计算机科学
COMPUTER SCIENCE

基于深度强化学习的四旋翼无人机自主控制方法

梁吉，王立松，黄昱洲，秦小麟

引用本文

梁吉，王立松，黄昱洲，秦小麟. 基于深度强化学习的四旋翼无人机自主控制方法[J]. 计算机科学，2023，50(11A)：220900257-7.

LIANG Ji, WANG Lisong, HUANG Yuzhou, QIN Xiaolin. Autonomous Control Algorithm for Quadrotor Based on Deep Reinforcement Learning [J]. Computer Science, 2023, 50(11A): 220900257-7.

相似文章推荐（请使用火狐或 IE 浏览器查看文章）

Similar articles recommended (Please use Firefox or IE to view the article)

自动化红队测试中强化学习策略的实现与验证

Implementation and Verification of Reinforcement Learning Strategy in Automated Red Teaming Testing

计算机科学, 2023, 50(11A): 230200162-6. https://doi.org/10.11896/jsjkx.230200162

基于SA-UCB算法的Android应用程序自动化测试方法

Automated Testing Method of Android Applications Based on SA-UCB Algorithm

计算机科学, 2023, 50(11A): 221200145-7. https://doi.org/10.11896/jsjkx.221200145

车载边缘计算网络中基于MAB的动态任务卸载方案研究

Study on Dynamic Task Offloading Scheme Based on MAB in Vehicular Edge Computing Network

计算机科学, 2023, 50(11A): 230200186-9. https://doi.org/10.11896/jsjkx.230200186

基于深度强化学习的无线异构网络中继决策研究

Study on Relay Decision in Wireless Heterogeneous Networks Based on Deep ReinforcementLearning

计算机科学, 2023, 50(11A): 221000088-5. https://doi.org/10.11896/jsjkx.221000088

云边协同计算中基于强化学习的依赖型任务调度方法

Dependency-aware Task Scheduling in Cloud-Edge Collaborative Computing Based on Reinforcement Learning

计算机科学, 2023, 50(11A): 220900076-8. https://doi.org/10.11896/jsjkx.220900076

计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.220900257

基于深度强化学习的四旋翼无人机自主控制方法

梁 吉 王立松 黄昱洲 秦小麟
南京航空航天大学计算机科学与技术学院 南京 211106
(2276835336@qq.com)

摘 要 随着无人机的广泛应用,无人机控制器的设计成为近年来广泛研究的热点。当前无人机中广泛使用的PID,MPC等控制算法受到参数难调节、模型构建复杂、计算量大等一系列因素的制约。针对上述问题,提出了一种基于深度强化学习的无人机自主控制方法。该方法通过神经网络拟合无人机控制器,直接将无人机的状态映射到舵机的输出以控制无人机运动,在不断与环境进行交互训练中即可得到一个通用的无人机控制器,有效地避免了参数调节、模型构建等复杂操作。同时,为进一步提高模型的收敛速度和准确性,在传统强化学习算法Soft Actor Critic(SAC)的基础之上引入专家信息,提出了ESAC算法,指导无人机对环境进行探索,以增强控制策略的易用性和扩展性。最后在无人机的位置控制以及轨迹跟踪任务中,通过与传统PID控制器和SAC,DDPG等强化学习算法构建的模型控制器进行对比,实验结果表明,通过ESAC算法构建的控制器能够达到与PID控制器同样甚至更优的控制效果,同时在稳定性和准确性上优于SAC和DDPG构建的控制器。

关键词:强化学习;四旋翼无人机;自主控制;专家策略
中图法分类号 TP391

Autonomous Control Algorithm for Quadrotor Based on Deep Reinforcement Learning

LIANG Ji,WANG Lisong,HUANG Yuzhou and QIN Xiaolin
College of Computer Science and Technology,Nanjing University of Aeronautics and Astronautics,Nanjing 211106,China

Abstract With the wide application of UAV,the design of UAV controller has become a hot research topic in recent years. The control algorithms such as PID and MPC widely used in UAV are restricted by a series of factors such as difficult parameter adjustment,complex model construction,and large amount of calculation. Aiming at the above problems,a UAV autonomous control method based on deep reinforcement learning is proposed. This method fits the UAV controller through a neural network,directly maps the state of the UAV to the output of the steering gear to control the movement of the UAV,and can obtain a general UAV controller in the continuous interactive training with the environment. This method effectively avoids complex operations such as parameter adjustment and model building. At the same time,in order to further improve the convergence speed and accuracy of the model,on the basis of the traditional reinforcement learning algorithm soft actor critic(SAC),by introducing expert information,an ESAC algorithm is proposed,which guides the UAV to explore the environment and enhances the ease of control strategy. Finally,in the position control and trajectory tracking tasks of the UAV,compared to the traditional PID controller and the model controller constructed by SAC,DDPG and other reinforcement learning algorithms,experimental results show that the controller constructed by the ESAC algorithm can achieve the same level as the PID controller,and it is better than the controller built by SAC and DDPG in stability and accuracy.

Keywords Reinforcement learning,Quadrotor,Autonomous control,Expert policy

1 引言

在过去的十几年里,随着传感器、嵌入式计算等相关技术的发展与应用,无人机(Unmanned Aerial Vehicle,UAV)以其结构简单、易操作等优势在各个领域都得到了广泛的应用[1]。尽管无人机在各个领域得到了充分的发展,但飞行控制系统的设计仍被认为是一个复杂的研究课题。一方面,由于飞行控制系统的控制频率在200~500 Hz之间,需要在极短的时间内收集传感器信息并进行处理,然后计算得到无人机的控制量。另一方面,由于飞行控制系统需要具备一定的自适应性,以有效应对复杂的外部环境变化。这对飞控系统的设计提出了较高的要求。因此进一步提高飞行控制系统的自适应性和简化控制系统设计具有非常大的挑战,并且具有十分重要的研究价值和意义。

传统的无人机飞行控制算法主要可以分为两类。一种为不依赖于无人机的动力学模型与模型无关的控制算法,例如PID控制算法[2],其思想为通过对无人机当前状态和期望状态之间的误差进行比例、积分、微分运算以得到无人机的控制量,调节当前状态与期望状态之间的误差使其趋近于0。PID控制算法对控制参数的设置十分敏感,参数的选取对控制

基金项目:国家自然科学基金(61972198)
This work was supported by the National Natural Science Foundation of China(61972198).
通信作者:秦小麟(qinxcs@nuaa.edu.cn)

效果起着至关重要的作用，当前 PID 算法的参数的选取往往是基于人类的先验经验，一经设置在整个飞行任务中不再改变，因此飞行控制系统很难发挥出最大的性能。另一类控制算法是基于无人机的动力学模型进行设计的，如 MPC[3]，LQR[4]等控制算法，其主要思想为通过构建无人机动力学模型来计算未来一定时间步内的最优控制量。而对于四旋翼无人机而言，其是一个典型的欠驱动非线性系统[5]，其 12 维状态（位置、速度、姿态、角速度）变量需要由 4 个舵机的转速来进行调节。对于这样一个复杂的系统，对其进行精确的建模存在着很大的难度，因此 MPC，LQR 等算法在无人机的飞行控制也很难发挥较大作用。

因此传统的飞行控制算法受到参数调节困难[6]、模型构建复杂[7]等一系列因素的制约，很难发挥其最优的性能以有效应对复杂的环境变化。同时，将人工智能技术应用到移动机器人控制领域中是当前的一个研究热点。而强化学习（RL）因其自身的特点在机器人控制领域具有十分广泛的应用，相比传统的监督式学习和非监督式学习，强化学习的模型的获取是在智能体（Agent）不断与环境进行交互中得到的，这种学习方式更符合人类的学习过程，因此其在机器人自主控制领域具有先天优势。

针对上文提出的传统无人机控制算法受到模型构建复杂、参数调节困难等问题，本文提出一种基于深度强化学习的无人机自主控制方法，直接利用神经网络拟合无人机控制器，输入无人机当前状态计算得到控制舵机转速的控制量。通过与环境不断进行交互，从而学习得到无人机的控制策略，避免了动力学模型构建、控制器参数调节等复杂操作。同时，为进一步加快控制策略的构建速度，降低策略学习的难度，本文在传统 SAC 的基础上引入专家策略，提出了 ESAC 算法，指导无人机与环境进行交互，缩小无人机的探索空间。最后本文在仿真环境中对所提算法进行验证，实验结果表明，在无人机的位置控制和轨迹跟踪任务中，相比传统的无人机控制方法，本文提出的的网络控制方法能够达到与 PID 控制器相近甚至优于其性能的控制效果。

2 相关工作

当前将人工智能技术应用于移动机器人自主问题是一个主要的研究方向[8-10]。而无人机飞行控制系统的设计主要分为基于无人机动力学模型的控制器和无模型的控制器两类。下面简单介绍这两类方法当前主要的研究工作。

在不依赖于动力学模型的无人机飞控系统设计中，PID 控制算法因其结构简单、易操作等优良特性成为当前无人机飞行控制算法中采用最广泛的控制算法，例如 PX4，APM 等开源控制系统均采用 PID 算法作为其控制算法[11]。

式（1）所示为一简单的 PID 算法数学表达式，其具体控制流程如图 1 所示。

$$o(t)=k_p e(t)+k_i\int e(t)+k_d\frac{\mathrm{d}}{\mathrm{d}t}e(t) \tag{1}$$

从式（1）可以看出比例、积分、微分参数（增益）k_p，k_i，k_d 对最终的控制结果起着至关重要的影响。同时当前 PID 控制器的参数设置往往是基于人类先验知识，需要经过反复的大量调整才能应用于控制器之中，因此飞控的性能很难达到最优状态。

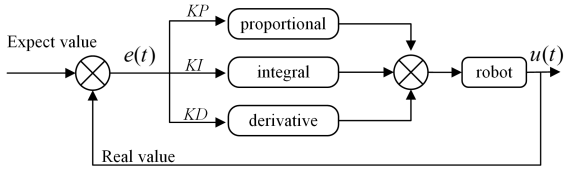


图 1 PID 控制器

Fig. 1 PID controller

针对于此，近年来许多参数自适应的算法相继被提出。例如 Leung 等[12]基于网格点的概念提出了 PID 参数调度控制器，实现了利用线性模型对非线性系统进行控制的目标。Papadopoulos 等[13]提出了一种基于幅度最优原则的 PID 控制器自动调节方法，仅解决了对于线性动力学模型的单输入单输出控制问题。然而，上述方法都是基于动力学模型已知这一假设，在使用过程中需要精确地对被控物体进行建模，这在实际应用中是很难做到的，对于四旋翼无人机这种多输入多输出的欠驱动非线性系统尤其如此。

更进一步，采用神经网络的方式直接将状态映射到动作，同样也是不依赖于无人机动力学模型的控制方法。例如 Cheng 等[14]采用强化学习的方式训练控制模型替代无人机中的控制算法并在无人机的轨迹跟踪任务中进行测试，但此方法依赖于固定场景，需要根据不同的场景构建不同的模型。Xie 等[15]利用深度强化学习的方法将模型控制应用到无人车的控制中，在该方法中，他们基于 DDPG[16]算法提出了 AsDDPG 算法以辅助 Agent 学习，并最终在无人车的轨迹跟踪任务中对算法的有效性进行了验证。Panerati 等[17]利用 pybullet 物理仿真引擎构建了无人机的仿真环境，并利用强化学习的方式训练无人机学会起飞，但该方法同样依然依赖于固定的场景，场景变化后控制模型将不再适用。Lopes 等[18]采用强化学习中的 PPO[19]算法，尝试构建无人机的控制器，同样该方法也依赖于训练场景。综上可以看出，在利用模型构建无人机的控制器的过程中，需要解决无人机动作空间大以及模型依赖于固定训练场景两个问题。

在另一类基于无人机动力学模型已知这一假设下的 MPC 控制算法的主要思想是在未来一定时间步内寻找最优的控制策略，其对安全性有一定保证，因此被广泛的应用于自动驾驶领域。因此 MPC 算法关键在于模型的构建，当前的一系列的研究工作[20-22]关注于从数据中构建被控物体（如机械手臂、无人机、无人船等）的动力学模型。其中 Kabzan 等[23]将收集大量的数据通过高斯模型拟合赛车动力学模型，将 MPC 控制算法成功应用于无人赛车的控制之中。Torrente 等[24]同样通过高斯模型拟合无人机的动力学模型，然后通过添加约束将 MPC 算法应用于无人机的轨迹跟踪控制中，但他们仅在 Y-Z 二维平面内完成了对无人机的控制。Lambert 等[25]通过强化学习方法构建无人机的状态动作预测模型，然后利用 MPC 算法完成了对无人机起飞的控制。

对于四旋翼无人机的控制，其 12 维状态变量（位置、速度、姿态、加速度）需要由 4 个电机的转速进行调节，相较于无人车、机械手臂等物体，其动力学模型更复杂，通过传统的动力学分析或基于数据的学习方式很难对其进行精确建模。同时由于飞控系统的控制频率很高，需要在很短的时间内计算得到控制量，而在 MPC 算法中，当预测步长 T 增大时，其计算量显著增大，很难在短时间内计算得到控制量，不能满足

实时控制的要求。因此 MPC 算法在无人机的自主控制应用中受到很大的限制。

综上所述，当前在无人机飞控中广泛使用的 PID 算法、MPC 算法由于受到参数调节困难、模型构建复杂、计算复杂度高等一系列的限制，飞控系统很难发挥出最优的控制性能。因此本文提出了基于强化学习的无人机自主控制算法，采用直接将无人机状态映射到控制输出的方式构建控制模型，在不需要任何无人机先验知识的前提下通过学习的方式即可获取到一个通用的无人机控制器，有效地规避了上述问题。

3　端到端无人机自主控制器

3.1　问题定义

标准的强化学习框架主要由一个学习代理（Agent）和马尔可夫决策过程（MDP）组成。Agent 表示智能体，MDP 可以由一个 5 元组 $(S,A,P_{ss'}^{a},R,\gamma)$ 表示，其中 S 表示智能体的状态空间，A 表示智能体所能采取的动作的集合，$P_{ss'}^{a}:S\times A\to S$ 表示智能体从当前状态 s 采取动作 a 转移到下一状态 s' 的转移概率矩阵，$R:S\times A\to R$ 表示智能体从当前状态 s 采取动作 a 环境给予的奖励，$\gamma\in(0,1)$ 表示折扣因子，代表未来奖赏值对当前动作选取的影响程度。具体的强化学习基本过程如图 2 所示，可以描述为在 t 时的 Agent 的状态为 s_t，根据策略函数 $\pi_\theta(a_t|s_t)$ 选取动作 a_t，然后按照状态转移函数 $P(s_{t+1}|s_t,a_t)$ 转移到下一状态 s_{t+1}，此时环境会根据智能体所采取的动作和状态返回奖励值 $r_t(s_t,a_t)$。强化学习的最终目标为期望累计奖赏最大以寻找到最优控制策略，其中累计奖赏和最终优化目标的定义如式（2）、式（3）所示。

$$G_t=r_t+\gamma r_{t+1}+\gamma^2 r_{t+2}+\cdots=\sum_{k=0}\gamma^k r_{k+1} \tag{2}$$

$$\pi^*=\max_{\pi}(E(\sum_{t=0}^{\infty}\gamma^t R(s_t,a_t,s_{t+1}))) \tag{3}$$

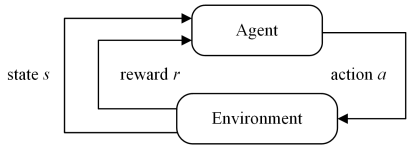


图 2　智能体与环境进行交互训练过程

Fig. 2　Interactive training process of agent and environment

因此基于深度强化学习的无人机网络控制器的构建可以建模为一个马尔可夫决策过程，即 S 表征无人机的状态集合，A 表征无人机舵机转速的集合，$P_{ss'}^{a}$ 为无人机控制器。因此控制器的构建即在无人机不断与环境进行交互的过程中，朝着最大化累计奖赏的方向优化控制策略，最终学会如何根据无人机当前自身状态和目标调整舵机转速。

在网络控制器构建过程中，由于无人机 4 个舵机转速空间较大，且无人机的姿态与性能对舵机速度变化十分敏感，转速的异常变化会导致无人机坠毁。直接利用强化学习方法对无人机的转速空间进行探索，存在着探索空间巨大、策略网络难以收敛、网络控制性能差等一系列问题。因此文中基于传统的强化学习算法 SAC 提出一种基于专家策略指导的 ESAC 算法，指导 Agent 对动作空间进行探索，以加速策略网络收敛，提升网络控制器的适用性和控制性能。

3.2　自主控制器设计

图 3 给出了基于强化学习的端到端的无人机自主控制算法的整体结构。Agent 根据无人机的当前策略函数计算得到无人机的控制量 $u(t)=\pi(\cdot|s_t)$，并将其应用于环境之中。接下来环境反馈回基于当前状态与动作的奖励值 r_t，无人机转移到下一状态 s_{t+1}。通过收集大量的无人机状态动作对 (s_t,a_t,s_{t+1},r_t) 并根据优化目标更新网络参数，然后通过重复此流程迭代更新网络参数，以最大化累计奖赏，寻找最优的控制策略。

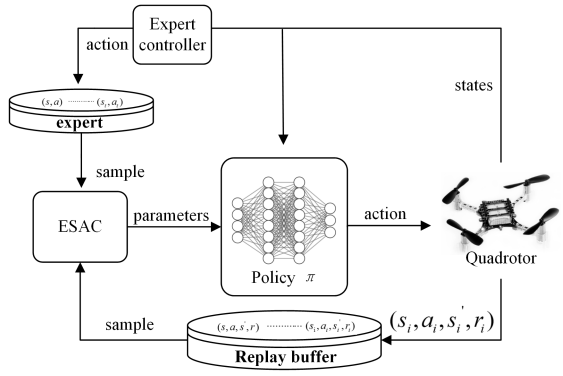


图 3　基于强化学习的四旋翼无人机控制

Fig. 3　RL for quadrotor control

无人机的状态 q 定义如式（4）所示，其中 x,y,z 为无人机的位置表示，$\dot{x},\dot{y},\dot{z}$ 为无人机在 3 个轴上的线速度表示，φ,θ,ψ 为无人机的姿态角表示，$\dot{\varphi},\dot{\theta},\dot{\psi}$ 表示无人机绕 x,y,z 轴的角速度。

$$q=(x,y,z,\dot{x},\dot{y},\dot{z},\varphi,\theta,\psi,\dot{\varphi},\dot{\theta},\dot{\psi})\in R^{13} \tag{4}$$

直接采用无人机状态 q 作为网络控制器的输入，存在着模型只能应用于单一场景且更换场景之后控制策略不能发挥作用这一问题。为提高模型的泛化能力，使得在定点悬停任务和轨迹跟踪等任务中不依赖于无人机的初始位置设置以及特定的训练场景，同时在不同的飞行任务中控制策略依然适用，文中采用位置的误差和无人机的状态两部分（见式（5））作为策略网络的输入，其中 pos_{err} 为当前位置与期望位置之间的误差，该误差采用二范数的形式进行定义即物理距离，vel 表示无人机当前时刻在固定坐标系下的线速度，ang 表示无人机当前时刻绕 3 个轴旋转的角速度，$\boldsymbol{M}$ 为机体坐标系到固定坐标系旋转矩阵。采用位置误差和状态两部份作为网络控制器的输入可以更好地表征期望位置和当前状态的关系，使得在不同的飞行场景和任务中决策策略依然能够发挥作用。

$$s=(pos_{err},vel,ang,\boldsymbol{M})\in R^{18} \tag{5}$$

进一步地，对于网络控制器的输出，由于四旋翼无人机每个舵机产生的升力 F_i 与转速 ω 的平方成正比，具体表达式如式（6）所示，其中 K_f 为比例系数。而当 4 个舵机的转速相等并且 4 个舵机产生的升力之和等于无人机的重力时，无人机达到平衡状态。因此为了减少 Agent 与环境的交互次数，降低了训练难度。文中采用式（7）作为最终的无人机控制量，其中 ω_h 为悬停转速，α 为网络控制器的输出，β 为缩放因子。即采用相对于平衡转速的偏移量作为网络控制器的输出，可以有效地提高无人机的稳定性，减少 Agent 与环境的交互次数，降低学习难度。

$$F_i=K_f\omega^2,i\in(1,2,3,4) \tag{6}$$

$$o=\omega_h(1+\beta a) \tag{7}$$

在强化学习算法中，奖励值函数是对 Agent 基于当前状态 s_t 所采取动作 a_t 的评价，是整个算法的核心所在，正确

反映状态与动作之间关系的奖励值函数设计可以引导 Agent 对环境进行探索，防止 Agent 陷入局部最优状态。文中采用的奖励函数由 5 部分组成，如式(8)所示，其中 α 为各部分之间的权重系数值均大于 0，p_e 为当前位置与期望位置之间的距离，v 和 q 分别表示无人机当前的线速度与角速度，其采用二范数进行定义，a 和 a_e 分别表示当前网络控制器的输出以及当前控制器的输出与上一轮控制器输出之间的差值，其同样也采用二范数的形式进行定义。其中 p_e 表示无人机距离目标位置的远近表征了任务完成的好坏，其余 4 项为稳定项，用于表征无人机飞行的稳定性。因此参数 $\alpha_1 > \alpha_i (i=2,3,4,5)$ 将完成任务作为优先考虑，其余 4 项用于对无人机的稳定性进行约束，取值范围相同。该奖励值函数设计的主体思想为期望位置误差越小越好，使得无人机更接近目标位置，同时期望速度和角速度越小越好，以防止其坠落，而对于决策的输出和两轮控制器输出的差值，同样期望其越小越好，此时网络控制器的输出为悬停转速，可以更好地保证无人机稳定飞行。

$$\begin{aligned} &\alpha=[\alpha_1,\alpha_2,\alpha_3,\alpha_4,\alpha_5] \\ &r=-\alpha[\|p_e\|,\|v\|,\|q\|,\|a\|,\|a_e\|]^{\mathrm{T}} \end{aligned} \tag{8}$$

3.3 ESAC 算法

由于网络控制器的输入为 18 维变量，输出为 4 维变量，需要 Agent 探索的动作空间巨大，传统的强化学习算法很难完成控制器的构建，直接采用强化学习训练的方式，决策模型难以收敛。因此本文提出了 ESAC 算法，通过在传统 SAC (soft actor critic)算法的基础上引入专家策略指导 Agent 探索，使其朝着专家策略方向进行探索更新，有助于缩小探索空间，加速网络控制器收敛。

ESAC 算法主要分为两个部分，第一部分为传统的 SAC 算法，通过不断收集 Agent 与环境交互产生的状态动作对根据 SAC 算法对网络进行更新，而 SAC 算法相比传统的强化学习方法，在期望最终累计奖赏最大的同时也期望动作的熵也最大，鼓励 Agent 对环境进行探索，防止其陷入局部最优状态，更适用于探索空间大等问题，其优化的目标如式(9)所示。第二部分为利用专家信息指导 Agent 对环境进行探索，根据收集的状态动作对计算专家基于当前状态所产生的动作，然后以专家的决策作为标签、以智能体的决策作为预测、以二者的均方误差损失作为监督损失进行训练。其特点为通过监督的方式拟合专家策略从而指导智能体训练，使其朝着专家策略的方向进行探索优化，从而可以有效地缩减智能体探索空间，加快模型的收敛速度，同时提升模型的性能。最终的优化目标如式(10)所示，其中 π_e 为专家策略，即期望当前策略与专家策略更接近，而当前策略与专家策略的差异采用均方误差的形式进行定义。图 4 给出了 ESAC 的整体结构。

$$\max_\pi(E(\sum_{t=0}^{\infty}\gamma^t(R(s_t,a_t,s_{t+1})+\alpha H(\pi(\cdot|s_t))))) \tag{9}$$

$$J(\pi)=MSE(\pi(\cdot|s_t),\pi_e(\cdot|s_t)) \tag{10}$$

ESAC 算法的整体结构依然为 Actor-Critic 结构。Actor 为策略网络即网络控制器，其输入为无人机的状态，输出为无人机的转速，Critic 为动作价值网络输入无人机当前的状态和采取的动作，输出为基于当前状态和动作的价值。在网络设置上，Actor 为一个全连接神经网络，表示为 P_θ，该网络共有 3 个隐含层，每层由 128 个神经单元组成，具体结构如图 5 所示。Critic 由 4 个相同的全连接神经网络组成，每个网络均包含 3 个隐含层并且每层由 128 个神经元组成，网络的具体结构如图 6 所示。在 Critic 的 4 个网络中，其中两个为用于计算状态动作价值的 Q 网络 $Q_{\phi,1}$，$Q_{\phi,2}$，以及两个目标 Q 网络 $Q_{\phi_{\mathrm{targ},1}}$，$Q_{\phi_{\mathrm{targ},2}}$，目标 Q 网络每间隔一段时间从计算状态动作价值的 Q 网络复制网络参数。文中所有网络中的神经元均采用 tanh()函数作为激活函数。

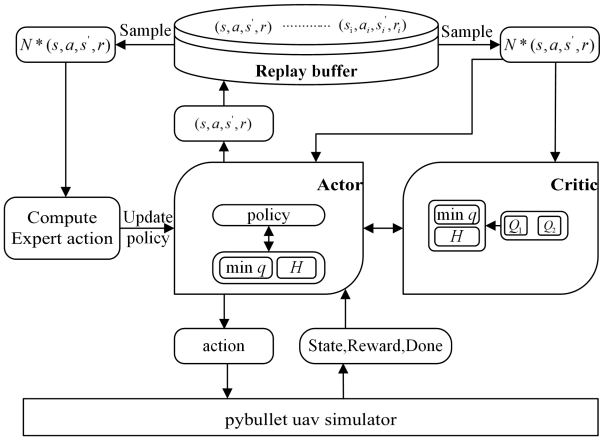


图 4 ESAC 整体结构

Fig. 4 Overall structure of ESAC

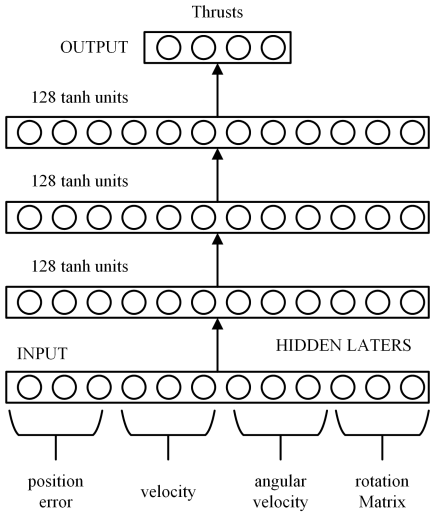


图 5 Policy 网络结构

Fig. 5 Structure of Policy neural network

在网络的更新过程中，对于 Critic 中的两个 Q 网络 $Q_{\phi,1}$，$Q_{\phi,2}$，其参数的更新需要先收集 Agent 与环境交互产生的状态动作对，然后根据目标 Q 网络计算当前状态动作对的目标值 y，计算过程如式(11)所示，然后根据目标值 y 采用梯度下降的方式更新 Q 网络，计算过程如式(12)所示。

$$\begin{aligned} &y(r,s',d)=r+\gamma(1-d)Q \\ &Q=(\min_{i=1,2}Q_{\phi_{\mathrm{targ},i}}(s',a)-\alpha\log\pi_\theta(a|s')) \\ &a\sim\pi_\theta(\cdot|s') \end{aligned} \tag{11}$$

$$\nabla_{\phi_i}\frac{1}{|B|}\sum_{(s,a,r,s',d)\in B}(Q_{\phi_i}-y(r,s',d))^2 \tag{12}$$

策略网络 P_θ 用于与环境进行交互，其根据无人机的状态和期望位置，计算得到无人机控制量 a 并应用于环境之中。此时间无人机转移到下一个状态信息 s'，同时环境返回奖励值信息 r，此时产生的四元组(s,a,r,s',d)会被存放到经验缓冲池(Replay Buffer)中，用于后续对网络的更新。对于该网络的更新，首先根据式(13)计算基于当前策略产生的动作的

熵，然后采用梯度上升的方式更新式(14)，即期望动作价值和动作熵最大[26]。

$$H=-\log\pi_\theta(a_\theta(s)|s) \tag{13}$$

$$\nabla\frac{1}{|B|}\sum_{s\in B}(\min_{i=1,2}Q_{\phi_i}(s,a_\theta(s))+\alpha H) \tag{14}$$

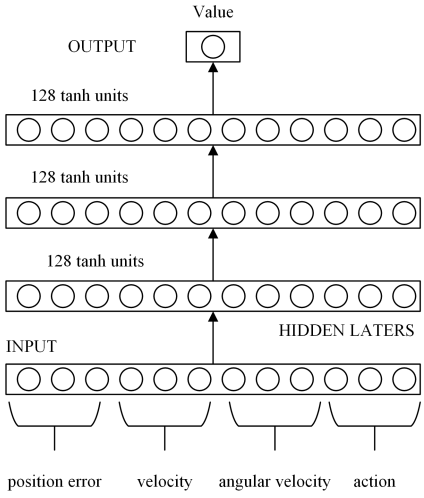


图 6　Critic 网络结构

Fig. 6　Structure of Critic neural network

具体算法如下所示，ESAC 基于 SAC 算法，主要引入第 15 行利用专家控制器对策略网络进行指导更新。

算法 1　ESAC

Input：位置误差和状态 s

Output：无人机舵机控制量 o

1. 初始化网络参数；
2. Repeat：
3.　根据当前决策策略选择动作 a～$\pi_\theta(\cdot|s)$；
4.　　计算电机转速 rpm_t；
5.　无人机转移到下一状态 s′；
6.　存储状态动作对 D←(s,a,r,s′,d)；
7.　if |D|＞M then：
8.　　for i in range(update times)：
9.　　sample B={(s,a,r,s′,d)}∈D；
10.　　根据式(11)计算目标值；
11.　　根据式(12)更新 Q 网络；
12.　　根据式(13)、(14)更新策略网络；
13.　　根据式(10)更新策略网络；
14.　　更新目标 Q 网络；
15.　end if
16. until convergence

4　实验与分析

4.1　实验设置

为了验证 ESAC 算法的有效性和实用性，实验中首先构建四旋翼无人机仿真环境并对其物理结构进行建模，图 7 给出了仿真环境中的无人机结构。在建模中使用统一机器人描述格式(Unified Robot Description Format，URDF)文件对四旋翼无人机进行表示，并基于 pybullet 机器人仿真平台搭建仿真环境，构建强化学习框架，使得仿真环境能够为 Agent 实时提供无人机的状态(位置、姿态、速度、角速度)信息，同时也能够接收 Agent 反馈的动作信息并应用于仿真环境之中，以改变无人机的状态。

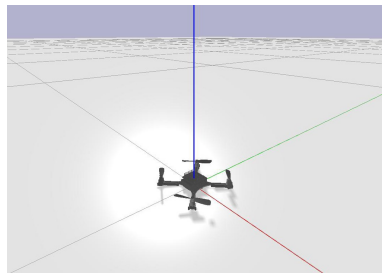


图 7　四旋翼无人机结构。

Fig. 7　Structure of quadrotor

对于无人机的机型，实验中采用与 Crazyfile 机型相同的仿真无人机，同时对与本实验无关的物理特性进行简化。在仿真中采用的无人机重量仅为 28g，其具体的物理特性如表 1 所列。对于专家策略，实验中采用 PID 控制算法，其能够根据无人机当前状态和期望状态给出控制量。对于控制频率，实验中使用的网络控制器的控制频率设置为 240 Hz。本文所有实验均在 PyTorch(Paszke et al. 2019) 中实现，并在 NVIDIA GTX 2080 GPU 上进行。

表 1　无人机物理学特性

Table 1　Physical properties of quadrotor

weights/g	I_{xx}，I_{yy}，I_{zz}/(kgm^2)
28	0.000014，0.000014，0.000022

为提高网络控制器的通用性，即构建的模型控制器在不同的场景下依然能够发挥作用，实验中在强化学习的每轮训练过程中随机初始化无人机状态(包括位置、姿态)和目标点位置，当 Agent 不断与环境交互到达目标点后，随机设置下一个目标点位置，直到 Agent 飞出设定区域范围或坠毁，本轮训练结束进入下一轮训练过程。对于到达目标点的定义采用式(15)的形式，即无人机的位置 p_c 和目标位置 p_t 的距离在一定范围内时就认为无人机已经到达期望位置，此时随机设置下一目标点位置。

$$\|p_c-p_t\|<\xi \tag{15}$$

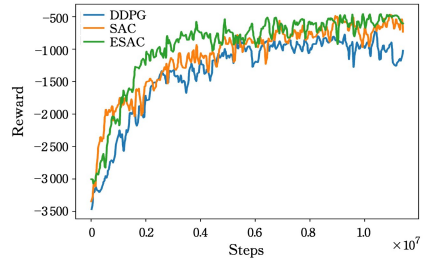


图 8　累计回报奖赏学习曲线

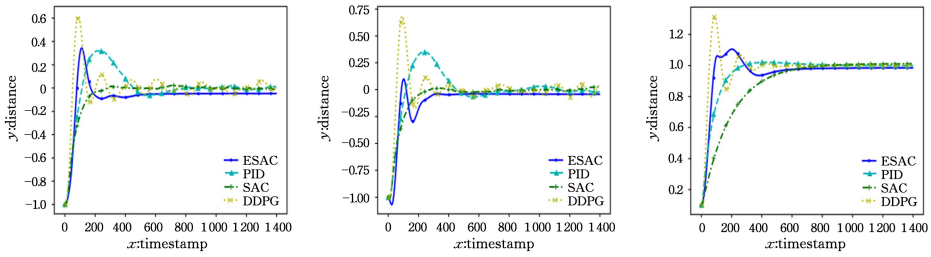
Fig. 8　Learning curve of accumulated reward

为了对 ESAC 算法的有效性进行有效评估，实验中采用的对比方法为 PID 算法和强化学习中的 SAC、DDPG 算法[27]。同时对于无人机飞行任务设置，实验中采用位置控制和轨迹跟踪两种控制任务，其中位置控制任务为 Agent 通过不断与环境进行交互将无人机悬停在期望位置；轨迹跟踪任务为 Agent 根据给定的期望轨迹点调整舵机转速以跟踪目标点，实验中采用的期望轨迹为圆形和正方形。

4.2　实验结果与分析

实验结果如图 8－图 10 所示，图 8 给出了 DDPG，SAC

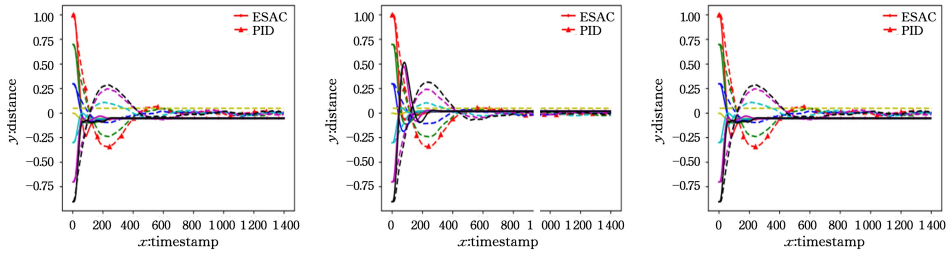
和 ESAC 3 种算法的平均累计奖赏随迭代轮次增加的变化曲线。图 9 给出了通过不同的强化学习算法构建的决策模型在位置控制任务中的控制效果，其中实线为 ESAC 算法构建的控制器，虚线分别为 PID，SAC，DDPG 构建的控制器。图 10 给出了对于不同的初始位置，在位置控制任务中无人机的位置变化范围，其中实线为 ESAC 控制效果，虚线为 PID 控制器的控制效果。图 10 给出了不同的控制算法在轨迹跟踪任务中的控制表现。



(a) position x (b) position y (c) position z

图 9 不同强化学习模型的控制效果

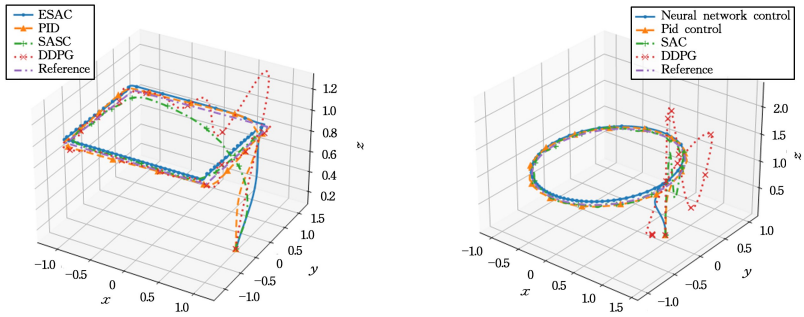
Fig. 9 Control effects of different RL models



(a) position x (b) position y (c) position z

图 10 ESAC 和 PID 的控制效果

Fig. 10 Control effects of ESAC and PID



(a)正方形轨迹 (b)圆形轨迹

图 11 无人机轨迹跟踪任务

Fig. 11 Trajectory tracking task

从图 8 可以看出，随着迭代轮次的增加，ESAC 相较于 SAC 和 DDPG 能够更快地收敛且获得更高的平均累计奖赏，这表明引入的专家知识对智能体的探索起到正向引导作用，使其能够快速学到一个较好的策略。

从图 9 中可以看出不同控制器在位置控制任务中的控制效果，对于无人机的位置调节，ESAC 算法构建的控制器相比专家 PID 控制器能够将无人机更快更稳定地调节到目标位置，同时相较于 SAC 和 DDPG 算法构建的控制器在调节过程中其波动更小且更加稳定。图 10 给出了在位置控制任务中随机选取 7 个初始点的前提下 ESAC 和 PID 的调节效果，PID 控制器和 ESAC 算法构建的控制器均能完成调节任务，但相较于 PID 控制器，ESAC 可以在更短的时间内将无人机悬停在期望位置且在调整过程中网络控制器的波动较小，不会在目标位置附近抖动。而同样如图 11 所示，在无人机的轨迹跟踪任务中，在 ESAC 构建的控制器的调节下，无人机的轨迹更接近于期望轨迹，同时在控制性能上达到了与 PID 控制器性能相近甚至优于其控制性能的效果，而 PID 控制器则需要复杂的参数调节以及不断尝试才能达到相同的控制效果。对于 SAC 和 DDPG 构建的控制器，由于响应较慢，波动较大，其轨迹与期望轨迹相差较大，最终在圆形轨迹的跟踪任务中 DDPG 控制器没能完成调节，无人机直接坠毁。

因此综合以上两组实验可以看出，文中提出的 ESAC 算法在传统 SAC 的基础上引入专家指导，使得决策策略在训练过程中能更快速收敛并获得更高的累积奖赏，具有收敛速度快、探索空间小、平均累计奖赏高等特点。同时，相较于 PID 算法，ESAC 不需要基于经验的参数调节和动力学模型构建，通过简单的训练即可达到与专家控制器性能接近甚至优于其控制性能的控制效果。

结束语 本文提出了一种基于强化学习的端到端的无人机自主控制算法，与专家控制器相比，网络控制器不需要复杂

的参数调节以及模型构建，直接将无人机的状态映射到舵机的输出，减少了中间的复杂计算过程。同时为了验证本文算法的有效性，设计了两组实验，分别在位置控制以及轨迹跟踪任务中对 ESAC 算法的控制性能进行了验证。实验结果表明，通过 ESAC 算法构建的控制器能够达到与 PID 控制器同样的控制效果，同时在稳定性和准确性上优于 SAC 和 DDPG 构建的模型控制器。未来将进一步深入研究网络控制器并将其应用于真实无人机中。

参 考 文 献

[1] MOAD I, SALAMI M, ANNAZ F, et al. A Review of Quadrotor Unmanned Aerial Vehicles: Applications, Architectural Design and Control Algorithms[J]. Journal of Intelligent & Robotic Systems, 2022, 104(2): 1-33.

[2] ANG K H, CHONG G, LI Y. PID control system analysis, design, and technology[J]. IEEE Transactions on Control Systems Technology, 2005, 13(4): 559-576.

[3] GARCIA C E, PRETT D M, MORARI M. Model predictive control: Theory and practice—A survey[J]. Automatica, 1989, 25(3): 335-348.

[4] ARGENTIM L M, REZENDE W C, SANTOS P E, et al. PID, LQR and LQR-PID on a quadcopter platform[C]// 2013 International Conference on Informatics, Electronics and Vision (ICIEV). IEEE, 2013: 1-6.

[5] EMRAN B J, NAJJARAN H. A review of quadrotor: An underactuated mechanical system[J]. Annual Reviews in Control, 2018, 46: 165-180.

[6] YU X, FAN Y, XU S, et al. A self-adaptive SAC-PID control approach based on reinforcement learning for mobile robots[J]. International Journal of Robust and Nonlinear Control, 2021.

[7] WILLIAMS G, WAGENER N, GOLDFAIN B, et al. Information theoretic MPC for model-based reinforcement learning[C]// 2017 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2017: 1714-1721.

[8] HWANGBO J, SA I, SIEGWART R, et al. Control of a quadrotor with reinforcement learning[J]. IEEE Robotics and Automation Letters, 2017, 2(4): 2096-2103.

[9] KOCH W, MANCUSO R, WEST R, et al. Reinforcement learning for UAV attitude control[J]. ACM Transactions on Cyber-Physical Systems, 2019, 3(2): 1-21.

[10] LEWIS F L, VRABIE D, VAMVOUDAKIS K G. Reinforcement learning and feedback control: Using natural decision methods to design optimal adaptive controllers[J]. IEEE Control Systems Magazine, 2012, 32(6): 76-105.

[11] MEIER L, HONEGGER D, POLLEFEYS M. PX4: A node-based multithreaded open source robotics framework for deeply embedded platforms[C]// 2015 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2015: 6235-6240.

[12] NG T C T, LEUNG F H F, TAM P K S. A simple gain scheduled PID controller with stability consideration based on a grid-point concept[C]// Proceeding of the IEEE International Symposium on Industrial Electronics (ISIE'97). IEEE, 1997: 1090-1094.

[13] PAPADOPOULOS K G, TSELEPIS N D, MARGARIS N I. On the automatic tuning of PID type controllers via the magnitude optimum criterion[C]// 2012 IEEE International Conference on Industrial Technology. IEEE, 2012: 869-874.

[14] PI C H, HU K C, CHENG S, et al. Low-level autonomous control and tracking of quadrotor using reinforcement learning[J]. Control Engineering Practice, 2020, 95: 104222.

[15] XIE L, WANG S, ROSA S, et al. Learning with training wheels: speeding up training with a simple controller for deep reinforcement learning[C]// 2018 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2018: 6276-6283.

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

[17] PANERATI J, ZHENG H, ZHOU S Q, et al. Learning to fly—a gym environment with pybullet physics for reinforcement learning of multi-agent quadcopter control[C]// 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2021: 7512-7519.

[18] LoPES G C, FERREIRA M, DA SILVA SIMOES A, et al. Intelligent control of a quadrotor with proximal policy optimization reinforcement learning[C]// 2018 Latin American Robotic Symposium, 2018 Brazilian Symposium on Robotics (SBR) and 2018 Workshop on Robotics in Education (WRE). IEEE, 2018: 503-508.

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

[20] FAN D D, AGHA-MOHAMMADI A, THEODOROU E A. Deep learning tubes for tube mpc[J]. arXiv: 2002.01587, 2020.

[21] BIEKER K, PEITZ S, BRUNTON S L, et al. Deep model predictive flow control with limited sensor data and online learning[J]. Theoretical and Computational Fluid Dynamics, 2020, 34(4): 577-591.

[22] LENZ I, KNEPPER R A, SAXENA A. DeepMPC: Learning deep latent features for model predictive control[C]// Robotics: Science and Systems. 2015.

[23] KABZAN J, HEWING L, LINIGER A, et al. Learning-based model predictive control for autonomous racing[J]. IEEE Robotics and Automation Letters, 2019, 4(4): 3363-3370.

[24] TORRENTE G, KAUFMANN E, FÖHN P, et al. Data-driven MPC for quadrotors[J]. IEEE Robotics and Automation Letters, 2021, 6(2): 3769-3776.

[25] LAMBERT N O, DREW D S, YACONELLI J, et al. Low-level control of a quadrotor with deep model-based reinforcement learning[J]. IEEE Robotics and Automation Letters, 2019, 4(4): 4224-4230.

[26] HAARNOJA T, ZHOU A, HARTIKAINEN K, et al. Soft actor-critic algorithms and applications[J]. arXiv: 1812.05905, 2018.

[27] JOSHUA A. Spinning Up in Deep Reinforcement Learning[OL]. https://github.com/openai/spinningup.

LIANG Ji, born in 1996, postgraduate. His main research interests include adaptive UAV control and reinforcement learning.

QIN Xiaolin, born in 1953, Ph.D, professor, is a member of China Computer Federation. His main research interests include data management, unmanned system and security in distributed environment.