aerospace

MDPI

Article

Research on Dual-Arm Control of Lunar Assisted Robot Based on Hierarchical Reinforcement Learning under Unstructured Environment

Weiyan Ren, Dapeng Han and Zhaokui Wang *

School of Aerospace Engineering, Tsinghua University, Beijing 100084, China; rwy17@mails.tsinghua.edu.cn (W.R.); dphan@tsinghua.edu.cn (D.H.)
* Correspondence: wangzk@tsinghua.edu.cn

Abstract: When a lunar assisted robot helps an astronaut turn over or transports the astronaut from the ground, the trajectory of the robot's dual arms should be automatically planned according to the unstructured environment on the lunar surface. In this paper, a dual-arm control strategy model of a lunar assisted robot based on hierarchical reinforcement learning is proposed, and the trajectory planning problem is modeled as a two-layer Markov decision process. In the training process, a reward function design method based on the idea of the artificial potential field method is proposed, and the reward information is fed back in a dense reward method, which significantly reduces the invalid exploration space and improves the learning efficiency. Large-scale tests are carried out in both simulated and physical environments, and the results demonstrate the effectiveness of the method proposed in this paper. This research is of great significance in respect of human–robot interaction, environmental interaction, and intelligent control of robots.

Keywords: lunar assisted robot; hierarchical reinforcement learning; dual-arm control; reward functions

Citation: Ren, W.; Han, D.; Wang, Z. Research on Dual-Arm Control of Lunar Assisted Robot Based on Hierarchical Reinforcement Learning under Unstructured Environment. *Aerospace* **2022**, *9*, 315. https://doi.org/10.3390/aerospace9060315

Academic Editor: Shuang Li

Received: 2 April 2022
Accepted: 8 June 2022
Published: 10 June 2022

1. Introduction

Outer space is an ultimate field for the application of robotics technology. After 50 years of manned landings on the moon, lunar exploration has once again become a hot spot in the field of manned spaceflight. The United States proposed the Artemis Program, China proposed the Chang'e Program and plans to launch the Chang'e 6 probe, India launched the Chandrayaan-2 probe, and Russia plans to launch the Luna 25 probe. The European Space Agency plans to land on the moon and establish a lunar base by 2025. The moon is the closest celestial body to the Earth in the solar system, and it is an extraterrestrial resource-gathering area where contemporary aerospace technology can effectively build a transportation system. The new round of lunar exploration planning is no longer limited to the display of technical capabilities, but is based on the development and utilization of cislunar space resources; it also comprises on-the-spot investigation of lunar construction and cislunar space industrialization, and it is more important for the frontier exploration of the expansion of human civilization to cislunar space. Lunar exploration involves complex system engineering. The lunar surface is far away, the environment is extreme, and human resources are scarce. Therefore, artificial-intelligence-based robotics will play an important role in manned lunar exploration activities [1,2]. Robot systems, which assist astronauts in fine operations and other tasks, play an important role in manned lunar landing missions. The intelligent sensing and dual-arm control level of a lunar assisted robot in an unstructured environment are the main factors limiting its application range [3]. Zhang introduced a method, CNN-vote, that applied convolutional neural networks to predict human intentions behind a single action, for improving robotic intelligence by training it to understand human intentions [4]. Therefore, it is of great theoretical significance

and practical application value to study human–computer interactions and environmental interactions between lunar assisted robots and astronauts in unstructured environments.

Research on the dual-arm control of robots is the key to promoting the practical application of lunar assisted robots. The commonly used manipulator control requires sensing and modeling of the astronaut's environment and manipulator [5]. This method has two major challenges: (1) it is difficult to model the unstructured environment on the lunar surface, and the solution efficiency is low; (2) when the environmental parameters change, the original model cannot easily solve the control problems in the new environment. In recent years, data-driven manipulator control methods have been proposed [6]. Li proposed a trajectory planning method for space station manipulators based on deep reinforcement learning, and combined this with the artificial potential field method to improve the convergence of training [7]. Dong developed a new autonomous incremental visual servo control law for the robotic manipulator to capture a non-cooperative target, where the control input is the incremental joint angle, to avoid multiple solutions in the existing inverse kinematics [8]. Cristian proposed a force control framework based on reinforcement learning on the control of rigid robot manipulators, combining traditional force control methods with the Soft Actor-Critic (SAC) algorithm to avoid damage to the environment during in the process of approaching the environment [9]. Xiong developed and compared an end-to-end deep reinforcement learning (DRL) strategy and a hybrid DRL strategy in controlling a cable-driven parallel robot; this study demonstrated that hybrid DRL strategies offer an alternative to accomplishing robotic manipulation tasks [10]. Asad used the PPO algorithm to conduct an in-depth study on a robot grasping task, and designed a reward and punishment function (RPF) with intensive rewards, but the RPF and requirements of the task were relatively simple [11]. As for the obstacle avoidance of the robotic arm in the process of moving, Evan proposed to use a collision detection method to punish the robot, but the robot did not receive a penalty signal when approaching the obstacle, which lengthened the training time [12]. Zhu presented a predictive visual servo kinematic control scheme for a robotic manipulator with an eye-in-hand configuration to perform autonomous capture of a non-cooperative space target with unknown motion [13]. Kei proposed a trajectory planning method for manipulators working in a constrained space, so as to avoid obstacles outside the constrained space, but the definition of the constrained space had a certain particularity [14]. Although reinforcement learning (RL) has achieved success in some fields, the problem of "dimension disaster" and long-term reliability allocation in robot control research leads to slow convergence or even difficult convergence of reinforcement learning training.

Research on the dual-arm control of redundant-degree-of-freedom robots is a very challenging problem. In an unstructured environment, the RL method of reinforcement learning is more challenging than the traditional reinforcement learning decision-making problem [1,15]. First of all, the trajectory of the dual arms of a redundant-freedom robot is of an exponential order of magnitude in joint action space. The challenge lies in the need for the robot to explore and learn the optimal trajectory in high-dimensional state space and maximum action space. At present, there is no mature solution or reinforcement learning algorithm. Secondly, in a three-dimensional space composed of astronauts, lunar rocks, and lunar robots, it is difficult for agents to converge in the intensive learning training of multi-objective interaction such as targets approaching, obstacle avoidance, and energy consumption reduction. Therefore, the two major challenges facing the dual-arm control problem of lunar assisted robots are as follows:

(1) How to avoid the exploration and decision-making of dual-arm joints in the maximum action space by constructing hierarchical strategies;

(2) How to set the reward function reasonably in multi-objective interaction to quickly explore the optimal strategy.

Based on the previous research of fine-grained posture recognition and visual tracking of astronauts [16–18], the high-dimensional data of astronauts' joints are simplified as singular postures, which effectively reduces the dimensionality by converting infinite postures

of the human body into 20 posture models. In this paper, a hierarchical reinforcement learning framework for a robot dual-arm control strategy model (HRL-DCS) based on hierarchical reinforcement learning is proposed. The trajectory planning problem is modeled as a two-level Markov decision process. The purposes of this paper are as follows: (1) to transform the nonlinear problem in the high-dimensional state space of the manipulator into a nonlinear problem in the low-dimensional state space and an approximate linear problem in the high-dimensional state space; (2) the dual-arm control problem of the lunar assisted robot is transformed into a sequential decision-making problem. The hierarchical reinforcement learning framework proposed in this paper solves the strategy representation problem by decomposing the strategy representation into sequential decision problems. The proposed methods are validated by simulation using a dual-arm robot manipulator.

2. Hierarchical Reinforcement Learning

In this study, the dual-arm control problem of a lunar assisted robot is modeled as a multi-objective Markov decision process (MOMDP). Considering the unstructured environment on the lunar surface, during the interaction between the assistant robot and astronauts, the agent needs to complete the goals of target approaching, obstacle avoidance, and task completion.

2.1. Multi-Objective Markov Decision Process

We model the dual-arm control problem of the lunar assisted robot as a multi-objective Markov decision process, and use HRL to optimize the dual-arm trajectory. MOMDP can be defined as a six-tuple $M = (I, S, A, P, R, \gamma)$, where I is the task space and a specific task $I \in I$ is sampled from the task distribution $\rho(I)$; S is the state space, and $\rho(s_0|I)$ is the initial state probability distribution; A is the action space; M: $S \times A \rightarrow S$ is a state transition function, $P(s_{t+1}|s_t, a_t)$ represents the probability that an action executed in a state transitions to a state; R: $S \times A \times I \rightarrow R$ is the reward function, and $\gamma \in [0, 1]$ is the reward attenuation factor.

We define the state $s_t = \left\{P_{AJ_i}, P_{BJ_i}, P_{Gj}, P_P, P_E, P_R\right\} (i = 1, 2, 3, 4, 5)$, where P_{AJ_i} denotes the parameters of the left arm joint of the lunar assisted robot, P_{BJ_i} denotes the parameters of the right arm joint of the robot, the position of the target area is P_{G_j}, the center of gravity of astronauts is P_P, the position of obstacles is P_E, and the position of the robot base is P_R. In the t-th interaction, the robot will accept a task I_t satisfying $I_t \in I$, and the corresponding feedback $r_t \in R$, the interaction time step is $T(I)$; $T(I)$ is the maximum length of the set episode for a certain task (the interaction process will end when the agent reaches the termination state, or it may end when the number of steps exceeds the maximum number of set episodes). In this case, I is the set of all target tasks of the robot, R is the set of all possible feedback, and the interaction state satisfies the Markov decision process.

In the above MOMDP process, given the history of the task $s_t = \{I, s_0, a_0, r_0, s_1, \cdots, s_t\}$, the conditional probability is $\pi : S \times A \rightarrow R$, where $\pi(a_t|s_t)$ represents the probability of the robot dual-arm control strategy in the t-th interaction. At this point, $\xi = (s_0, a_0, r_1, s_1, a_1, r_2, \cdots, s_T)$ is an episode or a trajectory of the robot interacting with the environment and $r_t \in \mathbb{R}$ is a reward function. The goal of HRL is to find a strategy function π^* that maximizes the expected cumulative reward value of the trajectory ξ obtained during the strategy π,

$$\begin{aligned} \pi^* &= \underset{\pi}{\operatorname{argmax}} \underset{\pi \in \Pi}{\eta(\pi)} \\ &= \underset{\pi}{\operatorname{argmax}} \underset{\pi \in \Pi}{E_{\xi \sim P_\xi^\pi}} \left[\sum_{t=0}^{T} \gamma^t r(s_t, a_t) \right] \end{aligned} \tag{1}$$

where $\eta(\pi)$ refers to the probability distribution satisfied by the trajectory ξ when the lunar robot interacts with the environment by using strategy π. $r(s_t, a_t) \in [0, r_{\max}]$ represents the reward function related to environmental feedback; γ is a super parameter that controls how future rewards decay; $\xi \sim P_\xi^\pi$ indicates that the trajectory ξ is generated by strategy π. In this study, a model-free method is adopted.

2.2. Multi-Objective Hierarchical Reinforcement Learning

The purpose of this study is to learn the dual-arm control strategy of a lunar robot through HRL to achieve the optimization goal of this study Specifically, we treat the original Markov Decision Process (MDP) problem as a multi-level MOMDP process; that is $\left(M^1, M^2, M^3, \cdots, M^k\right)$, (in this chapter, k is the number of layers of policies, the same below), with the following characteristics:

(1) The time granularity of each layer of strategy is different; that is, the number of steps N^i used to complete MOMDP actions in each layer of HRL is different, assuming $N^1 = 1$. Moreover, the length of the episode of each MOMDP layer is the time accuracy ratio of the upper and lower MOMDP layers.

(2) S and I of each layer of MOMDP can be mapped to the same metric space, so a distance function is needed to calculate the distance between the target state and the current state. Therefore, the reward function of each layer is related to the distance function, but the specific correlation is related to the determination of specific objectives in multi-objective optimization, as described below.

(3) The action space of high-level MOMDP is the task space of the low-level; that is, the strategic action of the high-level corresponds to the task goal of the low-level; that is, $A^{i+1} = I^{\mathrm{i}}$. In particular, the highest-level task is the original task ($I^k = I$), and the lowest-level action is the original action of the manipulator ($A^1 = A$).

3. Methodology

In order to solve the problem of the dual-arm control of robots with redundant degrees of freedom, we propose a training framework based on a layered strategy, and use a two-layer strategy to represent the robot dual-arm control method. Specifically, (1) the high-level strategy generates discrete sequence sub-targets in the state space of the end position of the low-dimensional manipulator, learns and explores the sub-targets, and optimizes the motion trajectory of the robot arms from a long-term perspective and the whole strategy trajectory. (2) The high-level strategy encourages diversity recommendation and full exploration of objectives, aiming to deal with the balance between exploration and utilization and avoid local optimization. (3) The low-level strategy is responsible for achieving the sub-goal parameterized by the high-level strategy; that is, optimizing the internal reward function in the high-dimensional double-arm joint space, and the low-level strategy is to balance the goal requirements such as goal approaching, obstacle avoidance, and energy consumption reduction.

3.1. Network Framework

In this paper, the HRL is proposed to solve the dual-arm control problem of a lunar robot in an unstructured environment. The proposed HRL-DCS has a two-layer framework; namely, the high-level subgoal-generation network (SGN) and the low-level network (LLN), as shown in Figure 1.

At a certain time t, the high-level SGN obtains the observation state s_t from the environment, and the high-level strategy π_h generates sub-targets $g_t \sim \pi_h(\cdot|s_t) \in \mathbb{R}$ according to the task code I and the observation state s_t. SGN runs at a relatively abstract level, decomposes complex tasks into sub-target sequences, generates diversified sub-targets g_t for the underlying strategy LLN, and controls the lunar robot to complete all tasks with reference to certain behavior sequences. The high-level strategy is mainly used to evaluate the target task and optimize the trajectory of the robot arms from a long-term perspective and the entire strategy trajectory. Specifically, the high-level strategy encourages the diverse recommendation of robot dual-arm targets by generating multiple sub-targets. This is because, if a strategy is blindly chosen without sufficient exploration, local optima or non-convergence may occur. However, when there are clear goals, blindly increasing the diversity of exploration may reduce the utilization efficiency. Therefore, the typical problem in reinforcement learning is faced in the research of robot dual-arm control in this paper;

that is, the high-level strategy has to deal with the balance of exploration and exploitation, in order to optimize the robot's dual-arm trajectory from a long-term perspective.

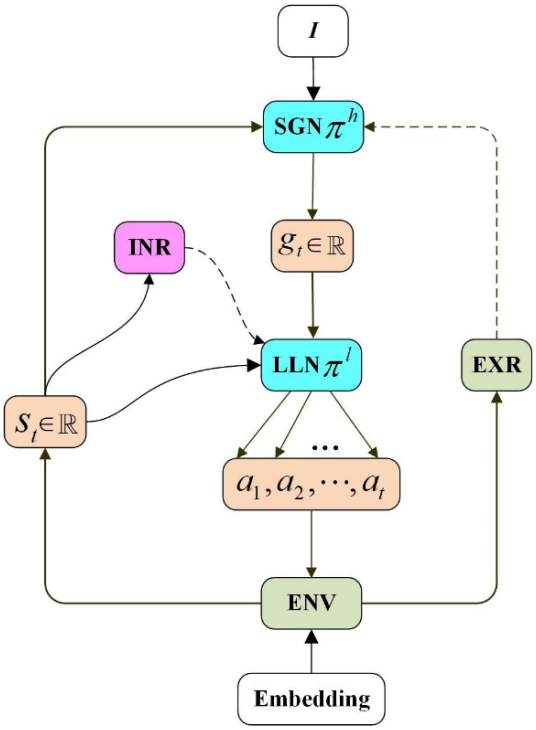


Figure 1. HRL-DCS network framework.

In the LLN, the underlying strategy π_l generates atomic actions $a_t \sim \pi_l(\cdot|s_t, g_t, a_{t-1})$ according to diversity sub-targets g_t, current state s_t, and previous actions a_{t-1}. Atomic action a_t is used to interact with astronauts and obstacles and obtain the corresponding reward r_t^l. Based on the state s_t and action a_t, the observed state is transferred to the new state s_{t+1}. The low-level strategy is mainly responsible for completing the sub-goals parameterized by the high-level strategy; that is, optimizing the internal reward function in the high-dimensional joint space. Specifically, the low-level strategy includes a target approach strategy and an obstacle avoidance model based on an artificial potential field. Among them, the target approach strategy is mainly used to evaluate the probability that the ends of the arms will reach the target area in a certain state. The obstacle avoidance model is used to maximize the intrinsic reward function for obstacle avoidance. The low-level strategy mainly balances the goal approach reward and obstacle avoidance requirement. SGN provides a parameterized internal reward function $r_t^l = (s_t, g_t, a_t)$ for LLN, and at the same time, SGN collects corresponding reward r_t and stores transition data tuples (s_t, g_t, a_t, s_{t+1}) of moment t for reinforcement learning. Similar to traditional HRL, SGN moves at a high-level time scale ($N^h = N$); that is, only at the time $t = 0, N, 2N, \cdots$ does the high-level SGN generate sub-target g_t. In this study, an embedding observation module is used to process image data, and the original input image in time t environment is mapped into abstract state vector s_t in N-dimensional embedding space.

3.2. Reward Functions

In Figure 1, the reward functions in HRL-DCS include high-level and low-level aspects, namely external reward (EXR) and internal reward (INR) functions. In the high-level strategy, the goal of SGN is to optimize the overall performance of the lunar robot's dual-arm trajectory. Therefore, we take EXR in N time steps as the instantaneous reward of SGN.

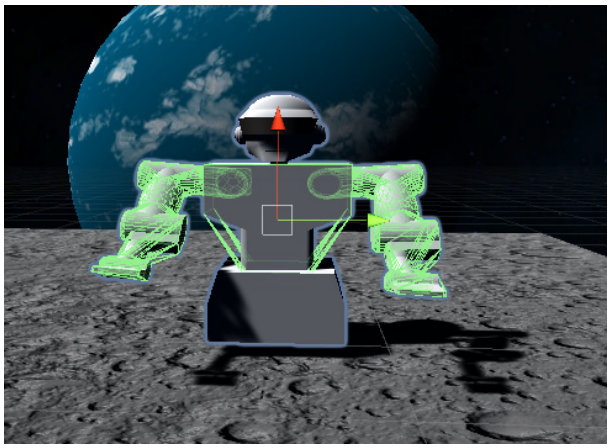
In the low-level strategy, LLN only focuses on how to accomplish the sub-goal of high-level policy parameterization through the interaction between agents and environment. The optimization goal of LLN is to complete target approaching, obstacle avoidance, and

rescue tasks with both arms of the lunar robot. LLN uses an internal reward function (shown in Equation (2)), and INR encourages the exploration of control strategies that simultaneously satisfy target proximity (the first part of Equation (2)), obstacle avoidance reliability (the second part of Equation (2)), energy saving (the third part of Equation (2)), and time reward function (the fourth part of Equation (2)).

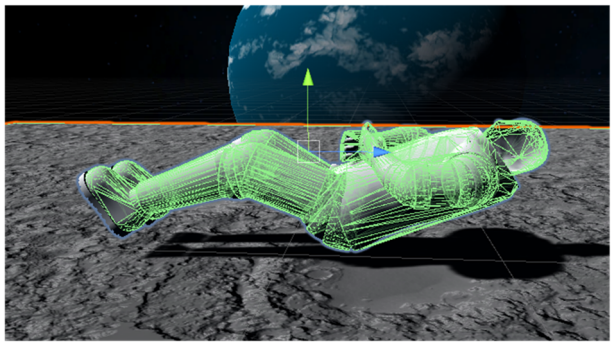
$$\begin{aligned} r_t^l &= \beta \exp\left[D_m^L(s_{t-1}, a_{t-1}) - D_t^L(s_t, a_t)\right] - \rho \log\left(\sum_{i=1}^{i}\sum_{j=1}^{j} \frac{\xi}{D_{j_i E_j}^L(s_t, a_t)} + 1\right) \\ &\quad - \varphi \sum_{i=1}^{n} \Delta\omega_i + \left(1 - \frac{0.9\phi}{\phi_{\max}}\right) \end{aligned} \tag{2}$$

The first part of INR is the target approach function: in this equation, Euclidean distance is used to measure the distance in three-dimensional space. At a certain time t, $D_t^L(s_t, a_t)$ represents the distance between the end of a lunar robot arm and astronauts, and $D_m^L(s_t, a_t)$ is the minimum value of the distance, $D_m^L(s_t, a_t) = \min\left[D_m^L(s_t, a_t), D_t^L(s_t, a_t)\right]$, $t > 1$, and the value of $D_m^L(s_t, a_t)$ is constantly updated with training. When $D_t^L(s_t, a_t) < D_m^L(s_{t-1}, a_{t-1})$, this indicates that the end of the robot arms are closer to the target area and is regarded as a reward signal; otherwise, it is regarded as a punishment signal.

The second part of INR is the obstacle avoidance function: $D_{j_i E_j}^L(s_t, a_t)$ represents the Euclidean distance between each joint of the robot arm and each mark point of obstacles, i represents the number of joints of the robot arm, and j represents the number of obstacle marks (as shown in Figure 2). We propose an obstacle avoidance model based on the idea of the artificial potential field method (AAPF), the reward function of obstacle avoidance is set with the reciprocal of the Euclidean distance between the end of arms and obstacles, and the sparse feedback of collision detection is changed into real-time dense feedback of distance.



(**a**)



(**b**)

Figure 2. Obstacle avoidance model. (**a**,**b**) Lunar robot collision detection configuration.

The third part of INR is the joint optimization function, which minimizes the angle change of the robot's arms under the premise of completing the rescue task. ω_i indicates the i-th joint angle of the robot; $\Delta\omega_i$ is the i-th joint angle change of the robot, $\Delta\omega_i = \omega_{i,t+1} - \omega_{i,t}$; n is a constant, indicating the number of arms and waist joints of the robot; φ is the joint optimization coefficient. The fourth term of INR is the time reward function, which aims to encourage agents to complete tasks in less time. In the equation, ϕ is the episode length when tasks are completed and $\phi_{\max}$ indicates the preset maximum number of training steps.

3.3. Network Structure

The two-layer learning network used in this chapter is shown in Figure 3, and each layer network has a unified modeling method. In the figure, yellow indicates the upper layer network structure and blue indicates the lower layer network structure. Both SGN and LLN use the long short-term memory (LSTM) [19] module to handle abstract states. SGN and LLN networks should not only detect the state of environment (agent), but also generalize in the task space. Because these networks receive mixed inputs, hierarchical networks need to converge at their respective levels.

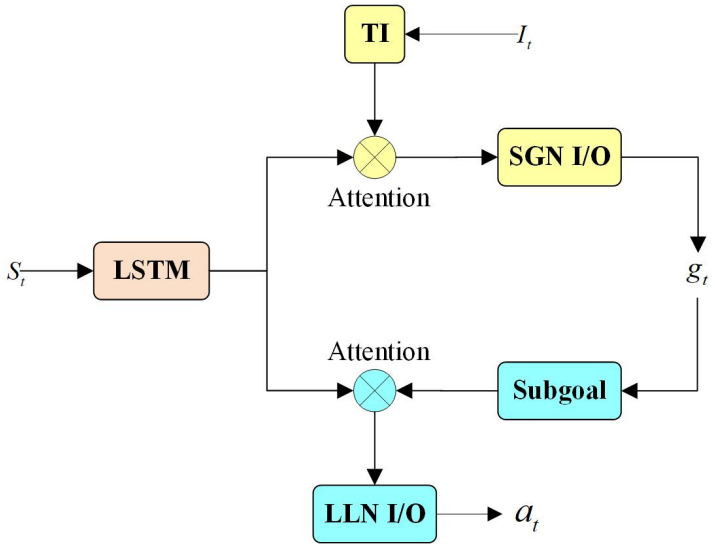


Figure 3. HRL-DCS network structure.

SGN mainly fuses states and tasks, while LLN mainly fuses states and sub-targets. SGN uses a multi-layer fully connected network to process task code I and output sub-target g_t. LLN uses a multi-layer fully connected network to process sub-target g_t and output action a_t. In order to ensure the boundedness of sub-targets, tanh is used as the activation function of the SGN output layer, and softmax is used as the activation function of LLN to output the probability of each action a_t.

In the dual-arm control problem of the lunar auxiliary robot, the motion of the manipulator is a continuous d-dimensional vector, so we model the parameterized sub-target as a continuous control problem. The deterministic strategy gradient is used to solve the reinforcement learning problem in high-dimensional state space, and the gradient maximization model is used to optimize the strategy.

4. Experiments

In this paper, Markov decision modeling is used to solve the control problem of the dual-arm robot. In this chapter simulation experiments and an ablation analysis are carried out to verify the advantages and reliability of HRL-DCS mentioned in this chapter. In the experiments we focused on two questions:

(1) Can the proposed method improve the reliability and success rate of exploration decision-making in a large-scale action space?

(2) Compared with the existing reinforcement learning methods, can the proposed method improve the training efficiency and convergence speed?

4.1. Experimental Setup

The dual-arm control problem of the robot is a process of finding the optimal and collision-free trajectory to reach the target state given the initial pose. In this study, the astronaut rescue problem is decomposed into two processes; that is, the lunar robot reaching the area between astronauts and the lunar surface with both arms is taken as the first goal, and the astronaut's pose reaching the target parameter is taken as the second goal. According to the kinematic model of the robot arm, the angular motion range of each joint of the robot arm is as follows:

$$\begin{aligned} q_{Ai} &= q_{A\text{int}ii} + 60 ContinuousAction[i] \\ q_{Bi} &= q_{B\text{int}ii} + 60 ContinuousAction[i] \end{aligned} \tag{3}$$

In the equation, q_{Ai} and q_{Bi} ($i = 1, 2, \ldots, 5$) respectively represent the joint angles of dual arm of the robot; $q_{\text{int}i}\{i = 1,\ 2,\ \ldots,\ 5\}$ denotes the initial values of each joint angle; $60 ContinuousAction[i]$ indicates that the output value of the joint angle will be calculated by Python API, and its variation range is $[-60, 60]$.

The training environment adopts a lightweight layout, and 16 agents are set for training, as shown in Figure 4. The maximum number of training steps is set as 700,000,000 (the number of steps is determined according to the specific training conditions, and the training can be stopped in advance when the ideal results are achieved), the entropy regularization intensity β is set as 1×10^{-3}, the acceptable difference range ε between the old and new strategies is 0.2, the number of hidden layers of the network is 128, and the reward signal parameter γ is set as 0.99.

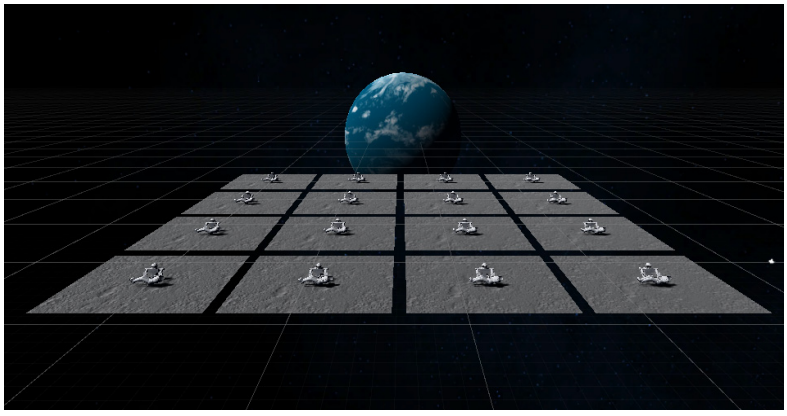


Figure 4. Training environment.

In the training process, we input $\left(R_{\text{int}i}, R_{goal}\right)$ at the beginning, the high-level SGN obtains the state s_t from the environment, and the strategy π_h generates sub-target g_t according to the task code I and the observed state s_t. The underlying strategy generates atomic action $a_t \sim \pi_l(\cdot|s_t, g_t, a_{t-1})$ according to sub-target g_t, current state s_t, and previous action a_{t-1}. Atomic action a_t is used to interact with astronauts and obstacles and obtain corresponding reward r_t^l. Based on the state s_t and action a_t, the observed state is transferred to the new state s_{t+1}. This process is repeated until the target state is reached. In the whole process, SGN collects the corresponding reward r_t and stores the transition data tuples (s_t, g_t, a_t, s_{t+1}) of time t for reinforcement learning, which is used to calculate the optimal trajectory strategy at any given starting position and target position in the workspace.

4.2. Evaluation Indicators

According to the reward and punishment function, the agent will train the optimal strategy from the initial pose to the target pose, and the result is shown in Figure 5. Among them, astronauts are placed at destination in the scene, and robots move randomly in a certain space to explore the trajectory of their dual-arms to rescue the astronauts.

Figure 5. Simulation results.

The criteria for judging whether the test is successful in this chapter are as follows: (1) the relative position of the astronaut's pose and center of gravity reaches the target state, (2) none of the joints of the robot's arms collide with obstacles (stage 1), and the test is successful when the above two conditions are met at the same time. In order to increase the difficulty and reliability of the test, the starting position of the robot base and the astronaut posture are random during the test. A total of 1000 experimental tests of the lunar assistant robot rescuing astronauts were carried out, which took 652 s and achieved a success rate of 98.6%.

4.3. Comparative Analysis

The purpose of the experiments in this Section is to verify the advantages of the proposed method in robot control, by comparing the HRL-DCS model with the traditional hierarchical reinforcement learning method FeUdal Networks (FuNs), and traditional reinforcement learning method, SAC.

(1) SAC: SAC is the traditional RL method. In the contrast experiment, we replace the reward function of SAC with AAPF, and test the effectiveness of the proposed layered framework under the same reward function.

(2) FuNs: FuNs is a commonly used hierarchical reinforcement learning method. Based on the goal-conditioned HRL method with a two-layer strategy structure, whether the proposed method has more advantages than FuNs will be tested.

(3) HRL-DCS-w/o AAPF: To further illustrate the role of AAPF in HRL, this Section adopts an ablation analysis method and sets up control group experiments by controlling variables. Specifically, based on HRL-DCS, the AAPF obstacle avoidance model was replaced by the common reward; that is, when the dual-arm joints contact with obstacles the value is -0.5, otherwise the value is $+0.5$. This comparison is used to verify the effectiveness of AAPF.

Under the premise of the same simulation parameters and hardware, the simulation results are shown in Figure 6. The abscissa in the graph represents the total number of training steps, which is independent of the computational efficiency of the hardware and the single-step speed of different algorithms. The ordinate represents the cumulative reward value obtained by the different methods. At the same time, based on different training models, 1000 groups of astronaut rescue experiments were carried out. The test results of different methods are shown in Table 1.

Based on the simulation results in Figure 6 and Table 1, the following results were obtained from the analysis of cumulative rewards, convergence rate, and test success rate:

(1) The SAC method has a faster convergence speed, but the cumulative reward is much lower than other methods, indicating that the method has a local optimal problem. From the analysis of training convergence speed, reward results, and test success rate, the SAC method cannot currently solve the exploration and decision-making of agents in high-dimensional state space and large action space in complex tasks. Hierarchical reinforcement learning methods have significant advantages when dealing with high-dimensional state space and large action space problems such as robot control.

(2) From the perspective of convergence speed, the training convergence speed of HRL-DCS proposed in this paper is slightly faster than that of FuNs; From the perspective of test success rate, the success rate (98.6%) of HRL-DCS is higher than that of FuNs, whose success rate is 91.8%. The results show that the HRL-DCS method proposed in this Section also has better performance than the current typical hierarchical reinforcement learning method FuNs when dealing with the dual-arm control of lunar robots in unstructured environments.

(3) Under the same hierarchical strategy and framework, the HRL-DCS method proposed in this paper is better than w/o AAPF in terms of cumulative reward and convergence speed, and the test success rate (98.6%) of HRL-DCS is also higher than the test success rate in w/o AAPF (87.3%). This means that the obstacle avoidance model AAPF proposed in this paper can optimize both the training speed of reinforcement learning and the success rate of tasks.

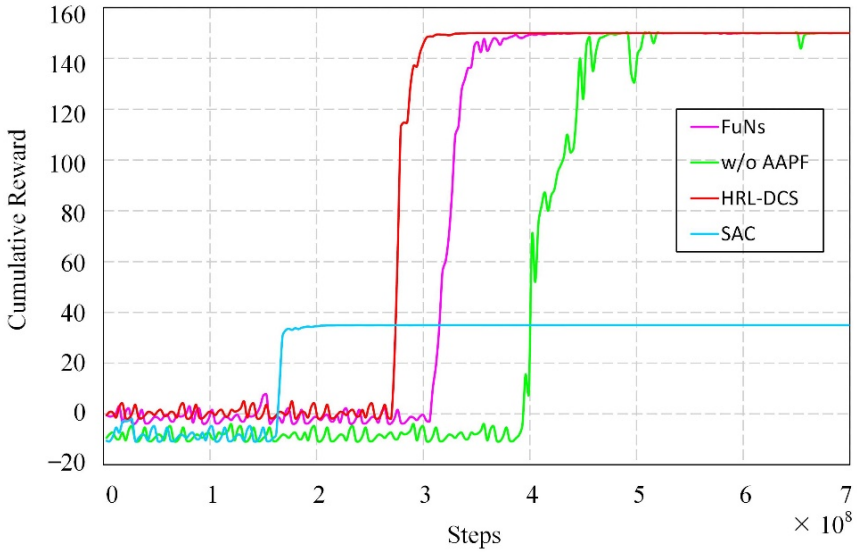


Figure 6. Training convergence of different models.

Table 1. Results of different models.

Model	Time (h)	# of Tests	Accuracy (%)
Soft Actor-Critic	—	1000	—
FeUdal Networks	67	1000	91.8
HRL-DCS-w/oAAPF	86	1000	87.3
HRL-DCS	59	1000	98.6

The advantages of the HRL-DCS framework proposed in this paper are as follows: (1) The number of joints at the end of the robot arm is much less than that of the arm. Therefore, the end of the robot arm in this paper simultaneously completes the tasks of obstacle avoidance and target approaching, while other non-end joints only perform obstacle avoidance tasks, and the design can reduce the state space and action space, enabling the agent to perform multi-objective exploration and training in low-dimensional space. (2) The training problem of high-dimensional joints is solved by a sequential decision-

making underlying strategy, which avoids multi-objective training of high-dimensional joints of the arms. (3) The reward mechanism of the underlying strategy can densely feedback the information of the target area, which improves the efficiency of exploration.

In addition, this paper also proposes a reward function design based on the idea of the artificial potential field method to feedback reward information in a dense reward manner. In robot control problems, HRL methods require each layer to reach the goal in order to obtain a reward. In this paper, the sub-goal of HRL is a continuous multi-dimensional vector, but the large three-dimensional action space of the two arms of the lunar robot makes the exploration space of the agent very large, which has a great impact on the learning efficiency and convergence. Based on this, the obstacle avoidance model based on the artificial potential field method proposed in this paper can make full use of the feedback information of the spatial position, which can quickly reduce the invalid exploration space and greatly improve the learning efficiency.

At the same time, the research of this paper is carried out on the basis of the research results of astronaut fine-grained attitude recognition [16]. The previous research simplifies the high-dimensional joint data of astronauts into low-dimensional human poses, which reduces the parameter dimension in reinforcement learning training. Based on the research results of fine-grained attitude recognition of astronauts, the robot can complete the rescue mission of astronauts in any posture.

5. Conclusions

In this work, we proposed a dual-arm control strategy model of a lunar assisted robot based on hierarchical reinforcement learning, and the dual-arm control problem was modeled as a two-layer Markov decision process. The high-layer strategy is responsible for generating discrete sub-targets in the state space of the robotic arm, learning and exploring the sub-targets, and optimizing the trajectory of the robot dual-arm from the long-term perspective and the whole strategy trajectory. The bottom strategy is responsible for achieving the parameterized sub-goal of the high-layer strategy; that is, optimizing the internal reward function in the joint space. The aim of the bottom strategy is to balance the demands of target approaching, obstacle avoidance, energy consumption reduction, etc. The proposed hierarchical strategy model greatly improves the efficiency of exploration decision-making of the dual-arm joint in 3-D action space. By setting up the reward function and the obstacle avoidance model based on the artificial potential field method, the aim of rapidly-exploring the optimal strategy in the multi-objective optimization was realized. In order to verify the method proposed in this paper, we chose the commonly used RL and HRL methods for comparison. Through simulation testing and ablation analysis, the advantages of AAPF in convergence speed, cumulative rewards, and target success rate were verified, and the advantages of HRL-DCS in dual-arm control were verified. In future work, we will test the validity of the proposed method in a 3-D environment using a dual-arm robot in physical environments.

Author Contributions: Conceptualization, W.R., D.H. and Z.W.; methodology, W.R., D.H. and Z.W.; software, W.R. and D.H.; validation, W.R. and D.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported the National Natural Science Foundation of China (No. U20B2056), and supported by Beijing Natural Science Foundation under grant number 1224039. The authors fully appreciate their financial support.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Hu, R.; Wang, Z.; Zhang, Y. A Lunar Robot Obstacle Avoidance Planning Method Using Deep Reinforcement Learning for Data Fusion. In Proceedings of the 2019 Chinese Automation Congress (CAC), Hangzhou, China, 22–24 November 2019; pp. 5365–5370.
2. Izzo, D.; Märtens, M.; Pan, B. A survey on artificial intelligence trends in spacecraft guidance dynamics and control. *Astrodynamics* **2019**, *3*, 287–299. [CrossRef]
3. Tang, G.; Hauser, K. A data-driven indirect method for nonlinear optimal control. *Astrodynamics* **2019**, *3*, 345–359. [CrossRef]
4. Zhang, L.; Li, S.; Xiong, H.; Diao, X.; Ma, O.; Wang, Z. Prediction of Intentions Behind a Single Human Action: An Application of Convolutional Neural Network. In Proceedings of the 2019 IEEE 9th Annual International Conference on CYBER Technology in Automation, Control, and In-telligent Systems (CYBER), Suzhou, China, 29 July–2 August 2019; pp. 670–676.
5. Nguyen-Tuong, D.; Peters, J. Model learning for robot control: A survey. *Cogn. Processing* **2011**, *12*, 319–340. [CrossRef] [PubMed]
6. Shirobokov, M.; Trofimov, S.; Ovchinnikov, M. Survey of machine learning techniques in spacecraft control design. *Acta Astronaut.* **2021**, *186*, 87–97. [CrossRef]
7. Li, Y.; Li, D.; Zhu, W.; Sun, J.; Zhang, X.; Li, S. Constrained Motion Planning of 7-DOF Space Manipulator via Deep Reinforcement Learning Combined with Artificial Potential Field. *Aerospace* **2022**, *9*, 163. [CrossRef]
8. Dong, G.; Zhu, Z.H. Incremental visual servo control of robotic manipulator for autonomous capture of non-cooperative target. *Adv. Robot.* **2016**, *30*, 1458–1465. [CrossRef]
9. Beltran-Hernandez, C.C.; Petit, D.; Ramirez-Alpizar, I.G.; Nishi, T.; Kikuchi, S.; Matsubara, T.; Harada, K. Learning force control for contact-rich manipulation tasks with rigid position-controlled robots. *IEEE Robot. Autom. Lett.* **2020**, *5*, 5709–5716. [CrossRef]
10. Xiong, H.; Ma, T.; Zhang, L.; Diao, X. Comparison of end-to-end and hybrid deep reinforcement learning strategies for controlling cable-driven parallel robots. *Neurocomputing* **2020**, *377*, 73–84. [CrossRef]
11. Shahid, A.A.; Roveda, L.; Piga, D.; Braghin, F. Learning Continuous Control Actions for Robotic Grasping with Reinforcement Learning. In Proceedings of the 2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Toronto, ON, Canada, 11–14 October 2020.
12. Prianto, E.; Kim, M.S.; Park, J.H.; Bae, J.H.; Kim, J.S. Path Planning for Multi-Arm Manipulators Using Deep Reinforcement Learning: Soft Actor–Critic with Hindsight Experience Replay. *Sensors* **2020**, *20*, 5911. [CrossRef]
13. Dong, G.; Zhu, Z.H. Predictive visual servo kinematic control for autonomous robotic capture of non-cooperative space target. *Acta Astronaut.* **2018**, *151*, 173–181. [CrossRef]
14. Ota, K.; Jha, D.K.; Oiki, T.; Miura, M.; Mariyama, T. Trajectory Optimization for Unknown Constrained Systems using Reinforcement Learning. In Proceedings of the 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Macau, China, 3–8 November 2019.
15. Moghaddam, B.M.; Chhabra, R. On the guidance, navigation and control of in-orbit space robotic missions: A survey and prospective vision. *Acta Astronaut.* **2021**, *184*, 70–100. [CrossRef]
16. Ren, W.; Ma, O.; Ji, H.; Liu, X. Human Posture Recognition Using a Hybrid of Fuzzy Logic and Machine Learning Approaches. *IEEE Access* **2020**, *8*, 135628–135639. [CrossRef]
17. Rui, Z.; Zhaokui, W.; Yulin, Z. A person-following nanosatellite for in-cabin astronaut assistance: System design and deep-learning-based astronaut visual tracking implementation. *Acta Astronaut.* **2019**, *162*, 121–134. [CrossRef]
18. Lingyun, G.; Lin, Z.; Zhaokui, W. Hierarchical Attention-Based Astronaut Gesture Recognition: A Dataset and CNN Model. *IEEE Access* **2020**, *8*, 68787–68798. [CrossRef]
19. Hochreiter, S.; Schmidhuber, J. Long Short-term Memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef] [PubMed]