ITC 4/52
Information Technology and Control
Vol. 52 / No. 4 / 2023
pp. 1010-1024
DOI 10.5755/j01.itc.52.4.33125

Design of Intelligent Controller for Aero-engine Based on TD3 Algorithm

Received 2023/01/02 | Accepted after revision 2023/10/26

HOW TO CITE: Zhu, J., Tang, W., Dong, J. (2023). Design of Intelligent Controller for Aero-engine Based on TD3 Algorithm. *Information Technology and Control*, 52(4), 1010-1024. https://doi.org/10.5755/j01.itc.52.4.33125

Design of Intelligent Controller for Aero-engine Based on TD3 Algorithm

Jianming Zhu, Wei Tang, Jianhua Dong
School of Automation, Northwestern Polytechnical University, Xi'an 710129, China;
e-mails: zhujm@mail.nwpu.edu.cn; tangwei@nwpu.edu.cn; djh@mail.nwpu.edu.cn

Corresponding author: tangwei@nwpu.edu.cn

Recently, higher structure complicacy and performance requirements of the aero-engine have brought higher demands on its control system. With the rapid development of artificial intelligence technology, the intelligent controller with self-learning ability will be able to make a great difference. In the paper, we propose an aero-engine intelligent controller design method based on twin delayed deep deterministic policy gradient (TD3) algorithm. The design method allows the intelligent controller to interact autonomously with the aero-engine system to acquire the optimal control sequence. The JT9D turbofan engine is used to introduce the controller design workflow proposed in the paper. First, the problem of aero-engine control is described as a Markov decision process for deep reinforcement learning (DRL) algorithms. Second, a complete intelligent controller design process is constructed by reasonably designing the network structures and reward function. Finally, the comparison simulations are carried out to verify the superior performance of the controller design methods. The simulation results indicate that low-pressure turbine speed has no overshoot, and the settling time does not exceed 0.88s during the engine acceleration process. In the deceleration process, the overshoot of the low-pressure turbine speed is limited to 0.74% and the settling time does not exceed about 0.6s. The results prove that the TD3 controller outperforms deep deterministic policy gradient (DDPG) and the proportional-integral-derivative (PID) in the speed tracking control.

KEYWORDS: TD3, intelligent control, turbofan engine, deep reinforcement learning, neural network.

1. Introduction

Aero-engines are highly complex multivariable control systems, which are characterized by nonlinearity, time-varying, and sensitivity to external environmental changes. The control method of the aircraft propulsion system is mainly based on PID control [19] with a simple and robust control struc-

ture. However, as aero-engine systems continue to evolve, they are expected to exhibit even more pronounced control characteristics, making the use of advanced control methods essential for improved performance. In recent years, scholars have proposed many improved control methods for aero-engine, such as Linear Quadratic Regulator (LQR) [14], H_∞ [30]. Though, most of these methods are for linear models. Then scholars also proposed nonlinear control methods such as gain scheduling control [4, 17], model predictive control [16], and sliding mode variable structure control [11] on this basis. All of these control methods require the establishment of a comparatively precise system model and design of accurate controller with the foundation. In practice, the complex aerothermodynamic processes make it challenging to develop an accurate model of the aircraft propulsion system, which in turn makes it difficult to control the aero-engine. Accordingly, model-free control algorithms on the basis of artificial intelligence provide a new solution.

Reinforcement learning (RL) [22] is a model-free control algorithm, and originates from dynamic programming and optimal control theory. Its fundamental approach is to perceive the states of the environment and select appropriate actions through trial-and-error learning, without relying on an explicit model of the system. The algorithm explores the optimal policy through repeated interaction with the environment, learning from the feedback received in the form of rewards or penalties. Q-learning [2] is a common reinforcement learning method that discretizes the action and state space to solve problems using tables. It has been widely studied and improved upon, leading to the development of SARSA [9], Deep Q-Network (DQN) [15], and Double-DQN [6]. The DQN algorithm is considered a pioneering work in the field of deep reinforcement learning as it combines reinforcement learning with deep learning techniques to address control problems based on visual perception. This algorithm has achieved fruitful results in discrete behavior decision-making. However, its effectiveness in high-dimensional continuous action spaces requires further investigation. In 2015, the researchers presented DDPG algorithm [10] to solve the dimensional explosion issue caused by the discretization of continuous space. The algorithm employs a network that outputs a certain value, which is provided by the deterministic policy gradient. As a result, the problem of continuous action and state space can be solved by the DDPG algorithm. Nevertheless, problems such as high estimation of value network in DDPG algorithm still exist. Considering these problems, TD3 algorithm on account of deep double Q-learning is then proposed [3]. Up to now, reinforcement learning has been studied in diverse areas including robotics [12, 23], spacecraft guidance [1, 7], flight control [24] and automatic text summarization [21]. However, there has been little research on the application of RL to aero-engine control. And these studies have focused on the DQN and DDPG algorithms.

This study proposes a model-free intelligent controller design method based on TD3 algorithm, which directly maps the state information of aero-engine operation to the control signals of the engine. The main contributions of the work are listed as follows:

1. The TD3 algorithm is applied to the intelligent control of turbofan engines, and a detailed design flow is given. This work provides a solid basis for future practical applications of reinforcement learning on turbofan engines.
2. This paper designs the neural network with gradient threshold limitations and a reward function centered on speed control within safety boundaries. The design ensures the practicality of the intelligent controller, and simulation results demonstrate the effectiveness of the trained control strategy.
3. The comparison simulation is conducted between the controller based on the TD3 algorithm and the DDPG-based controller, PID controller for the aero-engine accelerating and decelerating control tasks. The TD3-based controller exhibits superior performance compared to the other two controllers.

The rest of the paper is organized as follows: Section 2 shows some related works. Section 3 describes the aero-engine control problem as a Markov decision process and introduces the principle of related algorithms. Section 4 describes in detail the design process of the intelligent controller based on TD3 algorithm. Section 5 illustrates the simulation and analysis. Last, Section 6 concludes the content of the article.

2. Related Work

The aero-engine is a system with great uncertainty and needs to meet the demands of high performance, low fuel consumption rate and low noise during operation. The absence of a suitable aero-engine control system can lead to severe problems, such as compressor surge and speed stalling. The control methods of aero-engine can be classified into model-based control and model-free control. In general, integrating mathematical models of aero-engines with controller design is considered a model-based control algorithm. The latter approach does not require the establishment of a precise model of the controlled target and allows for direct controller design.

2.1. Model-based Control

Modern control theory is generally considered as model-based control algorithms. The approach is now extensively applied in the control of aero-engine. In applications of model-based control, the establishment or identification of the aero-engine model is the first issue to be considered. When the model is available, we would be able to design the controller.

Modeling an aero-engine with physical mechanisms heavily depends on the accuracy of its parameters. The model built by the identification method is demanded to reveal the dynamic properties of the aero-engine over a wide operating range. Therefore, the model-based control with the aero-engine is typically designed according to the following steps. Firstly, the operating envelope of the aero-engine is partitioned into several sub-regions based on specific points. In each of the regions, the mathematical model of the aero-engine can be constructed in diverse forms, e.g., small perturbation state space model [28, 29] and finite impulse response model [26]. Afterwards, controllers based on different theories are designed using these models. Finally, aero-engine control within the full envelope is implemented by means of gain scheduling. For example, the method of the LQG/LTR control was applied to the turbine speed control of aero-engine [8]. The simulations show the method can effectively reduce the turbine speed overshoot. Haiquan et al. [5] employed two degrees-of-freedom H_∞ loop-shaping method to realize the aero-engine control with improved robustness. Model predictive control (MPC) is also a typical model-based control approach that has gained wide attention and investigation in recent years. The model of aero-engine constitutes one of the most fundamental components of this control algorithm. This model is primarily designed to predict the dynamic output of engines based on historical data and future inputs. Using this information, real-time rolling optimization and feedback correction can be carried out. It is one of the most effective methods for dealing with constraint system control problems in engines.

However, the operation of an aero-engine involves a multitude of complex aerothermodynamic processes. Whether the modeling is based on mechanism or identification by data, it is difficult to avoid the large errors of model. Consequently, the performance of the aero-engine controller developed using a model-based approach will inevitably degrade to some extent. Based on that, the model-free control method would be a promising choice.

2.2. Model-free Control

Reinforcement learning (RL) is a model-free control algorithm. As we mentioned earlier, the design of model-free controllers rarely depends on any mathematical model of the controlled object. Instead, they rely entirely on the data obtained through interaction with the controlled systems. In highly unknown and uncertain nonlinear systems, these features of reinforcement learning algorithms offer prospective solutions to develop optimal controllers.

In the framework of reinforcement learning, more and more control problems are solved. However, the application of RL in aero-engine control is still in its early stages. Zhang et al. [25] suggested an optimal controller of aero-engine on steady-state operating point based on DRL algorithm. Zheng et al. [27] proposed that the deep Q-learning method can be utilized to enhance the aero-engine acceleration performance. Since the DQN algorithm cannot be applied directly to the continuous action space, this research suggests to find the action with the largest action value function at each step. This method requires evaluation over the entire action space, leading to the difficulty in real-time control. Miao et al. [13] proposed a transient controller design approach based on DDPG algorithm. The simulation results show that this approach can control the acceleration and deceleration process of turbofan engines and maintain the system

performance. Qian et al. [18] proposed the mathematical model of the turbofan engine which is in the polynomial state space form. And they employed the DDPG algorithm to design the intelligent controller. The simulation results indicate that the controller brings about a great performance improvement of the aero-engine.

Most of the studies mentioned above have focused on addressing aero-engine control problem with mathematical models and deep deterministic policy gradient algorithms. However, the DDPG algorithm suffers from the issues of sensitive hyperparameters and Q value overestimation, which can make it difficult to converge to the optimal policy. To address these issues, an aero-engine intelligent controller design method based on TD3 algorithm is proposed in this paper.

3. The Principle of Deep Reinforcement Learning

3.1. Markov Decision Process (MDP) Model of Aero-engine Control

This paper takes the JT9D as the object of study, which is a high bypass ratio dual-rotor turbofan engine. Its component-level model is developed based on the toolbox for the modeling and analysis of thermodynamic systems (T-MATS).

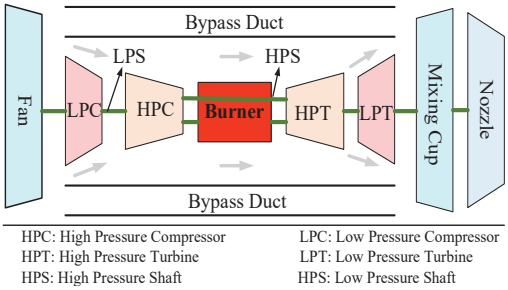
The following listed the operating conditions simulated by this model:

$$W = 674.22\,pps, h_t = 130BTU/(lbm*R)$$

$$T_t = 448.46degR,\ P_t = 5.528\,psia,\ Pamb = 3.626\ psia,$$

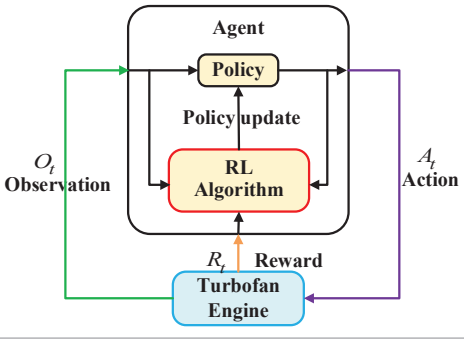
where W is the gas path flow of inlet, h_t is the total enthalpy, T_t is the inlet air temperature, P_t is the inlet air pressure,$Pamb$ is the ambient pressure. As shown in Figure 1, it mostly consists of following components: aircraft inlet, fan, low-pressure compressor (LPC), high-pressure compressor (HPC), burner, high-pressure turbine (HPT), low-pressure turbine (LPT), and tail nozzle. The purpose of the aero-engine controller is to obtain the most efficient engine performance and the excellent thrust response within a safe range. The primary task of a turbofan engine controller is the tracking control of the turbine speed. In real-world applications, the thrust of the turbofan engine cannot be directly measured. Therefore, the low-pressure turbine speed $n_L(t)$, which is proportional to the thrust, is commonly used as a measurement for control. The controller adjusts the fuel-air ratio $FAR(t)$, based on the error signal between the desired speed and the output speed of the turbofan turbine, which is regulated. The allowable range of the air-fuel ratio is [0.01,0.05].

Figure 1
The JT9D turbofan engine model



In reinforcement learning, the MDP model consists of the tuple $\langle S, A, P, r, \gamma \rangle$, where S is the set of states, A is the set of actions, γ is the discount factor, and $r(s, a)$ is the reward function. The value of the reward at a given point in time depends on the state and the action taken. The state transfer function, represented by $P(s'|s, a)$, determines the likelihood of reaching the next state s'. Reinforcement learning is a suitable approach for solving sequential decision problems. However, it is important to note that the system being solved must satisfy the assumption that the next state of the system is only dependent on the current state. Figure 2 illustrates the basic elements of reinforcement learning and the corresponding interaction process. At each

Figure 2
The schematic diagram of reinforcement learning



time step, the agent receives observations of the external environment and selects an action based on them. After the agent performs an action, the environment transitions to a new state and returns a reward to the agent. The reward at the current moment is called the immediate reward and is defined as r_t. The cumulative reward obtained by the agent from the environment is defined as $R_t = \sum_t^{t+N_t} \gamma r$, where the discount factor $\gamma (0 \leq \gamma \leq 1)$ represents the effect of the reward value at future moments on the current cumulative reward.

Aero-engine control system is a typical closed-loop feedback system. In the control loop, the control signal is calculated by the controller through the error between the reference signal and the system response. On this basis, we describe this control task in the MDP model as follows:

State *S*:The primary task of a turbofan engine controller is the tracking control of the turbine speed. Accordingly, the low-pressure turbine speed $n_L(t)$ and the error signale $e(t)$ between the given speed and the output speed of the turbofan turbine are the most necessary observation signals. The state space observed by the agent in this paper is defined as:

$$S_t = [n_L(t), e(t), \int e(t)dt]. \tag{1}$$

Action *A*: Agent's action is usually defined as a quantity related to the controller parameter or input. In this paper, the output of the intelligent controller is the action. The action space is therefore as follows:

$$A_t = [FAR(t)]. \tag{2}$$

Reward r_t: The reward function is a scalar feedback signal provided by the environment, indicating the agent's gain in selecting a specific action at a particular time step. Designing an appropriate reward function is crucial, as it requires prior knowledge of aero-engine control and is a significant indicator of control performance. The details of the reward function are described in Section 4.

Transition probability function *P*: The transition probability function can be replaced by physical and thermodynamic response of aero-engine in practice.

3.2. Basics of DDPG Algorithm

In reinforcement learning, the objective of the agent is to maximize the cumulative reward value by optimizing its own policy. The agent observes the current state *s* of the environment and selects the corresponding action *a* based on the learned policy μ. The action taken by the agent modifies the state of the environment, and the environment provides the agent with a reward and a new state *s'*. The state-action value function $Q^*(s, a)$ is updated by iterating through the Bellman equation.

$$Q^*(s,a) = E_{s' \sim S}[r + \gamma \max_{a'} Q^*(s',a')], \tag{3}$$

where *r* is reward function, the γ is the discount factor,E[·] is expected function, and the prime notation denotes the quantities at the next discrete time. The expected function is introduced due to the uncertainty of the state at the next moment.

In practice, a function is usually used to approximate the $Q^*(s, a)$, which means $Q^*(s, a) \approx Q^*(s, a|\theta)$. The parameter θ can be calculated by minimizing the loss function.

This loss function is defined as:

$$L_i(\theta_i) = E_{s' \sim S}[(y_i - Q(s,a \mid \theta_i))^2] \tag{4}$$

$$y_i = E_{s' \sim S}[r + \gamma \max Q(s',a' \mid \theta_{i-1})]. \tag{5}$$

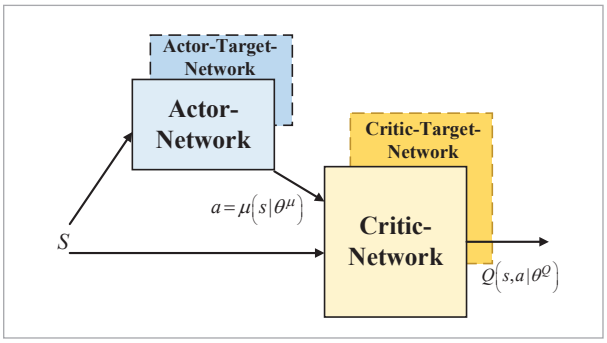
When the network parameters θ_{i-1} are constant, $L_i(\theta_i)$ is optimized. The variables of the loss function are differentiated to obtain the gradient equation:

$$\nabla_{\theta_i} L_i(\theta_i) = E[(r + \gamma \max_{a'} Q(s',a' \mid \theta_{i-1}) - Q(s,a \mid \theta_i))\nabla_{\theta_i} Q(s,a \mid \theta_i)] \tag{6}$$

The optimal policy is then derived by solving the Bellman equation. In the DQN algorithm, a critical technique is setting up the experience replay buffer. This buffer stores transition samples (s, a, r, s') generated during the agent's interaction with the environment. The use of it helps to reduce the correlation between consecutive samples and stabilizes the learning process.

The DDPG algorithm is an enhanced algorithm on a basis of the Actor-Critic network structure and introduces the target network. As shown in Figure 3, the target networks are created by replicating the original Actor and Critic neural networks. Therefore, they have the same network structure and initial parameters. By separating the functions of parameter

Figure 3
Network structure of DDPG algorithm



updating, strategy selection, and value function calculation, the learning process becomes more stable. The Actor neural network is responsible for iteratively updating the parameters and selecting the action based on the current state. The Actor target network selects the optimal action a' based on the next state sampled from the experience replay buffer which can be expressed as $a = \mu(s|\theta^{\mu})$. The Critic network is primarily responsible for computing the current Q value function $Q(s, a|\theta^{Q})$ and updating its parameters. The Critic target network is primarily utilized to calculate the target Q value which is involved in TD-error.

The equation for calculating the target Q value is:

$$y_i = r + \gamma Q\left(s', \mu(s' \mid \theta^{\mu'}) \mid \theta^{Q'}\right), \tag{7}$$

where y_i is target Q value, $\mu(s'|\theta^{\mu'})$ is the output of the Actor target network, $Q(s', \mu(s'|\theta^{\mu'})|\theta^{Q'})$ is the output of the Critic target network.

The Critic neural network is trained by minimizing TD-error:

$$L = \frac{1}{N}\sum_{i=1}^{N}(y_i - Q(s,a \mid \theta^{Q}))^2, \tag{8}$$

where $Q(s, a)|\theta^{Q})$ is the output of the Critic neural network, L is the square mean value of TD-error.

The Actor network then maps state through the policy to the specified action and updates the current policy. It is updated in order to obtain the optimal policy that maximizes the cumulative reward. Thus, the objective function for neural network training is defined as the expectation of cumulative rewards. The formula is as follows:

$$J \approx E\left[\sum_{i=t}^{T}\gamma^{(i-t)} r\left(s_i, a_i\right)\right], \tag{9}$$

where J is the objective function.

In 2014, Sliver et al. have proven the validity of the following formula [20]. The formula shows the relationship between the gradient of the objective function and the gradient of $Q(s, a|\theta^{Q})$.

$$\nabla_{\theta^{\mu}} J \approx E\left[\nabla_{\theta^{\mu}} Q\left(s, a \mid \theta^{Q}\right) \mid s = s_i, a = \mu\left(s_i \mid \theta^{\mu}\right)\right], \tag{10}$$

where $\nabla_{\theta^{\mu}} J$ is the gradient of the objective function, $\nabla_{\theta^{\mu}} Q(s, a| \theta^{Q})$ is the gradient of $Q(s, a| \theta^{Q})$.

During the training process, the Actor neural network is updated by means of batch samples. The final update formula is:

$$\nabla_{\theta^{\mu}} J \approx \frac{1}{N}\sum_{i=1}^{N}\nabla_a Q(s,a \mid \theta^{Q}) \mid s = s_i, a = \mu\left(s_i\right) \bullet \nabla_{\theta^{\mu}} \mu\left(s \mid \theta^{\mu}\right) \mid s_i \tag{11}$$

where N denotes the size of the batch samples.

3.3. Basics of TD3 Algorithm

The DDPG algorithm has practical limitations, including overestimation issues, which affect its performance in real-world applications. In response, the TD3 algorithm incorporates three key techniques to improve the training process: clipped double Q-learning, delayed policy updates, and target policy smoothing.

In Equation (3), the operation of taking the maximum value when calculating the state-action value function may lead to the problem of overestimating Q.Clipped double Q-learning is a technique used in the TD3 algorithm to mitigate the problem. This technique involves the use of two Q-functions instead of one. Both Q-functions are updated using the same target, but during the calculation of the target Q value, the smaller Q value is chosen to avoid overestimation. The equation is written as follows:

$$y = r + \gamma \min_{i=1,2} Q(s', a' \mid \theta_i^{Q}), \tag{12}$$

where y is target Q value in TD3 algorithm.

Based on the target policy μ, target policy smoothing derives the target action by adding a perturbation fac-

tor to each dimension of the action, ensuring that the target action takes a value that satisfies the condition $a_{low} \le a \le a_{high}$.The second key skill can be expressed as:

$$a' = clip(\mu(s') + o, a_{low}, a_{high}) \tag{13}$$

$$o \sim clip(N(0,\sigma), -c, c), \tag{14}$$

where a_{low} and a_{high} respectively indicate the maximum and minimum values of the action a, $N(0, \sigma)$ represents Gaussian noise, and $clip(x, -y, y)$ means each element of x is clipped to the effective range $[-y, y]$.The key skill of the TD3 algorithm is to address the issue of potential overestimation in the Q function approximator by quickly correcting any incorrect peaks through target policy smoothing.

The delayed update policy means that the parameters of the Critic network are updated more frequently than the parameters of the Actor network and the target network. By delaying the update of the Actor network and target network, the Critic network has more time to learn and provide more accurate Q value estimates.

The target network is updated by soft-update method. The update equation is:

$$\begin{aligned} \theta_i^{Q'} &\leftarrow \tau\theta_i^{Q} + (1-\tau)\theta_i^{Q'}, \\ \theta^{\mu'} &\leftarrow \tau\theta^{\mu} + (1-\tau)\theta^{\mu'} \end{aligned} \tag{15}$$

where τ is the soft update rate.

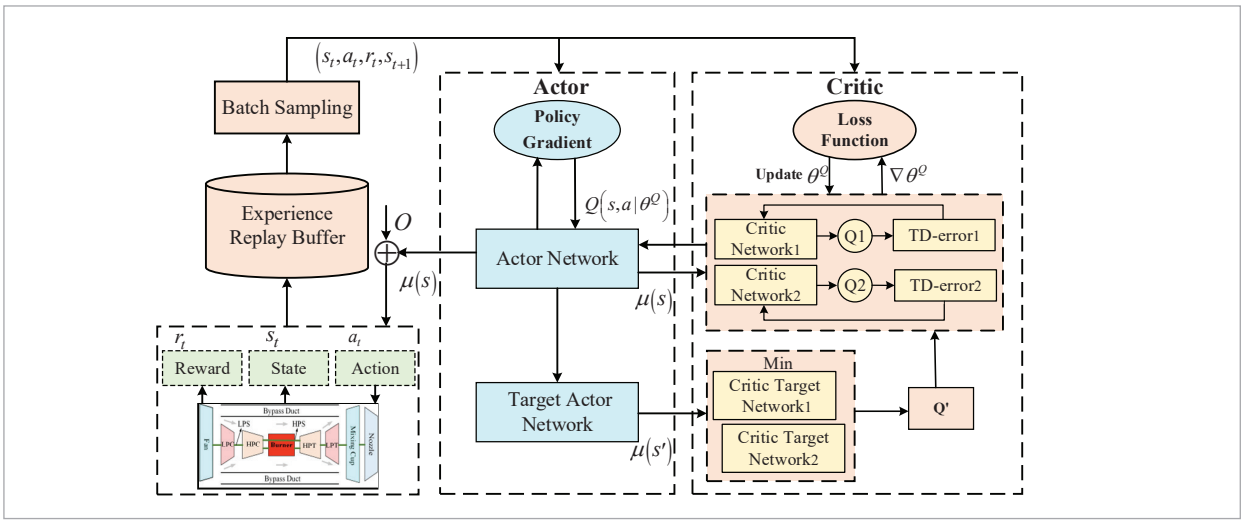
4. Design of Intelligent Controller

4.1. The Structure of the Intelligent Controller

The goal of the intelligent controller is to achieve optimal policy in the interaction with the aero-engine and to complete the control work. The optimal control policy guides it to maximize reward returns. The observations obtained from the environment provide information about the current state of the aero-engine system. And the speed reference signal is embedded in the reward function and observations. Figure 4 shows the workflow of the intelligent controller.

As illustrated in Figure 4, the TD3 algorithm structure involves six deep neural networks, consisting of one Actor network, one Actor target network, two Critic networks, and two Critic target networks. The Actor network generates a control policy based on the current state s_t and adds noise O to produce the current action a_t. The aero-engine executes the action a_t, obtains the current state information, and computes the current reward r_t according to the reward function. The control experience (s_t, a_t, r_t, s_{t+1}) acquired through exploration is subsequently stored to the experience replay buffer. The Actor and Critic neural networks are trained by learning from the relevant experiences in the experience replay buffer. The parameters of the Actor and Critic neural networks are updated based on the aforementioned rules.

Figure 4
The structure of the intelligent controller



4.2. The Design of the Reward Function

The reward function is an important part of the reinforcement learning algorithm. It guides the reinforcement learning agent to learn and affects the convergence speed of the algorithm. In this paper, speed tracking control of the aero-engine is the main assumption. Thus, it is necessary to maintain the controller effectiveness by setting some constraint boundaries for signals.

To enable the intelligent controller to track control of the reference signal, the continuous reward function is designed as:

$$r_1(t) = e^{-0.02*|e(t)|}. \tag{16}$$

The reward function $r_1(t)$ reflects the deviation between the reference signal and the output of the engine, and it is the core component of the reward signal. The agent is rewarded with a higher value if it can reduce the deviation between the low-pressure turbine speed and the reference speed. The value of reward function $r_1(t)$ is in the range of [0, 1]. It will enable the training neural network to converge more rapidly.

The reward functions $r_2(t)$ and $r_3(t)$ are discrete and serve as boundary penalty functions. The input must be constrained to a specified range for the aero-engine to operate normally. As a consequence, the reward function $r_2(t)$ is defined as:

$$r_2(t) = \begin{cases} 0 & 0.01 < FAR < 0.05 \\ -0.2 & other \end{cases}. \tag{17}$$

When the engine control signal fuel-air ratio FAR≤0.01 or FAR≥0.05, the intelligent controller is given a penalty of -0.2. The phenomenon of converging to the boundary value will be effectively alleviated by reward $r_2(t)$.

The reward function $r_3(t)$ is used as a stopping signal for the agent, and is designed to optimize the output of the system. This reward is shown Equation (16):

$$r_3(t) = \begin{cases} 0 & 2000 \le n_L \le 5000 \\ -100 & other \end{cases}. \tag{18}$$

When the output of the system exceeds the reasonable range, the agent would stop training in the round and receives a penalty of -100. This reward function can expedite the algorithm's convergence to a certain extent.

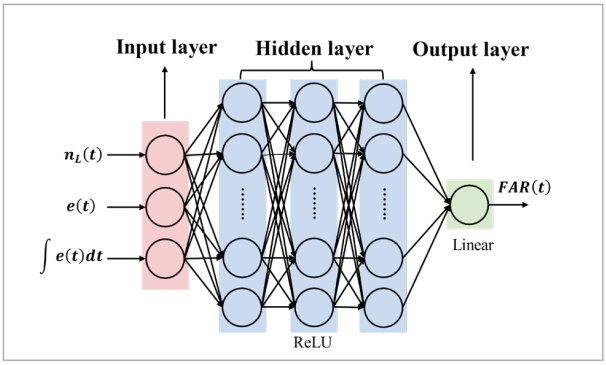
Therefore, the final reward received by the agent is:

$$r(t) = r_1(t) + r_2(t) + r_3(t). \tag{19}$$

4.3. The Design of Neural Network Structure

The neural network in this paper is a multi-layer feedforward neural network. The information about the Actor neural network is revealed in Figure 5. There are five layers in the network, which are one input layer, three hidden layers, one output layer. The input layer of the actor network has three neurons, which connected to three input variables, namely, for the low-pressure turbine speed $n_L(t)$, the error signal between the given speed and the output speed of the turbofan turbine $e(t)$ and the integral of error $\int e(t)dt$. The action value $FAR(t)$ is the output of actor network. The role of the first hidden layer is to map states into features and the activation function is the ReLU. The purpose of the last hidden layer is to normalize the output of the previous layer and exports the action values. To some extent, the Actor neural network can be referred to as an end-to-end control strategy. It can convert the low-pressure turbine speed and error information into a fuel flow signal without the need for manual design of intermediate control logic. Therefore, it is sufficient to deploy the trained Actor neural network on the embedded system. The embedded system only needs to handle the inference of the Actor neural network, which has much lower computational requirements than training the network. It only requires approximately 5000 floating-point

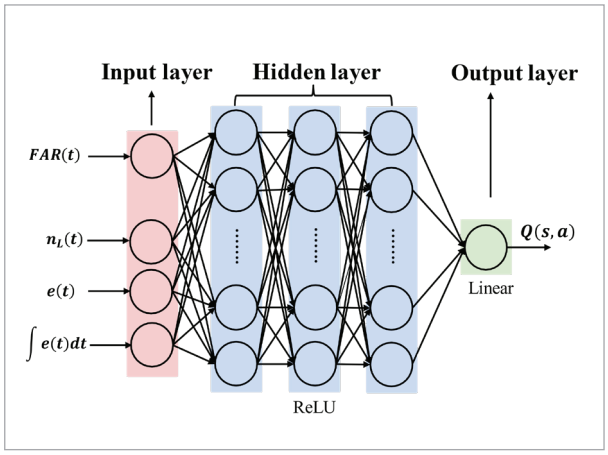
Figure 5
The structure of the actor neural network



operations. High-performance microcontrollers can achieve real-time inference and control.

The specific details of the Critic neural network are depicted in Figure 6. It also contains one input layer, three hidden layers and one output layer. The input layer of the critical neural network corresponds to a 3-dimensional state space and a 1-dimensional action space. The role of the first hidden layer is still to extract the input states as features, and the activation function is the ReLU. It allows for sparsity in the network and better exploration of relevant features. The input of the second hidden layer is the feature which is weighted and summarized by ReLU to output. The final hidden layer takes the output from the previous layer as input and uses an activation function to output Q values to the output layer.

Figure 6
The structure of the critic neural network



5. Simulation and Analysis

We perform some comparative simulations to validate the efficacy and superiority of the TD3 controller in this section. To verify the intelligent controller under different speed phases, the simulation experiments are conducted from the speed range of 2500 rpm to 4500 rpm. Both the acceleration and the deceleration process are considered in our simulations. For example, case 1: the acceleration process control of aero-engine; case 2: the deceleration process control of aero-engine. We then evaluate the tracking performances at different operating speed.

5.1. Simulation Setup

In the simulation, we compare the control performance of TD3, DDPG, and PID controller. For TD3 and DDPG controller, we choose the best control policy in the training process, and we also set some common parameters uniformly. The relevant parameters in the training process are shown in Table 1. As illustrated in Section 4.3, the hidden layers of both the Actor and Critic neural networks are set to three layers. Both the Actor and Critic neural networks contain hidden layers with 50, 25, and 25 neurons. The selection of the learning rate is one of the most critical hyperparameter, as it has a significant impact on the convergence and learning speed. The usage of a larger learning rate can lead to non-convergence of the neural networks. Conversely, a smaller learning rate may increase the probability of model convergence, but can also impact the rate of convergence. In this study, the learning rate for the Actor neural network was set to 0.0001, and the learning rate for the Critic neural network was set to 0.001. Gradient thresholding for neural networks typically involves gradient clipping, which can help mitigate the issue of exploding or vanishing gradients during training. As a result, the stability and performance of the algorithm can be improved. For the training of the Actor and Critic neural networks, a threshold of 1 was set for gradient clipping. The soft update parameter is utilized to update the target neural network. In this paper, the soft update rate is 0.001. In this paper, the experience re-

Table 1
The table of training parameters

Name of the parameter	Parameter Value
Number of hidden layers	3
Number of Actor hidden units	[50,25,25]
Number of Critic hidden units	[50,25,25]
The learning rate of Actor	0.0001
The gradient threshold of Actor	1
The learning rate of Critic	0.001
The gradient threshold of Critic	1
Soft update rate	0.001
Size of the replay buffer	1000000
Number of samples per minibatch	256

play buffer size was set to 1000000, which can store a greater amount of information. The neural network was trained using batch samples, and the size of batch samples is 256 in this study.

In addition, the PID controller is adjusted to obtain the best dynamic response. In the acceleration process control of aero-engine, the PID controller parameters are set as k_p = 0.000008, k_i = 0.000008, k_d = 0. In the deceleration process control of aero-engine, the PID controller parameters are set as k_p = 0.000007, k_i = 0.000006, k_d = 0.

5.2. Case 1: The Acceleration Process Control of Aero-engine

To verify the effect of the TD3 controller and the benefits of the reward function designed in Section 4.2, we gradually increase the low-pressure turbine speed from 2500 rpm to 4500 rpm in the simulation. Meanwhile, the equivalent experiments of the DDPG controller are completed. The principles of the two algorithms and the corresponding details are described above. There, we explore the differences between the TD3 algorithm and the DDPG algorithm in intelligent controller training design. The average reward curve is shown in Figure 7, and the specific performance indicators are listed in Table 2.

Figure 7 implies that the TD3 algorithm has achieved better performance on aero-engine control task. The blue curve and the red curve represent the average reward values of the TD3 algorithm and the DDPG algorithm during training. The average reward function value is obtained by computing the mean reward value over the current episode as well as all previous episodes. The shaded area shows the fluctuation range of the actual episode reward value. In the training process, the reward value of TD3 agent begins to rise around the 13th round. In previous explorations, the actions of the agent could easily touch the boundary value and stop training. In the following parts, the reward value fluctuates between 900 and 1000. However, the average reward for the DDPG algorithm fluctuates around 800. In Table 2, we present a comparative performance analysis of the DDPG and TD3 algorithms. The average reward value achieved at the point of final convergence for each algorithm is used as a metric for evaluating their performance. Additionally, the last column of the table reports the number of episodes required for the reward values of both algorithms to converge to a stable value. Table 2 indicates that the average reward value of TD3 algorithm is about 100 higher than that of DDPG algorithm. The simulation results show that TD3 algorithm overcomes Q value overestimation and converges to a better control policy.

Figure 7
The curve of the average reward value

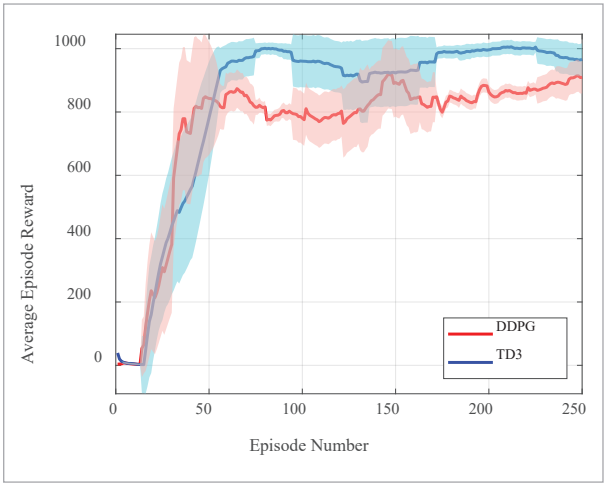


Table 2
The comparison of DDPG and TD3

Algorithm	Performance	Convergence
DDPG	858.519	172
TD3	988.206	180

The control performances of the three controllers are compared and the results are displayed in Figure 8. It is evident that all three controllers demonstrate a favorable impact on the tracking control of the low-pressure turbine speed. Compared with PID controller, the intelligent controllers based on reinforcement learning algorithm exhibit notably superior tracking performance. In each speed range, the intelligent controller can obtain the optimal control strategy according to the change of the speed. It generates smaller overshoot and faster response. Furthermore, the benefit of the intelligent controller gets apparent with the higher speed range. Consequently, at the acceleration process, the proposed controller provides a more stable and faster system response.

For the purpose of quantifying the control performance of three controllers, we list the relevant control

Table 3
Performance comparisons for Case 1

Speed Range(rpm)		2500-3000	3500-4000
Rising Time(s)	TD3	0.52	0.68
	DDPG	0.84	0.96
	PID	2.60	1.88
Setting Time(s)	TD3	0.84	0.88
	DDPG	0.86	0.96
	PID	2.92	3.52
Overshoot (%)	TD3	0.3862	0.1285
	DDPG	0.0704	0.1280
	PID	0.7677	1.2267

performance indicators at two speed levels. Clearly, the intelligent controller performs great self-learning ability on the optimal control policy. As illustrated in Table 3, the rising time for the three controllers when the low-pressure turbine speed increases from 2500 to 3000 rpm are 0.52, 0.84, and 2.6. Regarding the setting time, the TD3 controller demonstrates a reduction of 2.08 and 0.02 seconds compared to the PID controller and DDPG controller, respectively. There is no substantial variation in the amount of overshoot between the three controllers. However, after stabilization, the PID controller exhibits slightly more fluctuations compared to the other controllers. The scenario is essentially analogous in the speed range of 3500-4000, where the TD3 controller minimizes the setting time by 0.08 and 2.64 seconds compared to the other two controllers. Based on Figure 8, it can be observed that the PID controller displays the least overshoot only within a specific speed range (e.g., 3000-3500rpm) across a wide range of speeds. Yet the smaller overshoots are obtained at the cost of more settling time and more rise time. While for the similar overshoot, TD3 controller has a comparative advantage over DDPG controller in terms of rapidity. This provides additional support for the observation that the TD3 algorithm ultimately achieves a higher reward value than the DDPG algorithm. In short, the intelligent controller based on TD3 algorithm generates more rapid response with less speed overshoot than other controllers.

With the aim to properly measure the static and dynamic errors of the controllers, the integral absolute error (IAE) is introduced:

$$IAE = \int_0^{\infty} \left| n_{Lr} - n_L \right| dt \,, \tag{20}$$

where n_{Lr} and n_L are the reference speed and the low-pressure turbine speed of the aero-engine, re-

Figure 8
The acceleration process control of aero-engine

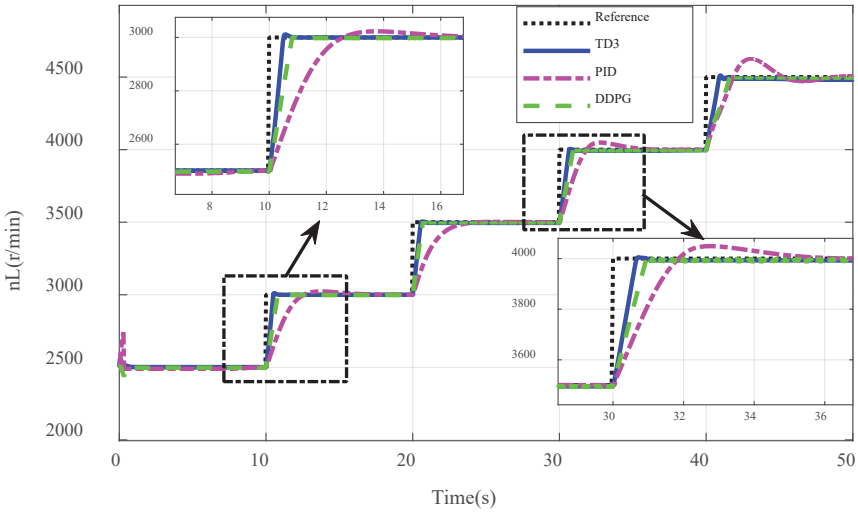
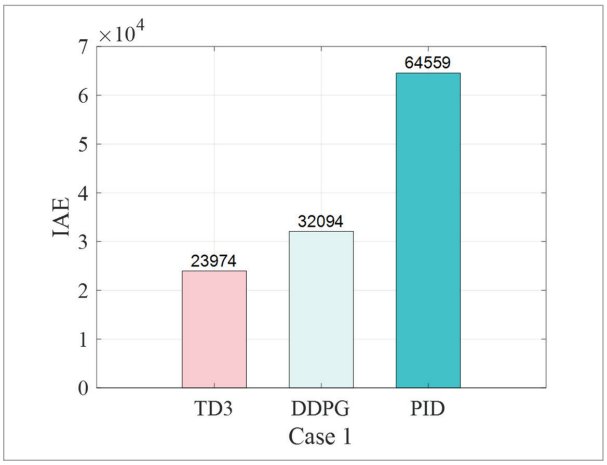


Figure 9
IAE of the speed tracking of Case 1



spectively. Figure 9 displays the IAE derived from the simulation outcomes of each controller throughout the speed-up phase. In comparison with the PID controller, TD3 controller yields 40585 reductions on integral absolute error. It is 8120 less than the error of DDPG controller. As a result, intelligent controller reveals more superior performance than traditional one in speed tracking control task. Moreover, the control policy obtained by TD3 algorithm is more superior.

5.3. Case 2: The Deceleration Process Control of Aero-engine

To further observe the performance of the TD3 controller under different working conditions, we also simulate the tracking control of descending step signal. We step down the low-pressure turbine speed signal from 4500rpm to 2500rpm. The performance comparison results between the DDPG algorithm and TD3 algorithm during the speed-down phase are similar to those presented in Section 5.2. This will not be further elaborated here. The control performance of the three controllers is shown in Figure 10.

Obviously, all the three controllers realize the speed tracking control in the descending speed section. Meanwhile, the intelligent controllers show more significant advantages in rapidity. The three controllers result in very similar effects on overshoot. Compared with Figure 8, traditional PID controller has higher overshoot in high speed range, as it adapts poor to fixed parameters. The control policy of the intelligent controller can fit the change of speed to achieve effective control at all speed.

To have a further investigation, we computed performance metrics for each of the three controllers in Table 4. In the 3000-3500rpm speed range, the advantage of the TD3 controller in terms of response speed

Figure 10
The deceleration process control of aero-engine

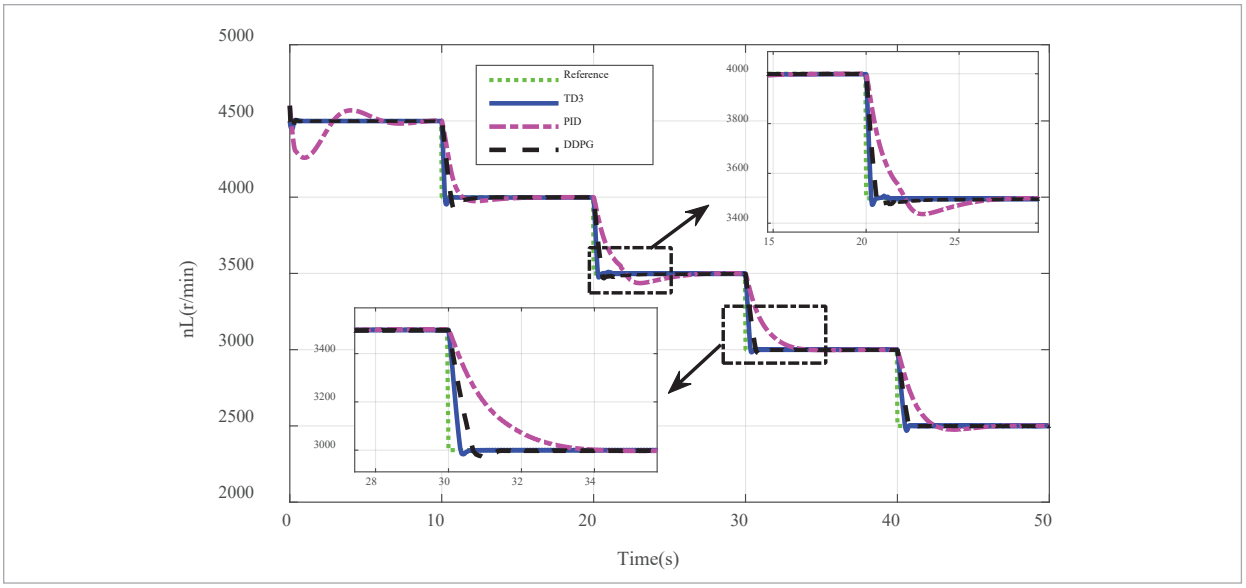
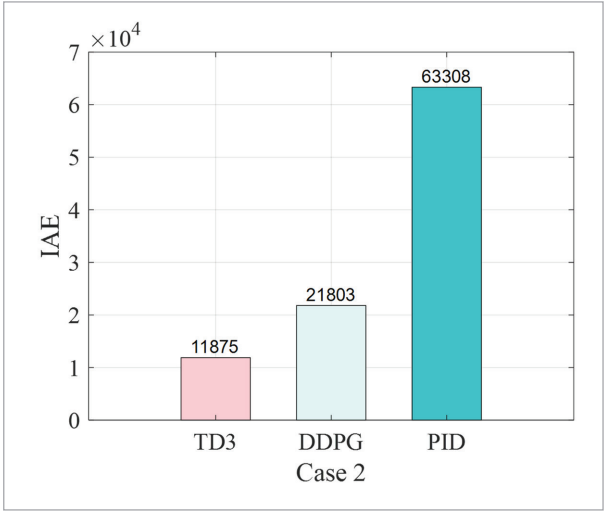


Table 4
Performance comparisons for Case 2

Speed Range(rpm)		3000-3500	3500-4000
Rising Time(s)	TD3	0.28	0.32
	DDPG	0.56	0.68
	PID	2.12	3.88
Setting Time(s)	TD3	0.56	0.60
	DDPG	3.36	1.56
	PID	6.24	3.88
Overshoot (%)	TD3	0.74	0.4951
	DDPG	0.91	0.8182
	PID	1.82	0

is more prominent. The rise time of the TD3 controller is significantly shorter than that of the DDPG and PID controllers by 0.28s and 1.84s, respectively. The setting time of three controllers are 0.56s, 3.36s and 6.24s. In the overshoot, the TD3 controller exhibits reductions of 0.17 and 1.08 compared to the other two controllers. In the speed range of 3500-4000rpm, it should be noted that although the PID controller does not exhibit overshoot, it results in a longer rise time and setting time. The TD3 controller demonstrates more outstanding results in terms of rise time and settling time. According to the simulation results, it is apparent that the reinforcement learning algorithm demonstrates exceptional performance in aero-engine control due to its optimal control characteristics. Between the TD3 and DDPG algorithms, the TD3 algorithm displays a more pronounced superiority.

Figure 11
IAE of the speed tracking of Case 2



The same as Case 1, the IAE is also used to analyze the control performance of the controllers. Figure 11 demonstrates that the integral absolute error of TD3 controller is the least. Compared with other controllers, it achieves 9928 and 51433 reductions on integral absolute error. Therefore, TD3 controller exhibits excellent tracking control performance of descending step signal.

6. Conclusion

In this paper, we propose an aero-engine intelligent controller design method based on TD3 algorithm. The major advantage of this method is its ability to design the controller without significant knowledge of the aero-engine model. The proposed approach presents an effective solution to compensate for the limitations of model-based control algorithms in aero-engines that involve complex aerodynamic and thermodynamic processes. The detailed design flow of intelligent controller is given which provides a solid basis for future practical applications of reinforcement learning on turbofan engines. The research confirms the feasibility and advantages of the method. The comparison simulation results prove that the TD3 controller enables tracking control of low-pressure turbine speed with faster response and less overshoot. Future work will extend this approach to include more input variables and cover a wider range of operating conditions. Additionally, the trained control strategies will be written into the embedded system for further testing and evaluation. We believe that guided by reinforcement learning theory, the aero-engine intelligent control method will achieve greater development and have strong effect in practice in the near future.

Acknowledgement

This work is supported by Advanced Jet Propulsion Creativity Center, AEAC (Project ID.HK-CX2020-02-019).

References

1. Arora, L., Dutta, A. Reinforcement Learning for Sequential Low-thrust Orbit Raising Problem. AIAA Scitech 2020 Forum, 2020, 2186. https://doi.org/10.2514/6.2020-2186
2. Clifton, J., Laber, E. Q-learning: Theory and Applications. Annual Review of Statistics and Its Application, 2020, 7, 279-301. https://doi.org/10.1146/annurev-statistics-031219-041220
3. Fujimoto, S., Hoof, H., Meger, D. Addressing Function Approximation Error in Actor-Critic Methods. International Conference on Machine Learning. PMLR, 2018,80,1587-1596.
4. Gou, L., Liu, Z., Fan, D., Zheng, H. Aeroengine Robust Gain-scheduling Control Based on Performance Degradation. IEEE Access, 2020, 8, 104857 - 104869. https://doi.org/10.1109/ACCESS.2020.2986336
5. Haiquan, W., Ying-qing, G., Jun, L., Guangyao, L. Aero-engine Control Design Using Two Degrees-of-Freedom H∞ Approach. 2008 2nd International Symposium on Systems and Control in Aerospace and Astronautics, 2008, 1-5. https://doi.org/10.1109/ISSCAA.2008.4776136
6. Hasselt, H. Double Q-learning. Advances in Neural Information Processing Systems, 2010, 23.
7. Hovell, K., Ulrich, S. On Deep Reinforcement Learning for Spacecraft Guidance. AIAA Scitech 2020 Forum, 2020, 1600. https://doi.org/10.2514/6.2020-1600
8. Jaw, L., Mattingly, J. Aircraft Engine Controls: Design. System Analysis, and Health Monitoring, 2009, 136-138. https://doi.org/10.2514/4.867057
9. Jiang, H., Gui, R., Chen, Z., Wu, L., Dang, J., Zhou, J. An Improved Sarsa (λ) Reinforcement Learning Algorithm for Wireless Communication Systems. IEEE Access, 2019, 7, 115418-115427. https://doi.org/10.1109/ACCESS.2019.2935255
10. Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., Wierstra, D. Continuous Control with Deep Reinforcement Learning. Computer Science, 2016, 8(6), A187.
11. Liu, S., Bai, J., Wang, Q., Wang, W. Tracking Controller Design for Aero-engine Based on Improved Multi-Power Reaching Law of Sliding Mode Control. International Journal of Aeronautical and Space Sciences, 2019, 20, 722-731. https://doi.org/10.1007/s42405-019-00159-4
12. Mehmood, A., Ali, A. Application of Deep Reinforcement Learning for Tracking Control of 3WD Omnidirectional Mobile Robot. Information Technology and Control, 2021, 50(3), 507-521. https://doi.org/10.5755/j01.itc.50.3.25979
13. Miao, K., Wang, X., Zhu, M., Yang, S., Pei, X., Jiang, Z. Transient Controller Design Based on Reinforcement Learning for a Turbofan Engine with Actuator Dynamics. Symmetry, 2022, 14(4), 684. https://doi.org/10.3390/sym14040684
14. Miller, D. E., Rossi, M. Simultaneous Stabilization with Near Optimal LQR Performance. IEEE Transactions on Automatic Control, 2001, 46(10), 1543-1555. https://doi.org/10.1109/9.956050
15. Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wiestra, D., Legg, S., Hassabis, D. Human Level Control Through Deep Reinforcement Learning. Nature, 2015, 518(7540),529-533. https://doi.org/10.1038/nature14236
16. Montazeri-Gh, M., Rasti, A., Jafari, A., Ehteshami, M. Design and Implementation of MPC for Turbofan Engine Control System. Aerospace Science and Technology, 2019, 92,99-113. https://doi.org/10.1016/j.ast.2019.05.061
17. Navarro-Tapia, D., Marcos, A., Bennani, S. The VEGA Launcher Atmospheric Control Problem: A Case for Linear Parameter-varying Synthesis. Journal of the Franklin Institute, 2022, 359(2), 899-927. https://doi.org/10.1016/j.jfranklin.2021.07.057
18. Qian, R. R., Feng, Y., Jiang, M., Liu, L. Design and Realization of Intelligent Aero-engine DDPG Controller. Journal of Physics: Conference Series, IOP Publishing, 2022, 2195(1),012056. https://doi.org/10.1088/1742-6596/2195/1/012056
19. Richter, H. Advanced Control of Turbofan Engines. Springer Science & Business Media, 2011. https://doi.org/10.1007/978-1-4614-1171-0
20. Silver, D., Lever, G., Heess, N., Degris, T., Wierstra, D., Riedmiller, M. Deterministic Policy Gradient Algorithms. International Conference on Machine Learning, PMLR, 2014, 32(1), 387-395.
21. Sun, G., Wang, Z., Zhao, J. Automatic Text Summarization Using Deep Reinforcement Learning and Beyond. Information Technology and Control, 2021, 50(3), 458-469. https://doi.org/10.5755/j01.itc.50.3.28047
22. Sutton, R. S., Barto, A. G. Reinforcement learning: An Introduction. MIT press, 2018.

23. Tan, J., Zhang, T., Coumans, E., Iscen, A., Bai, Y., Hafner, D., Bohez, S., Vanhoucke, V. Sim-to-Real: Learning Agile Locomotion for Quadruped Robots. Robotics: Science and Systems, 2018. https://doi.org/10.15607/RSS.2018.XIV.010

24. Xian, B., Zhang, X., Zhang, H., Gu, X. Robust Adaptive Control for a Small Unmanned Helicopter Using Reinforcement Learning. IEEE Transactions on Neural Networks and Learning Systems, 2021, 33(12), 7589-7597. https://doi.org/10.1109/TNNLS.2021.3085767

25. Zhang, H., Wei, S., Xu, G. Steady State Controller Design for Aero-engine Based on Reinforcement Learning NNs. 2017 29th Chinese Control And Decision Conference (CCDC), IEEE, 2017, 2168-2173. https://doi.org/10.1109/CCDC.2017.7978874

26. Zhao, L., Fan, D., Shan, W. Single-spool Turbofan Engine Model Identification. Journal of Propulsion Technology, 2008, 29(6), 733-736.

27. Zheng, Q., Jin, C., Hu, Z., Zhang, H. A Study of Aero-engine Control Method Based on Deep Reinforcement Learning. IEEE Access, 2019, 7, 55285-55289. https://doi.org/10.1109/ACCESS.2018.2883997

28. Zheng, T., Wang, X., Luo, X., Li, Q. Modified Method of Establishing the State Space Model of Aeroengine. Journal of Propulsion Technology, 2005, 26(1), 46-49.

29. Zhou, W., Shan, X., Geng, Z., Huang, J. Establishment of State Space Model of Turboshaft Engine with Self-optimized Method. Journal of Aerospace Power, 2008, 23(12), 2314-2320.

30. Zhu, K., Zhao, J., Dimirovski, G. M. H∞ Tracking Control for Switched LPV Systems with an Application to Aero-engines. IEEE/CAA Journal of Automatica Sinica, 2016, 5(3), 699-705. https://doi.org/10.1109/JAS.2016.7510025