计算机科学
COMPUTER SCIENCE

基于安全强化学习的航天器交会制导方法

幸林泉，肖应民，杨志斌，韦正旻，周勇，高赛军

引用本文

幸林泉，肖应民，杨志斌，韦正旻，周勇，高赛军基于安全强化学习的航天器交会制导方法[J]．计算机科学，2023，50(8)：271-279.

XING Linquan，XIAO Yingmin，YANG Zhibin，WEI Zhengmin，ZHOU Yong，GAO Saijun. Spacecraft Rendezvous Guidance Method Based on Safe Reinforcement Learning [J]. Computer Science，2023，50(8)：271-279.

计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.220700210

基于安全强化学习的航天器交会制导方法

幸林泉[1,2]　肖应民[1,2]　杨志斌[1,2]　韦正旻[1,2]　周　勇[1,2]　高赛军[3]
1 南京航空航天大学计算机科学与技术学院　南京 211106
2 高安全系统的软件开发与验证技术工信部重点实验室　南京 211106
3 上海航天电子技术研究所　上海 201109
(xinglq@nuaa.edu.cn)

摘　要　随着航天器交会对接任务越来越复杂,对其高效性、自主性和安全性的要求急剧增加。近年来,引入强化学习技术来解决航天器交会制导问题已经成为国际前沿热点。障碍物避撞对于确保航天器安全交会对接至关重要,而一般的强化学习算法没有对探索空间进行安全限制,这使得航天器交会制导策略设计面临挑战。为此,提出了基于安全强化学习的航天器交会制导方法。首先,设计避撞场景下航天器自主交会的马尔可夫模型,提出基于障碍预警与避撞约束的奖励机制,从而建立用于求解航天器交会制导策略的安全强化学习框架;其次,在该安全强化学习框架下,基于近端策略优化算法(PPO)和深度确定性策略梯度算法(DDPG)这两种深度强化学习算法生成了制导策略。实验结果表明,该方法能有效地进行障碍物避撞并以较高的精度完成交会。另外,通过分析两种算法的性能优劣和泛化能力,进一步证明了所提方法的有效性。

关键词:航天器交会制导;障碍物避撞;安全强化学习;近端策略优化;深度确定性策略梯度
中图法分类号　TP311

Spacecraft Rendezvous Guidance Method Based on Safe Reinforcement Learning

XING Linquan[1,2],XIAO Yingmin[1,2],YANG Zhibin[1,2],WEI Zhengmin[1,2],ZHOU Yong[1,2] and GAO Saijun[3]
1 School of Computer Science and Technology,Nanjing University of Aeronautics and Astronautics,Nanjing 211106,China
2 Key Laboratory of Safety-critical Software,Ministry of Industry and Information Technology,Nanjing 211106,China
3 Shanghai Aerospace Electronic Technology Institute,Shanghai 201109,China

Abstract　With the increasing complexity of spacecraft rendezvous and docking tasks,the requirements for its efficiency,autonomy and reliability are highly demanded. In recent years,the introduction of reinforcement learning technology to solve the problem of spacecraft rendezvous and guidance has become an international frontier hotspot. Obstacle avoidance is critical for safe spacecraft rendezvous,and the general reinforcement learning algorithm does not impose safety restrictions on space exploration,which make the design of spacecraft rendezvous guidance policy challenging. This paper proposes a spacecraft rendezvous guidance method based on safe reinforcement learning. First,a Markov model of autonomous spacecraft rendezvous in collision avoidance scenarios is designed,a reward mechanism based on obstacle warning and collision avoidance restraint is proposed,and thus a safe reinforcement learning framework for solving spacecraft rendezvous guidance strategy is established. Second,with the framework of safe reinforcement learning,guidance policies are generated based on two deep reinforcement learning algorithms,proximal policy optimization(PPO) and deep deterministic policy gradient(DDPG). Experimental results show that the method can effectively avoid obstacle and complete the rendezvous with high accuracy. In addition,the performance and generalization ability of the two algorithms are analyzed,which proves the effectiveness of the proposed method.

Keywords　Spacecraft rendezvous guidance,Obstacle avoidance,Safe reinforcement learning,Proximal policy optimization,Deep deterministic policy gradient

---

到稿日期:2022-07-24　返修日期:2022-11-04
基金项目:国家自然科学基金(62072233);国防基础科学研究计划(JCKY2020205C006);南京航空航天大学科研与实践创新计划(xcxjh20211604)
This work was supported by the National Natural Science Foundation of China(62072233),National Defense Basic Scientific Research Program of China(JCKY2020205C006) and Postgraduate Research & Practice Innovation Program of NUAA(xcxjh20211604).
通信作者:杨志斌(yangzhibin168@163.com)

1 引言

航天器自主交会(Autonomous Spacecraft Rendezvous)是一个追踪航天器自动接近目标航天器的过程,对航天器的最终位置、最终速度和最终姿态有严格的限制,是完成对接、维修、大范围结构集装、编队飞行等诸多任务的前提条件。随着航天任务的复杂性增加,高计算效率的制导策略生成面临挑战。传统的解决方案主要依赖于动态模型的最优化控制,需要大量的计算资源[1-2]。同时,随着太空探索的愈发频繁,航天器面临的太空障碍物也在不断增加,例如运动的其他飞行器和相对静止的物理机构或者空间碎片。航天器不与其他空间目标相撞是轨道飞行的迫切要求[3-4],因此,障碍物避撞对于确保航天器安全运行至关重要。使用传统控制方法难以开发出针对该类问题的高计算效率的解决方案[5],近年来,引入强化学习技术来解决航天器交会制导问题已经成为国际前沿热点。

强化学习(Reinforcement Learning,RL)作为一种最优化控制方法,在诸多领域的应用越来越广泛,如自动驾驶[6]、移动边缘计算[7]、推荐系统[8]以及工业物联网[9]等。在复杂制导任务的驱动下,一些研究将强化学习方法引入航天器自主交会领域,以增强航天器的制导能力。

目前已有一些在航天器交会制导领域利用强化学习取得的进展。Wang 等提出了一种基于深度确定性策略梯度(Deep Deterministic Policy Gradient,DDPG)算法的小推力自主交会制导方法[10]。Hovell 等提出了用于航天器制导的深度制导方法,并采用了分布式深度确定性策略梯度(Distributed Distributional Deep Deterministic Policy Gradient,D4PG)算法[11]。文献[12]研究了线性多脉冲交会任务中的行为克隆和强化学习问题。

这些已有研究较少考虑航天器交会过程的障碍物避撞问题。为了解决诸如障碍物避撞这类安全问题,学术界提出安全强化学习(Safe Reinforcement Learning)的思想。安全强化学习的目标是在满足一定安全约束的条件下最大化累计回报。Garcia 等[13]将安全强化学习分为两类,一类是修改最优准则(如最坏情况准则、风险敏感准则、约束准则等)的方法,通过调整累计回报的最大化目标来减少风险;另一类是约束探索空间的方法,使用先验知识或者基于风险度量的方法对智能体的动作进行约束,例如文献[6]使用基于动作约束的深度强化学习方法实现安全自动驾驶。在航天器交会制导领域,Broida 等研究了 RL 在卫星交会任务闭环控制中的应用,使用了近端策略优化(Proximal Policy Optimization,PPO)算法[5],设置了以目标航天器为中心的保护区,最终能使追踪航天器与目标航天器不发生碰撞,但该研究没有考虑交会过程中的空间碎片等障碍物避撞问题。

本文在研究航天器自主交会制导的前期工作中,使用基于模型的方法,以实现强化学习的可预测性[14]。本文进一步考虑了障碍物避撞问题。障碍物可以分为动态障碍物与静态障碍物,我们主要研究静态障碍物的规避,提出基于安全强化学习的航天器交会制导方法,主要贡献如下:

首先,设计避撞场景下航天器自主交会的马尔可夫模型,并提出基于障碍预警与避撞约束的奖励机制,即在交会过程中设置静态警戒区,航天器进入警戒区则表示航天器处于危险状态,此时给予航天器负奖励以对动作进行惩罚。

其次,在上述马尔可夫模型与奖励机制组成的安全强化学习框架下,基于 PPO 和 DDPG 两种深度强化学习算法生成制导策略。实验结果表明,该方法能有效地进行障碍物避撞并以较高的精度完成交会。

最后,通过分析两种算法的性能优劣和泛化能力,进一步证明本文方法的有效性。

2 背景知识

2.1 航天器交会对接

航天器交会对接也称空间交会对接,是两个航天器在轨道上按预定的位置、速度和时间会合(交会),然后经姿态对准、靠拢直至在结构上连接成一体(对接)的全部飞行动作过程[15]。进行空间交会对接的两个航天器,通常一个被称为追踪航天器,另一个被称为目标航天器。在交会对接过程中,通常追踪航天器是主动的,一般通过改变追踪航天器相对于目标航天器的位置和姿态分阶段实现两个航天器的交会对接。对于一项完整的交会对接任务,追踪航天器的飞行阶段通常包含以下阶段:待发段、发射段、远距离导引段、近距离自主控制段、对接段、组合体运行段、撤离段、返回再入段等。本文重点考虑近距离自主控制段,近距离自主控制段需要充分考虑两个航天器间的相对运动特性,所采用的制导、导航和控制策略有别于其他阶段,最能体现交会对接技术的特点,并且整个交会对接任务对该阶段控制系统的性能要求最多也最为严格,所以近距离自主控制段一直是交会对接制导、导航与控制理论方法研究的热点和工程设计的重点。

本文所考虑的近距离自主控制段场景如图 1 所示。目标航天器在一个以地球为圆心的圆形轨道上运行,两个航天器之间存在障碍物,追踪航天器需要在不与空间中障碍物发生碰撞的前提下与目标航天器完成交会,如图 1 虚线所示。

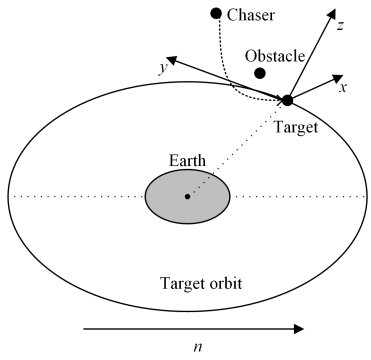


图 1 航天器自主交会

Fig. 1 Autonomous spacecraft rendezvous

目标航天器的质心被定义为原点,x 轴是沿地球中心到目标航天器质量中心的方向,y 轴是沿目标航天器轨道的切线方向,z 轴与它们形成右手坐标系。坐标系以地球质心为圆心,R 为轨道半径,n 为旋转的轨道角速度。根据牛顿运动理论,可以得到以下非线性相对运动模型[16]:

$$\begin{bmatrix}\ddot{x}\\ \ddot{y}\\ \ddot{z}\end{bmatrix}=\begin{bmatrix}2n\dot{y}+n^2x-\dfrac{\mu(R+x)}{[(R+x)^2+y^2+z^2]^{\frac{3}{2}}}+\dfrac{\mu}{R^2}\\ n^2y-2n\dot{x}-\dfrac{\mu y}{[(R+x)^2+y^2+z^2]^{\frac{3}{2}}}\\ -\dfrac{\mu z}{[(R+x)^2+y^2+z^2]^{\frac{3}{2}}}\end{bmatrix}+a_f \tag{1}$$

其中，μ 是引力常量，由于两个航天器之间的距离远小于目标航天器与地球中心之间的距离，因此可以对方程(1)进行线性化，以获得近似的线性相对运动模型，称为 C-W 方程：

$$\begin{bmatrix}\ddot{x}\\ \ddot{y}\\ \ddot{z}\end{bmatrix}=\begin{bmatrix}3n^2x+2n\dot{y}\\ -2n\dot{x}\\ -n^2z\end{bmatrix}+a_f \tag{2}$$

推力加速度 a_f 由沿 3 个轴的 3 个加速度值组成：

$$a_f=[\boldsymbol{u}_x,\boldsymbol{u}_y,\boldsymbol{u}_z]^{\mathrm{T}} \tag{3}$$

其中，$\boldsymbol{u}_x,\boldsymbol{u}_y,\boldsymbol{u}_z$ 是追踪航天器分别在 3 个轴方向上的控制力输入。那么方程(3)就可以转换成如下方程组：

$$\begin{cases}\ddot{x}=3n^2x+2n\dot{y}+\boldsymbol{u}_x\\ \ddot{y}=-2n\dot{x}+\boldsymbol{u}_y\\ \ddot{z}=-n^2z+\boldsymbol{u}_z\end{cases} \tag{4}$$

由式(4)可知，在相对运动中，z 方向的运动是独立的，z 方向上控制器可以单独考虑，因此本文研究基于 x-y 二维平面的航天器交会问题，最终相对运动方程如式(5)所示：

$$\begin{cases}\ddot{x}=3n^2x+2n\dot{y}+\boldsymbol{u}_x\\ \ddot{y}=-2n\dot{x}+\boldsymbol{u}_y\end{cases} \tag{5}$$

2.2 强化学习

强化学习是智能体以试错的方式进行学习，通过与环境进行交互获得的奖励来引导行为，目标是使智能体获取最大的奖励。由于外部环境提供的信息很少，强化学习系统必须靠自身的经验进行学习。通过这种方式，强化学习在行动-评估的环境中获得知识，并改善行动方案以适应环境。强化学习的基本原理是：如果智能体的某个行为策略在环境中产生积极的奖励，那么智能体在未来产生这个行为策略的趋势便会加强。智能体的目标是在每个状态中发现最优策略，以最大化期望的折扣奖励之和。

强化学习与其他类型的机器学习的主要区别在于，其学习过程涉及智能体与其环境之间的交互。智能体与环境之间的交互被建模为马尔可夫决策过程(Markov Decision Process，MDP)，它是一个五元组$\langle S,A,R(s,a),T(s'|s,a),\gamma\rangle$。其中 S 是状态集；A 是动作集；$R(s,a)$是奖励函数，代表智能体在状态 s 采取动作 a 所获得的奖励；$T(s'|s,a)$表示智能体在状态 s 采取动作 a 而到达 s'的概率；而 γ 则是优先考虑短期奖励的折扣因子，通常情况下 $\gamma\in[0,1]$。MDP 的解是一种策略 $\pi(s)$，它将状态映射到动作，以最大限度地提高随时间积累的奖励。

深度强化学习(Deep Reinforcement Learning，DRL)将强化学习和深度学习的优点相结合，是目前强化学习的研究热点。DRL 的框架如图 2 所示，智能体根据当前环境输出策略，策略是由深度神经网络产生的，网络输入当前状态 s，输出动作 a，环境接收到动作后根据奖励函数返回当前的奖励 r，同时根据状态转移规则更新至下一个状态。在下一个周期中，智能体根据新的状态 s'输出新的动作 a'，该动作再次与环境交互。智能体不断地与环境交互并生成大量数据，DRL 算法利用这些数据来修改自己的动作策略，然后与环境再次交互生成新的数据，接着使用新的数据进一步提高算法的性能。经过多次迭代学习，智能体最终可以学习到完成任务的最优策略。

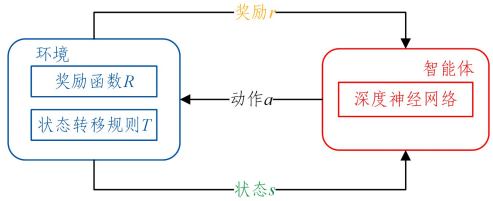


图 2 深度强化学习框架

Fig. 2 Deep reinforcement learning framework

目前流行的 DRL 算法很多，本文使用的是 PPO 算法[17]和 DDPG 算法[18]。PPO 算法是一种在线策略(On-Policy)方法，它是信任域策略优化(Trust Region Policy Optimization，TRPO)算法[19]的改进版。相较于 TRPO 算法，PPO 更简单，实现更容易，并且性能表现更好。PPO 的训练在与环境交互采样数据以及利用随机梯度上升优化一个替代目标函数之间交替进行，相较于标准梯度策略算法，每次数据采样只能进行一次梯度更新，PPO 算法所使用的目标函数能够利用同一批次数据进行多次梯度更新。DDPG 算法是一种离线策略(Off-Policy)方法，它是在 DQN 算法[20]的基础之上进行改进的，解决了原始 DQN 算法不能处理连续动作以及高维状态空间的问题，可以根据状态信息得到确定性连续动作输出。

3 实现方法

本文提出的航天器自主交会制导的安全强化学习框架如图 3 所示。智能体根据当前状态通过推力发动机输出动作策略 a，然后根据状态转移方程计算出下一个状态值 s。该状态值一方面会输入给智能体，另一方面也会被用来计算奖励值，计算奖励值时，需考虑是否满足安全约束，通过本文提出的基于障碍预警与避撞约束的奖励机制来计算出最终的奖励值 r。最后在训练过程中，还需要判断当前训练回合是否满足中止条件。

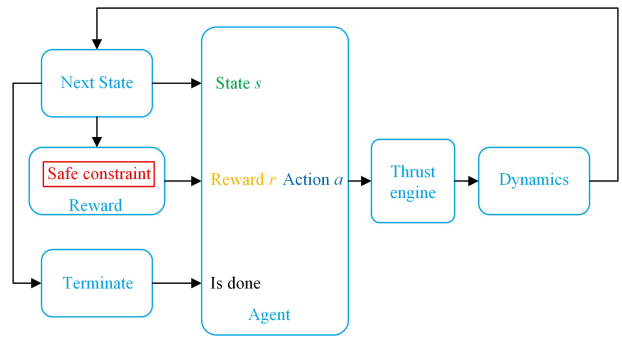


图 3 航天器自主交会制导的安全强化学习框架

Fig. 3 Safety reinforcement learning framework for spacecraft autonomous rendezvous

在上述框架下，需要针对性地进行环境设置与算法实现。接下来首先介绍航天器自主交会环境设置，即 MDP 环境建模，包括场景假设和 MDP 的各个要素的设计，其次分别介绍基于 PPO 算法和基于 DDPG 算法的航天器交会制导策略生成方法实现。

3.1 MDP 环境建模

(1)场景假设

如 2.1 节所述，航天器交会过程中，z 方向的运动是独立的，因此 z 方向上的控制器可以单独考虑。本文研究基于 x-y 二维平面的航天器交会问题，如图 4 所示。目标航天器位于原点，其轨道角速度 $n=0.0011068\ \mathrm{rad/s}$，虚线框为追踪航天器的出发区域，它是一个以坐标(450 m，450 m)为中心、边长为 100 m 的正方形区域。航天器交会的目标是追踪航天器从出发区域以相对静止的状态出发，通过观察追踪航天器的状态信息，控制发动机推力，使得追踪航天器尽可能地靠近目标航天器。同时，追踪器与目标器之间存在一个包含障碍物的禁区，它是一个以坐标(110 m，110 m)为中心、边长为 20 m 的正方形区域，航天器在交会过程中，应该避免进入该区域。

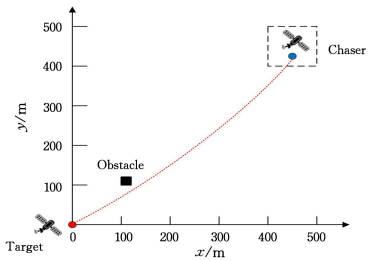


图 4 二维平面的航天器交会场景

Fig. 4 Spacecraft rendezvous scene in 2D plane

(2)状态空间

状态变量的选取应保证能够反映交会效果，从而确保航天器能根据观测的状态输出正确的控制指令。C-W 方程中的相对位置和速度是反映航天器接近情况的关键指标，因此选取的状态变量为 $s=[x,y,\dot{x},\dot{y}]$，分别为追踪航天器相对于目标航天器的沿 x 轴和 y 轴的位置分量和速度分量。这样的状态设置能保证追踪航天器可以获取到足够的关于交会过程中的状态信息，保证交会控制的性能。在算法训练过程中，应对追踪航天器的位置进行范围限定，避免追踪航天器与目标航天器相距过远而进行无意义的训练。综上，根据上文的场景假设可设置状态变量的空间范围，如表 1 所列。

表 1 状态变量

Table 1 State variables

状态	范围	初始范围
x	$[-200\ \mathrm{m},600\ \mathrm{m}]$	$[400\ \mathrm{m},500\ \mathrm{m}]$
y	$[-200\ \mathrm{m},600\ \mathrm{m}]$	$[400\ \mathrm{m},500\ \mathrm{m}]$
$\dot{x}$	$[-\infty,+\infty]$	$[0\ \mathrm{m/s},0\ \mathrm{m/s}]$
$\dot{y}$	$[-\infty,+\infty]$	$[0\ \mathrm{m/s},0\ \mathrm{m/s}]$

(3)动作空间

动作空间由二维向量组成，分别表示 x 轴方向和 y 轴方向的追踪航天器发动机推力 $\boldsymbol{u}_x$ 和 $\boldsymbol{u}_y$。根据实际硬件情况，它们应该为连续值，并且应该满足限幅条件，范围设为[−1，1]。此处假设追踪航天器加速度在数值上等于发动机推力大小。

(4)状态转移方程

状态转移方程基于 C-W 方程和欧拉方法获得，根据方程(5)，可以计算当前推力下追踪航天器在 x 轴和 y 轴方向上分别获得的加速度，下一个状态的速度和位置可以进行如下更新(此处设置的时间步长 $\Delta t=1\ \mathrm{s}$)：

$$\begin{bmatrix} x' \\ y' \\ \dot{x}' \\ \dot{y}' \end{bmatrix} \leftarrow \begin{bmatrix} x+\dot{x}\Delta t \\ y+\dot{y}\Delta t \\ \dot{x}+\ddot{x}\Delta t \\ \dot{y}+\ddot{y}\Delta t \end{bmatrix} \tag{6}$$

(5)奖励函数

在强化学习中，奖励是智能体判断动作好坏的主要信息来源，故奖励函数的设计至关重要。本文提出了基于碰撞规避的奖励机制，考虑了多方面的因素，具体设计如下。

首先，根据终止条件设计奖励。定义 $error=\sqrt{x^2+y^2+\dot{x}^2+\dot{y}^2}$，即状态向量的 L2 范数来描述与目标状态的误差，当 $error\leqslant 0.5$ 时认为两个航天器完成交会，任务成功，将获得一个较大的正奖励。而航天器在与环境交互过程中发生超时、与障碍物碰撞、超出表 1 中设置的范围等情况时宣告任务失败，将会得到一个较大的负奖励。这些奖励是在追踪航天器到达一些关键状态时得到的，故属于稀疏奖励，它的构成如表 2 所列。其中，$steps$ 表示当前的累计时间步长，而 max_steps 表示一个回合的最大时间步长，此处设为 400 s。

表 2 稀疏奖励

Table 2 Sparse rewards

描述	奖励
成功交会($error\leqslant 0.5$)	$+10*(3-steps/max\_steps)$
与障碍物碰撞	−100
越界	−100
超时($steps>max\_steps$)	−100

其次，在交会过程中，状态值包括追踪航天器的相对位置和速度，根据表 1 可以发现追踪航天器的状态空间很大，如果仅在到达终止条件时给奖励，那么奖励将过于稀疏，不利于完成任务，因此还需要设计密集奖励。密集奖励是追踪航天器每一时间步长都可以得到的奖励，有利于追踪航天器完成任务。具体奖励函数如表 3 所列，主要分为两部分：基于 $error$ 的奖励和基于障碍预警与避撞约束的奖励。描述如下：

基于 $error$ 的奖励：在每一步中，都会根据当前的 $error$ 值给出相应奖励，$error$ 越小，说明追踪航天器越靠近目标航天器，则奖励越大。这样设计的目的是在使得追踪航天器在每一步中都能获得一个基于当前状态的奖励，从而避免奖励过于稀疏导致的训练困难。特别地，当 $error\leqslant 1.0$ 时，还会有额外的奖励，以进一步促进追踪航天器靠近目标航天器。

基于障碍预警与避撞约束的奖励：如果只在发生碰撞的情况下才给智能体较大的负奖励，即使最终模型收敛，追踪航天器仍然可能与障碍物发生碰撞，因为追踪航天器无法利用少量的碰撞数据进行有效训练。为了在交会过程中提前

预警，设置了警戒区，即在禁区各边向外延展 20 m 范围内的区域。特别地，如果追踪航天器进入了警戒区，不会终止当前回合，而是实现了一个软约束，追踪航天器在进入警戒区的每个时间步都会获得一个较大的负奖励。该软约束的目的是在帮助追踪航天器远离禁区的同时允许它在训练探索期间进入警戒区收集有用的数据。这种方法使得策略更新更加顺畅，并能够加快学习速度，改进最终的策略。

表 3 密集奖励

Table 3 Dense rewards

描述	奖励
基于当前状态的奖励	$-0.001*(error)$
如果 $error\leqslant 1.0$	$+1$
如果进入警戒区	-10

3.2 基于 PPO 的交会制导策略生成方法实现

PPO 算法是一种基于策略梯度的强化学习算法，它直接学习一个策略，即 $\pi_\theta(a|s)$，其中 $\boldsymbol{\theta}$ 表示该策略的参数向量。PPO 算法是一种基于 Actor-Critic 的方法，即有两个网络，一个网络用来参数化智能体的策略，称作 Actor；另一个网络用来参数化当前策略的价值函数，称作 Critic。根据当前的状态信息，Critic 估计预期的奖励，Actor 则估计策略，如果得到的奖励比 Critic 估计的好，那么 Actor 会增加选择该动作的概率，反之则会降低概率。PPO 算法通过设置单次更新距离极限来稳定策略，从而改进了 Actor-Critic 方法。

通过引入裁剪函数，PPO 算法优化了如下目标函数：

$$L^{\mathrm{CLIP}}(\boldsymbol{\theta})=\hat{E}_t[\min(R_t(\boldsymbol{\theta})\hat{A}_t, clip(R_t(\boldsymbol{\theta}),1-\varepsilon,1+\varepsilon)\hat{A}_t)] \tag{7}$$

其中比例 $R_t(\boldsymbol{\theta})$ 定义如下：

$$R_t(\boldsymbol{\theta})=\frac{\pi_{\boldsymbol{\theta}}(a_t|s_t)}{\pi_{\boldsymbol{\theta}_{\mathrm{old}}}(a_t|s_t)} \tag{8}$$

它描述了当前策略分布 $\pi_{\boldsymbol{\theta}}$ 与前一次策略分布 $\pi_{\boldsymbol{\theta}_{\mathrm{old}}}$ 的差距。ε 是一个超参数，通常设置为 0.2。$\hat{A}_t$ 是优势函数，有多种定义方式。本文使用了广义优势估计(Generalized Advantage Estimator，GAE)[21]，定义如下：

$$\hat{A}_t^{\mathrm{GAE}(\gamma,\lambda)}=\sum_{l=0}^{\infty}(\gamma\lambda)^l\delta_{t+1}^V \tag{9}$$

其中，$\delta_t^V=r_t+\gamma V(s_{t+1})-V(s_t)$，$V(s_t)$是由 Critic 估计出的状态 s 的价值，λ 是一个超参数。GAE 可以有效减小梯度估计的方差，同时有利于面对奖励稀疏时的智能体的训练，该方法一经提出之后就被广泛地应用在各种 DRL 算法实现中。

基于 PPO 的交会制导策略生成算法实现如算法 1 所示。

算法 1 基于 PPO 的交会制导策略生成算法

1. 初始化策略参数 θ_0 和价值函数参数 ϕ_0
2. for k=0,1,2,3…：
3. 追踪航天器在第 3.1 节设定的交会环境中运行策略 $\pi_k=\pi(\theta_k)$，根据状态转移方程收集一系列的经验数据 $\mathcal{D}_k=\{\tau_i\}$
4. 基于提出的障碍预警与避撞约束的奖励机制计算累计折扣奖励 $\hat{R}_t$
5. 根据式(9)计算优势函数估计值 $\hat{A}_t$
6. 用随机梯度上升最大化 PPO-clip 目标函数更新策略：

$$\theta_{k+1}=\mathrm{argmax}\frac{1}{|\mathcal{D}_k|T}\sum_{\tau\in\mathcal{D}_k}\sum_{t=0}^{T}\min(R_t(\theta)A^{\pi_{\theta_k}}, \mathrm{clip}(R_t(\theta),1-\varepsilon,1+\varepsilon)A^{\pi_{\theta_k}})$$

7. 通过随机梯度下降最小化均方误差更新价值函数：

$$\phi_{k+1}=\arg\min\frac{1}{|\mathcal{D}_k|T}\sum_{\tau\in\mathcal{D}_k}\sum_{t=0}^{T}(V_\phi(s_t)-\hat{R}_t)^2$$

8. end for

PPO 算法的具体神经网络结构以及参数分别如表 4 和表 5 所列。Actor 网络的输入层是由追踪航天器的四维状态空间 S 组成，Actor 网络输出了两组相互独立的高斯分布的均值与方差，分别代表两个方向的动作值分布，中间隐藏层节点数是 256，使用的激活函数是 ReLU。在训练过程中，将当前状态 s 输入到 Actor 网络中，Actor 网络输出两个方向的动作值分布，再根据这两个分布进行采样得到具体的动作，并分别用 tanh 激活函数将其限制在[−1，1]之间从而得到 $\boldsymbol{u}_x$ 和 $\boldsymbol{u}_y$，追踪航天器再与环境交互得到新的状态，具体过程是：根据动作 $\boldsymbol{u}_x$，$\boldsymbol{u}_y$ 和 C-W 方程，即式(5)计算出前推力下追踪航天器在 x 轴和 y 轴方向上分别获得的加速度 $\ddot{x}$ 和 $\ddot{y}$，然后根据状态转移方程，即式(6)，计算出下一个状态 $s'=[x',y',\dot{x}',\dot{y}']$。

Critic 网络的输入层和中间隐藏层与 Actor 一致，Critic 网络输出层是一维标量，用于评估当前状态的价值，即 $V(s)$。一方面，Critic 网络通过训练优化自身的参数，对状态的价值做出更准确的预测，另一方面，它引导着 Actor 网络参数的更新方向。

神经网络训练优化器采用了 Adam 优化器，这是深度学习领域使用最广泛的优化器。同时基于 3.1 节中的奖励函数设计，设置折扣因子=0.99，PPO 算法总共与环境交互了 3×10^6 次。

表 4 PPO 算法的神经网络结构

Table 4 Neural network structure of PPO algorithm

层	Actor 网络		Critic 网络	
	节点数	激活函数	节点数	激活函数
输入层	4	ReLU	4	ReLU
隐藏层 1	256	ReLU	256	ReLU
隐藏层 2	256	ReLU	256	ReLU
输出层	2	Tanh	1	Linear

表 5 PPO 算法的参数

Table 5 Parameters of PPO algorithm

参数	值
折扣因子 γ	0.99
Actor 学习率	0.000 03
Critic 学习率	0.000 03
优化器	Adam
训练轮次	3 000
Clip 函数参数 ε	0.2
GAE 超参数 λ	0.98
每轮步数 *steps_per_epoch*	1 000
总步数 *total_steps*	3 000 000

3.3 基于 DDPG 的交会制导策略生成方法实现

DDPG 算法同样也采用 Actor-Critic 架构，网络部分由 Actor 网络 $\mu(s|\theta^\mu)$、Critic 网络 $Q(s,a|\theta^Q)$以及与 Actor 网络相对应的演员家目标网络(Target-Actor)$\mu(s|\theta^{\mu'})$和与 Critic 网络对应的评论家目标网络(Target-Critic)$Q(s,a|\theta^{Q'})$组成，

此外还包含用于增加环境探索能力的随机噪声 N 和以离线策略的方式为网络提供训练的经验回放池(Replay-Buffer)。

Actor 网络使用一组参数 θ^{μ} 来代表当前的确定性策略，通过该策略输出动作，而累计奖励 $Q^{\pi}=\mathbb{E}[R_t|s_t,a_t]$ 与动作相关，通过梯度上升对 θ^{μ} 进行更新，可以使 Q^{π} 上升。Critic 网络使用一组参数 θ^{Q} 来估计当前状态动作下的 Q 值，Q 值以链式法则的形式对 Actor 网络梯度更新产生影响，因此准确的 Q 值对网络收敛有着非常重要的影响，通过最小化损失函数对 θ^{Q} 进行更新，可以使 Q 值更加准确。Target-Actor 网络通过参数 $\theta^{\mu'}$ 来估计目标动作，Target-Critic 网络通过参数 $\theta^{Q'}$ 来估计目标 Q 值。式(10)给出了目标网络的参数更新方式，采用滑动平均的方式，与真实网络存在一定延迟。τ 为滑动平均系数，在实际应用中，为了切断数据相关性，需保证目标网络与真实网络存在一定的差异，所以 τ 值远远小于 1。

$$\begin{aligned}\theta^{Q'}&\leftarrow\tau\theta^{Q}+(1-\tau)\theta^{Q'}\\ \theta^{\mu'}&\leftarrow\tau\theta^{\mu}+(1-\tau)\theta^{\mu'}\end{aligned}\tag{10}$$

基于 DDPG 的交会制导策略生成算法如算法 2 所示。

算法 2 基于 DDPG 的交会制导策略生成算法

1. 初始化 Actor 网络 $\mu(s|\theta^{\mu})$ 和 Critic 网络 $Q(s,a|\theta^{Q})$
2. 初始化目标网络 μ' 和 Q'，其权重分别与 μ 和 Q 一致
3. 初始化经验回放缓冲区 R
4. for k=0,1,2,3…:
5. 初始化追踪航天器动作探索的随机过程即噪声 $\mathcal{N}$
6. 根据 3.1 节状态空间初始化追踪航天器状态 s_1
7. for t=0,1,2,3…T:
8. 根据当前策略和噪声选择动作 $a_t=\mu(s_t|\theta^{\mu})+\mathcal{N}t$
9. 追踪航天器根据动作 a_t 产生相应发动机推力，并基于障碍预警与避撞约束的奖励机制与状态转移方程计算奖励 r_t 和下一个状态 s_{t+1}
10. 存储经验数据(s_t,a_t,r_t,s_{t+1})到 R 中
11. 从 R 中随机采样一个小批量的转移(s_i,a_i,r_i,s_{i+1})，令 $y_i=r_i+\gamma Q'(s_{i+1},\mu'(s_{i+1}|\theta^{\mu'})|\theta^{Q'})$
12. 通过最小化损失函数来更新 Critic：

$$L=\frac{1}{N}\sum_i(y_i-Q(s_i,a_i|\theta^{Q}))^2$$

13. 使用采样策略梯度更新 Actor 的策略：

$$\nabla_{\theta^{\mu}}J\approx\frac{1}{N}\sum_i\nabla_a Q(s,a|\theta^{Q})|_{s=s_i,a=\mu(s_i)}\nabla_{\theta^{\mu}}\mu(s|\theta^{\mu})|_{s_i}$$

14. 使用式(10)更新目标网络
15. end for
16. end for

DDPG 算法具体的神经网络结构以及参数分别如表 6 和表 7 所列。为了方便后续的比较分析，DDPG 算法与 PPO 算法的神经网络架构基本一致，两种算法都有的常规参数(如折扣因子、学习率、优化器等)也保持一致。Actor 网络输入层和中间隐藏层与 PPO 算法的 Actor 网络保持一致，不同的是，Actor 网络输出的不是分布，而是确定的动作值。Critic 网络的输入除了四维状态外，还增加了二维动作变量。Critic 网络输出层是一维标量，用以评估当前策略的价值，即 $Q(s,a)$，其余参数与 Actor 网络保持一致。DDPG 算法总共与环境交互了 3×10^6 次。

表 6 DDPG 算法的神经网络结构

Table 6 Neural network structure of DDPG algorithm

层	Actor 网络		Critic 网络	
	节点数	激活函数	节点数	激活函数
输入层	4	ReLU	6	ReLU
隐藏层 1	256	ReLU	256	ReLU
隐藏层 2	256	ReLU	256	ReLU
输出层	2	Tanh	1	Linear

表 7 DDPG 算法的参数

Table 7 Parameters of DDPG algorithm

参数	值
折扣因子 γ	0.99
Actor 学习率	0.000 03
Critic 学习率	0.000 03
优化器	Adam
训练轮次	3 000
经验回放池容量	2 000 000
采样批量大小 *batch_size*	256
软更新系数 τ	0.003
每轮步数 *steps_per_epoch*	1 000
总步数 *total_steps*	3 000 000

4 实验结果及分析

本文实验均在 Windows10 工作站进行，CPU 为 16 核 Intel Xeon E5-2620，GPU 为 GTX 1080Ti，工作站内存容量为 384 GB。两种算法都训练了 3×10^6 步。使用的编程语言为 Python3.6，并基于 PyTorch 训练网络，同时使用 GPU 加速，PPO 算法和 DDPG 算法的训练分别花费了大约 3 h 和 15 h。

4.1 仿真实验

追踪航天器感知自身的状态，包括相对位置和相对速度，并输入到策略网络中，由策略网络输出策略。如第 3 节所述，PPO 生成的策略网络输出的是两组相互独立的高斯分布的均值与方差，追踪航天器直接将均值作为策略从而产生相应的发动机推力，而 DDPG 算法生成的策略网络输出的是确定性动作值，可直接被追踪航天器作为策略产生相应的发动机推力，接着根据状态转移方程更新追踪航天器的下一个状态，最后重复上述操作直至完成任务。

实验设置初始状态 $s_0=[476.13\,\text{m},467.85\,\text{m},0\,\text{m/s},0\,\text{m/s}]$，在训练得到的策略网络控制下，航天器交会轨迹如图 5 所示，可以看到追踪航天器最终能够到达目标航天器附近并稳定在该处，同时在交会过程中没有与障碍物发生碰撞。

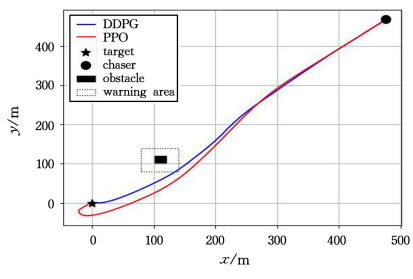


图 5 航天器交会轨迹

Fig. 5 Spacecraft rendezvous trajectory

图 6 给出了追踪航天器在 PPO 算法和 DDPG 算法生成的策略下的交会过程的相对位置和相对速度的变化，从图中可以看出，在距离较远时，追踪航天器会以较快的相对速度

靠近目标航天器，但随着与目标航天器的距离越来越近，追踪航天器相对速度逐渐减小，靠近目标航天器后趋近于零。可以看到，经过训练，追踪航天器能逼近目标航天器，并稳定在附近，最终相对位置和速度都趋近于零，完成交会任务。

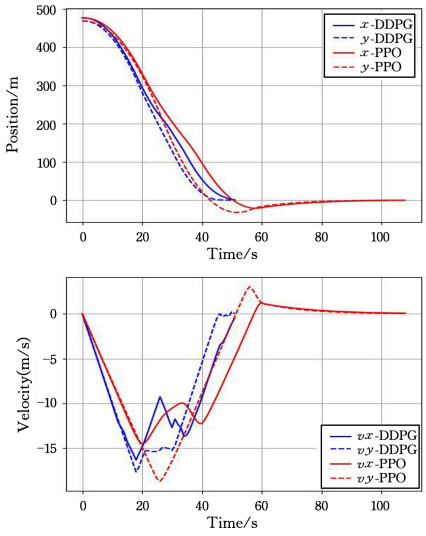


图 6 航天器交会的相对位置和速度变化

Fig. 6 Spacecraft rendezvous relative position and relative velocity

为了证明本文方法的有效性，还基于 PPO 算法和 DDPG 算法训练了另外一组未考虑障碍预警与避撞约束的奖励机制的策略网络，与本文方法进行对比。在训练得到的策略网络控制下，航天器交会轨迹如图 7 和图 8 所示，虚线表示未考虑障碍预警与避撞约束的奖励机制所训练出来的策略网络生成的轨迹，实线则是本文提出的方法所训练出来的策略网络生成的轨迹。可以看到，在未考虑障碍预警与避撞约束的奖励机制下，追踪航天器最终会与障碍物发生碰撞，而根据本文提出的方法，无论是 PPO 还是 DDPG 算法训练出来的智能体都能使航天器避开障碍物并完成交会。

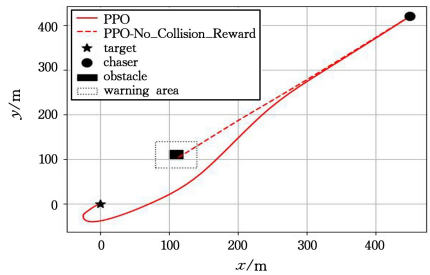


图 7 基于 PPO 算法的交会轨迹

Fig. 7 Rendezvous trajectory based on PPO algorithm

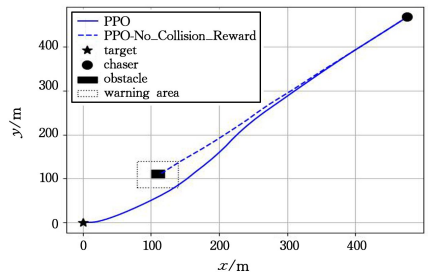


图 8 基于 DDPG 算法的交会轨迹

Fig. 8 Rendezvous trajectory based on DDPG algorithm

进一步地，实验随机选取了 100 个测试点作为追踪航天器的出发起点，以测试两种不同算法训练出来的策略网络产生的策略能否完成交会任务，结果如表 8 所列。从表 8 可以发现，基于本文提出的方法，两种算法的成功率都达到了 100%，而在未考虑障碍预警与避撞约束的奖励机制的情况下，两种算法都有较高的碰撞率，证明了本文提出方法的有效性。

表 8 两种算法的成功率比较

Table 8 Comparison of success rate of two algorithms

算法	成功率/%	碰撞率/%	是否有避撞奖励机制
PPO	100	0	是
DDPG	100	0	是
PPO	69	31	否
DDPG	73	27	否

4.2 算法比较

为了对比两种算法及其生成的策略网络的性能优劣，从训练效率、训练时间、任务效果这 3 个角度进行比较。

(1)训练效率

从平均回合奖励、成功交会的平均回合比率、发生碰撞的平均回合比率来比较训练效率。

图 9 给出了两种算法的学习曲线，它描述了在训练过程中的平均回合奖励变化关系，即所有回合累计奖励的平均值，这也是 DRL 算法最大化的目标。

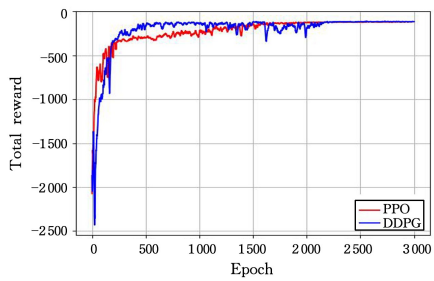


图 9 平均回合奖励

Fig. 9 Average episode rewards

图 10 给出了两种算法在训练过程中使航天器能够成功交会的平均回合比率的变化关系，以衡量算法对该任务的训练效率。达到更高的成功率所需的迭代次数更少，则表明训练效率更高。

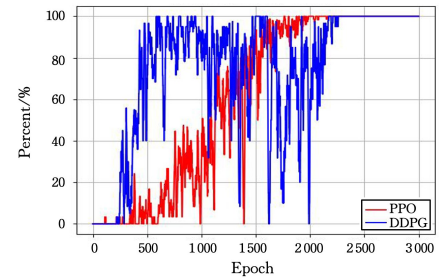


图 10 成功交会的平均回合比率

Fig. 10 Average percentage of episodes of successful rendezvous

图 11 给出了两种算法在训练过程中使航天器发生碰撞的平均回合比率的变化关系。

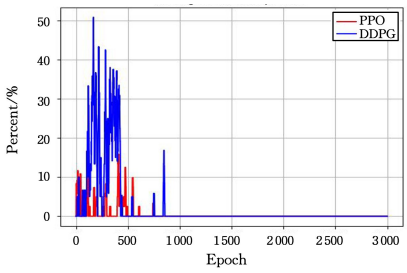


图 11 发生碰撞的平均回合比率

Fig. 11 Average percentage of episodes of collision

从图 9 来看，PPO 算法平均回合奖励收敛稍快一些，在 2000 轮迭代左右收敛，并且增长过程稳定，而 DDPG 算法在 2500 轮迭代左右收敛。从图 10 来看，随着训练的进行，DDPG 算法率先到达 100% 的成功率，但 DDPG 算法浮动频率更大，PPO 算法和 DDPG 算法交会成功率最终都能达到 100%。从图 11 来看，DDPG 算法在前期训练中会频繁与障碍物发生碰撞，而 PPO 算法在前期训练中发生碰撞的概率比 DDPG 算法小得多，两者在后期均保持着零碰撞概率。综上，在训练期间 PPO 算法训练效率更高，稳定性更好。

(2)训练时间

在同样的环境下，基于 PyTorch 训练网络，并且使用 GPU 加速，DDPG 算法完成训练需耗时 15 h 左右，而 PPO 算法仅需 3 h 左右，即 PPO 算法的训练速度大约是 DDPG 算法的 5 倍。从表 5 和表 7 的参数设置可以发现，即使 PPO 算法和环境的交互次数与 DDPG 算法一样，它仍然比 DDPG 算法快得多，主要原因在于 DDPG 算法是离线策略方法，它需要一个较大的经验回放池，每次采样出的数据都存放在池中，并且数据会被多次取出进行训练网络，而每轮训练时，又要多次从经验回放池中读取小批量数据以更新网络。相比 PPO 算法而言，DDPG 算法更新网络更为频繁。另外，PPO 算法实现更容易，对超参数敏感性低，而 DDPG 实现复杂，并且超参数多，调参较为繁琐。

(3)任务效果

实验随机选取了 100 个测试点作为追踪航天器的启动起点，来测试两种不同算法训练出来的策略网络实际的交会效果，结果如表 9 所列。平均交会终点表示 100 次测试中所有成功交会的情况下追踪航天器最终停留的平均状态值，可以用来衡量交会精度。从表 9 中可以发现，两者的成功率都达到了 100%，且都能达到较高的交会精度，但在平均奖励和平均回合长度方面，DDPG 算法比 PPO 算法表现更好。

表 9 两种算法的任务效果比较

Table 9 Comparison of rendezvous effects between two algorithms

算法	平均奖励	成功率/%	平均交会终点	平均回合长度
PPO	−137.80	100	[0.35 m, 0.30 m, 0.06 m/s, 0.03 m/s]	107.56
DDPG	−121.55	100	[0.14 m, 0.31 m, 0.11 m/s, 0.08 m/s]	59.40

4.3 扩展性测试

为了检验 DRL 的泛化能力，实验使用了新的测试点来测试具有未接受过训练的观察状态的策略。同样，实验随机选取了 100 个测试点作为追踪航天器的启动起点，但与之前不同的是，追踪航天器出发位置由表 1 设置的[400 m, 500 m]扩展到[600 m, 800 m]，观察两种算法在新的状态下的任务完成度，结果如表 10 所列。图 12 给出了用两种算法在其中一组扩展测试点上所产生的追踪航天器交会轨迹和追踪航天器相对位置(即 x 和 y)随时间的变化关系。从表 10 中可以发现，两者训练出来的网络在扩展测试范围后都能使航天器避开障碍物并完成交会，任务成功率都为 100%，依然保持着较高的交会精度，这体现了本文方法具有一定的泛化能力。进一步地，对比两种算法发现，无论从平均奖励还是从平均回合长度来看，DDPG 算法都是表现更好的一方，这与 4.2 节分析的情况一致，即 DDPG 算法的实际任务效果更好。

表 10 经过扩展测试范围后的两种算法的任务效果比较

Table 10 Comparison of rendezvous effects between two algorithms after expanded test range

算法	平均奖励	成功率/%	平均交会终点	平均回合长度
PPO	−303.30	100	[0.33 m, 0.32 m, 0.05 m/s, 0.03 m/s]	136.94
DDPG	−266.20	100	[0.09 m, 0.35 m, 0.07 m/s, 0.06 m/s]	78.56

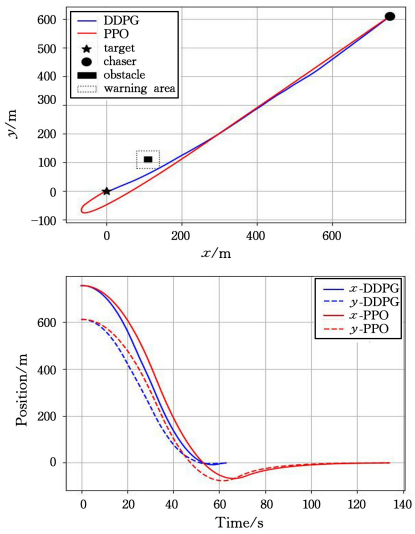


图 12 扩展测试范围后的交会轨迹和相对位置变化

Fig. 12 Spacecraft rendezvous trajectory and relative position with extended position range

综上所述，可以得出如下结论：

(1) 仿真实验结果表明，本文提出的方法能有效地进行障碍物避撞并以较高的精度完成交会。

(2) 算法比较结果表明，两种算法各有优劣，PPO 算法训练效率更高并且更加稳定和快速，而 DDPG 算法的实际任务效果更好。

(3) 扩展性测试结果表明，本文提出的方法能有效地应用于没有接受过训练的任务，具有较好的泛化能力。

结束语 本文提出了基于安全强化学习的航天器交会制导方法。首先，设计避撞场景下航天器自主交会的马尔可夫模型，并提出基于障碍预警与避撞约束的奖励机制，从而建立用于求解航天器交会制导策略的安全强化学习框架。其次，

在该安全强化学习框架下,基于 PPO 和 DDPG 两种深度强化学习算法生成制导策略。实验结果表明,该方法能有效地进行障碍物避撞并以较高的精度完成交会。另外,还进行了两种算法的比较和扩展性测试,从而进一步证明了本文方法的有效性。

在未来的工作中,我们将进一步考虑动态障碍物避撞的航天器安全交会制导问题,研究基于安全盾牌(Shielding)和运行时监控(Runtime Monitor)的安全强化学习方法。另外,航天器制导控制系统是一类安全关键系统,系统的安全性与可靠性极为重要,因此我们将研究安全强化学习模型的形式化验证方法。

参 考 文 献

[1] BOYARKO G,YAKIMENKO O,ROMANO M. Optimal rendezvous trajectories of a controlled spacecraft and a tumbling object[J]. Journal of Guidance, Control, and dynamics, 2011, 34(4):1239-1252.

[2] WEISS A,BALDWIN M,ERWIN R S,et al. Model predictive control for spacecraft rendezvous and docking: Strategies for handling constraints and case studies[J]. IEEE Transactions on Control Systems Technology,2015,23(4):1638-1647.

[3] XU D D,ZHANG J. A collision-avoidance control algorithm for spacecraft proximity operations based on improved artificial potential function[J]. Chinese Journal of Theoretical and Applied Mechanics,2020,52(6):1581-1589.

[4] DUTTA S,MISRA A K. Convex optimization of collision avoidance maneuvers in the presence of uncertainty[J]. Acta Astronautica,2022,197:257-268.

[5] BROIDA J,LINARES R. Spacecraft rendezvous guidance in cluttered environments via reinforcement learning[C]// 29th AAS/AIAA Space Flight Mechanics Meeting. American Astronautical Society Ka'anapali,Hawaii,2019:1-15.

[6] DAI S S,LIU Q. Action Constrained Deep Reinforcement Learning Based Safe Automatic Driving Method[J]. Computer Science,2021,48(9):235-243.

[7] XIE W C,LI B,DAI Y Y. PPO Based Task Offloading Scheme in Aerial Reconfigurable Intelligent Surface-assisted Edge Computing[J]. Computer Science,2022,49(6):3-11.

[8] HONG Z L,LAI J,CAO L,et al. Study on Intelligent Recommendation Method of Dueling Network Reinforcement Learning Based on Regret Exploration[J]. Computer Science, 2022, 49(6):149-157.

[9] LI B B,SONG J R,DU Q Y,et al. DRL-IDS:Deep Reinforcement Learning Based Intrusion Detection System for Industrial Internet of Things[J]. Computer Science,2021,48(7):47-54.

[10] WANG X,WANG G,CHEN Y,et al. Autonomous Rendezvous Guidance via Deep Reinforcement Learning[C]// 2020 Chinese Control and Decision Conference(CCDC). IEEE, 2020: 1848-1853.

[11] HOVELL K,ULRICH S. Deep reinforcement learning for spacecraft proximity operations guidance[J]. Journal of Spacecraft and Rockets,2021,58(2):254-264.

[12] FEDERICI L,BENEDIKTER B,ZAVOLI A. Machine Learning Techniques for Autonomous Spacecraft Guidance during Proximity Operations[C]// AIAA Scitech 2021 Forum. 2021.

[13] GARCIA J,FERNANDEZ F. A comprehensive survey on safe reinforcement learning[J]. Journal of Machine Learning Research,2015,16(1):1437-1480.

[14] YANG Z B,XING L Q,GU Z H,et al. Model-based Reinforcement Learning and Neural Network-based Policy Compression for Spacecraft Rendezvous On Resource-Constrained Embedded Systems[J]. IEEE Transactions on Industrial Informatics,2022, 19(1):1107-1116.

[15] ZHOU J P. Space rendezvous and docking technology[M]. National Defense Industry Press,2013.

[16] SCHAUB H,JUNKINS J L. Analytical mechanics of space systems[M]. AIAA,2003.

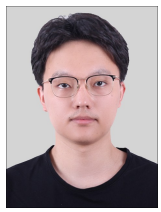
[17] SCHULMAN J,WOLSKI F,DHARIWAL P,et al. Proximal policy optimization algorithms[J]. arXiv:1707.06347,2017.

[18] LILLICRAP T P,HUNT J J,PRITZEL A,et al. Continuous control with deep reinforcement learning[J]. arXiv: 1509.02971,2015.

[19] SCHULMAN J,LEVINE S,ABBEEL P,et al. Trust region policy optimization[C]// International Conference on Machine Learning. PMLR,2015:1889-1897.

[20] MNIH V,KAVUKCUOGLU K,SILVER D,et al. Playing atari with deep reinforcement learning[J]. arXiv:1312.5602,2013.

[21] SCHULMAN J,MORITZ P,LEVINE S,et al. High-dimensional continuous control using generalized advantage estimation[J]. arXiv:1506.02438,2015.



XING Linquan, born in 1998, postgraduate. His main research interests include reinforcement learning and safety-critical software.

YANG Zhibin, born in 1982, Ph.D, professor, postdoctoral researcher. His main research interests include safety-critical system, formal verification and AI software engineering.

(责任编辑:何杨)