IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received October 30, 2018, accepted November 19, 2018, date of current version May 6, 2019.

Digital Object Identifier 10.1109/ACCESS.2018.2883997

A Study of Aero-Engine Control Method Based on Deep Reinforcement Learning

QIANGANG ZHENG, CHONGWEN JIN, ZHONGZHI HU, AND HAIBO ZHANG
Nanjing University of Aeronautics and Astronautics, Jiangsu Province Key Laboratory of Aerospace Power System, Nanjing 210016, China

Corresponding author: Qiangang Zheng (zhqg@nuaa.edu.cn)

This was supported in part by the National Natural Science Foundation of China under Grant 51576096, in part by Q. Lan and the 333 Project, in part by the Research Funds for Central Universities under Grant NF2018003, in part by Six Talents Peak Project of Jiangsu Province.

ABSTRACT A novel aero-engine control method based on deep reinforcement learning (DRL) is proposed to improve the engine response ability. The Q-learning that is model free and can be performed online is adopted. For improving the learning capacity of DRL, the online sliding window deep neural network (OL-SW-DNN) is proposed and adopted to estimate the action value function. The OL-SW-DNN selects the nearest point data with certain length as training data and is insensitivity to the noise. Finally, the comparison simulations of the proposed method with the proportion–integration–differentiation (PID) that is the most commonly used as an engine controller algorithm in industry are conducted to verify the validity of the proposed method. The results show that, compared with the PID, the acceleration time of the proposed method decreased by 1.525 s under the premise of satisfying all engine limits.

INDEX TERMS Aero-engine control method, response ability, deep reinforcement learning, on line, deep neural network.

I. INTRODUCTION

When aircraft performs in some situations, such as landing, take-off, aircraft overshoot and some emergency situations, the engine is required quickly response from one working state to another working state [1]–[3]. The less consumed time during the transient process means the better engine response performance. Therefore, how to improve engine response ability is an important index during designing engine control system. The most popular engine control method is Proportional-Integral (PI), which has simple structure, strong robustness and convenience for adjustment [4]. However, the aero-engine is a multivariable, strong nonlinear, strong coupling relationship, time delayed and poor working conditions controlled plant [5]. Therefore, the traditional control method - PI is hard to get the best response ability [6].

In recent years, the control method based on Deep Reinforcement Learning (DRL)that is model free, adopts Deep Learning (DL) technology, and will obtain best response performance as learning time goes on, which has arouse the research interests of many researchers [7], [8]. Schuitema *et al.* [9] proposed a controller based on Reinforcement Learning (RL) algorithm for a passive dynamic walking robot. Wang *et al.* [10] adopted Q-Learning algorithm to choose the PD controller parameters for biped robot walking on uneven surfaces. Ziqiang *et al.* [11] designed Q-Learning controller based on Back-Propagation (BP) Neural Networks to control 2D biped robot. Mnih *et al.* [12] presented deep reinforcement learning based on convolutional neural network and successfully learned control policies directly from high-dimensional sensory input. Lillicrap *et al.* [13] proposed an actor-critic, model-free algorithm based on the deterministic policy gradient that can operate over continuous action spaces. Oh *et al.* [14] introduced a new set of RL tasks in Minecraft (a flexible 3D world) and used these tasks to systematically compare and contrast existing DRL architectures with new memory-based DRL architectures. The above works and other works about the application of DRL [15], [16] got great control effect. However, there are seldom involved in the application of DRL to engine control.

Therefore, a new aero-engine control method based on DRL is proposed here to enhance engine response ability. This paper is organized as follows. In Section II, the control structures of the engine control system are given. In Section III, the engine controller based on DRL is designed firstly, and the On Line Sliding Window Deep Neural Network (OL-SW-DNN) is proposed. In Section IV, to verify the validity of the proposed method, the comparison simulations with the PID are carried out. In Section V, some conclusions are given.

II. THE CONTROL STRUCTURE OF AERO-ENGINE CONTROL SYSTEM

Figure 1 and Figure 2 give the control structures of the traditional aero-engine control system and the proposed one based on DRL respectively. They mainly consist of controller method block, acceleration limits block, deceleration limits block, MIN blocks and MAX blocks. The acceleration limits block calculates the fuel flow while engine satisfies surge margin limits of fan and compressor, rotor speed limits of fan and compressor, the temperature limit of turbine inlet temperature and the other physical limits. The target of deceleration limits block calculates the fuel flow to overcome flameout phenomenon. The MIN and MAX blocks get the maximum or minimum value of the inputs. The controller block calculates the fuel flow to get desired thrust. The traditional engine control system always adopts Proportional-Integral (PI) or Proportion-Integration-Differentiation (PID) as control method. The reinforcement learning will make engine more and more intelligent. For improving the learning ability of RL, the Deep Neural Network (DNN) is adopted to estimate action-value function of RL. Therefore, a new controller based on DRL is proposed here to improve engine response ability.

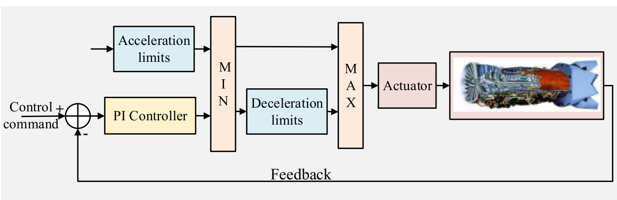

FIGURE 1. The control structure of traditional aero-engine control system.

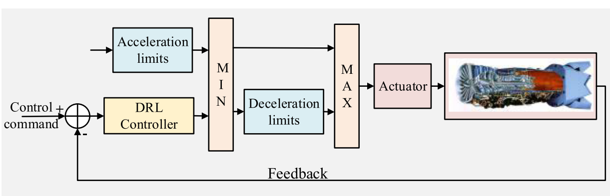

FIGURE 2. The control structure of DRL controller for aero-engine.

III. THE ENGINE CONTROLLER BASED ON DEEP REINFORCEMENT LEARNING

A. THE PRINCIPLE OF Q-LEARNING

Q-Learning is model free and selects the next action based on Q table or DNN without estimating the control object. The transient process of aero-engine is a strong nonlinear process. Therefore, the Q-Learning is selected to update action value function $Q(s, a)$, where s is engine state, a is an action or control input of engine. At j episode, with probability ε selecting a random action a_j otherwise selecting $a_j = \max_a Q(s_j, a)$. Then the updating rule of $Q(s, a)$ can be described as:

$$Q_j(s,a) = \begin{cases} (1-\alpha)Q_j(s_j,a_j) + \alpha[r_j + \gamma \\ \max\limits_{a_{j+1}} Q(s_{j+1},a_{j+1})] \quad \text{for non-terminal } s_t \\ r_j \quad \text{for terminal } s_t \end{cases} \tag{1}$$

where α is the learning rate, r is the reward, γ is the discount rate, s_t is terminal engine state.

In order to make engine fast switch from a working state to another one, the reward r_k is calculated as following:

$$\begin{aligned} r_j = {} & [\mathbf{r}(j) - \hat{\mathbf{r}}(j)]^T Q[\mathbf{r}(j) - \hat{\mathbf{r}}(j)] \\ & + [\mathbf{u}(j) - \mathbf{u}(j-1)]^T R[\mathbf{u}(j) - \mathbf{u}(j-1)] \end{aligned} \tag{2}$$

where $\hat{\mathbf{r}}$ is measured or estimated value of control objective, such as rotor speed, engine pressure ratio, $\mathbf{u}$ is control variable vector (action a, fuel flow in aero-engine control), Q and R are positive definite and symmetric. A larger value of Q will make much faster response of the engine.

B. THE ON-LINE SLIDING WINDOW DEEP NEURAL NETWORK

The DNN is a non-linear mapping for the multi-input multi-output system [18], [19] and can be described as follow:

$$\mathbf{y} = f_{DNN}(\mathbf{x}) \tag{3}$$

where $\mathbf{x}$ is an input vector and $\mathbf{y}$ is an output vector. In order to keep the engine dynamic characteristics and improve the estimation precision of the DNN model, the input consists of current and past fuel flow W_{fb}, past the speed of fan rotor N_f, the speed of compressor rotor N_c, the surge margin of fan S_{mf}, the surge margin of compressor S_{mc}, and the inlet temperature of high pressure turbine T_{41}. And the output is action value function $Q(s, a)$. As shown in Eq.(4), the input and output of DNN is:

$$\begin{cases} \boldsymbol{x} = [W_{fb}(j), W_{fb}(j-1), \cdots, W_{fb}(j-m_1); \\ \quad N_f(j-1), N_f(j-2), \cdots, N_f(j-m_2); \\ \quad N_c(j-1), N_c(j-2), \cdots, N_c(j-m_3); \\ \quad S_{mf}(j-1), S_{mf}(j-2), \cdots, S_{mf}(j-m_4); \\ \quad S_{mc}(j-1), S_{mc}(j-2), \cdots, S_{mc}(j-m_5); \\ \quad T_{41}(j-1), T_{41}(j-2), \cdots, T_{41}(j-m_6)] \\ y = Q(s,a) \end{cases} \tag{4}$$

Because the engine can be often simplified to an object with two degrees of freedom, $m_1, m_2, \cdots, m_6$ are all set to 2 in this paper.

The structure of DNN is shown in Figure 3. The DNN has deeper hidden layer than traditional neural network. The increase of the hidden layers of DNN will improve the fitting capacity of DNN. The each hidden layer of DNN is defined as:

$$\mathbf{a}^{l+1} = \mathbf{W}^l \mathbf{h}^l + \mathbf{b}^l \tag{5}$$

$$\mathbf{h}^{l+1} = \sigma(\mathbf{a}^{l+1}) \tag{6}$$

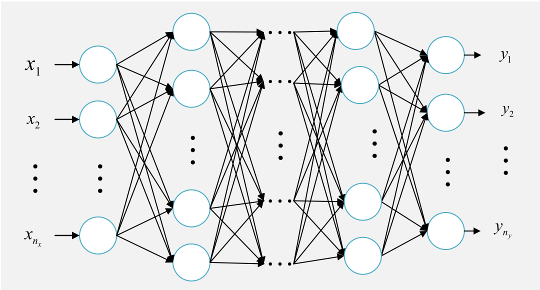

FIGURE 3. The structure of DNN.

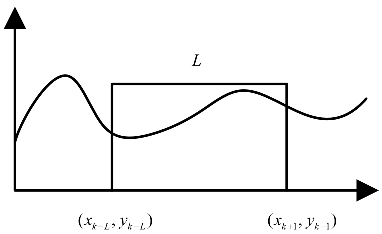

FIGURE 4. The principle of sliding window.

where $\mathbf{W}^l$ is weight matrix, $\mathbf{b}^l$ is offset vector, $\boldsymbol{\sigma}$ is activation function, $\mathbf{h}^l$ (for $l > 0$) is the output of the l-th hidden layer, $l = 1, 2, \cdots, n_l$, n_l is the number of layers. Set $\mathbf{h}_i^0 = \mathbf{x}_i$, $i = 1, 2, \cdots, N$, N is the size of the training set.

The traditional on-line deep neural network always only selects one data point to computes gradient at each iteration, which has better real time performance. Nevertheless, only selecting one training point is sensitivity to the noise and might not be a best direction choice. Therefore, in order to improve the robustness of NN, as shown in Figure 3, the OL-SW-DNN is proposed and applied to controller design. The OL-SW-DNN selects the nearest point data of length L as at each iteration of the training data. The loss function of OL-SW-DNN is described as:

$$J_1(\mathbf{W}, \mathbf{b}; \mathbf{x}, \mathbf{y}) = \min_{\mathbf{W},\mathbf{b}} \sum_{j=k-L}^{k+1} \frac{1}{2} \left\| f_{DNN}(\mathbf{x}_j) - \mathbf{y}_j \right\|^2 \tag{7}$$

At each iteration, $\mathbf{W}$, $\mathbf{b}$ is updated as follows:

$$W_{ij}^l \leftarrow W_{ij}^l + \eta \nabla W_{ij}^l \tag{8}$$

$$b_i^l \leftarrow b_i^l + \eta \nabla b_i^l \tag{9}$$

where η is the learning rate. As shown in Figure 5, the back-propagation, which quickly solves the gradient of network parameters, is applied to calculate ∇W_{ij}^l and ∇b_i^l.

The gradients of $\mathbf{W}$, $\mathbf{b}$ are calculated as:

$$\frac{\partial J(\mathbf{W}, \mathbf{b}; \mathbf{x}, \mathbf{y})}{\partial W_{ij}^l} = \mathbf{h}_j^l \boldsymbol{\delta}_j^{l+1} \tag{10}$$

$$\frac{\partial J(\mathbf{W}, \mathbf{b}; \mathbf{x}, \mathbf{y})}{\partial b_j^l} = \boldsymbol{\delta}_j^{l+1} \tag{11}$$

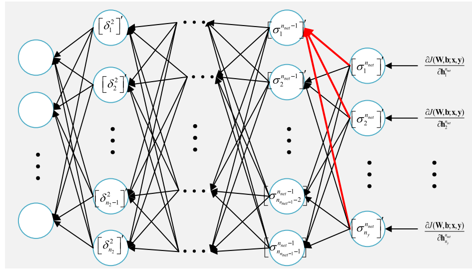

FIGURE 5. The principle of Back-propagation algorithm.

where $\boldsymbol{\delta}^l$ is:

$$\boldsymbol{\delta}^l = \left[\mathbf{W}^l\right]^T \boldsymbol{\delta}^{l+1} \otimes \left[\boldsymbol{\sigma}^l\right]' \tag{12}$$

where $l = n_{net}, n_{net} - 2, \cdots, 2$, let $\otimes$ is Hadamard product, $\mathbf{x} \otimes \mathbf{y} = [x_1 y_1, x_2 y_2, \cdots x_n y_n]^T$.

Suppose $\bar{\boldsymbol{\delta}}^{n_{net}}$ is

$$\begin{aligned} \boldsymbol{\delta}^{n_{net}} &= \frac{\partial J(\boldsymbol{W}, \boldsymbol{b}; \boldsymbol{x}, \boldsymbol{y})}{\partial \boldsymbol{a}^{n_{net}}} = \frac{\partial J(\boldsymbol{W}, \boldsymbol{b}; \boldsymbol{x}, \boldsymbol{y})}{\partial \boldsymbol{h}^{n_{net}}} \otimes \frac{\partial \boldsymbol{h}^{n_{net}}}{\partial \boldsymbol{a}^{n_{net}}} \\ &= \frac{\partial J(\boldsymbol{W}, \boldsymbol{b}; \boldsymbol{x}, \boldsymbol{y})}{\partial \boldsymbol{h}^{n_{net}}} \otimes \left[\boldsymbol{\sigma}^{n_{net}}\right]' \end{aligned} \tag{13}$$

where n_{net} is the number of network layer.

IV. SIMULATION AND ANALYSIS OF THE DRL CONTROLLER

In order to verify the effectiveness of the proposed method, the comparison simulations of the PID and the proposed method are conducted here. The engine acceleration process is the strongest nonlinear process among the transient process. Therefore, the acceleration process is selected as simulation process of these two method. The engine operation condition of these two method is the standard atmospheric state at height $H = 0km$, Mach number Ma=0. The starting point is the engine steady working state when power level angle PLA=20°. The ending point is the engine steady working state when PLA=70°. The simulation results of the proposed method and the PID are as shown in Figure 6. The parameters of engine in the Figure have been normalized. Through debugging, the structure of OL-SW-DNN is chosen as [13,15,12,10,10,1]. The decay parameter $\lambda = 10^{-5}$. Learning rate $\alpha = 0.00002$. Momentum factor $\eta = 0.6$. $L = 25$.

As shown in Figure 6(a), the times for thrust increases to 95% thrust of the design point in proposed method and PID are 3.7 seconds and 5.225 seconds respectively. It can be easily inferred that the proposed method has much faster than PID control method, and the acceleration time nearly decreased by 1.525 seconds. The main reason is that the DRL will learn experience from the history and make engine more and more intelligent. Moreover, the OL-SW-DNN has strong fitting capacity and let the proposed method has stronger learning ability.

As shown in Figure 6(g), during the acceleration process of the engine, the working points move along the surge

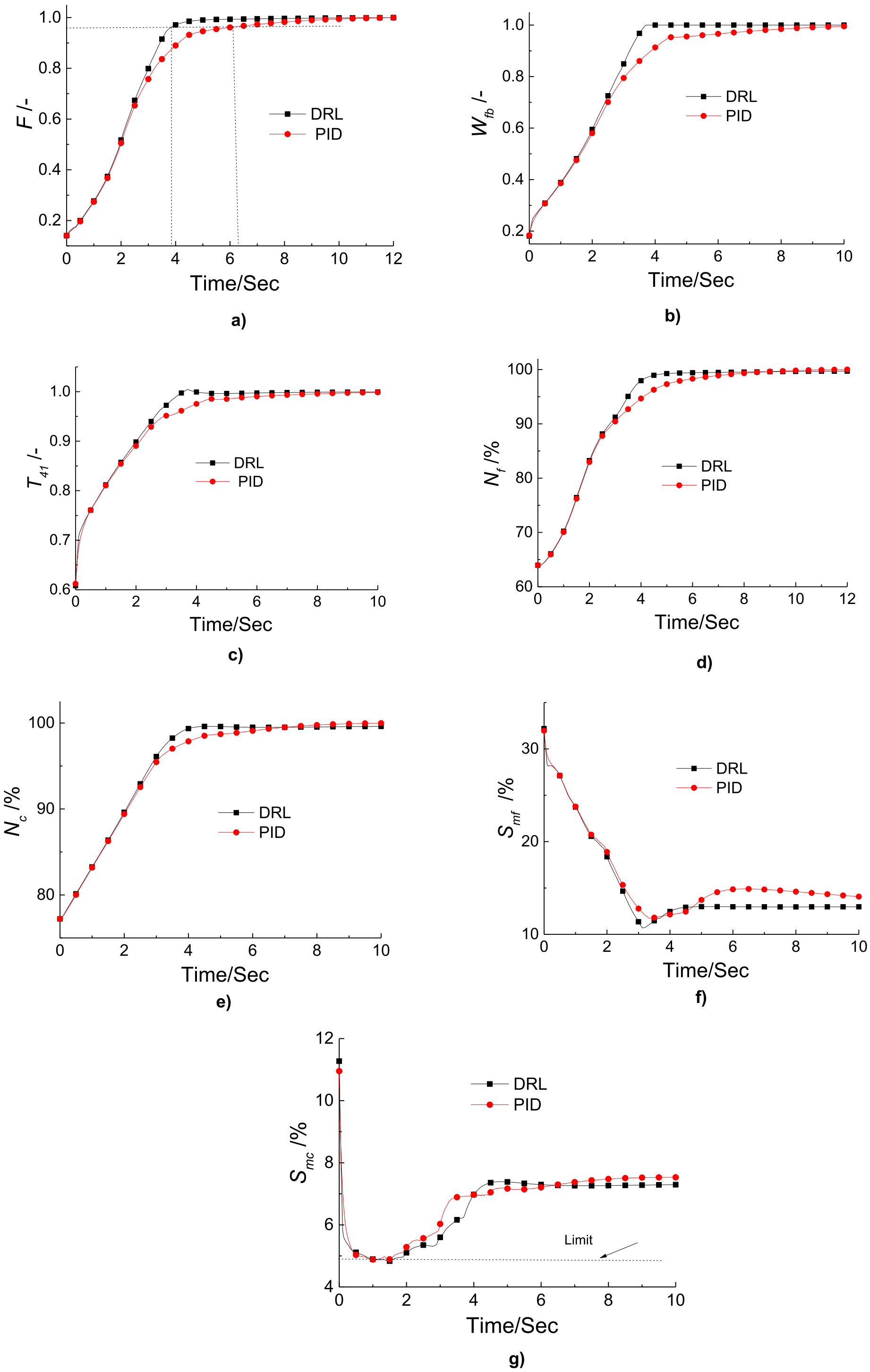

FIGURE 6. The engine acceleration simulation of DRL and PID. a) The response of F. b) The control input of W_{fb}. c) The response of T_{41}. d) The response of N_f. e) The response of N_c. f) The response of S_{mf}. g) The response of S_{mc}.

limit, which usually regarded as the fastest route in engine theory. As shown in Figure 6(c)~6(f), when the proposed method is applied during the acceleration process of the engine, the engine did not reach over-temperature, over-speed or occur-surge. This demonstrates that the proposed control method has high control precision and response speed.

V. CONCLUSIONS

A new engine control method based on deep reinforcement learning is proposed in this paper. For improving the control effect of the proposed control method, the on-line sliding window deep neural network is applied to fitting the action value function. The engine acceleration simulations of DRL and PID show that the proposed control method has much better response ability. Compared with the PID, the acceleration time of the proposed method decreased by 1.525 seconds, while the limits of engine are all satisfied during the acceleration process.

APPENDIX

NOMENCLATURE

Symbol	Explanation
H	Height
Ma	Mach number
PLA	Power level angle
W_{fb}	Fuel flow
F	Engine thrust
T_{41}	High pressure turbine inlet temperature
N_f	Fan rotor speed
N_c	Compressor rotor speed
S_{mf}	Fan surge margin
S_{mc}	Compressor surge marge

REFERENCES

[1] C. A. Skira and M. Agnello, "Control systems for the next century's fighter engines," *J. Eng. Gas Turbines Power*, vol. 114, no. 4, pp. 749–754, 1992.

[2] Q. Zheng, L. Miao, H. Zhang, and Z. Ye, "On-board real-time optimization control for turbofan engine thrust under flight emergency condition," *Proc. Inst. Mech. Eng., I, J. Syst. Control Eng.*, vol. 231, no. 7, pp. 554–566, 2017.

[3] Q. Zheng, H. Zhang, L. Miao, and F. Sun, "On-board real-time optimization control for turbo-fan engine life extending," *Int. J. Turbo Jet-Engines*, vol. 34, no. 4, pp. 321–332, 2017.

[4] H. Richter, *Advanced Control of Turbofan Engines*. New York, NY, USA: Springer Science & Business Media, 2011.

[5] L. C. Jaw and J. D. Mattingly, *Aircraft Engine Controls: Design, System Analysis, and Health Monitoring*. Reston, VA, USA: American Institute of Aeronautics and Astronautics, 2009.

[6] A. Kwiatkowski, H. Werner, J. P. Blath, A. Ali, and M. Schultalbers, "Linear parameter varying PID controller design for charge control of a spark-ignited engine," *Control Eng. Pract.*, vol. 17, no. 11, pp. 1307–1317, 2009.

[7] C. Blundell *et al.* (2016). "Model-free episodic control." [Online]. Available: https://arxiv.org/abs/1606.04460

[8] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: MIT Press, 1998.

[9] E. Schuitema, D. G. E. Hobbelen, P. P. Jonker, M. Wisse, and J. G. D. Karssen, "Using a controller based on reinforcement learning for a passive dynamic walking robot," in *Proc. 5th IEEE-RAS Int. Conf. Humanoid Robots*, Dec. 2005, pp. 232–237.

[10] S. Wang, J. Braaksma, R. Babuska, and D. Hobbelen, "Reinforcement learning control for biped robot walking on uneven surfaces," in *Proc. IEEE Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2006, pp. 4173–4178.

[11] P. Ziqiang, P. Gang, and Y. Ling, "Learning biped locomotion based on Q-learning and neural networks," in *Advances in Automation and Robotics*, vol. 1. Berlin, Germany: Springer, 2011, pp. 313–321.

[12] V. Mnih *et al.* (2013). "Playing Atari with deep reinforcement learning." [Online]. Available: https://arxiv.org/abs/1312.5602

[13] T. P. Lillicrap *et al.* (2015). "Continuous control with deep reinforcement learning." [Online]. Available: https://arxiv.org/abs/1509.02971

[14] J. Oh, V. Chockalingam, S. Singh, and H. Lee. (2016). "Control of memory, active perception, and action in minecraft." [Online]. Available: https://arxiv.org/abs/1605.09128

[15] V. François-Lavet, R. Fonteneau, and D. Ernst. (2015). "How to discount deep reinforcement learning: Towards new dynamic strategies." [Online]. Available: https://arxiv.org/abs/1512.02011

[16] J. N. Foerster, Y. M. Assael, N. de Freitas, and S. Whiteson. (2016). "Learning to communicate to solve riddles with deep distributed recurrent Q-networks." [Online]. Available: https://arxiv.org/abs/1602.02672

[17] V. Mnih *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[18] Q. Zheng, H. Zhang, Y. Li, and Z. Hu, "Aero-engine on-board dynamic adaptive MGD neural network model within a large flight envelope," *IEEE Access*, vol. 6, pp. 45755–45761, 2018.

[19] I. Goodfellow *et al.*, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.

QIANGANG ZHENG received the Ph.D. degree from the Nanjing University of Aeronautics and Astronautics (NUAA) in 2018. He is currently a Teacher with the College of Energy and Power Engineering, NUAA. His research interests include modeling, fault diagnosis, optimization control, and model predictive control of aero-engines.

CHONGWEN JIN is currently pursuing the M.D. degree in aerospace propulsion theory and engineering with the College of Energy and Power Engineering, Nanjing University of Aeronautics and Astronautics. His research interests include aircraft engine modeling, switching control, and predictive control.

ZHONGZHI HU received the B.S. degree in automatic control from the Beijing Institute of Technology, China, and the M.S. and Ph.D. degrees in electrical engineering from The University of British Columbia, Canada. He has almost 30 years of extensive experience in modeling and simulation, controls and PHM for gas turbine, wind turbine, and other industry applications, as a Researcher and an Engineer, including 15 years of research and development and engineering positions working for general electric. He is currently a Professor with the Nanjing University of Aeronautics and Astronautics, Nanjing, China. He has published over 40 refereed conference and journal papers. He holds five U.S. patents. His research interests are in gas turbine modeling, simulation, controls, and PHM.

HAIBO ZHANG received the Ph.D. degree from the Nanjing University of Aeronautics and Astronautics (NUAA) in 2005. He is currently a Professor with the College of Energy and Power Engineering, NUAA. His main research interests include modeling, fault diagnosis, and the control of aero-engines.