_c \in \mathbb{Z}^+$ 是未知阶数。

假设系统满足以下条件:

假设 1: 对于任何时间 $t \in \mathbb{Z}^+$, 控制器 $C_i(\cdot)$ 相对于误差输入 $e_i(t+1), e_i(t-1), \dots, e_i(t-n_e+2)$ 和控制输入 $u_i(t), u_i(t-1), \dots, u_i(t-n_c)$ 的偏导数是连续的, 其中 n_e, n_c 是预设的常数。

假设 2: 控制器(2)满足广义 Lipschitz 条件, 即:

$$|\Delta u_i(t)| \leq L \|\zeta_i(t)\| \|\Delta u_i(t)\| \leq L \|\zeta_i(t)\| \|\Delta u_i(t)\| \leq L \|\zeta_i(t)\| \quad (3)$$

其中, $L > 0$ 是 Lipschitz 常数, $\zeta_i(t) = [\Delta e_i(t+1), \dots, \Delta e_i(t-n_e+2), \Delta u_i(t-1), \dots, \Delta u_i(t-n_c)]^T$, $\Delta u_i(t) = u_i(t) - u_i(t-1)$ 。

假设 3: 存在理想的控制器参数, 使得 $e_i(t+1) = 0$ 成立。

假设 4: 虚拟领导者(编号 0)的输出, 即理想轨迹 $y_d(t)$, 至少被一个跟随者(follower)智能体所知, 并且该智能体的信息可以沿着拓扑结构的有向路径传播至所有智能体。

引理 1: 对于满足假设 1~3 的系统智能体 i , 根据文献[14], 理想控制器(2)可以转换为:

$$\Delta u_i(t) = \bar{\zeta}_i^T(t) \psi_i(t) \quad (4)$$

$\bar{\zeta}_i(t) = [-e_i(t), \dots, \Delta e_i(t-n_e+2), \Delta u_i(t-1), \dots, \Delta u_i(t-n_c)]^T$ 和 $\psi_i(t) = [\psi_{i,1}(t), \dots, \psi_{i,n_e+n_c}(t)]^T$ 是有界的。

引理 2: 对于方程(1), 采用等效动态线性化(Equivalent Dynamic Linearization, EDL)技术[15], 可推导出系统(1)的等效数据模型, 并将其表示为部分形式动态线性化(Partial Form Dynamic Linearization, PFDL)形式:

$$y_i(t+1) = y_i(t) + v_i^T(t) \phi_i(t) \quad (5)$$

$$v_i^T(t) = [\Delta u_i(t), \dots, \Delta u_i(t-n_u+1), \Delta y_i(t), \dots, \Delta y_i(t-n_y+1)], \quad \phi_i(t) = [\phi_{i,1}(t), \dots, \phi_{i,n_u+n_y}(t)]^T。$$

随后, 基于数据模型参数 $\phi_i(t)$ 的调整规则, 在每个采样时刻的更新公式如下:

$$\hat{\phi}_i(t) = \hat{\phi}_i(t-1) - \Gamma \hat{e}_i(t) v_i(t-1) \quad (6)$$

其中, Γ 是步长矩阵, $\hat{e}_i(t) = \hat{y}_i(t) - y_i(t)$ 。因此, PFDL 形式(5)可以转换为:

$$\hat{y}_i(t+1) = y_i(t) + v_i^T(t) \hat{\phi}_i(t) \quad (7)$$

3. 强化学习算法设计

在第三部分中, 提出了一种创新的无模型自适应控制方法, 该方法结合了强化学习技术。其主要创新点在于构建了一种基于强化学习算法的 Actor-Critic 神经网络框架的控制器架构。此外, 通过将动态线

性化技术与强化学习过程相结合,可以保证系统的稳定性。

3.1. 价值函数

价值函数 $r_i(t) \in \mathbb{R}$ 由一致性误差 $\xi_i(t)$ 表示, 即:

$$r_i(t) = \alpha |\xi_i(t)| \quad (8)$$

$$\xi_i(t) = \sum_{j \in N_i} a_{ij} (y_j(t) - y_i(t)) + d_i (y_d(t) - y_i(t)) \quad (9)$$

其中, $\alpha > 0$ 是预设参数。

性能指标 $Q_i(t) \in \mathbb{R}$ 表示为:

$$Q_i(t) = \beta^N r_i(t+1) + \beta^{N-1} r_i(t+2) + \dots + \beta^{k+1} r_i(N) + \dots \quad (10)$$

其中, $0 < \beta < 1$ 和 $N > 0$ 为预设常数, 该公式可简化为以下形式:

$$Q_i(t) = \min_{u_i(t)} (\beta Q_i(t-1) - \beta^{N+1} r_i(t)) \quad (11)$$

3.2. Actor-Critic 神经网络

Critic Network 用于估计值函数 $Q_i(t)$, 而 Actor Network 用于逼近控制器参数 $\psi_i(t)$ 。它们的数学表达式如下:

$$\hat{Q}i(t) = \hat{W}i, c^T(t) H_{i,c}(t) \quad (12)$$

$$\hat{\psi}i(t) = \hat{W}i, a^T(t) H_{i,a}(t) \quad (13)$$

$$\hat{\psi}_i(t) = \text{sat}(\underline{\hat{\psi}}, \bar{\hat{\psi}})$$

其中, $\underline{\hat{\psi}}$ 和 $\bar{\hat{\psi}}$ 为预定义常数, $\hat{W}_{i,c}^T$ 和 $\hat{W}_{i,a}^T$ 分别表示评论神经网络和演员神经网络的权重矩阵。演员网络和评论网络的输出激活分别表示为 $H_{i,a}$ 和 $H_{i,c}$ 。

$$h_{i,c,m}(t) = \exp\left(-|\xi_i(t) - c_{c,m}|^2 / \gamma_i^2\right), m \in \{1, \dots, L_c\} \quad (14)$$

$$h_{i,a,n}(t) = \exp\left(-|\xi_i(t) - c_{a,n}|^2 / \gamma_i^2\right), n \in \{1, \dots, L_a\} \quad (15)$$

其中, $h_{i,c,m}$ 和 $h_{i,a,n}$ 分别表示评论神经网络的第 m 层隐藏层激活和演员神经网络的第 n 层隐藏层激活, γ_i 和 c 分别指定隐藏层的宽度和中心。评论神经网络的目标函数表示如下:

$$E_{i,c}(t) = \frac{1}{2} e_{i,c}^2(t) \quad (16)$$

其中, $e_{i,c}(t) = \hat{Q}i(t) - \beta(\hat{Q}i(t-1) - \beta^N r_i(t))$, 权重更新规则如下:

$$\hat{W}i, c(t+1) = \hat{W}i, c(t) - \eta_c H_{i,c}(t) (\hat{Q}i(t) + \beta^{N+1} r_i(t) - \beta \hat{Q}i(t-1)) \quad (17)$$

$$\hat{W}_{i,a}^T(t+1) = \hat{W}_{i,a}^T(t-1) - \frac{\eta_a \bar{\xi}_i(t-1) (e_i(t) - \hat{Q}_{i,a}(t)) \hat{\phi}_{i,1}(t-1) H_{i,a}^T(t-1)}{1 + |\bar{\xi}_i(t-1)|^2 |H_{i,a}(t-1)|^2 |\hat{\phi}_{i,1}(t-1)|^2} \quad (18)$$

其中, $\eta_a, \eta_c \in \mathbb{R}$ 为更新速率, $\hat{Q}_{i,a}(t) = \hat{Q}i(t) \alpha \text{sign}(\xi_i(t)) \beta^{N+1}$ 。随后, 控制信号表示为:

$$\Delta u_i(t) = \bar{\xi}^T i(t) \hat{\psi}i(t) \quad (19)$$

为了更好地说明本文讨论的控制技术, RLMFAC 算法的架构如图 1 所示。

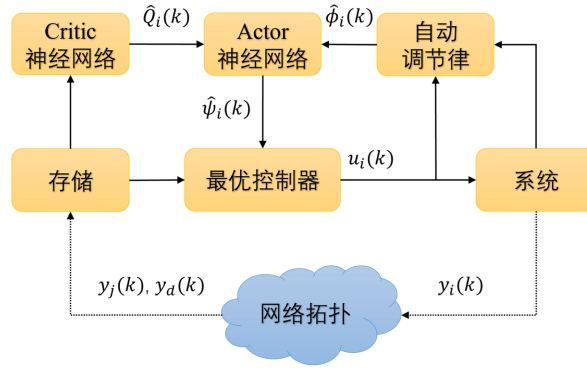


Figure 1. The architecture of RLMAFC algorithm

图 1. RLMFAC 算法结构

备注 1: 由于系统模型不可直接获取, 因此无法直接计算 $\phi_i(t-1)$ 。因此, 在 RLMFAC 算法中, 必须使用数据模型来对 $\phi_i(t-1)$ 进行逼近。

备注 2: 与传统 MFAC 方法不同, 以往方法通常使用输入输出数据逼近 $\partial y_i(t+1)/\partial u_i(t)$ 和 $y_i(t+1)$, 而当前 RLMFAC 策略在控制器设计过程中无需逼近 $y_i(t+1)$, 从而大大拓展了其在实际应用中的适用范围。

4. 仿真验证

本节展示了仿真结果, 以验证所提出算法在多智能体系统(MASs)中的有效性。

对于包含四个智能体及一个领导者(编号 0)的 MAS, 其网络拓扑结构如图 2 所示。系统的动力学方程如下:

$$y_1(t+1) = 1.2u_1(t)(5 + \cos(2y_1(t)u_1(t))) + 0.8\sin(y_1(t))$$

$$y_2(t+1) = 0.8u_2(t)(4 + \cos(3y_2(t)u_2(t))) + \sin(y_2(t))$$

$$y_3(t+1) = 1.1u_3(t)(3 + \cos(4y_3(t)u_3(t))) + 1.2\sin(y_3(t))$$

$$y_4(t+1) = 0.9u_4(t)(6 + \cos(y_4(t)u_4(t))) + 0.6\sin(y_4(t))$$

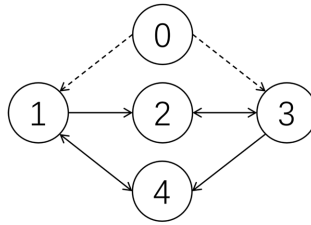


Figure 2. The network topology structure of the MAS

图 2. 多智能体系统网络拓扑结构

其中, $t \in \mathbb{Z}^+$ 表示采样时间。理想轨迹 $y_d(t)$ 定义如下:

$$y_d(t) = 0.55 + 0.25 \left(\sin\left(\frac{2\pi t}{50}\right) + \sin\left(\frac{2\pi t}{100}\right) + \sin\left(\frac{2\pi t}{150}\right) \right)$$

系统的初始条件设定如下:

$\phi_i(1)=[1,1,1,1]^T$, $u_i(1)=0$, $y_i(1)=0$, $\hat{Q}_i(1)=0$, $\hat{W}_{i,c}(1)=0_{27 \times 1}$, $H_{i,c}(1)=0_{27 \times 1}$, $\hat{W}_{i,a}(1)=0_{9 \times 4}$, $H_{i,a}(1)=0_{9 \times 4}$ 。

仿真参数设定如下:

$\Gamma=0.001 \times I_{4 \times 4}$, $L=9$, $N=4$, $n_u=1$, $n_y=3$, $n_c=1$, $n_e=3$, $\eta_a=0.06$, $\eta_c=0.03$, $\alpha=7$, $\beta=0.1$ 。

仿真结果如图3~5所示。图3展示了RLMFAC算法对理想轨迹 $y_d(t)$ 的跟踪性能,图4显示了相应的跟踪误差,而图5展示了控制信号的轨迹。

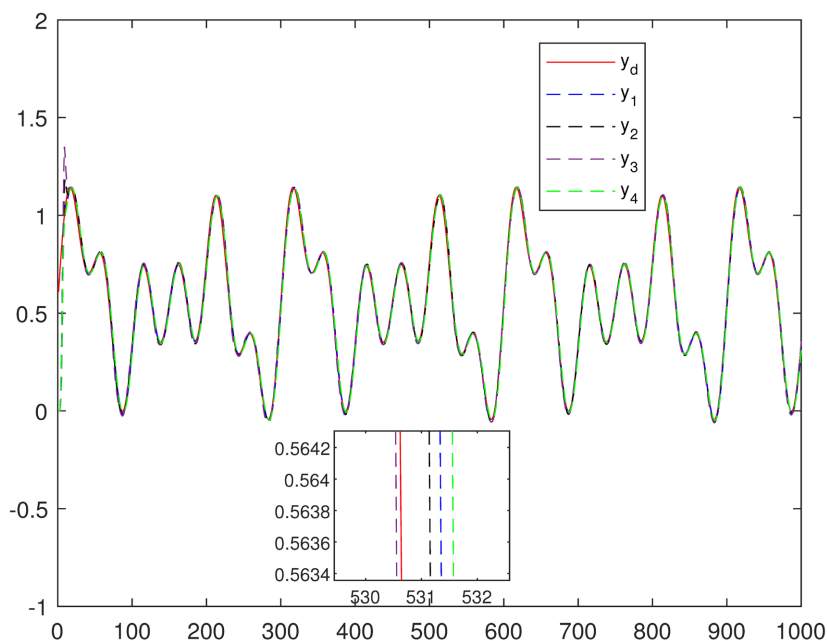


Figure 3. The tracking performances of RLMFAC algorithm

图3. RLMFAC 算法的跟踪效果

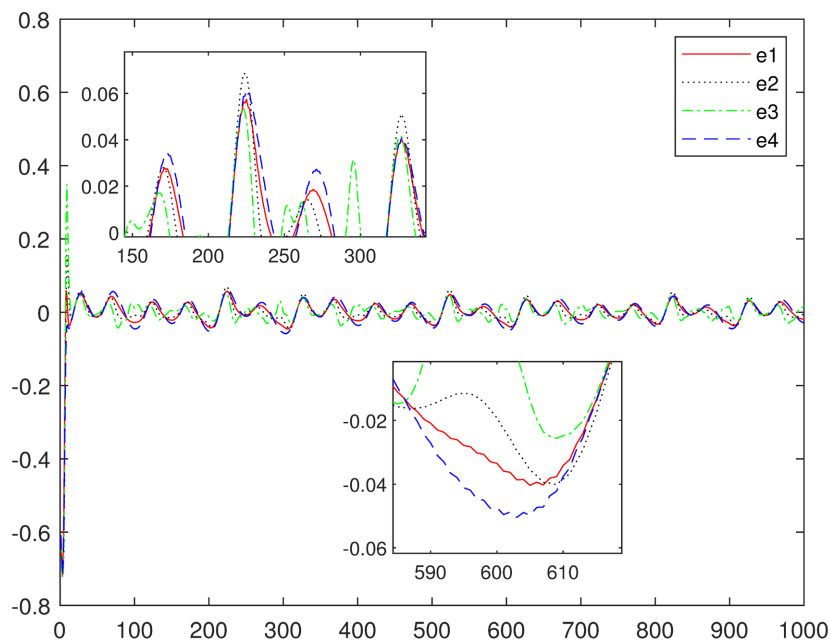


Figure 4. The tracking errors of RLMFAC algorithm

图4. RLMFAC 算法的跟踪误差

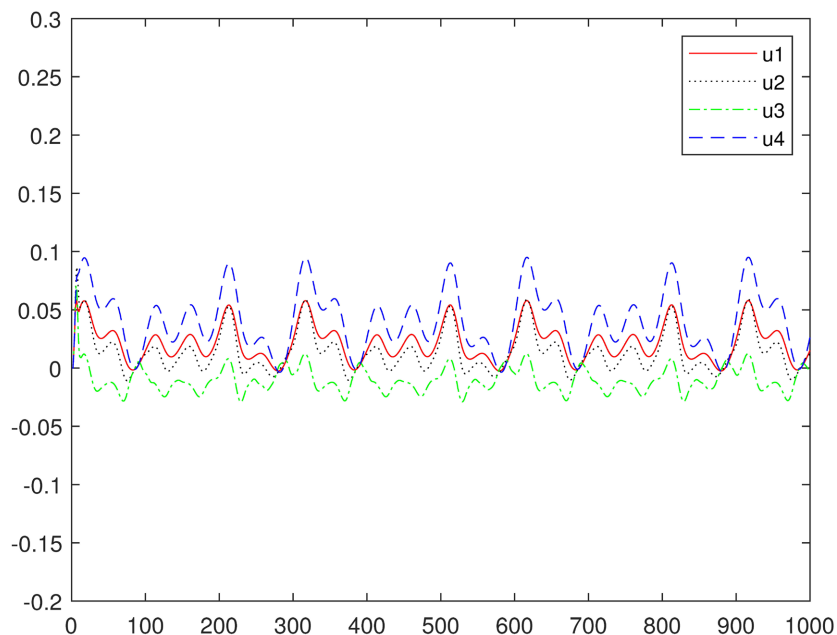


Figure 5. The trajectories of input signals
图 5. 输入信号轨迹

仿真结果表明, RLMFAC 算法在 MASs 中具有良好的性能。因此, 该算法能够有效应用于非线性离散时间系统。

5. 结论

在本研究中, 提出了 RLMFAC 算法, 旨在解决离散时间非线性多智能体系统中的一致性跟踪问题。该方法的核心基于专门设计的值函数策略, 以应对 MASs 的复杂性挑战。仿真结果表明, RLMFAC 算法在跟踪理想轨迹方面具有良好的效果。然而, 该算法在跟踪快速变化的目标时仍存在一定的困难。因此, 使 RLMFAC 算法适应各种实际应用场景仍然是我们持续研究的目标。

参考文献

- [1] Schilling, M., Melnik, A., Ohl, F.W., Ritter, H.J. and Hammer, B. (2021) Decentralized Control and Local Information for Robust and Adaptive Decentralized Deep Reinforcement Learning. *Neural Networks*, **144**, 699-725. <https://doi.org/10.1016/j.neunet.2021.09.017>
- [2] Wang, N., Gao, Y. and Zhang, X. (2021) Data-Driven Performance-Prescribed Reinforcement Learning Control of an Unmanned Surface Vehicle. *IEEE Transactions on Neural Networks and Learning Systems*, **32**, 5456-5467. <https://doi.org/10.1109/tnnls.2021.3056444>
- [3] Zhang, Y., Chu, B. and Shu, Z. (2019) A Preliminary Study on the Relationship between Iterative Learning Control and Reinforcement Learning. *IFAC-PapersOnLine*, **52**, 314-319. <https://doi.org/10.1016/j.ifacol.2019.12.669>
- [4] Yue, B., Su, M., Jin, X. and Che, W. (2022) Event-Triggered MFAC of Nonlinear NCSs against Sensor Faults and Dos Attacks. *IEEE Transactions on Circuits and Systems II: Express Briefs*, **69**, 4409-4413. <https://doi.org/10.1109/tcsii.2022.3178881>
- [5] Liao, Y., Jiang, Q., Du, T. and Jiang, W. (2020) Redefined Output Model-Free Adaptive Control Method and Unmanned Surface Vehicle Heading Control. *IEEE Journal of Oceanic Engineering*, **45**, 714-723. <https://doi.org/10.1109/joe.2019.2896397>
- [6] Wang, X., Karimi, H.R., Shen, M., Liu, D., Li, L. and Shi, J. (2022) Neural Network-Based Event-Triggered Data-Driven Control of Disturbed Nonlinear Systems with Quantized Input. *Neural Networks*, **156**, 152-159. <https://doi.org/10.1016/j.neunet.2022.09.021>

-
- [7] Dorri, A., Kanhere, S.S. and Jurdak, R. (2018) Multi-Agent Systems: A Survey. *IEEE Access*, **6**, 28573-28593. <https://doi.org/10.1109/access.2018.2831228>
 - [8] Chen, F. and Ren, W. (2019) On the Control of Multi-Agent Systems: A Survey. *Foundations and Trends® in Systems and Control*, **6**, 339-499. <https://doi.org/10.1561/26000000019>
 - [9] Olfati-Saber, R., Fax, J.A. and Murray, R.M. (2007) Consensus and Cooperation in Networked Multi-Agent Systems. *Proceedings of the IEEE*, **95**, 215-233. <https://doi.org/10.1109/jproc.2006.887293>
 - [10] Amirkhani, A. and Barshooi, A.H. (2021) Consensus in Multi-Agent Systems: A Review. *Artificial Intelligence Review*, **55**, 3897-3935. <https://doi.org/10.1007/s10462-021-10097-x>
 - [11] Zhao, W., Chen, G., Xie, X., Xia, J. and Park, J.H. (2023) Sampled-Data Exponential Consensus of Multi-Agent Systems with Lipschitz Nonlinearities. *Neural Networks*, **167**, 763-774. <https://doi.org/10.1016/j.neunet.2023.09.003>
 - [12] Ren, H., Liu, R., Cheng, Z., Ma, H. and Li, H. (2024) Data-Driven Event-Triggered Control for Nonlinear Multi-Agent Systems with Uniform Quantization. *IEEE Transactions on Circuits and Systems II: Express Briefs*, **71**, 712-716. <https://doi.org/10.1109/tcsii.2023.3305946>
 - [13] Ma, H., Li, H., Lu, R. and Huang, T. (2020) Adaptive Event-Triggered Control for a Class of Nonlinear Systems with Periodic Disturbances. *Science China Information Sciences*, **63**, Article ID: 150212. <https://doi.org/10.1007/s11432-019-2680-1>
 - [14] Zhu, Y. and Hou, Z. (2014) Data-Driven MFAC for a Class of Discrete-Time Nonlinear Systems with RBFNN. *IEEE Transactions on Neural Networks and Learning Systems*, **25**, 1013-1020. <https://doi.org/10.1109/tnnls.2013.2291792>
 - [15] Hou, Z., Chi, R. and Gao, H. (2017) An Overview of Dynamic-Linearization-Based Data-Driven Control and Applications. *IEEE Transactions on Industrial Electronics*, **64**, 4076-4090. <https://doi.org/10.1109/tie.2016.2636126>