Research on an End-to-End Motion Control Method for Extraterrestrial Exploration Mobile Robots Based on Physics-Informed Reinforcement Learning

Abstract. When performing tasks on unknown, unstructured planetary surfaces, extraterrestrial exploration mobile robots face severe challenges posed by complex terrain such as soft-soil entrapment. Conventional modular motion-control methods involve many stages, are complex to implement, and struggle to accurately model strongly nonlinear terrain such as soft-soil sinkage. Conventional data-driven end-to-end reinforcement-learning methods, on the other hand, produce control commands that lack any guarantee of physical feasibility, risking violation of hardware constraints on wheel speed and steering angle and thereby causing structural damage or even mission failure. To address these problems, this paper proposes an end-to-end motion-control method based on physics-informed reinforcement learning. The method builds an end-to-end motion-control model based on a Transformer encoder-decoder architecture and designs a Physics-Informed Reward Function (PIRF). By jointly encoding the two-stage escape-navigation task knowledge together with engineering physical constraints (steering angle, steering-angle rate, wheel speed, etc.) as reward terms, the policy output is explicitly guided to fall within the hardware safety boundaries during training. On a high-fidelity physics-simulation platform, simulation experiments were designed covering multiple test scenarios. Under entrapment configurations not seen during training, the model attained a 100% mission success rate; throughout motion, the steering angle, steering-angle rate, and wheel speed satisfied the preset physical limits. An ablation comparison further shows that the physical-constraint terms in the PIRF improve the convergence stability of training. These results verify the effectiveness and engineering applicability of the proposed method and can provide technical support for the safe mobility of extraterrestrial exploration mobile robots over complex terrain.

Keywords: End-to-End Motion Control, Reinforcement Learning, Extraterrestrial Exploration, Safe Mobility.

1 Introduction

Deep-space exploration refers to exploration activities directed at the Moon and celestial bodies and space beyond it, including lunar exploration, Mars exploration, and asteroid exploration [1]. As an important component of extraterrestrial exploration activities, deep-space exploration carries major strategic significance for advancing space-science research and the utilization of extraterrestrial resources [2,3].

When executing tasks on unknown, unstructured planetary surfaces, extraterrestrial exploration mobile robots face severe challenges such as harsh environments, scarce prior information, and large communication delays; their autonomous motion-control

capability is the key to mission success [4,5]. Conventional modular motion-control methods (perception-mapping-planning-control) are highly complex to implement, have poor environmental adaptability, and struggle to accurately model complex terrain such as soft-soil sinkage [6]. End-to-end motion control aims to map sensor inputs directly to control commands, offering a new approach to autonomous decision-making in complex environments [7,8]. Among the available techniques, reinforcement learning (RL), by virtue of its ability to optimize policies through trial-and-error interaction with the environment, has become an important technical route for realizing end-to-end motion control [9-11]. However, in extraterrestrial-exploration scenarios, conventional data-driven RL methods still face two core challenges. First, such methods depend heavily on the scale of the training data and require techniques such as data cleaning and automatic labeling to ensure the reliability of training samples [11]; yet in extraterrestrial-exploration scenarios, the cost of acquiring real-environment data is high and samples are scarce, making it difficult to meet these data requirements [12]. Second, the inherent end-to-end mapping property of such methods renders the decision process non-interpretable, and the generated action sequences may violate physical-feasibility constraints, typically manifesting as abrupt changes in wheel speed, steering angles exceeding mechanical limits, and accelerations beyond the rated capability of the motors [13]. Such limit-exceeding actions are not only difficult to deploy stably on a real robot but also further widen the sim-to-real transfer gap. For extraterrestrial-exploration robots, hardware-limit constraints constitute safety boundaries that must be strictly observed, and any limit-exceeding operation may cause irreversible hardware damage or even mission failure [14].

In recent years, the physics-guided machine-learning paradigm has attracted broad attention [15]. This paradigm aims to inject prior knowledge such as physical laws, engineering constraints, and task structure into data-driven models in various forms, thereby improving learning efficiency and guaranteeing the physical feasibility and interpretability of the outputs. According to the manner in which knowledge is injected, existing studies can be grouped roughly into four categories. (i) Data pre-/post-processing, which applies physics-based corrections before data enter the model or after it produces an output; for example, Zhou et al. used high-fidelity physical simulation to generate reliable synthetic training data for deep-learning imaging in complex scattering media [16]. (ii) Training optimization, which introduces physical information or constraints into the loss or reward function; a representative example is the Physics-Informed Neural Network (PINN), which embeds governing partial differential equations into the loss so that the model satisfies physical constraints while fitting the data [17]. (iii) Network-architecture design, which embeds a physical-dynamics module into the network; for instance, Greydanus et al. built a Hamiltonian-based architecture that learns energy conservation automatically [18]. (iv) Hybrid methods, which split the problem into analytically tractable and intractable parts handled by a physical model and a learning model, respectively; Shi et al. combined an analytical gait controller with a learned residual module for legged locomotion [19].

Among these four categories, the training-optimization method, which introduces physical information or constraints directly into the reward function, offers the following advantages: it does not alter the model structure, is simple to implement, and

is highly compatible with existing reinforcement-learning algorithms; at the same time it deeply embeds physical knowledge into the training process, effectively constraining the behavior space of the agent and markedly improving data efficiency and generalization. For these reasons it has become one of the most widely used physics-guided strategies at present.

Targeting complex-terrain scenarios such as soft-soil sinkage in extraterrestrial exploration, and taking a wheeled mobile robot as the research object, this paper proposes an end-to-end motion-control method based on physics-informed reinforcement learning. Relying only on sensor data and local coordinate information, the method enables the robot to autonomously escape from entrapment and safely reach the target point; meanwhile, by designing physical-constraint reward terms, engineering knowledge is incorporated into the learning process, effectively constraining the reasonableness of the actions output by the robot.

2 End-to-End Motion-Control Method Based on Physics-Informed Reinforcement Learning

2.1 End-to-End Motion-Control Intelligent Model

The end-to-end motion-control model proposed in this paper adopts an encoder-aggregator-decoder architecture, the overall structure of which is shown in Fig. 1.

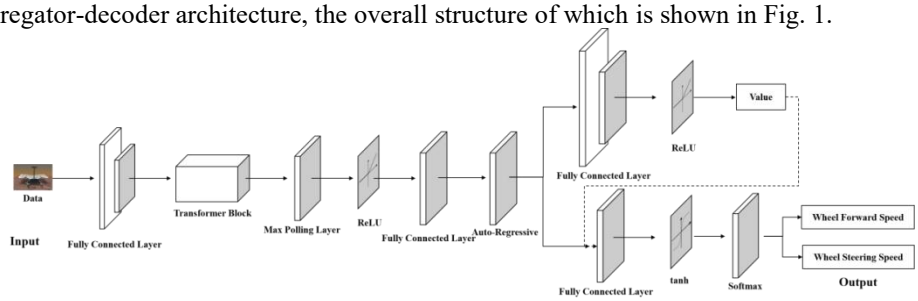

Fig. 1. Transformer-based encoder-decoder end-to-end motion-control intelligent model.

The model consists of an encoder, an aggregator, and a decoder. The encoder extracts features from the input perception information (robot pose, wheel state, and sinkage-detection signals) by cascading two fully-connected layers (64 and 32 nodes), a single Transformer block, and a max-pooling layer. The Transformer block uses two parallel attention heads of dimension 4 to model global dependencies within the feature sequence, while the max-pooling layer compresses the sequence into a fixed-length vector; all encoder layers use ReLU. The aggregator then fuses these features via a 256-node fully-connected layer and a 128-dimensional auto-regressive component that captures short-term temporal dependencies at low computational cost. The decoder adopts an Actor-Critic architecture: the Actor (a 256-unit layer with tanh, followed by Softmax) outputs a discrete action distribution from which the highest-probability action is selected, and the Critic (two layers of 512 and 256 nodes with ReLU) estimates

the state value to guide policy optimization. Through end-to-end learning, the model maps perception inputs directly to control commands.

2.2 Design of the Physics-Informed Reward Function

To improve the generalization capability, learning efficiency, and engineering feasibility of the model, this paper designs a Physics-Informed Reward Function (PIRF). Its design draws on the idea of the Physics-Informed Neural Network (PINN), which embeds partial differential equations into the loss function to constrain the network output and thereby explicitly encode physical laws [17]; here, engineering physical constraints (such as steering-angle limits and wheel-speed limits) are incorporated into the reinforcement-learning reward function in the form of soft-constraint penalty terms. Unlike directly applying a hard truncation to the output, this approach guides the agent to proactively avoid limit-exceeding actions while optimizing the task objective, all while keeping the policy differentiable and training stable, so that its output balances task performance and physical feasibility, thereby improving the generalization capability and engineering practicality of the model in complex physical environments.

The PIRF consists of two parts: first, a two-stage escape-navigation reward that encodes structured task knowledge to guide the robot in completing the composite task objective; and second, a physics-constraint reward that constrains the actions output by the robot to a feasible region satisfying engineering requirements.

Escape-Navigation Two-Stage Reward. In the soft-soil-sinkage scenario of extraterrestrial exploration, the robot must complete a composite task composed of escape and navigation. To this end, this paper designs a two-stage reward function and switches it dynamically according to whether the robot is in an entrapped state.

(1) Escape reward. The escape reward is activated when the robot is in an entrapped state and is used to guide the agent to escape the entrapment region as quickly as possible. It consists of an intermediate reward and a sparse reward: the intermediate reward gives positive feedback for an increase in the robot's speed, while the sparse reward rewards the terminal event of escape completion.

$$\begin{cases} R_{\text{escape-inter}} = -b + a \times \left(\boldsymbol{v}_t^{\,2} - \boldsymbol{v}_{t-1}^{\,2} \right) \\ R_{\text{escape-sparse}} = c \\ R_{\text{escape}} = R_{\text{escape-inter}} + R_{\text{escape-sparse}} \end{cases} \tag{1}$$

where $R_{\text{escape-inter}}$ is the escape intermediate reward, $R_{\text{escape-sparse}}$ is the escape sparse reward, R_{escape} is the escape reward, $-b$ is the negative-feedback constant term, a is the intermediate-reward coefficient, c is the sparse-reward constant, $\boldsymbol{v}_t$ is the current speed of the robot, and $\boldsymbol{v}_{t-1}$ is the speed of the robot at the previous step.

(2) Target reward. The target reward is activated when the robot has not become entrapped or has already completed escape, and is used to guide the agent to reach the target point efficiently. It likewise consists of an intermediate reward and a sparse reward: the intermediate reward gives positive feedback for the reduction in distance

between the robot and the target point, while the sparse reward rewards the terminal event of reaching the target point.

$$\begin{cases} R_{\text{goal-inter}} = -b' + a' \times (\|(x_{t-1}, y_{t-1}) - (x_{\text{goal}}, y_{\text{goal}})\| - \|(x_t, y_t) - (x_{\text{goal}}, y_{\text{goal}})\|) \\ R_{\text{goal-sparse}} = c' \\ R_{\text{goal}} = R_{\text{goal-inter}} + R_{\text{goal-sparse}} \end{cases} \tag{2}$$

where $R_{\text{goal-inter}}$ is the target intermediate reward, $R_{\text{goal-sparse}}$ is the target sparse reward, R_{goal} is the target reward, $-b'$ is the negative-feedback constant term, a' is the intermediate-reward coefficient, c' is the sparse-reward constant, (x_{t-1}, y_{t-1}) is the position of the robot at the previous step, (x_t, y_t) is the current position of the robot, and $(x_{\text{goal}}, y_{\text{goal}})$ is the position of the target point.

Physical-Constraint Reward. The physical-constraint reward is used to lead the robot's actions to the engineering-feasible region: when state parameters such as wheel speed and steering angle exceed the preset limits, a negative reward related to the degree of limit exceedance is applied, thereby continuously penalizing limit-exceeding actions during training and guiding the model to output control commands that satisfy the physical constraints.

$$R_{\text{constraint}} = -d \times \left| \boldsymbol{A} - \boldsymbol{A}_{\text{constraint}} \right| \tag{3}$$

where $R_{\text{constraint}}$ is the constraint reward and $-d$ is corresponding coefficient, $\boldsymbol{A}$ is a robot state parameter, and $\boldsymbol{A}_{\text{constraint}}$ is the corresponding physically-compliant state parameter of the robot.

Fused Reward Function. To coordinate the dual task objectives of escaping and reaching the target point, the following fused reward function, i.e. the reward function oriented toward safe mobility, is constructed:

$$R = \alpha \cdot R_{\text{escape}} + \beta \cdot R_{\text{goal}} + R_{\text{constraint}} \tag{4}$$

where α and β denote the weighting coefficients of the escape reward and the target reward, respectively. When the robot is entrapped: $\alpha = 1, \beta = 0$ and when the robot has not encountered entrapment or has escaped: $\alpha = 0, \beta = 1$.

2.3 Training Algorithm Based on the PPO Policy

This paper uses the PPO reinforcement-learning algorithm [20] to train the end-to-end motion-control model. First, the agent continuously interacts with the environment to generate training data (the agent outputs an action command a_t to the environment, and the environment returns the state parameters s_t and generates a reward r_t fed back to the agent), with the optimization objective of maximizing the cumulative reward obtained by the agent during its interaction with the environment. Second, the PPO

algorithm continuously updates and iterates the policy function of the agent model so as to learn the optimal policy for the corresponding environment; the physics-constraint-based reward function is evaluated through the designed reward mechanism to generate the reward for the current state.

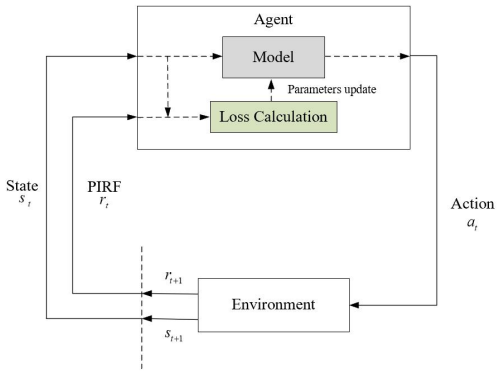

Fig. 2. Reinforcement-learning flow of the model.

3 Experimental Design and Result Analysis

3.1 Environment and Parameter Settings

The experiments in this paper are conducted on the AGX Dynamics high-fidelity physics-simulation engine. This engine provides accurate rigid-body dynamics, a wheel-terrain contact model, and soft-soil-sinkage simulation capability, and can fairly realistically characterize the motion behavior of an extraterrestrial-exploration robot over complex terrain.

The training scenario is set as an initial single-wheel-entrapment condition, in which the left-front wheel of the robot is trapped in a soft-soil region and the target point is set at coordinates (-3, -3). This scenario requires the agent to first complete escape and then navigate to the target point. The robot states collected during training include position, speed, steering angle, steering-angle rate, and the sinkage-detection signal.

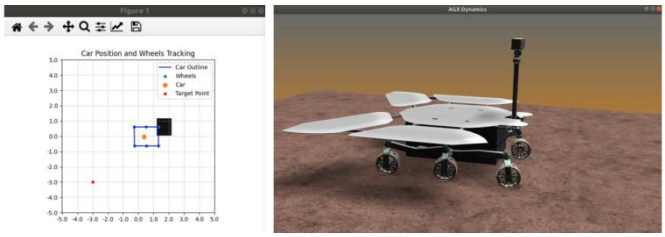

Fig. 3. Training scenario. Left: two-dimensional schematic, where the black area is the soft-soil-sinkage region, the robot's left-front wheel is entrapped, the positions of the six wheels and of the robot are indicated, and the asterisk denotes the target-point position. Right: corresponding three-dimensional schematic.

3.2 Test Scenarios

To verify the generalization capability of the model, complex sinkage regions and target points different from the training scenario were further set up as test scenarios, including a fully non-entrapped traverse, in-motion entrapment, and initial entrapment.

Table 1. Test scenarios.

Training scenario	Test scenario		Target point
Initial entrapment: single-wheel (left-front)	Fully non-entrapped traverse		(-3, -3)
	In-motion entrapment		
	Initial entrapment:	single-wheel	
		double-wheel	
		triple-wheel	

In the initial-entrapment scenario, each type of test scenario further covers entrapment configurations at different wheel positions. For example, single-wheel entrapment is divided into six cases (left-rear, left-middle, left-front, right-front, right-middle, right-rear); double-wheel entrapment is divided into 15 combinations such as left-rear + left-middle, left-rear + left-front, and left-rear + right-front; and triple-wheel entrapment includes 20 typical configurations. This design is intended to comprehensively evaluate the generalization and adaptation capability of the model to different sinkage patterns.

3.3 Test Experiment Validation

The test scenarios were set up, and the trained model was deployed in these scenarios to carry out test experiments for the fully non-entrapped traverse, in-motion entrapment, and initial entrapment, respectively.

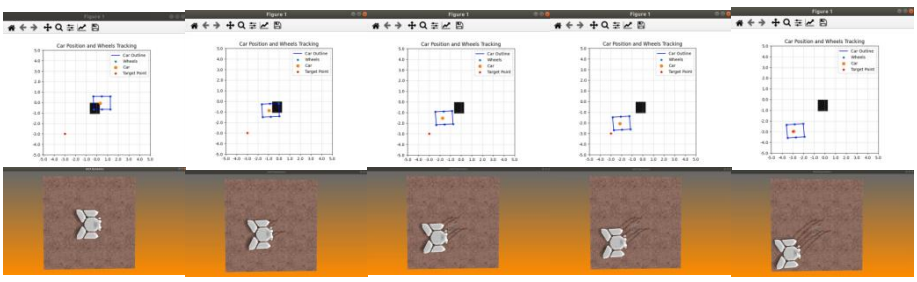

Fig. 4. Schematic of the test experiment. In sequence: the initial entrapment state, the escape process, successful escape, navigation toward the target point, and successful arrival at the target point.

From left to right, Fig. 4 successively shows the complete process of the mobile robot in an initial-entrapment (single-wheel, right-rear) test experiment. Throughout this process, the wheel-drive and steering actions of the robot are output directly end-to-end by the proposed model.

To improve the reliability of the experimental results, no fewer than three repeated trials were conducted for each type of test scenario, and the average of the data from

multiple trials was taken as the experimental result for that scenario. In each test trial, it was recorded whether the mobile robot successfully escaped, whether it successfully reached the target point, and the average number of motion-control steps.

3.4 Experimental Results

Through the above test experiments, the experimental results were obtained as shown in Table 2.

Table 2. Experimental results.

Test scenario		Escape success rate	Target-arrival rate	Avg. control steps
Fully non-entrapped traverse		100%	100%	89
In-motion entrapment		100%	100%	122
Initial entrapment:	single-wheel	100%	100%	95
	double-wheel	100%	100%	107
	triple-wheel	100%	100%	115

The above experimental results show the following.

(1) After being trained on the training scenario, the motion-control model achieves a 100% target-arrival rate in the fully non-entrapped traverse test scenario. In the in-motion-entrapment and initial-entrapment test scenarios, both the escape success rate and the target-arrival rate are 100%. These results indicate that the model possesses good end-to-end motion-control capability and generalization capability in complex sinkage scenarios, verifying the effectiveness of the proposed safe-mobility-oriented end-to-end motion-control method.

(2) The average number of steps increases with the complexity of the entrapment. This trend is consistent with expectations: the more wheels are entrapped, the more complex and time-consuming the coordinated actions required to escape. The number of steps in the in-motion-entrapment scenario (122 steps) is higher than in each of the initial-entrapment scenarios, because the robot must first complete a segment of non-entrapped navigation, encounter entrapment midway, and then escape, giving the overall task a longer course.

3.5 Analysis of Key State-Quantity Evolution Curves

This section further analyzes the evolution curves of key state quantities of the model in typical test scenarios. The following figures present the complete-process data of the robot from entrapment to escape and then to arrival at the target point, for an initial-entrapment scenario.

Fig. 5 shows the initial state and the motion trajectory of the mobile robot in the test scenario: the left-middle, right-front, and right-rear wheels are all trapped in the soft-soil region marked in black; the asterisk denotes the target-point position; the robot's motion trajectory is a continuous, smooth curve connecting the starting position with the target point, indicating that the model successfully realizes end-to-end motion control from the initial position to the target point.

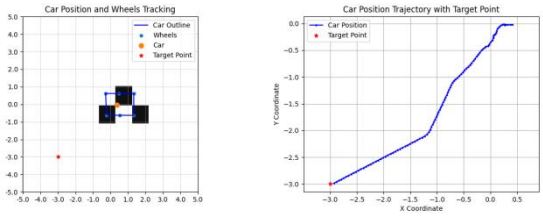

Fig. 5. Left: initial state. Right: motion trajectory in the test scenario.

Fig. 6 presents the steering-angle and steering-angle-rate curves of the mobile robot. During the escape stage (approximately the first 40 simulation steps), the steering angle changes frequently and with large amplitude, indicating that the model adjusts the relative position of the wheels and the soft soil to seek an escape path; correspondingly, the steering-angle rate also exhibits several large fluctuations (with a peak of about 0.262 rad/s). After escape is completed, the steering-angle adjustment becomes smoother, the robot proceeds toward the target point along an approximately straight line, and the steering-angle rate converges to near 0. Throughout the process, the steering angle always remains within ±15° and the steering-angle rate does not exceed 0.262 rad/s, both strictly satisfying the preset physical limits, indicating that the PIRF can effectively constrain the physical feasibility of the policy output.

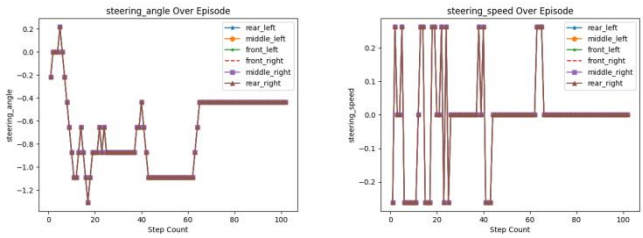

Fig. 6. Left: steering-angle curve. Right: steering-angle-rate curve.

Fig. 7 presents the velocity curve of the mobile robot. During the escape stage, the velocity fluctuates markedly as the robot adjusts its steering and drive to break free of the sinkage region. It then rises and stabilizes at about 0.06−0.07 m/s, indicating a smooth transition into the navigation stage, where the curve stays steady with no abrupt changes or limit exceedances. This confirms that the method effectively constrains the physical feasibility of the action output, ensuring safe and stable motion.

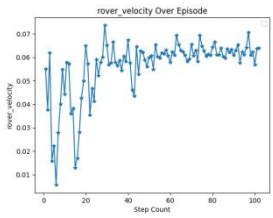

Fig. 7. Velocity curve.

The above analysis shows that the proposed method enables the agent to autonomously complete the composite escape-and-navigation task under strict

satisfaction of the physical constraints, and maintains good generalization capability and action feasibility under entrapment configurations not covered during training.

3.6 Ablation Experiment

To verify the effectiveness of the physical-constraint reward term in the PIRF, the following ablation comparison experiment was designed.

(1) Constrained model: adopts the complete PIRF fused reward function (task reward plus engineering-constraint penalty term).

(2) Unconstrained model: adopts only the task reward, with the engineering-constraint penalty term removed.

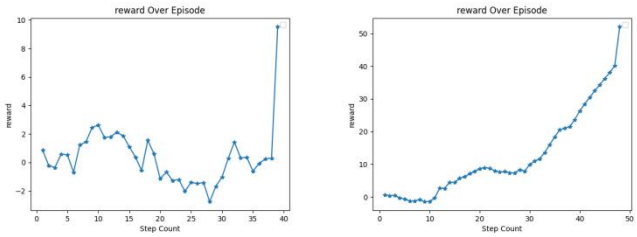

Fig. 8. Reward-function curves during model training. Left: reward-function curve of the unconstrained model. Right: reward-function curve of the constrained model.

As can be seen from Fig. 8, owing to the lack of guidance from physical mechanisms and engineering rules, the reward curve of the unconstrained model fluctuates continuously during training, reflecting an unstable training process. In contrast, the PIRF fused reward curve rises steadily. This comparison shows that introducing the physical-constraint penalty term plays a clear role in improving training stability, and further enhances the generalization capability of the model.

4 Conclusion

This paper addresses the safe-motion-control problem of extraterrestrial-exploration mobile robots in unknown, complex environments by proposing an end-to-end control method based on physics-informed reinforcement learning. Built on a Transformer encoder-decoder architecture, the method introduces a Physics-Informed Reward Function (PIRF) that jointly encodes two-stage escape-navigation task knowledge and engineering physical constraints (steering angle, steering-angle rate, wheel speed) as reward terms. Multi-scenario validation on the AGX Dynamics platform shows that, under entrapment configurations unseen during training, the model attains a 100% mission success rate, while the steering angle, steering-angle rate, and wheel speed all stay within the preset physical limits — confirming both strong generalization and effective constraint of the policy output within hardware safety boundaries. The ablation comparison further shows that the physical-constraint penalty term turns an oscillating reward curve into a steadily converging one, improving learning efficiency and training stability. Overall, the method offers technical support for the safe mobility of such robots over complex terrain.

References

1. Wu, W., Yu, D.: Development and future key technologies of deep-space exploration. Journal of Deep Space Exploration 1(1), 5-17 (2014).
2. Zhang, R., Huang, J., He, R., et al.: A review of the development of asteroid exploration. Journal of Deep Space Exploration 6(5), 417-423,455 (2019).
3. Wu, K., Li, W., Zhang, Z., et al.: Progress in autonomous-intelligence technology for landing and roving exploration of extraterrestrial bodies. Journal of Astronautics 43(9), 1129 (2022).
4. Wu, W., Liu, J., Tang, Y., et al.: China's lunar exploration program. Journal of Deep Space Exploration 6(5), 405-416 (2019).
5. Yu, D., Wu, X., Wu, W.: A review of the technical development of China's lunar exploration program. Journal of Deep Space Exploration 3(4), 307-314 (2016).
6. Xing, Y., Liu, X., Teng, B., et al.: Autonomous local obstacle-avoidance planning for lunar-surface roving exploration. Control Theory & Applications 36(12), 2042-2046 (2019).
7. Zhang, X., Gao, H., Zhao, J., et al.: A survey of autonomous-driving technology based on deep learning. Journal of Tsinghua University (Science and Technology) 58(4), 438-444 (2018).
8. Wang, T., Luo, Y., Liu, J., et al.: End-to-end self-driving approach independent of irrelevant roadside objects with auto-encoder. IEEE Transactions on Intelligent Transportation Systems 23(1), 641-650 (2020).
9. Chen, Y., Tian, D., Lin, C., et al.: A survey of end-to-end autonomous-driving systems. Journal of Image and Graphics 29(11), 3216-3237 (2024).
10. Pan, F., Bao, H.: Research progress on reinforcement-learning-based autonomous-driving control technology. Journal of Image and Graphics 26(1), 28-35 (2021).
11. Chen, L., Wu, P., Chitta, K., et al.: End-to-end autonomous driving: challenges and frontiers. IEEE Transactions on Pattern Analysis and Machine Intelligence 46(12), 10164-10183 (2024).
12. Zhang, H.J., Xing, Y., Yang, W.Q.: Terrain classifier based on contrastive learning for extraterrestrial surface exploration. In: International Conference on Computational Vision and Robotics, pp. 144-153. Springer Nature Switzerland, Cham (2024).
13. Lee, J., Schroth, L., Klemm, V., et al.: Evaluation of constrained reinforcement-learning algorithms for legged locomotion. arXiv preprint arXiv:2309.15430 (2023).
14. Belmonte-Baeza, A., Cazorla, M., García, G.J., et al.: Autonomous legged mobile manipulation for lunar-surface operations via constrained reinforcement learning. In: 2025 International Conference on Space Robotics (iSpaRo), pp. 588-595. IEEE, Sendai (2025).
15. Cao, H., Mao, Y., Sha, L., et al.: Physics-regulated deep reinforcement learning: invariant embeddings. In: International Conference on Learning Representations, pp. 3712-3756 (2024).
16. Zhou, B., et al.: Learning from better simulation: creating highly realistic synthetic data for deep learning in scattering media. Advanced Photonics Nexus 4(5), 056007 (2025).
17. Raissi, M., Perdikaris, P., Karniadakis, G.E.: Physics-informed neural networks: a deep-learning framework for solving forward and inverse problems involving nonlinear partial differential equations. Journal of Computational Physics 378, 686-707 (2019).
18. Greydanus, S., Dzamba, M., Yosinski, J.: Hamiltonian neural networks. In: Advances in Neural Information Processing Systems, vol. 32, pp. 15353-15363. Curran Associates, Inc., Vancouver (2019).

19. Shi, J., Li, Q., Su, Z.: FCNAUP: learning hybrid locomotion skills - learn to exploit residual actions and modulate model-based gait control. Frontiers in Robotics and AI 10, 1180408 (2023).
20. Schulman, J., Wolski, F., Dhariwal, P., et al.: Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347 (2017).