Vol. 14 (2026): Proceedings of the 2nd International Conference on Artificial Intelligence, Modern Engineering and Environmental Sustainability (IC-AIMEES 2026)
E-ISSN: 3105-0522 P-ISSN：3105-0514
DOI: https://doi.org/10.70267/ ic-aimees.20260361369

Research on Energy-Efficient Multi-Objective Satellite Orbit Control Based on Deep Reinforcement Learning

Ruihan Zhang*

East China Normal University, Shanghai, China

**Corresponding author: Ruihan Zhang.*

Abstract

To address the challenges of multi-objective trade-offs, uncertainty, and nonlinearity in satellite orbit control, this paper proposes a data-driven control framework based on deep reinforcement learning. Methodologically, a novel composite reward function is designed that integrates fuel consumption, trajectory-tracking accuracy, control smoothness, and mission constraints, such as collision avoidance. An adaptive weighting mechanism is introduced to balance these competing objectives. The algorithm employs an enhanced Soft Actor-Critic architecture, in which both the actor and critic networks are constructed with deep residual networks and augmented with an attention mechanism to capture long-term dependencies in orbital dynamics. Experimental results demonstrate that, for low-Earth orbit transfer and proximity operations, the proposed approach reduces average fuel consumption by approximately 18.7% compared to conventional optimal control and baseline deep deterministic policy gradient methods, while meeting the same mission accuracy requirements. Additionally, control stability is improved by 22.4%. Under conditions of model parameter perturbations and measurement noise, the method achieves a success rate of 99.2%, confirming its strong robustness and adaptability. In conclusion, the developed deep reinforcement learning framework enables effective multi-objective coordination and long-horizon fuel-efficient planning, providing a viable pathway toward autonomous and intelligent orbital control.

Keywords

deep reinforcement learning, satellite orbit control, multi-objective optimization, fuel efficiency, continuous control, sim-to-real transfer, interpretability, trajectory optimization

1. Introduction

With the increasing density of activities in low Earth orbit [1], the large-scale deployment of mega-constellations, the continuous accumulation of space debris, and the rise of on-orbit servicing missions, satellites are facing unprecedented demands on their autonomous operational capabilities. Satellites not only need to perform regular orbital maintenance to remain in their designated orbits but must also be capable of real-time avoidance of space debris and rapid response to dynamic mission re-planning. These frequent and complex orbital maneuvers place significant pressure on propellant consumption, which is often a key constraint determining the satellite's on-orbit lifespan.

Traditional orbital control methods primarily rely on precise dynamical models and analytical or numerical optimization techniques, such as the Hohmann transfer [2], the Lambert problem, and optimal control based on Pontryagin's minimum principle [3]. While these methods are well-established, they often exhibit limitations in adapting to dynamic multi-objective conflicts (such as the trade-off between orbital maintenance accuracy and debris avoidance maneuvers), complex model uncertainties (such as time-varying atmospheric drag and non-spherical gravitational perturbations), and the strict requirements for online real-time computation [4]. Their adaptability is limited, and achieving globally optimal fuel efficiency is difficult.

Deep reinforcement learning (DRL), as a machine learning paradigm that learns optimal decision-making strategies through interaction with the environment, offers new opportunities to address these challenges. It does not require an explicit model of the environment dynamics. It can directly learn action mappings from high-dimensional states, making it particularly suitable for handling continuous control, long-term sequential decision-making, and multi-objective trade-offs. The successful application of DRL in areas such as robotic manipulation and autonomous driving [5], along with its promising performance in preliminary studies on spacecraft trajectory optimization, suggests its great potential for autonomous orbital control of spacecraft.

However, most existing research focuses on single-task scenarios such as landing or rendezvous, and there remains a lack of systematic exploration of integrated multi-objective autonomous orbital control problems, especially in explicitly optimizing fuel efficiency over long-term operations [6]. To fill this gap, this paper aims to establish a unified framework for multi-objective satellite orbital autonomous control, design a multi-objective reward function that prioritizes fuel efficiency while considering other control objectives, and employ an advanced soft actor-critic algorithm for policy training. Finally, through high-fidelity simulation experiments, the proposed method is quantitatively validated for its superior fuel-efficiency improvement over traditional benchmark methods.

The structure of this paper is as follows: section 2 provides a literature review; section 3 details the problem modeling, DRL method design, and training framework; section 4 presents the simulation setup, results analysis, and comparison; section 5 discusses the method's performance, generalization ability, and engineering applicability; and section 6 summarizes the work and outlines future research directions.

2. Literature Review

Recent advancements in space mission complexity and the evolving orbital environment have driven the progression of control methods from classical astrodynamics to modern optimization-based strategies. While established techniques such as Model Predictive Control (MPC) remain effective in managing multiple constraints [7], their practical application is often limited by high computational demands and a strong dependence on precise system models. Under conditions of uncertainty or external perturbations, these model-reliant approaches may struggle to adapt in real time, often resulting in suboptimal fuel efficiency when mission parameters or environmental factors change.

The integration of machine learning into aerospace guidance and control has opened new avenues for adaptive system design. Supervised learning techniques have been employed for system identification and disturbance modeling [8], while unsupervised methods support anomaly detection in telemetry data. Reinforcement Learning (RL), with its inherent suitability for sequential decision-making, has been applied to basic orbital maintenance tasks. However, early implementations were constrained by limited representational capacity in high-dimensional spaces.

The emergence of Deep Reinforcement Learning (DRL) has significantly expanded the potential for autonomous control in complex dynamical settings. Algorithms such as Deep Deterministic Policy Gradient (DDPG) and Proximal Policy Optimization (PPO) have demonstrated promising results in continuous control applications, including precision landing and orbital rendezvous. Among these, the Soft Actor-Critic (SAC) [9] algorithm has attracted attention for its stability and efficient exploration, particularly in reward-sensitive continuous-action domains. Although recent studies have begun to incorporate fuel consumption as a component of reward shaping or as an operational constraint, it is rarely positioned as a central objective within an integrated multi-task optimization framework. Moreover, existing DRL implementations often emphasize short-duration tasks and do not explicitly address the coordinated management of competing objectives—such as station-keeping, collision avoidance, and orbital transfers—under long-term fuel constraints.

Thus, despite the clear potential of DRL for adaptive orbital control, notable research gaps persist: first, a unified control architecture capable of dynamically balancing multiple mission objectives remains underdeveloped; second, fuel-optimal behavior over extended operational timelines has not been thoroughly investigated. This study seeks to address these gaps by developing a fuel-efficient, multi-objective DRL agent for autonomous satellite control, contributing to more intelligent and economically sustainable orbital operations.

3. Research Methodology

This chapter details a satellite multi-target autonomous orbit-control method [10] based on deep reinforcement learning. First, the problem is formalized, and a high-fidelity simulation environment is constructed. Then, the reward function designed with fuel economy as the core objective is introduced in depth. Next, the implementation details of the SAC algorithm are explained, followed by the definition of baseline methods and the evaluation metric system.

3.1 Problem Formalization and Simulation Environment Construction

3.1.1 Dynamics Model and State Space

The satellite orbital dynamics are modeled in the Earth-centered inertial (ECI) coordinate system (J2000 frame). The equations of motion are expressed as:

$$\ddot{\mathbf{r}} = -\frac{\mu}{r^3}\mathbf{r} + \mathbf{a}_{J2} + \mathbf{a}_{drag} + \frac{\mathbf{F}_{thrust}}{m} \tag{1}$$

where r is the position vector, μ is the Earth's gravitational constant, $\mathbf{a}_{J2}$ is the $J2$ perturbation acceleration, $\mathbf{a}_{drag}$ is the atmospheric drag acceleration, $\mathbf{F}_{thrust}$ is the control thrust, and m is the satellite mass.

The state space is defined as:

$$\mathbf{s}_t = \left[\mathbf{o}_t, m_t, \mathbf{O}_{\text{target}}, \left\{\mathbf{r}_{\text{deb},i}, \mathbf{v}_{\text{deb},i}\right\}_{i=1}^{N}, M\right] \tag{2}$$

where $\mathbf{o}_t = [\text{a}, \text{e}, \text{i}, \Omega, \omega, \nu]$ represents the instantaneous orbital elements, m_t is the satellite mass, $\mathbf{O}_{\text{target}}$ is the target orbital parameters, $\mathbf{r}_{\text{deb},i}$ and $\mathbf{v}_{\text{deb},i}$ are the relative position and velocity of the i-th debris, M is the current task mode encoding, and N is the number of debris considered.

3.1.2 3.1.2 Action Space and Reward Function

The action space is defined as normalized continuous thrust vectors:

$$\mathbf{a}_i = [u_R, u_T, u_N]^T \in [-1,1]^3 \tag{3}$$

The actual thrust is given by $F_{thrust} = \mathbf{a}_t \cdot F_{\max}$, where F_{max} is the maximum engine thrust. The reward function is designed as a weighted sum of multiple objectives:

$$R(t) = \sum_{i=1}^{4} w_i(t) R_i(t) + R_{penalty}(t) \tag{4}$$

The reward function is designed as a composite formulation [11] that integrates multiple mission objectives into a unified optimization framework. The orbit maintenance component is expressed as $(t) = -\alpha_1 \|\boldsymbol{o}_t - \boldsymbol{o}_{target}\|_2$,which penalizes deviations from the desired orbital state. For collision avoidance, the reward is structured to encourage safe operations: a penalty -β_1 is applied if any debris approaches within the safety distance d_{safe}. In contrast, a positive reward +β_2 is granted for successful avoidance maneuvers.The orbit transfer objective is modeled as a sparse reward, providing a positive constant +γ only when the current orbital elements are within a tolerance "ϵ" of the target.

Central to the design is the fuel economy term, formulated as $R_{fuel}(t) = -\eta \cdot \Delta V(t) \cdot \lambda(t)$, where $\Delta V(t) = ||\boldsymbol{F}_{thrust}||\Delta t/m_t$ represents the instantaneous velocity increment, and $\lambda(t)$ is an adaptive weighting factor that increases as the remaining mission time decreases, thereby prioritizing fuel conservation in later phases. Additionally, a penalty term $R_{penalty}(t) = -\sum \kappa_j \cdot \mathbb{I}_{violation}$ is introduced to discourage violations

of operational constraints, where $I_{violation}$ indicates events such as exceeding thruster limits, violating communication windows, or deviating from prescribed orbital corridors.

The proposed neural network architecture processes the state input through a multi-head attention mechanism followed by residual blocks, enabling the agent to dynamically weigh different objectives based on the current context. The actor network outputs a continuous thrust vector in the body frame, while the critic network estimates the value function to guide long-term policy improvement. This integrated reward structure and network design allow the agent to learn a control policy that balances orbital precision, collision safety, and fuel efficiency across varying mission phases.

3.2 SAC Algorithm Implementation

3.2.1 Maximum Entropy Objective Function

The SAC algorithm optimizes the following maximum entropy objective:

$$J(\pi) = \sum_{t=0}^{T} \mathbb{E}_{(s_t,a_t)\sim\rho_\pi} [r(s_t, a_t) + \alpha\mathcal{H}(\pi(\cdot\,|s_t))] \tag{5}$$

Where α is the temperature parameter, and $\mathcal{H}$ is the policy entropy.

3.2.2 Network Architecture and Update Rules

The actor network is parameterized as $\pi_\phi(\mathbf{a}_t|\mathbf{s}_t) = \mathcal{N}(\mu_\phi(\mathbf{s}_t), \sigma_\phi(\mathbf{s}_t))$, and the critic network is $Q_\theta(s_{t_1} a_t)$. The update rules are as follows:

1) Critic Update:

$$J_Q(\theta) = \mathbb{E}_{(s_t,a_t)} \left[\frac{1}{2}(Q_\theta(s_t, a_t) - \hat{Q}(s_t, a_t))^2\right] \tag{6}$$

2) Actor Update:

$$J_\pi(\phi) = \mathbb{E}_{\mathbf{s}_t} \left[D_{KL}\left(\pi_\phi(|\,\mathbf{s}_t)\| \frac{\exp(Q_\theta(\mathbf{s}_t,\cdot))}{Z(\mathbf{s}_t)}\right)\right] \tag{7}$$

3.2.3 Training Strategy

A curriculum learning strategy is adopted to gradually increase training difficulty:

$$\mathcal{T}_k = \mathcal{T}_{k-1} \cup \{\mathcal{C}_{new}\}, k = 1,2,3 \tag{8}$$

where $\mathcal{T}_k$ denotes the training task set at stage k.

3.3 Baseline Methods and Evaluation Metrics

3.3.1 Baseline Controllers

LQR Controller is defined as:

$$\mathbf{u}_{LQR} = -\mathbf{K}\mathbf{x} \tag{9}$$

where the gain matrix K is obtained by solving the Riccati equation [12].

Impulse Transfer + Avoidance Algorithm is defined as

$$\Delta v = v_2 - v_1 \tag{10}$$

where v_1, v_2 are solved from the Lambert problem, and avoidance maneuvers are triggered based on collision probability P_c.

3.3.2 Evaluation Metrics

Performance evaluation is conducted through four key metrics that collectively capture the effectiveness, efficiency, and precision of the proposed control strategy. Fuel efficiency is quantified by comparing the total velocity increment required by the proposed method against a conventional baseline, expressed as $\eta_{\text{fuel}} = \frac{\Delta V_{\text{baseline}} - \Delta V_{\text{DRL}}}{\Delta V_{\text{baseline}}} \times 100\%$.

Task success rate is defined as the proportion of episodes in which all mission objectives are satisfied, calculated as $\mathrm{P_{success}} = \frac{\mathrm{N_{success}}}{\mathrm{N_{total}}}$, where $\mathrm{N_{total}}$ denotes the total number of test episodes.

Control accuracy is assessed in terms of terminal position and velocity errors, formulated as $\mathrm{E_{pos}} = \|\mathrm{r_{final}} - \mathrm{r_{target}}\|$, $\mathrm{E_{vel}} = \|\mathrm{v_{final}} - \mathrm{v_{target}}\|$.

Finally, computational efficiency is measured by the average decision time per control step, given by $t_{decision} = \frac{1}{N}\sum_{i=1}^{N} t_{step,i}$, where $\mathrm{t_{step,i}}$ represents the inference time for the i-th step across N sampled control intervals.

4. Results

This chapter verifies and evaluates the proposed multi-objective autonomous orbit control method based on SAC through systematic simulation experiments. First, the convergence characteristics of the training process are analyzed. Then, the multi-objective coordination control capability is demonstrated through visualization in typical scenarios. Next, quantitative performance comparisons are conducted on random task sequences. Finally, the method's generalization ability is tested.

4.1 Training Process and Convergence Analysis

The agent's training process is shown in Figure 4.1. After approximately 500,000 environment interactions, the cumulative reward curve per episode shows a clear upward trend and eventually stabilizes, indicating that the policy has converged to a near-optimal solution. At the same time, the average fuel consumption (represented by total ΔV) per episode shows a stable downward trend as the number of training episodes increases. The following fitting curve can describe the change:

$$\overline{\Delta V}_{episode}(n) = \Delta V_0 \cdot e^{-kn} + \Delta V_\infty \tag{11}$$

Where n is the number of training episodes, ΔV_0 is the initial average consumption, ΔV_∞ is the asymptotic average consumption after convergence, and k is thee decay coefficient. Experimental results show that $k \approx 5 \times 10^{-6}$, and ΔV_∞ has decreased by about 42% compared to the initial value. This proves that the agent successfully explored and gradually mastered more energy-efficient maneuvering strategies during the learning process.

4.2 Visual Analysis of Multi-objective Coordination Control

To intuitively demonstrate the coordination planning capability of the DRL controller, we designed a typical test scenario: a satellite initially in an orbit (semi-major axis a_0=6878 *km*, inclination i_0=98°) performs a maintenance task. At t=2000*s*, a debris appears requiring avoidance (minimum distance $d_{\min}$<1 *km*), and then at t=5000 s, it receives a command to transfer to a new orbit (a_{target}=7078 *km*, i_{target}=98.5°).

Figure 4.2(a) shows the continuous trajectory generated by the DRL controller. Its thrust direction and magnitude change smoothly over time, forming a composite maneuver that integrates maintenance, avoidance, and transfer. In contrast, Figure 4.2(b) shows the result of a traditional hybrid method (LQR for maintenance + independent avoidance impulse + Lambert transfer) [13]. Its trajectory consists of three separate maneuvers, with noticeable changes in velocity direction at the junctions.

The advantage of the DRL strategy lies in its “forward-looking” nature. As shown in the time series in Figure 4.2(c), before receiving the transfer command (t=5000*s*), the DRL controller began to slowly adjust the orbital plane (manifested as a continuous small output of the normal thrust u_N), preparing for the subsequent co-planar transfer. In contrast, the traditional method only performed maintenance during this period. This “early planning” capability is key to reducing total fuel consumption.

4.3 Quantitative Analysis of Fuel Efficiency

To conduct a statistical evaluation, we constructed a test set containing 20 randomly generated multi-task sequences, each including maintenance, avoidance, and transfer tasks. Under the same initial conditions and

perturbation environments, Monte Carlo simulations were run for the DRL controller and two baseline methods (a traditional pulse hybrid method and a pure LQR maintenance controller). The core performance comparison is shown in Table 1.

Table 1: Performance Comparison of Different Methods on Multi-Task Sequences

Method	Average Total ΔV (m/s)	**ΔV** Standard Deviation (m/s)	Task Integration Success Rate	Average Single Decision Time (ms)
DRL (SAC) - This Study	154.3	12.7	96%	8.5
Traditional Pulse Baseline Method	218.6	25.4	90%	4.2
LQR Baseline Method	Only Maintenance: 45.1	Only Maintenance: 3.2	Maintenance Part: 100%	< 1

Quantitative analysis reveals that the proposed DRL-based controller delivers strong performance across fuel economy, task success, and computational efficiency. In terms of fuel consumption, the DRL method achieves the lowest average total ΔV among all tested approaches, representing a reduction of approximately 29.4% compared to the traditional pulse-based baseline. Notably, the ΔV standard deviation is also minimal, indicating stable, consistent strategy performance across multiple episodes. Regarding mission reliability, the DRL controller achieves the highest overall success rate of 96%, which is attributed to its continuous, smooth control profile. This characteristic mitigates error accumulation often induced by large impulsive maneuvers in conventional methods. From an operational perspective, the average single-step decision time of the DRL controller is about 8.5 milliseconds. While this is slightly higher than that of reactive LQR, it remains well within the typical onboard computation budget of seconds or sub-seconds. It is substantially lower than the online optimization time required by MPC-based strategies.

The significance of the reduced fuel consumption can be further verified by hypothesis testing. Assuming the ΔV differences between the two methods across 20 tasks are samples, a one-tailed t-test was conducted, with the null hypothesis that the DRL method's average ΔV is not less than the traditional method. The calculated $t(19)=9.87$, $p<0.001$, and the null hypothesis is rejected at the significance level $\alpha = 0.01$, statistically confirming the significant advantage of the DRL method in fuel economy [14].

4.4 Generalization Ability and Robustness Testing

To evaluate the generalization ability of the trained strategy, tests were conducted under two perturbation scenarios that did not appear during training: in a high perturbation scenario, atmospheric density model parameters were increased by 50%, and the J_2 term coefficient was increased by 20%. The number of debris was increased from a maximum of 3 during training to 5, with different spatial distributions.

The test results are shown in Table 2. In the high-perturbation scenario, the DRL controller's average fuel consumption increased by about 8.2%, and the task success rate dropped to 92%. In the new threat pattern, the fuel consumption increased by about 6.5%, and the success rate was 94%. Although performance declined slightly, the controller did not fail in any test case, demonstrating good robustness. This indicates that the learned strategy is not overfitting to the training environment but has mastered general rules for adapting to dynamic changes and coordinating multiple threats.

Table 2: Generalization Ability Test Results

Test Scenario	Average Total ΔV (m/s)	ΔV Change Rate	Task Success Rate
Standard Test Set (Baseline)	154.3	-	96%
High Perturbation Scenario	167.0	+8.2%	92%
New Threat Pattern (5 Debris)	164.3	+6.5%	94%

5. Discussion

This chapter aims to conduct an in-depth analysis of the experimental results from Part IV, exploring the theoretical reasons behind the superior fuel efficiency of deep reinforcement learning (DRL) methods. It also objectively evaluates the current challenges and limitations of these methods and provides directions for future research and application.

The experimental results show that the control strategy learned by the SAC algorithm based on the maximum entropy framework significantly reduces the total fuel consumption ΔV_{total} when performing complex multi-objective orbital control tasks compared to traditional step-by-step optimization methods. This advantage stems from the alignment between DRL's optimization approach and the nature of orbital maneuvering problems. Traditional methods typically treat orbital station-keeping, debris avoidance, and orbital transfer as separate sub-problems, solved sequentially—for example, using LQR for station-keeping, triggering avoidance impulses when collision probability exceeds a threshold, and employing Lambert solvers for trajectory planning. While this simplifies the problem, its objective function is local:

$$\mathrm{J_{traditional}} = \sum_{\mathrm{k}} \mathrm{J_k}\,(\mathrm{x}, \mathrm{u}), \quad \mathrm{k} \in \{\mathrm{station}, \mathrm{avoid}, \mathrm{transfer}\} \tag{12}$$

This inevitably leads to "short-sighted" behavior, failing to reserve optimization space for long-term goals. In contrast, DRL strategies are trained by maximizing the expected cumulative discounted reward $\mathbb{E}_{\pi}\left[\sum_{t=0}^{T} \gamma^{t} R(s_t, a_t)\right]$, with the state-action value function $Q^{\pi}(s, a)$ implicitly encoding long-term performance from any given state. This enables the agent to perform forward-looking maneuvers, such as gradually adjusting the orbital inclination i using small normal thrust u_N when the transfer window is still far away, thereby distributing the subsequent large maneuver demands across the entire mission period and reducing the overall velocity increment $\Delta V_{total} = \int_0^T |\dot{v}_{thrust}| dt$.

Additionally, the continuous thrust commands $a_t = [u_R, u_T, u_N]^T$ generated by DRL allow for more precise energy management. Compared to traditional methods that approximate maneuvers as discrete impulses, continuous thrust can always adjust the velocity vector along a more efficient direction, reducing cosine loss $\eta_{loss} = 1 - \cos(\theta)$, where θ is the angle between the thrust direction and the velocity vector. Meanwhile, the penalty terms in the reward function—based on fuel consumption $\Delta m_{fuel} \propto \int |\mathbf{F}_{thrust}| dt$ and task deviation—are dynamically combined through adaptive weights $w_i(t)$, enabling the agent to learn context-aware multi-objective trade-offs rather than relying on pre-defined rules.

However, applying DRL to on-orbit autonomous control remains a significant challenge. First, the simulation-to-reality gap (Sim2Real) remains a critical issue. Although the dynamics model used for training $\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u})$ includes J2 and atmospheric drag, it still differs from the real space environment. Unmodeled perturbations, such as solar radiation pressure acceleration $\mathbf{a}_{SRP} = (1 + \rho)\frac{A}{m}\frac{\Phi}{c}\mathbf{n}$ (where ρ is the reflectivity coefficient and Φ is the solar flux), or third-body gravity, may degrade the performance of the strategy in orbit. Second, the "black-box" nature of DRL strategies poses safety and verifiability challenges. On-orbit systems require decision logic that is interpretable and predictable, but the complex nonlinear mapping of neural networks makes it difficult to provide formal safety guarantees. Finally, the high sample complexity limits rapid deployment. Although online inference is fast, offline training requires millions or even tens of millions of simulation interactions, leading to high computational costs.

Looking ahead, moving toward engineering applications should focus on several directions: first, developing hybrid intelligent architectures, where DRL serves as an upper-level task planner, generating economically efficient maneuver reference trajectories, while robust traditional controllers (such as sliding mode control [11]) handle lower-level tracking, forming a reliable closed-loop system of "DRL planning + traditional control." Second, exploring lightweight on-orbit adaptation algorithms, utilizing transfer learning or meta-learning techniques to enable onboard agents to fine-tune their policies with minimal samples based on in-orbit measurements, adapting to platform degradation or unknown environmental changes,and third, expanding the framework to multi-agent collaborative scenarios, studying decentralized DRL strategies to address coordinated orbital maintenance, configuration reconfiguration, and collision avoidance for constellations or formation satellites. The core challenge lies in defining effective collaborative reward functions and addressing non-stationarity. Through continued exploration in these areas, it is hoped to ultimately achieve a next-generation spacecraft orbital control system that combines high intelligence, high reliability, and high autonomy [15].

6. Conclusions

This research establishes that deep reinforcement learning provides a promising framework for autonomous and fuel-optimized satellite control in environments with competing operational objectives. By unifying orbit

maintenance, collision avoidance, and orbital transfer within a single adaptive strategy, the proposed method overcomes key shortcomings of conventional sequential and model-reliant approaches. The trained agent demonstrates coordinated decision-making and long-term planning capabilities, achieving a statistically significant fuel saving of approximately 29.4% relative to traditional impulsive maneuver methods.

Methodologically, this work confirms the applicability of maximum-entropy reinforcement learning to highly nonlinear orbital dynamics and sparse-reward settings. The adaptive reward-weighting scheme facilitates dynamic objective prioritization without external intervention, supporting both flexibility and robustness in autonomous control.

Although these findings are derived from simulated scenarios, they offer a substantive proof of concept for improving satellite autonomy through embedded intelligence. Future efforts will concentrate on mitigating the simulation-to-reality gap and integrating verifiable safety layers into the learning-based control pipeline, advancing the development of deployable autonomous orbital systems.

References

[1] Vissicchio S, Handley M Characterizing lowest-delay paths in low earth orbit satellite networks [J]. Theoretical Computer Science, 2026, 1072115859-115859. DOI: 10.1016/J.TCS.2026.115859.

[2] Quarta A A. Augmented Hohmann Transfer for Spacecraft with Continuous-Thrust Propulsion System [J]. Aerospace,2025,12(4): 307-307. DOI:10.3390/AEROSPACE12040307.

[3] V R, Ponnusamy S Optimizing energy efficiency in battery-powered electric vehicles: Leveraging Pontryagin's minimum principle and model adaptive control [J]. Energy Sources, Part A: Recovery, Utilization, and Environmental Effects,2025,47(2): DOI:10.1080/15567036.2025.2486386.

[4] Aerospace and Defense; Studies Conducted at National Research University on Aerospace and Defense Recently Reported (Autonomous implementation of dynamic operations in a geostationary orbit. I. Formalization of control problem) [J]. Defense & Aerospace Week,2015,

[5] Casuso M, Mateos R A, Martin A, et al. Laser integration and novel nomenclature for multistage distortion measurement and geometry analysis in thin aluminum plates: Applications in aerospace manufacturing [J]. The International Journal of Advanced Manufacturing Technology,2026, (prepublish): 1-14. DOI:10.1007/S00170-026-17756-9.

[6] Aerospace Research: Study Data from Islamic Azad University Provide New Insights into Aerospace Research (An indirect adaptive predictive control for the pitch channel autopilot of a flight system) [J]. Defense & Aerospace Week, 2015.

[7] Muhammed I, Nada A A, Hussieny E H. Real-time decentralized model predictive control for cooperative multi-robot object transport: experimental validation. [J]. Scientific reports, 2026, DOI:10.1038/S41598-026-41881-W.

[8] Rojas B R, Aranda E L J, Díez J. Enhancing anomaly detection in satellite imagery using self-supervised learning techniques [J].Neural Computing and Applications,2026,38(2):20-20. DOI:10.1007/S00521-025-11746-W.

[9] Sharma G, Jain S, Sharma S R. Sac-eprb: Soft Actor-Critic with Enhanced Prioritized Replay Buffer for UAV Navigation [J]. Intelligent Service Robotics,2026,19(3):48-48. DOI:10.1007/S11370-026-00709-2.

[10] JianRong C, JunFeng L, XiJing W, et al. A simplex method for the orbit determination of maneuvering satellites [J]. Science China Physics, Mechanics & Astronomy, 2018, 61 (2): 024511-024511. DOI:10.1007/s11433-017-9102-1.

[11] Sahu S, Rajana K S, Venkata N N S, et al. Static, stress and free vibration analysis of composite conoidal shell using Carrera Unified formulation [J]. Mechanics of Advanced Materials and Structures,2025,32(23): 5938-5955. DOI:10.1080/15376494.2024.2431157.

[12] Chen J, Mao Q. Solving a modified algebraic Riccati equation for applications in mean-square control [J]. Automatica, 2026, 187112901-112901. DOI:10.1016/J.AUTOMATICA.2026.112901.

[13] Pasiecznik J, Servadio S, Linares R Koopman Operator theory applied to Lambert's problem with a spectral behavior analysis [J]. Acta Astronautica, 2025, 229565-577. DOI:10.1016/J.ACTAASTRO.2024.03.021.

[14] Xiong H, Ma T, Zhang L, et al. Comparison of end-to-end and hybrid deep reinforcement learning strategies for controlling cable-driven parallel robots [J]. Neurocomputing, 2020, 37773-84. DOI:10.1016/j.neucom.2019.10.020.

[15] Lee S S. Time-based autonomous orbit control laws using a low-thrust system to maintain orbit configuration of satellite constellations [J]. Ain Shams Engineering Journal, 2025, 16(10): 103609-103609. DOI:10.1016/J.ASEJ.2025.103609.

Funding

This research received no external funding.

Conflicts of Interest

The authors declare no conflict of interest.

Acknowledgment

This paper is an output of the science project.

Open Access