DOI: 10.19650/j.cnki.cjsi.J2514211

基于深度强化学习的空间捕获自主决策*

黄 成,殷振凯,邢爱佳,于智龙
(哈尔滨理工大学自动化学院 哈尔滨 150080)

摘 要:针对航天器机械臂在复杂空间环境下执行旋转目标捕获任务时的自主决策问题,提出了一种改进的分布式深度确定性策略梯度的决策方法,以进一步增强捕获任务的自主决策能力,其中捕获航天器装备有三自由度的机械臂用于执行捕获操作,而目标航天器则处于固定位置并以恒定角速度进行旋转。为了提升空间捕获系统在复杂环境下的探索能力,设计了一种基于状态熵最大化的内部奖励探索机制:该机制通过计算当前状态与最小批量样本中各状态之间的欧氏距离,选取其中最小距离并通过熵计算将其转化为内部奖励,再与外部奖励进行线性叠加,构成最终的总奖励值,进而提升算法的收敛速度。与此同时,进一步构造了一种双网络结构:即通过两个价值网络分别对候选动作进行并行评估,并由两个策略网络选择价值更优的动作并输出执行,同时引入奖励重塑函数对奖励信号进行重塑,以降低算法估计的偏差,同时提高样本效率。最后,通过与多种主流强化学习算法进行仿真对比,验证了所提方法的有效性和优越性。具体实验结果表明:改进后的 D4PG 算法在奖励值方面提升了 32.25%,在收敛速度方面提升了 3.08%,显著提高了航天器机械臂执行空间捕获任务的自主决策能力。

关键词: 空间捕获;D4PG 算法;内部奖励探索;奖励重塑;双网络结构
中图分类号: TH166 **文献标识码**: A **国家标准学科分类代码**: 510.80

Autonomous decision-making for spatial capture based on deep reinforcement learning

Huang Cheng, Yin Zhenkai, Xing Aijia, Yu Zhilong
(*School of Automation, Harbin University of Science and Technology, Harbin 150080, China*)

Abstract: To address the autonomous decision-making challenges of a spacecraft manipulator performing a rotating target capture task in a complex space environment, this article proposes an improved distributed deep deterministic policy gradient decision-making method to further enhance the autonomous decision-making capabilities of the capture task. The capture spacecraft is equipped with a three-degree-of-freedom manipulator for capture, while the target spacecraft is fixed and rotates at a constant angular velocity. To improve the exploration capability of the space capture system in complex environments, this article designs an internal reward exploration mechanism based on state entropy maximization. This mechanism calculates the Euclidean distance between the current state and each state in a minibatch, selects the minimum distance, and converts it into an internal reward through entropy calculation. This reward is then linearly superimposed with the external reward to form the final total reward, thereby improving the algorithm's convergence speed. Furthermore, this article constructs a dual-network architecture. Two value networks evaluate candidate actions in parallel, and two policy networks select and execute the action with the best value. A reward reshaping function is introduced to reshape the reward signal to reduce estimation bias and improve sample efficiency. Finally, simulations and comparisons with several mainstream reinforcement learning algorithms, evaluate the effectiveness and superiority of the proposed method. Specific experimental results show that the improved D4PG algorithm has increased the reward value by 32.25% and the convergence speed by 3.08%, significantly improving the autonomous decision-making ability of the spacecraft robotic arm in performing space capture missions.

Keywords: space capture; D4PG algorithm; internal reward exploration; reward reshaping; dual network structure

收稿日期:2025-06-30 Received Date: 2025-06-30
*基金项目:国家自然科学基金(52102455)、黑龙江省自然科学基金(LH2023F032)项目资助

0 引　　言

随着航天技术的进步,各航天强国积极发展下一代智能化航天器,并在轨开展捕获、交会对接等复杂空间任务试验。当前星上制导控制系统主要依靠“星地大回路”模式对航天器进行测量和控制,受制于测控弧段和通信时延等因素,难以自主、实时、有效地应对复杂空间任务。因此,研究航天器智能自主决策方法势在必行,通过人工智能技术对航天器训练赋能,提高应对复杂空间任务的能力[1]。空间捕获是复杂空间任务中最具代表性的一类场景,由于空间目标尤其是非合作目标先验知识不完备、运动行为不配合的特性,难以计算自身的期望状态,从而无法预先规划轨迹,故捕获方的策略需要随着目标方状态的变化而调整。因此,求解空间捕获决策问题受到了国内外学者的广泛关注[2]。

在空间捕获任务中,执行器与航天器的连接方式主要分为柔性连接和刚性连接两类[3-4]。柔性连接主要采用飞网或飞爪作为捕获工具[5-6],相比之下,刚性连接通常采用机械臂或抓取器来捕获非合作目标,在动态环境中可表现出快速、高效的捕获能力,且在捕获过程中能够显著削弱外部扰动的影响,更适用于复杂空间任务中的精细化操作[7]。因此,针对空间捕获任务中的稳定控制需求,采用刚性连接来研究其自主决策方法。目前针对决策方法的研究可以分成两种:一种是微分对策理论;另一种则是强化学习方法。微分对策理论的思路最早由美国兰德公司的 Isaacs[8] 提出,并广泛应用于飞行器拦截[9]、轨道追逃[10]等领域。微分对策理论通过动态规划或者极大值原理求解双边最优轨迹,使得双方达到纳什均衡,这一求解过程需要建立精确的数学模型,并进行大量的计算,在实际应用过程中可能存在求解成本较高、精度低等问题,难以满足刚性连接空间捕获任务对实时性、准确性的需求,在空间捕获领域的应用相对较少[11-12]。

随着人工智能技术的不断发展,强化学习方法受到广泛关注,该方法不需要建立精确的数学模型,而是引入深度神经网络和奖励函数与环境进行交互,通过反馈的奖励信号形成有效的学习策略,从而增强自主学习能力。目前,深度强化学习算法已在路径规划[13-14]、姿态控制[15-16]、空战博弈[17-18]和轨道拦截制导[19-20]等领域有着广泛的应用。文献[21]采用近端策略优化(proximal policy optimization, PPO)算法并结合基于模型的反馈线性化控制器来控制末端执行器完成空间捕获任务;文献[22]则提出基于双延迟深度确定性策略梯度(twin delayed deep deterministic policy gradient, TD3)算法的捕获方法,并设计了 7 种不同奖励函数的轨迹规划器以及跟踪器,从而提高捕获任务的成功率;文献[23]采用柔性动作-评价(soft actor-critic, SAC)算法并结合位姿规划方法实施捕获,从而提高算法的稳定性。在上述研究中 PPO 算法通过限制策略更新幅度,有效提高了训练的稳定性以及智能体的决策能力,最终促使算法达到了更优的收敛状态。类似地,TD3 算法通过引入双重网络以及延迟更新等机制,显著增强了算法的稳定性和收敛速度。SAC 算法通过增加熵来增强策略的随机性,从而促进更多的探索,提升后续的学习效率。然而,在面对高维且复杂的空间捕获环境时,上述 3 种算法可能会展现出一定的局限性,这主要是由于在处理高度复杂且多维的状态空间时,这些算法的收敛速度普遍较为缓慢,进而限制了它们迅速发现较优策略的能力。因此,为了在空间捕获任务中实现快速响应和高效决策,需要采用更全面和自主性更高的策略和技术手段来应对。文献[24]中采用深度确定性策略梯度(deep deterministic policy gradient, DDPG)算法来实现空间捕获,该算法能够在环境中有效地调整控制策略,快速适应环境的变化。然而,在高维状态空间下,其应用仍存在一定的局限性。因此,文献[25]提出分布式深度确定性策略梯度(distributed distributional deep deterministic policy gradient, D4PG)算法,该算法引入了分布式架构,允许多个智能体并行训练,从而增加了智能体所收集经验的多样性并提高了算法的学习效率,使得智能体能够更快速的找到较优策略。此外,该算法还采用了分布式价值估计,为决策过程提供了更全面的信息,在复杂或不确定性的环境中,考虑更多不确定性因素,增强了决策过程的鲁棒性。

综合考虑刚性连接空间捕获任务对自主决策方法稳定性和快速性的需求,本文以 D4PG 算法作为基础算法,并在此基础上进行改进,进一步优化算法的性能。主要创新点为:1) 提出了一种融合最大熵理论以及 K 最近邻算法的内部探索机制,通过将状态转化为内部奖励促使智能体进行探索,以进一步的解决算法在高维状态空间下的探索难题。2) 构造了一种双网络结构来选择较优的动作并执行,同时对奖励值进行重塑,降低估计偏差,提高算法的稳定性。

1 问题描述与建模

1.1 动力学模型与控制框架建立

本文研究的空间捕获任务是一个复杂而动态的过程,涉及两个航天器在平面环境中精确协同工作,具体捕获情况如图 1 所示。其中,实线框内表示两个航天器的起始位置,虚线框则描绘了捕获过程,而点划线框内则标示了捕获成功的状态。

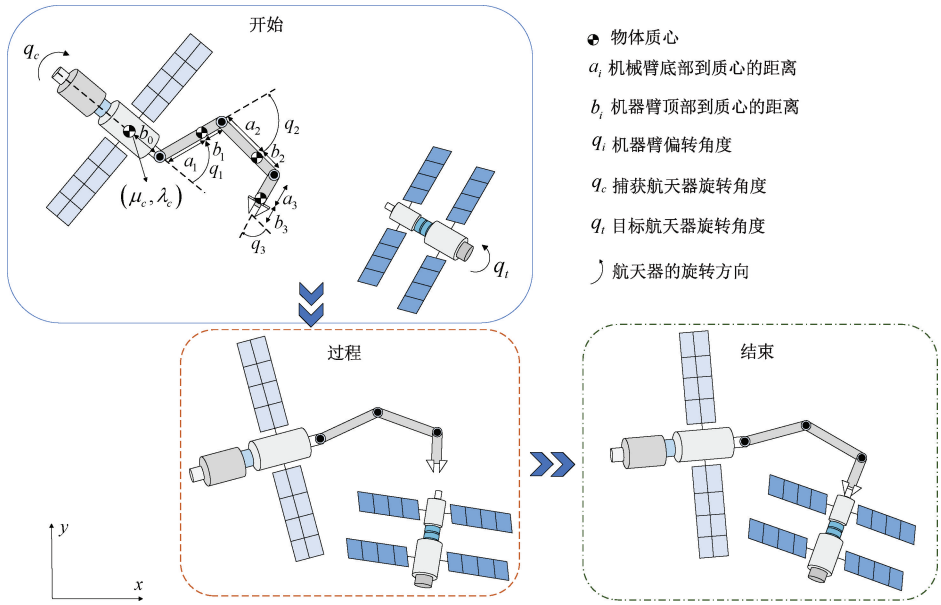


图 1 空间捕获示意图

Fig. 1 Diagram of spatial capture

具体的过程可以描述为:捕获航天器装备有三自由度的机械臂并且通过施加力矩来控制其动作,使末端执行器逐步接近对接端口,从而实现捕获。目标航天器设有一个对接端口,并以恒定的角速度在固定位置上进行旋转。当末端执行器与对接端口的距离<0.04 m 时,即视为捕获成功。此外,为了便于对目标航天器实施后续操作,需确保捕获后整体的航天器以及机械臂保持稳定,即使其旋转角速度尽可能的降至为 0。

由于本文需确保实时掌握捕获航天器和机械臂的状态,因此采用动力学方程来进行描述,具体联合体的动力学建模式如式(1)所示。

$$\boldsymbol{M}(\boldsymbol{\tau}_c)\ddot{\boldsymbol{\tau}}_c + \boldsymbol{C}(\boldsymbol{\tau}_c,\dot{\boldsymbol{\tau}}_c)\dot{\boldsymbol{\tau}}_c = \boldsymbol{u}_c \tag{1}$$

式中:$\boldsymbol{M}$ 是质量矩阵;$\boldsymbol{C}$ 是科里奥利矩阵;$\boldsymbol{\tau}_c$ 为捕获航天器的状态;$\boldsymbol{u}_c$ 是控制的力矩[26]。为了进一步的提高控制精度,引入了积分控制器。通过利用航天器机械臂联合体的加速度差来生成控制力矩,具体如式(2)所示。

$$\boldsymbol{u}_t = \boldsymbol{u}_{t-1} + k_i \times \boldsymbol{a}_e(t) \tag{2}$$

式中:$\boldsymbol{u}_t$ 为联合体当前时刻的力矩;$\boldsymbol{u}_{t-1}$ 为前一时刻的力矩;$\boldsymbol{\alpha}_e(t) = |\boldsymbol{\alpha}_{desired} - \boldsymbol{\alpha}_t|$ 为当前时刻的加速度误差,$\boldsymbol{\alpha}_{desired}$ 表示期望的加速度,$\boldsymbol{\alpha}_t$ 为当前时刻的加速度。在此基础上,还引入了前馈补偿策略,以进一步优化控制性能。前馈补偿公式可以用式(3)进行表示。

$$\boldsymbol{u}_{ff} = \boldsymbol{M}(\boldsymbol{\tau}_c)\boldsymbol{\alpha}_{desired} + \boldsymbol{C}(\boldsymbol{\tau}_c,\dot{\boldsymbol{\tau}}_c)\boldsymbol{v}_{desired} \tag{3}$$

式中:$\boldsymbol{v}_{desired}$ 表示期望的速度。前馈补偿策略能够有效抵消由于惯性和非线性因素带来的力矩误差,确保捕获方能够顺利完成捕获任务。因此,总的力矩可以表示两个线性叠加和,具体可以用式(4)表示。

$$\boldsymbol{u}_{total} = \boldsymbol{u}_t + \boldsymbol{u}_{ff} \tag{4}$$

综上,整体控制框架如图 2 所示。

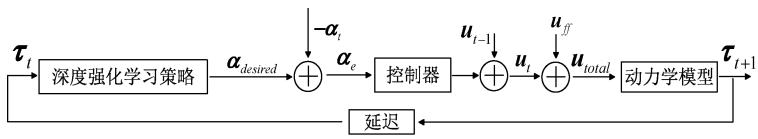


图 2 整体框架

Fig. 2 Overall framework diagram

1.2 马尔可夫决策过程模型建立

本文主要采用深度强化学习算法来设计捕获方的自主运动策略。在此之前,必须先将空间捕获系统建模为马尔可夫决策过程(Markov decision processes, MDP)。该模型主要由 5 个元素组成,具体设计为:

1) 状态空间 $\boldsymbol{\tau}$:空间捕获系统的控制目的是使捕获航天器上的末端执行器能够快速、稳定地接近目标航天器的对接端口,因此利用捕获航天器的位置

$\boldsymbol{x}=[\mu_c \quad \lambda_c \quad q_c]$ 和速度 $\dot{\boldsymbol{x}}$ 以及机械臂的偏转角度 $\boldsymbol{q}=[q_1 \quad q_2 \quad q_3]$ 和角速度 $\dot{\boldsymbol{q}}$ 来构建状态。此外,为了使捕获航天器能够及时预测并响应目标航天器的运动趋势,故将目标航天器的角速度 $\dot{\boldsymbol{q}}_t$ 也纳入状态空间中。另外,为了进一步提高捕获任务的准确率,还需要评估当前状态和目标状态之间的距离和方向,以便做出更准确的控制决策,即 $[\mu_t-\mu_c \quad \lambda_t-\lambda_c \quad q_t-q_c]$。因此 MDP 状态空间被定义为式(5),即:

$$\boldsymbol{\tau}=[\boldsymbol{\tau}_c \quad \dot{\boldsymbol{\tau}}_c \quad \dot{\boldsymbol{q}}_t \quad \mu_t-\mu_c \quad \lambda_t-\lambda_c \quad q_t-q_c]^{\mathrm{T}}\in R^{16} \tag{5}$$

上式中 $\boldsymbol{\tau}_c=[\mu_c \quad \lambda_c \quad q_c \quad q_1 \quad q_2 \quad q_3]$,相关参数均在图 1 中进行定义。

2) 动作空间 $\boldsymbol{a}$:为了使捕获航天器能够有效适应目标航天器的动态变化,并快速地调整其位置和姿态,因此引入捕获航天器的线性加速度和角加速度来构建动作空间。同时,为了精确控制末端执行器,使其能够顺利接触到对接端口,本文还将三自由度机械臂的旋转角加速度纳入了动作空间中。综上所述,动作空间如式(6)所示。

$$\boldsymbol{a}=[\boldsymbol{\sigma}_x \quad \boldsymbol{\sigma}_y \quad \boldsymbol{\alpha}_c \quad \boldsymbol{\alpha}_1 \quad \boldsymbol{\alpha}_2 \quad \boldsymbol{\alpha}_3] \tag{6}$$

式中:$\boldsymbol{\sigma}_x$、$\boldsymbol{\sigma}_y$ 为捕获航天器 X、Y 轴方向上的线性加速度;$\boldsymbol{\alpha}_c$ 为捕获航天器旋转的角加速度;$\boldsymbol{\alpha}_1$、$\boldsymbol{\alpha}_2$、$\boldsymbol{\alpha}_3$ 为三自由度机械臂的旋转角加速度。

3) 状态转移概率 $P(\boldsymbol{\tau}'|\boldsymbol{\tau},\boldsymbol{a})$:该概率函数描述了捕获方在状态 $\boldsymbol{\tau}$ 下执行动作 $\boldsymbol{a}$ 后,转移到下一个状态 $\boldsymbol{\tau}'$ 的可能性。此转移概率仅依赖当前状态和动作,独立于之前的状态和动作。

4) 奖励函数 $r(\boldsymbol{\tau},\boldsymbol{a})$:奖励函数用于评估捕获方在状态 $\boldsymbol{\tau}$ 下采取动作 $\boldsymbol{a}$ 时的表现,从而引导其优化控制策略,以实现精确的捕获。具体奖励值设计为:(1)捕获成功给予奖励;(2)末端执行器距离对接端口<0.1 m 时给予中途奖励;(3)操纵机械臂达到角度极限时给予惩罚;(4)捕获航天器偏离目标航天器时给予惩罚;(5)其他情况下给予 0 奖励。奖励函数如式(7)所示。

$$r=\begin{cases}100-50\left|\sin\left(q_c+q_1+q_2+q_3-\right.\right.\\ \left.q_t-\pi/2\right)\left|-50\right\|\dot{\boldsymbol{p}}_e-\dot{\boldsymbol{d}}_t\left\|-\right.\\ 50\left|\dot{\boldsymbol{q}}_c+\dot{\boldsymbol{q}}_1+\dot{\boldsymbol{q}}_2+\dot{\boldsymbol{q}}_3-\dot{\boldsymbol{q}}_t\right|-25\left|\boldsymbol{h}_{cmt}\right|,\\ \qquad \left\|d_t-p_e\right\|\leqslant 0.04\ \mathrm{m}\\ 25, \quad \left\|d_t-p_e\right\|\leqslant 0.1\ \mathrm{m}\\ -5, \quad \left|q_1\right|\cup\left|q_2\right|\cup\left|q_3\right|\geqslant\pi/2\\ -100, \quad \tau_c\notin[0,3.5]\cup\lambda_c\notin[0,2.4]\\ 0, \quad \text{其他}\end{cases} \tag{7}$$

式中:p_e 表示末端执行器的位置;d_t 表示目标航天器对接端口的位置;$\boldsymbol{h}_{cmt}$ 表示捕获成功后的总的角动量,具体的计算公式如式(8)所示。

$$\boldsymbol{h}_{cmt}=\boldsymbol{h}_{cm}+\boldsymbol{S}(\boldsymbol{p}_{cm}-\boldsymbol{p}_{cmt})\times\boldsymbol{l}_{cm}+\boldsymbol{h}_t+\boldsymbol{S}(\boldsymbol{p}_t-\boldsymbol{p}_{cmt})\times\boldsymbol{l}_t \tag{8}$$

式中:$\boldsymbol{h}_{cm}$、$\boldsymbol{l}_{cm}$、$\boldsymbol{p}_{cm}$ 分别表示捕获航天器的角动量、线动量以及质心的位置;$\boldsymbol{h}_t$、$\boldsymbol{l}_t$、$\boldsymbol{p}_t$ 则分别表示目标航天器的角动量、线动量以及质心的位置;$\boldsymbol{p}_{cmt}$ 表示捕获成功后两个航天器整体的质心位置;$\boldsymbol{S}$ 则表示一个反对称矩阵,可以将叉乘运算转换成矩阵乘法运算,具体如式(9)所示。

$$\boldsymbol{S}(\boldsymbol{b})=\begin{pmatrix}0 & -\boldsymbol{b}_z & \boldsymbol{b}_y\\ \boldsymbol{b}_z & 0 & -\boldsymbol{b}_x\\ -\boldsymbol{b}_y & \boldsymbol{b}_x & 0\end{pmatrix} \tag{9}$$

若智能体在行动中超出时间的限制或者以任何方式与目标航天器发生碰撞,则认定捕获失败,反馈的奖励值为 0,并立即终止该回合。根据奖励函数的设定,奖励值的界限为[-100,125],理论最大预期奖励为 125,即捕获航天器接近目标航天器获得奖励 25 与完美对接成功的奖励 100,但在实际操作中,这一最大预期奖励较难实现。

5) 折扣因子 γ:折扣因子影响着智能体在学习过程中如何平衡即时奖励与未来奖励的权重。一般而言,折扣因子的取值范围为[0,1]。折扣因子 γ 的值越接近于 1,表示智能体在决策时越重视未来的奖励;反之,则表示智能体将更加关注当前的奖励,本文设计 $\gamma=0.95$。

2　基于 D4PG 的空间捕获自主决策方法设计

本章分析了 D4PG 算法学习策略的过程,并针对其探索不足的问题,提出了一种基于状态的内部奖励探索机制。此外,为了进一步的提升算法的稳定性,还提出了双网络结构以及奖励重塑的方法,以减少估计偏差带来的影响。算法集成图如图 3 所示。

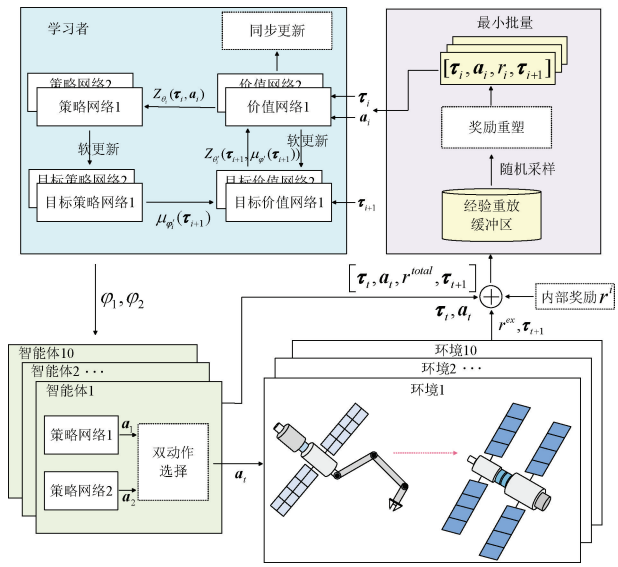


图 3　算法集成结构

Fig. 3　Algorithm integration structure diagram

2.1 分布式深度确定性策略梯度

D4PG 算法是在传统的 DDPG 算法的基础上进行改进,引入分布式训练机制,通过 n 个并行的 Actor 去与环境进行交互,高效地收集经验,这些经验被存储在一个共享的经验回放区中,供学习者进行采样学习,以便进一步优化其策略,D4PG 的这一改进机制显著提升了算法的学习效率。此外,D4PG 算法采用了分布式的价值估计方法,即价值网络的输出不再是单一的 Q 值,而是状态-动作对的奖励分布 $Z(\boldsymbol{\tau},\boldsymbol{a})$,这种改进使得智能体在策略评估时能够更加全面,同时也使得算法在面对复杂任务时适应性更强。相应的,分布式贝尔曼方程如式(10)所示。

$$\hat{Z}(\boldsymbol{\tau},\boldsymbol{a}) = r + \gamma Z_{\theta'}(\boldsymbol{\tau}',\mu_{\varphi'}(\boldsymbol{\tau}')) \tag{10}$$

式中:θ' 表示目标价值网络的参数;φ' 表示目标策略网络的参数;$\hat{Z}(\boldsymbol{\tau},\boldsymbol{a})$ 为目标分布;$Z_{\theta'}(\boldsymbol{\tau}',\mu_{\varphi'}(\boldsymbol{\tau}'))$ 为目标价值网络来估计下一个时刻状态-动作的奖励分布;r 为即时奖励。为了更新价值网络参数,需要计算价值网络输出的奖励分布 $Z_{\theta}(\boldsymbol{\tau},\boldsymbol{a})$ 与目标分布 $\hat{Z}(\boldsymbol{\tau},\boldsymbol{a})$ 之间的差异。计算这两个分布差异的公式被定义为损失函数,具体如式(11)所示。

$$L(\theta) = E_{\rho}[D_{w}(Z_{\theta}(\boldsymbol{\tau}_i,\boldsymbol{a}_i),\hat{Z}(\boldsymbol{\tau}_i,\boldsymbol{a}_i))] \tag{11}$$

式中:D_w 表示两个分布的距离值;ρ 表示从经验缓冲区采样的数据;θ 为价值网络的参数。相应的,策略网络的目标函数为 $J(\varphi) = E[Z_{\theta}(\boldsymbol{\tau},\boldsymbol{a})]$,对其进行求导可以得到式(12),即:

$$\nabla_{\varphi}J(\varphi) = E_{\rho}[\nabla_{a}Z_{\theta}(\boldsymbol{\tau},\boldsymbol{a})|_{a=\mu_{\varphi}(\tau)}\nabla_{\varphi}\mu_{\varphi}(\boldsymbol{\tau})] \tag{12}$$

式中:φ 表示策略网络的参数。采用梯度法来更新网络参数,具体可以由式(13)来计算。

$$\begin{cases}\theta \leftarrow \theta - \eta_1 \nabla_{\theta}L(\theta)\\ \varphi \leftarrow \varphi + \eta_2 \nabla_{\varphi}J(\varphi)\end{cases} \tag{13}$$

式中:η_1 为价值网络的学习率;η_2 为策略网络的学习率。

2.2 基于状态的内部奖励探索机制设计

尽管 D4PG 算法引进了分布式训练结构和价值估计方法,但仍沿用了传统的确定性策略方法,因此提升智能体的空间探索能力尤为重要。传统的探索方法主要采用高斯或者 OU(Ornstein-Uhlenbeck)噪声进行探索,加入噪声虽然在一定程度上提升了智能体的探索能力,但在面对高维度复杂环境下的任务,依然存在较大的局限性,无法完全满足任务对于高效性和稳定性的需求。为了解决上述问题,本文提出了一种基于状态的内部奖励探索(state-based internal reward exploration, SBIRE)的方法。该方法将智能体自身的状态转化为内部奖励与智能体在环境中交互产生的外部奖励以一定的权重进行叠加,以此引导探索,从而弥补可能出现的外部奖励稀疏或延迟等问题。然而,该方法也面临着一个巨大的挑战,即如何将状态和奖励有效地联系起来。为了解决这个问题,引入了状态熵的概念,熵值越大,就说明数据随机性越强。假设 X 是一个随机变量并且服从分布 p,其中子集满足 $\mathcal{X} \subset B^n$,那么存在熵为:$H(X) = -E_{x\sim p(x)}[\log(p(x))]$。如果分布 p 是高维的,那么就需要对熵进行估计。本文采用 K-最近邻状态熵估计器,计算公式如式(14)和(15)所示。

$$\hat{H}_N^k(X) = \frac{1}{N}\sum_{i=1}^{N}\log\left(\frac{N\times(\|d_i - d_i^{K\text{-}NN}\|_2^n)\times\pi^{\frac{n}{2}}}{k\times\Gamma\left(\frac{n}{2}+1\right)}\right) + T_k \tag{14}$$

$$\propto \frac{1}{N}\sum_{i=1}^{N}\log\|d_i - d_i^{K\text{-}NN}\|_2 \tag{15}$$

式中:N 为样本数量,即 $\{d_i\}_{i=1}^{N}$ 的数量;$d_i^{K\text{-}NN}$ 为在样本集 $\{d_i\}_{i=1}^{N}$ 内采样 d_i 的 K 最近邻值;T_k 是一个偏差矫正项;Γ 为伽马函数;n 为 X 的维度,满足 $n>0$;$\pi \approx 3.141\,592\,6$。

此外,为了有效的估计状态熵,利用随机编码器(random encoder, RE)对状态进行编码生成特征向量,并结合 K-最近邻方法[27-28]进行状态熵估计。RE 的网络如图 4 所示。

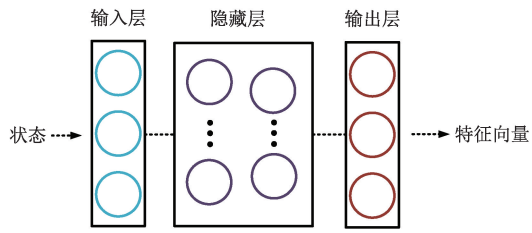


图 4 RE 网络示意图

Fig. 4 RE network diagram

假设 RE 网络的参数为 ψ,且该参数在初始化后不进行更新。该网络的主要作用是将输入的状态转换成特征向量,即 $\boldsymbol{l} = f_{\psi}(\boldsymbol{\tau})$,在获得状态的特征表示后,智能体利用 K-最近邻方法搜索已编码的状态之间的最小距离差值,将该差值与常数 g 进行叠加并做对数计算,获得高维数据下的最大化熵,并将熵值转化为内部奖励,如式(16)所示。

$$r^i(\boldsymbol{\tau}_t) = \log(\|\boldsymbol{l}_t - \boldsymbol{l}_i^{K\text{-}NN}\|_2 + g) \tag{16}$$

式中:$\boldsymbol{l}_t$ 为随机编码器的输出;$\boldsymbol{l}_i^{K\text{-}NN}$ 为矢量 $\boldsymbol{l}_t$ 的 K 最近邻矢量值;g 为超参数,本文设定 $g = 1.5$。总的奖励值设计为外部奖励和内部奖励线性叠加构成,如式(17)所示。

$$r_t^{total} = r^e(\boldsymbol{\tau}_t,\boldsymbol{a}_t) + \alpha_{step}\times r^i(\boldsymbol{\tau}_t) \tag{17}$$

式中:$\alpha_{step} \geqslant 0$ 为超参数,决定智能体是进行探索还是利用。为了鼓励智能体在接近任务结束时尽可能多地关注外部奖励,因此内部奖励权重 α_{step} 应该随着训练次数增

加逐渐衰减至 0,具体如式(18) 所示。

$$\alpha_{step} = \alpha_0 \times \beta_{step} \tag{18}$$

式中:α_0 为初始内部奖励所占的权重;β_{step} 为衰减率。综上所述,内部奖励的具体计算流程如图 5 所示。

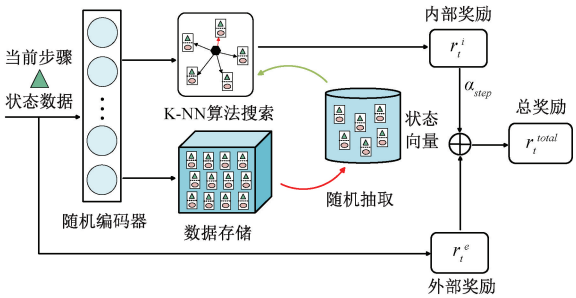


图 5　内部奖励流程

Fig. 5　Internal reward flowchart

为了提高训练效率,本文将随机编码器输出的特征向量存储在一个新的数据存储区中,当该存储区内的数据达到预先设定的阈值时,算法就会随机采样 30 个状态向量,并与当前状态的特征向量进行比较,找到欧式距离最小的一组状态数据。随后,利用式(16) 和(17) 分别计算当前状态的内部奖励以及总奖励值。综上,本文提出的 SBIRE 的伪代码如算法 1 所示。

算法 1:基于状态的内部探索

初始化重放缓冲区 R
初始化随机编码器的参数 ψ
for $0 < t < T$ **do**
　收集状态转移数组 $s_t = (\boldsymbol{\tau}_t, \boldsymbol{a}_t, \boldsymbol{\tau}_{t+1}, r^e)$
　计算 $\boldsymbol{l}_t = f_\psi(\boldsymbol{\tau}_t)$,并将 $\{s_t, \boldsymbol{l}_t\}$ 存入到重放缓冲区 R
　If length $(R) > N$ **do**
　　随机抽取小批量样本 $\{(s_i, \boldsymbol{l}_i)\}_{i=1}^{n}$
　　for $i = 1$ to n **do**
　　计算欧式距离 $\|\boldsymbol{l}_t - \boldsymbol{l}_i\|$,并搜索 $\boldsymbol{l}_i^{K\text{-}NN}$
　　end
　　计算内部奖励 $r^i(\boldsymbol{\tau}_t) = \log(\|\boldsymbol{l}_t - \boldsymbol{l}_i^{K\text{-}NN}\|_2 + g)$
　　更新 $\alpha_{step} = \alpha_0 \cdot \beta_{step}$
　　计算总奖励值 $r_t^{total} = r^e(\boldsymbol{\tau}_t, \boldsymbol{a}_t) + \alpha_{step} \cdot r^i(\boldsymbol{\tau}_t)$
　end
　将 $s_t = (\boldsymbol{\tau}_t, \boldsymbol{a}_t, \boldsymbol{\tau}_{t+1}, r_t^{total}, \boldsymbol{l}_t)$ 存入重放缓冲区 R
end

2.3　双网络结构及奖励重塑方法设计

D4PG 算法是基于 DDPG 的进一步改进,而 DDPG 又是深度 Q 网络(deep Q-network, DQN) 在连续空间的版本,尽管 D4PG 在保留目标网络、经验回放等 DQN 技巧的同时,也引入了分布式 Q 值估计来改善其表现,但同样也继承了 DQN 的一个核心问题,即 Q 值高估现象。D4PG 算法的目标是在每个状态 $\boldsymbol{\tau}$ 下找到能够使得期望 Q 值最大的动作 $\boldsymbol{a}$,即 $\max_{a \in A}\{E[Z_\theta(\boldsymbol{\tau}, \boldsymbol{a})]\}$,且在实际应用中,样本数据不可避免地会存在噪声 ξ,所以 Q 值高估现象产生的主要原因与策略中的最大化操作以及数据中的噪声密切相关。如果所选择动作的 Q 值包含较大的噪声成分时,会导致该值出现估计偏差,并且会在后续 Q 值更新中被进一步放大,进而导致偏差的累积。这种高估问题会使算法在学习过程中变得不稳定,尤其是在复杂的连续控制任务中,这个问题变得尤为严重。

针对上述高估问题,传统的主流观点是直接将双 Q 学习(double Q-learning) 思想应用到 actor-critic 架构中,通过交替使用两个价值网络对 Q 值进行估计。然而,该方法仍然无法完全避免高估问题。为了解决上述问题,本文提出了双动作选择-同步双 Q 学习-保守优势学习(double-action selection-simultaneous double Q-learning-conservative advantage learning, DSC) 方法,其中同步双 Q 学习-保守优势学习方法对传统的 double Q-learning 进行了改进,不仅引入了两个价值网络同时也使用了两个策略网络。其核心思想是同时对两个价值网络进行更新并通过重塑奖励值的方法来降低估计的偏差,同时提高样本效率。奖励重塑的具体公式如式(19) 所示。

$$\begin{cases} r_1 = r(\boldsymbol{\tau}, \boldsymbol{a}) + \beta \sum_i^N \{\min[\mathrm{var}(Z_{\theta'_1}(\boldsymbol{\tau}, \boldsymbol{a})), \\ \mathrm{var}(Z_{\theta'_2}(\boldsymbol{\tau}, \boldsymbol{a}))] - Z_{\theta'_1}(\boldsymbol{\tau}, \mu_1(\boldsymbol{\tau}))\} V \\ r_2 = r(\boldsymbol{\tau}, \boldsymbol{a}) + \beta \sum_i^N \{\min[\mathrm{var}(Z_{\theta'_1}(\boldsymbol{\tau}, \boldsymbol{a})), \\ \mathrm{var}(Z_{\theta'_2}(\boldsymbol{\tau}, \boldsymbol{a}))] - Z_{\theta'_2}(\boldsymbol{\tau}, \mu_2(\boldsymbol{\tau}))\} V_i \end{cases} \tag{19}$$

式中:$r(\boldsymbol{\tau}, \boldsymbol{a})$ 为即时奖励值;θ'_1、θ'_2 为两个目标价值网络的参数;$\mu_1(\boldsymbol{\tau})$、$\mu_2(\boldsymbol{\tau})$ 为两个策略网络对给定状态 $\boldsymbol{\tau}$ 选择的动作;var 表示方差计算;N 表示 bin 的总数量;V_i 表示 bin 值。β 是一个超参数,本文设定为 0.019。

为了进一步降低高估现象并保证算法的稳定性,本文还创新性的引入了双动作选择(double-action selection, DAS) 方法,该方法使用两个策略网络对输入的状态 $\boldsymbol{\tau}$ 生成两个候选动作,即:$\boldsymbol{a}_1 = \mu_1(\boldsymbol{\tau})$ 和 $\boldsymbol{a}_2 = \mu_2(\boldsymbol{\tau})$。随后,DAS 利用两个独立的价值网络对候选动作进行估计,得到 4 个 Q 值分布,具体如式(20) 所示。

$$\begin{cases} Z_1 = Z_{\theta_1}(\boldsymbol{\tau}, \boldsymbol{a}_1) \\ Z_2 = Z_{\theta_2}(\boldsymbol{\tau}, \boldsymbol{a}_1) \\ Z_3 = Z_{\theta_1}(\boldsymbol{\tau}, \boldsymbol{a}_2) \\ Z_4 = Z_{\theta_2}(\boldsymbol{\tau}, \boldsymbol{a}_2) \end{cases} \tag{20}$$

式中:θ_1、θ_2 表示两个价值网络的参数。对上述 4 个 Q 值分布进行加权求和计算,选择加权和最大的 Q 值分布所

对应的动作并执行，如式(21)、(22)所示。

$$q_i = \sum_{i=1}^{4}\sum_{j=1}^{n} Z_{ij} \cdot V_{ij} \tag{21}$$

$$\boldsymbol{a} = \max(d_1, d_2) \tag{22}$$

式中：Z_{ij} 表示每个 Q 值分布的概率；V_{ij} 表示 bin 值；n 则表示 bin 的总数量；$d_1 = q_1 + q_2$；$d_2 = q_3 + q_4$。通过这种双动作选择机制，能够更有效地选择最优动作，并进一步避免高估问题。DAS 计算流程如图 6 所示。

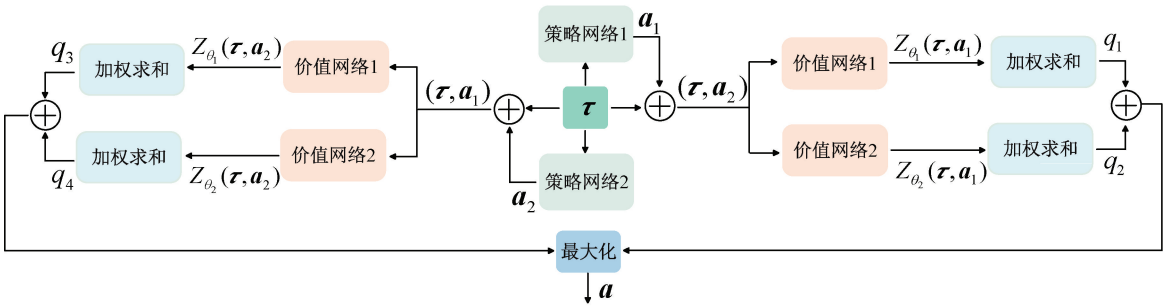


图 6 DAS 过程结构

Fig. 6 DAS process structure diagram

在奖励值重塑及双动作选择后，利用反 Q 值估计方法来计算相应的目标分布值，具体如式(23)所示。

$$\begin{cases}\hat{Z}_1(\boldsymbol{\tau},\boldsymbol{a}) = r_1 + \gamma \cdot Z_{\theta'_2}(\boldsymbol{\tau}', \mu_{\varphi'_1}(\boldsymbol{\tau}'))\\ \hat{Z}_2(\boldsymbol{\tau},\boldsymbol{a}) = r_2 + \gamma \cdot Z_{\theta'_1}(\boldsymbol{\tau}', \mu_{\varphi'_2}(\boldsymbol{\tau}'))\end{cases} \tag{23}$$

式中：r_1、r_2 表示重塑的奖励值。在此基础上，利用式(11)来计算两个价值网络和其目标分布的损失函数，并更新网络参数。综上，本文提出的 DSC 算法的伪代码如算法 2 所示。

算法 2：DSC 算法

初始化价值网络参数 θ_1, θ_2
初始化策略网络参数 φ_1, φ_2
初始化目标网络参数 $(\theta'_1, \theta'_2) \leftarrow (\theta_1, \theta_2)$，$(\varphi'_1, \varphi'_2) \leftarrow (\varphi_1, \varphi_2)$
初始化重放缓冲区 $\boldsymbol{R}$
for $0 < t < T$ **do**
  根据式(22)选择输出的动作 $\boldsymbol{a}_t$
  对动作加入噪声 ε 并执行 $\boldsymbol{a}_t \sim \boldsymbol{a}_t + \varepsilon$，获得新状态 $\boldsymbol{\tau}_{t+1}$ 和奖励 r_t
  将状态转移数组 $(\boldsymbol{\tau}_t, \boldsymbol{a}_t, \boldsymbol{\tau}_{t+1}, r_t)$ 放入重放缓冲区 R
  从重放缓冲区 R 中抽取 N 个小批量状态转移数组样本 $(\boldsymbol{\tau}_i, \boldsymbol{a}_i, \boldsymbol{\tau}_{i+1}, r_i)$
  根据式(19)重塑奖励值 r_1, r_2
  根据式(23)和(11)计算对应的目标分布以及损失函数
  计算 $\nabla_{\theta_i} L(\theta_i)$，$\nabla_{\varphi_i} J(\varphi_i)$，并利用梯度法更新相应参数：
  $\theta_i \leftarrow \theta_i - \lambda_1 \nabla_{\theta_i} L(\theta_i)$，$\varphi_i \leftarrow \varphi_i + \lambda_2 \nabla_{\varphi_i} J(\varphi_i)$
  更新目标网络的参数：
  $\theta'_i \leftarrow \zeta_1 \theta_i + (1 - \zeta_1)\theta'_i$，$\varphi'_i \leftarrow \zeta_2 \varphi_i + (1 - \zeta_2)\varphi'_i$
end

3 仿真与数据分析

3.1 仿真参数

在空间捕获任务中，本文采用 D4PG 算法进行训练，其中智能体数量设定为 10。航天器模型设定为正方体形状，边长为 0.3 m，其质量以及转动惯量等关键参数如表 1 所示。在捕获过程中，考虑到捕获航天器和机械臂的行动存在较大的随机性，因此需要对其运动范围、速度以及加速度进行限制，确保捕获任务可以顺利完成，如表 2 所示。

表 1 关键参数表

Table 1 DSC algorithm pseudo code

物体	a_i/m	b_i/m	m_i/kg	I_i/(kg·m^2)
捕获航天器		0.251 5	11.211 0	2.022×10^{-1}
机械臂 1	0.196 8	0.107 7	0.345 0	3.704×10^{-3}
机械臂 2	0.198 2	0.106 3	0.335 0	3.506×10^{-3}
机械臂 3	0.062 1	0.025 2	0.111 0	1.060×10^{-4}
目标航天器		0.240 6	12.039 0	2.257×10^{-1}

表 2 模型参数限制表

Table 2 Model parameter restriction table

参数	位置	速度	加速度
μ_c	[0 m, 3.5 m]	±0.1 m/s	±0.02 m/s^2
λ_c	[0 m, 2.4 m]	±0.1 m/s	±0.02 m/s^2
q_c	无限制	±π/12 rad/s	±0.05 rad/s^2
q_1	±π/2 rad	±π/6 rad/s	±0.1 rad/s^2
q_2	±π/2 rad	±π/6 rad/s	±0.1 rad/s^2
q_3	±π/2 rad	±π/6 rad/s	±0.1 rad/s^2

针对上述参数变化范围,对捕获航天器、机械臂以及目标航天器的位置以及角度进行初始化设置,并允许在不同回合中进行一定范围内的随机化。具体初始化设置如表 3 所示。特别地,对于两方的速度设置,规定捕获方的初始速度在每回合开始时均都设置为 0,而目标方的角速度则在每回合开始时在 $\pm 10 \times np.pi/180$(rad/s) 范围内随机初始化。

表 3　初始位置和随机化

Table 3　Initial position and randomization

参数	初始位置	随机化
μ_c/m	1.2	±0.05
λ_c/m	1.2	±0.05
q_c/rad	0	±π
q_1/rad	0	±π/2
q_2/rad	0	±π/2
q_3/rad	0	±π/2
μ_t/m	2.3	±0.05
λ_t/m	1.2	±0.05
q_t/rad	0	±π

改进后的 D4PG 算法包含 8 个网络,每个网络均设置为 2 层全连接网络,且每层神经元数量分别为 400 和 300 个。Actor 网络的输入为状态,输出则为动作,网络激活函数设置为 ReLU。Critic 网络的输入为状态-动作对,输出则为奖励分布的 51 个概率值,激活函数同样设置为 ReLU。神经网络的优化器采用 Adam。具体训练过程的超参数以及网络结构超参数设置分别为表 4 和 5 所示。

表 4　训练过程超参数

Table 4　Training process hyperparameters

参数	值
策略网络学习率	0.000 1
价值网络学习率	0.000 1
目标网络更新率	0.001
折扣因子	0.95
最小批量大小	256
重放缓冲区大小	10^6
高斯噪声	0.333 3
时间步长	0.2
每回合最大步数	300
最大回合数	500 000
最大迭代次数	900 000

表 5　网络每层的超参数

Table 5　Hyperparameters of each layer of the network

策略网络层	类型	神经元数量	价值网络层	类型	神经元数量
1	输入层	16	1-1	状态输入	16
2	全连接层	400	2	全连接层	400(连接至 1-1)
3	全连接层	300	3	全连接层	300(连接至 2)
4	输出层	6	1-2	动作输入	6
-	-	-	4	全连接层	神经元:300 (连接至 1-2)
-	-	-	5	相加层	连接至 3 和 4
-	-	-	6	输出层	神经元: bin 的数量

基于上述设定的参数,本文进行了自主决策算法分析实验,来验证所提出方法的可行性以及优越性。

3.2　仿真结果与分析

1)经典环境下算法性能对比

本文在 LunarLanderContinuous-v2 环境下对比评估了 4 种强化学习算法:PPO、TD3、DDPG 以及 D4PG 算法。其中 D4PG 算法采用分布式架构,其余 3 种为传统的单一架构算法。评估过程中分别记录了各算法的最终奖励值、收敛所需回合数以及在 100 次独立测试中的最大奖励值、最小奖励值和平均奖励值。相关结果如表 6 和 7 所示。

表 6　算法性能对比

Table 6　Algorithm performance comparison table

性能指标	PPO	TD3	DDPG	D4PG
奖励值	251.178	240.748	241.359	265.547
收敛回合	218	245	230	163

表 7　100 次测试奖励值对比表

Table 7　Comparison of reward values for 100 tests

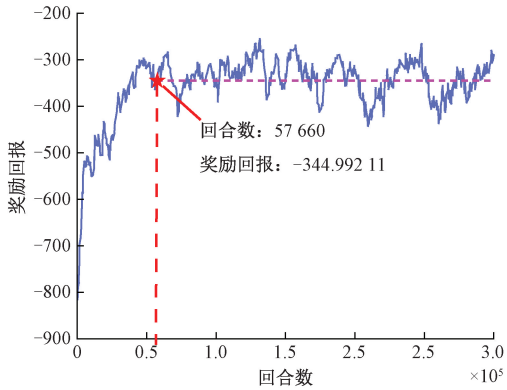
奖励值	PPO	TD3	DDPG	D4PG
最大值	258.418	251.536	257.247	268.625
最小值	243.236	235.376	231.223	259.759
平均值	250.836	243.813	245.459	265.291

由表 6 可知,D4PG 算法的奖励值最终可达到 265.547,显著高于其他算法,表明其具备较优的策略性能。同时,该算法在 163 回合内即可实现收敛,表现出更快的学习效率。进一步从表 7 分析可知,D4PG 算法在 100 次测试中奖励值稳定分布于[259.759,268.625]范围,波动幅度较小,且平均奖励值达到 265.291。相比于其他算法,该算法体现出较强的稳定性与鲁棒性。综上

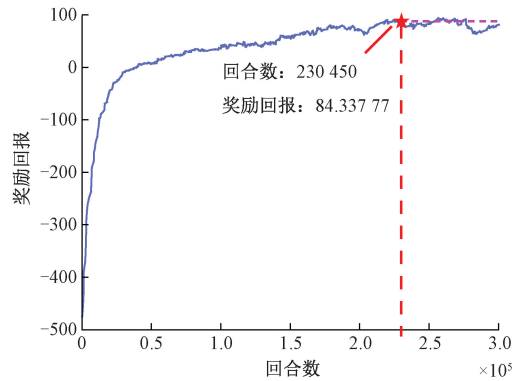
所述,D4PG 算法凭借其分布式架构,在策略表现、收敛速度以及训练稳定性等方面均优于传统的单一结构算法。因此,本文选取 D4PG 作为后续研究空间捕获自主决策问题的基础模型。

2)决策算法优越性验证

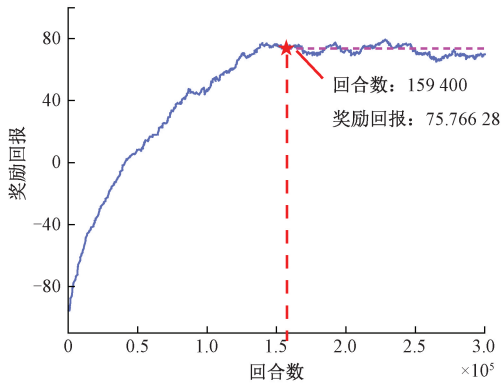
进一步的,在复杂且高维的空间捕获环境中本文对比了 DDPG、常见的分布式双延迟深度确定性策略梯度(distributed twin delayed deep deterministic policy gradient algorithm, DTD3)以及 D4PG 这 3 种强化学习算法,着重分析了 3 种算法的稳定性、奖励值表现以及收敛速度等方面的差异,具体的奖励曲线如图 7 所示。



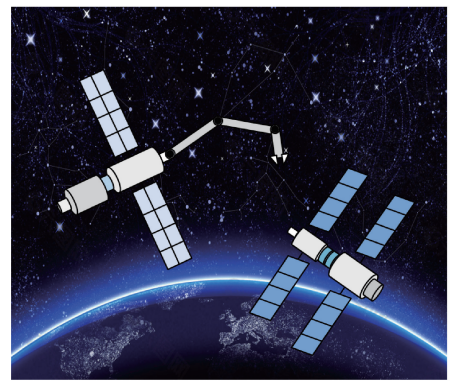
(a) DDPG算法奖励曲线
(a) DDPG algorithm reward curve



(b) DTD3算法奖励曲线
(b) DTD3 algorithm reward curve



(c) D4PG算法奖励曲线
(c) D4PG algorithm reward curve



(d) 捕获示意图
(d) Capture schematic

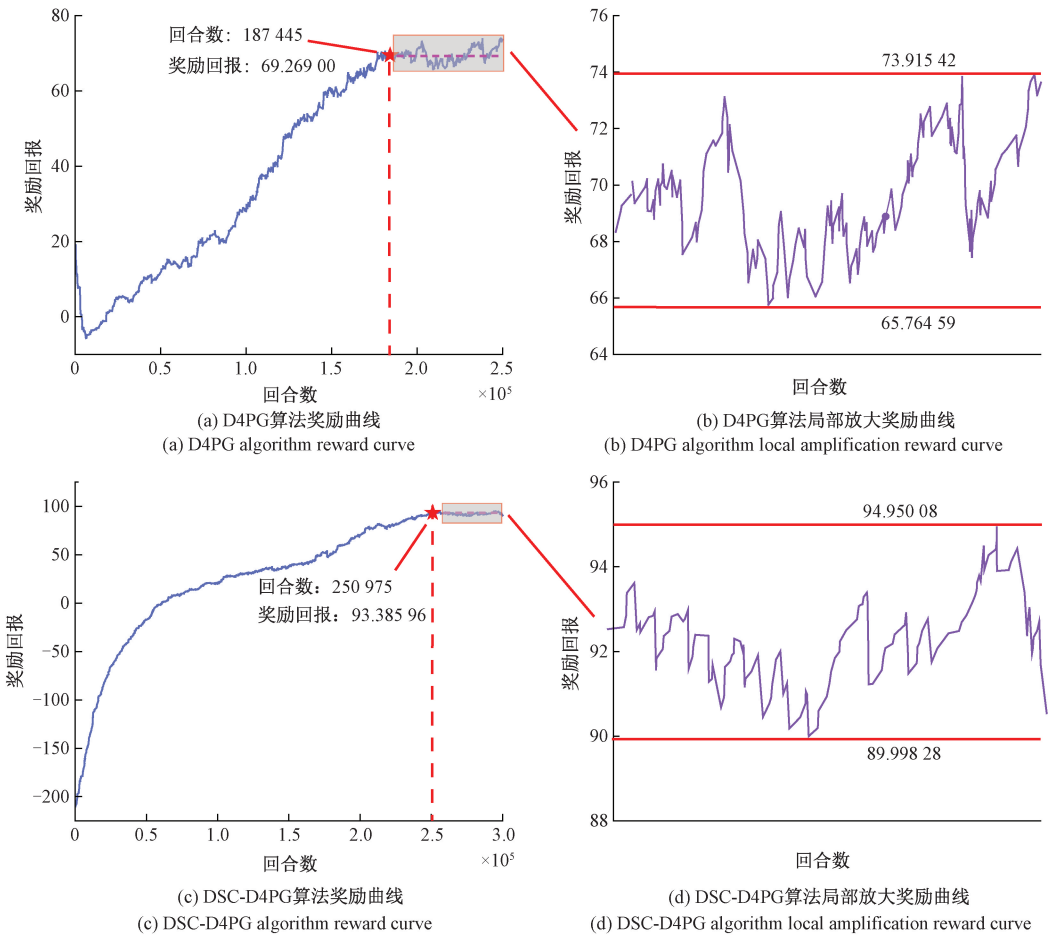
图 7 算法奖励曲线示意图

Fig. 7 Diagram of the algorithm reward curve

图 7(a)展示了 DDPG 算法的奖励曲线,该算法在第 57 660 回合时就开始逐渐收敛,对应的奖励值为 -344.992 11。然而,该算法在收敛过程中仍然在[-424.434 73,-255.108 071]范围存在较大的波动,表现出明显的不稳定性,且奖励值也无法进一步的提升。由于不稳定的策略可能会导致捕获过程中产生一些误操作,从而造成任务失败或资源浪费。因此,为了确保捕获任务的成功,提升算法的稳定性是至关重要的。图 7(b)描绘了 DTD3 算法的奖励曲线,该算法在经历了 230 450 回合训练后开始逐渐收敛,且奖励值从 DDPG 算法的 -344.992 11 提升至 84.337 77。同时,收敛过程中波动范围也缩减至[66.007 36,93.598 45]范围,展现出更好的稳定性。然而,相比较 DDPG 算法而言,DTD3 算法的收敛速度有所减慢,未能满足任务的快速性需求。图 7(c)为 D4PG 算法的奖励曲线,该算法在 159 400 回合开始收敛,奖励值为 75.766 28,且波动幅度也进一步的缩小至[66.938 14,79.327 37],表明算法策略逐渐趋于成熟和稳定。虽然其奖励值略低于 DTD3 算法,但是 D4PG 算法在稳定性方面表现更优,且收敛速度也明显加快。结合以上数据,综合考虑本文任务对收敛性和稳定性的要求,最终选择 D4PG 算法用于空间捕获任务。

3)空间捕获的自主决策仿真

由于 D4PG 算法存在 Q 值估计偏差问题,故本文针对性的提出了解决方法,该方法结合双网络结构进行动作选择以及奖励重塑,旨在有效的降低高估现象的发生,对比结果如图 8 所示。



(a) D4PG算法奖励曲线
(a) D4PG algorithm reward curve

(b) D4PG算法局部放大奖励曲线
(b) D4PG algorithm local amplification reward curve

(c) DSC-D4PG算法奖励曲线
(c) DSC-D4PG algorithm reward curve

(d) DSC-D4PG算法局部放大奖励曲线
(d) DSC-D4PG algorithm local amplification reward curve

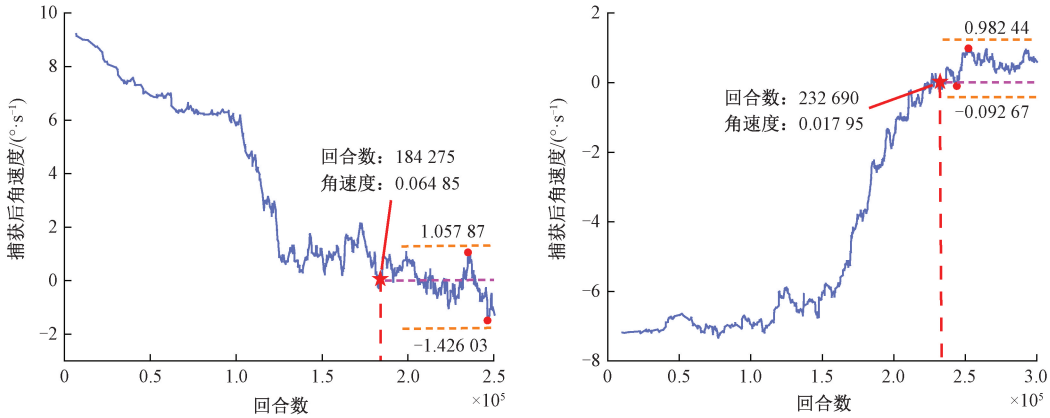
图 8　奖励曲线示意图

Fig. 8　Reward curve diagram

图 8(a)展示了 D4PG 算法的奖励曲线,可以看出该算法在训练初始阶段时奖励值较低,这是因为初始策略表现较差,产生的动作容易使捕获航天器偏离目标航天器或者机械臂超出关节界限,从而受到较大的惩罚。然而,随着训练的进行,策略逐渐优化,奖励值也稳步提升,在 187 445 回合时学到了较优的策略,此时奖励值为 69.269 00。图 8(b)为收敛部分的放大曲线,可以发现该算法在收敛后奖励值在[65.764 59,73.915 42]范围波动,稳定性较差。相比之下,图 8(c)为结合 DSC 后的奖励曲线,该算法在训练 250 975 回合时找到较优的策略,此时收获的奖励值提升至 93.385 96,进一步观察图 8(d)的放大曲线发现改进后的算法波动范围缩减至[89.998 28,94.950 08]。整体来看,结合 DSC 方法后,奖励值提升了约 34.82%,稳定性提高了约 39.25%,但收敛速度相较原始 D4PG 算法有所下降,降低了约 33.88%。

此外,本文还对捕获后组合航天器的角速度进行了记录,具体结果如图 9 所示。由于角速度仅在捕获成功后开始记录,因此曲线并非从 0 开始的。

图 9(a)为 D4PG 算法的角速度曲线,可以观察到在初始策略的作用下,捕获后角速度较大,无法满足将角速度尽可能降至为 0 的需求,但随着策略不断优化,角速度在第 184 275 回合降至为 0.064 85。然而,由于捕获任务比较复杂,角速度无法始终保持在一个稳定值,而是在[-1.426 03,0.982 442]范围进行波动。图 9(b)为改进后算法的角速度曲线,尽管训练回合数有所增加,但角速度的稳定性显著提升,其波动范围缩小至[-0.092 67,0.982 44]。与原始 D4PG 算法相比,角速度稳定性提高了约 55.35%,但收敛速度下降了约 26.27%。由上述分析可知,融合 DSC 方法能够有效提升算法的稳定性,但由于探索能力有限,收敛速度相对减缓,难以满足复杂任务对快速性的要求。因此,为了提升收敛速度,引入基于状态的内部探索机制,以更好的平衡探索与利用,结果如图 10 所示。

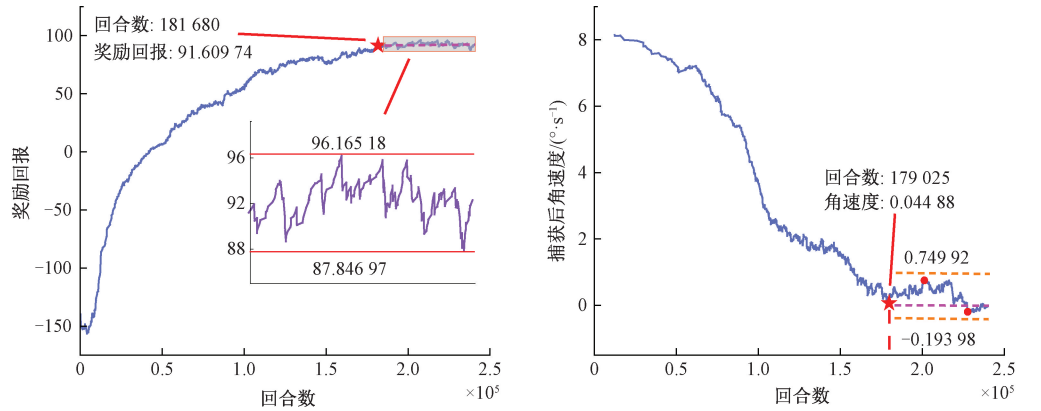


(a) D4PG捕获后角速度曲线
(a) D4PG post-capture angular velocity curve

(b) D4PG+SDQ-CAL捕获后角速度曲线
(b) D4PG+SDQ-CAL post-capture angular velocity curve

图 9 角速度曲线示意图

Fig. 9 Angular velocity curve diagram



(a) SBIRE-DSC-D4PG 算法奖励曲线
(a) SBIRE-DSC-D4PG algorithm reward curve

(b) SBIRE-DSC-D4PG算法捕获后角速度曲线
(b)SBIRE-DSC-D4PG algorithm post-capture angular velocity curve

图 10 奖励和角速度曲线示意图

Fig. 10 Diagram of reward and angular velocity curve

图 10(a)展示了引入 SBIRE 探索机制后的奖励曲线,加入探索机制后,算法在第 181 680 回合便学得较优策略,对应的奖励值为 91.609 74。放大图进一步显示了算法收敛阶段的表现,其奖励值波动范围为[87.846 97, 96.165 18]。与融合 DSC 的算法相比,SBIRE 探索方法将收敛速度提升了约 21.92%;相较原始 D4PG 算法,奖励值提升了约 32.25%,收敛速度提高了约 3.08%。图 10(b)表示捕获后的角速度曲线。结果表明,加入探索机制后,算法在第 179 025 回合即可将角速度降至为 0.044 88,并且维持在[−0.193 98,0.749 92]范围,相比原始 D4PG 算法,收敛速度提高了 2.85%。综上所述,本文所提方法在提升奖励值与加快收敛速度方面均表现出显著优势,有效验证了其整体性能的有效性和优越性。

3.3 真实实验验证

为验证本文所提方法的有效性和可靠性,本文开展了仿真迁移实验。实验平台由固定底座的 UR5 机械臂、Robotiq-2f-140 二指夹爪以及深度相机构成。由于仿真环境为二维,而实际实验处于三维空间中,故在实验中进行了扩维处理:即在决策网络模型的输入端额外引入由相机实时获取目标航天器的 z 轴位置信息,从而实现机械臂在三维工作空间内的目标捕获。实验过程中,深度相机实时采集图像信息并传输至运动控制主线程,作为决策网络的输入。经过训练的决策网络能够对输入的信息进行实时推理,生成机械臂末端的可靠追踪位姿;随后,通过逆运动学算法将该位姿转换为各关节的运动角度,实现机械臂的精确控制。但由于地面实验无法真实模拟太空中物体的失重环境,因此该实验与仿真实验在部分条件上存在差异:在仿真中目标航天器具有自旋状态,而在本实验中,则对应为目标航天器在任意姿态下的空间捕获任务。实验结果如图 11 和表 8 所示。

图 11　不同姿态捕获实验结果

Fig. 11　Experimental results of different pose capture

表 8　不同姿态捕获实验数据结果

Table 8　Experimental data results of different posture capture

姿态 1 时间/s	姿态 1 奖励值	姿态 2 时间/s	姿态 2 奖励值	姿态 3 时间/s	姿态 3 奖励值
0	0	0	0	0	0
5	0	6	0	5	0
10	0	12	0	10	0
15	0	18	0	15	0
20	0	24	0	20	0
22	96.2	27	98.3	23	94.8

由图 11 和表 8 可知,无论目标航天器处于何种姿态,机械臂均能顺利完成捕获任务。在整个实验过程中,初始阶段的奖励值始终保持稳定,表明系统未受到惩罚;当捕获成功后,奖励值显著上升,表明捕获任务顺利完成并获得了较高的奖励值回报。该结果充分表明,本文所提出的决策策略在不同姿态下均具备良好的有效性和鲁棒性,进一步验证了本文所提方法的可行性与可靠性。

4　结　　论

本文将分布式深度确定性策略梯度算法与基于状态的内部探索和双网络结构以及奖励重塑方法相结合研究了空间捕获任务中自主运动决策问题,研究成果及结论总结为:

1)分布式深度确定性策略梯度算法能够提升经验池中信息的多样性以及算法的学习效率,使捕获方具备类人行为的决策能力,从而更好地应对各种突发情况。

2)双网络结构以及奖励重塑方法有效的降低了估计偏差,增强了算法的稳定性。

3)基于状态的内部探索方法使算法在探索与利用之间实现了更好的平衡,有效提升了算法的收敛速度。

未来的研究将进一步的扩展目标方的自主能力,使其能够自主决策以躲避捕获方,从而形成比较复杂的捕获机制。此外,如何在捕获后保持整体航天器的角速度始终为 0,从而实现更高的稳定性,也将是后续工作的重点方向之一。

参考文献

[1] 周俊峰. 基于微分对策理论的航天器追逃控制方法研究[D]. 哈尔滨:哈尔滨工程大学, 2021.
ZHOU J F. Research on control method for spacecraft pursuit-evasion based on differential game theory[D]. Harbin: Harbin Engineering University, 2021.

[2] 孙永军,王钤, 刘伊威, 等. 空间非合作目标捕获方法综述[J]. 国防科技大学学报, 2020, 42(3): 74-90.
SUN Y J, WANG Q, LIU Y W, et al. A survey of non-cooperative target capturing methods[J]. Journal of National University of Defense Technology, 2020, 42(3): 74-90.

[3] ZHANG X Y, LIU X F, WANG M M, et al. A new strategy for capturing a noncooperative spacecraft by a robotic arm[J]. Multibody System Dynamics, 2023, 59(2): 143-169.

[4] 张宇, 张豪杰, 程彬, 等. 空间柔性绳网多碎片捕获动力学研究[J]. 宇航总体技术, 2023, 7(6): 42-50.
ZHANG Y, ZHANG H J, CHENG B, et al. Capture dynamics simulations of a flexible net for multiple space debris pieces removal[J]. Astronautical Systems Engineering Technology, 2023, 7(6): 42-50.

[5] 陈鹏旭. 飞网式空间目标捕获技术研究[D]. 沈阳:沈阳工业大学, 2023.
CHEN P X. Research on capture technology of tether-net for space targets[D]. Shenyang: Shenyang University of Technology, 2023.

[6] 惠倩倩. 空间飞爪在轨捕获碰撞动力学及抓捕后稳定控制研究[D]. 西安:长安大学, 2024.
HUI Q Q. Research on collision dynamics and post-capture stabilization control of space claws for on-orbit capture[D]. Xi′an : Chang′an University, 2024.

[7] ZENG L, GAO B, ZHU CH, et al. Design and validation of maintenance system for space manipulator grapple fixture[C]. 2023 14th International Conference on Reliability, Maintainability and Safety, 2023: 938-942.

[8] ISAACS R. Differential games: A mathematical theory

with applications to warfare and pursuit, control and optimization[M]. Courier Corporation, 1999.

[9] 杜雪松, 张东俊, 诸葛浩. 拦截高超声速飞行器的微分对策制导律[J]. 航天控制, 2024, 42(3): 29-34.
DU X S, ZHANG D J, ZHUGE H. Differential game guidance law for intercepting hypersonic vehicle[J]. Aerospace Control, 2024, 42(3): 29-34.

[10] 罗亚中, 李振瑜, 祝海. 航天器轨道追逃微分对策研究综述[J]. 中国科学:技术科学, 2020, 50(12): 1533-1545.
LUO Y ZH, LI ZH Y, ZHU H. Survey on spacecraft orbital pursuit-evasion differential games[J]. Scientia Sinica(Technologica), 2020, 50(12): 1533-1545.

[11] 耿远卓, 袁利, 黄煌, 等. 基于终端诱导强化学习的航天器轨道追逃博弈[J]. 自动化学报, 2023, 49(5): 974-984.
GENG Y ZH, YUAN L, HUANG H, et al. Terminal-guidance based reinforcement learning for orbital pursuit-evasion game of the spacecraft[J]. Acta Automatica Sinica, 2023, 49(5): 974-984.

[12] 王若冰, 王晓芳. 一种结合 MADDPG 和对比学习的无人机追逃博弈方法[J]. 宇航学报, 2024, 45(2): 262-272.
WANG R B, WANG X F. An algorithm for UAV pursuit-evasion game based on MADDPG and contrastive learning[J]. Journal of Astronautics, 2024, 45(2): 262-272.

[13] CHEN J L, JIANG Y, PAN H R, et al. Path planning in complex environments using attention-based deep deterministic policy gradient[J]. Electronics, 2024, 13(18): 3746.

[14] 毛剑琳, 贺志刚, 张书凡, 等. 面向多到一任务交接的多机器人协作路径规划算法[J]. 仪器仪表学报, 2024, 45(9): 237-248.
MAO J L, HE ZH G, ZHANG SH F, et al. Multi-robot collaborative path planning algorithm for many-to-one task handover[J]. Chinese Journal of Scientific Instrument, 2024, 45(9): 237-248.

[15] 宁睿, 刘志晨, 刘毅, 等. 强化学习引导变导纳控制的机械臂精密轴孔装配模糊位姿估计与精确调整[J]. 仪器仪表学报, 2024, 45(11): 170-177.
NING R, LIU ZH CH, LIU Y, et al. Reinforcement learning guided variable admittance control for fuzzy pose estimation and precise adjustment of robotic precise peg-in-hole assembly[J]. Chinese Journal of Scientific Instrument, 2024, 45(11): 170-177.

[16] 候磊, 贾贝熙, 杜子亮, 等. 深度强化学习在无人机智能控制中的应用研究[J]. 战术导弹技术, 2024(6): 107-117.
HOU L, JIA B X, DU Z L, et al. Application research of deep reinforcement learning in intelligent control of unmanned aerial vehicle[J]. Tactical Missile Technology, 2024(6): 107-117.

[17] ZHAO SH W, WANG J CH, XU H T, et al. Composite observer-based optimal attitude-tracking control with reinforcement learning for hypersonic vehicles[J]. IEEE Transactions on Cybernetics, 2023, 53(2): 913-926.

[18] JIA L Y, CAI CH T, WANG X M, et al. Multi-intent autonomous decision-making for air combat with deep reinforcement learning[J]. Applied Intelligence, 2023, 53(23): 29076-29093.

[19] 王梓屹, 吕江涛. 一种基于射向约束的预警卫星弹道估计方法[J]. 电子测量技术, 2022, 45(5): 152-156.
WANG Z Y, LYU J T. Method for estimating missile trajectory by early warning satellite based on direction constraint[J]. Electronic Measurement Technology, 2022, 45(5): 152-156.

[20] 向贤财. 基于强化学习的无人机空战非完全信息博弈模型研究[D]. 北京:中国科学院大学, 2024.
XIANG X C. Research on the game model under incomplete information for UAV air combat based on reinforcement learning[D]. Beijing: University of Chinese Academy of Sciences, 2024.

[21] D'AMBROSIO M, CAPRA L, BRANDONISIO A, et al. Redundant space manipulator autonomous guidance for in-orbit servicing via deep reinforcement learning[J]. Aerospace, 2024, 11(5): 341.

[22] SONG B Y, LI J Q, LIU X Y, et al. A trajectory planning method for capture operation of space robotic arm based on deep reinforcement learning[J]. Journal of Computing and Information Science in Engineering, 2024, 24(9): 091003.

[23] PENG ZH T, WANG CH. Reinforcement learning-based pose coordination planning capture strategy for space non-cooperative targets[J]. Aerospace, 2024, 11(9): 706.

[24] BLAISE J, BAZZOCCHI M C F. Space manipulator collision avoidance using a deep reinforcement learning

control[J]. Aerospace, 2023, 10(9): 778.
[25] BARTH-MARON G, HOFFMAN M W, BUDDEN D, et al. Distributed distributional deterministic policy gradients[J]. ArXiv preprint arXiv:1804.08617, 2018.
[26] CRAIN A, ULRICH S. Experimental validation of pseudospectral-based optimal trajectory planning for free-floating robots[J]. Journal of Guidance, Control, and Dynamics, 2019, 42(8): 1726-1742.
[27] 韩莹, 陈熙. 一种基于融合特征聚类和随机配置网络的轴承剩余寿命预测方法[J]. 电子测量与仪器学报, 2024, 38(4): 128-139.
HANG Y, CHEN X. Bearing residual life prediction method based on fusion feature clustering and stochastic configuration networks[J]. Journal of Electronic Measurement and Instrumentation, 2024, 38(4): 128-139.
[28] 陈玉明, 蔡国强, 卢俊文, 等. 一种邻域粒 K 均值聚类方法[J]. 控制与决策, 2023, 38(3): 857-864.
CHEN Y M, CAI G Q, LU J W, et al. A neighborhood granular K-means clustering method[J]. Control and Decision, 2023, 38(3): 857-864.

作者简介

黄成(通信作者),2012 年于哈尔滨工程大学获得硕士学位,2018 年于哈尔滨工业大学获得博士学位,现为哈尔滨理工大学教授,主要研究方向为航天器姿轨控制及自主交会对接、空间非合作目标捕获、群体智能。
E-mail:huangchengsunxi@163.com

Huang Cheng (Corresponding author) received his M. Sc. degree from Harbin Engineering University in 2012 and his Ph. D. degree from Harbin Institute of Technology in 2018. He is currently a professor at Harbin University of Science and Technology. His research interests include spacecraft attitude and orbit control and autonomous rendezvous and docking, space non-cooperative target acquisition, and swarm intelligence.

殷振凯,2019 年于哈尔滨理工大学获得学士学位,现为哈尔滨理工大学研究生,主要研究方向为深度强化学习、航天器空间捕获和轨道转移。
E-mail:lxyz20190608@163.com

Yin Zhenkai received his B. Sc. degree from Harbin University of Science and Technology in 2019. He is currently a graduate student at Harbin University of Science and Technology. His main research interests include deep reinforcement learning, spacecraft capture, and orbit transfer.

邢爱佳,2022 年于内蒙古科技大学获得学士学位,现为哈尔滨理工大学研究生,主要研究方向为深度强化学习、机器人智能控制、群体智能控制.
E-mail:xingaijia2022@163.com

Xing Aijia received her B. Sc. degree from Inner Mongolia University of Science and Technology in 2022. She is currently a graduate student at Harbin University of Science and Technology. Her research focuses on deep reinforcement learning, robot intelligent control, and swarm intelligent control.

于智龙,2007 年于哈尔滨理工大学获得硕士学位,2013 年于东北林业大学获得博士学位,现为哈尔滨理工大学副教授,主要研究方向为输配电系统状态监测、人工智能与机器学习。
E-mail:zlyu@hrbust.edu.cn

Yu Zhilong received his M. Sc. degree from Harbin University of Science and Technology in 2007 and his Ph. D. degree from Northeast Forestry University in 2013. He is currently an associate professor at Harbin University of Science and Technology. His main research interests include power transmission and distribution system condition monitoring, artificial intelligence and machine learning.