A Parallel Control Method For Zero-Sum Games With Unknown Time-varying System

Qinglai Wei, Zhenhua Zhu, Jie Zhang, Feiyue Wang

Abstract—In this paper, based on ACP approach, a parallel control method is proposed for zero-sum games of unknown time-varying systems. The process of constructing a sequence of artificial systems, implementing the computational experiments and conducting the parallel execution is presented. The artificial systems are constructed to model the real system. Computational experiments adopting adaptive dynamic programming (ADP) are shown to derive control laws for the a sequence of artificial systems. The purpose of the parallel execution step is to derive the control laws for the real system. Finally, simulation experiments are provided to show the effectiveness of the proposed method.

***Index Terms*—Zero-sum games, parallel control, ACP, adaptive dynamic programming (ADP)**

1. INTRODUCTION

Zero-sum game (ZSG) is a classical problem in the game theory. For the linear systems, the algebraic Riccati equation is solved for ZSG. For the nonlinear systems, the Hamilton-Jacobi-Isaacs equation (HIJE) is solved, which is difficult to solve directly. Hence, in [1]–[3], utilizing the successive approximation method, the HJIE is converted to a series of linear differential equations for known systems.

With the research for the ZSG developing, model predictive control (MPC) methods are used to solve the ZSG with known dynamics. In [4], MPC is used for solving the ZSG between two heterogeneous players, which are an unmanned aerial vehicle and an unmanned ground vehicle respectively. Besides, ADP is introduced to solve ZSG. In [5], two structures about ADP are proposed for solving the ZSG of known systems. In [6], the value iteration with history information is proposed for solving the ZSG of known discrete-time systems. For the unknown systems, in [7], by using the neural networks to approximate the systems, the policy iteration is proposed for solving the HJIE. In [8], utilizing the fuzzy models to identify the unknown systems, ADP is used to solve the ZSG.

This work was supported in part by the National Key R&D Program of China under Grants 2021YFE0206100;in part by the National Natural Science Foundation of China under Grant 62073321, 62273036;in part by National Defense Basic Scientific Research Program JCKY2019203C029; in part by the Science and Technology Development Fund, Macau SAR under Grants FDCT-22-009-MISE, 0060/2021/A20015/2020/AMJ;in part by the State Key Lab of Rail Traffic Control & Safety RCS2021K005.(Corresponding authors: Jie Zhang; Fei-Yue Wang.)

The authors are with the State Key Laboratory for Management and Control of Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, and with the School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing 100049, and also with the Institute of Systems Engineering, Macau University of Science and Technology, Macau 999078, China (e-mail: qinglai.wei@ia.ac.cn; zhuzhenhua2020@ia.ac.cn; jie.zhang@ia.ac.cn; feiyue.wang@ia.ac.cn).

In most of above papers, the systems are time-invariant and known. For the unknown systems, the traditional model-based methods utilize neural networks and fuzzy systems to identify the unknown systems. However, for the unknown time-varying systems, the accurate physical models are difficult to obtain for whole time horizon. Inspired by [9], the ACP-based parallel control [10] is introduced to handle the ZSG for unknown time-varying systems.

The ACP approach [10] is originally proposed for the purpose of modeling, analysis and contrl of complex systems. It consists of the following three steps: 1) Construct artificial systems. As the accurate physical model for a nonlinear system is difficult to obtain, the artificial systems are necessary to model the dynamic of the real system. Note that the artificial systems are not meant to be accurate to the real system but for the purpose of analysis. 2) Implement the computational experiments on the artificial systems. Because of the complexity of the real system, it is difficult to carry out computational experiments on the real system. Therefore, it is necessary to find the results of the equivalent problems in the real system from the artificial systems. 3) Implement the parallel execution. Because an artificial system does not approximate the real system, the interation is necessary to control and manage the real system. The control policies for the real system are derived through the parallel execution.

In this paper, an ACP-based parallel control method is proposed to solve the ZSG of the unknown time-varying systems. Using the ACP approach, the method obtains the control laws for the real system through constructing artificial systems, implementing computational experiments and conducting parallel execution.

This paper is organized as follows. In Section II, the problem about ZSG is formulated. In Section III, the details of the proposed method are presented. In Section IV, simulation results and analysis are presented to demonstrate the effectiveness of the proposed method.

2. PROBLEM FORMULATION

Consider a class of unknown time-varying systems:

$$s_{k+1} = \mathscr{F}(s_k, u_{1k}, u_{2k}, k), \quad k = 0, 1, 2, ..., \tag{1}$$

where $\mathscr{F}(\cdot)$ is the system function, $u_{1k} \in \mathbb{R}^m$ and $u_{2k} \in \mathbb{R}^q$ are the control vectors of two players, $s_k \in \mathbb{R}^n$ is the state vector of the system.

The performance index function of the system (1) is defined as follows:

$$\mathcal{J}(s_k, u_{1k}, u_{2k}, k) = \sum_{i=k}^{\infty} U(s_i, u_{1i}, u_{2i}, i) \tag{2}$$

where $U(s_i, u_{1i}, u_{2i}, i)$ is the utility function.

The goal of the ZSG is to obtain the control pair (u_{1k}^*, u_{2k}^*) such that

$$\mathcal{J}^*(s_k, k) = \min_{u_{1k}} \max_{u_{2k}} \mathcal{J}(s_k, u_{1k}, u_{2k}, k), \tag{3}$$

According to the Bellman optimality principle [11], the optimal performance index $\mathcal{J}^*(s_k, k)$ is written as:

$$\mathcal{J}^*(s_k, k) = \min_{u_{1k}} \max_{u_{2k}} (U(s_k, u_{1k}, u_{2k}, k) + \mathcal{J}^*(s_{k+1}, k+1)). \tag{4}$$

It is difficult to solve the equation (4) directly. To overcome the difficulty, we propose a parallel control method to derive the control laws for the real system (1).

3. METHOD

In this section, the proposed method is presented in detail. First, a sequence of artificial systems are constructed to model the system (1). Second, we implement computational experiments via ADP to obtain the control laws for the artificial systems. Third, through the interaction parallel execution step, the suboptimal control laws for the real system are derived.

3.1. Artificial Systems

In many real world scenarios, precise physical models are difficult to obtain. Hence, artificial systems are introduced to model the real system (1). Since the real system (1) is time-varying, a sequence of artificial systems are constructed to model the real system. The artificial systems are written in the following form:

$$\begin{aligned} \hat{s}_{k+1} &= \mathcal{F}(s_k, u_{1k}, u_{2k}, k) \\ &= \begin{cases} \mathcal{F}_0(s_k, u_{1k}, u_{2k}) & k_0 \le k < k_1, \\ ... & \\ \mathcal{F}_\tau(s_k, u_{1k}, u_{2k}) & k_\tau \le k < k_{\tau+1}, \\ ... & \end{cases} \end{aligned} \tag{5}$$

where $\mathcal{F}_\tau(\cdot), \tau = 0, 1, ...$ is an artificial system function in time horizon $[k_\tau, k_{\tau+1})$, $k_0, k_1, ...$ are positive integers.

In this paper, multilayer perceptrons(MLPs) are used to construct the artificial systems, an artificial system function in time horizon $[k_\tau, k_{\tau+1})$ is expressed as:

$$\hat{s}_{k+1}^\tau = \mathcal{F}_\tau(s_k, u_{1k}, u_{2k}) = W_\tau^\top \phi_\tau(s_k, u_{1k}, u_{2k}), \tag{6}$$

where W_τ are the weight matrix of MLPs, $\phi_\tau(\cdot)$ is the activation functions of MLPs.

In the process of constructing artificial systems, the gradient descend algorithm [12] is used to update the weights of MLPs. When the weights of MLPs converge, the real system (1) is modelled by the artificial system. The process of constructing artificial systems is shown in Algorithm 1.

Algorithm 1 The process of constructing artificial systems

Initialization:
1: Collect observational data from the real system
2: Construct MLPs like (6)
3: Give the modeling precision $\epsilon > 0$ and learning rate α
Iteration:
4: Using the gradient descend algorithm to update the weights of MLPs
5: If $\|W_{\tau(i)} - W_{\tau(i-1)}\|_2 > \epsilon$ holds, go to step 4
Return: the weight matrix $W_{\tau(i)}$

3.2. Computational Experiments

Since the real system is complex and unknown, carrying out computational experiments on the real system is difficult. Therefore, computational experiments are implemented on the artificial systems via ADP to evaluate and derive control laws for the artificial systems.

To distinguish from the state of the real system (1), we introduce the artificial system in time horizon $[k_\tau, k_{\tau+1})$:

$$z_{k+1} = \mathcal{F}_\tau(z_k, u_{1k}, u_{2k}), \tag{7}$$

where $z_k \in \mathbb{R}^n$ denotes the state in the artificial systems.

Assume saddle point exists [13], computational experiments via ADP is used to derive the control laws for the artificial system in certian time horizon. According to ADP method [14], the value function and control laws are obtained. For a sequence of artificial systems, there are a sequece of value funxtions and control laws written as:

$$\mathcal{V}(z_k, k) = \begin{cases} \mathscr{V}^0(z_k) & k_0 \le k < k_1, \\ ..., & \\ \mathscr{V}^\tau(z_k) & k_\tau \le k < k_{\tau+1}, \\ .... & \end{cases} \tag{8}$$

$$\mathcal{U}_1(z_k, k) = \begin{cases} \mathscr{U}_1^0(z_k) & k_0 \le k < k_1, \\ ..., & \\ \mathscr{U}_1^\tau(z_k) & k_\tau \le k < k_{\tau+1}, \\ ..., & \end{cases} \tag{9}$$

and

$$\mathcal{U}_2(z_k, k) = \begin{cases} \mathscr{U}_2^0(z_k) & k_0 \le k < k_1, \\ ..., & \\ \mathscr{U}_2^\tau(z_k) & k_\tau \le k < k_{\tau+1}, \\ ..., & \end{cases} \tag{10}$$

In practical applications for computational experiments, MLPs are used to approximate the control laws and the value functions. The structure diagram of the computational experiments is given in Fig. 1. The critic module approximates the value functions, two actor modules approximate the control laws of two players and the artificial system module is the artificial systems constructed in III-A.

For $k \in [k_\tau, k_{\tau+1})$, $\tau = 0, 1, 2, ...$, the approximation functions of the control laws and the value functions are expressed as:

$$\hat{\mathscr{V}}^\tau(z_k) = W_c^{\top(\tau)} \phi_c(z_k), \tag{11}$$

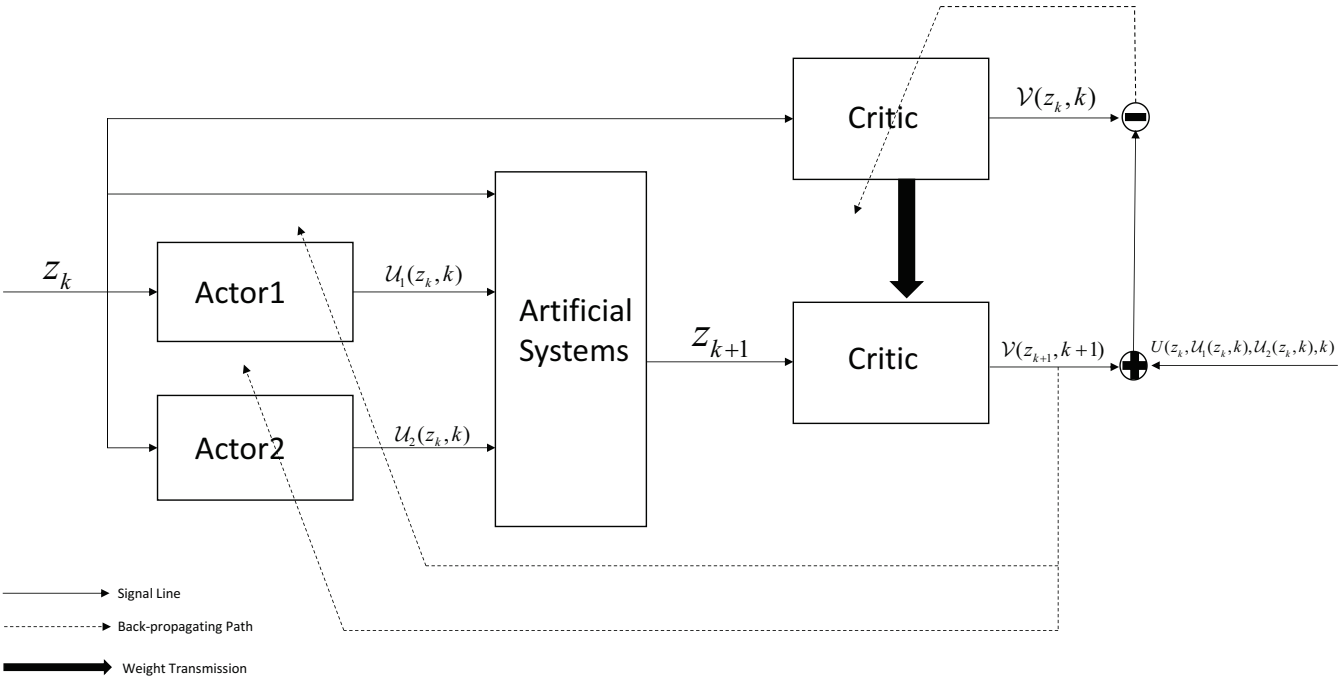


Fig. 1. The structure diagram of the computational experiments for a sequence of artificial systems

$$\hat{\mathscr{U}}_1^{\tau}(z_k) = W_{u1}^{\top(\tau)}\phi_{u1}(z_k), \tag{12}$$

and

$$\hat{\mathscr{U}}_2^{\tau}(z_k) = W_{u2}^{\top(\tau)}\phi_{u2}(z_k), \tag{13}$$

where W_c^{τ},W_{u1}^{τ} and W_{u2}^{τ} are the weight matrix of MLPs, $\phi_c(\cdot)$,$\phi_{u1}(\cdot)$ and $\phi_{u2}(\cdot)$ are the activation functions of MLPs.

In the process of iteration for computational experiments, the weights of MLPs are tuned by the the gradient descend algorithm. When the weights of MLPs converge, the control laws for the artificial systems are derived. The details about the process of computational experiments are presented in Algorithm 2.

Algorithm 2 Computational Experiments Algorithm

Initialization:

1: Give an initial value function $\mathscr{V}_0^{\tau}(z_k) = 0$
2: Give the computation precision $\epsilon > 0$
3: Construct three MLPs for approximating the value function and control laws

Iteration:

4: Compute the iterative value functions and control laws by using ADP
5: Using the gradient descend algorithm to update the weights of three MLPs
6: If $\|W_{c(i+1)} - W_{c(i)}\|_2 > \epsilon$ holds, go to step 4

Return: $W_{u(i)}^{\tau}, W_{w(i)}^{\tau}, W_{c(i)}^{\tau}$

3.3. Parallel Execution

After constructing the artificial systems and implementing the computational experiments, the control laws for the artificial systems are derived. As we mention above, an artificial system in certain time horizon does not approximate the real system, hence, feedback and interaction are necessary to obtain the control laws for the real system.

In parallel execution step, the control laws derived from computational experiments are implemented not only on the artificial systems but also on the real system, then the difference about performance of the artificial systems and the real system is evaluated. If the diffirence is not large, the control laws derived from computational experiments continue to be applied to the real system. Otherwise, it is necessary to update the artificial systems and conduct the computational experiments again.

Through constructing the artificial systems, implementing the computational experiments and conducting the parallel execution, the control laws for the real system are obtained until the state of the real system is stable.

We present a detailed process of the parallel execution in Algorithm 3

4. Simulation Study

In this section, to show the effectiveness of the proposed method, we choose an example obtained from [9] with some modifications, the dynamics of the system is as follows.

Algorithm 3 Parallel Execution Algorithm

Initialization:

1: Give the computation precision ϵ_1 and ϵ_2
2: Let $\tau = 0$, $k = 0$, $k_\tau = k$
3: Give initial state s_0 and z_0, let z_0 = s_0

Iteration:

4: Implement Algorithm 1 to construct the artificial system (6) under ϵ_1
5: Obtain the weights of two action modules W_u^τ, W_w^τ and the weights of the critic module W_c^τ by using Algorihtm 2
6: Implement the control laws on the artificial system (6) and the real system (1) and evaluate the difference:
if the difference is larger than a threshold
go to step 4
else
if $||s_k|| > \epsilon_2$, go to step 6

$$\begin{pmatrix} s_{1(k+1)} \\ s_{2(k+1)} \end{pmatrix} = \begin{pmatrix} s_{1k} + 0.1s_{2k} \\ -0.1s_{1k} + 1.1s_{2k} - 0.1s_{1k}^2 s_{2k} \end{pmatrix} + B_k u_{1k} + C_k u_{2k} \tag{14}$$

where

$$B_k = \begin{bmatrix} -0.5 + 0.2\sin(k) & 0 \\ 0 & -0.5 + 0.2\sin(k) \end{bmatrix} \tag{15}$$

and

$$C_k = \begin{bmatrix} 0.1 + 0.05\cos(k) & 0 \\ 0 & 0.1 + 0.05\cos(k) \end{bmatrix}. \tag{16}$$

Let utility function $U(s_i, u_{1i}, u_{2i}) = s_i^\top Q s_i + u_{1i}^\top R u_{1i} - \gamma^2 u_{2i}^\top u_{2i}$, where $Q = I_1 \in \mathbb{R}^{2\times 2}$, $R = I_2 \in \mathbb{R}^{2\times 2}$.

In the Artificial Systems, multilayer perceptrons (MLPs) are used model the real system. For the MLPs, The structure is 6–8–2 and the activation functions of the output layer and the hidden layer are purelin functions and tansig functions respectively. The bias of the output layer and the hidden layer are zero. The training process of weights for MLPs is shown in Algorithm 1 with model precision $\epsilon = 0.001$ and the learning rate $\alpha = 0.02$.

Next, In the Computational Experiments, three MLPs, which are two action networks and critic networks, are used as basis functions to approximate the control laws of two players and the value function. The structures of two action networks and critic networks are chosen as 2–8–2, 2–8–2, and 2–8–1. For all three MLPs,the activation functions of the output layer and the hidden layer are purelin functions and tansig functions respectively. The bias of the output layer and the hidden layer are zero. Implement computational experiments by Algorithm 2 with the computation precision $\epsilon = 0.001$ and the learning rate $\alpha = 0.02$ to evaluate and derive the control laws for multiple artificial systems.

Then, in the parallel execution, we implement Algorithm 3 to derive the control laws for the real system. After implementing the proposed method, the trajectory of the states for the real system is shown in Fig. 2. According to the Fig. 2, the states of system converge to the equilibrium, showing the effectiveness of the proposed method.

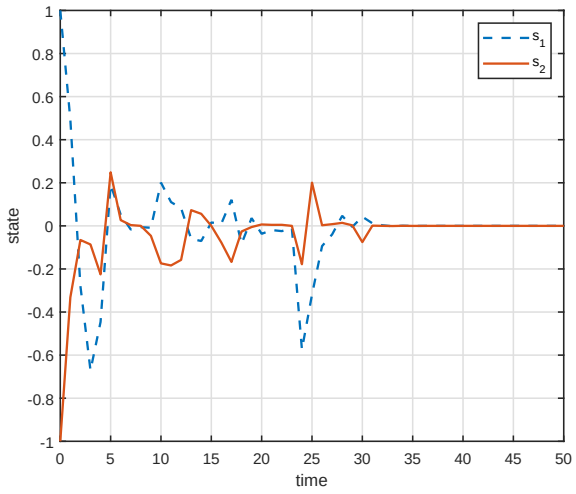


Fig. 2. The trajectory of the states for the real system

REFERENCES

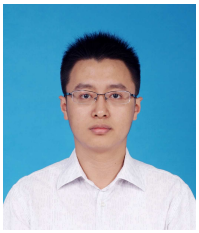
[1] A. VANDERSCHAFT, "L2-gain analysis of nonlinear-systems and nonlinear state feedback-h-infinity control," *IEEE Transactions on Automatic Control*, vol. 37, no. 6, pp. 770–784, Jun 1992.
[2] M. Abu-Khalaf, F. L. Lewis, and J. Huang, "Policy iterations on the hamilton-jacobi-isaacs equation for h-infinity state feedback control with input saturation," *IEEE Transactions on Automatic Control*, vol. 51, no. 12, pp. 1989–1995, Dec 2006.
[3] S. Mehraeen, T. Dierks, S. Jagannathan, and M. L. Crow, "Zero-sum two-player game theoretic formulation of affine nonlinear discrete-time systems using neural networks," *IEEE Transactions on Cybernetics*, vol. 43, no. 6, pp. 1641–1655, Dec 2013.
[4] G. Tzannetos, P. Marantos, and K. J. Kyriakopoulos, "A competitive differential game between an unmanned aerial and a ground vehicle using model predictive control," in *the 24th Mediterranean Conference on Control and Automation*, Athens, Greece, Jun. 2016, pp. 21–24.
[5] A. Al-Tamimi, M. Abu-Khalaf, and F. L. Lewis, "Adaptive critic designs for discrete-time zero-sum games with application to h-infinity control," *IEEE Transactions on Systems Man and Cybernetics Part B-Cybernetics*, vol. 37, no. 1, pp. 240–247, Feb 2007.
[6] M. Zhao, D. Wang, M. Ha, and J. Qiao, "Evolving and incremental value iteration schemes for nonlinear discrete-time zero-sum games," *IEEE Transactions on Cybernetics*, early access, doi: 10.1109/TCYB.2022.3198078.
[7] H. Zhang, C. Qin, B. Jiang, and Y. Luo, "Online adaptive policy learning algorithm for h-infinity state feedback control of unknown affine nonlinear discrete-time systems," *IEEE Transactions on Cybernetics*, vol. 44, no. 12, pp. 2706–2718, Dec 2014.
[8] H. Zhang, H. Su, K. Zhang, and Y. Luo, "Event-triggered adaptive dynamic programming for non-zero-sum games of unknown nonlinear systems via generalized fuzzy hyperbolic models," *IEEE Transactions on Fuzzy Systems*, vol. 27, no. 11, pp. 2202–2214, Nov 2019.
[9] Q. Wei, L. Wang, J. Lu, and F.-Y. Wang, "Discrete-time self-learning parallel control," *IEEE Transactions on Systems Man and Cybernetics-Systems*, vol. 52, no. 1, pp. 192–204, Jan 2022.
[10] F.-Y. Wang, "Parallel control and management for intelligent transportation systems: Concepts, architectures, and applications," *IEEE Transactions on Intelligent Transportation Systems*, vol. 11, no. 3, pp. 630–638, Sep. 2010.
[11] R. Bellman and R. E. Kalaba, *Dynamic programming and modern control theory*. New York: Citeseer, 1965.
[12] J. Nocedal and S. J. Wright, *Numerical Optimization*. New York: Springer-Verlag, 1999.
[13] T. Steven, *Game theory: an introduction*. Princeton: Princeton university press, 2013.

[14] Q. Wei, D. Liu, Q. Lin, and R. Song, "Adaptive dynamic programming for discrete-time zero-sum games," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 4, pp. 957–969, Apr 2018.

Qinglai Wei (M'11-SM'20) received the B.S. degree in Automation, and the Ph.D. degree in control theory and control engineering, from the Northeastern University, Shenyang, China, in 2002 and 2009, respectively. From 2009–2011, he was a postdoctoral fellow with The State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing, China. He is currently a professor of the institute and the associate director of the laboratory. He has authored four books, and published over 80 international journal papers. His research interests include adaptive dynamic programming, neural-networks-based control, optimal control, nonlinear systems and their industrial applications.

He is the Secretary of IEEE Computational Intelligence Society (CIS) Beijing Chapter since 2015. He was guest editors for several international journals. He was a recipient of IEEE/CAA Journal of Automatica Sinica Best Paper Award, IEEE System, Man, and Cybernetics Society Andrew P. Sage Best Transactions Paper Award, IEEE Transactions on Neural Networks and Learning Systems Outstanding Paper Award, the Outstanding Paper Award of Acta Automatica Sinica, IEEE 6th Data Driven Control and Learning Systems Conference (DDCLS2017) Best Paper Award, and Zhang Siying Outstanding Paper Award of Chinese Control and Decision Conference (CCDC). He was a recipient of Shuang-Chuang Talents in Jiangsu Province, China, Young Researcher Award of Asia Pacific Neural Network Society (APNNS), Young Scientst Award and Yang Jiachi Tech Award of Chinese Association of Automation (CAA). He is a Board of Governors (BOG) member of the International Neural Network Society (INNS) and a council member of CAA.

Zhenhua Zhu received the bachelors degree in mechanical engineering from the Beihang University (BUAA), Beijing, China, in 2019. He is currently pursuing the master degree with The State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing, China, and the School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing, China. His current research interests include adaptive dynamic programming and optimal control.



Jie Zhang (M'16) received the B.Sc. degree in information and computing science from Tsinghua University, Beijing, China, in 2005, and the Ph.D. degree in technology of computer application from the University of Chinese Academy of Sciences, Beijing, in 2015. He has been an Associate Professor with the State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing, since 2016. His current research interests include parallel control, mechanism design, optimal control, and multiagent reinforcement learning.

Fei-Yue Wang (S'87-M'89-SM'94-F'03) received the Ph. D. degree in computer and systems engineering from the Rensselaer Polytechnic Institute, Troy, NY, USA, in 1990. He joined The University of Arizona, Tucson, AZ, USA, in 1990, and became a Professor and the Director of the Robotics and Automation Laboratory and the Program in Advanced Research for Complex Systems. In 1999, he founded the Intelligent Control and Systems Engineering Center, Institute of Automation, Chinese Academy of Sciences (CAS), Beijing, China, under the support of the Outstanding Chinese Talents Program from the State Planning Council, and in 2002, was appointed as the Director of the Key Laboratory of Complex Systems and Intelligence Science, CAS, and Vice President of Institute of Automation, CAS, in 2006. In 2011, he became the State Specially Appointed Expert and the Founding Director of the State Key Laboratory for Management and Control of Complex Systems. He has been the Chief Judge of Intelligent Vehicles Future Challenge since 2009 and Director of China Intelligent Vehicles Proving Center, Changshu since 2015. He is currently the Director of Intels International Collaborative Research Institute on Parallel Driving with CAS and Tsinghua University, Beijing, China.

His research interests include methods and applications for parallel intelligence, social computing, and knowledge automation. He is a Fellow of INCOSE, IFAC, ASME, and AAAS. In 2007, he was the recipient of the National Prize in Natural Sciences of China, numerous best papers awards from IEEE Transactions, and became an Outstanding Scientist of ACM for his work in intelligent control and social computing. He was the recipient of the IEEE ITS Outstanding Application and Research awards in 2009, 2011, and 2015, respectively, IEEE SMC Norbert Wiener Award in 2014, and became the IFAC Pavel J. Nowacki Distinguished Lecturer in 2021.

Since 1997, he has been the General or Program Chair of more than 30 IEEE, INFORMS, IFAC, ACM, and ASME conferences. He was the President of the IEEE ITS Society from 2005 to 2007, IEEE Council of RFID from 2019 to 2021, Chinese Association for Science and Technology, USA, in 2005, American Zhu Kezhen Education Foundation from 2007 to 2008, Vice President of the ACM China Council from 2010 to 2011, Vice President and the Secretary General of the Chinese Association of Automation from 2008 to 2018, Vice President of IEEE Systems, Man, and Cybernetics Society from 2019 to 2021. He was the Founding Editor-in-Chief (EiC) of the International Journal of Intelligent Control and Systems from 1995 to 2000, IEEE ITS Magazine from 2006 to 2007, IEEE/CAA Journal of Automatica Sinica from 2014 to 2017, Chinas Journal of Command and Control from 2015 to 2021, and Chinas Journal of Intelligent Science and Technology from 2019 to 2021. He was the EiC of the IEEE Intelligent Systems from 2009 to 2012, IEEE Transactions on Intelligent Transportation Systems from 2009 to 2016, IEEE Transactions on Computational Social Systems from 2017 to 2020. He is currently the President of CAAs Supervision Council and new EiC of the IEEE Transactions on Intelligent Vehicles.