航 空 学 报
Acta Aeronautica et Astronautica Sinica
Feb. 15 2025 Vol. 46 No. 3
ISSN 1000-6893 CN 11-1929/V
http://hkxb.buaa.edu.cn hkxb@buaa.edu.cn

引用格式：杨敏，刘关俊，周子渊. 基于安全强化学习的月球着陆器控制[J]. 航空学报，2025，46(3)：630553. YANG M，LIU G J，ZHOU Z Y. Control of lunar landers based on secure reinforcement learning[J]. Acta Aeronautica et Astronautica Sinica，2025，46(3)：630553(in Chinese). doi:10.7527/S1000-6893.2024.30553

深空光电测量与智能感知技术专栏

基于安全强化学习的月球着陆器控制

杨敏，刘关俊*，周子渊

同济大学 计算机科学与技术系，上海 201804

摘 要：在月球着陆任务中，着陆器必须在极端环境下进行精确操作，并且通常面临着通信延迟的挑战，这些因素严重限制了地面控制的实时操作能力。针对这些挑战，研究提出了一种基于半马尔可夫决策过程(SMDP)的深度强化学习安全性提升框架，旨在提高航天器自主着陆的操作安全性。为了实现状态空间的压缩并保持决策过程的关键特征，该框架通过对历史轨迹的马尔可夫决策过程(MDP)压缩成SMDP，并根据压缩后的轨迹数据构建抽象SMDP状态转移图，然后识别潜在风险的关键状态-动作对，并实施实时监控及干预，有效提高了航天器的自主着陆安全性。采用了反向广度优先搜索方法，搜索出对任务结果有决定性影响的状态-动作对，并通过搭建的状态-动作监控器实现对模型的实时调整。实验结果显示，该框架在不需增加额外传感器或显著改变现有系统配置的条件下，能够在预训练的深度Q网络(DQN)、Dueling DQN、DDQN模型上，提升月球着陆器在模拟环境中的任务成功率高达22%，在预设的安全性评价标准下，该框架能提升最高42%的安全性。此外，虚拟环境中的模拟结果展示了该框架在月球着陆等复杂航天任务中的实际应用潜力，可以有效提升操作安全性和效率。

关键词：深度强化学习；自主着陆；抽象SMDP状态转移图；安全性提升；实时监控；反向广度优先搜索

中图分类号：V448；TP391.9 **文献标识码**：A **文章编号**：1000-6893(2025)03-630553-14

在执行地外天体探测任务时，确保航天器的安全着陆至关重要[1]。然而，由于太空恶劣的环境及其高度不确定性，例如存在不同行星的地形多样性和复杂的大气条件等因素，极大地增加了着陆过程的复杂性。此外，由于航天器与地球之间存在的通信延迟限制，航天器必须依赖机载传感器、控制系统进行精确的导航、控制。同时，航天器燃料的限制进一步给航天器的安全着陆带来了困难。为了确保在极端条件下航天器也能够成功着陆[2]，设计具备高度自主性、适应性的航天器系统变得尤为迫切。因此，在航天器的自主着陆任务中，除了采用先进的传感器、成像技术来实时感知着陆环境，还需要不断提高航天器的通信能力，确保操作的实时性。另外，在面对动态变化的环境时，要保证航天器安全着陆还必须采用高效的控制策略[3]。这些控制策略不仅要处理复杂的环境因素，还需要应对不可预见的外部干扰。

为了提高航天器着陆的准确性和安全性，研究者已经探索了不同的控制算法，这些算法也被用于轨迹生成、轨迹跟踪。例如，文献[4]提出了自抗扰控制器(Active Disturbance Rejection

收稿日期：2024-04-19；**退修日期**：2024-05-07；**录用日期**：2024-07-24；**网络出版时间**：2024-08-21 09:42
网络出版地址：https://hkxb.buaa.edu.cn/CN/Y2025/V46/I3/630553
基金项目：国家自然科学基金(62172299，62032019)；北京控制工程研究所空间光电测量与感知实验室开放基金(LabSOMP-2023-03)；中央高校基本科研业务费专项资金(2023-4-YB-05)；上海市科技创新行动计划(22511105500)
* 通信作者. E-mail：liuguanjun@tongji.edu.cn

Control, ADRC)，其继承了比例积分微分(Proportional-Integral-Derivative, PID)的优点，通过对ADRC处理大气模型和飞行器空气动力误差的能力的研究，表明ADRC比PID更适合跟踪问题。文献[5]首次在火星大气进入引导中应用扩展状态观测器来估计不确定性，并提出了一种终端滑模控制(Terminal Sliding Mode Control, TSMC)算法来跟踪参考轨迹，通过结合扩展状态观测器和TSMC，实现了状态对参考轨迹的快速收敛。文献[6]提出了基于障碍Lyapunov函数(Barrier Lyapunov Function, BLF)的滑模控制，用于火星大气进入轨迹跟踪，并证明了在存在外部干扰、输入饱和的情况下具有有限收敛时间的稳定性。文献[7]提出了固定时间非奇异终端滑模(Fixed-Time Nonsingular Terminal Sliding Mod, FTNTSM)面，并设计了连续自适应固定时间非奇异终端滑模控制(Continuous Adaptive FTNTSM Control, AFTNTSMC)方法，特别适合在不确定情况下的火星着陆跟踪控制。

与其他控制算法一样，优化和预测控制算法也被用于航天器着陆问题。文献[8]利用几何力学和非线性模型预测控制(Nonlinear Model Predictive Control, NMPC)技术，解决了航天器在月球着陆阶段的精确着陆问题。文献[9]采用了融合鲁棒Tube模型预测控制(Model Predictive Control, Tube-MPC)思想的序列凸优化方法，专门针对火星精确着陆动力下降段的自主轨迹规划进行研究。文献[10]开发的基于动力下降阶段的火星着陆约束优化预测控制算法，即显式MPC(Explicit Model Predictive Control, EMPC)，也进一步推动了火星着陆器的研究。文献[11]提出了动态安全裕度指标，其中安全裕度指标考虑并定量描述了着陆器的状态不确定性，进一步推导了避险引导律。这些文献展示出在实际航天器着陆任务中，依赖精确的环境模型、预先设定的参数能够提高航天器着陆准确性，但它们在灵活性、独立性方面存在限制，特别是着陆器的控制策略通常通过离线计算完成，并由地面控制中心上传至着陆器，在存在通信延迟的太空环境中这种方法限制了着陆器在面对未知情况时的应对能力。

为了提高着陆器的自主性并实现实时控制，文献[12]提出了一种基于元启发式粒子群优化(Particle Swarm Optimization, PSO)的次优指导，通过逆动力学、轨迹多项式近似的组合，克服了实时机载应用中的计算成本高昂的问题。然而，文献[12]中算法的效率依赖于对环境动力学模型的准确知识，这在实际高度动态的行星环境中，获取这些信息往往非常困难。因此，面对高度动态、不可预测的环境条件时，实时性、灵活性成为设计航天器控制系统时的关键需求。

近年来，强化学习以其强大的自适应能力，为解决这些挑战提供了新的可能性[13-14]。强化学习允许航天器与复杂不确定的环境进行交互学习逐渐优化控制策略，并能根据传感器的输入作出实时的决策，能够更好地适应太空中未知或变化的环境条件[15-16]。尽管强化学习在计算资源方面的需求可能较高，但一旦模型训练完成，它可以迅速部署到航天器上，并支持实时操作。例如，文献[17]引入了一种采用演员-间接(Actor-Indirect)架构的交互式深度强化学习算法，提升了处理月球燃料最优着陆问题的实时性能。文献[18]探讨了将深度强化学习算法用于航天器决策问题的可行性，提出了一种保证安全性的自主航天器指令、控制方法，表现出深度强化学习相比于其他黑箱优化工具或启发式方法的优势。文献[19]提出了一个能够自主导航并安全着陆在月球表面的基于强化学习的月球着陆器控制系统，表现出强化学习在开发月球着陆器自主控制系统方面的潜力。文献[20]通过引导成本学习(Guided Cost Learning, GCL)算法，并基于生成的专家演示集，成功地为月球着陆器环境生成了反映期望行为的奖励函数。文献[21]通过评估深度Q网络(Deep Q-Network, DQN)、双重深度Q网络(Double Deep Q-Network, DDQN)、策略梯度方法在月球着陆问题上的应用，证明了这些深度强化学习(Deep Reinforcement Learning, DRL)算法能够促进成功且燃油高效的航天器着陆。

然而，上述基于强化学习的月球着陆器控制策略主要专注于提升算法效率，快速实现高奖励

收敛，但将训练好的强化学习模型部署于着陆器时，在安全性方面仍然存在缺陷，这是由于在训练过程中未显式地考虑安全约束，难以保证着陆策略能避免潜在的危险后果。为了应对这一挑战，研究者提出了使用约束马尔可夫决策过程（MDP）框架，在算法的优化过程中直接加入安全约束。该框架通过要求预期累积成本维持在预定阈值下，来诱导更安全的行为。然而，这种方法允许在不超过设定阈值的情况下存在安全违规，这可能仍然不适合于航天器着陆这种安全关键应用[22]。因此，为了进一步提升安全性，文献[23]提出了一种改进的概率约束强化学习方法，通过引入更精确的梯度表达式、降低方差的技术，增强了算法可靠性、安全性。此外，文献[24]提出了一个数据驱动安全层，该层使用预测控制、可达性分析作为决策过程中的过滤器，来确保在现实世界中实现高安全性标准。

虽然现有的安全强化学习方法能提高了系统的安全性，但在实际航天器着陆环境中的应用仍面临复杂性、开发成本大的问题[25]。针对这一问题，为确保整个DRL系统的安全性，文献[26]提出了循环训练框架Trainify，该框架通过形式化验证方法来保证DRL系统的安全性。然而，Trainify框架依赖于准确的环境动态模型[27]，在许多实际应用中会受到限制。除了形式化验证，测试方法也被用来发现强化学习模型的缺陷，文献[28-29]提出了一种基于搜索的方法，通过定位参考轨迹来解决基于DRL的任务，以评估DRL智能体如何避免错误动作。这种方法虽然在马里奥游戏环境中表现出色，但它依赖于专家轨迹作为指导，并且难以应用于动力学未知的航天器着陆任务。因此，现有的测试方法难以在未知动力学环境中有效地识别、处理关键的状态-动作对，面对实际的航天器着陆任务可能会导致不良结果，例如着陆器在回合结束发生坠毁或者没有精准降落等情况。

针对月球着陆器环境提出一种DRL安全性提升框架，该框架首先通过状态-动作压缩模块将原始的马尔可夫决策过程（MDP）转化为更易于处理的半马尔可夫决策过程（Semi-MDP，SMDP），通过K-means聚类将状态空间划分为有限的状态簇。在此基础上，构建抽象SMDP状态转移图，并采用反向搜索方法从失败或不利的终止状态开始，识别出对任务失败具有决定性影响的关键状态-动作对。所提框架中的策略纠正方法不依赖于准确的环境动态模型，以期在预训练模型未达到最优效果时，通过较小的干预提高模型的性能。

1 模型框架

针对月球着陆器环境，提出一个安全性提升框架，专门用于从历史轨迹数据中识别深度强化学习策略中关键的抽象状态-动作对，如图1所示。该框架提供了实时监控和干预能力，可以显著提高DRL系统在处理月球着陆器任务时的安全性。

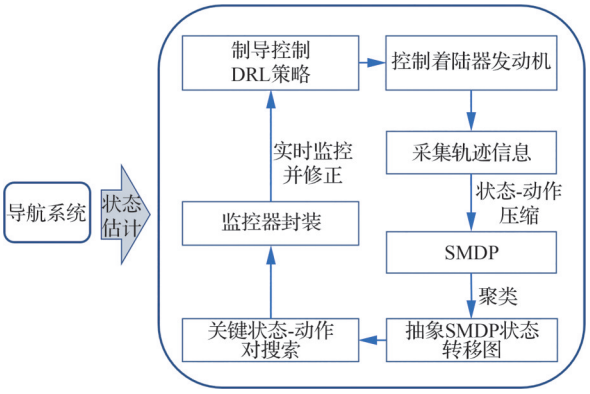


图1 安全性DRL提升框架

Fig. 1 Safety DRL improvement framework

首先，通过测试收集预训练的月球登陆器模型的执行轨迹。随后，通过反向广度优先搜索算法来搜索对月球着陆任务结果或奖励有决定性影响的抽象状态-动作对。这个过程完全依赖于环境状态、着陆器的动作和最终的任务成功/失败，将价值函数或策略网络当作黑盒，不考虑其复杂性。然后，将识别的关键状态-动作对封装到监控器中。监控器可以在模型的下一次执行中实施实时监控和纠正，以提高现有预训练深度强化学习模型在月球着陆器环境中的安全性。安全性DRL提升框架主要包括3个步骤。

1）状态-动作压缩

为了减少状态空间的复杂性，同时保持决策过程的关键特征，采用状态-动作压缩模块将原始

的状态动作的马尔可夫决策过程(MDP)简化为半马尔可夫决策过程(SMDP),并应用K-means聚类算法,将状态空间划分为有限数量的状态簇。基于这些状态簇,构建一个抽象SMDP状态转移图,其中状态对应于抽象后的状态,转移概率根据从历史轨迹数据中观察到的频率进行估计。

2) 识别关键状态-动作对

利用抽象SMDP状态转移图,采用反向广度优先搜索方法,从失败状态或终止的不利状态开始回溯,以识别对任务结果或奖励有关键影响的状态-动作对。这个步骤旨在确定那些关键的抽象状态、可能导致不良结果的关键动作。

3) 实时监控和纠正

将识别出的关键状态-动作对整合到监控器中,这些监控器在DRL算法执行期间进行实时监控。一旦检测到模型在关键抽象状态下,要做出不良动作,监控器就会启动纠正机制来干预、改变模型的决策,引导DRL系统趋向结果更优。

2 深度强化学习

在强化学习中,问题通常被表述为一个马尔科夫决策过程[30],可表示为一个四元组(S,A,P,R),分别表示状态空间、动作空间、状态转移函数、奖励函数。强化学习的目标是通过优化策略π来最大化累积预期$\mathbb{E}\left[\sum_{t=0}^{\infty}\gamma^t R_t\right]$,其中,$\gamma$为折扣因子;$\sum_{t=0}^{\infty}\gamma^t R_t$为从时间步$t$到无穷远未来的累积奖励。

DQN算法是强化学习中的一个基础算法[31],基于Q学习的框架,核心是使用深度神经网络来近似Q函数,损失函数为

$$L(\theta)=\mathop{\mathbb{E}}_{(s,a,r,s')\sim U(D)}\left[\left(r+\gamma\max_{a'}Q(s',a';\theta^-)-Q(s,a;\theta)\right)^2\right] \tag{1}$$

式中:$U(D)$为从经验回放池D中采样得到的样本批次,存储了过去的转移记录,每一条记录包含在给定状态s下采取动作a后得到的奖励r和下一个状态s';$r+\gamma\max_{a'}Q(s',a';\theta^-)$为目标$Q$值;$Q(s,a;\theta)$为当前策略下对状态$s$、动作$a$的$Q$值预测;$\theta^-$、$\theta$为网络参数。DQN通过最小化预测$Q$值与目标$Q$值之间的平方误差来更新网络参数。

Dueling DQN[32]在DQN的基础上,将网络结构分为两个路径:一个计算状态值函数$V(s;\theta,\beta)$,另一个计算每个动作的优势函数$A(s,a;\theta,\alpha)$,Q值的计算公式为

$$Q(s,a;\theta,\alpha,\beta)=V(s;\theta,\beta)+\left(A(s,a;\theta,\alpha)-\frac{1}{|A|}\sum_{a'}A(s,a';\theta,\alpha)\right) \tag{2}$$

式中:α、β为网络参数。

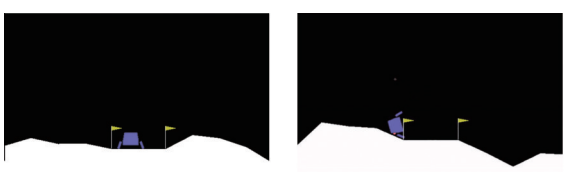
DDQN解决了标准DQN中可能出现的过高估计Q值的问题[33],在选择、评估最佳动作时使用了2个网络

$$Q(s,a;\theta)=r+\gamma Q\left(s',\ \arg\max_{a'}Q(s',a';\theta);\ \theta^-\right) \tag{3}$$

本文主要在这3个深度强化学习算法上进行安全性提升。

3 月球着陆器环境

月球着陆器环境LunarLander-v2是由Open AI Gym提供的一个物理模拟环境[34],基于Box2D实现,模拟了着陆器在月球表面着陆的场景,环境包含月球重力、风力、风向扰动力。如图2所示,LunarLander-v2环境包含一个具有3个不同方向的引擎的着陆器,分别在左侧、右侧、底部,月球表面的两面旗帜表示着陆位置。着陆器的初始状态从视图顶部中心位置开始,具有随机的初始力。目标是控制着陆器引擎能够安全平稳着陆在指定区域,同时最小化燃料消耗,并避免着陆过程中的碰撞或过快下降。图2(a)显示了着陆器成功着陆到指定位置的情况,图2(b)显示了着陆器任务失败的实例情形。



(a) 安全着陆 (b) 失败着陆

图 2 LunarLander-v2环境

Fig. 2 LunarLander-v2 environment

3.1 状态空间

在月球着陆器仿真环境中，状态空间 $\boldsymbol{S}$ 为所有可能出现的状态的集合，每个状态由8个实数值构成，分别为着陆器中心的 X 轴坐标 x（相对水平位置）、Y 轴坐标 y（相对高度）、沿 X 轴的速度 v_x、沿 Y 轴的速度 v_y、角度 θ'、角速度 ω'，以及左、右腿接触地面的二元指示变量（1表示接触、0表示未接触）g_{l}、g_{r}。其中状态的各个维度的约束为

$$\begin{cases} x, y \in [-1.5, 1.5] \\ v_x, v_y \in [-5.0, 5.0] \\ \theta' \in [-\pi, \pi] \\ \omega' \in [-5.0, 5.0] \\ g_{\mathrm{l}}, g_{\mathrm{r}} \in \{0, 1\} \end{cases} \tag{4}$$

3.2 动作空间

动作空间 A 由4个离散动作组成，可以表示为 $A=\{0,1,2,3\}$，其中动作0表示不激活任何引擎；动作1表示激活主引擎，对着陆器产生向上推力，动作2表示激活左侧引擎，产生向右推力，动作3表示激活右侧引擎，产生向左推力。

3.3 奖励函数

奖励函数 $R(s,a,s')$ 定义了在状态 s 执行动作 a 并转移到状态 s' 时获得的即时奖励。奖励函数的设置旨在鼓励安全着陆并惩罚燃料消耗、过快下降、角度偏离。通过这种奖励机制，强化学习算法可以在探索与利用之间找到平衡，学习如何有效地控制着陆器安全着陆。具体地，奖励包括由以下分奖励组成，用于量化不同的状态和动作对总奖励的贡献，计算公式为

$$\begin{cases} R_{\text{距离}} = -100\sqrt{x^2+y^2} \\ R_{\text{速度}} = -100\sqrt{v_x^2+v_y^2} \\ R_{\text{角度}} = -100|\theta'| \\ R_{\text{腿接地}} = 10g_{\mathrm{l}} + 10g_{\mathrm{r}} \\ R_{\text{引擎}} = -0.3m_{\text{power}} \end{cases} \tag{5}$$

除了上述即时奖励外，当着陆器成功着陆或坠毁时，则该回合终止，并获得额外的终止奖励（−100或者+100），如果着陆器成功着陆，环境会返回较大的正奖励（+100），如果着陆器坠毁，环境会返回较大的负奖励（−100）。

3.4 终止条件

终止条件的设定是为了确保DRL算法能在合理的时间内从每个仿真回合中学习到有效的策略，在任务失败或成功时结束仿真回合，以进行下一个回合的学习。LunarLander-v2环境的终止条件有以下4个：

1）着陆器成功着陆。

2）着陆器与月球表面不是通过腿接触，则认为是坠毁。

3）着陆器的水平位置超出了边界，则认为着陆器超出视野。

4）着陆器在固定的时间步内保持静止，例如因为速度过低而进入休眠状态。

为增强仿真效率，本文新增了一个时间约束终止条件：着陆器在400个时间步内没有成功着陆，仿真将终止，同时认为任务失败。

这个时间约束可以避免无效的长时间仿真，确保算法训练的时效性、针对性。

3.5 安全性

在LunarLander-v2环境的设置中，如果在连续100次着陆尝试中，平均得分超过200分，则认为模型已训练好，满足着陆器的着陆要求，即人为任务成功。但值得注意的是，满足最大化奖励不一定满足安全性能，着陆器的安全性需要通过最终的终止状态来判断，因此重新定义着陆器的安全性。若 s_T 为着陆器的终止状态，则LunarLander-v2环境中着陆器的安全性可以定义为函数 $\mathrm{Safety}(s_T)$，满足条件

$$\mathrm{Safety}(s_T) = \begin{cases} 1 & |x| \leqslant 0.1 \wedge |y| \leqslant 0.01 \wedge \\ & |\theta'| \leqslant 0.1 \wedge g_{\mathrm{l}} = 1 \wedge g_{\mathrm{r}} = 1 \\ 0 & \text{其他} \end{cases} \tag{6}$$

由于着陆平台的位置始终在(0,0)，设置着陆器的位置、角度满足固定范围，左、右腿均成功接触地面，可以尽可能保证着陆器着陆时的安全性。如果满足条件，则认为着陆器处于安全状态；反之，则认为着陆器处于不良状态，即飞船未安全着陆。

如图3所示为DQN模型的测试阶段，从图中可以看出，即使奖励满足平均得分200分要求时，该回合的着陆也不一定满足安全性要求。

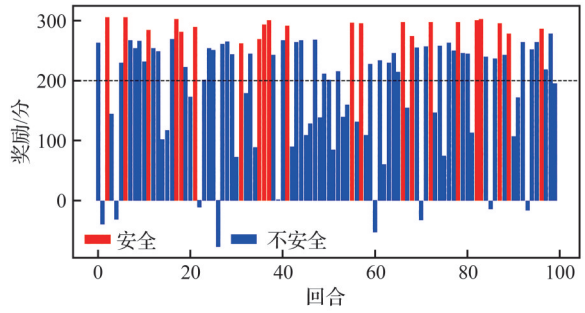


图 3 测试DQN模型的性能

Fig. 3 Testing the performance of the DQN Model

4 概率策略提取

通过对状态-动作压缩来生成抽象SMDP，根据压缩后的SMDP计算概率动作分布，然后构建抽象SMDP状态转移图来提取、表示深度强化学习系统的策略。

首先通过测试训练好的深度强化学习模型，采集M个轨迹：$\mathcal{T}=\{\tau_1,\tau_2,\cdots,\tau_M\}$。对于每个轨迹$\tau$，记录不同时间步$t$的状态$s_t$、动作$a_t$，则轨迹表示为从初始状态到终止状态或特定数量时间步T的状态动作对序列：$\tau=((s_0,a_0),(s_1,a_1),\cdots,(s_{T-1},a_{T-1}))$，可以简化为$\tau=(s_0,a_0,s_1,a_1,\cdots,s_{T-1},a_{T-1})$。对于每个轨迹，$R(\tau)=\{0,1\}$定义了每个回合的任务是否安全，若安全为1，否则为0，$R(\tau)$可以被视为一种简化的奖励函数。

在描述这些轨迹时，通常使用马尔可夫决策过程(MDP)。MDP用于描述系统在状态之间的随机转移过程，包含4个元素：状态空间S、动作空间A、概率转移函数$P(s'|s,a)$、简化的轨迹奖励函数$R(\tau)$。

4.1 状态-动作压缩

受文献[35]的启发，引入状态-动作压缩模块，将原始的MDP映射为一个更为精简的半马尔可夫决策过程(Semi-MDP，SMDP)。状态-动作压缩的核心是找到一个合适的状态度量来衡量轨迹中状态之间的相似性。为了减少状态、动作的数量，同时保持决策过程中的重要决策信息，采用了两阶段压缩策略：首先对每个回合压缩为SMDP，随后将这些SMDP的结果进一步聚类。

在状态-动作压缩过程中，首先定义相似状态，对于相邻状态s_i、s_{i+1}需要同时满足2个条件：

1）s_i、s_{i+1}的下一步动作相同，满足动作一致性，即$a_i=a_{i+1}$。

2）状态s_i、s_{i+1}的连续变量的欧几里得距离小于阈值ϵ，且离散变量相同。

因此，可以得到，若2个状态相似(在轨迹中连续)，则满足公式

$$\begin{cases} a_i=a_{i+1} & \\ \sqrt{\sum_{j=1}^{n}(s_{i,j}-s_{i+1,j})^2}<\epsilon & n\text{为连续变量} \\ s_{i,n}=s_{i+1,n} & n\text{为离散变量} \end{cases} \tag{7}$$

若多个相邻状态$s_1,s_2,\cdots,s_n$满足相似性条件，则通过计算它们的均值来合并这些状态，即对于连续变量，计算$s'=\frac{1}{n}\sum_{i=1}^{n}s_i$，并补充相同的离散变量。由于相邻状态的下一步动作相同，基于动作一致性、状态相似性，通过状态-动作压缩，原始的每个轨迹$\tau=(s_0,a_0,s_1,a_1,\cdots,s_{T-1},a_{T-1})$被简化为$\tau'=(s'_0,a'_0,s'_1,a'_1,\cdots,s'_{T'-1},a'_{T'-1})$，其中简化后的轨迹$\tau'$有$T'$个时间步，并且有$T'<T$。

基于简化后的SMDP，采用K-means聚类算法$f_{k\text{-means}}$将SMDP的所有回合的状态进行聚类。由于前六维数据为连续向量，后两维数据为二值化离散向量，对所有状态进行标准化处理。一方面，将特征表示向量作为对状态的整体表示，满足K-means对输入数据的特征表示要求；另一方面，由于K-means的复杂性低，且能高效地处理大数据集。

根据这个距离，可以将状态空间S分割成K个簇，聚类中心集合为$C=\{C_1,C_2,\cdots,C_K\}$，将每个状态s分配到与其距离$d(s,C_i)$最小的聚类中心的簇中，其中

$$d(s,C_i)=\sqrt{\sum_{j=1}^{n}(s_j-C_{i,j})^2} \tag{8}$$

式中：s_j为状态s的第j个维度表示；$C_{i,j}$为聚类中心C_i的第j个维度表示。

通过这种聚类表示，可以将状态空间S转换为多个抽象概率决策模块C_k[36]。需要强调的是，针对每个抽象决策模块，文献[36]采取了确定性动作，而在本文中使用了概率性动作。在C_k中，每个可能的动作都有一个概率分布，而不是一个确定的动作。状态-动作的两阶段压缩模块的如算法1所示。假设在相似的状态聚类中，状态之间可能表现出相似的动作偏好。将相似状态聚类到一起有助于提取概率策略，可以更好地利用状态之间的相似性来估计动作的转移概率。在假设这些状态在动作偏好方面表现出相似的行为的情况下，这种离散化便于通过分组接近的状态来提取概率策略。

算法1 状态-动作两阶段压缩模块

输入：原始轨迹集合$\mathcal{T}=\{\tau_1,\tau_2,\cdots,\tau_M\}$，阈值$\epsilon$，聚类数量$K$.

初始化：空集合$\mathcal{T}'$

For 每个轨迹τ **do**

  初始化压缩后的轨迹τ'，临时状态列表[s]

  For 轨迹中的每个状态-动作对(s_i,a_i) **do**

    For $i=1:T$

      获取下一个状态-动作对(s_{i+1},a_{i+1})；

        如果满足式(7)，则将s_{i+1}添加到[s]中；

      计算[s]中连续变量的均值，并添加相同离散变量，形成压缩状态s'；

      将压缩状态s'和当前动作添加到压缩后的轨迹τ'中；

      更新当前状态和动作，并重置[s]；

  将压缩后的轨迹τ'添加到集合$\mathcal{T}'$中；

对集合$\mathcal{T}'$的所有状态标准化；

根据K-means计算压缩后的$\mathcal{T}'$的所有状态进行聚类，得到聚类中心；

根据式(8)计算每个状态与聚类中心的距离，将每个状态分配到最近的聚类中心；

输出：新轨迹$\mathcal{T}'=\{\tau_1',\tau_2',\cdots,\tau_M'\}$，抽象概率决策模块聚类中心$C=\{C_1,C_2,\cdots,C_K\}$.

4.2 计算概率动作分布

通过将每个状态聚类到对应的C_k后，需要估计C_k簇中每个动作的选择概率。对于C_k中的每个状态-动作-下一状态三元组(s_t,a_t,s_{t+1})，计算动作a_t的选择概率为

$$P\left(s_{t+1}|s_t,a_t\right)=\frac{\mathrm{Count}\left(s_t,a_t,s_{t+1}\right)}{\sum_{s'\in S}\mathrm{Count}\left(s_t,a_t,s'\right)} \tag{9}$$

然后，估计每个C_k内的动作概率分布。通过聚合C_k中所有状态的转移概率，获得C_k中选择每个动作a的概率

$$P\left(a|C_k\right)=\sum_{s_t\in C_k}P\left(s_{t+1}|s_t,a\right) \tag{10}$$

动作概率估计方法如算法2所示，其中每个C_k内的动作概率表示捕捉了整个状态空间中动作偏好的不确定性。

算法2 动作概率估计

输入：状态-动作压缩模块得到的新轨迹$\mathcal{T}'=\{\tau_1',\tau_2',\cdots,\tau_M'\}$；抽象概率决策模块聚类中心集合$C$.

For 每个C_k **do**

  初始化动作计数矩阵Count←0；

  For 每个轨迹τ_m

    For τ_m中的每个状态-动作-下一个状态三元组(s_t,a_t,s_{t+1})；

      If s_t在C_k中

        增加动作计数：Count(a_t)←Count$(a_t)+1$；

归一化Count以通过式(9)估计动作选择概率；

通过式(10)聚合转移概率估计C_k内的动作选择概率$P(a|C_k)$；

输出：每个C_k内动作的概率分布

4.3 构建抽象SMDP状态转移图

在获得每个概率决策单元中的概率动作分布后，可以根据抽象SMDP、概率动作分布构建抽象SMDP状态转移图，来表示基于DRL的月球着陆器系统的行为。抽象SMDP状态转移图是一种图形化的表示，由1组节点表示状态，通过有向边表示状态之间的转移，并通过边上的标签表示转移的概率。

抽象SMDP状态转移图是有限分支的，目的是模拟抽象状态之间的概率转移。因此，将抽象SMDP状态转移图定义为一个三元组$(C,A,\boldsymbol{P})$。其中，C为抽象状态集合；A为动作的有限集合；$\boldsymbol{P}$为抽象状态-动作转移概率矩阵。状态-动作转移概率矩阵$\boldsymbol{P}$为基于每个C_k的动作选择概率、采集的轨迹中观察到的状态转移构建的。矩阵$\boldsymbol{P}$中每个元素的计算公式为

$$P_{ij}(a)=P(s_{t+1}|s_t,a) \tag{11}$$

式中：$P_{ij}(a)$为状态抽象状态C_i选择动作a转移到C_j的概率，且$s_t\in C_i,s_{t+1}\in C_j$。

5 关键抽象状态-动作对识别与纠正

抽象SMDP状态转移图作为DRL系统行为的抽象表示，展示了不同状态中动作选择的概率本质，用于进一步分析和优化DRL系统性能。通过对抽象SMDP状态转移图的关键抽象状态-动作对的识别，可以增强基于DRL的月球着陆器系统的决策。在深入研究关键状态-动作的识别之前，首先提出以下相关的定义。

定义1 一个关键的抽象状态-动作对(C_k^*, a^*)定义为：C_k^*中动作a^*的选择将对深度强化学习系统的最终结果产生最显著的影响。满足条件

$$(C_k^*, a^*) = \underset{(C_k, a)\in C\times A}{\arg\max}\ P_{\text{fail}}(C_k, a) \tag{12}$$

式中：P_{fail}为在抽象状态C_k下做动作a导致最终结果失败的概率，可以根据转移概率矩阵$\boldsymbol{P}$来确定。由于抽象状态集合C、动作空间A都是有限且非空集集合，则可以得到：存在至少1个关键抽象状态-动作对，导致结果失败概率最大化。

5.1 关键抽象状态-动作对识别

为了提高识别关键抽象状态-动作对的效率，使用反向优先搜索(BFS)方法。一方面，从失败状态往前回溯，只需要考虑实际导致失败的路径，而不是从所有可能的起始抽象状态-动作对开始并尝试所有可能的路径。这种搜索方法可以显著减少需要考虑的路径数量并降低计算量。另一方面，通过从失败状态开始回溯，可以避免在多个路径中重新计算相同的抽象状态-动作对。一旦计算出从某个抽象状态-动作对进入失败状态的概率，就可以将其存储起来并在需要时重用，而不是在每个新路径中重新计算。

首先将失败状态由集合F表示，加入队列Queue中。集合F中的每个抽象状态都被赋予一个高的初始影响得分，表明它们对失败场景的最大贡献。算法从队列Queue中出队一个抽象状态-动作对(C_k, a)，并搜索前序的可以直接达到(C_k, a)的抽象状态-动作对，对于每个前序(C_i, a_i)，计算转移概率$P_{ik}(a_i)$，表示从(C_i, a_i)到(C_k, a)的转换可能性。每个前序的影响得分$I(\cdot)$会更新，来反映其对失败状态的累积贡献。计算公式为

$$I\left[(C_k, a_i)\right] = I\left[(C_k, a_i)\right] + P_{ik}(a_i)\, I\left[(C_k, a)\right] \tag{13}$$

这个迭代过程会完全访问导致失败状态的所有路径。该算法最后将影响分数最高的状态-动作对识别为关键抽象状态-动作对。这种关键的抽象状态-动作对(C_k^*, a^*)被认为是导致失败状态的最关键因素。具体的关键抽象状态-动作对识别算法流程如算法3所示。在确定了关键抽象状态-动作对后，可以在深度强化学习系统内实施监控并纠正机制。

算法3 使用反向BFS识别关键抽象状态-动作对

输入：所有失败状态集合F、动作选择概率$P(a|C_k)$.

初始化：一个空队列Queue，一个空集Visited来追溯访问的抽象状态-动作对(C_k, a)，一个字典I存储每个(C_k, a)对失败的影响.

```
For 每个失败状态 s_fail ∈ F do
    将失败状态 s_fail ∈ F 排队进入 Queue;
    I[s_fail] ← 1   //假设失败状态有最大影响
While Queue不是空 do
    (C_k, a) ← Queue. dequeue();
    If (C_k, a)未被访问过 then
        添加(C_k, a)到Visited;
        For 每个能达到(C_k, a)的前序(C_i, a_i) do
            基于式(11)计算从(C_i, a_i)到(C_k, a)的转移概率;
            基于式(13)更新(C_k, a)对失败的影响;
            将(C_i, a_i)进入队列Queue;
基于式(12)计算对失败状态具有最大影响的抽象状态-动作对;
```

输出：导致失败状态的关键抽象状态-动作对(C_k^*, a^*).

5.2 构造监控器

在识别关键抽象状态-动作对后，通过修改关键(C_k^*, a^*)中的动作来提高任务成功的概率。首先计算C_k^*中每个动作的导致结果成功的概率，将概率最大的动作设置为a_{new}^*，在模型下一次运行的监控器中替换原始动作a^*，计算公式为

$$a_{\text{new}}^* = \underset{a\in A}{\arg\max}\ P_{\text{success}}(a, C_k^*) \tag{14}$$

式中：P_{success}为在C_k^*内执行动作a时，导致结果成功的概率，计算公式为

$$P_{\text{success}}(a, C_k^*) = \sum_{C_j\in C, C_j\neq F} P(C_j|C_k^*, a) \tag{15}$$

最优动作a_{new}^*最大化了在关键C_k^*内的成功

概率。对于每个动作$a\in A$,成功概率是根据该动作下C_k^*的所有预期成功转移的概率之和来计算的。为了提高预训练深度强化学习模型的性能,围绕关键C_k^*、不良动作a^*、新动作a_{new}^*构建一个监控器,当C_k^*出现时监督模型的决策,如果动作与a^*一致,则监控器修改动作为a_{new}^*来进行干预。该监控模块可确保深度强化学习系统在关键抽象状态中不会持续采取导致次优结果的操作,从而提高模型的安全性。

6 实 验

6.1 预训练模型

为了评估提出的关键抽象状态-动作对搜索方法的有效性,在修改后的环境中,对3种不同的深度强化学习算法——经典的DQN,及其变体Dueling DQN、DDQN进行实验研究。实验中,每种算法的模型都经过了充分的训练,具体来说,每个模型都在相同的环境中训练了600个回合。3种模型的超参数相同,其中,学习率l_{r}设置为0.000 75,折扣因子γ设置为0.99,批处理参数为64。

在训练过程中,密切监控了每个深度强化学习模型的奖励收敛情况,如图4所示,3个子图分别为DQN、Dueling DQN、DDQN的奖励变化情况,当模型在训练过程中的平均奖励达到200分时,可以认为着陆器的着陆性能已经满足设计要求。

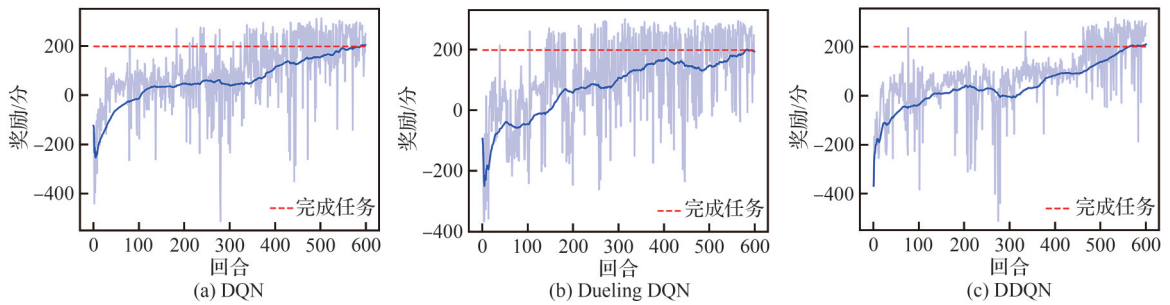


图4 模型训练阶段

Fig. 4 Model training stage

6.2 实验设置

为了验证提出的关键抽象状态-动作对搜索和实时监控方法的有效性,通过实验对比装备监控机制的DRL月球登陆器模型与未装备监控机制的模型。实验中采集的预训练模型测试的轨迹为100回合,每个回合包含400个时间步,每个状态为8维。针对每个回合,采用状态-时间压缩,其中状态-时间压缩中的阈值设置为$\epsilon=0.03$。根据压缩后的状态和动作,聚合所有轨迹的状态进行K-means聚类,在选择聚类数量时,采用了肘部法则,肘部法则通过绘制聚类数量与聚类误差之间的关系图来确定最佳的聚类数量。然而,实验中发现此关系图没有明显的弯曲点,可能是由于数据的分布特征导致的,本文考虑从集合{20, 30, 40, 50}中选择聚类数量。

基于聚类结果,构建抽象SMDP状态转移图。如图5所示为预训练DQN模型轨迹的抽象SMDP状态转移图示例,在该状态转移图中,状态是聚类后形成的状态簇,展示了不同状态簇在执行不同动作时的状态转移概率。

此外,通过反向广度优先搜索算法,可以识别对任务结果具有决定性影响的关键抽象状态-动作对。从导致失败的集合中往前回溯,直到找到导致失败概率最大的关键状态簇与对应的动作,并探索出导致结果成功的概率最大的最优动作,封装为监控器。在监控器中,当前状态到达相应关键状态簇时,如果模型控制输出了导致失败概率最大的关键错误动作,则纠正其为最优动作。这样可以减少不良结果的发生,从而显著提升月球登陆器模型在各种操作条件下的稳定性、安全性。

表1列出了导致失败结果的直接前序状态-动作对,包括各状态-动作对的发生概率。这些数据提供了关于失败前序状态的直接视角。在

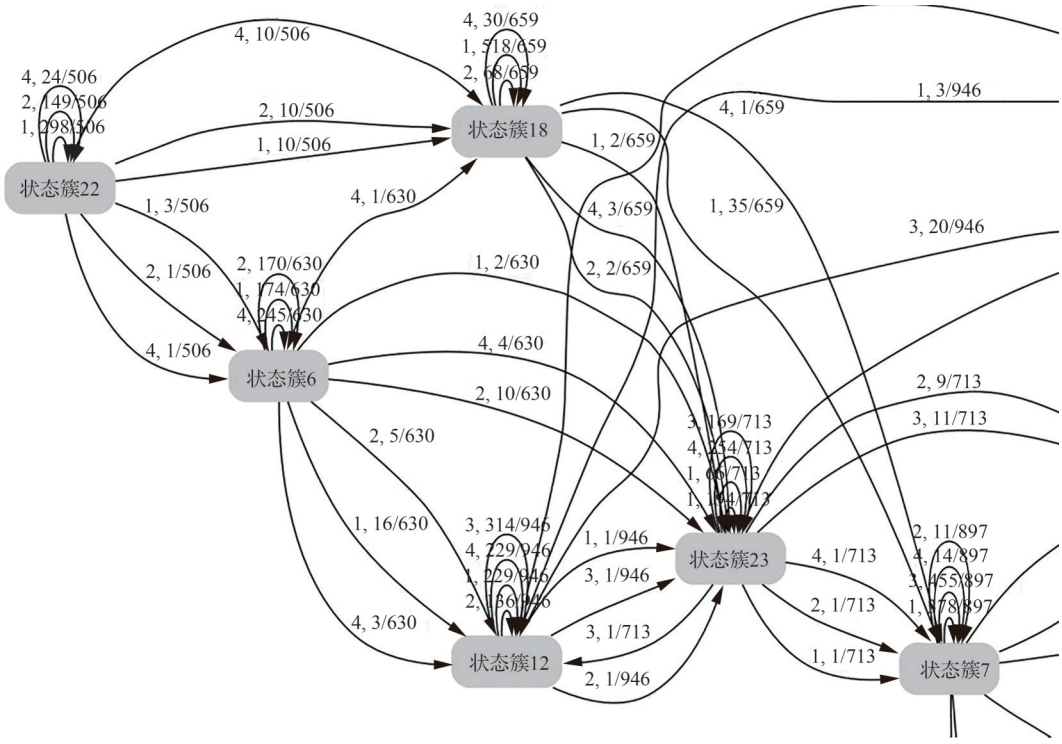


图 5 生成的抽象SMDP状态转移图示例

Fig. 5 Example of generated abstract SMDP state transition diagram

表1中,在监控器中应当尽量避免状态进入0、20、24、25、28号状态簇,若进入这些状态簇,则应尽量避免采用表1中所对应的动作集。若必然要采用这些动作集,则应当采用导致失败概率最小的动作。以表1中经聚类后的24号状态簇为例,其做动作1、2、4都有可能会导致最终结果的失败,其中做动作1时导致失败的概率为4/3 650,做动作2导致失败的概率为8/3 650,做动作4导致失败的概率为2/3 650。若已进入24号状态簇,且必须不选择动作1、2、4中的某一个,则应当选择动作3以尽量降低导致失败的概率。

表 1 导致失败结果的直接前序状态-动作

Table 1 Immediate preceding state-action resulting in failed result

状态簇	动作	概率
0	1	3/1 395
24	4	2/3 650
	1	4/3 650
	2	8/3 650
25	4	2/130
	3	1/130
20	2	2/332
28	1	1/40
	4	1/40

6.3 监控器纠正结果

基于100个回合的轨迹中搜索到的关键抽象状态-动作对,对比分析月球着陆器在未监控与监控状态下的性能差异。为此,进行了额外的300个回合的实验,以评估模型在2种情况下的性能。性能评估主要依据平均奖励、任务成功率2个指标。

性能评估的实验结果如表2所示,虽然预训练Dueling DQN模型在测试中的平均奖励大于200,但其任务完成率却相对较低,这表明在大多数回合中,由于时间步的限制,着陆器未能在规定时间步内成功着陆。在平均奖励方面,监控模型相较于未监控模型展示出了更高的得分。在任务成功率方面,监控模型同样表现出较高的成功率,例如在Dueling DQN模型在监控状态下的任务完成率提升了22%,从67%提高到89%。

表 2 监控状态下与未监控状态下的模型性能比较

Table 2 Comparison of model performance under monitored and unmonitored conditions

预训练模型	奖励		任务成功率/%		
	未监控	监控器纠正	未监控	监控器纠正	成功率提升
DQN	197.07	**235.87**	68	80	**12**
Dueling DQN	223.05	**248.53**	67	89	**22**
DDQN	147.08	**216.85**	66	81	**15**

在本文设置的安全性评价标准下,将未经监控与经监控器纠正后的安全性能进行对比,结果如表3所示。从表中可以看出,所有模型在经过监控器纠正后,安全性都有所提升。例如DQN模型在未监控状态下的安全性为22%,经过监控器纠正后提升到64%,安全性提升了42%。

表 3 着陆器安全性能对比

Table 3 Comparison of lander safety performance

预训练模型	安全性/%		
	未监控	监控器纠正	安全性提升
DQN	22	64	42
Dueling DQN	71	75	4
DDQN	40	49	9

综上所述,监控器纠正策略通过对策略的适当调整,在提升预训练模型累积奖励的同时,也能够降低潜在的风险,增强其安全性,进一步证实了实时监控、纠正关键状态-动作对的框架在提升着陆器安全性方面的有效性。

6.4 鲁棒性分析

鲁棒性是指系统即使面对输入数据的变化和扰动,也能维持其性能的能力,这在动态和不可预测的环境中尤为重要[37]。为了模拟月球着陆器在实际应用中可能遇到的情况,对状态的前6个维度进行了扰动,扰动的标准差分别设为[0.01,0.01,0.05,0.05,0.05,0.05],其中水平位置、垂直位置扰动标准差为0.01,水平速度、垂直速度扰动标准差为0.05,角度扰动标准差为0.05 rad,角速度扰动标准差为0.05 rad/s。通过这样的设定,能够模拟月球着陆器系统在实际应用中可能遇到的扰动。

评估了不同模型在未扰动前、扰动后、监控后的平均奖励对比。如表4所示,所有预训练模型在面对扰动后平均奖励都有所下降,但加入监控机制后,平均奖励普遍提升。

表 4 添加扰动后平均奖励对比

Table 4 Comparison of average reward after adding perturbation

预训练模型	平均奖励对比/分		
	未扰动	扰动后	监控器纠正
DQN	197.07	110.20	133.56
Dueling DQN	223.05	139.18	164.93
DDQN	147.08	134.52	176.51

此外,评估了在本文设置的安全性评价标准下,不同模型在未经监控与经监控器纠正后的安全性能,如表5所示。从表中可以看出,所有模型在遭受扰动后安全性普遍降低,这可能由于扰动改变了环境特性或影响了智能体的决策过程。然而,在监控器的帮助下,各模型的安全性都得到了一定程度的提升。值得注意的是,Dueling DQN在扰动后安全性表现大幅下降,从71%降至0,表明扰动对该模型影响最大,而经过监控器纠正后,其安全性能也得到了一定程度的提升。这些实验结果说明了扰动对月球着陆器性能的影响,还验证了监控机制在提高任务执行过程中的安全性方面的具有一定的鲁棒性。

表 5 添加扰动后安全性能对比

Table 5 Comparison of safety performance after adding perturbation

预训练模型	安全性/%		
	未扰动	扰动后	安全性
DQN	22	17	19
Dueling DQN	71	0	14
DDQN	40	16	28

7 结 论

实时监控和调整关键状态-动作对的方法显著提升了基于DRL的月球登陆器模型的性能,也能广泛应用于其他各种DRL算法,特别是在平均奖励和任务完成率这2个关键指标上,监控模型

均优于未监控模型,在任务完成率上最高可提升22%,鲁棒性实验展示了在具有扰动的环境中,也可以提升其安全性。实验结果展示了该框架在月球着陆等复杂航天任务中的实际应用潜力,可以有效提升操作安全性。

本文所提出的安全性提升框架主要是对已经训练好的DRL策略进行微调,而没有直接干预策略的训练过程。鉴于此,在未来的研究中,利用关键状态-动作对来设计约束条件,来指导策略的迭代优化也是一个重要的研究方向。然而,在DRL训练中,强制在某些状态下执行特定动作可能会限制模型的探索能力,从而潜在地导致性能下降。因此,探索如何在提升安全性与保持策略优化之间取得平衡,是未来工作的关键。

另外,本文方法也具有一定的空间局限性,只适用于离散动作空间,不适用于连续动作空间。在未来的研究中,通过改进现有的安全性提升框架,可以满足更多不同类型任务的需求。

参 考 文 献

[1] SMIRNOV N N. Safety in space[J]. Acta Astronautica, 2023, 204: 679-681.

[2] TIPALDI M, IERVOLINO R, MASSENIO P R. Reinforcement learning in spacecraft control applications: Advances, prospects, and challenges[J]. Annual Reviews in Control, 2022, 54: 1-23.

[3] LORENZ R D. Planetary landings with terrain sensing and hazard avoidance: A review[J]. Advances in Space Research, 2023, 71(1): 1-15.

[4] XIA Y Q, CHEN R F, PU F, et al. Active disturbance rejection control for drag tracking in Mars entry guidance [J]. Advances in Space Research, 2014, 53(5): 853-861.

[5] DAI J, XIA Y Q. Mars atmospheric entry guidance for reference trajectory tracking[J]. Aerospace Science and Technology, 2015, 45: 335-345.

[6] LONG J T, ZHU S Y, CUI P Y, et al. Barrier Lyapunov function based sliding mode control for Mars atmospheric entry trajectory tracking with input saturation constraint[J]. Aerospace Science and Technology, 2020, 106: 106213.

[7] SHEN G H, XIA Y Q, ZHANG J H, et al. Adaptive fixed-time trajectory tracking control for Mars entry vehicle [J]. Nonlinear Dynamics, 2020, 102(4): 2687-2698.

[8] DANG Q Q, GUI H C, LIU K, et al. Relaxed-constraint pinpoint lunar landing using geometric mechanics and model predictive control[J]. Journal of Guidance, Control, and Dynamics, 2020, 43(9): 1617-1630.

[9] 邓云山, 夏元清, 孙中奇, 等. 扰动环境下火星精确着陆自主轨迹规划方法[J]. 航空学报, 2021, 42(11): 524834.
DENG Y S, XIA Y Q, SUN Z Q, et al. Autonomous trajectory planning method for Mars precise landing in disturbed environment[J]. Acta Aeronautica et Astronautica Sinica, 2021, 42(11): 524834 (in Chinese).

[10] KHALID A, JAFFERY M H, JAVED M Y, et al. Performance analysis of Mars-powered descent-based landing in a constrained optimization control framework[J]. Energies, 2021, 14(24): 8493.

[11] YUAN X, ZHU S Y, YU Z S, et al. Hazard avoidance guidance for planetary landing using a dynamic safety margin index[C]//2018 IEEE Aerospace Conference. Piscataway: IEEE Press, 2018: 1-11.

[12] D'AMBROSIO A, CARBONE A, SPILLER D, et al. PSO-based soft lunar landing with hazard avoidance: Analysis and experimentation[J]. Aerospace, 2021, 8(7): 195.

[13] SHAKYA A K, PILLAI G, CHAKRABARTY S. Reinforcement learning algorithms: A brief survey[J]. Expert Systems with Applications, 2023, 231: 120495.

[14] ZHOU Z Y, LIU G J, TANG Y. Multi-agent reinforcement learning: methods, applications, visionary prospects, and challenges[DB/OL]. arXiv preprint: 2305.10091, 2023.

[15] 高锡珍, 汤亮, 黄煌. 深度强化学习技术在地外探测自主操控中的应用与挑战[J]. 航空学报, 2023, 44(6): 026762.
GAO X Z, TANG L, HUANG H. Deep reinforcement learning in autonomous manipulation for celestial bodies exploration: Applications and challenges[J]. Acta Aeronautica et Astronautica Sinica, 2023, 44(6): 026762 (in Chinese).

[16] MOHOLKAR U R, PATIL D D. Comprehensive survey on agent based deep learning techniques for space landing missions[J]. International Journal of Intelligent Systems and Applications in Engineering, 2024, 12(16S): 188-200.

[17] CHENG L, WANG Z B, JIANG F H. Real-time control for fuel-optimal Moon landing based on an interactive deep reinforcement learning algorithm[J]. Astrodynamics, 2019, 3(4): 375-386.

[18] HARRIS A, VALADE T, TEIL T, et al. Generation of spacecraft operations procedures using deep reinforcement learning[J]. Journal of Spacecraft and Rockets, 2022, 59(2): 611-626.

[19] MALI R, KANDE N, MANDWADE S, et al. Lunar

lander using reinforcement learning algorithm[C]//2023 7th International Conference on Computing, Communication, Control and Automation (ICCUBEA). Piscataway: IEEE Press, 2023: 1-5.

[20] DHARRAO D, GITE S, WALAMBE R. Guided cost learning for lunar lander environment using human demonstrated expert trajectories[C]//2023 International Conference on Advances in Intelligent Computing and Applications (AICAPS). Piscataway: IEEE Press, 2023: 1-6.

[21] SHEN D L. Comparison of three deep reinforcement learning algorithms for solving the lunar lander problem [M]//Advances in Intelligent Systems Research. Dordrecht: Atlantis Press International BV, 2024: 187-199.

[22] GU S D, YANG L, DU Y L, et al. A review of safe reinforcement learning: Methods, theory and applications [DB/OL]. arXiv preprint: 2205.10330, 2022.

[23] CHEN W Q, SUBRAMANIAN D, PATERNAIN S. Probabilistic constraint for safety-critical reinforcement learning[J]. IEEE Transactions on Automatic Control, 2024, 69(10): 6789-6804.

[24] SELIM M, ALANWAR A, EL-KHARASHI M W, et al. Safe reinforcement learning using data-driven predictive control[C]//2022 5th International Conference on Communications, Signal Processing, and their Applications (ICCSPA). Piscataway: IEEE Press, 2022: 1-6.

[25] BRUNKE L, GREEFF M, HALL A W, et al. Safe learning in robotics: From learning-based control to safe reinforcement learning[J]. Annual Review of Control, Robotics, and Autonomous Systems, 2022, 5: 411-444.

[26] JIN P, TIAN J X, ZHI D P, et al. Trainify: A CEGAR-driven training and verification framework for safe deep reinforcement learning[C]//International Conference on Computer Aided Verification. Cham: Springer, 2022: 193-218.

[27] ZHI D P, WANG P X, CHEN C, et al. Robustness verification of deep reinforcement learning based control systems using reward martingales[J]. Proceedings of the AAAI Conference on Artificial Intelligence, 2024, 38 (18): 19992-20000.

[28] TAPPLER M, CÓRDOBA F C, AICHERNIG B K, et al. Search-based testing of reinforcement learning[DB/OL]. arXiv preprint: 2205.04887, 2022.

[29] TAPPLER M, PFERSCHER A, AICHERNIG B K, et al. Learning and repair of deep reinforcement learning policies from fuzz-testing data[C]//Proceedings of the IEEE/ACM 46th International Conference on Software Engineering. New York: ACM, 2024: 1-13.

[30] WANG H N, LIU N, ZHANG Y Y, et al. Deep reinforcement learning: A survey[J]. Frontiers of Information Technology & Electronic Engineering, 2020, 21 (12): 1726-1744.

[31] MNIH V, KAVUKCUOGLU K, SILVER D, et al. Human-level control through deep reinforcement learning [J]. Nature, 2015, 518(7540): 529-533.

[32] WANG Z Y, SCHAUL T, HESSEL M, et al. Dueling network architectures for deep reinforcement learning [C]//Proceedings of the 33rd International Conference on International Conference on Machine Learni. New York: ACM, 2016, 48: 1995-2003.

[33] VAN HASSELT H, GUEZ A, SILVER D. Deep reinforcement learning with double Q-learning[J]. Proceedings of the AAAI Conference on Artificial Intelligence, 2016, 30(1): 2094-2100.

[34] BROCKMAN G, CHEUNG V, PETTERSSON L, et al. OpenAI gym[DB/OL]. arXiv preprint: 1606.01540, 2016.

[35] GUO S Q, YAN Q, SU X, et al. State-temporal compression in reinforcement learning with the reward-restricted geodesic metric[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2022, 44(9): 5572-5589.

[36] JIN P, WANG Y, ZHANG M. Efficient LTL model checking of deep reinforcement learning systems using policy extraction[C]//The 34th International Conference on Software Engineering and Knowledge Engineering. San Francisco: KSI Research Inc., 2022: 357-362.

[37] KORKMAZ E. Adversarial robust deep reinforcement learning requires redefining robustness[J]. Proceedings of the AAAI Conference on Artificial Intelligence, 2023, 37(7): 8369-8377.

（责任编辑：李丹）

Control of lunar landers based on secure reinforcement learning

YANG Min, LIU Guanjun*, ZHOU Ziyuan

Department of Computer Science and Technology, Tongji University, Shanghai 201804, China

Abstract: In lunar landing missions, the lander must perform precise operations in extreme environments and often faces the challenge of communication delays. These factors severely limit the real-time operation capabilities of ground control. In response to these challenges, this study proposes a Deep Reinforcement Learning (DRL) framework for safety enhancement based on the Semi-Markov Decision Process (SMDP) to improve the operational safety of autonomous spacecraft landing. To compress the state space and maintain the key characteristics of the decision-making process, this framework compresses the Markov Decision Process (MDP) of the historical trajectory into a SMDP, and constructs an abstract SMDP state transition diagram based on the compressed trajectory. Then, the key state-action pairs of potential risks are identified, and the real-time monitoring and intervention strategy is implemented. The framework effectively improves the safety of the spacecraft's autonomous landing. Furthermore, the reverse breadth first search method is used to search for the state-action pairs that have decisive impact on task results, and real-time adjustment of the model is realized through the built state-action monitor. Experimental results show that this framework increases the mission success rate of the lunar lander by up to 22% in a simulated environment on the pre-trained Deep Q-Network (DQN), Dueling DQN, and DDQN models without adding additional sensors or significantly changing the existing system configuration. According to the preset safety evaluation standards, the framework can improve safety by up to 42%. In addition, simulation results in a virtual environment demonstrate the practical application potential of this framework in complex space missions such as lunar landing, which can effectively improve operational safety and efficiency.

Keywords: deep reinforcement learning; autonomous landing; abstract SMDP state transition diagram; safety enhancement; real-time monitoring; reverse breadth-first search

Received: 2024-04-19; **Revised**: 2024-05-07; **Accepted**: 2024-07-24; **Published online**: 2024-08-21 09:42
URL: https://hkxb.buaa.edu.cn/CN/Y2025/V46/I3/630553
Foundation items: National Natural Science Foundation of China (62172299, 62032019); Space Optoelectronic Measurement and Perception Lab., Beijing Institute of Control Engineering (LabSOMP-2023-03); The Fundamental Research Funds for the Central Universities (2023-4-YB-05); Shanghai Technological Innovation Action Plan (22511105500)
* **Corresponding author.** E-mail: liuguanjun@tongji.edu.cn