自动化学报
ACTA AUTOMATICA SINICA

基于终端诱导强化学习的航天器轨道追逃博弈

耿远卓 袁利 黄煌 汤亮

Terminal-guidance Based Reinforcement-learning for Orbital Pursuit-evasion Game of the Spacecraft

GENG Yuan-Zhuo, YUAN Li, HUANG Huang, TANG Liang

在线阅读 View online: https://doi.org/10.16383/j.aas.c220204

基于终端诱导强化学习的航天器轨道追逃博弈

耿远卓[1, 2] 袁 利[2, 3] 黄 煌[1, 2] 汤 亮[1, 2]

摘 要 针对脉冲推力航天器轨道追逃博弈问题, 提出一种基于强化学习的决策方法, 实现追踪星在指定时刻抵近至逃逸星的特定区域, 其中两星都具备自主博弈能力. 首先, 充分考虑追踪星和逃逸星的燃料约束、推力约束、决策周期约束、运动范围约束等实际约束条件, 建立锥形安全接近区及追逃博弈过程的数学模型; 其次, 为了提升航天器面对不确定博弈对抗场景的自主决策能力, 以近端策略优化 (Proximal policy optimization, PPO) 算法框架为基础, 采用左右互搏的方式同时训练追踪星和逃逸星, 交替提升两星的决策能力; 在此基础上, 为了在指定时刻完成追逃任务, 提出一种终端诱导的奖励函数设计方法, 基于 CW (Clohessy Wiltshire) 方程预测两星在终端时刻的相对误差, 并将该预测误差引入奖励函数中, 有效引导追踪星在指定时刻进入逃逸星的安全接近区. 与现有基于当前误差设计奖励函数的方法相比, 所提方法能够有效提高追击成功率. 最后, 通过与其他学习方法仿真对比, 验证提出的训练方法和奖励函数设计方法的有效性和优越性.

关键词 航天器追逃, 智能博弈, 近端策略优化, 奖励函数设计, 终端诱导

引用格式 耿远卓, 袁利, 黄煌, 汤亮. 基于终端诱导强化学习的航天器轨道追逃博弈. 自动化学报, 2023, **49**(5): 974−984

DOI 10.16383/j.aas.c220204

Terminal-guidance Based Reinforcement-learning for Orbital Pursuit-evasion Game of the Spacecraft

GENG Yuan-Zhuo[1, 2] YUAN Li[2, 3] HUANG Huang[1, 2] TANG Liang[1, 2]

Abstract This paper addresses the problem of orbital pursuit-evasion of two multi-impulse satellites with strong maneuver ability. A reinforcement-learning based approach is proposed to train two satellites such that the pursuer can reach to a specific region adjacent to the evader at the appointed time. First, by taking fuel limits, control force limits, control frequency, and range of motion into consideration, the model for conical approach region and orbital dynamics of relative motion between two satellites is established. Based on this model, to enhance the ability of confronting with the situations with high uncertainties, the proximal policy optimization (PPO) scheme is adopted to train the pursuer and the evader alternately. Moreover, to accomplish the pursuit or evasion at the appointed time, a new kind of reward function is designed based on the final predicted error, which guides the pursuer to approach the evader approximately at the prescribed time. Compared with existing reward function design methods based on the current error, the proposed method in this paper can effectively enhance the success rate of pursuit. Finally, the simulation comparisons are conducted to show the superiority of the terminal-guidance reward function proposed in this paper over traditional reward function design approaches.

Key words Spacecraft pursuit and evasion, intelligent game, proximal policy optimization (PPO), reward function design, terminal-guidance

Citation Geng Yuan-Zhuo, Yuan Li, Huang Huang, Tang Liang. Terminal-guidance based reinforcement-learning for orbital pursuit-evasion game of the spacecraft. *Acta Automatica Sinica*, 2023, **49**(5): 974−984

随着航天技术的进步, 空间态势日益复杂, 各航天强国积极发展下一代智能化航天器, 并在轨开展绕飞、巡视、交会对接等新技术试验. 面对日益复杂的空间态势, 传统依赖地面指控的决策方式难以及时应对边界模糊、变化迅速的时敏空间任务[1]. 因此, 研究新一代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, 提高应对不确定博弈对抗场景的能力.

轨道追逃是博弈态势中最具代表性的一类场景[2]. 轨道追逃博弈本质上属于双边连续最优决策问题, 由于对方策略的不可预知性, 难以计算自身的期望状态, 从而无法预先规划轨迹, 任何一方的策略都

收稿日期 2022-03-21 录用日期 2022-09-26
Manuscript received March 21, 2022; accepted September 26, 2022
国家自然科学基金 (U21B6001), 中国博士后科学基金 (2022M722994) 资助
Supported by National Natural Science Foundation of China (U21B6001) and China Postdoctoral Science Foundation (2022M722994)
本文责任编委 穆朝絮
Recommended by Associate Editor MU Chao-Xu
1. 北京控制工程研究所 北京 100094 2. 空间智能控制技术重点实验室 北京 100094 3. 中国空间技术研究院 北京 100094
1. Beijing Institute of Control Engineering, Beijing 100094 2. Science and Technology on Space Intelligent Control Laboratory, Beijing 100094 3. China Academy of Space Technology, Beijing 100094

需要随着对方策略的变化而调整. 因此, 求解轨道追逃博弈问题具有较大的挑战.

目前的追逃博弈研究可以分为两种方法, 一种是在最优控制的框架下, 基于微分对策理论求解双边最优轨迹; 另一种是在强化学习的框架下, 通过学习训练使双方具备自主决策能力, 完成追击或者逃逸任务. 以下对上述两种方法简要介绍和评析.

基于微分对策求解追逃博弈问题的思路最早由美国兰德公司的 Isaacs 提出[3]. 该方法与最优控制理论相辅相成, 通过动态规划或者极大值原理求解双边最优轨迹, 双方达到纳什均衡. 求解该问题的关键是求解非线性偏微分 HJI (Hamilton-Jacobi-Isaacs) 方程, 然而对于大部分场景, 该方程没有解析解. 因此, 很多学者基于极大值原理将最优化问题转化为两点边值问题, 然后采用打靶法[4]、半直接配点法[5]、高斯伪谱法[6] 等方法进行求解. 针对追逃问题, 文献 [7] 定义了捕获区和逃逸区, 两者分界面称之为界栅. 界栅对于追逃双方十分重要, 是塑造各自态势优势的关键因素. 因此, 微分对策的相关研究很多都聚焦于求解界栅[8−13].

目前微分对策的研究成果大多集中于连续小推力控制, 然而对于高轨航天器, 大多数采用大推力脉冲变轨方式, 上述方法难以直接应用, 相应研究成果较少[14−15]. 尤其当考虑决策周期、燃料上限、推力幅值上限这三类强约束时, 微分对策求解难度较大.

随着人工智能技术的发展, 强化学习受到广泛关注. 强化学习的核心是通过智能体与环境交互, 得到自身奖励, 引导智能体朝向期望目标运动[16]. 强化学习尤其适合解决时变不确定对抗任务, 在围棋、Dogfight 等任务中表现突出, 可在一定规则限制下战胜人类. 因此, 利用强化学习技术对航天器进行赋能, 为航天器在轨博弈提供了一种新的求解思路.

强化学习领域最具代表性的是行动者−评论家 (Actor-critic) 算法架构. 其核心是通过行动者产生动作策略, 然后通过评论家对当前策略进行评价, 指导动作策略的调整. 该框架衍生出包含置信域策略优化算法[17]、近端策略优化算法 (Proximal policy optimization, PPO)[18] 等在内的众多算法. 行动者−评论家架构由于训练过程指导信息完备, 有助于降低训练方差, 且评论家与控制理论中的反馈功能相似, 因此成为强化学习和传统控制理论结合的纽带[19]. 本文针对轨道追逃博弈, 采用 PPO 算法的训练架构, 通过在数字环境中开展多回合博弈训练, 得到最优的轨道机动策略.

由于航天器的研制成本较高, 其制导控制策略一般较为简单和保守. 目前在姿轨控任务中实际应用的人工智能技术仅为专家系统[20]. Izzo 等[20] 将神经网络应用于多航天器位姿协同控制, 并在国际空间站内开展了试验. 文献 [21] 综述了机器学习技术在航天器导航、制导与控制领域的应用情况, 总结了人工智能应用的难点和未来发展方向. 强化学习作为机器学习的一个分支, 在航天领域的理论研究主要集中在星际转移、行星着陆、在轨操控等任务的轨迹规划. Gaudet 等[22−23] 采用 PPO 算法框架, 设计了火星动力下降段的制导策略. 文献 [24] 利用强化学习产生制导策略, 然后设计控制律进行跟踪, 有助于解决训练模型和实际模型不一致的问题. 针对行星转移轨道设计问题, Zavoli 等[25] 设计了参数自适应的稀疏奖励函数, 基于 PPO 算法训练航天器, 使之在测量误差、执行机构偏差和环境干扰影响情况下依然能够以最少燃料消耗精准转移.

这种由强化学习训练得到的机动策略完全依赖于试错机制, 不存在解析解, 难以分析算法的稳定性和收敛性. 但是由于可以将推力饱和、碰撞规避等约束条件融入训练过程, 其灵活性更强, 智能性更高. 另一类研究是在传统的制导与控制算法框架下, 引入强化学习, 以提升系统性能[26−28]. 这种决策方式针对确定性任务可以取得较好的效果, 但是难以解决复杂约束多、不确定性强的博弈任务.

对于空间博弈问题, 目前强化学习技术应用较少. 文献 [29] 基于 PPO 算法设计了多机协同空战决策流程. 赵毓等[30] 针对多星碎片规避协同决策问题, 提出一种奖励函数设计方法, 用历史数据的经验信息代替传统人为设计的奖励函数, 提高了算法的学习效率. 文献 [31] 针对非合作目标交会问题, 提出分支深度强化学习架构, 将强化学习的策略输入给模糊推理模型, 通过模糊推理将连续状态空间离散化, 解决了行为数量与映射规则的组合增长问题.

综上, 现有航天器强化学习主要针对的是目标确定、模型未知的非对抗任务. 对于具有强博弈能力的非合作航天器, 目前利用强化学习对航天器赋能的研究较少. 同时, 对于时敏空间任务, 时间约束对于任务成败十分重要. 本文针对带有终端时刻约束的航天器轨道追逃博弈问题开展研究, 主要工作及创新点如下: 提出一种终端诱导的奖励函数设计方法, 充分利用轨道动力学特性, 通过轨迹预测, 估计终端时刻追踪星相对目标星的位置误差, 并利用该预测误差设计奖励函数. 现有强化学习算法中,

奖励函数的设计大都基于当前时刻的误差, 未能充分利用已知的轨道动力学模型. 对于指定时刻追逃的博弈问题, 基于即时误差设计的奖励函数会加大训练难度, 降低算法的收敛速度. 本文提出的奖励函数设计方法能够引导追踪星在终端时刻附近抵近逃逸星, 满足时间约束, 同时提升训练效率.

1 问题描述

本文假设逃逸星及追踪星运行于地球同步轨道 (Geostationary orbit, GEO) 附近, 如图 1 所示, 追踪星需要通过轨道机动, 在指定时刻到达逃逸星的锥形安全接近走廊内, 为后续交会对接等任务奠定基础. 逃逸星具有一定的轨道机动能力, 可根据追踪星的位置速度进行规避, 使得追踪星远离自身的安全接近走廊. 同时, 逃逸星为了维持其原有的对地通信、气象观测等任务, 需要尽可能保持原有轨位及对地指向.

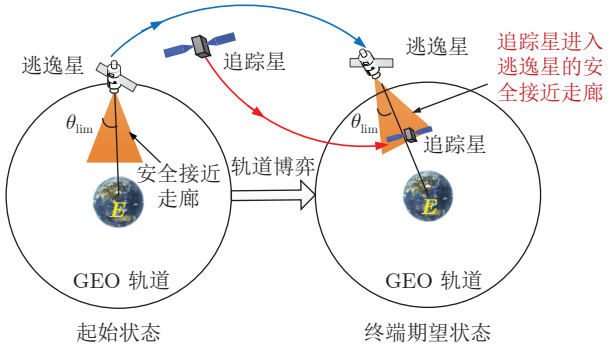


图 1 轨道博弈位置变化示意图

Fig. 1 The orbit change for the pursuit-evasion game

由于本文仅考虑近距离交会阶段, 整个运行过程中两者相对距离远小于 GEO 的轨道半径, 因此可采用 CW (Clohessy Wiltshire) 方程近似描述两者的相对轨道运动, 如下式所示

$$\begin{cases} \ddot{r}_{io\_x} - 2\omega_o \dot{r}_{io\_z} = \dfrac{F_{ix}}{m_i} \\ \ddot{r}_{io\_y} + \omega_o^2 r_{io\_y} = \dfrac{F_{iy}}{m_i} \\ \ddot{r}_{io\_z} + 2\omega_o \dot{r}_{io\_x} - 3\omega_o^2 r_{io\_z} = \dfrac{F_{iz}}{m_i} \end{cases} \tag{1}$$

式中, 变量下标 $i = \{p, e\}$ 代表追踪星 (pursuer) 或者逃逸星 (evader). $\boldsymbol{r}_{io}^o = [r_{io\_x}, r_{io\_y}, r_{io\_z}]^{\mathrm{T}}$ 为航天器相对于 CW 参考系的位置矢量在 CW 参考轨道系中的表示. $\boldsymbol{v}_{io}^o = [\dot{r}_{io\_x}, \dot{r}_{io\_y}, \dot{r}_{io\_z}]^{\mathrm{T}}$ 为航天器相对于 CW 参考系的速度. 各矢量定义如图 2 所示. 其中 CW 参考轨道系的原点在 GEO 轨道上, $\boldsymbol{x}_o$ 沿

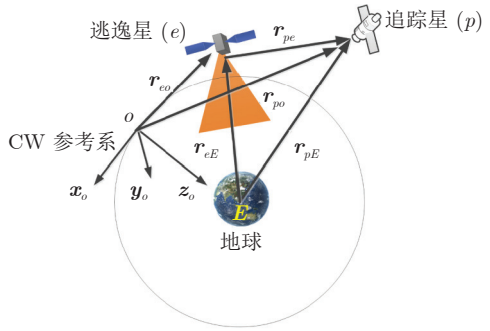


图 2 相对位置矢量示意图

Fig. 2 The diagram of the relative position

轨道速度方向, $\boldsymbol{z}_o$ 指向地心, m_i 为航天器质量. $\boldsymbol{y}_o = \boldsymbol{z}_o \times \boldsymbol{x}_o$, $F_i = [F_{ix}, F_{iy}, F_{iz}]^{\mathrm{T}}$ 为航天器推力矢量, 本文考虑常值推力, 追踪星和逃逸星沿本体三轴的推力幅值固定, 仅可通过调节发动机的开机时间调节速度脉冲增量.

对于式 (1) 所示的线性系统, 状态方程可显式求解, 得到状态随时间的变化方程如下

$$\boldsymbol{X}(t) = \boldsymbol{\Phi}(t, t_0)\boldsymbol{X}(t_0) + \int_{t_0}^{t} \boldsymbol{\Phi}_v(t, \tau)\boldsymbol{a}\mathrm{d}\tau \tag{2}$$

其中, $\boldsymbol{X}(t) = [\boldsymbol{r}_{io}^o(t)^{\mathrm{T}}, \boldsymbol{v}_{io}^o(t)^{\mathrm{T}}]^{\mathrm{T}}$, $\boldsymbol{a} = \boldsymbol{F_i}/m_i$, $\boldsymbol{X}(t_0) = [\boldsymbol{r}_{io}^o(t_0)^{\mathrm{T}}, \boldsymbol{v}_{io}^o(t_0)^{\mathrm{T}}]^{\mathrm{T}}$. $\boldsymbol{\Phi}(t, t_0)$ 为状态转移矩阵, 具体表达式详见文献 [15].

根据式 (2) 可得到任意时刻追踪星及逃逸星在 CW 参考系中的位置和速度

$$\boldsymbol{X}(t) = \boldsymbol{\Phi}(t, t_0)\boldsymbol{X}(t_0^+) \tag{3}$$

其中, $\boldsymbol{X}(t_0^+) = [\boldsymbol{r}_{io}^o(t_0^-)^{\mathrm{T}}, (\boldsymbol{v}_{io}^o(t_0^-) + \Delta\boldsymbol{V}(t_0))^{\mathrm{T}}]^{\mathrm{T}}$, 表征在 t_0 时刻进行脉冲推力后的轨道位置及速度. 在任意时刻 t, 根据式 (3) 可预测在无控条件下终端时刻 t_{f} 追踪星相对于逃逸星的位置

$$\boldsymbol{r}_{pe}^o(t_{\mathrm{f}}) = \boldsymbol{r}_{po}^o(t_{\mathrm{f}}) - \boldsymbol{r}_{eo}^o(t_{\mathrm{f}}) = [\boldsymbol{\Phi}_{pp}(t_{\mathrm{f}}, t), \boldsymbol{\Phi}_{pv}(t_{\mathrm{f}}, t)] \begin{bmatrix} \boldsymbol{r}_{po}^o(t) - \boldsymbol{r}_{eo}^o(t) \\ \boldsymbol{v}_{po}^o(t^+) - \boldsymbol{v}_{eo}^o(t^+) \end{bmatrix} \tag{4}$$

追踪星为了能够在终端时刻 t_{f} 进入图 1 所示的安全接近走廊, 其终端位置需要满足

$$\begin{cases} \|\boldsymbol{r}_{pe}^o(t_{\mathrm{f}})\| \le r_{\lim} \\ \theta_e(t_{\mathrm{f}}) = \arccos\left(\dfrac{\boldsymbol{r}_{eE}(t_{\mathrm{f}}) \cdot \boldsymbol{r}_{pe}(t_{\mathrm{f}})}{\|\boldsymbol{r}_{eE}(t_{\mathrm{f}})\| \, \|\boldsymbol{r}_{pe}(t_{\mathrm{f}})\|}\right) \le \theta_{\lim} \end{cases} \tag{5}$$

其中, θ_e 表示追踪星–逃逸星连线 $\boldsymbol{r}_{pe}$ 与逃逸星–地球连线 $\boldsymbol{r}_{eE}$ 的夹角, $\theta_{\lim}$ 为安全接近走廊的锥角.

逃逸星的目的是通过轨道机动, 使得追踪星在终端时刻无法进入自身的安全接近走廊, 即

$$\begin{cases} \left\|\boldsymbol{r}_{pe}^{o}(t_{\mathrm{f}})\right\| > r_{\mathrm{lim}} \\ 或 \\ \theta_e(t_{\mathrm{f}}) = \arccos\left(\dfrac{\boldsymbol{r}_{eE}(t_{\mathrm{f}})\cdot\boldsymbol{r}_{pe}(t_{\mathrm{f}})}{\left\|\boldsymbol{r}_{eE}(t_{\mathrm{f}})\right\|\left\|\boldsymbol{r}_{pe}(t_{\mathrm{f}})\right\|}\right) > \theta_{\mathrm{lim}} \end{cases} \tag{6}$$

同时, 为了保持原始轨位, 逃逸星相对 t_0 时刻的位置需要满足如下约束

$$\left\|\boldsymbol{r}_{eo}(t) - \boldsymbol{r}_{eo}(t_0)\right\| < r_{e\_\mathrm{lim}} \tag{7}$$

除此之外, 综合考虑实际任务中对机动航天器的定轨时间、数据处理时间等制约因素, 追踪星和逃逸星的决策频率 f_{pd} 和 f_{ed} 需要满足

$$\begin{cases} f_{pd} < f_{pd\_\max} \\ f_{ed} < f_{ed\_\max} \end{cases} \tag{8}$$

发动机单次开机时间约束

$$\begin{cases} T_{p\_\mathrm{Time}} < T_{p\_\mathrm{Time}\_\max} \\ T_{e\_\mathrm{Time}} < T_{e\_\mathrm{Time}\_\max} \end{cases} \tag{9}$$

总的速度增量约束

$$\begin{cases} \left\|\Delta\boldsymbol{V}_p\right\| \le \Delta V_{p\_\max} \\ \left\|\Delta\boldsymbol{V}_e\right\| \le \Delta V_{e\_\max} \end{cases} \tag{10}$$

本文的目标就是通过设计追踪星和逃逸星机动策略, 在满足式 (7) ~ 式 (10) 的约束条件下, 实现式 (5) 和式 (6) 的追逃目标. 该问题是典型的双方博弈问题, 博弈三要素分别阐述如下: 1) 参与博弈的智能体集合为 $N = \{p, e\}$, p 表示追踪星, e 表示逃逸星; 2) 动作集合为 $\boldsymbol{U} = \{\Delta\boldsymbol{V}_p, \Delta\boldsymbol{V}_e\}$, 由追踪星和逃逸星的速度脉冲组成; 3) 效用函数集合 $J = \{J_p, J_e\}$, J_p 和 J_e 表示追踪星和逃逸星的损失函数, 具体表达式详见式 (11).

2　基于 PPO 算法的终端诱导强化学习决策

本节将在 PPO 算法的框架下, 提出一种新颖的终端诱导奖励函数设计方法, 引导追踪星在终端指定时刻进入逃逸星的安全接近区域.

2.1　训练架构设计

本文采用左右互搏的训练模式, 同时对追踪星和逃逸星的神经网络进行训练, 两星分别拥有各自的决策网络. 两星的训练算法相同, 都采用 PPO 算法对各自神经网络进行权值更新. 两星训练框架如图 3 所示.

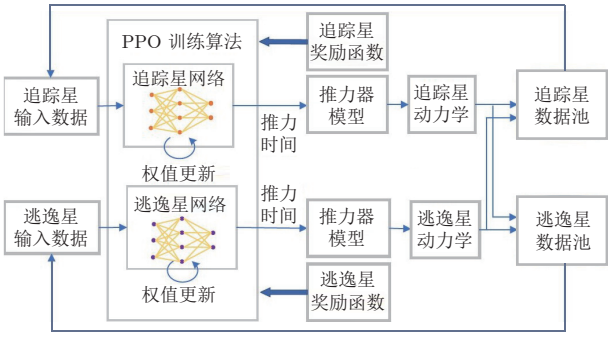


图 3　两星训练框架示意图

Fig. 3　The schematic diagram for training framework of the two spacecraft

在构建神经网络时, 本文将价值网络和动作网络融合为了一个网络, 如图 4 所示. 网络的输出为各轴推力时间以及状态价值函数. 追踪星和逃逸星的网络结构相同.

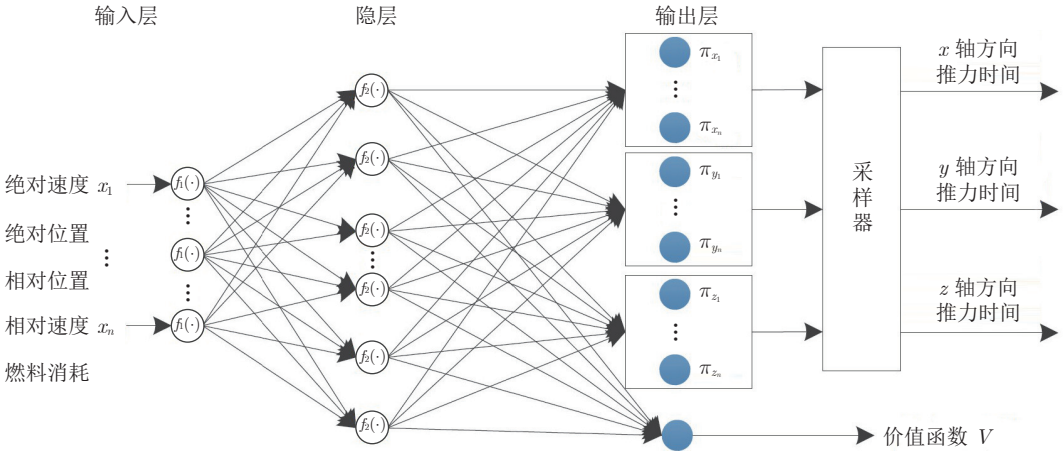


图 4　神经网络信息流向示意图

Fig. 4　The schematic diagram for the data flow in the network

2.2 PPO 算法设计

由于本文在训练追踪星和逃逸星时, 两星都采用 PPO 算法更新网络权值, 区别仅体现在两星的输入数据结构以及奖励函数. 本节将介绍 PPO 算法的原理和设计过程, 第 2.3 节将重点介绍两星各自的奖励函数设计方法.

PPO 算法融合了策略梯度方法和基于价值函数方法的优点, 将策略 (执行各个动作的概率) 表征为网络权值参数 θ 的函数, 同时, 为了降低学习训练过程的方差, 引入价值函数作为基线.

由于强化学习是无监督学习, 其没有预先标注的样本, 因此神经网络的损失函数难以直接得到. 为了训练神经网络, 必须定义合理的损失函数. 本文定义的损失函数为三部分的期望

$$L_t^{\text{Total}}(\theta)=\mathrm{E}_t\left(L_t^{\text{CLIP}}(\theta)-c_1L_t^{VF}(\theta)+c_2S\left[\pi_\theta\right](s_t)\right) \tag{11}$$

神经网络通过训练, 使得 $L_t^{\text{Total}}(\theta)$ 达到极大值. $L_t^{\text{Total}}(\theta)$ 的值与追踪星和逃逸星每一步的奖励函数有关, 两星的奖励函数在第 2.3 节详细介绍. $L_t^{\text{CLIP}}(\theta)$ 为与策略优势相关的函数, 定义为

$$L_t^{\text{CLIP}}(\theta)=\mathrm{E}_t\left(\min\left\{r_t(\theta)\hat{A}_t,\, clip\left(r_t(\theta),\,1-\varepsilon,\,1+\varepsilon\right)\hat{A}_r\right\}\right) \tag{12}$$

其中, $r_t(\theta)=\pi_\theta(a_t\mid s_t)/\pi_{\theta_{\text{old}}}(a_t\mid s_t)$ 为新旧策略的概率比, $\pi_\theta(a_t|s_t)$ 是以参数 θ 表示的在状态 s_t 条件下选择动作 a_t 的概率; $\pi_{\theta_{\text{old}}}(a_t|s_t)$ 是以参数 θ_{old} 表示的在状态 s_t 条件下选择动作 a_t 的概率, θ_{old} 为 θ 的历史值, 在经过一定步数的训练后将 θ 的数值传递给 θ_{old}. $\hat{A}_t$ 为优势函数, 定义为当前的动作价值函数 $q_{\pi_\theta}(s_t,\,a_t)$ 与状态价值函数 $V_{\pi_\theta}(s_t)$ 的差值, 如式 (13) 所示, 其表征当前动作 a_t 相较于策略 π_θ 的优势, R_t 为 t 时刻执行完动作 a_t 所获得的收益, 即单步奖励函数. $V_{\pi_\theta}(s_t)$ 表征在当前状态 s_t 条件下, 继续以策略 π_θ 控制系统运行, 整个过程累积得到的收益, 即剩余代价函数. $q_{\pi_\theta}(s_t,\,a_t)$ 表征在当前状态 s_t 采用动作 a_t, 转移至状态 s_{t+1}, 然后在后续过程中利用策略 π_θ 控制系统运行, 整个过程累积得到的收益.

$$\begin{cases}\hat{A}_t=q_{\pi_\theta}(s_t,\,a_t)-V_{\pi_\theta}(s_t)\\ V_{\pi_\theta}(s_t)=\mathrm{E}_{\pi_\theta}\left(\displaystyle\sum_{k=0}^{T-t-1}\gamma^kR_{t+k+1}\right)\\ q_{\pi_\theta}(s_t,\,a_t)=\mathrm{E}_{\pi_\theta}\left(\displaystyle\sum_{k=0}^{T-t-1}\gamma^kR_{t+k+1}\,\middle|\,s_t,\,a_t\right)\end{cases} \tag{13}$$

式 (11) 中 $L_t^{VF}(\theta)$ 为价值函数对应的损失函数, 其表示状态对应的真实价值函数与估计值的差值. 由于 s_t 对应的价值函数 $V_{\pi_\theta}(s_t)$ 未知, 一般采用神经网络对其估计, 如图 4 所示, 其可以表示为神经网络权值参数 θ 的函数, 即 $V_{\pi_\theta}(s_t)\approx\hat{V}_\pi^\theta(s_t)$. 对于某一条轨迹, 状态 s_t 处的真实价值函数可由下式估算得到

$$V_{\pi_\theta}(s_t)=\sum_{k=0}^{T-t-1}\gamma^kR_{t+k+1}\approx R_{t+1}+\gamma\hat{V}_\pi^\theta(s_{t+1}) \tag{14}$$

则价值函数的损失函数可表示为

$$L_t^{VF}(\theta)=\left(\sum_{k=0}^{T-t-1}\gamma^kR_{t+k+1}-\hat{V}_\pi^\theta(s_t)\right)^2 \tag{15}$$

$S[\pi_\theta](s_t)$ 为交叉熵, 表征策略的随机程度, 加入该项可以有效提升系统的探索能力. 对于离散动作 $a\in\{a_1,\,a_2,\,\cdots,\,a_{n_a}\}$, 交叉熵可表示为

$$S\left[\pi_\theta\right](s_t)=-\sum_{i=1}^{n_a}\pi_\theta\left(a_i\mid s_t\right)\ln\pi_\theta\left(a_i\mid s_t\right) \tag{16}$$

当 $\pi_\theta(a_1\mid s_t)=\pi_\theta(a_2\mid s_t)=\cdots=\pi_\theta(a_{n_a}\mid s_t)=1/n_a$ 时, $S[\pi_\theta](s_t)$ 达到极大值, 此时每个动作被选择的概率相同, 策略的不确定性最大.

本文将动作网络和价值网络融合为一个网络, 用式 (11) 的损失函数同时表征状态价值函数误差和动作代理优势, 神经网络的权值利用下式的梯度下降法进行更新

$$\theta\leftarrow\theta+\alpha_{lr}\frac{\partial L_t^{\text{Total}}(\theta)}{\partial\theta} \tag{17}$$

神经网络训练的伪代码如图 5 所示. 训练过程共包含 N_{episode} 个回合, 每个回合时长为 $T=dt\times N_{\text{step}}$, dt 为仿真步长, N_{step} 为每回合总步数. 在每个回合中, 首先对追踪星和逃逸星的位置在一定范围内随机初始化, 以此来提升网络对于不同初始位置的适应性. 然后将当前时刻两航天器的状态输入给图 4 所示的神经网络, 神经网络输出各个方向的推力时间, 经过式 (1) 的轨道动力学, 得到下一时刻的状态. 重复上述过程, 储存多条轨迹的相关数据, 直到存储的数据量达到训练需求, 利用梯度下降法更新网络参数.

2.3 奖励函数设计

1) 追踪星奖励函数

本文以轨道动力学模型为引导, 提出一种终端诱导的奖励函数设计方法, 预测追踪星和逃逸星在终端时刻的位置偏差, 并基于此设计奖励函数, 提

环境设置: 设置仿真步长、终端时刻、航天器总燃料、
推力上限、运动范围、训练数据长度 …

for i=1:N_{episode}
(1) 初始化: 随机初始化航天器位置和速度
for t=1:N_{step}
(2) 产生动作: 将当前航天器相对位置、速度输入给网络,
网络输出推力器开机时间;
(3) 状态更新: 将推力输入给轨道动力学,
得到下一时刻的状态和奖励;
(4) 状态收集: 将每一时刻的航天器相对位置、速度、奖励、动作存储;
if 数据长度n=mini_batch_n:
(5) 反向计算每一时刻的状态价值函数
$V(s_n)=\hat{V}_{\pi}^{\theta_{\text{old}}}(s_n)$
$V(s_{n-k})=\gamma^k V(s_n)+\sum_{l=0}^{k-1}\gamma^l R_{n-k+l}$
(6) 存储状态价值函数、相对位置、速度、奖励和动作;
else:
(7) 返回步骤 (2), 继续采集数据
end
if 轨迹数量=batch_n:
(8) 利用当前的网络参数 θ 替换旧的参数 θ_{old}
(9) 利用批数据更新网络参数, 如式 (30) 所示
else:
(10) 返回步骤 (1), 继续采集数据
end

图 5　PPO 算法训练伪代码

Fig. 5　Pseudo code for the PPO algorithm

升追踪星的预见性.

在当前时刻 t, 追踪星的奖励函数设计如下

$$R_p(t)=\begin{cases}\delta_{t_{\text{f}}}+0.95^{N_{\text{step}}-t}\left(\delta_{c1}-\alpha_c\left\|\hat{\boldsymbol{r}}_{pe}^{o}\left(t_{\text{f}}\right)\right\|-\beta_c\hat{\theta}_e\left(t_{\text{f}}\right)\right)-\\ 0.95^t\lambda_c\left\|\Delta\boldsymbol{V}_p(t)\right\|_1,\ \text{若}\left\|\hat{\boldsymbol{r}}_{pe}^{o}\left(t_{\text{f}}\right)\right\|<r_{\lim}\ \text{且}\ \theta_e<\theta_{\lim}\\ \delta_{c2}-\alpha_c\left\|\hat{\boldsymbol{r}}_{pe}^{o}\left(t_{\text{f}}\right)\right\|-\beta_c\hat{\theta}_e\left(t_{\text{f}}\right)-\\ 0.95^t\lambda_c\left\|\Delta\boldsymbol{V}_p(t)\right\|_1,\ \text{否则}\end{cases}$$

$$\delta_{t_{\text{f}}}=\begin{cases}30, & \text{若}\ t=t_{\text{f}},\ \left\|\hat{\boldsymbol{r}}_{pe}^{o}\left(t_{\text{f}}\right)\right\|<r_{\lim}\text{且}\ \theta_e<\theta_{\lim}\\ 0, & \text{否则}\end{cases}\tag{18}$$

其中, $\hat{\boldsymbol{r}}_{pe}^{o}\left(t_{\text{f}}\right)$ 为根据式 (4) 预测的无控条件下终端时刻两星的相对位置, $\hat{\theta}_e\left(t_{\text{f}}\right)$ 为由 $\hat{\boldsymbol{r}}_{pe}^{o}\left(t_{\text{f}}\right)$ 解算出的终端时刻两星连线与安全接近走廊中心线的夹角. $\Delta\boldsymbol{V}_p(t)$ 表示 t 时刻追踪星三轴速度脉冲. 采用分段的设计思路, 根据预测结果, 若在无控状态下追踪星在 t_{f} 时刻能够进入逃逸星的安全接近区域, 则赋予追踪星一个较大的常值奖励 δ_{c1}, 否则赋予其较小的常值奖励 δ_{c2}. δ_{c1} 和 δ_{c2} 的主要目的是通过持续赋予追踪星正值奖励, 使得追踪星保持学习进化的动力, 提高网络训练效率. $\delta_{t_{\text{f}}}$ 为终端奖励, 如果追踪星通过机动策略, 在 t_{f} 时刻进入到逃逸星的安全接近区域, 则赋予其较大的奖励.

2) 逃逸星奖励函数

$$R_e(t)=\begin{cases}\delta_{\lim}-\delta_{t_{\text{f}}}-0.95^t\lambda_t\left\|\Delta\boldsymbol{V}_e(t)\right\|_1-\\ 0.95^{N_{\text{step}}-t}\left(\delta_{t1}-\alpha_t\left\|\hat{\boldsymbol{r}}_{pe}^{o}\left(t_{\text{f}}\right)\right\|-\beta_t\hat{\theta}_e\left(t_{\text{f}}\right)\right),\\ \text{若}\left\|\hat{\boldsymbol{r}}_{pe}^{o}\left(t_{\text{f}}\right)\right\|<r_{\lim}\ \text{且}\ \theta_e<\theta_{\lim}\\ \delta_{\lim}-\left(\delta_{t2}-\alpha_t\left\|\hat{\boldsymbol{r}}_{pe}^{o}\left(t_{\text{f}}\right)\right\|-\beta_t\hat{\theta}_e\left(t_{\text{f}}\right)\right)-\\ 0.95^t\lambda_t\left\|\Delta\boldsymbol{V}_e(t)\right\|_1,\ \text{否则}\end{cases}$$

$$\delta_{t_{\text{f}}}=\begin{cases}30, & \text{若}\ t=t_{\text{f}},\ \left\|\hat{\boldsymbol{r}}_{pe}^{o}\left(t_{\text{f}}\right)\right\|<r_{\lim}\ \text{且}\ \theta_e<\theta_{\lim}\\ 0, & \text{否则}\end{cases}$$

$$\delta_{\lim}=\begin{cases}-5, & \text{若}\ \left\|\boldsymbol{r}_{eo}(t)-\boldsymbol{r}_{eo}(t_0)\right\|>r_{e\_\lim}\\ 0, & \text{否则}\end{cases}\tag{19}$$

由于逃逸星的目的与追踪星相反, 因此在设计奖励函数时应直接对追踪星奖励中的相对距离和角度项符号取反, 同时加入对逃逸星运动范围的限制项 $\delta_{\lim}$.

3　仿真分析

本节将针对追逃博弈问题, 采用 PPO 算法同时对追踪星和逃逸星进行训练, 构造相互博弈的动态仿真场景, 给出两星在训练过程中各回合的收益、相对距离、燃料消耗等变化曲线, 对比不同奖励函数和不同算法对训练收敛情况的影响. 当两星完成训练后, 决策网络的权值不再变化, 进行打靶验证.

3.1　仿真参数设置

本文的博弈场景选择在 GEO 附近, 轨道半径为 36 000 km, 周期为 86 400 s. 两星的轨道运动采用二体动力学模型驱动. 参考坐标系 (CW 系) 的原点选择在 GEO 轨道上, 星下点经纬度为 (100°E, 0°N), 两星的相对运动采用式 (1) 的动力学描述. 在仿真中, 逃逸星初始位置与 CW 参考系原点重合, 相对 CW 参考系速度为零. 综合考虑追踪星燃料上限和终端时刻约束, 为了使追踪星具备在终端时刻抵近逃逸星的能力, 设置追踪星的初始位置为距离逃逸星 100 km ~ 150 km 的球壳内. 仿真步长 6 000 s, 每回合步数为 60 步, 终端时刻 $t_{\text{f}}=36\,000$ s.

两星的质量、最大推力、决策周期等参数如表 1 所示. 逃逸星安全接近区的半锥角 $\theta_{\lim}=30°$, 母线长度 $r_{\lim}=50$ km. 博弈输赢判定条件为: 如果追踪星在终端时刻进入逃逸星的安全接近区, 则追踪星胜利, 否则逃逸星胜利.

训练过程依托 2 块 Intel Xeon 6240 CPU 以及 1 块 Tesla V100 GPU 等硬件资源支撑.

训练采用 3 隐层全连接网络, 输入层包含自身绝对位置、绝对速度、相对位置、相对速度、燃料消耗共计 13 维变量; 各隐层节点数分别为 256、128、128. 输出层包含航天器三轴推力时间以及状态价值函数, 本文将航天器各轴推力离散为 11 个变量, 因此, 输出层共有 34 个神经元. 各隐层神经元的激活函数选择为 ReLU 函数, 输出层的激活函数选择双曲正切函数. 神经网络结构详见图 4.

PPO 训练算法参数以及奖励函数相关参数在表 2 中给出.

3.2 训练过程展示

为了说明本文提出的终端诱导奖励函数设计方法相比于传统的基于当前时刻误差设计奖励函数方法的优势, 进行仿真对比. 为了对比的公平性, 基于两种方法设计的奖励函数的逻辑与结构相同 (如式 (18) 和式 (19) 所示), 只是将相对位置变量进行变换. 本文的奖励函数中使用相对位置的终端预测值 $\hat{\boldsymbol{r}}_{pe}^{o}(t_{\mathrm{f}})$, 而传统的奖励函数中使用相对位置的当前时刻值 $\boldsymbol{r}_{pe}^{o}(t)$. 另一方面, 为了验证本文 PPO 算法相较于其他学习算法的优势, 与强化学习中的 SAC (Soft actor critic) 算法进行对比.

在 Python 仿真环境中对追踪星和逃逸星进行训练, 训练过程曲线如图 6 ~ 图 9 所示. 从图 6 和图 7 可以看出, SAC 算法在训练 10 000 回合后收敛, 而采用终端诱导奖励的 PPO 算法训练 600 回合即可收敛. SAC 算法的探索能力驱使其持续探索新的策略, 导致收益出现剧烈抖动. 采用传统即时奖励的 PPO 算法在 600 回合后陷入局部极小值, 未探索到进入安全接近走廊的策略, 直到 5 000 回合后, 追踪星探索到新的策略, 使得追踪星在终端时刻进入到逃逸星的安全接近走廊, 其得到较大的终端奖励如式 (18) 所示, 跳出了局部极小值, 收敛到全局最优解.

图 8 为每个回合的终端时刻 $t = 36\,000$ s 追踪星和逃逸星相对距离, 在本文设计的奖励函数引导下, 追踪星在训练 600 回合后, 能够在终端时刻抵近至距离逃逸星 50 km 以内, 同时, 式 (6) 定义的两星连线与安全接近区域中线夹角达到 30° 附近, 如图 9 所示.

而采用传统的基于当前误差设计奖励函数在 600 回合后陷入局部极小值, 其与逃逸星的终端距离和角度分别保持在 60 km 和 40°, 直到 5 000 回合后才成功进入逃逸星的安全接近走廊. 虽然采用终端诱导奖励和传统即时奖励最终都能使追踪星学到抵近策略, 但是终端诱导奖励的训练效率明显高于即时奖励, 这是因为即时奖励没有利用轨道动力学特性. SAC 算法只能训练追踪星抵近到距离逃逸星 75 km 处, 之后由于逃逸星探索到更优的逃逸策略, 导致追踪星难以进一步抵近.

3.3 打靶验证

追踪星和逃逸星完成网络训练后, 网络参数保持不变, 将两星的相对位置和速度输入给网络, 网络输出相应的推力时间.

采用第 10 000 局训练得到的神经网络进行追逃决策. 进行 200 局轨道博弈, 追踪星相对逃逸星的距离及角度随时间变化曲线如图 10 所示. 在本文设计的奖励函数引导下, 追踪星不会过早进入逃逸星 50 km 的安全接近区域, 以此节省燃料消耗. 由于逃逸星通过训练也提升了逃逸能力, 因此有一定的机率避免追踪星抵近, 最终追踪星的追击成功率为 97%.

选择其中一局的结果进行展示, 如图 11 所示.

表 1 追踪星和逃逸星参数设置

Table 1 Parameters of the pursuer and evader

博弈对象	决策周期 (s)	各轴推质比 (N/kg)	各轴单次速度增量上限 (m/s)	总速度增量上限 (m/s)
追踪星	600	20/500	1.6	320
逃逸星	600	20/500	1.6	240

表 2 PPO 算法相关参数

Table 2 Parameters for the PPO algorithm

参数描述	参数数值
学习率	$\alpha_{lr} = 0.0002$
损失函数相关参数	$\varepsilon = 0.1,\ c_1 = 0.5,\ c_2 = 0.01$
训练所需轨迹条数	$\mathrm{Batch\_n} = 128$
追踪星奖励函数参数	$\delta_{c1} = 1$, $\delta_{c2} = 0.1$, $\alpha_c = 4 \times 10^{-6}$, $\beta_c = 1/\pi$, $\lambda_c = 0.5$
逃逸星奖励函数参数	$\delta_{t1} = 1$, $\delta_{t2} = 0.1$, $\alpha_t = 4 \times 10^{-6}$, $\beta_t = 1/\pi$, $\lambda_t = 5/6$

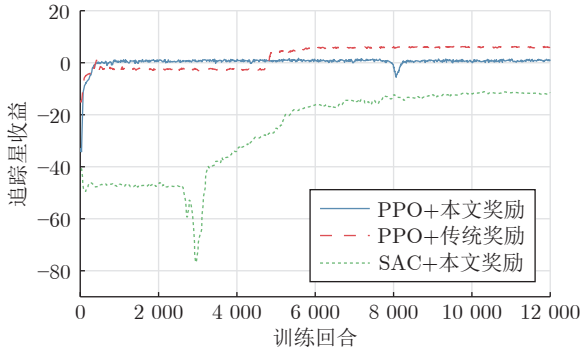


图 6　追踪星训练过程收益曲线

Fig. 6　Pursuer total reward during the training

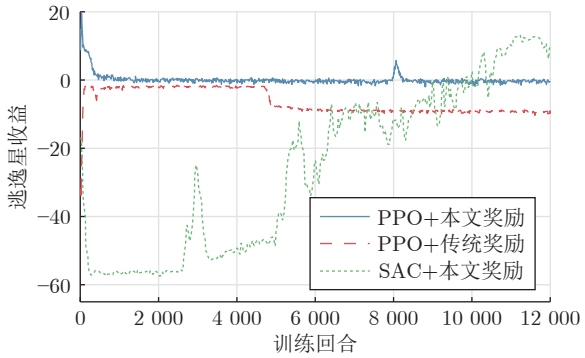


图 7　逃逸星训练过程收益曲线

Fig. 7　Evader total reward during the training

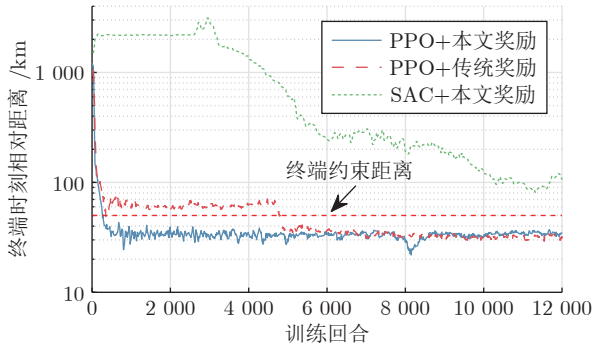


图 8　训练过程中两星终端相对距离

Fig. 8　The final distance between pursuer and evader during the training

追踪星和逃逸星不断进行轨道博弈, 追踪星在仿真结束时成功进入到逃逸星的安全接近区域.

如果采用传统设计奖励函数的方法训练追踪星和逃逸星, 打靶结果如图 12 所示, 追踪星在终端时刻抵近至逃逸星的安全接近区的成功率为 92%. 对比图 10 和图 12 可以发现, 基于传统方法设计的奖励函数引导追踪星较早进入逃逸星的安全接近区域.

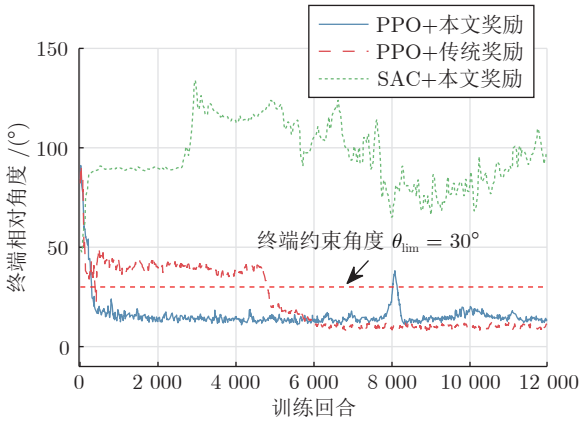
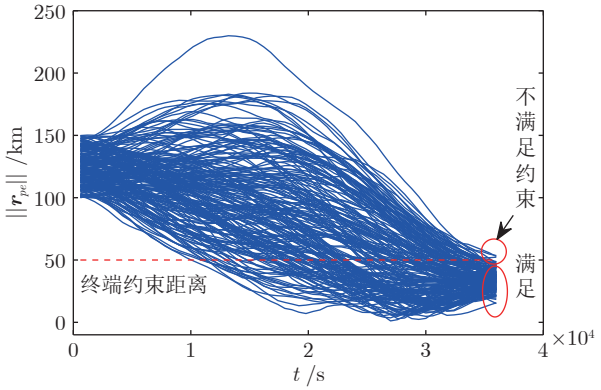


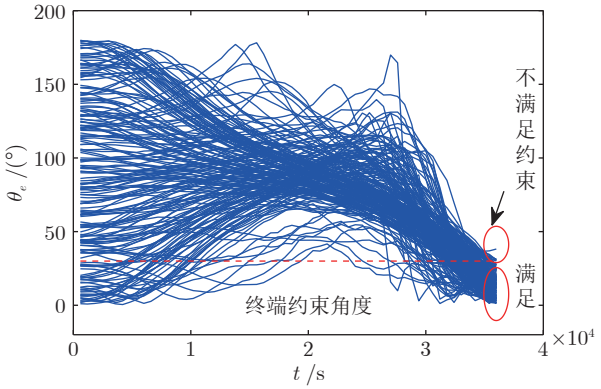
图 9　训练过程中终端时刻两星与安全接近区域中心线夹角

Fig. 9　The final angle between the pursuer-evader direction and the conic center axis during the training



(a) 追踪星和逃逸星的相对距离

(a) The distance between pursuer and evader



(b) 追踪星－逃逸星连线与安全接近区中心线夹角

(b) The angle between the direction of pursuer-evader and the center axis of the conic region

图 10　基于本文奖励函数设计方法的多局打靶结果

Fig. 10　The Monte-Carlo results for the reward function design method in this paper

博弈均衡点是博弈系统的稳定平衡点, 当双方达到均衡解时, 任意一方无法通过只改变自身策略

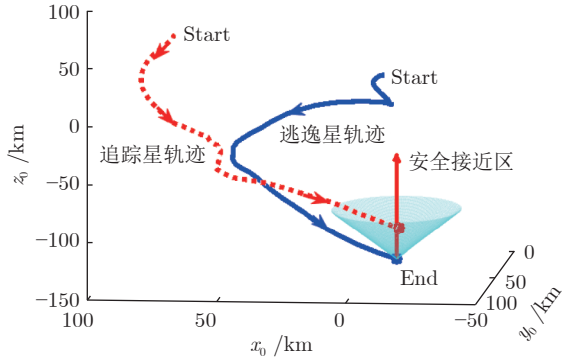
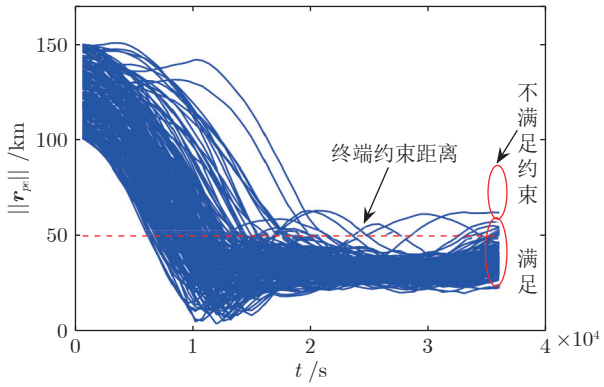


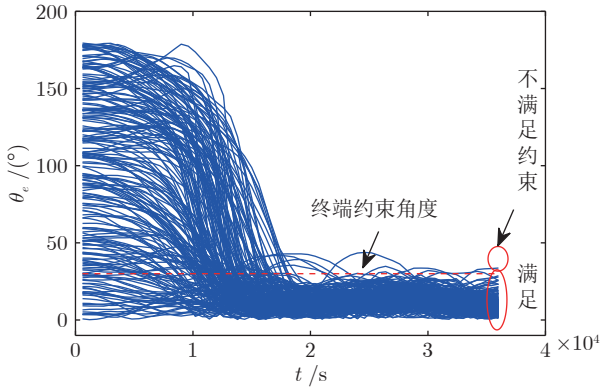
图 11 本文奖励函数对应的一局轨迹

Fig. 11 The trajectory in one episode based on the reward function designed in this paper



(a) 追踪星和逃逸星的相对距离

(a) The distance between pursuer and evader



(b) 追踪星－逃逸星连线与安全接近区中心线夹角

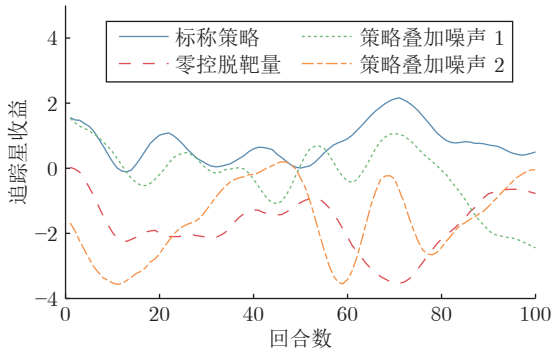
(b) The angle between the direction of pursuer-evader and the center axis of the conic region

图 12 基于传统奖励函数设计方法的多局打靶结果

Fig. 12 The Monte-Carlo results for the traditional reward function design method

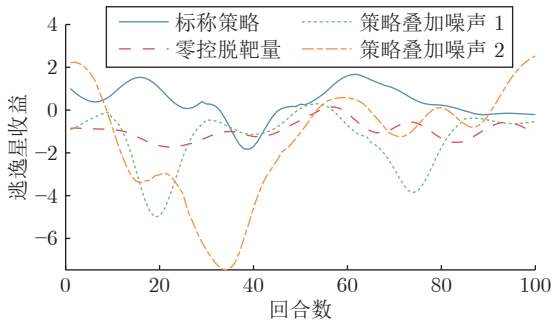
而提高收益. 从两星的训练曲线可以看出, 在 10000 回合过后两星的收益收敛到稳定状态, 如图 6 和图 7 所示.

选择第 10 000 回合的神经网络模型作为追踪星和逃逸星的标称决策策略, 进行 100 回合本地打靶, 每回合双方的收益如图 13 所示; 然后, 令逃逸星仍采用第 10000 回合的网络模型, 而追踪星改变策略, 通过对追踪星的输出动作添加不同幅度的噪声, 以及采用基于零控脱靶量的追踪策略, 使其决策策略偏离第 10000 回合的训练结果, 进行 100 回合打靶. 可以看出, 追踪星的收益都低于采用第 10000 回合的打靶结果; 类似地, 如图 13(b) 所示, 令追踪星采用第 10000 回合的网络模型进行决策, 而逃逸星改变逃逸策略, 可见逃逸星的收益要低于采用第 10000 回合的打靶结果, 从而说明了系统运行于博弈均衡解.



(a) 追踪星不同策略的收益

(a) Pursuer reward with different strategies



(b) 逃逸星不同策略的收益

(b) Evader reward with different strategies

图 13 博弈均衡态验证结果

Fig. 13 The results for verification of the saddle point

为进一步说明本文算法的优势, 将其分别与基于传统奖励的 PPO 算法、SAC 算法以及基于零控脱靶量[32] 的追逃算法进行打靶成功率对比. 试验结果如表 3 所示. 本文设计的奖励函数可驱使追踪星以更高的概率完成对采用不同策略的逃逸星的抵近任务, 从而说明本文学习算法和奖励函数的智能性和适应性. 传统即时奖励也能使得追踪星取得较高成功率, 但是如前文所述, 其训练效率低. 而 SAC 算法由于探索空间过大, 收敛性较差, 在轨道博弈任务中成功率较低.

表 3　不同追逃策略的追踪成功率

Table 3　Success rate of the pursuer for different pursuing and evasion strategies

逃逸星	追踪星			
	PPO+本文奖励	PPO+传统奖励	SAC+本文奖励	零控脱靶量法
PPO+本文奖励	97%	89%	0%	92%
PPO+传统奖励	99%	92%	2%	98%
SAC+本文奖励	100%	61%	7%	100%
零控脱靶量法	99%	99%	9%	98%

4　结论

本文针对脉冲推力控制下的轨道追逃博弈问题, 充分考虑了实际任务需求及约束, 建立了追踪星和逃逸星的追逃轨道动力学模型及终端约束模型. 在此基础上, 针对强博弈态势, 提出一种终端诱导的奖励函数设计方法, 通过预测双方在终端时刻的相对位置, 并将其引入奖励函数, 克服了现有方法在解决指定终端时刻博弈问题的不足. 同时, 通过轨迹预测, 将动力学知识引入神经网络, 有效提升了训练效率和对抗成功率. 通过仿真对比, 本文提出的奖励函数能够快速引导追踪星学习到追踪策略, 使追踪星在终端时刻进入到逃逸星的安全接近区, 而传统的奖励函数由于没有利用轨道动力学知识, 学习效率较低.

References

1 Yuan Li. Spacecraft intelligent autonomous control technology toward uncertain environment. *Journal of Astronautics*, 2021, **42**(7): 839−849
(袁利. 面向不确定环境的航天器智能自主控制技术. 宇航学报, 2021, **42**(7): 839−849)

2 Zhao Li-Ran, Dang Zhao-Hui, Zhang Yu-Lin. Orbital game: Concepts, principles and methods. *Journal of Command and Control*, 2021, **7**(3): 215−224
(赵力冉, 党朝辉, 张育林. 空间轨道博弈: 概念, 原理与方法. 指挥与控制学报, 2021, **7**(3): 215−224)

3 Isaacs R. *Differential Games: A Mathematical Theory With Applications to Warfare and Pursuit, Control and Optimization*. London: John Wiley, 1999.

4 Chertovskih R, Karamzin D, Khalil N T, Pereira F L. An indirect method for regular state-constrained optimal control problems in flow fields. *IEEE Transactions on Automatic Control*, 2020, **66**(2): 787−793

5 Hao Zhi-Wei, Sun Song-Tao, Zhang Qiu-Hua, Chen Ying. Application of semi-direct collocation method for solving pursuit-evasion problems of spacecraft. *Journal of Astronautics*, 2019, **40**(6): 628−635
(郝志伟, 孙松涛, 张秋华, 谌颖. 半直接配点法在航天器追逃问题求解中的应用. 宇航学报, 2019, **40**(6): 628−635)

6 Li Y, Chen W C, Yang L. Multistage linear Gauss pseudospectral method for piecewise continuous nonlinear optimal control problems. *IEEE Transactions on Aerospace and Electronic Systems*, 2021, **57**(4): 2298−2310

7 Luo Ya-Zhong, Li Zhen-Yu, Zhu Hai. Survey on spacecraft orbital pursuit-evasion differential games. *Scientia Sinica Technologica*, 2020, **50**(12): 1533−1545
(罗亚中, 李振瑜, 祝海. 航天器轨道追逃微分对策研究综述. 中国科学: 技术科学, 2020, **50**(12): 1533−1545)

8 Hafer W T, Reed H L, Turner J D, Pham K. Sensitivity methods applied to orbital pursuit evasion. *Journal of Guidance, Control, and Dynamics*, 2015, **38**(6): 1118−1126

9 Oyler D W, Kabamba P T, Girard A R. Pursuit-evasion games in the presence of obstacles. *Automatica*, 2016, **65**: 1−11

10 Shen H X, Casalino L. Revisit of the three-dimensional orbital pursuit-evasion game. *Journal of Guidance, Control, and Dynamics*, 2018, **41**(8): 1823−1831

11 Li Z, Zhu H, Yang Z, Luo Y Z. Saddle point of orbital pursuit-evasion game under J2-perturbed dynamics. *Journal of Guidance, Control, and Dynamics*, 2020, **43**(9): 1733−1739

12 Perelman A, Shima T, Rusnak I. Cooperative differential games strategies for active aircraft protection from a homing missile. *Journal of Guidance, Control, and Dynamics*, 2011, **34**(3): 761−773

13 Liu Kun, Zheng Xiao-Shuai, Lin Ye-Ming, Han Le, Xia Yuan-Qing. Design of optimal strategies for the pursuit-evasion problem based on differential game. *Acta Automatica Sinica*, 2021, **47**(8): 1840−1854
(刘坤, 郑晓帅, 林业茗, 韩乐, 夏元清. 基于微分博弈的追逃问题最优策略设计. 自动化学报, 2021, **47**(8): 1840−1854)

14 Venigalla C, Scheeres D J. Delta-V-based analysis of spacecraft pursuit-evasion games. *Journal of Guidance, Control, and Dynamics*, 2021, **44**(11): 1961−1971

15 Yu Da-Teng. Approaches for the Spacecraft Security Defense and Evasion Maneuver Method [Ph.D. dissertation], National University of Defense Technology, China, 2017
(于大腾. 空间飞行器安全防护规避机动方法研究 [博士论文], 国防科技大学, 中国, 2017)

16 Li Zhen, Fan Jia-Lu, Jiang Yi, Chai Tian-You. A model-free H_∞ control method based on off-policy with output data feedback. *Acta Automatica Sinica*, 2021, **47**(9): 2182−2193
(李臻, 范家璐, 姜艺, 柴天佑. 一种基于 Off-Policy 的无模型输出数据反馈 H_∞ 控制方法. 自动化学报, 2021, **47**(9): 2182−2193)

17 Schulman J, Levine S, Moritz P, Jordan M I, Abbeel P. Trust region policy optimization. In: Proceedings of the International Conference on Machine Learning. Guangzhou, China: International Machine Learning Society (IMLS), 2015. 1889−1897

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

19 Han M H, Zhang L X, Wang J, Pan W. Actor-critic reinforcement learning for control with stability guarantee. *IEEE Robotics and Automation Letters*, 2020, **5**(4): 6217−6224

20 Izzo D, Simoes L F, de Croon G C H E. An evolutionary robotics approach for the distributed control of satellite formations. *Evolutionary Intelligence*, 2014, **7**(2): 107−118

21 Shirobokov M, Trofimov S, Ovchinnikov M. Survey of machine learning techniques in spacecraft control design. *Acta Astronautica*, 2021, **186**: 87−97

22 Gaudet B, Linares R, Furfaro R. Adaptive guidance and integrated navigation with reinforcement meta-learning. *Acta Astronautica*, 2020, **169**: 180−190

23 Gaudet B, Linares R, Furfaro R. Deep reinforcement learning for six degree-of-freedom planetary landing. *Advances in Space Research*, 2020, **65**(7): 1723−1741

24 Hovell K, Ulrich S. Deep reinforcement learning for spacecraft proximity operations guidance. *Journal of Spacecraft and Rockets*, 2021, **58**(2): 254−264

25 Zavoli A, Federici L. Reinforcement learning for robust trajectory design of interplanetary missions. *Journal of Guidance, Control, and Dynamics*, 2021, **44**(8): 1440−1453

26 Koch W, Mancuso R, West R, Bestavros A. Reinforcement learning for UAV attitude control. *ACM Transactions on Cyber-Physical Systems*, 2019, **3**(2): 1−21

27 Furfaro R, Scorsoglio A, Linares R, Massari M. Adaptive generalized ZEM-ZEV feedback guidance for planetary landing via a deep reinforcement learning approach. *Acta Astronautica*, 2020, **171**: 156−171

28 Wang X, Shi P, Wen C X, Zhao Y S. Design of parameter-self-tuning controller based on reinforcement learning for tracking noncooperative targets in space. *IEEE Transactions on Aerospace and Electronic Systems*, 2020, **56**(6): 4192−4208

29 Shi Wei, Feng Yang-He, Cheng Guang-Quan, Huang Hong-Lan, Huang Jin-Cai, Liu Zhong, et al. Research on multi-aircraft cooperative air combat method based on deep reinforcement learning. *Acta Automatica Sinica*, 2021, **47**(7): 1610−1623
(施伟, 冯旸赫, 程光权, 黄红蓝, 黄金才, 刘忠, 等. 基于深度强化学习的多机协同空战方法研究. 自动化学报, 2021, **47**(7): 1610−1623)

30 Zhao Yu, Guo Ji-Feng, Yan Peng, Bai Cheng-Chao. Self-learning-based multiple spacecraft evasion decision making simulation under sparse reward condition. *Journal of System Simulation*, 2021, **33**(8): 1766−1774
(赵毓, 郭继峰, 颜鹏, 白成超. 稀疏奖励下多航天器规避决策自学习仿真. 系统仿真学报, 2021, **33**(8): 1766−1774)

31 Liu Bing-Yan, Ye Xiong-Bing, Gao Yong, Wang Xin-Bo, Ni Lei. Strategy solution of non-coorerative target pursuit-evasion game based on branching deep reinforcement learning. *Acta Aeronautica et Astronautica Sinica*, 2020, **41**(10): 348−358
(刘冰雁, 叶雄兵, 高勇, 王新波, 倪蕾. 基于分支深度强化学习的非合作目标追逃博弈策略求解. 航空学报, 2020, **41**(10): 348−358)

32 Wang Chun-Bao, Ye Dong, Sun Zhao-Wei, Sun Chu-Qi. Adaptive game strategy of spacecraft terminal interception. *Journal of Astronautics*, 2020, **41**(3): 309−318
(王淳宝, 叶东, 孙兆伟, 孙楚琦. 航天器末端拦截自适应博弈策略. 宇航学报, 2020, **41**(3): 309−318)

耿远卓 北京控制工程研究所工程师. 主要研究方向为航天器轨迹规划与控制.
E-mail: gengyz_hit@163.com
(**GENG Yuan-Zhuo** Engineer at Beijing Institute of Control Engineering. His research interest covers spacecraft guidance and control.)

袁 利 中国空间技术研究院研究员. 主要研究方向为航天器自主控制, 鲁棒容错控制技术. 本文通信作者.
E-mail: yuanli@spacechina.com
(**YUAN Li** Researcher at China Academy of Space Technology. His research interest covers spacecraft autonomous control and robust fault-tolerant control. Corresponding author of this paper.)

黄 煌 北京控制工程研究所研究员. 主要研究方向为航天器智能决策与控制. E-mail: hhuang33@163.com
(**HUANG Huang** Researcher at Beijing Institute of Control Engineering. Her research interest covers spacecraft intelligent guidance and control.)

汤 亮 北京控制工程研究所研究员. 主要研究方向为航天器动力学与控制. E-mail: tl614@sina.com
(**TANG Liang** Researcher at Beijing Institute of Control Engineering. His research interest covers spacecraft dynamics and control.)