计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
Computer Systems & Applications,2022,31(7):93−98 [doi: 10.15888/j.cnki.csa.008591]

E-mail: csa@iscas.ac.cn
http://www.c-s-a.org.cn
Tel: +86-10-62661041

基于强化学习算法的智能飞控开发系统①

罗　杰[1], 董志岩[1,2], 翟　鹏[1], 张立华[1,2]

[1](复旦大学 工程与应用技术研究院, 上海 200433)
[2](季华实验室, 佛山 528200)
通信作者: 董志岩, E-mail: dongzhiyan@fudan.edu.cn

摘　要: 无人机控制器的设计开发是一项复杂的系统工程, 传统的基于代码编程的开发方式存在开发难度大、周期长及错误率高等缺点. 同时, 强化学习智能飞控算法虽在仿真中取得很好的性能, 但在实际中仍缺乏一套完备的开发系统. 本文提出一套基于模型的智能飞控开发系统, 使用模块化编程及自动代码生成技术, 将强化学习算法应用于飞控的嵌入式开发与部署. 该系统可以实现强化学习算法的训练仿真、测试及硬件部署, 旨在提升以强化学习为代表的智能控制算法的部署速度, 同时降低智能飞行控制系统的开发难度.

关键词: 无人机; 强化学习; 智能控制; 基于模式设计; 开发系统; 航迹规划

引用格式: 罗杰,董志岩,翟鹏,张立华.基于强化学习算法的智能飞控开发系统.计算机系统应用,2022,31(7):93–98. http://www.c-s-a.org.cn/1003-3254/8591.html

Intelligent Flight Control Development System Based on Reinforcement Learning

LUO Jie[1], DONG Zhi-Yan[1,2], ZHAI Peng[1], ZHANG Li-Hua[1,2]

[1](Academy for Engineering and Technology, Fudan University, Shanghai 200433, China)
[2](Ji Hua Laboratory, Foshan 528200, China)

Abstract: The design and development of unmanned aerial vehicle (UAV) controllers are complex system engineering. The traditional development method based on code programming has the disadvantages of difficult development, long cycle, and high error rate. Although the intelligent flight control algorithm based on reinforcement learning has achieved good performance in simulation, it still lacks a complete development system in practice. This study presents a model-based development system for intelligent flight control, applying the reinforcement learning algorithm to the embedded development and deployment for flight control with modular programming and automatic code generation technologies. The system is equipped for the training simulation, testing, and hardware deployment of the reinforcement learning algorithm, and it is expected to improve the deployment speed of intelligent control algorithms represented by reinforcement learning and to reduce the development difficulty of intelligent flight control systems.

Key words: unmanned aerial vehicle (UAV); reinforcement learning; intelligent control; model-based design; development system; path planning

随着科技的发展, 无人驾驶飞行器 (UAV) 开始在各种复杂场景中取得应用[1−4], 由于无人机具有体积小、质量轻、机动性好、易于控制、造价相对较低、危险系数小以及隐蔽性能好等优点, 在军事和民用领域都具有广泛的应用前景. 因此国内外均对无人机的机体结构及飞行控制展开了深入的研究, 并取得了不错的成果[5,6].

传统的无人机飞行控制器多采用比例-积分-微分

① 基金项目: 广东省基础与应用基础研究基金 (2019A1515110352); 季华实验室开放课题 (X190021TB194); 科技创新 2025 重大专项 (2020Z073)
收稿时间: 2021-10-18; 修改时间: 2021-11-17; 采用时间: 2021-11-30; csa 在线出版时间: 2022-05-31

(PID) 控制算法, 这种基于 PID 算法在稳定环境中可以达到很好的控制性能, 然而在面临复杂场景时, 往往容易受到外界干扰的影响, 且无法保证稳定飞行. 这对飞行控制器的创新提出了更高的要求, 最近的研究表明[7–9], 基于强化学习的智能控制算法在仿真中表现出了极好的性能, 这为无人机飞控开发提供了新的方向. 目前的行业痛点是在实际中仍然缺乏一套快速的智能无人机飞控开发系统.

为了提高无人机智能飞行控制器的开发速度, 本文提出一种基于模型的智能飞控开发系统. 该系统可以实现强化学习控制算法的仿真测试及快速硬件部署, 控制器开发采用基于模型的设计方式, 可以有效避免代码编程方式的弊端, 并大大提高控制器开发速度. 本研究还提供了一套仿真测试平台, 我们将开发的控制器在仿真平台和真实环境中进行飞行测试, 验证了该开发系统的有效性.

1 相关工作

智能飞行控制系统的开发是一个亟待解决并突破的研究领域[10], 研究表明, 强化学习是实现飞控智能化的一个重要途径, 目前基于强化学习实现无人机控制的理论研究已取得了突出的成果[7,11].

基于强化学习的智能算法具备实现飞行控制的仿真与验证. Koch 等人利用强化学习近端策略优化 (PPO) 算法[12] 实现了无人机仿真控制, 经过训练的无人机姿态控制器在仿真环境中可以实现稳定飞行并表现出了超过 PID 控制器的性能. 文献 [13] 中提出了一种基于强化学习的新误差卷积神经网络控制器设计方法, 并用于复合式无人机的飞行控制, 该研究缩小了虚拟仿真和真实环境之间的控制性能的差距, 实现了强化学习在实际环境中的应用. 文献 [14] 中提出了一种新的强化学习控制算法, 该算法比现有的算法更适用于控制四旋翼飞行器, 特别是在非常苛刻的初始化条件下, 仍可以自动调整四旋翼飞行器处于稳定的悬停状态. 更加令人瞩目的是, 文献 [15] 以强化学习理论为基础, 提出了一种用于训练神经网络仿真并将其编译为可在嵌入式硬件上运行的工具链, 但是开发方式仍为代码式编程, 对智能飞控的开发需要很高的门槛.

尽管强化学习智能飞控算法已在仿真中取得突出成就, 但在实际中仍缺乏一套完备的强化学习飞行控制器开发平台.

2 强化学习控制器开发平台

本文在智能飞控领域已有研究的基础上, 针对目前智能飞控开发存在的痛点, 提出了一套完备的无人机智能飞行控制开发系统, 整个系统框架如图 1 所示.

图 1 系统架构图

2.1 系统架构

本系统采用 4 层架构的模式, 分别为: (1) 数据存储层; (2) 强化学习网络训练层; (3) 控制器开发层; (4) 硬件测试层. 其中数据存储层根据不同数据类型分数据库存储, 对于关系型数据, 如不同飞行器模型及不同强化学习超参数对应的控制器性能, 存储在 MySQL 数据库中. 对于非关系型数据使用 MongoDB 数据进行存储, MongoDB 数据库是一个基于分布式文件存储的数据库, 适用于数据量大的存储场景. 在本系统中, 需要使用服务器进行强化学习训练, 每次训练的神经网络参数, 采用 MongoDB 分布式集群的存储方式.

强化学习网络训练层是指进行强化学习控制器网络训练的层, 本层采用强化学习作为飞行控制器, 需要有一个通用的训练环境来进行强化学习训练. 系统选用戴尔 R940 服务器来搭建仿真训练环境, 并在 Gazebo 仿真模拟器中建立了一个四旋翼模型, 该模型可以根据强化学习神经网络输出的电机控制量, 在俯仰、横

滚、偏航 3 个方向上改变四旋翼姿态. 控制器开发层主要使用基于模式的设计方法 (MDB), 利用 Simulink 提供的无人机自驾仪开发支持包 (Pixhawk pilot support package, PSP) 进行控制器设计, 并利用自动代码生成技术将控制器部署到 Pixhawk 硬件中. 下面分层介绍整个系统的实现原理.

2.2 强化学习训练层

强化学习算法的基本原理是通过让智能体与环境不断交互来学习最优策略, 以实现回报最大化或完成特定目标. 整个交互过程如图 2 所示, 在某一时刻t, 智能体从环境中获得状态值S_t, 根据当前状态值并经过特定策略的评估, 执行最优动作并获取下一时刻的状态值S_{t+1}. 其中状态转换定义为转换到状态s'的概率, 即当前状态和动作分别为s和a, 转换到状态s'的概率可以表示为$p_r\{s_{t+1}=s'|s_t=s,a_t=a\}$. 智能体的行为由其策略$\pi$定义, 该策略$\pi$本质上是对特定状态应采取动作的映射.

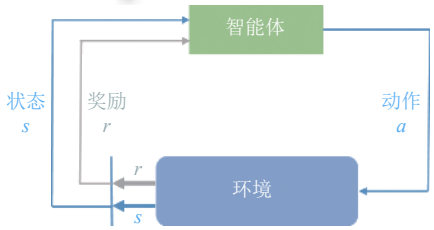

图 2 强化学习交互图

本系统在搭建训练平台时, 将环境建模为一个四旋翼无人机模型, 用于模拟无人机在无重力条件下的飞行 (模拟重力只需要在垂直方向加一个力的分量, 在进行强化学习训练时, 去除重力作用可以规避很多不必要的问题, 后续实验只需要平衡重力即可). 如图 3 所示, 整个仿真环境利用 Gazebo 仿真模拟器完成搭建, 其中无人机模型符合动力学特性, 可以根据输入的信号驱动电机并改变飞行姿态.

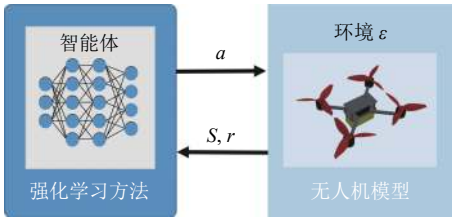

图 3 强化学习仿真环境中神经网络与无人机模型交互图

根据强化学习算法的理论基础, 将智能体建模为一个 4 层的强化学习神经网络, 其中输入层有 9 个节点, 代表从 Gazebo 环境中获取 9 个状态信息; 输出层有 4 个节点, 代表电机的 4 个输出控制信号; 中间是 2 层具有 32 个节点的隐藏层, 整体构成了强化学习控制器网络.

神经网络控制器以无人机飞行时的角度、角速度、角速度误差组成了 9 维矢量作为输入, 将无人机的输入状态向量定义为:

$$s=\left(\phi,\theta,\omega,\beta,\gamma,D_u,D_v,D_w,D_\psi\right)^{\mathrm{T}}\in R^9$$

其中, ϕ,θ分别表示俯仰角和横滚角, ω,β,γ分别表示无人机的三轴角速度, D_u,D_v,D_w是地方坐标系中期望速度与当前速度分量之间的差异, D_ψ是目标偏航角与当前偏航角之间的差异.

在强化学习智能体网络的训练中, 本实验使用近端策略优化 (PPO) 算法, 该算法在强化学习领域有着广泛的应用, 在运动控制领域中具有成功的先例 (如半猎豹实验, 足式机器人等). 同时, OpenAI 的 Baselines 项目[16] 中提供了 PPO 算法的通用 API, 本文直接使用 Baselines 提供的 PPO 算法训练神经网络.

在每一个训练步骤中, 使用智能体网络指定的动作在 Gazebo 模型中执行一个模拟步骤, 每个模拟步骤需要返回一个奖励以评估给定动作的执行情况. 本文在每个模拟步骤的强化学习奖励函数由 3 部分组成: 飞行时长、飞行稳定性和速度跟踪误差. 飞行时长和飞行稳定性这两项可以使飞行器在保持稳定飞行的同时, 尽可能飞行更长的时间, 速度跟踪误差用来衡量智能体对输入指令的跟踪情况. 因此, 本文将奖励函数定义为:

$$R=r-sum\left(\left\|v_t-v_{\text{target}}\right\|\right)-\|\omega\|^2$$

其中, r是一个不变的存活奖励, 用来反映飞行器飞行的时长, 每个时间步不断累加, 飞行的时间越久, 累积奖励越大, 这有利于获得更长的飞行时间. $-\|\omega\|^2$项通过最小化角速度来防止机体抖动, 以尽可能使机体保持稳定飞行. $-sum\left(\left\|v_t-v_{\text{target}}\right\|\right)$项求和每个速度分量误差的绝对值, 由于奖励是负数, 该项表示惩罚, 以最小化跟踪误差, 从而尽可能准确地跟踪目标速度.

整个训练过程在一台拥有 72 核 CPU 和 250 GB 内存的戴尔 R940 服务器上进行, 通过使用并行计算, 训练 100 万步的 PPO 算法大约需要 1 h. 通过记录每个训练周期智能体获得的奖励, 可以得到如图 4 所

示的 reward 曲线图, 通常在训练结束之前就实现了收敛.

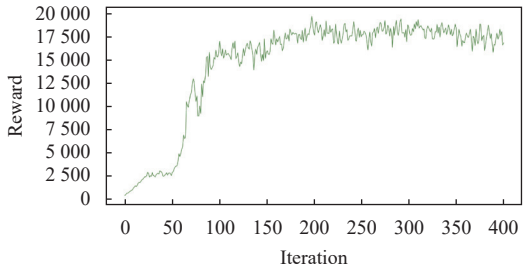

图 4 强化学习训练过程中 reward 曲线图

在每个交互周期中, 强化学习神经网络根据状态输入获得 4 个电机的输出值, 并作用于无人机动力学模型, 无人机根据不同的控制量来调节飞行姿态, 以减小实际速度与期望速度之间的误差并获得最优的奖励回报. 经过不断的训练优化, 使强化学习神经网络得到收敛, 我们将调节的超参数及网络控制性能保存到 MySQL 数据库中, 将训练好的网络参数存储到 MongoDB 数据库中, 以供控制器开发平台使用.

2.3 控制器开发层

本层将训练完成的强化学习神经网络参数用于无人机飞行控制器的设计实现, 整体设计采用基于模型设计方式取代传统代码编程的方式. 基于模型设计将敏捷原则延伸到包括物理组件和软件在内的系统开发工作, 从需求捕获、系统架构和组件设计, 到实现、验证、测试和部署, 基于模型设计可以贯穿整个开发周期.

通过手动编码来开发复杂的飞行控制器是一项艰难而又不可靠的任务, 难以避免编码错误、逻辑错误或未知漏洞带来的不正确的结果. Simulink 是一款值得信赖的 MBD 开发工具, 通过模块化编程来避免手动编码开发存在的问题, 为飞行控制器的开发提供了捷径. 除此之外, Simulink 具有的自动代码生成能力可以根据模块化的控制器自动生成可执行的控制器软件, 实现强化学习控制算法的快速部署. 因此, 本层使用 Simulink 来进行控制器开发.

如图 5 所示, 基于强化学习的飞行控制器主要包括以下几个模块: 控制信号输入模块、神经网络参数接口模块、计算网络输入模块以及强化学习控制系统模块, 模块内部采用独立的子系统, 分别设计以完成特定的内部功能. 其中信号输入模块读取遥控器 RC 信号, 遥控器的控制信号主要是对无人机机体速度、姿态角以及油门驱动的控制, 同时将归一化后的控制信号传递到网络计算模块进行当前状态值的计算. 网络计算模块根据控制信号输入以及传感器获取的无人机姿态角数据计算出强化学习神经网络的状态输入, 即 $s=\left(\phi,\theta,\omega,\beta,\gamma,D_u,D_v,D_w,D_\psi\right)^{\mathrm{T}}\in R^9$, 并作为当前时刻的状态量输入到强化学习控制系统中.

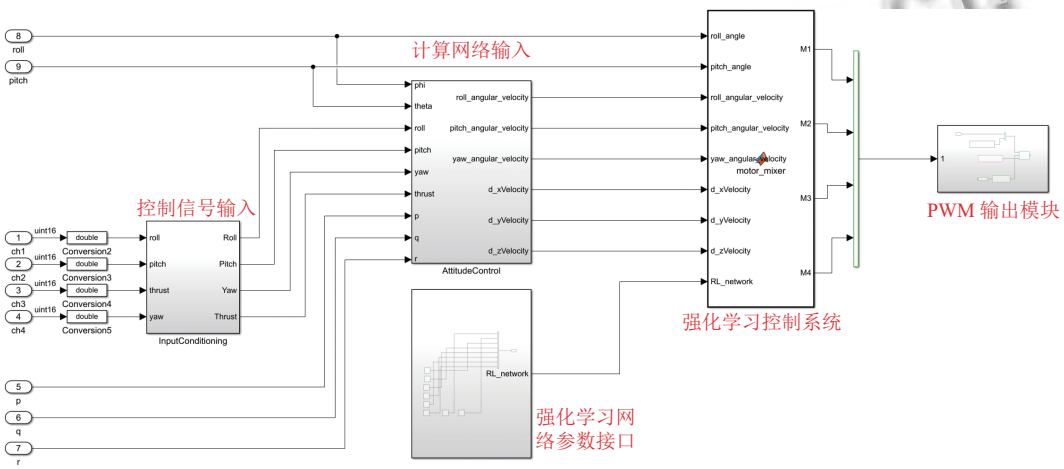

图 5 基于模型的智能飞控设计

在 Simulink 模型中, 需要一个“控制器参数接口”模块用于接收从仿真计算机获得的神经网络参数. 我们将仿真环境中训练好的网络参数保存到 Matlab 可以读取的“.mat”文件中, 并通过 TCP/IP 通信来连接主机和服务器, 将参数导入到“控制器参数接口”模块. 最终, 网络参数与状态输入信号一起传入强化学习控制系统模块中, 在每一次循环中, 该模块根据获取到的状态输入s, 经过强化学习神经网络的前向传递, 计算出

下一步采取的动作a. 动作a对应的是四旋翼 4 个电机的输出油门控制量, 油门控制量可直接作用于无人机并通过改变电机转速来改变四旋翼的飞行姿态, 驱动无人机进行飞行并进行姿态控制.

3 最终成果

为了验证强化学习飞行控制器的性能, 我们搭建了一套半实物仿真测试平台和硬件测试平台, 并进行仿真测试实验以及真机飞行实验.

3.1 半实物仿真测试平台

硬件在环仿真 (HIL) 利用硬件在仿真实验系统的回路中进行实时仿真, 通过在计算机仿真回路中加入一些实物, 并建立数学模型, 将实物的动态特性和物理规律在计算机上运行试验, 从而为物理部件创造一个仿真环境.

硬件在环仿真系统已被证明在加快无人驾驶飞行器的控制系统的开发速度方面的有效性, 在无人机控制器设计完成之后, 需要测试验证控制器的控制性能, 若直接进行无人机实物测试, 可能会发生无法预测的故障. 为了避免安全问题, 可以先进行 HIL 实验, 来测试控制器的控制性能. 这是控制器测试的第一步, 根据 HIL 测试的结果, 可对控制器进行适当的调整.

如图 6 所示, 本研究以现有的无人机仿真软件为基础搭建仿真平台, 仿真环境中包含一架小型四旋翼无人机模型, 为了与强化学习训练环境中的四旋翼无人机保持尽可能的一致, 实验时选用重量为 440 g, 轴距为 225 mm 的“X”结构四旋翼无人机. 同时, 无人机在仿真环境中飞行无气流、风力等环境因素的影响, 可以很好地规避其他因素对控制器性能的影响.

图 6 半实物仿真测试平台

HIL 实验中, 首先将开发的强化学习控制器部署到 Pixhawk 硬件, 并将 Pixhawk 硬件与无人机仿真软件建立连接, 之后通过遥控器控制飞控硬件发出驱动信号, 并控制仿真无人机飞行. 最后, 可以在仿真平台中观察无人机的各项飞行数据及飞行轨迹, 并进行分析实验. 软件界面中, 通过三维场景视窗可以观察无人机在仿真环境中的位置和姿态; 轨迹视窗可以记录无人机在仿真环境中的水平飞行轨迹; 参数视窗用于记录无人机在飞行过程中电机转速、姿态角数据、速度数据以及位置数据. 记录实时采集的数据, 并进行对比实验, 最终用于验证所提出开发平台的性能.

开发的半实物仿真测试平台可以替代真实无人机进行控制器性能实验, 在仿真环境中可以规避突发的安全问题和无法预测的故障, 可以作为控制器测试的第一步.

3.2 真机测试平台

经过第一步控制器的硬件在环测试后, 需要进行真机测试, 这样才能进一步验证开发的控制器在真实环境中的可用性. 本研究开发的智能飞行控制器可以通过自动代码生成将控制器固件部署到 Pixhawk 硬件中, 并安装在真实四旋翼无人机上飞行. 如图 7 所示, 搭建了针对特定四旋翼无人机的硬件测试平台, 图 7(a) 包括一个用于测试无人机飞行姿态角的云台装置, 可将无人机安装在云台上固定, 并测试记录在飞行过程中的姿态角, 用于对控制器跟踪性能的分析. 图 7(b) 是在一个小型四旋翼无人机上进行的飞行测试, 我们让飞手在空旷地带控制四旋翼无人机飞行, 可以看出本研究提出的智能飞控开发系统可以在实际中使用, 并具有很好的控制性能.

(a) 云台测试装置 (b) 真机飞行测试

图 7 控制器真机测试

4 结论与展望

本文提出了一套完备的无人机智能飞行控制系统仿真、测试及部署的一体化平台. 基于 MBD 开发工具, 使用模块化编程以及自动代码生成技术将强化学习算法部署到 Pixhawk 硬件中, 并实现了真实无人机的飞行测试. 该平台可大大减小智能控制器开发成本以及规避代码开发中的错误. 未来的工作中, 我们将进

一步拓展平台的功能, 以适用于不同无人机机型的飞控开发. 同时将部署平台与更多硬件连接交互, 以实现各种复杂的智能控制系统, 让强化学习控制算法在实际中得到更好的应用.

参考文献

1 陈帅, 尹洋, 杨全顺. 基于深度学习的无人机入侵检测方法. 计算机系统应用, 2021, 30(4): 32–38. [doi: 10.15888/j.cnki.csa.007894]

2 张静, 张洁, 燕正亮, 等. 面向无人机风机巡检的光照条件分析方法. 计算机系统应用, 2021, 30(6): 162–167. [doi: 10.15888/j.cnki.csa.007977]

3 高尚文, 组家奎, 陶德臣. 无人机综合检测与仿真系统. 计算机系统应用, 2020, 29(10): 68–74. [doi: 10.15888/j.cnki.csa.007532]

4 刘松林, 朱永丰, 张哲, 等. 基于卷积神经网络的无人机油气管线巡检监察系统. 计算机系统应用, 2018, 27(12): 40–46. [doi: 10.15888/j.cnki.csa.006668]

5 周子栋, 陈至坤, 赵志佳. 四旋翼无人机飞控算法综述. 网络安全技术与应用, 2019, (9): 33–35. [doi: 10.3969/j.issn.1009-6833.2019.09.019]

6 Fan BK, Li Y, Zhang RY, *et al*. Review on the technological development and application of UAV systems. Chinese Journal of Electronics, 2020, 29(2): 199–207. [doi: 10.1049/cje.2019.12.006]

7 Koch W, Mancuso R, West R, *et al*. Reinforcement learning for UAV attitude control. ACM Transactions on Cyber-Physical Systems, 2019, 3(2): 1–21.

8 Zhang W, Song K, Rong XW, *et al*. Coarse-to-fine UAV target tracking with deep reinforcement learning. IEEE Transactions on Automation Science and Engineering, 2019, 16(4): 1522–1530. [doi: 10.1109/TASE.2018.2877499]

9 秦世引, 陈锋, 张永飞. 小型无人机纵向姿态模糊自适应 PID 控制与仿真. 智能系统学报, 2008, 3(2): 121–128.

10 Santoso F, Garratt MA, Anavatti SG. State-of-the-art intelligent flight control systems in unmanned aerial vehicles. IEEE Transactions on Automation Science and Engineering, 2018, 15(2): 613–627. [doi: 10.1109/TASE.2017.2651109]

11 张友安, 马国欣, 刘京茂, 等. 固定翼无人机强化学习控制建模与算法设计. 飞行力学, 2019, 37(4): 88–91, 96.

12 Schulman J, Wolski F, Dhariwal P, *et al*. Proximal policy optimization algorithm. arXiv: 1707.06347, 2017.

13 Xu J, Du T, Foshey M, *et al*. Learning to fly: Computational controller design for hybrid UAVs with reinforcement learning. ACM Transactions on Graphics, 2019, 38(4): 1–12.

14 Hwangbo J, Sa I, Siegwart R, *et al*. Control of a Quadrotor with reinforcement learning. IEEE Robotics and Automation Letters, 2017, 2(4): 2096–2103. [doi: 10.1109/LRA.2017.2720851]

15 Koch W, Mancuso R, Bestavros A. Neuroflight: Next generation flight control firmware. arXiv: 1901.06553, 2019.

16 Dhariwal P, Hesse C, Klimov O, *et al*. Openai baselines. https://github.com/openai/baselines.

(校对责编: 牛欣悦)