IM-TD3

A Reinforcement Learning Approach for Liquid Rocket Engine Start-Up Optimization

Liu, Yuwei; Li, Yang; Cheng, Yuqiang; Pan, Wei; Wu, Jianjun

DOI
10.1109/TAES.2024.3471494

Publication date
2025

Document Version
Final published version

Published in
IEEE Transactions on Aerospace and Electronic Systems

IM-TD3: A Reinforcement Learning Approach for Liquid Rocket Engine Start-Up Optimization

YUWEI LIU
National University of Defense Technology, Changsha, China

YANG LI
Chinese Academy of Sciences, Shanghai, China

YUQIANG CHENG
National University of Defense Technology, Changsha, China

WEI PAN, Member, IEEE
Delft University of Technology, Delft, Netherlands

JIANJUN WU
National University of Defense Technology, Changsha, China

With advancements in reusable liquid rocket engine technology to meet the diverse demands of space missions, engine systems have become increasingly complex. In most cases, these engines rely on stable open-loop control and closed-loop regulation systems. However, due to the high degree of coupling and nonlinear dynamics within the system, most transient adjustments still depend on open-loop control. Open-loop control often fails to provide the optimal control strategy when encountering external disturbances. To address this issue, we introduce the intrinsically motivated twin delayed deep deterministic (TD3) algorithm, specifically designed for the startup process of LOX/Kerosene high-pressure staged combustion engine. This approach leverages intrinsic motivation to enable the algorithm to adapt to the abrupt parameter changes during the start-up process. A series of comprehensive experiments were conducted to verify the effectiveness of our method. The experimental results demonstrate that our method outperforms both the PID method and previous researchers' reinforcement learning methods based on the TD3 algorithm and DDPG, achieving a faster and more stable start-up process and significantly enhancing engine performance.

Received 7 June 2024; revised 3 September 2024; accepted 17 September 2024. Date of publication 14 October 2024; date of current version 14 April 2025.

DOI. No. 10.1109/TAES.2024.3471494

Refereeing of this contribution was handled by C.-H. Lee.

This work was supported in part by the China Scholarship Council under Grant 202306110030 and in part by the National Natural Science Foundation of China (NSFC) Innovative Research Group Project "Matter Transport and Energy Transformation in Space Power System" under Grant T2221002.

Authors's addresses: Yuwei Liu, Yuqiang Cheng, and Jianjun Wu are with the College of Aerospace Science and Engineering, National University of Defense Technology, Changsha 410073, China, E-mail: (liuyuwei19@nudt.edu.cn; cheng_yuqiang@163.com; jjwu@nudt.edu.cn); Yang Li is with the Shanghai Institute of Microsystem and Information Technology, Chinese Academy of Sciences, Shanghai 201210, China, E-mail: (yang.li-4@outlook.com); Wei Pan is with the Department of Cognitive Robotics, Delft University of Technology, 2628 CD Delft, The Netherlands, E-mail: (panweihit@gmail.com). *(Corresponding authors: Jianjun Wu; Wei Pan.)*

I. INTRODUCTION

With the increasing complexity of human space missions and the in-depth exploration and utilization of space, the performance requirements for rocket engines are also increasing [1], [2]. The liquid propellant rocket engine (LRE), as the most important power device at present, needs to undergo complex transient dynamic processes during launch and flight, such as ignition, thrust adjustment, and shutdown [3]. The accurate control of these transient processes is crucial to ensuring the reliability and safety of the engine. Although traditional control methods perform well in many cases, their limitations gradually become apparent when dealing with highly nonlinear, multivariable, and extreme operating conditions [4]. Furthermore, since SpaceX achieved the engineering of reusable rocket engines, reusable engines have also emerged as a prominent research focus in the field of engine studies, presenting new challenges and potential solutions to traditional control methods [5], [6], [7].

Therefore, with respect to the control issues of rocket engines, particularly for reusable engines, researching control over operational condition variations is crucial. This research must ensure not only precise control of engine performance during individual launches but also consider long-term maintainability and reusability [8], [9]. Minimizing damage to engine components is essential, as it significantly impacts engine stability, reliability, and reusability [10]. Consequently, the control system must be capable of adapting in real time to changes in engine conditions, and it should adjust control strategies to address potential performance degradation and fault progression over extended periods of use [11]. Furthermore, as the complexity and frequency of launch missions increase, traditional control methods may struggle with the requisite response speed and adaptability needed for rapidly evolving environments and parameters [12]. In traditional engine start-up control, most valve control logic is developed through extensive ground testing. While these control strategies ensure that the engine operates within a safe and reliable range, they are inadequate for current closed-loop control systems and future reusable engines. This traditional approach cannot effectively respond to unexpected situations during flight or address conditions not considered during ground tests. Moreover, it does not take into account the engine's lifespan and long-term maintainability, necessitating maintenance after each use, which prolongs the launch cycle. Therefore, as the demands of space missions continue to grow, the need for optimized engine control will also increase.

In this context, reinforcement learning (RL) offers a novel possibility. It continuously optimizes control strategies by learning from interactions with the environment, thereby adapting to changes in the environment and its own performance [13]. RL has been extensively researched for a variety of task requirements in the aerospace domain, including drone trajectory optimization [14], [15], aviation braking [16], spacecraft lunar landings [17], spacecraft maneuvers in space [18], and launch vehicle guidance [19], [20].

Falcone and Pulnam [21] implemented deep reinforcement learning (DRL) within the domain of aviation braking, and designed a 3-D reward function to ensure a stable learning trajectory for RL agents. They proposed a deep Q-learning architecture tailored for aviation brake maneuver planning and decision-making. This architecture leverages parallel simulations and a directional exploration strategy that utilizes locally observable environments. A Comparative analysis with the most recent autonomous pneumatic braking heuristic algorithms was conducted. The findings indicated that DRL methods are capable of making more robust decisions under extreme conditions. Tipaldi et al. [22] analyzed multiple specific applications of current RL technology in the aerospace field, such as guidance, navigation, and control systems for spacecraft landing on planets, orbit transfer, trajectory planning for interstellar missions, attitude control systems for spacecraft, guidance for rendezvous and docking and approach maneuver scenarios, constellation orbit control, as well as on-board decision-making, operation scheduling, and rover path planning for spacecraft. They explored how to meet the mission requirements of spacecraft through RL solutions and implement a controller (i.e., RL agent) that is robust to system uncertainty and can adapt to constantly changing environments. For each application field, the core elements of the RL framework were also discussed, including reward functions, RL algorithms, and environmental models for RL agent training, which are key guiding factors for solving spacecraft control problems through the RL framework. Maicke [23] conducted a comprehensive review of the application of machine learning in rocket propulsion systems, focusing on four primary areas: 1) fault diagnosis, 2) modeling assistance, 3) control, and 4) experimental data analysis. He discussed the reasons behind the relatively slow adoption of machine learning in this field compared to computational fluid dynamics. These reasons include:

1) the state of the propulsion system directly impacts the overall state of the rocket, necessitating a conservative approach to development to ensure safety;
2) effective application of machine learning in rocket propulsion not only demands expertise in machine learning algorithms but also requires a deep understanding of the specific field to develop appropriate machine learning tools for propulsion systems;
3) challenges related to data reliability and extrapolation, such as acquiring reliable experimental data for propulsion systems and using this data to create a high-fidelity database, are critical for advancing the use of interest-based learning tools in propulsion.

In subsequent studies, Dresia et al. [24] investigated the benefits of using neural network controllers with open-loop sequences. Before use, train the control using a simulated environment. The neural network was found to be closer to the reference values used for testing. They hope to expand control research to explain further unexpected events, including adaptation to exercise operations in unstable situations. Zavoli et al. [25] used neural networks as an alternative model for predicting the performance of hybrid rocket engines. Establish a medium-fidelity HRE trajectory model for generating sample data for network training. Finally, the trained network is used as an alternative HRE model in the multidisciplinary optimization process to determine the optimal HRE design and ascent trajectory for the Vettore Europeo di Generatione Avanzata-derived three-stage launch vehicle, where HRE replaces the last two stages of the original four-stage rocket. Oestreich et al. [26] proposed a six-degree-of-freedom rotating target docking excitation strategy using RL for spacecraft rendezvous and docking problems, which is effectively applied to satellite services and automatic docking of orbital debris. Adopting a proximal strategy optimization algorithm, this method performs well in simulated environments, optimizing performance, and controlling costs.

With the continuous strengthening of research on the application of artificial intelligence in spacecraft guidance, navigation, and control, Brandoniso et al. [27] adopted DRL to reconstruct the shape of unknown target objects in relative dynamic scenes, achieving adaptive guidance and control. Similarly, in the context of spacecraft navigation and guidance, Ciabatti et al. [28] employed DRL alongside transfer learning techniques. These methods were used to train models that excel in specific environments. Importantly, they successfully transferred these strategies to new environments while maintaining their optimal performance. Hörger et al. [29] conducted preliminary development of a controller based on RL for the efficient operation of reusable engines. Their study focused on a propulsion system using N_2O/C_2H_6 as green propellants. The control objectives included adjusting the mixture ratio and combustion pressure. The existing test bench was modeled in EcosimPro / ESPSS and controlled through a simulation model using DRL for controller training. Preliminary experiments validated the fundamental capabilities of the RL-based controller in actual rocket propulsion systems, marking an important step toward safe and intelligent engine control. In addition, Horger et al. [30] proposed a controller for cold gas thruster pressure based on a RL method. Through simulation and experimental data, they demonstrated the controller's stability and accuracy in various pressure setpoints and target pressure tracking tasks. However, the Horger's approach involved continuous open-loop control of thrust, with limited consideration given to transient closed-loop control. Amarthya et al. [31] applied RL to spacecraft automatic thrust vector control systems. By analyzing sensor data and

simulation models, the agent autonomously determined the optimal thrust vector and magnitude, achieving efficient closed-loop control. However, they only employed a greedy algorithm, and the trained model did not fully converge. Waxenegger-Wilfing et al. [32] studied a certain type of gas-cycle rocket engine system and analyzed the control strategy of engine transient regulation using the RL method.

Although researchers have extensively applied RL across the aerospace sector, studies on its application to LRE systems are relatively scarce. Notably, the work by Waxenegger [33] and colleagues on RL control for open-cycle engines made some progress but still faces significant limitations. For instance, their research did not analyze LOX/Kerosene high-pressure staged combustion engine, nor did it sufficiently consider the sequence and coordination of valve opening times. As a result, it remains unclear whether RL control methods based on the TD3 algorithm are applicable to actions that require a specific sequence. For reusable engines, an incorrect valve sequence can severely impact the engine's service life. Based on this, we believe it is necessary to delve deeper into this issue. Consequently, this article aims to optimize the start-up and transient control of liquid rocket engines by improving RL algorithms and control strategies.

Our research mainly includes the following points.

1) By incorporating RL into the startup process of an LOX/Kerosene high-pressure staged combustion engine and considering the characteristics of parameter variations during startup, we successfully transformed the engine startup problem into a RL problem. This was achieved by defining the observation and action spaces and utilizing an engine simulation platform alongside RL algorithms.
2) Based on the characteristics of the startup process in the gas generator staged combustion cycle engine, we designed a segmented reward function that includes both hard and soft constraints. In addition, we proposed improvements to the TD3 algorithm, enabling the reward function to stabilize quickly during training. We further designed, trained, and evaluated an RL controller.
3) We conducted a quantitative comparison between open-loop control and PID controllers in engineering applications and compared these methods with those used by previous researchers.

II. PRELIMINARIES AND MODELING

In this section, we will briefly introduce the relevant concepts of RL and the LOX/Kerosene high-pressure staged combustion engine study in our work. RL is a type of machine learning [33]. DRL is an organic combination of RL and deep learning, which breaks through the limitation of traditional tabular RL methods that can only use low-dimensional inputs and have better feature extraction ability than other function fitting methods. It is considered the most promising development direction in the field of artificial intelligence.

A. Basic Theory of Reinforcement Learning

RL can be abstracted as a Markov decision process, where the agent observes its state S_t at time t and selects action a_t based on strategy π. The environment feedbacks the reward r_{t+1} to the agent for the next moment, and the agent enters a new state S_{t+1}.

According to different learning objectives, RL can be divided into two categories: 1) value-based RL algorithms and 2) policy-based RL algorithms. The goal of value-based RL algorithms is to learn an optimal action state value function $Q^*(s, a)$, and typical algorithms include Q-learning, state-action-reward-state-action (SARSA), etc. The goal of policy-based RL algorithms is to learn an optimal policy π^*, and typical algorithms include trust region policy optimization (TRPO), proximal policy optimization (PPO), etc. On the basis of these two methods, Konda and Tsitsiklis proposed an actor-critic algorithm [34], which combines value-based methods with policy-based methods, while learning both the policy function and value function. Compared with strategy-based and value-based RL algorithms, the actor-critic algorithm has advantages, such as high sample utilization, small variance in value function estimation, and fast training speed.

B. TD3 Algorithm

DRL is centered around RL, and on this basis, it utilizes the powerful fitting ability of artificial neural networks to fit the policy function and state action value function $Q(s, a)$. The deep deterministic policy gradient algorithm (DDPG) combines the actor-critic algorithm with the deep Q-network algorithm (DQN) and is a typical DRL algorithm. The DDPG algorithm incorporates a target network into the actor-critic algorithm framework, making network training more stable. However, there are issues with high estimation and high variance in the DDPG algorithm.

In response to issues identified in the DDPG algorithm, the twin delayed deep deterministic policy gradient (TD3) algorithm offers several enhancements. To address the problem of overestimation inherent in the DDPG's critic network, TD3 introduces a second critic network. This dual critic setup operates by fitting the agent's action-state value function using both networks. During the evaluation of the action-state value function, both critic networks provide separate evaluations, and the lower of the two values is selected as the definitive action-state value for the agent. The method for updating the action-state value in the TD3 algorithm is structured as follows:

$$y \leftarrow r + \gamma \min_{i=1,2} Q_{\theta'_i}\left(s', \pi_\phi\left(s'\right)\right). \tag{1}$$

In the formula, $Q_{\theta'_1}(s', \pi_\phi(s'))$ and $Q_{\theta'_2}(s', \pi_\phi(s'))$ represent the estimated values from the two critic networks, respectively.

The high squared error prevalent in the DDPG algorithm can lead to several adverse effects, such as a reduced learning rate, decreased learning performance, and an unstable learning process. The TD3 algorithm addresses this issue by introducing regularization methods, namely, the integration

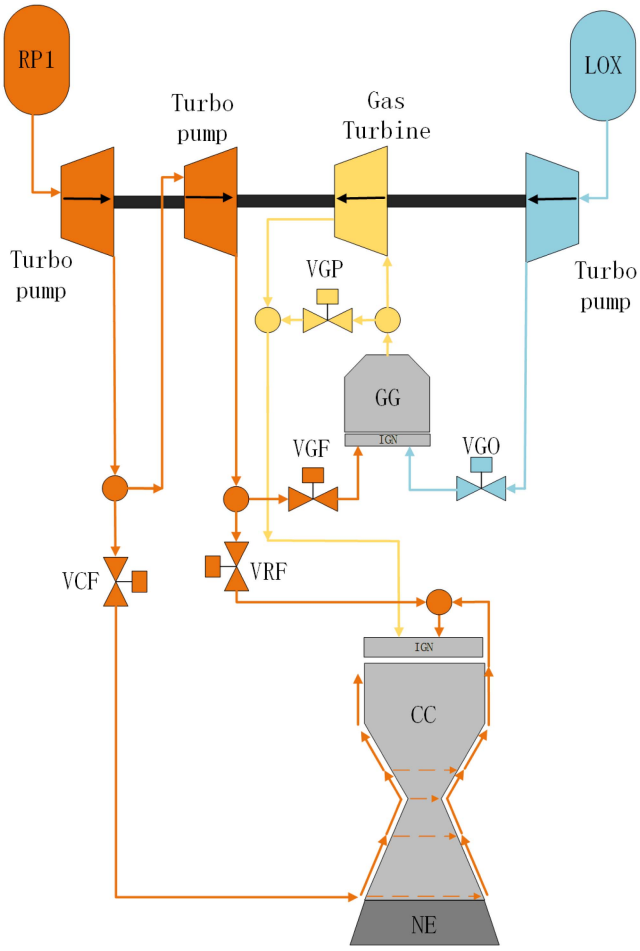

Fig. 1. Structural diagram of LOX/Kerosene high-pressure staged combustion engine.

of random noise into the actions of the intelligent agent. This modification in the agent's actions, post noise addition, aims to enhance the robustness and stability of the learning process

$$a \leftarrow \pi_{\phi'}\left(s'\right) + \varepsilon \quad \varepsilon \sim \mathrm{clip}\left(N\left(0, \tilde{\sigma}\right), -c, c\right). \tag{2}$$

In the formula: $\pi_{\phi'}(s')$ is the action output by the actor network; ε is a random noise.

The TD3 algorithm uses gradient descent for updating, and the gradient formula of the strategy network is as follows:

$$\nabla_{\phi} J(\phi) = N^{-1} \sum \nabla_a Q_{\theta_1}(s, a)|_{a=\pi_{\phi}(s)} \nabla_{\phi} \pi_{\phi}(s). \tag{3}$$

In the formula: $Q_{\theta_1}(s, a)$ is the value of the action state, $\pi_{\phi}(s)$ is the strategy of the intelligent agent, and ∇ is the sign of the gradient.

However, in traditional RL algorithms, each update is randomly sampled from the experience pool, and the selected samples are used to update the neural network. The selection of samples directly determines the convergence speed and results of the algorithm.

C. Rocket Engine System

In this article, an LOX/Kerosene high-pressure staged combustion engine is studied and its system diagram is shown in Fig. 1. The engine operates on a staged combustion cycle, where a small amount of propellant is burned in the precombustion chamber and a gas generator is employed. Low-temperature liquid oxygen and room-temperature kerosene were used as oxidants and fuels, respectively. This gas generator operates with an oxygen-rich mixture ratio. The generated gas is used to drive the gas turbine. This turbine is mechanically connected to the pump, which provides power. The pump then pressurizes the fuel and oxidizer before delivering them to the combustion chamber.

In this system, kerosene serves as a cooling medium to cool the main combustion chamber before all of it enters the gas generator for combustion. Typically, the Laval nozzle includes a noncooled nozzle extension (NE) that accelerates the combustion gas and generates thrust. The engine operating conditions are regulated by five flow control valves (VGO, VGF, VCF, VRF, VGP). VGO is a gas generator oxidizer valve used to regulate the mixing ratio of the gas generator; VGF and VCF are the gas generator fuel regulating valve and the combustion chamber fuel valve, respectively. VRF is a fuel flow control valve that can regulate the flow of the gas generator at low thrust while ensuring sufficient cooling of the thrust chamber. VGP is a gas bypass valve located downstream of the gas generator and upstream of the turbine, which can be used to regulate the power of the turbine and adjust the mixing ratio of the entire system. IGN represents the ignition system of the gas generator and main combustion chamber.

The use of RL for rocket engine control has the following advantages.

1) Rocket engines are complex systems with high thermodynamic fluid mechanical coupling and high energy release ratios. For traditional PID control, some parameters need to be analyzed for observability and controllability, or complex control logic needs to be designed. RL does not require a closed-loop or display of mathematical equations for the entire system and directly uses nonlinear simulation models.
2) The working process of rocket engines is complex, and any disturbance in the environment may have an impact on the engine's state. LRE did not consider the impact of various usage scenarios on the engine during the initial design process, especially the current reusable engine. RL can learn environmental dynamics by simulating training RL strategies (such as neural network weights).
3) RL can design different rewards for different task stages to achieve different goals.

The current shortcomings of RL:

The design of reward functions requires a lot of manual experience.

III. INTRINSICALLY MOTIVATED TD3 (IM-TD3)

To better adapt the TD3 algorithm for the start-up process of the LOX/Kerosene high-pressure staged combustion

engine, we propose the intrinsically motivated TD3 (IM-TD3) algorithm. Improvements to the algorithm include preprocessing, selecting appropriate observation and action spaces, and designing a reward function specifically for the engine startup process.

A. Preprocessing

This algorithm incorporates two main improvements over the standard TD3: 1) conducting ten iterations of updates during the training process; and 2) employing an annealing learning rate. These enhancements can further increase the stability and efficiency of the algorithm, particularly suitable for handling the complex or multimodal reward structures designed for the engine start-up process, especially in the face of "discontinuous" states encountered during engine start-up.

Next, we will detail the two key improvements made. First, during the training process, we perform ten iterations of updates instead of a single data sampling and network update. Each time the training function is called, ten independent iterative updates are executed. During engine start-up, if the start-up is successful, the engine's state parameters can undergo sudden changes, with some parameters experiencing multiple shifts within a fraction of a second. In such cases, a single round of data sampling and updating may not adequately capture all significant state changes in a timely manner. In each training iteration, the algorithm samples multiple data batches from the experience replay buffer. This not only enhances the utilization rate of data but also, through multiple samplings, increases the diversity of the training process, helping the algorithm to capture a more varied set of state-action relationships.

During the start-up process of a rocket engine, in addition to dramatic parameter changes, there are also issues, such as start-up failures and unstable combustion. This requires the model to handle sudden state changes and fluctuations. By computing the cumulative loss from ten iterations of updates and averaging it, we can smooth the training process, reduce random fluctuations in parameter updates, and thereby enhance the algorithm's stability. Another advantage of this method is that it allows for gradual adjustments of the learning rate during the training process to meet the needs of different training phases. Through the annealing learning rate strategy, it facilitates a smooth transition from rapid learning to fine-tuning.

On the other hand, the IM-TD3 algorithm incorporates an annealing learning rate. An annealing learning rate typically involves progressively reducing the learning rate as training advances. During the engine start-up process, it is essential, particularly in the early stages, to experiment with various valve opening sequences and degrees to achieve successful engine ignition. After the engine starts, excessive valve actuation should be avoided as it can cause parameter oscillations. The benefits of implementing an annealing learning rate include the following.

1) *Enhanced Initial Learning Efficiency:* A higher learning rate in the early stages allows the algorithm to converge quickly, reducing significant losses and enhancing learning efficiency. Maintaining a higher initial learning rate could help the algorithm escape local minima or saddle points, encouraging the agent to explore different actions. This is particularly crucial for navigating the complex environments associated with the engines discussed in this article, ensuring rapid performance improvements and successful engine start-up.
2) *Increased Stability in Later Phases:* To prevent system parameter fluctuations and oscillations after the engine has successfully started, the learning rate's gradual reduction decreases the model's update steps. This adjustment helps stabilize the model during the later stages of learning, refining its response to the environment. As the learning rate diminishes, the algorithm may stabilize at a global optimum or a favorable local optimum, minimizing excessive adjustments as the learning nears an optimized state and preventing oscillations.

B. Observation and Action Space

To Train and Use IM-TD3: It is necessary to define the observation space and action space of the agent. Observation space, which refers to the variables received by the agent from the environment at each time step, requires sufficient information to clearly define the current state of the system. In this article, we set the observation space as follows:

$$S = \left[P_G, P_C, F, n_t, n_{\text{fpp}}, \text{MR}_{\text{GG}}, \text{Pos}_{\text{VGO}}, \text{Pos}_{\text{VGF}}, \text{Pos}_{\text{VCF}}\right]. \tag{4}$$

It contains nine state parameters, where P_G, P_C, F, n_t, n_{fpp}, and MR_{GG} are the pressure of the gas generator, the pressure of the main combustion chamber, the magnitude of thrust, the speed of the main turbine, the speed of the fuel prepressure pump, and the mixing ratio of the gas generator. Pos_{VGO}, Pos_{VGF}, and Pos_{VCF} are the opening of the controlled valve. Normalize the observation space using steady-state reference values. Therefore, the observation space and method proposed for this engine are not limited to the simulation environment in this article. In other engine systems or environments, some parameters, such as engine turbine efficiency, cannot be directly measured.

The action space A of the agent consists of the opening degrees of three valves.

$$A = [\text{Pos}_{\text{VGO}}, \text{Pos}_{\text{VGF}}, \text{Pos}_{\text{VCF}}]. \tag{5}$$

At each time step, the RL agent receives environmental observation results and sends control signals to the engine's control valve. The interaction frequency between the RL agent and the environment is 25 Hz.

C. Reward Sharping

The rewards used for training IM-TD3 and evaluating the actions of valves consist of the following different

components:

$$\text{Reward} = r_{\text{target}} + r_f + r_{\text{mix}} + r_{\text{pos}} + r_{\text{act}}. \tag{6}$$

The first item is

$$r_{\text{target}} = 1 - \sum_{\varepsilon_i} clip\left(\left|\frac{\varepsilon_i - \varepsilon_{i,\text{ref}}}{\varepsilon_{i,\text{ref}}}\right|, 0.2\right) \tag{7}$$

where $\varepsilon_i \in [P_G, P_C, F, n_t, n_{\text{fpp}}]$ is the target value for different parameters. Each reward component in this item is trimmed to a maximum value of 0.2 to improve the cumulative reward during training balance initiation and steady-state periods. The second reward is

$$r_f = 1 - \text{clip}\left(|F - F_{\text{ref}}/F_{\text{ref}}|, 1\right). \tag{8}$$

Target thrust is one of the important indicators of the engine system, which directly reflects the performance and load capacity of the engine system. The third reward is

$$r_{\text{mix}} = 1 - \begin{cases} \frac{\text{MR}_{\text{GG,ref}} - \text{MR}_{\text{GG}}}{\text{MR}_{\text{GG,ref}}}, & if \frac{\text{MR}_{\text{GG}}}{\text{MR}_{\text{GG,ref}}} < 1 \\ 0, & \text{otherwise.} \end{cases} \tag{9}$$

If the mixing ratio of the gas generator is lower than the preset mixing ratio, it will cause the temperature of the gas generator to rise, causing high-temperature gas to impact the gas turbine, which may cause damage to the gas turbine. The next reward is

$$r_{\text{pos}} = \begin{cases} \left(\sum_i \text{Act}_i\right)/3, & \text{if Time} < 4 \\ 1 - \frac{|S_{\text{VGO}}| + |S_{\text{VGF}}| + |S_{\text{VCF}}|}{3}, & \text{otherwise} \end{cases} \tag{10}$$

where $\text{Act}_i \in [\text{Pos}_{\text{VGO}}, \text{Pos}_{\text{VGF}}, \text{Pos}_{\text{VCF}}]$ represents the opening degrees of the three valves, encouraging attempts to open the valves before the system reaches stability; represents the change in valve position between two-time steps before and after the valve, which is used to penalize the reciprocating action of the valve. Through this term, it can suppress the oscillation caused by the agent frequently acting as the valve during the system start-up process. The last reward

$$r_{\text{act}} = \begin{cases} 1, & \text{if } \text{AT}_{\text{VGO}} < \text{AT}_{\text{VGF}} < \text{AT}_{\text{VCF}} \\ 0.5, & \text{if } (\text{AT}_{\text{VGO}} < \text{AT}_{\text{VGF}} \text{ or} \\ & \quad \text{AT}_{\text{VGO}} < \text{AT}_{\text{VCF}} \text{ or} \\ & \quad \text{AT}_{\text{VGF}} < \text{AT}_{\text{VCF}}) \text{ and} \\ & \quad \text{not } (\text{AT}_{\text{VGO}} < \text{AT}_{\text{VGF}} < \text{AT}_{\text{VCF}}). \end{cases} \tag{11}$$

Among them, AT represents the opening time of the valve. This reward is designed based on the characteristics of the LOX/Kerosene high-pressure staged combustion engine system and engine testing experience, ensuring that the oxidizer valve opens earlier than the fuel valve and the gas generator fuel valve opens earlier than the main combustion chamber fuel valve. This condition-triggered reward is expected to enable the agent to use the "discontinuous" state of the engine during the training process, ensuring that VGO, VGF, and VCF can be opened in the correct order.

As previously mentioned, the design of the reward function is crucial in balancing multiple control objectives, including ensuring that valves open in the correct sequence, rapidly achieving target operating conditions, minimizing steady-state error, reducing system overshoot, and decreasing valve oscillations. These control objectives are reflected in the various components of the reward function, each targeting specific performance indicators.

Algorithm 1: Intrinsically Motivated TD3 (IM-TD3).

Initialize critic networks Q_{θ_1}, Q_{θ_2}, and actor network π_ϕ with random parameters θ_1, θ_2, ϕ.
Initialize target networks $\theta_1' \leftarrow \theta_1$, $\theta_2' \leftarrow \theta_2$, $\phi' \leftarrow \phi$.
Initialize replay buffer $\mathcal{B}$ and learning rate schedulers.
for $t = 1$ to T **do**
  for $i = 1$ to 10 **do** ▷ Perform ten iterative training updates
    Sample replay buffer to get transitions (s, a, r, s', d).
    Disable gradient calculations for target updates:
      Compute target action $a' = \pi_{\phi'}(s') + \text{clip}(\mathcal{N}(0, \sigma), -c, c)$.
      Compute target Q values $Q' = \min(Q_{\theta_1'}(s', a'), Q_{\theta_2'}(s', a'))$.
      Compute target $Q_{\text{target}} = r + (1 - d) \cdot \gamma \cdot Q'$.
    Update critic networks using mean square error (MSE) loss: $\text{MSE}(Q_\theta(s, a), Q_{\text{target}})$.
    if i mod policy_freq $= 0$ **then**
      Update actor network by maximizing critic's Q values.
      Soft update target networks θ_i' and ϕ' .
    end if
    Adjust learning rates using schedulers.
  end for
end for

The engine control methodology based on IM-TD3 necessitates the training and evaluation of the agent. During the training phase, the agent generates different actions through its policy and receives rewards through interaction with the environment. These rewards are used to adjust the policy, aiming to identify the optimal control strategy. During the evaluation phase, the policy remains constant while the performance of the agent is assessed under varying conditions.

In our study, random noise was incorporated into the policy for action generation during the training process to ensure the agent learned appropriate control strategies. The design conditions considered were thrusts of 1200 and 920 kN. In the evaluation phase, we calculated the values of rewards under different control logics and assessed the strategic performance at both design conditions. The control actions related to the state were deterministic, with exploration noise set at 0.001. For further detailed discussions on continuous control problems in RL, Riedmiller has extensively elaborated in literature [35].

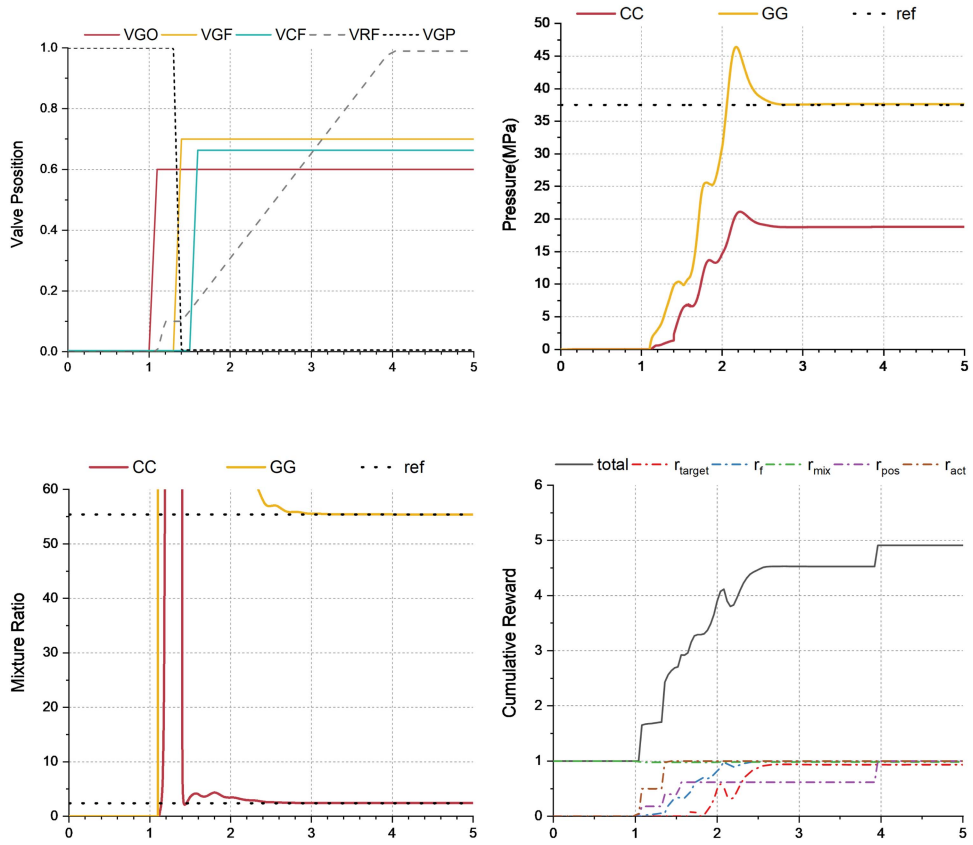

Fig. 2. 1200-kN open-loop start timing, parameter changes, and reward acquisition.

IV. INTELLIGENT AGENT TRAINING AND SYSTEM SIMULATION

In this article, we utilize Simulink for modeling and simulation. Simulink offers robust tools and logic libraries that facilitate the simulation of systems ranging from simple dynamics to complex control systems. As depicted in Fig. 1, an engine system was established using Simulink.

The start-up of the engine involves complex discontinuous processes, such as cold filling, exhaust, and ignition. In this study, the filling process of the engine was ignored, and it was assumed that the propellant had already been filled to the main valves and immediately entered the gas generator and combustion chamber when the valves opened.

When the system starts working, first open the liquid oxygen main valve in front of the gas generator. Under the pressure of the storage tank, liquid oxygen enters the gas generator. Subsequently, the fuel in the starting box compresses the ignition agent in the ignition duct and enters the gas generator, igniting and burning with the preentered liquid oxygen. The fuel in the starting box is controlled by the gas generator valve to enter and participate in combustion at a small flow rate. This article simplifies the simulation of this process by controlling the ON/OFFstate of the liquid oxygen main valve and the fuel valve of the gas generator. When the engine enters the primary process, the fuel valve in the combustion chamber opens, and the throttle valve is in a low flow state. For the simulation of this process, this article uses the ON/OFFstate of the flow control valve to achieve this. At the beginning of ignition, the fuel flow entering the gas generator is the starting flow rate. The mixing ratio of the gas generator rapidly decreases, and the temperature of the oxygen-rich gas increases, driving the main turbine to rotate. Therefore, the time sequence of valve opening and the size of valve opening in a gas engine determine the thermodynamic conditions and mechanical stresses experienced by the components during the engine starting process. Incorrect starting sequence may cause difficulty in starting the engine, or even damage the engine.

Fig. 2 shows the open-loop starting sequence obtained from the experiment. First, the VGP valve is in an open state from the beginning, then the VGO valve opens at $t = 1.1$ s, and second, the VGF valve opens at $t = 1.4$ s. Usually, the fuel will ignite smoothly at $t = 1.5$ s. To ensure the smooth entry of fuel into the main combustion chamber, it is necessary to open the VCF and close the VGP to ensure sufficient power output from the turbine. Therefore, starting at $t = 1.4$ s, VGP is closed, and all the gas from the gas generator is used for turbine work. Pressure is established in front of the VCF valve. When $t = 1.6$ s, open VCF, and the main combustion chamber begins to burn. During the entire start-up process, VGP is a flow-regulating valve that increases the fuel of the gas generator to maintain combustion stability. The engine reaches a steady state after approximately 4 s. In Fig. 2, the valve switch settings were adjusted to achieve the main combustion chamber pressure of approximately 18 MPa and the gas generator chamber

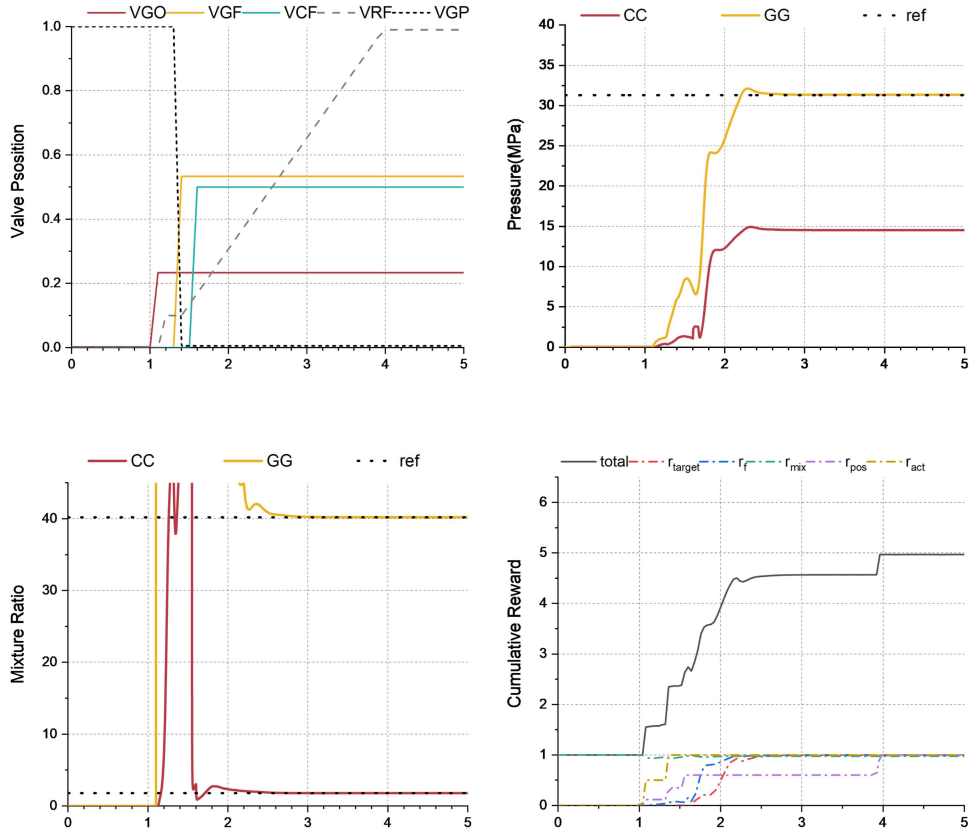

Fig. 3. 920-kN open-loop start timing, parameter changes, and reward acquisition.

pressure of approximately 37 MPa. The mixture ratios were approximately 55.5 for the gas generator and approximately 2.4 for the main combustion chamber. From the results shown in Fig. 2, we can observe that the overshoot in the gas generator is greater than in the combustion chamber. This is because the gas generator operates at a higher pressure and has a smaller volume compared to the combustion chamber, making it more sensitive to pressure fluctuations. In addition, since the gas generator operates under oxygen-rich combustion, its mixture ratio decreases over time. It is also evident from the reward function that the mixture ratio of the gas generator consistently remains equal to or greater than the target value, which causes the $r_{\text{mix}} = 1$ shortly after the engine starts the process.

In addition, Fig. 2 displays the distribution of cumulative rewards for open-loop control under 1200-kN operating conditions, comprising five distinct reward components. In the open-loop control strategy, because the valves are activated only once, the reward components r_{pos} and r_{act} contribute significantly to the total reward. This outcome highlights the importance of optimizing valve operations in the design of the reward function and underscores the need for meticulous adjustments during the actual control process.

We examined various thrust reference values, specifically 1200 and 920 kN. At the thrust of 920 kN, the main combustion chamber pressure was approximately 14 MPa, and the gas generator chamber pressure was about 31 MPa. The mixture ratio for the gas generator was around 40, while that for the main combustion chamber was approximately 1.8. The timing sequence of the valve activation remained unchanged; however, the degree of valve opening was altered, as depicted in Fig. 3. As shown in the results in Fig. 3, compared to the 1200-kN operating condition, the 920-kN condition exhibits smaller overshoot in both the gas generator and the main combustion chamber, but the system's response time is longer. However, the total reward trends of the two operating conditions are relatively similar. In the following analysis, we simulated and assessed the dynamic characteristic curves of an open-loop start-up sequence, a set of PID controllers, the TD3 algorithm, and the IM-TD3 algorithm.

V. SIMULATION RESULT AND DISCUSSION ANALYSIS

In this section, based on the simulation platform, we evaluated the performance of RL in engine control using steady-state error and cumulative rewards. In addition, we compared our results with the TD3 algorithm-based RL method used by Waxenegger et al. [30] and the DDPG algorithm-based RL method. However, the use of the TD3 algorithm in this engine model did not yield the desired outcomes, resulting only in curves of parameter changes associated with unsuccessful start-ups. We first presented the parameter variation trends of open-loop control, PID control, and RL control under 1200-kN operating conditions, as shown in Fig. 4.

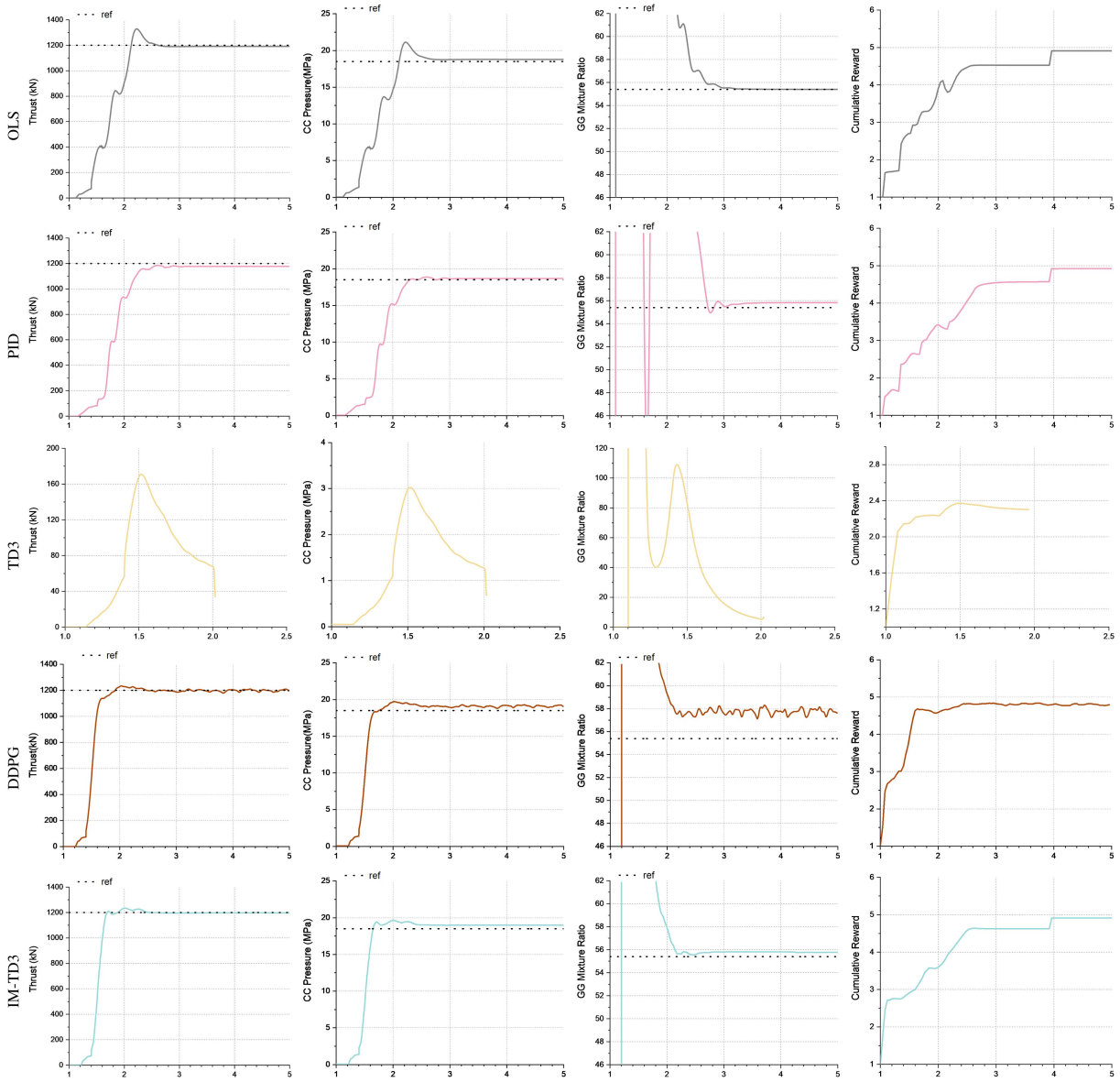

Fig. 4. Under the 1200-kN condition, the trends in parameter changes and the rewards obtained through Open-loop control, PID control, the TD3 method, the DDPG method and IM-TD3 method.

From Fig. 4, the first row shows the results of open-loop control during engine start-up. In this mode, the steady-state values are close to the design conditions, but there is significant parameter fluctuation at start-up, characterized by noticeable oscillations and overshoots. This could potentially impact the entire engine system, reducing the engine's lifespan. For the 920-kN condition, the mixture ratio in the gas generator under open-loop control is below the values required by the design conditions. This could lead to overheating of the gas generator, and the high-temperature gases produced might damage the turbine.

In closed-loop control, parameter fluctuations are significantly reduced, and the engine's overshoot is effectively minimized. We utilized three PID controllers to regulate the opening of the VGO, VGF, and VCF, thereby controlling the combustion chamber pressure, gas generator mixture ratio, and thrust. These adjustments in closed-loop control contribute to more stable and reliable engine operation, aligning with the operational requirements and enhancing overall performance and safety. The variables in the system are coupled, for example, changing VGO will have an impact on the other two variables. Therefore, to avoid system oscillation, we adopted three independent PID controllers for control. From Fig. 4, it can be seen that in PID control, the overshoot and fluctuation of system parameters are smaller than those in open-loop control, but it will increase the time for the system to reach a steady state. Due to the signal composition of the PID controller, the output shape has certain limitations and cannot provide the optimal control signal. Therefore, there is a deviation between the mixing ratio of thrust and gas generator and the design value. During the start-up process, there may be fluctuations in the mixing ratio in the gas generator, which may cause instantaneous temperature peaks in the gas generator. Although there is

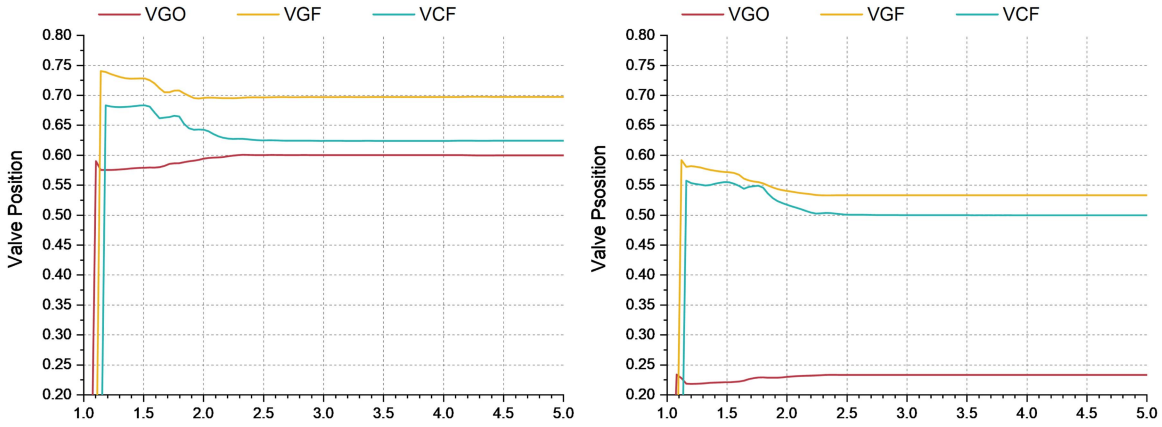

Fig. 5. Valve adjustment using IM-TD3 RL method under 1200-kN (left) and 920-kN (right) operating conditions.

a deviation between the mixing ratio and the design value in PID control, in steady-state, this value is greater than the expected mixing ratio, the fuel ratio of the gas generator is lower than the expected value, and the gas generator will not overheat. Moreover, compared with the other two types of control, PID control achieved the lowest reward under 920-kN operating conditions.

The third row represents the parameter changes under the RL method based on the TD3 algorithm. Despite trying various parameter configurations, the TD3 algorithm failed to successfully start the engine, possibly due to its limited ability to handle the complex dynamics within the system. From the attempted startup processes, we observed significant fluctuations in parameters, such as thrust, combustion chamber pressure, and mixture ratio, indicating that the TD3 algorithm lacks control stability and response efficiency in this system.

The fourth row shows the parameter variations based on the DDPG algorithm for the RL control method. The results indicate that the DDPG algorithm exhibits fluctuations during control, with a significant deviation between the gas generator's mixture ratio and the target value. However, it achieves a relatively high total reward in the initial stage. However, under the 920-kN operating condition, the DDPG algorithm causes significant oscillations in the gas generator's mixture ratio, resulting in a lower cumulative reward compared to the IM-TD3 algorithm.

The fifth row shows the starting parameters of the IM-TD3 RL method, and the adjustment curves of VGO, VGF, and VCF under 1200- and 920-kN conditions are shown in Fig. 5, respectively. Compared with Figs. 2 and 3, the valve does not open monotonically but has a certain degree of variation. It is precisely because of such changes that IM-TD3 can start faster than PID. As shown in Fig. 4, it can effectively control the pressure in the combustion chamber and engine thrust. In addition, the IM-TD3 directly considers the issue of overheating in the gas generator, ensuring that the mixing ratio of the gas generator is within the design target range. Similar to PID control, the IM-TD3 effectively reduces parameter fluctuations during the start-up process and achieves values closer to the design specifications. The IM-TD3 adjusts the valves based on the relationship between the current valve status and the target values, as the valve states are incorporated into the state space during the IM-TD3 training process. In Figs. 4 and 6, during the IM-TD3 adjustment, there were some overshoots in the pressure and thrust of the main combustion chamber. The reason for this phenomenon was that the IM-TD3 aimed to make the engine reach a steady state faster in order to pursue greater cumulative rewards. Therefore, a balance was made between the speed of the overshoot and reaching a steady state. From the characteristics of this strategy, we can analyze that IM-TD3 RL has the potential to dynamically adjust actions based on real-time data under different conditions, optimizing engine performance.

To quantitatively assess the efficacy of this approach, Table I presents the steady-state values, cumulative rewards, and total relative errors obtained by open-loop control, PID control, DDPG, and IM-TD3 under 1200- and 920-kN operating conditions.

From Table I, it can be seen that the cumulative relative error in open-loop control is the smallest, indicating that the deviation of each parameter in open-loop control is within an acceptable range. However, for cumulative rewards, the DDPG and the IM-TD3 are higher than open-loop control, indicating that open-loop control can still be improved. It is not difficult to see from Figs. 4 and 6 that there are significant overshoots and fluctuations in the starting process of open-loop control. In reality, if the valve can open nonlinearly, then the engine can achieve a fast and smooth flexible starting. In engineering practice, it is unlikely to obtain the optimal flexible starting scheme through repeated experiments. Therefore, it is necessary to choose open-loop starting with some acceptable deviations, rather than spending a lot of time and money searching for the optimal valve starting sequence. But for reusable engines, as the engine is used, its components will wear out, its performance will gradually deteriorate, and other disturbances in space may cause deviations from the original open-loop control target operating conditions. Therefore, closed-loop control is needed to provide feedback and eliminate these interferences.

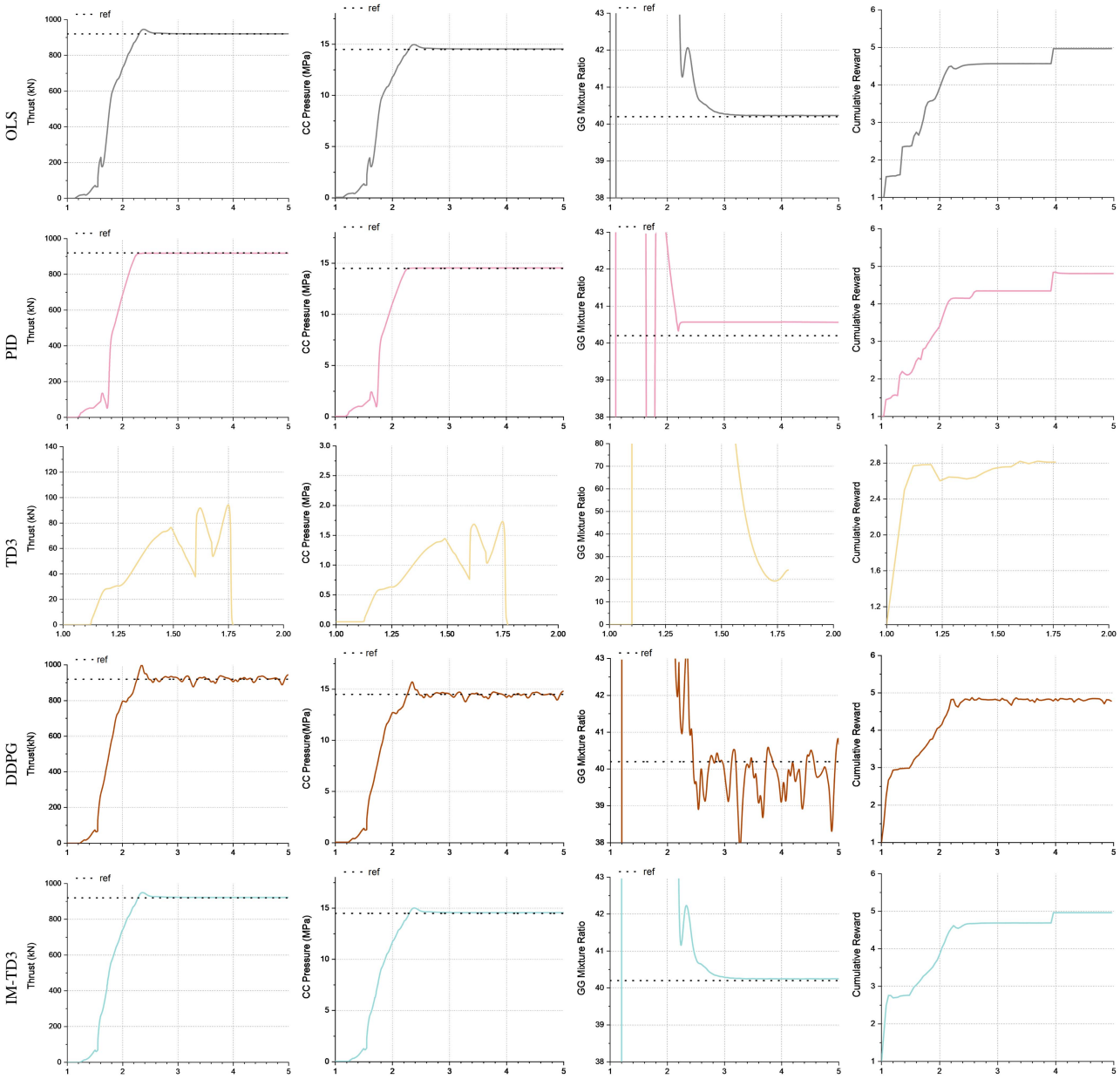

Fig. 6. Under the 920-kN condition, the trends in parameter changes and the rewards obtained through open-loop control, PID control, the TD3 method, the DDPG method, and IM-TD3 method.

TABLE I
Performance Summary of Control Algorithms Under Different Target Thrusts

Target Thrust (kN)	Algorithm	Reward	Steady-state Values			Relative Error
			Thrust(kN)	P_{cc}	MR_{GG}	
1200	OLS	431.49	1191.26	18.79	**55.41**	**0.0514**
	PID	420.06	1177.17	**18.68**	55.83	0.0646
	DDPG	**456.45**	1195.6	19.05	57.63	0.1023
	IM-TD3	444.67	**1197.09**	18.99	55.77	0.0641
920	OLS	435.77	**920.88**	14.54	**40.23**	**0.0045**
	PID	413.28	917.84	**14.53**	40.56	0.0134
	DDPG	451.73	930.42	14.80	40.67	000437
	IM-TD3	**452.45**	922.24	14.57	40.25	0.0085

* Relative Error $= \sum_{\varepsilon_i} \left(\left| \frac{\varepsilon_i - \varepsilon_{i,\mathrm{ref}}}{\varepsilon_{i,\mathrm{ref}}} \right| \right)$

Boldface is the closest thing to a desirable result.

VI. CONCLUSION

This article proposes an RL control method for transient regulation during the start-up process of an LOX/Kerosene high-pressure staged combustion engine. Utilizing the engine simulation platform, the IM-TD3 algorithm was employed to learn the optimal strategy. The start-up process under various operating conditions was examined and compared with the open-loop control and PID control used in the experiment. The results indicate that the IM-TD3 achieves the maximum reward by minimizing parameter fluctuations within an acceptable steady-state error range and allows for control through real-time interaction. However, some aspects of this research require further improvement in future studies.

This article analyzes the transient control of the starting process. Not considering engine performance degradation or other interfering factors is crucial for repeatable engines. Therefore, in future work, it is necessary to study state estimation and interference suppression. On the other hand, complex models may also generate prediction errors due to factors not taken into account or model errors. Therefore, it is necessary for the controller to have sufficient robustness.

The generalizability of the controller is also a critical consideration. When designing control strategies for rocket engines, it is crucial to ensure that the controller possesses robust generalizability. This requires the controller to function effectively across a variety of operating environments and to adapt to various potential system changes. This necessitates a careful distinction in the reward function between constraints that affect engine safety and those that impact performance.

Specifically, constraints that affect safety can be considered as hard constraints, while those impacting performance can be viewed as soft constraints. Hard constraints primarily focus on the safe operation of the engine; violating these constraints could lead to equipment damage or system failure. To reinforce such constraints, negative rewards can be designed to immediately terminate unsafe control strategies. For example, if the temperature of the gas generator exceeds safe thresholds, potentially damaging the turbine, a rule could be set to issue a significant negative reward and immediately stop the strategy if the mixture ratio falls below a specific safety threshold. Soft constraints, on the other hand, are generally associated with engine operational efficiency and performance optimization. Although violating soft constraints does not immediately endanger system safety, it can lead to suboptimal performance or reduced efficiency. In these cases, smaller negative rewards can effectively guide the control strategy to meet performance requirements while avoiding unnecessary operational extremes. For instance, if the engine cannot quickly reach a predetermined operational state during start-up, a moderate negative reward could be designed to encourage optimization of the start-up process without needing to halt training entirely.

By incorporating RL algorithms capable of offline learning, we can enhance model training speed and deploy the trained models from offline to online environments. The primary reason for the lengthy training time in RL is its reliance on online interaction with the environment. In practical engineering, extensive ground tests for training are not feasible. However, using test run data or offline data to train the model can not only improve the efficiency of RL but also make it more suitable for applications like engines, where data are scarce.

Particularly for reusable rocket engines, RL can refine its decision-making process by utilizing historical flight data and real-time feedback, thus enhancing operational efficiency and extending the engine's lifespan. By implementing such an intelligent control system, we can progress toward more efficient, economical, and sustainable space launch services.

ACKNOWLEDGMENT

The authors would like to thank all the staff and fellow researchers for providing technical and academic support.

REFERENCES

[1] P. Jin, Z. Chen, R. Li, Y. Li, and G. Cai, "Opportunistic preventive maintenance scheduling for multi-unit reusable rocket engine system based on the variable maintenance task window method," *Aerosp. Sci. Technol.*, vol. 121, 2022, Art. no. 107346.

[2] Y. Jin, X. Xu, Q. Yang, and S. Zhu, "Numerical investigation of flame appearance and heat flux and in a deep-throttling variable thrust rocket engine," *Aerosp. Sci. Technol.*, vol. 88, pp. 457–467, 2019.

[3] R. W. Cooper, D. J. Martin, N. Wunderlin, J. Pitot, and M. J. Brooks, "Facility design and cold flow testing operations for an 18kN LOX/kerosene liquid rocket engine," in *Proc. AIAA SCITECH 2024 Forum*, 2024, Art. no. 1399.

[4] Y. Liu, Y. Cheng, and J. Wu, "Research progress of intelligent control methods in space propulsion systems," *ACTA Aeronauticaet Astronautica Sinica*, vol. 44, no. 15, 2023, Art. no. 528505.

[5] K. Dresia et al., "Multidisciplinary design optimization of reusable launch vehicles for different propellants and objectives," *J. Spacecraft Rockets*, vol. 58, no. 4, pp. 1017–1029, 2021.

[6] A. Nebylov and V. Nebylov, "Reusable space planes challenges and control problems," *IFAC-PapersOnLine*, vol. 49, no. 17, pp. 480–485, 2016.

[7] S. Pérez-Roca et al., "A survey of automatic control methods for liquid-propellant rocket engines," *Prog. Aerosp. Sci.*, vol. 107, pp. 63–84, 2019.

[8] S. Jiawen and S. Bing, "Thermal-structural analysis of regeneratively-cooled thrust chamber wall in reusable lox/methane rocket engines," *Chin. J. Aeronaut.*, vol. 30, no. 3, pp. 1043–1053, 2017.

[9] W. Bauer et al., "DLR reusability flight experiment refex," *Acta Astronautica*, vol. 168, pp. 57–68, 2020.

[10] T. Chen, J. Li, P. Jin, and G. Cai, "Reusable rocket engine preventive maintenance scheduling using genetic algorithm," *Rel. Eng. Syst. Saf.*, vol. 114, pp. 52–60, 2013.

[11] K. Lai, F. K. Leung, B. Tao, and S. Wang, "Practices of preventive maintenance and replacement for engines: A case study," *Eur. J. Oper. Res.*, vol. 124, no. 2, pp. 294–306, 2000.

[12] S. Pérez-Roca et al., "An MPC approach to transient control of liquid-propellant rocket engines," *IFAC-PapersOnLine*, vol. 52, no. 12, pp. 268–273, 2019.

[13] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Netw.*, vol. 61, pp. 85–117, 2015.

[14] Q. Wei, Z. Yang, H. Su, and L. Wang, "Monte Carlo-based reinforcement learning control for unmanned aerial vehicle systems," *Neurocomputing*, vol. 507, pp. 282–291, 2022.

[15] J. Yang, H. Zhang, H. Wang, X. Li, and C. Wang, "Heterogeneous unmanned swarm formation containment control based on reinforcement learning," *Aerosp. Sci. Technol.*, vol. 150, 2024, Art. no. 109186.

[16] G. Falcone and Z. R. Putnam, "Autonomous decision-making for aerobraking via parallel randomized deep reinforcement learning," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 59, no. 3, pp. 3055–3070, Jun. 2023.
[17] R. Furfaro, A. Scorsoglio, R. Linares, and M. Massari, "Adaptive generalized Zem-Zev feedback guidance for planetary landing via a deep reinforcement learning approach," *Acta Astronautica*, vol. 171, pp. 156–171, 2020.
[18] D. Miller, J. A. Englander, and R. Linares, "Interplanetary low-thrust design using proximal policy optimization," in *Proc. AAS/AIAA Astrodyn. Specialist Conf.*, 2019, Art. no. AAS 19-779.
[19] H. Dong, X. Zhao, and H. Yang, "Reinforcement learning-based approximate optimal control for attitude reorientation under state constraints," *IEEE Trans. Control Syst. Technol.*, vol. 29, no. 4, pp. 1664–1673, Jul. 2021.
[20] J. G. Elkins, R. Sood, and C. Rumpf, "Autonomous spacecraft attitude control using deep reinforcement learning," in *Proc. Int. Astronautical Congr.*, 2020, pp. 1–13.
[21] G. Falcone and Z. R. Putnam, "Deep reinforcement learning for autonomous aerobraking maneuver planning," in *Proc. AIAA SCITECH 2022 Forum*, 2022, Art. no. 2497.
[22] M. Tipaldi, R. Iervolino, and P. R. Massenio, "Reinforcement learning in spacecraft control applications: Advances, prospects, and challenges," *Annu. Rev. Control*, vol. 54, pp. 1–23, 2022.
[23] B. A. Maicke, "A survey of machine learning in rocket propulsion applications," in *Proc. AIAA Aviation 2023 Forum*, 2023, Art. no. 4033.
[24] K. Dresia, G. Waxenegger-Wilfing, R. H. Dos Santos Hahn, J. C. Deeken, and M. Oschwald, "Nonlinear control of an expander-bleed rocket engine using reinforcement learning," in *Proc. Int. Space Propulsion Conf.*, 2021, pp. 1–10.
[25] A. Zavoli, P. M. Zolla, L. Federici, M. T. Migliorino, and D. Bianchi, "Machine learning techniques for flight performance prediction of hybrid rocket engines," in *Proc. AIAA Propulsion Energy Forum*, 2021, Art. no. 3506.
[26] C. E. Oestreich, R. Linares, and R. Gondhalekar, "Autonomous six-degree-of-freedom spacecraft docking with rotating targets via reinforcement learning," *J. Aerosp. Inf. Syst.*, vol. 18, no. 7, pp. 417–428, 2021.
[27] A. Brandonisio, L. Capra, and M. Lavagna, "Spacecraft adaptive deep reinforcement learning guidance with input state uncertainties in relative motion scenario," in *Proc. AIAA Scitech 2023 Forum*, 2023, Art. no. 1439.
[28] G. Ciabatti, S. Daftry, and R. Capobianco, "Learning transferable policies for autonomous planetary landing via deep reinforcement learning," in *Proc. Accelerating Space Commerce, Exploration, New Discov. Conf.*, 2021, Art. no. 4006.
[29] T. Hörger, K. Dresia, G. Waxenegger-Wilfing, L. K. Werling, and S. Schlechtriem, "Preliminary investigation of robust reinforcement learning for control of an existing green propellant thruster," in *Proc. AIAA Propulsion Energy Forum*, 2021, Art. no. 3223.
[30] T. Hörger, L. Werling, K. Dresia, G. Waxenegger-Wilfing, and S. Schlechtriem, "Experimental and simulative evaluation of a reinforcement learning based cold gas thrust chamber pressure controller," *Acta Astronautica*, vol. 219, pp. 128–137, 2024.
[31] R. Amarthya, A. Agrawal, Y. Chawla, and A. K. Mishra, "Autonomous thrust vector control using machine learning in physics simulations: Enhancing performance in guidance," in *Proc. Int. Conf. Power Eng. Intell. Syst.*, 2023, pp. 459–472.
[32] G. Waxenegger-Wilfing, K. Dresia, J. Deeken, and M. Oschwald, "A reinforcement learning approach for transient control of liquid rocket engines," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 57, no. 5, pp. 2938–2952, Oct. 2021.
[33] R. S. Sutton and A. G. Barto, *Barto Reinforcement Learning: An Introduction*. Cambridge, MA, USA: MIT Press, 2018.
[34] V. R. Konda and J. N. Tsitsiklis, "On actor-critic algorithms," *SIAM J. Control Optim.*, vol. 42, no. 4, pp. 1143–1166, 2003.
[35] M. Riedmiller, "10 steps and some tricks to set up neural reinforcement controllers," in *Neural Networks: Tricks of the Trade*, 2nd ed. Berlin, Germany: Springer, 2012, pp. 735–757.

Yuwei Liu received the B.E. degree in aircraft propulsion engineering from the College of Aviation Engineering, Civil Aviation Flight University of China, Deyang, China, in 2019, and the M.E. degree in aerospace engineering from the College of Aerospace Science and Engineering, National University of Defense Technology, Changsha, China, in 2021. He is currently working toward the Ph.D. degree in aerospace science and technology with the National University of Defense Technology, Changsha, China.

His research interests include propulsion system dynamics and control.

Yang Li received the bachelor's degree in digital media technology from School of Software, the Dalian University of Technology, Dalian, China, in 2019, and the master's degree in information and communication engineering from the University of Chinese Academy of Sciences, Beijing, China, in 2022.

He has authored or coauthored publications in prestigious conferences and journals, including International Conference on Machine Learning, International Conference on Learning Representations, Annual Meeting of the Association for Computational Linguistics, *Transactions on Machine Learning Research*, IEEE TRANSACTIONS ON MOBILE COMPUTING, IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, and so on. His research interests include multiagent systems, human-AI/Robot collaboration, and application of AI.

Yuqiang Cheng received the M.S. and Ph.D. degrees in aerospace science and technology from the National University of Defense Technology, Changsha, China, in 2004 and 2009, respectively.

He is currently a Professor with the College of Aerospace Science and Engineering, National University of Defense Technology, Changsha, China. His research interests include advanced space power technology, liquid propellant rocket engine system dynamics and health monitoring, reusable rocket engine fife assessment.

Wei Pan (Member, IEEE) received the Ph.D. degree in bioengineering from Imperial College London, London, U.K., in 2016.

Until 2018, he was a Project Leader with DJI, Shenzhen, China, responsible for machine learning research for DJI drones and AI accelerator. He is currently an Assistant Professor with the Delft University of Technology, Delft, Netherlands, and the University of Manchester, Manchester, U.K. His research interests include machine learning and control theory with applications in robotics.

Jianjun Wu received the Ph.D. degree in aerospace propulsion technology from the National University of Defense Technology, Changsha, China.

He is currently a Professor with the College of Aerospace Science and Engineering, National University of Defense Technology. His research interests include liquid propellant rocket engine propulsion technology and space electric propulsion technology.