

Intelligent Full-Envelope Optimization of Transient Control Laws for Adaptive Cycle Engines via Transfer Learning–Enhanced Deep Reinforcement Learning

Ming Qiao[1], Yingzhi Huang[2,*], Linfeng Gou[1]

[1]Northwestern Polytechnical University, Xi'an 710072, China
[2]Shenzhen Research Institute of Northwestern Polytechnical University, Shenzhen 518057, China

Abstract

Adaptive cycle engines can adjust bypass ratio and operating modes to achieve low fuel consumption during subsonic cruise and high thrust in supersonic flight. However, the optimization of transient control laws remains highly challenging due to strong aerothermodynamic coupling, pronounced nonlinear transient behavior, and coordinated control of multiple actuators. To address these challenges, this paper first employs the deep deterministic policy gradient (DDPG) algorithm to optimize the transient control laws at multiple typical operating points. Considering the high training cost of deep reinforcement learning and the inherent limitations of conventional similarity and isothermal scaling methods in full-envelope control law extension, a transfer-learning-based closed-loop extension framework is further proposed. The optimized control policies obtained at individual operating points are reused as the prior knowledge by sharing their neural network parameters, enabling efficient transfer to similar target conditions. Meanwhile, spectral clustering is employed to partition the engine flight envelope into characteristic subregions, providing a structured guidance for the transfer learning (TL) and mitigating error accumulation across the envelope. Simulation results demonstrate that the proposed TL-DDPG approach can effectively coordinate six control variables to accomplish wide-range acceleration of a three-bypass adaptive cycle engine over a wide flight envelope, while satisfying the turbine inlet temperature and the compressor surge margin constraints. Compared with conventional DDPG, the proposed method reduces the training episodes by more than 60% and exhibits strong robustness and generalization capability under cross-region operating conditions. The results indicate that the proposed framework provides an efficient and reliable closed-loop solution for full-envelope transient control law extension of adaptive cycle engines.

Keywords: Adaptive cycle engine; Deep reinforcement learning; Transfer learning; Transient optimization; Full-flight-envelope extension

1. Introduction

The adaptive cycle engine (ACE) can achieve a wider range of performance matching among different flight missions by changing bypass flow paths, equivalent bypass ratio, and multiple variable-geometry components [1]. Compared with conventional turbofan engines, ACE can switch between high-thrust and low-fuel-consumption modes according to mission requirements. During subsonic cruise, propulsion efficiency can be improved by increasing the bypass flow rate and equivalent bypass ratio; during supersonic dash or high-maneuvering flight, high thrust demand can be satisfied by reducing the equivalent bypass ratio and increasing the specific thrust [2]. Therefore, the optimization of control schedules not only determines whether the steady-state performance potential of the engine can be fully exploited, but also directly affects its dynamic response capability and safe operating boundaries under multi-mission profiles.

The ACE control object is characterized by multiple operating modes, multiple variable actuators, and strongly coupled constraints [3]. For a three-bypass ACE, variables such as the main combustor fuel flow, nozzle area, FLADE branch, front and rear variable-area bypass injectors, and variable guide vanes jointly affect fan/compressor matching, bypass flow distribution, turbine inlet

https://doi.org/10.71945/icas2026_0584

temperature, and thrust output. The core task of transient control schedule design is to determine the coordinated time-varying laws of multiple control variables when the engine accelerates or decelerates from one steady-state operating point to another. This process requires not only a shortened regulation time, but also that rotor speeds, turbine inlet temperature, compressor surge margins, and actuator positions always satisfy safety constraints [4]. Therefore, the transient control schedule design of ACE is essentially a sequential optimization problem with multivariable coupling, multiple constraints, and long-horizon dynamic characteristics [5].

However, an actual flight mission does not correspond to only a single acceleration or deceleration process at one operating condition. Instead, it is a continuous-envelope mission composed of different altitudes, Mach numbers, inlet boundary conditions, and thrust demands [6]. Even if an optimal transient control schedule is obtained at a representative operating condition, it remains necessary to solve how this schedule can be extended to neighboring conditions, different flight regions, and multimode operating scenarios [7]. Therefore, ACE transient control schedule optimization should focus not only on dynamic performance at a single operating condition, but also on control schedule extension and rapid generation over the full flight envelope.

Traditional aero-engine control schedule extension usually relies on similarity conversion, isothermal-line conversion, and open-loop gain scheduling methods [8–10]. These methods have good engineering practicability for conventional configurations and local operating ranges, but they still have certain limitations in ACE full-envelope control. On the one hand, similarity conversion is mainly established around similar parameters such as rotor speed, flow rate, temperature, and pressure, making it difficult to fully describe the coordinated regulation among multiple variable-geometry components [11]. On the other hand, this method usually assumes that gas thermodynamic properties change only slightly. When high altitude, high Mach number, or combustion product variation causes significant changes in specific heat at constant pressure and the specific heat ratio, strict thermodynamic similarity cannot be guaranteed even if geometric similarity and kinematic similarity are approximately satisfied. The isothermal-line method can reduce conversion errors within a local region, but interpolation or gain scheduling is still commonly required between adjacent isothermal lines. Under complex flight envelopes and multimode switching conditions, this may easily introduce discontinuities in control schedules or insufficient constraint margins [12]. Therefore, it is necessary to establish a control schedule optimization method that can simultaneously consider transient dynamic performance, safety constraints, and full-envelope extension efficiency.

Deep reinforcement learning (DRL) provides a new approach for the optimization of complex aero-engine transient control schedules [13]. Unlike optimization methods that rely on explicit model gradients or local linearization, DRL can directly learn the mapping from states to control variables through interaction with the simulation environment, making it suitable for sequential decision-making problems with continuous states, continuous actions, and multiple coupled constraints [14]. Deep deterministic policy gradient (DDPG) is an Actor–Critic algorithm for continuous action spaces. It has been widely used in continuous control tasks [15] and has gradually been applied to aero-engine transient control, full-envelope acceleration control, and performance optimization problems [16]. However, if the DDPG policy is trained from scratch at every flight-envelope operating point, the number of training samples and the computational cost will increase rapidly with the number of operating points, making it difficult to meet the engineering efficiency requirements of ACE full-envelope control schedule design [17].

To balance closed-loop control accuracy and full-envelope extension efficiency, this paper proposes a transfer-learning-enhanced deep reinforcement learning method for transient control schedule optimization. First, DDPG is used to train a high-quality transient control policy at representative source operating conditions. Then, the shared parameters of the policy network and value network are transferred to the target operating condition as prior knowledge, and the policy is fine-tuned through a small number of interactions with the target environment. Considering that aerodynamic–thermodynamic boundaries and engine operating states are not completely similar among different flight conditions, spectral clustering is further adopted to partition the flight envelope. Altitude, Mach number, engine inlet total temperature, and engine inlet total pressure are used to characterize operating-condition similarity. Based on the partition results, the source-task selection and transfer

path are determined, thereby reducing the risk of negative transfer caused by blind migration. Transfer learning can reuse the control experience obtained from the source task, improve the learning efficiency of the target task, and reduce the demand for repeated training samples [18]. Spectral clustering can characterize complex data distributions using the similarity structure among samples, making it suitable for partitioning nonlinear operating-condition spaces [19].

The main contributions of this paper are summarized as follows: an MDP formulation and a six-variable DDPG optimization framework are established for ACE transient control; a spectral-clustering-guided TL-DDPG closed-loop extension architecture is constructed for full-envelope control schedule generation; and both intra-cluster and inter-cluster transfer cases are conducted to verify the effectiveness of transfer learning in improving training efficiency and enhancing the generalization capability of control schedules.

2. Adaptive Cycle Engine Model

This paper considers a three-bypass adaptive cycle engine equipped with a fan-on-blade bypass duct engine fan and a core driven fan stage (CDFS) as the research object. The main components of the engine include the inlet, FLADE fan, front fan, mode select valve (MSV), CDFS, front variable-area bypass injector (FVABI), front mixer, high-pressure compressor (HPC), main combustor, high-pressure turbine (HPT), low-pressure turbine (LPT), rear variable-area bypass injector (RVABI), rear mixer, afterburner, main nozzle and FLADE nozzle. The schematic configuration of the engine is shown in Figure 1.

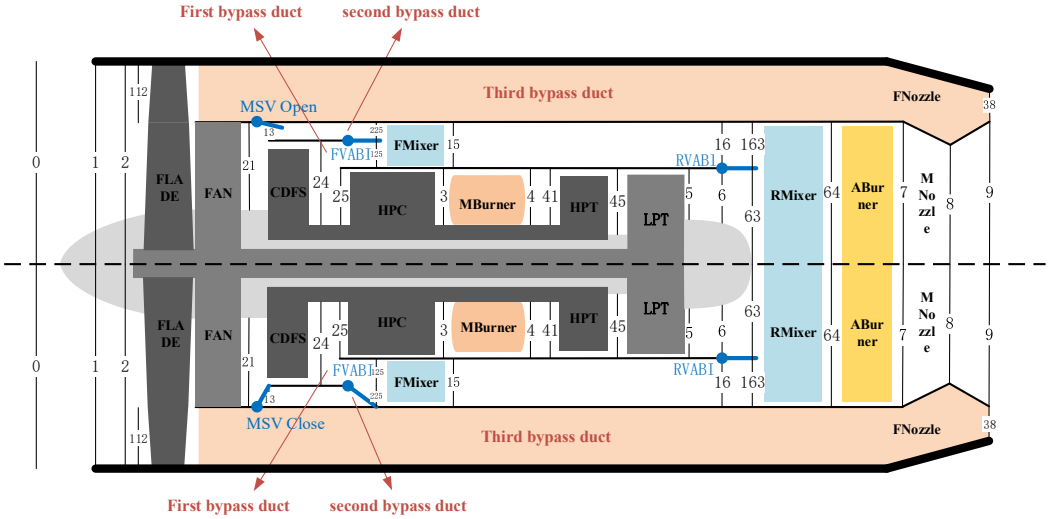


Figure 1 – Schematic of the three-bypass adaptive cycle engine structure and typical stations.

The typical station numbers and their corresponding physical meanings in the three-bypass adaptive cycle engine are summarized in Table 1.

Table 1 – Typical station numbers and their meanings.

Station number	Meaning	Station number	Meaning
0	Ambient environment	1	Inlet entrance
2	Front fan inlet	112	FLADE fan inlet
21	Core fan inlet	13	MSV duct inlet
24	FVABI duct inlet	25	High-pressure compressor inlet
125	Inner inlet of front mixer	225	Outer inlet of front mixer
3	Combustor inlet	4	High-pressure turbine inlet
45	Low-pressure turbine inlet	63/163	Stations related to rear mixer
64	Afterburner outlet	38	FLADE nozzle outlet

According to the states of the mode select valve (MSV) and the FLADE branch, the engine can form different bypass-flow configurations. In the M13 mode, the first and third bypass ducts are open, while the second bypass duct is closed. This mode is suitable for high-thrust and high-specific-thrust requirements. In the M3 mode, all three bypass ducts participate in the flow, and the equivalent bypass ratio increases, making it more suitable for cruise or low-fuel-consumption missions. Under different operating modes, the flow distribution between the core and bypass ducts, rotor power balance, and nozzle matching relationships all vary, which gives the transient control schedule obvious multivariable coupling characteristics. Therefore, this paper focuses on the optimization of the transient control schedule in a typical three-bypass mode.

3. Full-envelope transient control schedule optimization method based on TL-DDPG

Based on the engine model and optimization objective described above, this section proposes an ACE transient control schedule optimization method based on DDPG and transfer learning. First, the single-operating-point transient control schedule optimization problem is formulated as a Markov decision process with continuous states and continuous actions. The state space, action space, reward function and action dynamic update mechanism are designed so that the DDPG agent can learn the time-varying control law under safety constraints and actuator dynamic constraints. Furthermore, to address the large sample demand and high computational cost caused by point-by-point training over the full flight envelope, operating-condition similarity analysis and spectral clustering are introduced to partition the flight envelope. On this basis, a TL-DDPG parameter transfer architecture is constructed to realize the rapid transfer of control experience from the source operating condition to the target operating condition. The proposed method provides a unified framework for single-point optimization and full-envelope extension of ACE transient control schedules.

3.1 Single-operating-point transient control schedule optimization based on DDPG

The core task of transient control schedule optimization for an adaptive cycle engine is to coordinate the time-varying actions of multiple variable actuators under given initial and target steady-state operating points, so that the engine can complete the acceleration process as quickly as possible while satisfying constraints on rotor speed, temperature, surge margin and actuator physical limits. Since this problem involves continuous states, continuous actions, multivariable coupling and long-horizon dynamic characteristics, conventional methods based on fixed acceleration schedules or single-variable control laws cannot fully exploit the cooperative control capability of multiple variable components. Therefore, the acceleration control schedule optimization problem at a single operating condition is first formulated as a Markov decision process, and the deep deterministic policy gradient algorithm is adopted to optimize the closed-loop policy of time-varying control variables.

The engine acceleration process is represented as a Markov decision process, which can be expressed as the following five-tuple:

$$\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, R, \gamma) \tag{1}$$

where $\mathcal{S}$ is the state space, describing the current engine state and safety margins; $\mathcal{A}$ is the action space, corresponding to the continuous control commands issued by the agent at each control interval; P is the engine state-transition law under a given state and action; R is the reward function, which evaluates acceleration performance, safety constraints, and action feasibility; and γ is the discount factor used to balance immediate and future rewards.

DDPG is an Actor-Critic method for continuous action spaces. The Actor network takes the current state s_t as input and outputs the action according to a deterministic policy:

$$a_t = \mu(s_t \mid \theta^{\mu}) \tag{2}$$

The Critic network estimates the state-action value function:

$$Q(s_t, a_t \mid \theta^{Q}) \tag{3}$$

and evaluates the contribution of the current action to the long-term cumulative reward. To improve training stability, DDPG uses four networks: an online Actor, a target Actor, an online Critic, and a target Critic. The target-network parameters are slowly transferred from the online networks through

soft update, namely:

$$\theta^{\mu'} \leftarrow \tau\theta^{\mu} + (1-\tau)\theta^{\mu'} \tag{4}$$

$$\theta^{Q'} \leftarrow \tau\theta^{Q} + (1-\tau)\theta^{Q'} \tag{5}$$

where τ is the soft-update coefficient.

At each interaction step, the Actor generates an action from the engine state and superimposes exploration noise to improve policy search. After the simulation environment executes the action, it returns the next state $s_{t+1}$and the immediate reward r_t, and the transition sample $(s_t, a_t, r_t, s_{t+1})$is stored in the replay buffer. During training, mini-batches are randomly sampled from the replay buffer to reduce temporal correlation among adjacent samples. The Critic target is given by:

$$y_i = r_i + \gamma Q'\left(s_{i+1}, \mu'(s_{i+1} \mid \theta^{\mu'}) \mid \theta^{Q'}\right) \tag{6}$$

and the Critic is updated by minimizing the mean-squared Bellman error:

$$L(\theta^{Q}) = \frac{1}{N}\sum_{i=1}^{N}[y_i - Q(s_i, a_i \mid \theta^{Q})]^2 \tag{7}$$

The Actor is updated according to the value gradient provided by the Critic:

$$\nabla_{\theta^{\mu}} J \approx \frac{1}{N}\sum_{i=1}^{N} \nabla_a Q(s, a \mid \theta^{Q})\mid_{s=s_i, a=\mu(s_i)} \nabla_{\theta^{\mu}} \mu(s \mid \theta^{\mu})\mid_{s=s_i} \tag{8}$$

Consequently, the Actor gradually adjusts its policy toward actions that can obtain higher long-term rewards and ultimately forms a closed-loop transient control policy for a single operating condition. The DDPG-based closed-loop optimization framework for ACE transient control laws is shown in Figure 2.

An ACE contains multiple variable actuators. If all adjustable variables are used as action inputs, the policy search dimension increases substantially and training convergence becomes slower. Considering each actuator's influence on transient response, flow distribution, and nozzle matching, this study selects the main-combustor fuel flow W_{fm}, the main-nozzle throat area A_8, the main-nozzle exit area A_9, the FLADE bypass tail-nozzle exit area A_{38}, the FVABI outer-bypass area scaling coefficient dAs_{FVABI}, and the RVABI outer-bypass area scaling coefficient dAs_{RVABI}as the action variables, namely:

$$a_t = \left[W_{fm}, A_8, A_9, A_{38}, dAs_{FVABI}, dAs_{RVABI}\right]^T \tag{9}$$

where W_{fm} mainly affects turbine work and rotor acceleration capability; A_8 and A_9 affect the main-nozzle expansion matching and the component operating match; A_{38} adjusts the FLADE branch exhaust capacity; and dAs_{FVABI} and dAs_{RVABI} affect the flow distribution and mixing match of the front and rear variable-area bypass injectors, respectively. This six-dimensional action space preserves the dominant control degrees of freedom for ACE transient regulation with a relatively low dimension.

The state space should reflect both the acceleration target and safety boundaries. This study uses fan speed n_L, FLADE fan surge margin SM_{FLADE}, front-fan surge margin SM_{FAN}, CDFS surge margin SM_{CDFS}, high-pressure-compressor surge margin SM_C, and main-combustor outlet temperature $T_{t4}$to construct the state vector:

$$s_t = [n_L, SM_{FLADE}, SM_{FAN}, SM_{CDFS}, SM_C, T_{t4}]^T \tag{10}$$

where n_L describes the current acceleration progress, the four surge margins constrain the stable operating boundaries of the compression components, and T_{t4} limits the hot-section temperature. This state design enables the agent not only to accelerate the engine according to speed error, but also to sense the proximity to each safety boundary, thereby balancing rapid response and safety.

The reward function guides the agent to learn high-performance acceleration policies that satisfy engineering constraints. The single-step reward is designed as:

$$r_t = r_{n,t} + r_{s,t} + r_{u,t} + r_{g,t} \tag{11}$$

where the speed-error reward term is

$$r_{n,t} = -k_n \left(\frac{n^*_{L,t} - n_{L,t}}{n^*_{L,t}} \right)^2 \tag{12}$$

the safety-constraint penalty term is

$$r_{s,t} = -k_{SM} \sum_{j=1}^{4} \left[max\left(0, SM_{min} - SM_{j,t}\right) \right]^2 - k_T \left[max\left(0, T_{t4,t} - T_{t4,max}\right) \right]^2 \tag{13}$$

the action-bound penalty term is

$$r_{u,t} = -k_u \sum_{i=1}^{6} \left[max\left(0, \lambda_{i,t} - \lambda_{max}\right)^2 + max\left(0, \lambda_{min} - \lambda_{i,t}\right)^2 \right] \tag{14}$$

where the normalized action is

$$\lambda_{i,t} = \frac{u_{i,t} - u_{i,min}}{u_{i,max} - u_{i,min}} \tag{15}$$

the target-reaching reward term is

$$r_{g,t} = \begin{cases} R_g, & | n^*_{L,t} - n_{L,t} | \leq \varepsilon_n \\ 0, & | n^*_{L,t} - n_{L,t} | > \varepsilon_n \end{cases} \tag{16}$$

where $n^*_{L,t}$ is the target fan speed, $n_{L,t}$ is the current fan speed; $SM_{j,t}$ denotes the surge margin of the j-th compression component, including the FLADE fan, front fan, CDFS, and high-pressure compressor; $SM_{\min}$ is the lower surge-margin limit; $T_{t4,t}$ is the current turbine-inlet temperature, and $T_{t4,\max}$ is the temperature upper limit; $u_{i,t}$ is the i-th control variable, and $u_{i,\min}$ and $u_{i,\max}$ are its physical lower and upper bounds, respectively; k_n, k_{SM}, k_T, and k_u are the corresponding penalty weights; R_g is the target-reaching reward, and ε_n is the allowable target-speed error.

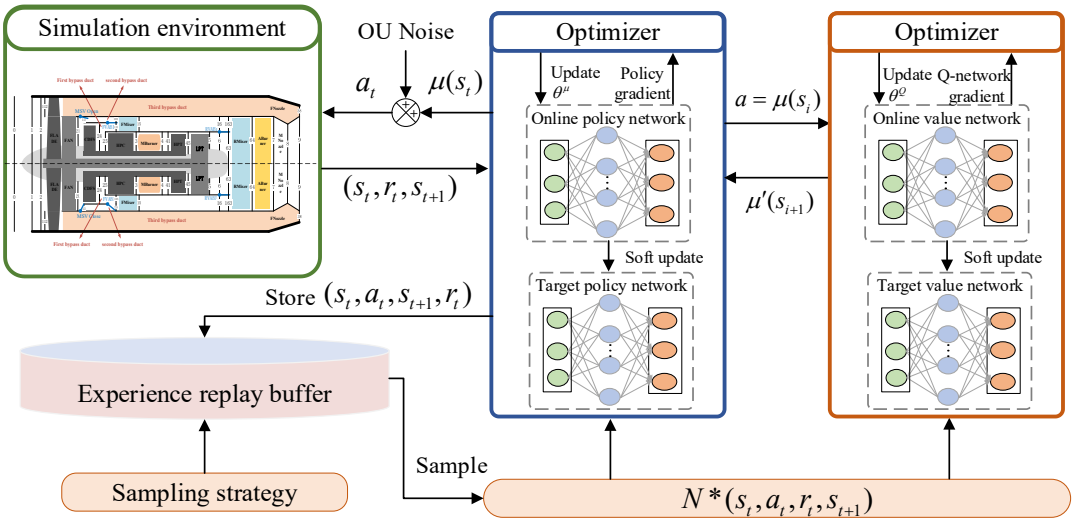


Figure 2 – DDPG-based closed-loop optimization framework for ACE transient control laws.

Table 2 – DDPG action space and bounds.

Action variable	Lower bound	Upper bound	Physical meaning
W_{fm}	1.2 kg/s	2.1 kg/s	Main-combustor fuel flow
A_8	0.20 m²	0.30 m²	Main-nozzle throat area
A_9	0.20 m²	0.30 m²	Main-nozzle exit area
A_{38}	0.040 m²	0.065 m²	FLADE bypass tail-nozzle exit area
dAs_{FVABI}	0.90	1.00	FVABI outer-bypass area scaling coefficient
dAs_{RVABI}	0.90	1.00	RVABI outer-bypass area scaling coefficient

Compared with directly terminating out-of-bound samples, soft penalties retain gradient information near the boundaries, enabling the agent to evaluate states that are close to, but have not yet crossed, safety limits. This is particularly important for an ACE, because an optimal acceleration process often requires the CDFS or other compression components to operate close to the minimum stable-margin boundary. If the penalty is too rigid, the policy tends to become overly conservative; if the penalty is insufficient, inadequate surge margin or overtemperature risk may occur. The selected action variables and their bounds are summarized in Table 2. The state variables and corresponding safety limits used in the DDPG framework are listed in Table 3.

Table 3 – DDPG state space and safety limits.

State variable	Lower bound	Upper bound	Description
n_L	-	105%	Fan percentage speed
SM_{FLADE}	10%	-	FLADE-fan surge margin
SM_{FAN}	10%	-	Front-fan surge margin
SM_{CDFS}	10%	-	Core-driven fan stage surge margin
SM_C	10%	-	High-pressure-compressor surge margin
T_{t4}	-	1850 K	Main-combustor outlet temperature / turbine-inlet temperature limit

3.2 Error Analysis of Similarity-Based Conversion

For geometrically similar gas-turbine engines, or engines of the same configuration, conventional full-envelope control-law extension is usually based on similarity theory. Its basic idea is that, when changes in flight altitude and Mach number change the engine inlet total temperature and total pressure, corrected speed, corrected flow, and corrected control quantities are used to map different operating conditions into a unified similarity-parameter space, and the control law obtained at the reference condition is then extended to the target condition. Let the engine inlet total temperature and total pressure be $T_{t0}$and P_{t0}, respectively. The temperature and pressure correction factors are defined as:

$$\theta = \frac{T_{t0}}{T_{ref}} \tag{17}$$

$$\delta = \frac{P_{t0}}{P_{ref}} \tag{18}$$

where T_{ref} and P_{ref} are the reference temperature and pressure, respectively. For speed and flow, the commonly used corrected relationships are:

$$n_c = \frac{n}{\sqrt{\theta}} \tag{19}$$

$$\dot{m}_c = \dot{m}\frac{\sqrt{\theta}}{\delta} \tag{20}$$

where n_c is corrected speed, $\dot{m}_c$ is corrected flow. If source and target conditions are denoted by superscripts $s$and t, respectively, and are assumed to have the same corrected speed and corrected flow, then speed and flow under the target condition can be obtained from the source condition as:

$$n^t = n^s\sqrt{\frac{\theta^t}{\theta^s}} \tag{21}$$

$$\dot{m}^t = \dot{m}^s\frac{\delta^t}{\delta^s}\sqrt{\frac{\theta^s}{\theta^t}} \tag{22}$$

For fuel flow, conventional methods generally use a correction relationship in the same form as that for airflow:

$$W_{f,c} = W_f\frac{\sqrt{\theta}}{\delta} \tag{23}$$

Therefore, the fuel flow under the target condition can be written as:

$$W_f^t = W_f^s \frac{\delta^t}{\delta^s}\sqrt{\frac{\theta^s}{\theta^t}} \tag{24}$$

For nozzle areas and variable-geometry mechanisms, conventional extension usually does not directly apply temperature-pressure correction; instead, it keeps the normalized opening or relative adjustment consistent. For an adjustable area A_i, its normalized opening is:

$$\lambda_i = \frac{A_i - A_{i,min}}{A_{i,max} - A_{i,min}} \tag{25}$$

If the source and target conditions keep the same normalized opening, then

$$\lambda_i^t = \lambda_i^s \tag{26}$$

For the six control variables selected in this study, the conventional similarity-based extension can be summarized as:

$$W_{fm}^t = W_{fm}^s \frac{\delta^t}{\delta^s}\sqrt{\frac{\theta^s}{\theta^t}} \tag{27}$$

$$\lambda_{A_8}^t = \lambda_{A_8}^s, \lambda_{A_9}^t = \lambda_{A_9}^s, \lambda_{A_{38}}^t = \lambda_{A_{38}}^s \tag{28}$$

$$dAs_{FVABI}^t \approx dAs_{FVABI}^s, dAs_{RVABI}^t \approx dAs_{RVABI}^s \tag{29}$$

where W_{fm} is the main-combustor fuel flow; A_8, A_9 and A_{38} are the main-nozzle throat area, main-nozzle exit area, and FLADE bypass tail-nozzle exit area, respectively; dAs_{FVABI} and dAs_{RVABI} are the area scaling coefficients of the front and rear variable-area bypass injectors, respectively.

The above similarity-based conversion is simple in form and suitable for control-law extension of conventional engine configurations over a local operating range. For an adaptive cycle engine, however, the FLADE branch, the front and rear variable-area bypass injectors, and multiple bypass flow paths change the inner/outer-bypass flow distribution, compression-component matching relationship, and rotor power balance. As a result, identical corrected fuel flow or geometric opening does not necessarily provide identical acceleration capability or safety margin. In addition, variations in altitude and Mach number change the inlet total temperature, inlet total pressure, and component operating points, so the source-condition optimal control law is no longer strictly optimal at the target condition. Therefore, this study decomposes full-envelope extension into two levels: identification of similar operating conditions and closed-loop policy transfer. Spectral clustering is first used to partition operating regions with similar inlet features and flight states; TL-DDPG then transfers the source-condition policy parameters to the target condition and continues closed-loop fine-tuning in the target environment. This avoids both the control-variable mapping errors caused by pure similarity conversion and the high computational cost of retraining at each operating condition.

3.3 Spectral-Clustering-Based Flight Envelope Partitioning

Spectral clustering regards operating points in the flight envelope as nodes of an undirected weighted graph, constructs a weight matrix from node similarities, and uses the eigenvectors of the graph Laplacian to reveal the low-dimensional clustering structure of the data. Compared with direct K-means clustering, spectral clustering is more suitable for the non-convex feature space jointly formed by altitude, Mach number, inlet total temperature, and inlet total pressure.

In this study, operating-condition samples are uniformly generated over the flight-envelope range of altitude $0 \sim 12\mathrm{km}$ and Mach number $0 \sim 1.2$, and the corresponding engine inlet total temperature T_{t0} and inlet total pressure P_{t0} are calculated. The feature vector of the i-th operating point is defined as:

$$x_i = \left[H_i, Ma_i, T_{t0,i}, P_{t0,i}\right]^T \tag{30}$$

where H_i and Ma_i are altitude and Mach number, respectively, and $T_{t0,i}$ and $P_{t0,i}$ are the engine inlet total temperature and total pressure at that condition, respectively. To remove differences in feature dimensions and magnitudes, each feature dimension is standardized using Z-score normalization:

$$\bar{x}_{i,k} = \frac{x_{i,k} - \mu_k}{\sigma_k} \tag{31}$$

where $x_{i,k}$ is the i-th sample's k-th feature, μ_k and σ_k are the mean and standard deviation of the k-th feature, respectively, and $\bar{x}_{i,k}$ is the standardized feature value. The standardized operating-condition feature vector is denoted as $\bar{x}_i$.

A Gaussian kernel function is then used to construct the similarity matrix between operating points:

$$W_{ij} = exp\left(-\frac{\|\bar{x}_i - \bar{x}_j\|^2}{2\sigma^2}\right) \tag{32}$$

where W_{ij} denotes the similarity between the i-th and j-th operating points, and σ is the kernel-width parameter. The degree matrix D is constructed from the similarity matrix W, with diagonal elements

$$D_{ii} = \sum_j W_{ij} \tag{33}$$

The symmetric normalized Laplacian matrix is then obtained as:

$$L_{sym} = I - D^{-\frac{1}{2}} W D^{-\frac{1}{2}} \tag{34}$$

From L_{sym}, select the eigenvectors corresponding to the K smallest eigenvalues to form the spectral-space representation matrix:

$$Z = [v_1, v_2, \cdots, v_K] \tag{35}$$

K-means clustering is then applied to the row vectors of Z to obtain operating-condition partitions in the flight envelope. After clustering, the within-cluster sample mean can be used as the representative operating point of the partition:

$$c_k = \frac{1}{|\Omega_k|} \sum_{x_i \in \Omega_k} x_i \tag{36}$$

where Ω_k denotes the sample set in the k-th cluster, $|\Omega_k|$ is the number of samples in that cluster, and c_k is the corresponding cluster center. The cluster center can serve as the representative condition for subsequent transfer learning and is used to determine source-task selection and the target-condition transfer path.

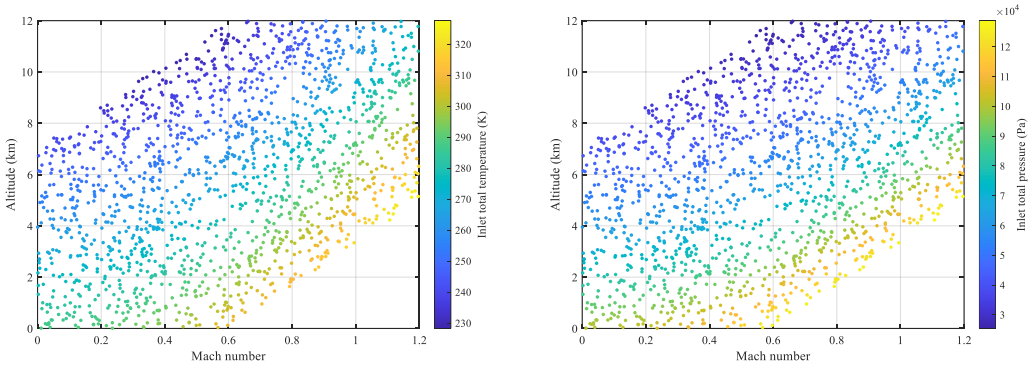


Figure 3 – Flight-envelope sample feature distribution.

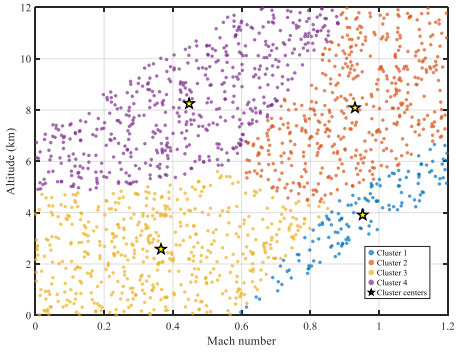


Figure 4 – Spectral-clustering partitioning result.

The flight-envelope sample features are shown in Figure 3, including the sample distribution on the Mach-number-altitude plane and the variations of inlet total temperature and inlet total pressure over the operating conditions. The corresponding spectral-clustering partitioning result is shown in Figure 4. The inlet total temperature mainly increases with Mach number, whereas the inlet total pressure is affected by both altitude and Mach number. Spectral clustering can form relatively continuous regions of similar operating conditions in this nonlinear feature space, providing a basis for subsequent source-task selection. The representative conditions used for transfer validation in this study are listed in Table 4.

Table 4 – Coordinates of spectral-clustering center operating conditions.

Cluster center No.	Mach number Ma	Altitude H/km
1	0.9523	3.9097
2	0.9299	8.0859
3	0.3654	2.5767
4	0.4472	8.2492

3.4 TL-DDPG Parameter Transfer Architecture

The objective of transfer learning is to reuse the control knowledge learned from the source task at the target condition, rather than simply copying the source-condition control trajectories. This study adopts feature-knowledge transfer: the network parameters in the source-task Actor and Critic that extract state, action, and value features are transferred to the target task as the initial parameters of the target networks. The target environment still maintains closed-loop interaction with the agent, so the policy can be fine-tuned according to the new inlet conditions, constraint boundaries, and state responses.

Let the Actor and Critic network parameters trained at the source condition be θ_μ^s and θ_Q^s, respectively, and let the corresponding target-condition network parameters be θ_μ^t and θ_Q^t. The parameter-transfer process can be expressed as:

$$\theta_\mu^t \leftarrow \theta_\mu^s \tag{37}$$

$$\theta_Q^t \leftarrow \theta_Q^s \tag{38}$$

where θ_μ^s contains the source-condition Actor network parameters related to state-feature extraction and action generation, and θ_Q^s contains the source-condition Critic network parameters related to state-action value estimation. After this initialization, the target-condition agent no longer explores from a random policy, but continues learning from the acceleration-control experience formed at the source condition.

For the target networks, their initial parameters are synchronized from the target-condition online networks:

$$\theta_{\mu'}^t \leftarrow \theta_\mu^t \tag{39}$$

$$\theta_{Q'}^t \leftarrow \theta_Q^t \tag{40}$$

where $\theta_{\mu'}^t$ and $\theta_{Q'}^t$ are the target-Actor network and target-Critic network parameters under the target condition, respectively.

The TL-DDPG procedure consists of four stages. First, in the experience-learning stage, DDPG is trained at the source condition to obtain stable Actor/Critic parameters and replay-buffer samples. Second, in the partition-identification stage, the altitude, Mach number, inlet total temperature, and inlet total pressure of the target condition are used to determine its cluster partition, and the policy at the corresponding cluster center is selected as the source task. The target-condition feature vector can be written as:

$$x_t = \left[H_t, Ma_t, T_{t0,t}, P_{t0,t}\right]^T \tag{41}$$

Third, in the parameter-transfer stage, the source network parameters are assigned to the target

networks while the DDPG target-network soft-update structure is retained. Fourth, in the experience-replay and fine-tuning stage, the agent continues sampling in the target environment, updates the Critic using the TD error, and updates the Actor using the value gradient until the target reward or convergence criterion is reached.

During fine-tuning at the target condition, the agent interacts with the target environment and stores transition samples

$$\left(s_i^t, a_i^t, r_i^t, s_{i+1}^t\right) \tag{42}$$

The TD target of the Critic network is:

$$y_i^t = r_i^t + \gamma Q^{t'}\left(s_{i+1}^t, \mu^{t'}(s_{i+1}^t \mid \theta_{\mu'}^t) \mid \theta_{Q'}^t\right) \tag{43}$$

The corresponding Critic loss function is:

$$L(\theta_Q^t) = \frac{1}{N}\sum_{i=1}^{N}\left[y_i^t - Q^t(s_i^t, a_i^t \mid \theta_Q^t)\right]^2 \tag{44}$$

The Actor network is updated according to the action-value gradient provided by the Critic network:

$$\nabla_{\theta_\mu^t} J \approx \frac{1}{N}\sum_{i=1}^{N} \nabla_a Q^t(s, a \mid \theta_Q^t) \mid_{s=s_i^t, a=\mu^t(s_i^t)} \nabla_{\theta_\mu^t} \mu^t(s \mid \theta_\mu^t) \mid_{s=s_i^t} \tag{45}$$

The target networks are still updated by soft update:

$$\theta_{\mu'}^t \leftarrow \tau\theta_\mu^t + (1-\tau)\theta_{\mu'}^t \tag{46}$$

$$\theta_{Q'}^t \leftarrow \tau\theta_Q^t + (1-\tau)\theta_{Q'}^t \tag{47}$$

where τ is the soft-update coefficient. Through the above process, the target-condition policy can adapt to new engine dynamic characteristics based on the source-task parameters.

Based on the spectral-clustering partition and source-task selection results, the TL-DDPG full-envelope extension procedure constructed in this study is shown in Figure 5. After a target condition is input, its similar partition is first determined from altitude, Mach number, inlet total temperature, and inlet total pressure; the pre-trained policy parameters of the corresponding source-task center are then loaded. The target environment continues sampling and updating the Actor/Critic networks, enabling closed-loop fine-tuning of the control law under the new condition and avoiding the accumulation of open-loop conversion errors during envelope extension.

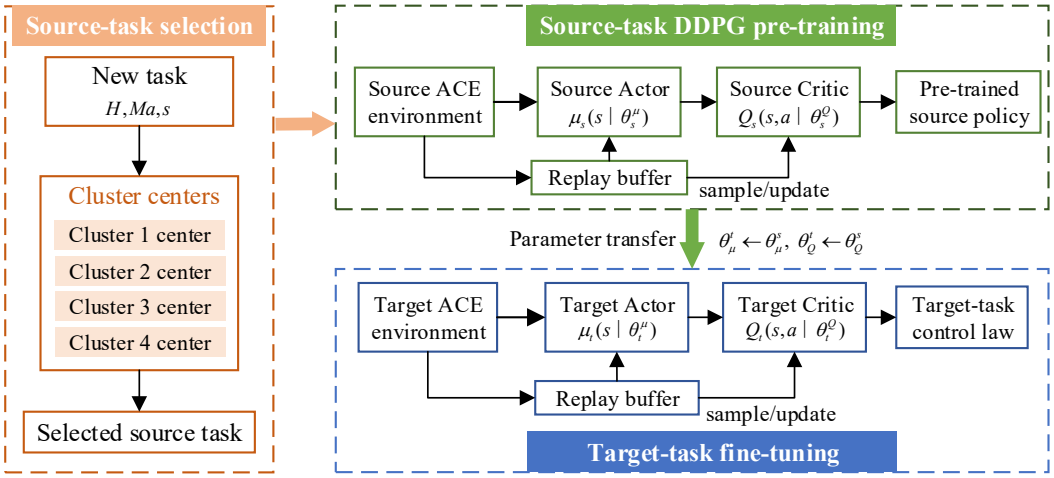


Figure 5 – TL-DDPG full-envelope extension procedure guided by spectral clustering.

4. Simulation Results

4.1 Intra-Partition Transfer Results

The single-condition simulation uses the ground-test condition under the representative three-bypass operating mode, with H=0 km and Ma=0, as the environment. The objective is to accelerate the fan percentage speed from 81% to 95%. The interaction environment is the adaptive cycle engine

model. The DDPG networks are composed of fully connected layers, and experience replay, target-network soft update, and OU noise are used together to improve sample efficiency and training stability. The main hyperparameters are listed in Table 5.

Table 5 – Main hyperparameters of the DDPG network.

Algorithm component	Parameter	Value
Experience replay	Buffer_size	1×10^{6}
Training setting	Simulation steps per episode	500
Training setting	Maximum episodes	2000
Training setting	Mini-batch size	128
Simulation setting	Simulation step size	0.02 s
Simulation setting	Maximum simulation time	10 s
Actor network	Layer structure	[6,120,60,6]
Critic network	Layer structure	[12,120,60,1]
Network update	Learning rate	1×10^{-4}
Target network	Soft-update coefficient	0.001

For the intra-partition transfer case, the operating condition H=1.0552 km and Ma=0.3066 is selected as the target task. The source task is the ground condition H=0 km and Ma=0, and the objective is to accelerate the corrected fan percentage speed from 81% to 95%. The target condition lies in a low-altitude and low-Mach-number region, and its inlet total temperature, inlet total pressure, and component matching state are close to those of the source condition. Therefore, the speed-promotion and constraint-avoidance features embedded in the source-task policy have strong transferability. Figures 6 to 8 show the training reward, main state responses, and six-variable control law for this case.

As shown in Figure 6, conventional DDPG starts exploration from random initialization and reaches the preset reward threshold at episode 199. After TL-DDPG reuses the source-task network parameters, the initial reward level is significantly improved, and the threshold is reached at episode 44, reducing the number of training episodes by 77.89%. Both methods enable the fan percentage speed to reach the 95% target smoothly with similar regulation time, indicating that transfer initialization does not degrade the closed-loop performance for target-speed tracking.

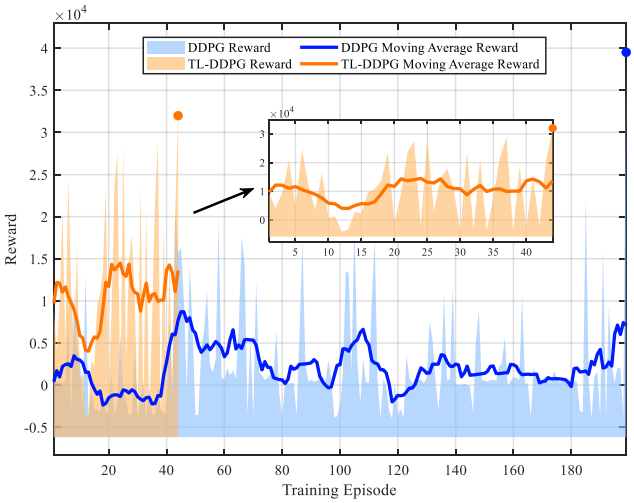


Figure 6 – Reward convergence curve for the intra-partition transfer case.

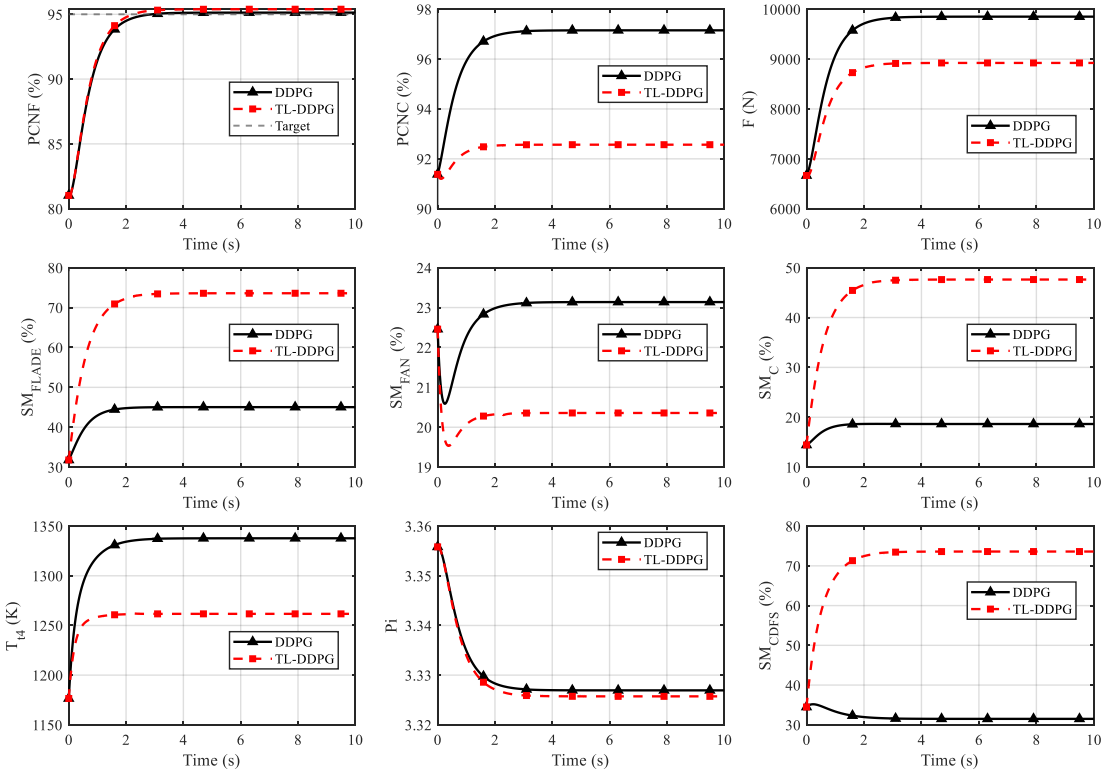


Figure 7 – Acceleration response and safety-constraint variations for the intra-partition transfer case.

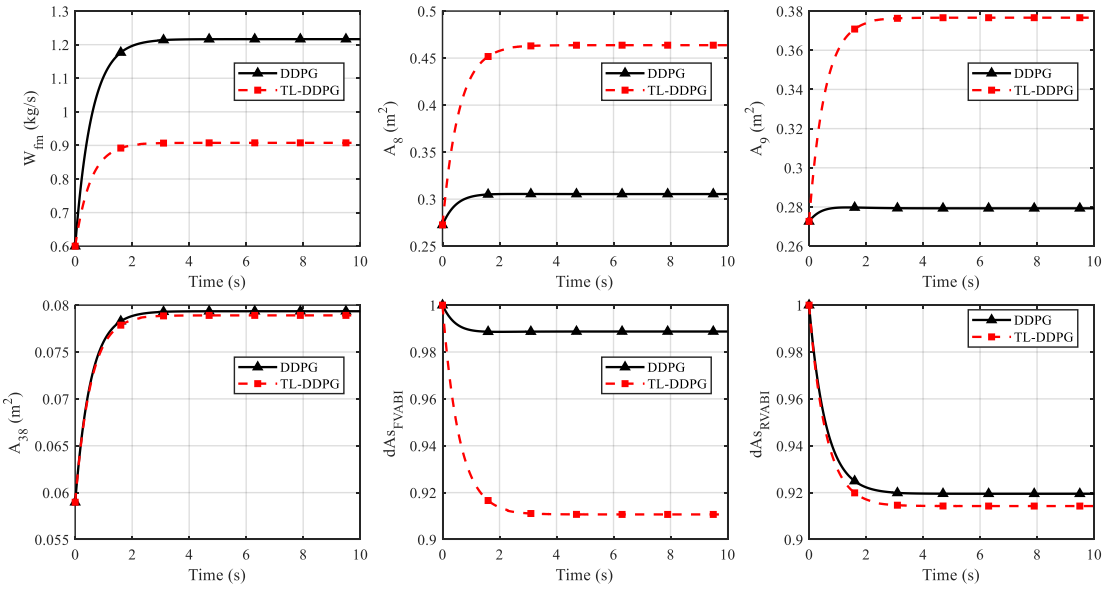


Figure 8 – Six-variable control law for the intra-partition transfer case.

Figures 7 and 8 further show that intra-partition TL-DDPG adopts a combination of lower fuel flow and larger main-nozzle area, resulting in lower turbine-inlet temperature, core speed, and thrust level than conventional DDPG, while significantly improving the CDFS and high-pressure-compressor surge margins. This indicates that, under the present reward design dominated by speed reaching and safety constraints, multiple feasible control laws exist for the target task, and TL-DDPG tends to obtain a safe feasible solution rapidly. If an engineering task also requires thrust matching or terminal steady-state matching, thrust-error, terminal-state-matching, or control-energy terms should be added to the reward function.

4.2 Cross-Partition Transfer Results

For the cross-partition transfer case, the operating condition H=4.4569 km and Ma=0.8186 is selected as the target task. Compared with the source task, this condition exhibits more pronounced

changes in inlet total temperature, inlet total pressure, and component operating match, and is therefore suitable for evaluating the adaptability of source-policy features during cross-region transfer. In the simulation setting, training stops when the episode reward reaches 10000. Figures 9 to 11 present the cross-partition transfer results.

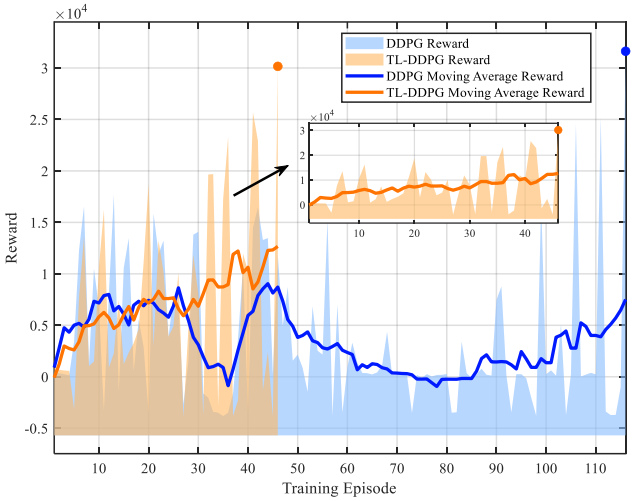


Figure 9 – Reward convergence curve for the cross-partition transfer case.

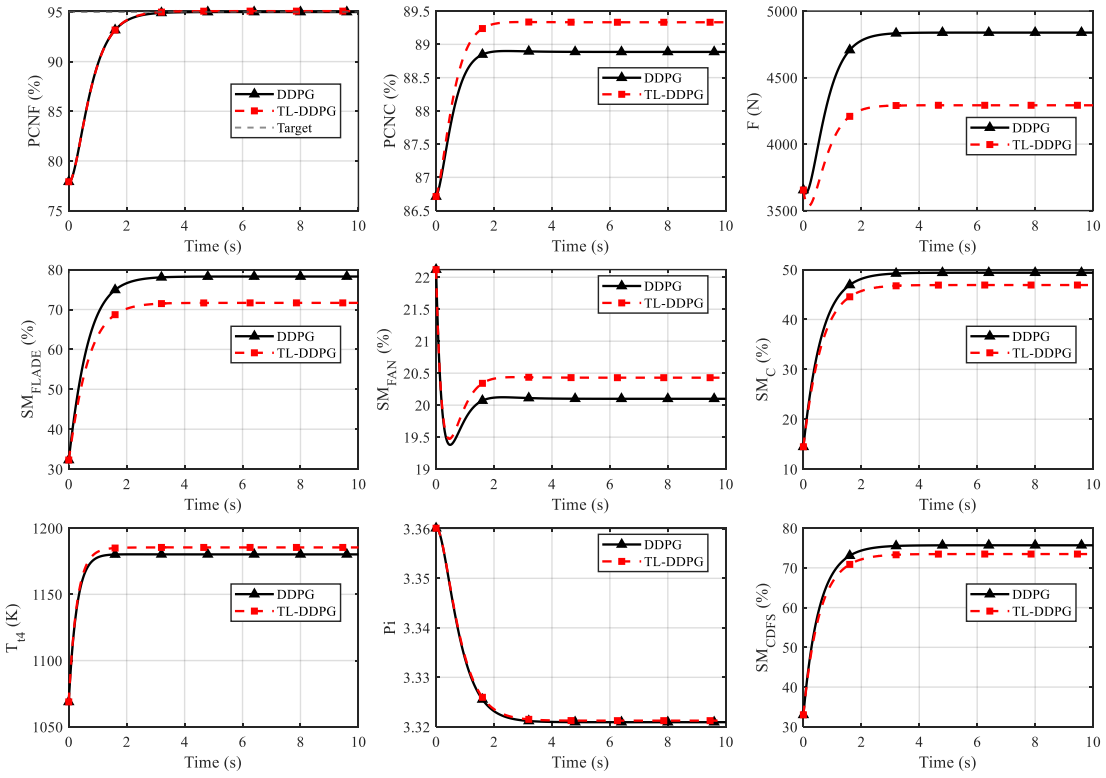


Figure 10 – Acceleration response and safety-constraint variations for the cross-partition transfer case.

As shown in Figure 9, conventional DDPG reaches the preset reward threshold at episode 116, whereas TL-DDPG reaches the threshold at episode 46, reducing the number of training episodes by 60.34%. Compared with the intra-partition case, the efficiency improvement of cross-partition transfer is smaller, indicating that operating-condition differences weaken the match between the source-policy prior and the target environment. Nevertheless, TL-DDPG still substantially shortens the random-exploration stage and rapidly forms a feasible policy that satisfies the target-speed requirement.

The state responses show that both policies can drive the fan percentage speed to converge to 95% within about 3 s. The surge margins remain above the 10% lower limit, and the turbine-inlet temperature is far below the 1850 K constraint. The control-law differences are mainly reflected in the main-nozzle exit area, FLADE nozzle area, and front/rear VABI area allocation, showing that the

policy must rebalance inlet/exhaust matching and stability margins under cross-partition conditions. These results indicate that spectral clustering can provide an effective basis for source-task selection in transfer learning. For wider envelopes, however, increasing the number of cluster centers, adopting multi-source weighted initialization, or introducing negative-transfer detection should still be considered. The training-efficiency comparison between DDPG and TL-DDPG is summarized in Table 6.

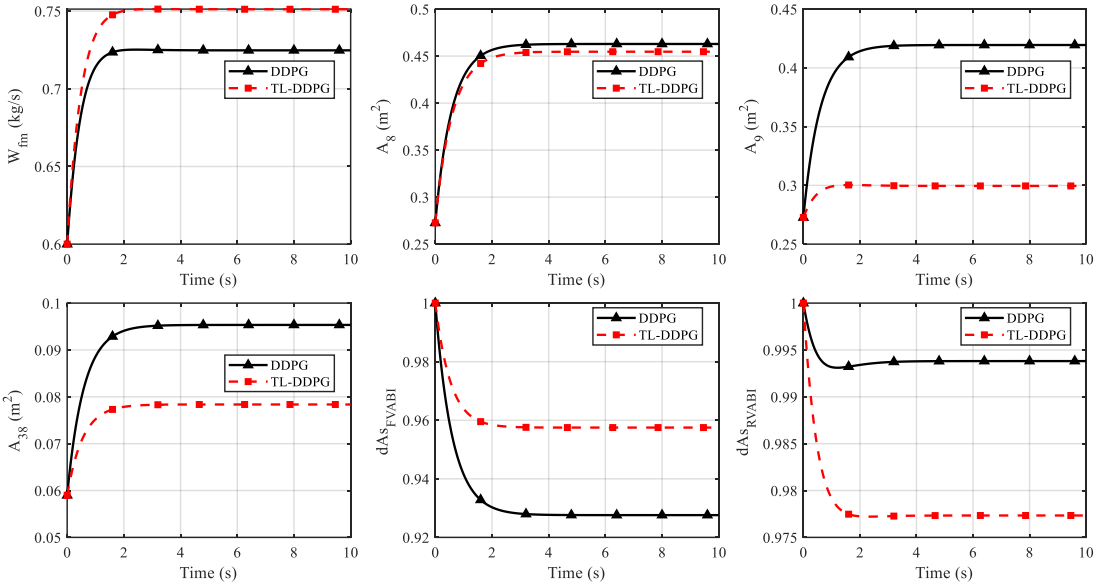


Figure 11 – Six-variable control law for the cross-partition transfer case.

Table 6 – Comparison of TL-DDPG training efficiency.

Case	Target condition	DDPG episodes	TL-DDPG episodes	Reduction
Intra-partition transfer	H=1.0552 km, Ma=0.3066	199	44	77.89%
Cross-partition transfer	H=4.4569 km, Ma=0.8186	116	46	60.34%

5. Conclusions

This paper proposes a transfer-learning-enhanced DDPG method for full-envelope transient control-law optimization of adaptive cycle engines. The ACE acceleration process is formulated as an MDP, and a six-variable DDPG framework is established to generate transient control laws while considering speed-tracking performance, surge-margin constraints, turbine-inlet-temperature limits, and actuator bounds.

To improve full-envelope extension efficiency, spectral clustering is introduced to partition the flight envelope using altitude, Mach number, inlet total temperature, and inlet total pressure. Based on the clustering results, source-task policies are transferred to target operating conditions and further fine-tuned through closed-loop interaction with the engine model.

Simulation results show that the proposed TL-DDPG method significantly reduces the training cost while maintaining feasible acceleration performance. Compared with conventional DDPG, the number of training episodes is reduced from 199 to 44 in the intra-partition case and from 116 to 46 in the cross-partition case, corresponding to reductions of 77.89% and 60.34%, respectively. In both cases, the fan speed reaches the target value, while surge margins and turbine-inlet temperature remain within the prescribed limits. These results indicate that the proposed method provides an efficient framework for full-envelope transient control-law extension of adaptive cycle engines.

6. Acknowledgement

This study is co-supported by the National Natural Science Foundation of China (No. 52305036) and

Guangdong Basic and Applied Basic Research Foundation (No. 2022A1515110005).

7. Contact Author Email Address

Yingzhi Huang, huangyingzhi@nwpu.edu.cn

8. Copyright Statement

The authors confirm that they, and/or their company or organization, hold copyright on all of the original material included in this paper. The authors also confirm that they have obtained permission, from the copyright holder of any third party material included in this paper, to publish it as part of their paper. The authors confirm that they give permission, or have obtained permission from the copyright holder of this paper, for the publication and distribution of this paper as part of the ICAS proceedings or as individual off-prints from the proceedings.

References

[1] Chen M, Deng L, Zhang J, et al. Review on the development status and performance design of adaptive cycle engines. *Propulsion and Power Research*, 2025.

[2] Zenkner S, Carvalho F, Brakmann R G, et al. Variable cycle engine concepts and component technologies: An overview. *Journal of Engineering for Gas Turbines and Power*, Vol. 147, No. 5, 051004, 2025.

[3] Wang H, Zhao S, Luo Q, et al. Investigation on the impact of variable geometry components on the matching characteristics of a three bypass adaptive cycle compression system. *Energy*, Vol. 321, 135153, 2025.

[4] Song K R, Zheng Q G, Zhang H B, et al. Design of acceleration control schedule for adaptive cycle engine based on direct simulation model. *Chinese Journal of Aeronautics*, Vol. 37, No. 6, pp 307-318, 2024.

[5] Zhao X Y, Li Q H, Pang S W, et al. Research on adaptive cycle engine mode transition control method. *Energies*, Vol. 17, No. 6, 1276, 2024.

[6] Chen Y, Lu S W, Zhou W X, et al. Global integration design method of acceleration and deceleration control schedule for variable cycle engine. *Chinese Journal of Aeronautics*, 103374, 2025.

[7] Cai C P, Zheng Q G, Wang Y, et al. Predictive control method for mode transition process of multi-mode turbine engine based on onboard adaptive composite model. *Energy*, Vol. 302, 131748, 2024.

[8] Walsh P P and Fletcher P. *Gas Turbine Performance*. 2nd edition, Blackwell Science, 2004.

[9] Kurzke J and Halliwell I. *Propulsion and Power: An Exploration of Gas Turbine Performance Modeling*. Springer, 2018.

[10] Rugh W J and Shamma J S. Research on gain scheduling. *Automatica*, Vol. 36, No. 10, pp 1401-1425, 2000.

[11] Mattingly J D. *Aircraft Engine Design*. 2nd edition, American Institute of Aeronautics and Astronautics, 2002.

[12] Lu P J, Zhang M C, Huang J Q, et al. Full-envelope acceleration control method of turbofan engine based on variable geometry compound adjustment. *Journal of Propulsion and Power*, Vol. 29, No. 3, pp 595-605, 2013.

[13] Mnih V, Kavukcuoglu K, Silver D, et al. Human-level control through deep reinforcement learning. *Nature*, Vol. 518, No. 7540, pp 529-533, 2015.

[14] Sutton R S and Barto A G. *Reinforcement Learning: An Introduction*. 2nd edition, MIT Press, 2018.

[15] Lillicrap T P, Hunt J J, Pritzel A, et al. Continuous control with deep reinforcement learning. Proc International Conference on Learning Representations, San Juan, Puerto Rico, 2016.

[16] Fang J, Zheng Q G, Liu W, et al. Deep reinforcement learning method for turbofan engine acceleration optimization problem within full flight envelope. *Aerospace Science and Technology*, Vol. 136, 108065, 2023.

[17] Gao W B, Pan M X, Zhou W X, et al. Acceleration control strategy for aero-engines based on model-free deep reinforcement learning method. *Aerospace Science and Technology*, Vol. 120, 107248, 2022.

[18] Taylor M E and Stone P. Transfer learning for reinforcement learning domains: A survey. *Journal of Machine Learning Research*, Vol. 10, pp 1633-1685, 2009.

[19] Von Luxburg U. A tutorial on spectral clustering. *Statistics and Computing*, Vol. 17, No. 4, pp 395-416, 2007.