Available online at www.sciencedirect.com

ScienceDirect

IFAC Papers Online CONFERENCE PAPER ARCHIVE

IFAC PapersOnLine 59-20 (2025) 2284–2289

Reinforcement Learning-Based Attitude Stabilization Control for Robot Astronauts

Liping Fang *, Liang Tang *, **, Jun Zhang*, **, Quan Hu***

* *Beijing Institute of Control Engineering, Beijing 100094, China (fangliping0818@163.com)*

** *Science and Technology on Space Intelligent Control Laboratory, Beijing 100094, China (tl614@sina.com, zhangjunsp@163.com)*

*** *Beijing Institute of Technology, Beijing 100081, China (Huquan@bit.edu.cn)*

Abstract: Robotic astronauts could play a crucial role in long-term duty and on-orbit experiments aboard space stations in future missions. Achieving attitude stabilization is critical for performing precision tasks. However, for robotic astronauts with high degrees of freedom, intricate motion coupling, and rich interactions, attitude control remains a significant challenge. A reinforcement learning-based framework is proposed to overcome this limitation, integrating curriculum learning with an Asymmetric Actor-Critic architecture and Proximal Policy Optimization (PPO). The approach is trained and validated within NVIDIA Isaac Gym, a high-performance GPU-accelerated physics simulation platform. The results demonstrate that the proposed policy enables rapid convergence of the robot's linear velocity, angular velocity, and attitude deviation, ensuring stable performance. Additionally, it shows strong generalization and robustness across varying initial conditions and curriculum levels. In conclusion, this strategy successfully achieves attitude stabilization control for robotic astronauts in space station environments, providing technical support for future on-orbit missions.

Keywords: Robot astronauts, Reinforcement learning, Attitude stabilization control, Curriculum learning

1. INTRODUCTION

The development of space intelligent robots for space exploration and utilization has advanced significantly in recent years (Li and Xie, 2022). Designed as humanoid robotic systems for on-orbit operations, robot astronauts can assist or replace human astronauts in tasks like space station maintenance and on-orbit experiments (Jiang et al., 2022). The idea was originally explored by NASA engineers in the 1990s. Robonaut-2 underwent on-orbit testing and performed maintenance tasks in the International Space Station from 2011 to 2018 (Baker et al., 2017). In August 2019, Skybot F-850 demonstrated assembly and monitoring tasks (Ma et al., 2023). However, limited by perception and control algorithms, both robots lacked lower limb mobility and relied heavily on teleoperation for upper limb tasks.

Due to microgravity, confined spaces, and unstructured environments of space stations, the dynamic modeling and control methods of robot astronauts differ significantly from ground robots (Fujiki et al., 2014). The strong coupling between limb motion and floating posture, combined with multiple contact points and frequent transitions, results in highly complex dynamics. Additionally, these systems are extremely sensitive to inertia, joint friction, contact dynamics, and external disruptions. These factors collectively pose substantial challenges to motion control design.

Recent advances in large language models (LLMs) and reinforcement learning (RL) have significantly enhanced robotic perception and control capabilities. Robot astronauts stabilize attitude through multi-limb contact with the environment, forming a contact-rich multibody system. This system exhibits distinct dynamics: over-actuated with closed-loop constraints during full-limb contact, while under-actuated when contact is reduced or lost. Traditional dynamics modeling lacks unified representations for these hybrid states, and model-based control struggle with high dimensionality, complex constraints, and computational inefficiency (Huang et al., 2024). In contrast, RL autonomously learns motion strategies from interaction data without explicit models (Xie et al., 2019). It adapts to disturbances and dynamic environments, offering robust control under uncertainty. Therefore, we propose an RL framework for robot-astronaut attitude stabilization in space capsules, integrating curriculum learning to progressively increase task difficulty and improve training efficiency. Simulation results confirm rapid stabilization across diverse initial conditions and curriculum levels, demonstrating robust adaptability.

The organization of this study is outlined as follows. Section 2 reviews relevant literature, followed by a detailed presentation of the developed RL control framework in Section 3. Simulation results are provided in Section 4, and Section 5 concludes with key insights and future research directions.

2. RELATED WORKS

Human astronauts achieve rapid, stable mobility within space stations by grasping handrails or pushing against capsule walls to initiate, accelerate, glide, and park. However, replicating such capabilities in robot astronauts remains challenging, prompting extensive research into motion control strategies.

Shen et al. (2024) proposed a strategy inspired by human acceleration and deceleration (AD) dynamics, where human interaction data were used to build arm dynamics. This strategy allowed robot astronauts to mimic human AD behavior and controll speed by pushing against capsule walls. However, it was limited to 2D motion and considered only a 2

10.1016/j.ifacol.2025.11.500

degrees of freedom (DOF) robotic arm. Jiang et al. (2019) developed a viscoelastic humanoid model featuring a mass-spring-damper, enabling stable XY-plane parking but neglecting handrail grasping torque, thus limiting full attitude control. In continuous motion, Zhang et al. (2023) studied robot astronauts mimicking human continuous propulsion with both arms against the spacecraft bulkhead. They proposed an optimization method utilizing the Artificial Bee Colony Algorithm for trajectory planning, contact positioning, and joint torque allocation. These model-based strategies, constrained by 2D assumptions or low-DOF formulations, lack scalability to full-body control in complex capsule scenarios.

RL has shown strong potential in attitude control and trajectory planning for free-floating robots. Rudin et al. (2021) leveraged deep RL for 3D pose adjustment and smooth landing in low-gravity environments with quadruped robots. Srivastava et al. (2023) first utilized the Proximal Policy Optimization (PPO) framework (Schulman et al., 2017) for 9-DOF synchronous control of a rotating robot, handling three independent tasks simultaneously: arm position, arm attitude, and base attitude control. Cao et al. (2023) introduced an Efficient Learning-based Path Tracking method for dual-arm robots to capture rotating non-cooperative targets, enhancing convergence via infinite norm rewards. However, few studies have explored RL control for robot astronauts in constrained multi-contact space capsule environments, especially for attitude stabilization.

To address these gaps, we develop a PPO-based RL framework. The policy learns attitude stabilization in space capsules through limb-wall interactions under weightless, multi-contact conditions. Curriculum learning progressively increases task difficulty to enhance convergence efficiency. This enables rapid stabilization and provides a foundation for downstream in-cabin operations.

3. METHOD

This section presents the overall framework of our control system, including the PPO-based policy architecture, reward function design, and curriculum learning strategy. The complete pipeline is illustrated in Figure 1.

3.1 Problem Formulation

The objective is to achieve attitude stabilization for a robot astronaut with random initial linear and angular velocities through limb contact with the capsule walls. The success criterion is defined as (1): $\Delta\theta \in [0, \pi]$ the minimal angular difference between the base quaternion $\boldsymbol{q}_{base}$ and the target quaternion $\boldsymbol{q}_{target}$, is calculated as twice the arccosine of the absolute value of their inner product, with $\bar{\theta}_{min}$ being the minimum angular threshold. $\boldsymbol{v}^b_{xyz}$ and $\boldsymbol{w}^b_{rpy}$ denote the base linear and angular velocities, while $\bar{v}_{min}$ and $\bar{w}_{min}$ are their corresponding minimum thresholds. The operator $\|\cdot\|_2$ is the Euclidean norm. In the simulation environment, the initial state is configured to avoid collisions with the capsule walls.

$$\begin{cases} \Delta\theta = 2 \times \arccos\left(\text{clamp}\left(\left|\boldsymbol{q}_{base} \cdot \boldsymbol{q}_{target}\right|, 0,1\right)\right) \le \bar{\theta}_{min} \\ \left\|\boldsymbol{v}^b_{xyz}\right\|_2 \le \bar{v}_{min} \\ \left\|\boldsymbol{w}^b_{rpy}\right\|_2 \le \bar{w}_{min} \end{cases} \quad (1)$$

The task termination includes six conditions: 1) collision failure, if the robot's head or torso collides with the capsule walls, i.e., $collision\_head \vee collision\_torso = True$; 2) out-of-bound failure, if the torso position exceeds the simulated capsule boundary along any coordinate axis, i.e., $\boldsymbol{p}_{torso} \notin [\boldsymbol{p}_{min}, \boldsymbol{p}_{max}]$, where where $\boldsymbol{p}_{min}$ and $\boldsymbol{p}_{max}$ define the spatial bounds of the capsule; 3) velocity limit failure, if the linear or angular velocity norm exceeds maximum thresholds, i.e., $\left\|\boldsymbol{v}^b_{xyz}\right\|_2 \ge \bar{v}_{max}$ or $\left\|\boldsymbol{w}^b_{rpy}\right\|_2 \ge \bar{w}_{max}$; 4) orientation deviation failure, if the angular deviation surpasses the allowable threshold, i.e., $\Delta\theta \ge \bar{\theta}_{max}$; 5) time_out failure, the episode reaches the maximum step length without task completion, i.e., $l > l_{max}$; 6) success, if the robot satisfies all stabilization thresholds as defined in (1) within the episode.

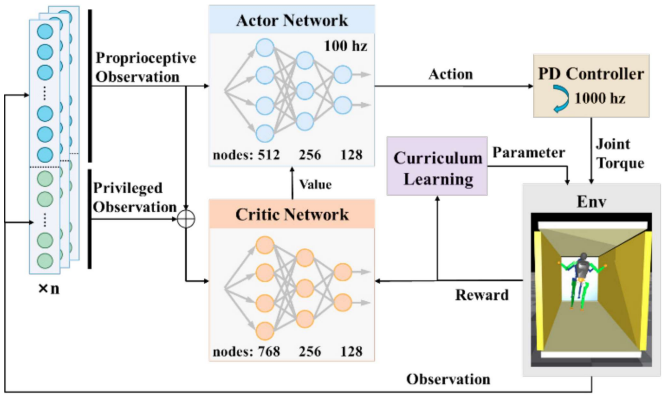

Figure 1. The control framework.

3.2 PPO-Based Control System Design

The robot astronaut model utilized in this study possesses 26 DOF in total. Each arm consists of 7 DOF, including 3 in the shoulder, 2 in the elbow, and 2 in the wrist. Each leg is equipped with 6 DOF, distributed as 3 in the hip, 1 in the knee, and 2 in the ankle. To control this high-dimensional system, we adopt a two-layer hierarchical architecture: a high-level RL policy, trained with PPO algorithm, runs at 100 Hz and outputs target joint positions, which are tracked by a low-level PD controller running at 1000 Hz.

We formalize the robot control problem as a Markov Decision Process, including state space S and action space A. The probability of transitioning to a new state $\boldsymbol{s}_{t+1}$ after performing an action $\boldsymbol{a}_t$ in the current state $\boldsymbol{s}_t$ is described by $P(\boldsymbol{s}_{t+1}|\boldsymbol{s}_t, \boldsymbol{a}_t)$. The discount factor γ helps balance the immediate rewards against those received in the future. The objective is to determine the optimal policy $\boldsymbol{\pi}^*$ that maximizes the total expected reward over time.

$$J_{\boldsymbol{\pi}} = \mathbb{E}[R_t] = \mathbb{E}\left[\sum_t \gamma^t r(\boldsymbol{s}_t, \boldsymbol{a}_t)\right] \quad (2)$$

In real-world scenarios, agents often only have access to partial observation. To tackle this challenge, we employ PPO algorithm and incorporate the Asymmetric Actor-Critic architecture (Pinto et al, 2017) along with privileged information during training to enhance decision-making. The actor network $\pi_\theta(\boldsymbol{a}|\boldsymbol{o}_{\le t})$ generates action distributions based on partial observation histories, while the critic network $V_\varphi(\boldsymbol{s})$

leverages privileged information to improve value estimation accuracy. The key formula for PPO is:

$$J^{CLIP}(\theta) = \mathbb{E}\left[min\left(k(\theta)\hat{A}_t, clip(k(\theta), 1-\varepsilon, 1+\varepsilon)\hat{A}_t\right)\right] \quad (3)$$

Here, $k(\theta) = \pi_\theta(\boldsymbol{a}|\boldsymbol{s})/\pi_{\theta_{\text{old}}}(\boldsymbol{a}|\boldsymbol{s})$ indicates the probability ratio comparing the new and existing policies. The advantage $\hat{A}_t$ quantifies the superiority of action $\boldsymbol{a}_t$ in state $\boldsymbol{s}_t$ over the expected value, calculated using Generalized Advantage Estimation (GAE) (Schulman et al., 2015). The hyperparameter ε constrains the update size for training stability. This objective balances exploration and exploitation.

In this work, we utilized multilayer perceptrons (MLPs) for both the actor and critic networks, each consisting of three hidden layers. The actor has 512, 256, and 128 nodes, and the critic has 768, 256, and 128 neurons. Across all layers, the Exponential Linear Unit (ELU) activation function is adopted.

3.3 Observations and actions

The robot astronaut's base position and Euler angles are denoted as $\boldsymbol{p}^b_{xyz}$ and $\boldsymbol{\theta}^b_{rpy}$, and the actuator joint positions and velocities represented by $\boldsymbol{\theta}_{joint}$ and $\dot{\boldsymbol{\theta}}_{joint}$. The actor network observes proprioceptive data, including joint states from encoders and base orientation and angular velocity from inertial sensors. Additionally, to capture action dynamics, it also introduces the previous action. The critic network, used only during training, further integrates privileged observation like base velocity, limb contact states, and target quaternion. Table 1 outlines the single-frame observation and their dimensions. To capture temporal dynamics, observation from the previous $\mathrm{n}-1$ frames are stacked as inputs, helping the network better understand model state transitions.

A continuous 26-D action space is adopted, consisting of the robot astronaut's absolute joint positions. All actions are normalized to $[-1,1]$, and converted into the expected joint positions $\boldsymbol{\theta}^{tar}_{joint}$ through the following linear transformation:

$$\boldsymbol{\theta}^{tar}_{joint} = \left(\boldsymbol{\theta}^{max}_{joint} - \boldsymbol{\theta}^{min}_{joint}\right)/2 \cdot \boldsymbol{a} + \left(\boldsymbol{\theta}^{max}_{joint} + \boldsymbol{\theta}^{min}_{joint}\right)/2 \quad (4)$$

where, $\boldsymbol{\theta}^{max}_{joint}$ and $\boldsymbol{\theta}^{min}_{joint}$ denote the maximum and minimum allowable positions for the robot joints. The expected joint positions are then converted into joint torques using the PD controller:

$$\boldsymbol{\tau} = K_p \cdot \left(\boldsymbol{\theta}^{tar}_{joint} - \boldsymbol{\theta}_{joint}\right) - K_d \cdot \dot{\boldsymbol{\theta}}_{joint} \quad (5)$$

here, K_p and K_d are the stiffness and damping coefficients.

3.4 Reward function

The reward function is designed to instruct the robot astronaut in learning efficient contact deceleration methods for attitude stabilization. It consists of immediate rewards and terminal rewards, as summarized in Table 2. The immediate rewards include: 1) velocity reward to encourage deceleration, 2) orientation reward for reaching the target quaternion, 3) joint tracking reward for following desired motions, and 4) regularization terms to promote smooth and efficient robot movement.

The terminal rewards include six parts: 1) a success reward r_{succ}, providing a large positive sparse reward for meeting orientation and velocity minimal thresholds; 2) a collision penalty r_{colli} for head or torso collisions with capsule walls; 3) an out-of-bounds penalty r_{out} for torso exceeding capsule boundaries; 4) an overspeed penalty $r_{overspeed}$ for linear or angular velocity norms surpassing thresholds; 5) an orientation deviation penalty r_{over_ori} for exceeding the allowable angular deviation; 6) a timeout penalty $r_{max\ \_length}$ for exceeding the step limit without success. For any given time step *t*, the overall reward is determined by the weighted combination of individual reward components r_i, where each component is weighted by μ_i, represented by the equation $r_t = \sum_i r_i \cdot \mu_i$.

Table 1. Detailed observation for the actor and critic networks

Single-frame observation	Dim	Actor	Critic
Joint position ($\boldsymbol{\theta}_{joint}$)	26	√	√
Joint velocity ($\dot{\boldsymbol{\theta}}_{joint}$)	26	√	√
Base angular velocity ($\boldsymbol{w}^b_{rpy}$)	3	√	√
Base Euler angle ($\boldsymbol{\theta}^b_{rpy}$)	3	√	√
Last action ($\boldsymbol{a}_{t-1}$)	26	√	√
Relative orientation ($\boldsymbol{q}_{diff}$)	4	√	√
Base linear velocity ($\boldsymbol{v}^b_{xyz}$)	3		√
Limb contact detection	4		√
Target orientation ($\boldsymbol{q}_{target}$)	4		√
Total dim		88	99

3.5 Curriculum learning

To improve the robustness and generalization ability of the robot astronaut across diverse scenarios, domain randomization is applied by adding perturbations to the robot's initial state, as detailed in Appendix A. To control perturbation magnitude while ensuring orientation diversity, only one direction of the Euler angle is perturbed at each initialization.

Considering the robot's high DOF, directly applying large disturbances may hinder training convergence. Therefore, a curriculum learning strategy is adopted to progressively increase task difficulty. Training starts with small disturbances and relaxed success thresholds for linear velocity and base orientation to facilitate early reward acquisition. Once the policy's average reward per step exceeds a predefined threshold, disturbances are expanded and success criteria tightened. The detailed settings and the overall curriculum procedure are illustrated in Appendix A.

4. SIMULATION RESULTS AND ANALYSIS

We developed a physics-based robot astronaut model in NVIDIA Isaac Gym (Makoviychuk et al, 2021), with full-body self-collision detection to prevent interpenetration artifacts. The simulated capsule's internal size was set to 4.0 m × 1.5 m × 1.8 m (L×W×H). Experiments were conducted on an Intel i9-14900K CPU and NVIDIA RTX 4090 GPU. The reward settings are as follows: success reward $r_{succ} = 130$, collision penalty $r_{colli} = -15$, out of bounds penalty $r_{out} = -15$, overspeed penalty $r_{overspeed} = -10$, orientation deviation penalty $r_{over\_orien} = -15$, timeout penalty $r_{max\ \_length} = -10$. The training hyperparameters are shown in Appendix B.

Table 2. Detailed design of reward function.

Reward	Equation(r_i)	Reward Scale(μ_i)	Explanation
Lin. vel. tracking	$\exp\left(-4 \times \left\|\boldsymbol{v}_{xyz}^{b}\right\|_2\right)$	1.6	To minimize the linear velocity
Ang. vel. tracking	$\exp\left(-3.5 \times \left\|\boldsymbol{w}_{rpy}^{b}\right\|_2\right)$	1.9	To minimize the angular velocity
Orientation tracking	$\exp(-4 \times \Delta\theta)$	2.3	To track desired orientation
Joint tracking	$\exp(-\|\boldsymbol{\theta}_{elbow}\|_2) + \exp(-\|\boldsymbol{\theta}_{knee}\|_2)$	1.8	To encourage full extension of the elbow and knee joints for easier contact
Dof pos limits	$RELU(\boldsymbol{\theta}_{joint} - \boldsymbol{\theta}_{joint}^{max}) + RELU(\boldsymbol{\theta}_{joint}^{min} - \boldsymbol{\theta}_{joint})$	-0.8	To penalize deviations beyond joint limits
Action smoothness	$\|\boldsymbol{a}_t - 2\boldsymbol{a}_{t-1} + \boldsymbol{a}_{t-2}\|_2$	-0.05	To encourage smooth movements
Limb large contact	$\sum_{i \in limb} clamp(\|\boldsymbol{F}_i\|_2 - 350,0,100)$	-0.02	To penalize excessive limb contact forces
Joint torques	$\left\|\boldsymbol{\tau}_{joint}\right\|_2$	-1e-6	To penalize high joint torques
Joint dof vel	$\left\|\dot{\boldsymbol{\theta}}_{joint}\right\|_2$	-1e-4	To penalize high joint velocities
Joint dof acc.	$\left\|\dot{\boldsymbol{\theta}}_{t,joint} - \dot{\boldsymbol{\theta}}_{t-1,joint}\right\|_2^2 / dt$	-1e-8	To penalize high joint accelerations
Terminal reward	$r_{succ} + r_{colli} + r_{out} + r_{overspeed} + r_{over\_ori} + r_{max\_length}$	1.0	A comprehensive terminal reward

To evaluate the proposed robot astronaut attitude stabilization strategy, the average episode success rate over 50 consecutive episodes was used as the evaluation metric. Figure 2 shows the evolution of reward values, success rate, and curriculum levels during training. During training, the mean reward increased and converged around 150, while the success rate initially peaked at approximately 60% before stabilizing at around 40%. Despite the performance drop associated with increased task difficulty, the overall high levels of reward and success rate indicate that curriculum learning effectively promoted progressive adaptation and improved policy robustness.

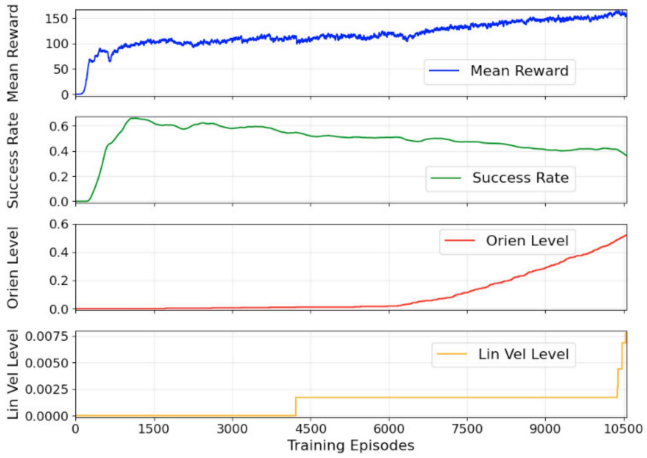

Figure 2. The training curves of Reward, success rate, and curriculum levels.

To assess the policy's stability and generalization, 50 forward simulation runs were conducted in a single environment with three different random seeds (42, 663, 9049). All tests were performed at the highest curriculum level achieved during training, with the results summarized in Table 3. The average episode success rate exceeded 60% across all seeds, confirming the policy's strong generalization and stability across diverse initial conditions.

Figure 3 illustrates the robot astronaut's attitude stabilization during a single forward run. After initial disturbances, the linear and angular velocities, as well as the angle deviation, rapidly decay within 0.2 seconds and subsequently exhibit small, stable oscillations. These results highlight the proposed control strategy's effectiveness in achieving fast disturbance rejection and maintaining low-variance motion stability.

Table 3. Comparison of average episode success rate across different seeds

Random seed value	Average episode success rate
42	62%
603	70%
9049	62%

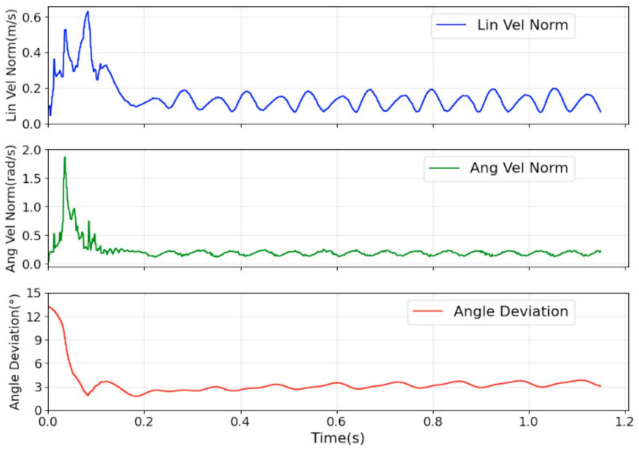

Figure 3. Robot astronaut's attitude stabilization process during a single forward run (seed = 42).

5. CONCLUSION

This paper has proposed an RL framework for attitude stabilization control of robotic astronauts in space capsules. We integrated the PPO algorithm with an Asymmetric Actor-Critic structure using privileged observation, and introduced

curriculum learning to enhance training efficiency and stability. Experimental results showed that under various random initial conditions, the method consistently achieved an average episode success rate of over 40%, with an initial peak at approximately 60%, and enabled rapid convergence of both velocity norms and attitude deviation. As task difficulty increased, curriculum learning maintained robust performance, adapting effectively to more complex environments. This approach successfully realized attitude control for robot astronauts aboard space stations, laying a solid foundation for operational tasks. Future work will expand the framework to attitude reorientation and wide-range movement control, addressing the diverse needs of on-orbit service.

REFERENCES

Baker, W., Kingston, Z., Moll, M., Badger, J. and Kavraki, L.E. (2017). Robonaut 2 and you: Specifying and executing complex operations. *2017 IEEE Workshop on Advanced Robotics and its Social Impacts (ARSO)*, 1-8.

Cao, Y., Wang, S., Zheng, X., Ma, W., Xie, X. and Liu, L. (2023). Reinforcement learning with prior policy guidance for motion planning of dual-arm free-floating space robot. *Aerospace Science and Technology*, 136, 108098.

Fujiki, S., Aoi, S., Senda, K. and Tsuchiya, K. (2014). Generation of adaptive splitbelt treadmill walking of a biped robot using learning of intralimb and interlimb coordinations. *2014 IEEE International Conference on Robotics and Automation (ICRA)*, 2392-2397.

Huang, W.C., Aydinoglu, A., Jin, W. and Posa, M. (2024). Adaptive contact-implicit model predictive control with online residual learning. *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 5822-5828.

Jiang, Z., Xu, J., Li, H. and Huang, Q. (2019). Stable parking control of a robot astronaut in a space station based on human dynamics. *IEEE Transactions on Robotics*, 36(2), 399-413.

Jiang, Z., Cao, X., Huang, X., Li, H. and Ceccarelli, M. (2022). Progress and development trend of space intelligent robot technology. *Space: Science & Technology.*

Li, L.F. and Xie, Y.C. (2022). Space robotic manipulation: A multi-task learning perspective. *Chinese Space Science and Technology*, *42*(3), 10-24.

Ma, B., Jiang, Z., Liu, Y. and Xie, Z. (2023). Advances in space robots for on-orbit servicing: A comprehensive review. *Advanced Intelligent Systems*, 5(8).

Makoviychuk, V., Wawrzyniak, L., Guo, Y., Lu, M., Storey, K., Macklin, M., Hoeller, D., Rudin, N., Allshire, A., Handa, A. and State, G. (2021). Isaac gym: High performance gpu-based physics simulation for robot learning. *arXiv preprint arXiv*:2108.10470.

Pinto, L., Andrychowicz, M., Welinder, P., Zaremba, W. and Abbeel, P. (2017). Asymmetric actor critic for image-based robot learning. *arXiv preprint arXiv*:1710.06542.

Rudin, N., Kolvenbach, H., Tsounis, V. and Hutter, M. (2021). Cat-like jumping and landing of legged robots in low gravity using deep reinforcement learning. *IEEE Transactions on Robotics*, 38(1), 317-328.

Schulman, J., Moritz, P., Levine, S., Jordan, M. and Abbeel, P. (2015). High-dimensional continuous control using generalized advantage estimation. *arXiv preprint arXiv*:1506.02438.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A. and Klimov, O. (2017). Proximal policy optimization algorithms. *arXiv preprint arXiv*:1707.06347.

Shen, M., Huang, X., Zhao, Y., Wang, Y., Li, H. and Jiang, Z. (2024). Human-like acceleration and deceleration control of a robot astronaut floating in a space station. *ISA transactions*, 148, 397-411.

Srivastava, R., Lima, R., Sah, R. and Das, K. (2023). Deep reinforcement learning based control of rotation floating space robots for proximity operations in pybullet. *2023 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 1224-1229.

Xie, Y.C., Wang, Y., Chen, A. and Li, L.F., (2019). Leaning based on-orbit servicing manipulation technology of space robot. *Aerospace Contrd and Application*, *45*(4), 25-37.

Zhang, Z., Wang, Z., Zhou, Z., Li, H., Zhang, Q., Zhou, Y., Li, X. and Liu, W. (2023). Omnidirectional Continuous Movement Method of Dual-Arm Robot in a Space Station. *Sensors*, 23(11), 5025.

Appendix A. DOMAIN RANDOMIZATION AND CURRICULUM LEARNING SETTINGS

Table 4 lists the domain randomization terms and parameter ranges. The curriculum settings for initial state perturbations and success thresholds are detailed in Table 5 and Table 6. And the overall curriculum procedure is shown in Table 7.

Table 4. Overview of domain randomization terms and corresponding parameter ranges.

Parameter	Unit	Range
Joint position	rad	[-0.1,0.1]
Base position (x)	m	[-1.0,1.0]
Base position (y, z)	m	[-0.1,0.05]
Linear velocity (y, z)	m/s	[-0.02,0.02]
Euler angle	°	[-5.0,5.0]
Angular velocity	rad/s	[-0.05,0.05]

Table 5. Curriculum settings for initial state perturbations.

Parameter	Range	Max Range	Increment
Linear velocity (m/s)	[±0.00,±0.02, ±0.02]	[±0.05,±0.2, ±0.2]	±0.015
Euler angle (°)	[±5.0,±5.0, ±5.0]	[±25.0,±25.0, ±25.0]	±3.0

Table 6. Curriculum settings for task success thresholds.

Parameter	Initial Threshold [min, max]	Min Threshold	Decrement
Linear velocity norm (m/s)	[0.25, 5.0]	[0.01, 1.0]	[0.03, 0.2]
Attitude deviation (°)	[12.0, 60.0]	[5.0, 30.0]	[0.5, 2.0]

Table 7. Pseudo-code of curriculum learning.

Algorithm 1: curriculum learning for robot astronaut attitude stabilization	
1:	Initialize base state sampling range $R_s = [r_s^{lin\_vel}, r_s^{orien}]$
2:	Initialize task goal thresholds $R_g = [r_g^{lin\_vel}, r_g^{orien}]$
3:	Initialize state range increment ΔR_s and task goal threshold decrement ΔR_g
4:	Initialize reward scales $\mu = [\mu^{lin\_vel}, \mu^{orien}]$
5:	Set curriculum level $c \leftarrow 1$
6:	**while** $c \leq C$ **do**
7:	**for** episode $m = 1, \ldots, M$ **do**
8:	Random sample initial state $\boldsymbol{s}_0 \sim (R_s)$
9:	**for** step $t = 0, \ldots, \mathrm{T} - 1$ **do**
10:	Sample action $\boldsymbol{a}_t \sim \pi_\theta(\boldsymbol{a}_t \mid \boldsymbol{o}_{\leq t})$ and execute it, observe reward $\boldsymbol{r}_t$ and next state $\boldsymbol{s}_{t+1}$, store transition $\langle \boldsymbol{s}_t, \boldsymbol{a}_t, \boldsymbol{r}_t, \boldsymbol{s}_{t+1} \rangle$ into replay buffer D, and update the actor and critic network
11:	**end for**
12:	**if** mean_reward_per_step $\geqslant \eta \odot \mu$, where $\eta = [\eta^{lin\_vel}, \eta^{orien}] = [0.6, 0.6]$
13:	Increase initial state range: $R_s \leftarrow clip(R_s + \Delta R_s,\ R_s^{min},\ R_s^{max})$
14:	Decrease task threshold: $R_g \leftarrow clip\left(R_g - \Delta R_g,\ R_g^{min},\ R_g^{max}\right)$
15:	Increase curriculum level: $c \leftarrow c + 1$
16:	**end if**
17:	**end for**
18:	**end while**

Appendix B. TRAINING HYPERPARAMETERS

Table 8. Hyperparameters

Parameter	Value
Training environments number	4096
Training epochs number	2
Batch size	4096×60
Episode length	600 steps
Discount factor	0.994
GAE discount factor	0.9
Entropy regularization coefficient	0.001
c1	0.8
c2	1.2
Learning rate	1e-6
Frame stack for actor network observation	15
Frame stack for critic network observation	3