基于强化学习的航天器姿态控制器设计

张瑞卿[1],钟　睿[1],徐　毅[2]
(1.北京航空航天大学 宇航学院,北京 102206;2.上海卫星工程研究所,上海 201109)

摘　要: 航天器在轨执行某些任务时,其质量参数会发生未知变化,传统控制方法在这种情况下控制效果不佳。本文提出基于强化学习的航天器姿态控制器设计方法,该方法在姿态控制器训练过程中不需要对航天器进行动力学建模,不依赖航天器的质量参数。当质量参数发生较大未知变化时,训练好的控制器仍然可以保持较好的控制效果。仿真测试表明:使用基于强化学习方法训练的控制器确实具有良好的鲁棒性。此外,回报函数的设计会明显影响姿态控制器的训练,因此对不同的回报函数设计进行了研究。

关键词: 航天器姿态控制; 鲁棒性; 强化学习; 神经网络; 回报函数
中图分类号: V 448.22^{+}3　　**文献标志码:** A　　**DOI:** 10.19328/j.cnki.2096-8655.2023.01.010

Satellite Attitude Control Based on Reinforcement Learning Method

ZHANG Ruiqing[1], ZHONG Rui[1], XU Yi[2]
(1.School of Astronautics, Beihang University, Beijing 102206, China;
2.Shanghai Institute of Satellite Engineering, Shanghai 201109, China)

Abstract: Owing to the growing complexity of space mission, classical control methods cannot meet the increasing high requirements for the robustness and adaptiveness of the satellite attitude control system. In this paper, a design method for the satellite attitude control system is proposed based on the reinforcement learning (RL) method. With the proposed method, it is not necessary to establish a dynamic model for the spacecraft in the training process of the attitude controller, and the satellite attitude control system is independent of the spacecraft mass parameters. Besides, when the mass parameters change, the trained controller can still maintain a good control effect. The test results show that the control system trained by the RL method has a stronger adaptive capability. In addition, since the design of the return function will significantly affect the training effect, different return function designs are also studied.

Key words: attitude control; robustness; reinforcement learning; neural network; reward function

0　引　言

航天器姿态控制系统是航天器系统中关键的分系统,姿态控制的效果直接影响到航天器有效载荷能否在轨正常工作,如低轨气象卫星需要通过姿态控制系统实现对月定标来完成任务[1]。传统广泛使用的比例-积分-微分(Proportion Integration Differentiation,PID)控制在设计控制器过程中,需要掌握准确的航天器质量参数。但航天器姿态动力学系统十分复杂,模型高度非线性,当航天器的质量参数发生较大改变且无法准备预测时(如捕获非合作对象[2]、燃料长期消耗[3]),PID控制设计的控制器会出现控制效果不佳,甚至是失效的情况[4]。此外,太空环境还存在很多不确定因素[5],这些都要求设计具备良好鲁棒性和自适应能力的姿态控制器。

传统强化学习方法,如Q-Learning算法,只能解决小规模、离散空间问题,并没有得到广泛的使用[6]。近年来,深度学习的研究得到快速发展,研究者们也尝试将深度学习和传统强化学习方法结合起来进行研究,进而研究出了很多著名的算法[7],如深度Q学

收稿日期: 2021-05-20; **修回日期:** 2021-08-19
基金项目: 国家自然科学基金(11772023);上海航天科技创新基金(SAST2019-040)
作者简介: 张瑞卿(1997—),男,硕士研究生,主要研究方向为航天器姿态智能控制系统。
通信作者: 钟　睿(1984—),男,博士,副教授,主要研究方向为绳系卫星动力学与控制、智能控制。

习算法(Deep Q-Network，DQN)[8]、深度确定性策略梯度算法(Deep Deterministic Policy Gradient，DDPG)[9]。其中，DDPG算法由于其状态空间和动作空间连续，被广泛应用于连续控制领域[10-13]。

将深度强化学习应用到控制领域时，通常需要根据控制系统的特点设计回报函数[14]。对于航天器的姿态镇定问题，如果认为只在进入精度范围内时获得奖励，那么当训练步数越多时，获得奖励的概率就越小，最终导致训练失败，这被称为稀疏回报问题[15]。吴恩达[16]提出了回报塑造概念，通过人为设计辅助回报函数引导算法收敛，可以减少训练时间，提升训练效果。在连续控制领域中，通常会设计与距离相关的辅助回报函数进行引导[17]。

拟使用DDPG算法对航天器姿态控制器进行设计。在设计过程中，首先，建立深度强化学习方法训练控制器所需的环境，设定回报函数，搭建基于Actor-Critic的神经网络框架；然后，使用DDPG算法对姿态控制器进行训练，迭代若干回合完成对姿态控制器的训练。

1　航天器姿态动力学建模

为了描述航天器姿态，规定参考坐标系为轨道坐标系 $ox_0y_0z_0$。在轨道坐标系 $ox_0y_0z_0$ 中，原点为航天器的质心 o，z_0 轴指向地心，x_0 轴指向轨道速度方向且与 z_0 轴方向垂直，y_0 轴与 x_0 轴、z_0 轴垂直且共同构成右手直角坐标系。采用由轨道坐标系($ox_0y_0z_0$)按 z、x、y 的顺序旋转到本体坐标系($ox_by_bz_b$)的欧拉角来描述航天器姿态，使用 ψ、φ、θ 分别表示偏航角、俯仰角、滚转角。

航天器姿态动力学方程为

$$\boldsymbol{I}(\mathrm{d}\boldsymbol{\omega}/\mathrm{d}t)+\boldsymbol{\omega}(\boldsymbol{I}\cdot\boldsymbol{\omega})=\boldsymbol{M} \tag{1}$$

式中：$\boldsymbol{I}$ 为航天器转动惯量；$\boldsymbol{\omega}$ 为姿态角速度；$\boldsymbol{M}$ 为作用在航天器上的力矩。

将此方程投影到航天器本体主轴坐标系中：

$$\begin{cases} I_x\dot{\omega}_x-(I_y-I_z)\omega_y\omega_z=M_x \\ I_y\dot{\omega}_y-(I_z-I_x)\omega_z\omega_x=M_y \\ I_z\dot{\omega}_z-(I_x-I_y)\omega_x\omega_y=M_z \end{cases} \tag{2}$$

式中：I_x、I_y、I_z 为航天器投影到本体系中的转动惯量；ω_x、ω_y、ω_z 为航天器投影到本体系的角速度；M_x、M_y、M_z 为航天器受到的力矩投影到本体系上的分量。

若只考虑航天器姿态镇定控制问题时，航天器姿态的欧拉角都是小量。考虑航天器绕地球旋转的轨道角速度，将姿态运动学方程代入式(2)后，可进一步将姿态动力学方程线化为线性常系数微分方程，即

$$\begin{cases} I_x\ddot{\varphi}+(I_y-I_z)\Omega^2\varphi-(I_x-I_y+I_z)\Omega\dot{\psi}=T_{cx}+T_{dgx} \\ I_y\ddot{\theta}=T_{cy}+T_{dgy} \\ I_z\ddot{\psi}+(I_x-I_y)\Omega^2\psi-(I_x-I_y+I_z)\Omega\dot{\varphi}=T_{cz} \end{cases} \tag{3}$$

式中：Ω 为轨道角速度；T_{cx}、T_{cy}、T_{cz} 为控制力矩；$\dot{\psi}$、$\dot{\varphi}$ 分别为 ψ、φ 的一阶导数；$\ddot{\varphi}$、$\ddot{\psi}$、$\ddot{\theta}$ 分别是 ψ、φ、θ 的二阶导数。

考虑重力梯度力矩，当卫星在小姿态角的情况下，投影到主坐标系下的重力梯度力矩为 T_{dgx} 和 T_{dgy}，其表达式为

$$\begin{cases} T_{dgx}=-3\Omega^2(I_y-I_z)\varphi \\ T_{dgy}=-3\Omega^2(I_x-I_z)\varphi \end{cases} \tag{4}$$

2　基于DDPG的航天器姿态控制器训练

2.1　DDPG算法原理

DDPG算法是一种基于Actor-Critic框架的算法。基于Actor-Critic框架的强化学习算法将值函数逼近的方法和策略逼近的方法结合在一起，使用策略逼近的思想来设计Actor，让Actor进行动作选择，保证了动作的连续性；而使用值函数逼近的思想设计Critic，Critic告诉Actor选择的动作是否合适，由于基于值函数逼近的方法可以做到单步更新，因此也提高了学习效率。在Actor和Critic交互过程中，Actor不断迭代，得到每一个状态下选择每一动作的合理概率，Critic也不断迭代，不断完善每个状态下选择每一个动作的奖惩值。

DDPG算法在Actor-Critic框架的基础上，将值函数逼近和策略函数逼近结合的同时，应用了DQN算法记忆库和冻结目标网络的方法，做到了动作空间和状态空间连续，也提高了学习效率。

DDPG算法在选择动作时，采用确定性策略 μ，即输出概率最大的动作，然后也采用了参数噪声 N 来增加对环境的探索：

$$a=\mu(s|\theta^{\mu})+N \tag{5}$$

式中：a 为实际得到的动作；$\mu(s_t|\theta^{\mu})$ 为神经网络参数 θ^{μ} 在状态 s 下根据确定性策略 μ 得到的动作。

可将DDPG算法的目标函数 $J(\theta^{\mu})$ 表示为

$$J(\theta^{\mu})=E_{\theta^{\mu}}\left[r_1+\gamma r_2+\gamma^2 r_3+\cdots+\gamma^{i-1}r_i\right] \tag{6}$$

式中：γ 为Agent短视的程度，也就是回报的衰减程度；r_i 为第 i 步的奖励；$E(\cdot)$ 为数学期望。

可以证明在采用确定性策略μ的DDPG算法中,目标函数$J(\theta^{\mu})$的梯度与动作值函数Q的期望梯度相等,故Actor网络的梯度为

$$\frac{\partial J(\theta^{\mu})}{\partial \theta^{\mu}}=E_{s}\left[\frac{\partial Q(s,a|\theta^{Q})}{\partial a}\frac{\partial \pi(s|\theta^{\mu})}{\partial \theta^{\mu}}\right] \tag{7}$$

$$\nabla_{\theta}J_{\beta}(\mu_{\theta})=\int_{S}\rho^{\beta}(s)\nabla_{\theta}\mu_{\theta}(s)Q^{\mu}(s,a)\big|_{a=\mu_{\theta}}\mathrm{d}s=E_{s\sim\rho^{\beta}}\left[\nabla_{\theta}\mu_{\theta}(s)Q^{\mu}(s,a)\big|_{a=\mu_{\theta}}\right] \tag{8}$$

式中:∇_{θ}为网络梯度;μ_{θ}为网络选择的策略。

而Critic网络的梯度为

$$\frac{\partial L(\theta^{Q})}{\partial \theta^{Q}}=E_{s,a,r,s'\sim D}\left[Q_{\text{target}}-Q(s,a|\theta^{Q})\frac{\partial Q(s,a|\theta^{Q})}{\partial \theta^{Q}}\right] \tag{9}$$

式中:$Q_{\text{target}}=r+\gamma Q'\left(s',\pi(s'|\theta^{\mu})|\theta^{Q}\right)$;$\theta^{Q}$为Critic网络的神经网络参数。

根据式(9)、式(10),可以对Actor网络、Critic网络的网络参数进行更新[9]。

2.2 训练流程

使用DDPG算法对姿态控制器进行训练时,首先建立航天器姿态动力学环境,并对姿态控制器随机进行初始化;然后姿态控制器根据当前姿态角和姿态角速度输出控制力矩,在控制力矩作用下航天器姿态角和姿态角速度发生改变,设置的回报函数会根据变化后的状态给出回报,算法将当前时刻的状态s_t姿态控制器输出的控制力矩a_t、回报r_t和下一个时刻的状态s_{t+1}生成样本(s_t,a_t,r_t,s_{t+1}),并存放在缓存区R中,之后从缓存区中随机抽取样本,对控制器进行训练,调整神经网络的参数,迭代若干次之后便可完成对姿态控制器的训练。具体训练流程如下。

步骤1 随机初始化Critic网络$Q(s,a|\theta^{Q})$和Actor网络$\mu(s|\theta^{\mu})$,权重分别为θ^{Q}和θ^{μ}。

步骤2 初始化目标网络的Q'和μ',权重分别为$\theta^{Q'}=\theta^{Q}$和$\theta^{\mu'}=\theta^{\mu}$。

步骤3 初始化缓存区R。

步骤4 设定训练的总回合数M,开始循环,循环步骤如下。

1) 为动作探索初始化一个参数噪声N_t;初始化状态s_1,并得到姿态角和姿态角速度的观测值;设定每回合的总控制时长T,开始每回合的循环;根据当前策略和探索动作的参数噪声选择动作,也即选择控制力矩$a_t=\mu(s_t|\theta^{\mu})+N_t$。

2) 执行控制力矩a_t,根据航天器姿态动力学模型,航天器的姿态角和姿态角速度发生改变。得到奖励或惩罚r_t,并观测新状态s_{t+1}。

3) 把(s_t,a_t,r_t,s_{t+1})作为样本传输到R中储存。

4) 从R中随机抽取minibatch个样本(s_t,a_t,r_t,s_{t+1})。

5) 设$y_i=r_i+\gamma Q'\left(s_{i+1},\mu'(s_{i+1}|\theta^{\mu'})|\theta^{Q'}\right)$。

6) 通过最小化误差来更新Critic网络:$L=\frac{1}{N}\sum_{i}\left(y_i-Q(s_i,a_i|\theta^{Q})\right)^2$。

7) 使用SGD更新Actor网络:$\nabla_{\theta^{\mu}}J\approx\frac{1}{N}\sum_{i}\nabla_{a}Q(s,a|\theta^{Q})\big|_{s=s_i,a=\mu(s_i)}\nabla_{\theta^{\mu}}\mu(s|\theta^{\mu})\big|_{s_i}$。

8) 更新目标网络:$\theta^{Q'}\leftarrow\tau\theta^{Q}+(1-\tau)\theta^{Q'}$,$\theta^{\mu'}\leftarrow\tau\theta^{\mu}+(1-\tau)\theta^{\mu'}$。

在每个步长中,循环上述1~8步,直到该回合结束。

步骤5 循环结束,得到训练好的姿态控制器。

2.3 回报函数

将回报函数设计为3部分:

$$r=r_1+r_2+r_3 \tag{10}$$

式中:r_1为当回合中姿态角和姿态角速度都满足目标精度范围内时的奖励,设置为常数,在训练后期,r_1的设置能够使姿态角和姿态角速度更快收敛到目标精度范围内,提高学习效率;r_2为当回合中姿态角或姿态角速度不满足目标精度时的惩罚,也即设计的辅助回报函数;r_3为当回合中姿态角或姿态角速度严重超出允许范围时的惩罚,设置为常数。r_3的设置,一方面可以避免控制时航天器出现翻滚的情况,另一方面也可以避免计算过程中因数值过大导致训练失败。

设辅助回报函数r_2为

$$r_2=-(l_0\alpha^{i}+l_1\omega^{i}+l_2M^{i}) \tag{11}$$

式中:$\alpha^{i}=|\psi|^{i}+|\theta|^{i}+|\varphi|^{i}$、$\omega^{i}=|\omega_x|^{i}+|\omega_y|^{i}+|\omega_z|^{i}$、$M^{i}=|T_{cx}|^{i}+|T_{cy}|^{i}+|T_{cz}|^{i}$分别为姿态角、角速度和控制力矩的惩罚项;参数$i$为计算时所取的指数。辅助回报函数在训练前期时,可引导姿态角,角速度和控制力矩通过训练收敛到0;l_0、l_1、l_2为比例系数,用于调整各惩罚项的大小关系,保证每一项都可以起作用。比例系数的设定应满足当达到目标精度时,使回报函数数值大小落在$[-1,1]$内,此时训练过程中数值比较稳定。

在进行参数调整时，首先，只保留姿态角惩罚项，调整l_0的大小，使得训练出来的控制器能够满足姿态角的目标精度；其次，加入角速度惩罚项，调整l_1的大小，使得训练出来的控制器能够满足角速度的目标精度；最后，加入力矩惩罚项，调整l_2的大小，使得角速度能够不再震荡。

3　仿真实验和结果分析

使用DDPG算法对姿态控制器进行训练，训练流程参考2.2节，对仿真中姿态动力学环境搭建和神经网络搭建的参数进行说明。

3.1　航天器姿态动力学环境

针对三轴稳定航天器的姿态镇定控制进行仿真。设航天器本体转动惯量$\boldsymbol{I}=\mathrm{diag}[220,210,58]\ \mathrm{kg\cdot m^2}$。航天器绕地球圆轨道运行，轨道角速度$\Omega=0.001\ \mathrm{rad/s}$。仿真时需考虑重力梯度力矩的影响。

为了能够更加充分地探索状态空间，训练时每回合初始时刻的姿态角和姿态角速度由系统在一定范围内随机生成。设训练时每回合初始时刻，航天器3个通道的姿态角和姿态角速度的分量在$-30^\circ\sim30^\circ$和$-10\sim10\ (^\circ)/\mathrm{s}$的范围内随机选择。

使用飞轮控制，设控制力矩范围为$-5\sim5\ \mathrm{N\cdot m}$。在选择控制力矩时加入Ornstein-Uhlenbeck噪声，噪声可以帮助算法更加充分地探索周围的环境，使训练效率和效果都大大提升。

3.2　神经网络和训练超参数

进行训练的最大步数为10^6，每回合最大时长40 s，采样时间为0.5 s，奖励衰减值γ为0.99。建立Actor部分的动作现实网络和动作估计网络、Critic部分的状态现实网络和状态估计网络时，所建立的神经网络均为结构相同的BP神经网络，使用ReLU函数作为神经网络的激活函数，中间的隐藏层神经元个数为256个，训练控制器使用的辅助回报函数为式(11)，选择$i=1$，其他条件保持不变。

3.3　仿真结果及分析

为了测试使用强化学习方法训练得到的姿态控制器不依赖于航天器的质量参数，使用训练好的姿态控制器对不同质量参数的受扰航天器实施控制。设初始时刻受扰航天器3个通道的姿态角均为30°，姿态角速度均为$10\ (^\circ)/\mathrm{s}$。设置3组不同质量参数的航天器，分别为训练时使用的航天器转动惯量$\boldsymbol{I}$，将转动惯量减小50%的$\boldsymbol{I}/2$和将转动惯量增加100%的$2\boldsymbol{I}$。3组测试中受扰航天器的姿态角、姿态角速度随时间的变化曲线如图1所示。

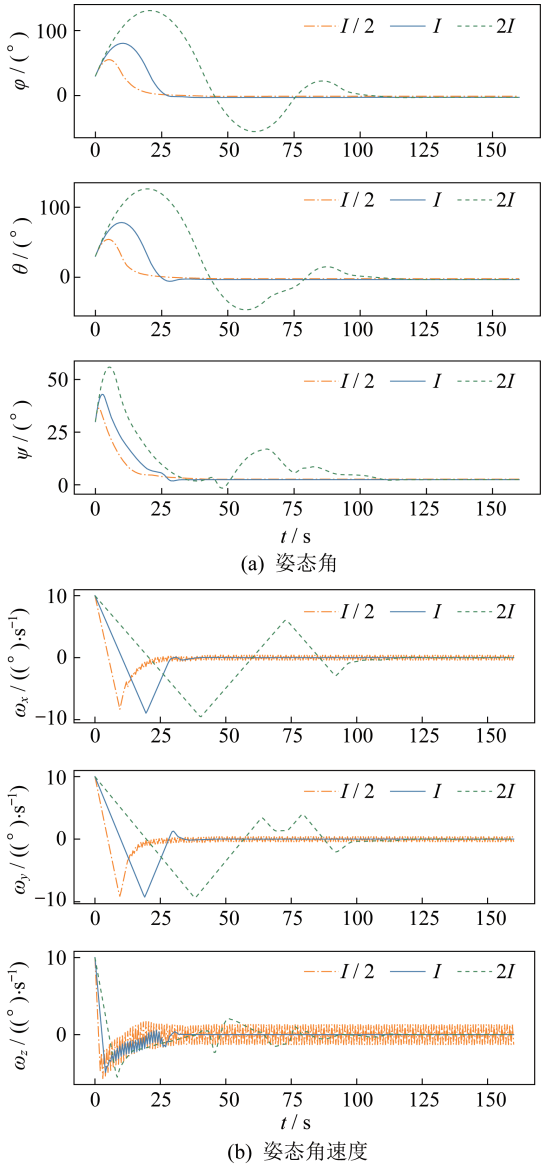


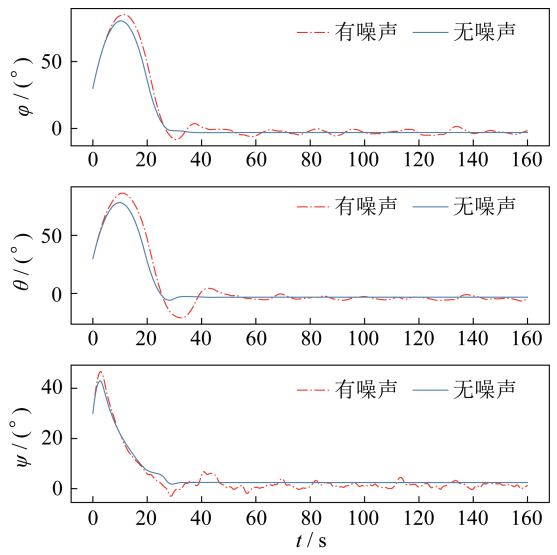
图1　不同转动惯量时姿态角和姿态角速度变化曲线

Fig. 1　Curves of the attitude angle and attitude angle velocity at different values of the inertia moment

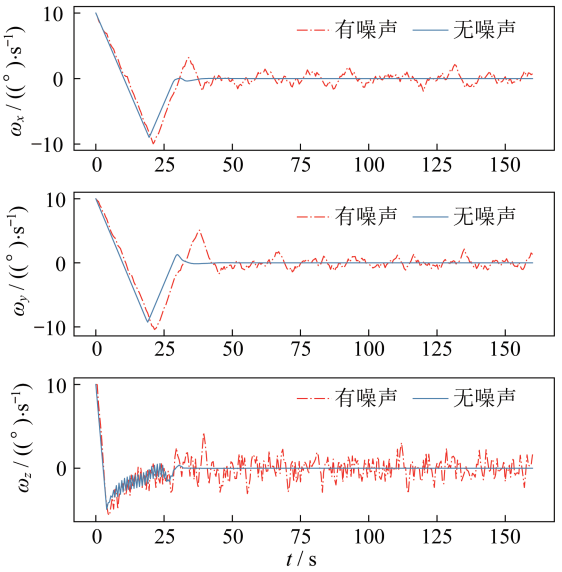
由图1(a)可知，尽管质量参数发生较大的改变，姿态角3条曲线最终都收敛到了0附近，并且满足精度要求，由于转动惯量发生变化，而力矩限制范围没有变，因此控制时间会随着转动惯量增大而增大。由图1(b)可知，当转动惯量减小50%时，角速度曲线出现了小幅的震荡，其中z轴的震荡幅度

最大,但仍然在误差允许范围内,没有出现发散的情况。通过对比图1中的曲线可以发现,尽管质量参数发生较大改变,经过DDPG算法训练的姿态控制器仍然能够较好地完成姿态控制任务,控制器对质量参数变化具有良好的鲁棒性。

测试训练好的控制器是否可以应对系统存在测量误差和存在外界干扰力矩的情况。设测量噪声在$-1^\circ \leqslant \varphi、\theta、\psi \leqslant 1^\circ$和$-1(^\circ)/\mathrm{s} \leqslant \omega_x、\omega_y、\omega_z \leqslant 1(^\circ)/\mathrm{s}$内随机产生,力矩噪声均值为通过策略选择得到的力矩值,噪声方差为2 N·m。仿真结果如图2所示。



(a) 姿态角



(b) 姿态角速度

图2 有无噪声时姿态角和姿态角速度变化曲线

Fig. 2 Curves of the attitude angle and attitude angle velocity with and without noise

图2为同时加入测量噪声和干扰力矩后,受扰航天器的姿态角、姿态角速度和控制力矩随时间的变化曲线图。由图2可知,当加入测量噪声和干扰力矩之后,控制系统的调节时间变化不大,但稳态误差有所增大,说明强化学习控制器可以做到在一定范围内的测量噪声和干扰力矩的作用下,使受扰控制器恢复姿态镇定。

对不同辅助回报函数进行实验,训练控制器使用的辅助回报函数为式(11),参数i分别选择0.5、1.0、2.0,分别代表选择了凸函数、线性函数、凹函数。使用不同辅助回报函数进行训练,训练得到的满足要求的控制器训练需要的步数和最终控制器的性能均有所不同。对不同辅助函数训练得到的控制器进行测试,并将测试结果进行整理,见表1。

表1 不同辅助回报函数训练效果比较

Tab. 1 Comparison of the training effects of different auxiliary reward functions

函数类型		$i=0.5$	$i=1$	$i=2$
训练步数		4 720	3 050	4 250
控制误差	$\varphi/(^\circ)$	0.214	0.308	0.645
	$\theta/(^\circ)$	0.263	0.533	0.749
	$\psi/(^\circ)$	0.260	0.690	0.624

由表1可知,训练得到控制器的精度随着i的增大而减小,而训练的步数则是当$i=1$时最少,但数量级相同。分析其原因,由于设置最终的控制精度绝对值小于1,此时若辅助回报函数取凹函数,则算法训练到后期接近目标控制精度时,辅助回报函数的数量级将会更小,计算出来的更新Critic梯度也会更小,导致后期辅助回报函数失效,此时无法再向更高的精度收敛,而使用凸函数则可以使精度更高。训练步数方面说明不同i的取值对训练步数的影响不大,需要考虑其他参数设置。

4 结束语

使用强化学习方法对航天器进行了姿态控制器设计。强化学习中,选择了能够用于连续控制领域的DDPG算法。DDPG算法能够通过与航天器姿态动力学环境进行互动,得到训练样本,然后随机选择训练样本,根据回报函数计算误差,并对Actor和Critic神经网络进行更新,最终通过迭代得到训练好的控制器。强化学习在整个训练过程中没有用到航天器的相关参数,表现出更好的鲁棒性。

通过仿真测试，验证了DDPG算法设计的控制器对航天器质量参数具有良好的鲁棒性，并且发现了控制器在环境中的力矩干扰和测量噪声也具有一定的控制能力。回报函数设计对强化学习训练效果具有很大影响，因此还对不同回报函数进行对比，实验结果表明，当控制精度绝对值小于1时，设置凹函数会提高控制器的精度。

但只考虑了强化学习在地面训练控制器后再上天在轨控制，而未考虑强化学习直接在轨进行学习控制，后面将进行在轨学习方面的研究。

参考文献

[1] 王金华，薄煜明，高旭东，等.FY-3(05)星主动对月定标控制技术研究[J].上海航天(中英文)，2021，38(2)：37-44.

[2] LIU E, YANG Y, YAN Y. Spacecraft attitude tracking for space debris removal using adaptive fuzzy sliding mode control [J]. Aerospace Science and Technology, 2020, 107: 1-13.

[3] 刘峰，岳宝增，马伯乐，等.燃料消耗下充液航天器等效动力学建模与分析[J].力学学报，2020，52(5)：1454-1464.

[4] 毛旭光，陈洲.航天器姿态控制算法研究综述[J].电脑与信息技术，2016，24(2)：25-29.

[5] SWEETING M N, HASHIDA Y, BEAN N P, et al. CERISE microsatellite recovery from first detected collision in low Earth orbit [J]. Acta Astronautica, 2004, 55(2): 139-147.

[6] KAELBLING L P, LITTMAN M L, MOORE A W. Reinforcement learning: a survey [J]. Journal of Artificial Intelligence Research, 1996, 4: 237-285.

[7] LI Y. Deep reinforcement learning: an overview [EB/OL]. (2018-11-25) [2020-11-19]. http://arxiv. org/abs/1701.07274.

[8] MNIH V, KAVUKCUOGLU K, SILVER D. Playing atari with deep reinforcement learning[EB/OL]. (2013-12-19)[2020-11-19]. http://arxiv.org/abs/1312.5602.

[9] LILLICRAP T P, HUNT J J, PRITZEL A, et al. Continuous control with deep reinforcement learning [EB/OL]. (2019-07-05) [2023-01-31]. http://arxiv.org/abs/1509.02971.

[10] YANG Q, ZHU Y, ZHANG J, et al. UAV air combat autonomous maneuver decision based on DDPG algorithm [C]// 2019 IEEE 15th International Conference on Control and Automation (ICCA). Edinburgh: IEEE, 2019: 37-42.

[11] WANG D, SHEN Y, SHA Q, et al. Adaptive DDPG design-based sliding-mode control for autonomous underwater vehicles at different speeds [C]// 2019 IEEE Underwater Technology (UT). Kaohsiung: IEEE, 2019: 1-5.

[12] LIU Y C, HUANG C Y. DDPG-based adaptive robust tracking control for aerial manipulators with decoupling approach [J]. IEEE Transactions on Cybernetics, 2021, 52(8): 8258-8271.

[13] WU X, LIU S, ZHANG T, et al. Motion control for biped robot via DDPG-based deep reinforcement learning [C]// 2018 WRC Symposium on Advanced Robotics and Automation (WRC SARA). Beijing: IEEE, 2018: 40-45.

[14] HENDERSON P, ISLAM R, BACHMAN P. Deep reinforcement learning that matters[EB/OL]. (2019-01-29)[2020-11-21]. http://arxiv.org/abs/1709.06560.

[15] MATHERON G, PERRIN N, SIGAUD O. The problem with DDPG: understanding failures in deterministic environments with sparse rewards [EB/OL]. (2019-11-26) [2021-05-07]. http://arxiv. org/abs/1911.11679.

[16] NG A Y, HARADA D, RUSSELL S J. Policy invariance under reward transformations: theory and application to reward shaping[C]// Proceedings of the Sixteenth International Conference on Machine Learning. San Francisco: Morgan Kaufmann Publishers Inc., 1999: 278-287.

[17] BERENJI H R, LEA R N, JANI Y, et al. Space shuttle attitude control by reinforcement learning and fuzzy logic [C]// Second IEEE International Conference on Fuzzy Systems. San Francisco: IEEE, 1993: 1396-1401.