中国科学技术大学

硕士学位论文

基于训练和执行双阶段联合设计的人机智能决策方法研究

作者姓名： 李明

学科专业： 控制科学与工程

导师姓名： 康宇 教授 赵云波 教授

完成时间： 二〇二三年五月二十九日

University of Science and Technology of China

A dissertation for master's degree

Research on human-machine intelligence decision making method based on two-stage joint design of training and execution

Author: Li Ming

Speciality: Control Science and Engineering

Supervisors: Prof. Yu Kang, Prof. Yun-Bo Zhao

Finished time: May 29, 2023

中国科学技术大学学位论文原创性声明

本人声明所呈交的学位论文，是本人在导师指导下进行研究工作所取得的成果。除已特别加以标注和致谢的地方外，论文中不包含任何他人已经发表或撰写过的研究成果。与我一同工作的同志对本研究所做的贡献均已在论文中作了明确的说明。

作者签名：李明　　签字日期：2023年5月29日

中国科学技术大学学位论文授权使用声明

作为申请学位的条件之一，学位论文著作权拥有者授权中国科学技术大学拥有学位论文的部分使用权，即：学校有权按有关规定向国家有关部门或机构送交论文的复印件和电子版，允许论文被查阅和借阅，可以将学位论文编入《中国学位论文全文数据库》等有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存、汇编学位论文。本人提交的电子文档的内容和纸质论文的内容相一致。

控阅的学位论文在解密后也遵守此规定。

☑公开　□控阅（____年）

作者签名：李明　　导师签名：

签字日期：2023年5月29日　　签字日期：2023年5月29日

摘　要

在人机混合智能系统中，人工智能赋能的机器智能和人类智能相互融合，在特定场景下可以超越单独人类或者机器的决策性能，成为当前的研究热点。但是，与传统的人机系统和人工智能算法不同，人机混合智能系统的决策效果不仅受到训练阶段人工智能算法性能的影响，比如算法的泛化性和鲁棒性，而且也会受到执行阶段人类和机器决策混合方法的影响，比如人类和机器控制权的分配。如何从整体上优化人机混合智能系统的决策性能，是当下重要的研究课题。

本文面向深度强化学习算法驱动的人机混合智能决策系统的序贯决策问题，同时从算法的训练端和执行端出发，通过引入人类智能的方式提高系统决策的鲁棒性和安全性，最终提高人机混合智能系统的决策性能。本文工作主要包含以下三个方面：

(1) 针对强化学习算法驱动的人机共享控制系统的序贯决策问题，在训练阶段提出了基于人类策略限制下人在环上强化学习算法，避免机器做出危险的行为，同时提高了算法的采样效率；在执行阶段提出了包含人类决策评估的仲裁机制，舍弃了人类错误的决策，提高了系统的整体性能。实验结果表明，此方法成功提高了算法训练的采样效率和系统执行任务的成功率。

(2) 针对多机竞速场景下强化学习算法驱动的人机介入控制系统的序贯决策问题，在训练阶段引入了包含人类反馈奖励的奖励函数组，以引导机器理解竞速规则，减少了执行阶段人类的介入次数；在执行阶段引入了人类的两级介入机制，避免违背规则或者容易造成事故的行为出现，同时降低了人类介入时的操作负担。实验结果表明，此方法缩短了无人机的单圈耗时，提高了系统决策的安全裕度，并且减轻了人类的介入负担。

(3) 针对上述人机混合序贯决策方法，本文以旋翼无人机为背景，搭建了从仿真到现实的人机实验平台，提出了算法部署到真实物理场景的整体流程和框架，并针对提出的多机竞速场景下强化学习算法驱动的人机介入控制方法，进行了现实场景下的算法验证。

关键词：人机混合智能系统；强化学习；共享控制；介入控制；序贯决策

ABSTRACT

In human-machine hybrid intelligence systems, AI-enabled machine intelligence and human intelligence are integrated with each other and can surpass the decision making performance of individual human or machine in specific scenarios, which has become a current research hotspot. However, unlike traditional human-machine systems and AI algorithms, the decision making effect of human-machine hybrid intelligence systems is not only influenced by the performance of AI algorithms in the training phase, such as the generalization and robustness of the algorithms, but also by the hybrid approach of human and machine decision making in the execution phase, such as the allocation of human and machine control. How to optimize the decision-making performance of human-machine hybrid intelligent systems as a whole is an important research topic nowadays.

This thesis is oriented to the problem of sequential decision making in human-machine hybrid intelligent decision making system driven by deep reinforcement learning algorithm, while improving the robustness and safety of system decision making by introducing human intelligence from both the training side and execution side of the algorithm, and finally improving the decision making performance of human-machine hybrid intelligent system. The work in this thesis contains the following three main aspects:

(1) For the sequential decision problem of human-machine shared control system driven by reinforcement learning algorithm, a human-in-the-loop reinforcement learning algorithm based on human policy constraints is proposed in the training phase to avoid dangerous behaviors of the machine while improving the sampling efficiency of the algorithm; an arbitration mechanism including human decision evaluation is proposed in the execution phase to discard the human wrong decisions and improve the overall performance of the system. Experimental results show that this method successfully improves the sampling efficiency of algorithm training and the success rate of the system in executing tasks.

(2) For the sequential decision problem of human-machine traded control system driven by reinforcement learning algorithm in the multi-drone racing scenario, a reward function groups containing human feedback rewards is introduced in the training phase to guide the machine to understand the racing rules and reduce the number of human interventions in the execution phase; A two-level human intervention mechanism is intro-

duced in the execution phase to avoid the appearance of rule-breaking or accident-prone behaviors, and to reduce the operational burden of human intervention. The experimental results show that this method shortens the lap time of the drone, improves the safety margin of the system decision, and reduces the burden of human intervention.

(3) For the above human-machine hybrid decision-making method, this thesis builds a human-machine experimental platform from simulation to reality in the context of multirotor UAVs, proposes an overall process and framework for algorithm deployment to real physical scenarios, and conducts algorithm validation in realistic scenarios for the proposed reinforcement learning algorithm-driven human-machine traded control method in multi-drone racing scenarios.

Key Words: Human-machine hybrid intelligent system；Reinforcement learning；Shared control；Traded control；Sequential decision-making

目　录

插 图 清 单

表 格 清 单

第 1 章　绪　　论

本章首先介绍了人机混合智能系统的诞生背景和研究意义，然后阐述了人工智能赋能下的人机混合智能系统的研究现状和问题，最后总结了本文的主要工作，并说明了全文的结构安排。

1.1　研究背景和意义

随着自动化水平的不断发展，人类正在逐渐的从繁杂的劳动中解放出来，可以发现的是大多数能够通过自动化设备独立完成的任务中，人类的身影在渐渐淡出：洗碗机、扫地机器人等家用机器人，汽车和飞机等出行工具，以及工厂的全自动生产流水线等。自动化的经典目标是用自动化设备和计算机取代人工控制、计划和解决问题[1]，所以大部分的自动化控制系统都不将人类视作系统的一部分。但是，一方面，目前的自动化控制系统还是难以在大多数的领域中替代人类；另一方面，即使是对于高度自动化的系统[2]，比如电力网络，系统的监督、调整、维护、扩展和改进都离不开人类。因此，将人类和自动化机器视作一个有机整体来研究是必要的。

广义来说，人类与机器相互依存、影响、协同而构成的整体被称为“人机系统”。人机系统将人类和机器分别视作两个动态的单元或者是不同的团队，通过协作的方式实现一个总体的任务和目标，其中包括参与者之间的动态任务分配[3]。传统自动化系统的控制任务是静态的，系统是根据确定性的任务而设计的。但是，大多数人机系统所考虑的场景更为复杂，任务往往是动态的，可能会出现系统设计者无法提前建模的情况。此时系统只有具有一定的自由度，才能够应对设计者无法预见的情况。在人类与机器的交互过程中，一方面，人类所做出的决策需要和机器结合；另一方面，机器做出的自主行动具有一定自由度的情况下，机器可能会引发意料之外的危险事故，此时需要引入人类的监督和控制。

现有研究大部分集中在人机系统中人类和机器交互所带来的复杂性和难度[4-6]，本文更为关注其中两个主要的复杂性来源：不确定性和风险。首先，人机系统的任务往往具有动态性，而动态性本身包含着不确定性，即机器无法完全控制所有的情况，意外的因素随时可能会影响系统的动态变化过程，内部的突发事件甚至会直接改变整个系统过程。而这种不确定性，很多情况下是不能通过概率学建模的。其次，动态性本身也意味着风险，机器在假设外的场景中可能会做出错误的决策，此时对于机器而言行动是具有风险性的，对于人类操作员而言控制的风险是如果管理不当可能会失去对局势的把控。因此在进行人机系统决策

的相关研究时，人机交互过程中存在的不确定性和风险是不可被忽视的。

近些年来，人机交互与合作的相关研究取得了较大的发展。目前，随着以深度学习为核心的人工智能 (Artificial Intelligence，AI) 算法不断革新，机器的自主能力进一步得到加强，这进一步推动了人机混合智能系统的诞生。与传统的人机系统相比，人机混合智能系统中的机器具有人工智能算法赋予的智能和自主性[7]。在这类系统中，机器在原本拥有自动化能力的基础上，进一步具备自主能力，整个人机系统的能力也将会进一步得到加强。如图1.1，以高级辅助驾驶系统 (Advanced Driver Assistance System，ADAS) 为例，汽车原本的按照指令进行驾驶的能力可以视作自动化能力，而通过人工智能算法训练后机器能够进行环境感知和路径规划的能力可以视作自主能力，在两者能力的加持下，原本系统的能力得到了进一步增强。

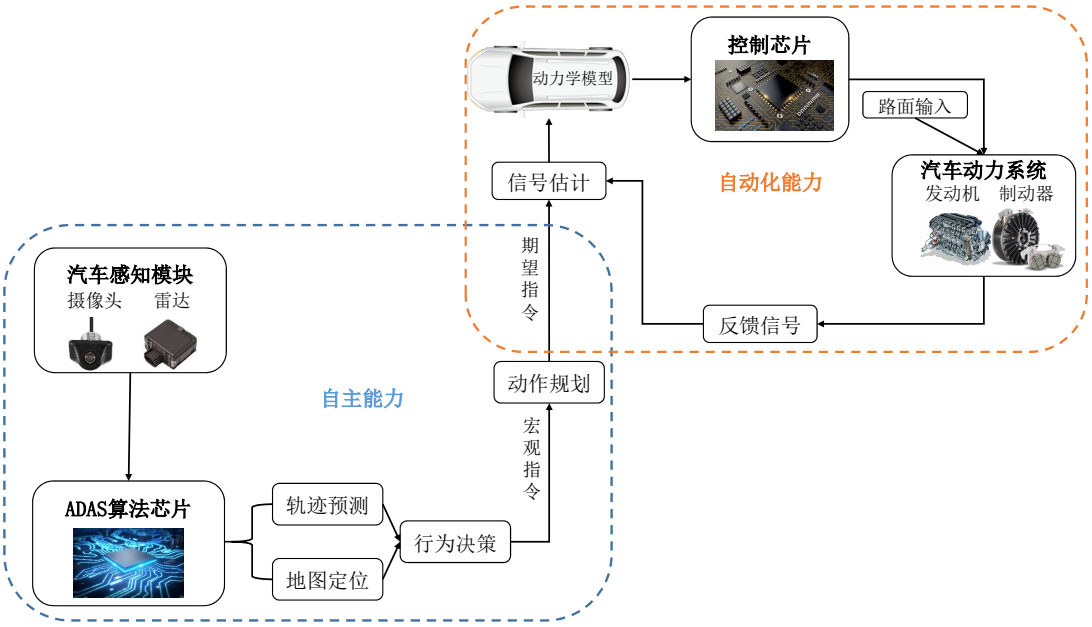


图 1.1　高级辅助驾驶系统决策和控制框图

虽然与传统人机系统相比，人机混合智能系统中的机器不仅具备自动化能力还具有一定的自主能力，但是人工智能算法的引入给人机系统本身带来了额外的问题：

- 不确定性：由于人工智能算法缺乏可解释性，算法输出的结果具有不确定性，并且这种不确定性难以量化，虽然目前有相关文章利用贝叶斯对深度学习网络进行不确定性的量化和分析，但是现有的研究还难以达到真正的解决深度学习不确定性问题的程度。
- 鲁棒性和泛化性差：深度学习算法本身缺乏足够的鲁棒性，在分类问题中，大量的研究发现在原始样本中加入轻微的干扰样本后，算法会做出错误的

分类；在决策相关的研究中，最典型的就是由于现实环境和模拟环境的差距，导致强化学习算法难以迁移到真实物理环境中。

随着人工智能算法的出现，原本的人机系统交互过程中存在的不确定性和风险这两个复杂性来源并没有减弱和消失，反而由于算法的不可解释性、不确定性和鲁棒性差，系统的不确定性和风险很可能会进一步增加。

从人机混合智能系统的整体考虑，由于人类的能力一般是不可调整的，系统决策的性能主要受到以下两方面的影响：一方面，训练阶段机器算法本身的自主能力，比如机器算法的泛化性和鲁棒性能够影响最终系统执行的性能，当机器本身的自主能力够强时，人类和机器争夺控制权的次数也会得到减少，提高了系统决策的平稳性；另一方面，执行阶段人类和机器决策的混合方式，机器和人类的控制权直接决定着最终系统的决策性能，比如当机器遇到难以处理的情况时，人类能够及时接过控制权，保证系统的正常运行，或者是当辅助驾驶过程中人类打瞌睡时，机器能够及时代替人类进行车道保持，以保证驾驶的安全。

1.1.1　将人类请入执行端

虽然人工智能算法给系统本身带来了额外的不确定性和风险，但人类具有较强的推理能力、适应性和鲁棒性，这一点能和机器形成互补。目前，大多数人机系统都是通过将人请入执行端，通过人类独有的智能弥补机器本身存在的不足，典型的场景比如：

- 任务目标过于复杂：机器难以理解太过复杂、因时而变或者是不可编程的任务目标。例如，霍金的智能轮椅中装载了文字输入预测算法，这种算法除了以霍金先前的作品作为训练数据外，还需要根据霍金本人的变化增加数据进行训练调整，此时霍金的目标是不断变化且无法预测的。在上述这类情况中，系统必须将人类请入决策环中。
- 人类绝对控制：尽管很多设备和仪器可以自主运行，但是在执行过程中需要保持人类的绝对控制权，最典型的就是武器系统，智能导弹可以追踪，但是在发射出去后需要保证人类改变其攻击目标的能力。
- 增强系统智能：机器和人类都有自己独特的优势。机器有高精度的运算能力和强大的记忆能力，物理上还有足够的硬度；人类有独特的认知能力，物理上还有足够的灵活度。例如，当机器认知能力不够或者受到干扰时，人类智能的鲁棒性更强，对于系统整体性能有更为稳定的提升；外科医生和达芬奇手术机器人可以协作完成单独机器或者人都无法完成的外科手术。

在这类系统中，机器能够自主感知环境，部分系统中机器还可以根据人的指令或者历史动作确认目标，随后做出相应决策。与此同时，人类进行决策，最终将两方的决策混合后得到最终的决策。人类并不要求每时每刻都参与决策，而如

何更好地混合两方的决策，正是这类系统的关键所在。如今这类人机混合智能系统在多个领域得到了应用，比如自动驾驶[8]、患者援助项目[9]、远程遥控[10]等。人机混合智能系统的发展要求人类和机器共同进化，人类和机器相互学习并且不断改进。随着人工智能技术和认知神经科学的不断发展和进步，人机系统的性能也会不断提升，并且在更广泛的领域发挥重要作用。

1.1.2　将人类请入训练端

人机混合智能系统的决策也受到训练阶段算法性能的影响。深度学习这类数据驱动的方法在信息丰富的场景表现优越，算法可以直接从广泛的样例中学习潜在的模式，通过不断重复的观察和比较来修改自己的决策进程。其中，深度强化学习算法，已经被证明在易于交互且具有精心设计的奖励函数的场景中，性能表现非常出色。深度强化学习技术的发展，使得人机系统中的机器拥有更强的自主能力，进一步推进了系统的发展[11]。

然而，深度强化学习方法在现实应用中还存在许多尚未解决的问题：一方面，明确的奖励函数是不存在的，并且设计高精度的仿真环境是很困难的；另一方面，硬件上的交互十分昂贵，并且容易发生比较严重的事故；更进一步，强化学习要求机器从与环境的交互中不断获得反馈，根据反馈来调整自身的策略，最终找到符合目标奖励的行为策略，而这种试错的过程对数据量的要求非常高，但是实际采样效率却很低，这也是深度强化学习算法的通病。考虑到机器需要从头学起，但是人类的领域知识可以促进机器学习的进步，因此将人类请进训练环(Human-in-the-loop，HITL) 的方法应运而生，目标是整合人类的知识和经验，以最小的成本训练出最精确的模型。

从数据的角度出发，人在环上机器学习方法按照递进关系可分为以下三类[12]：(1) 通过数据处理的方式提高模型性能；(2) 通过介入模型训练提高模型性能；(3) 独立设计人在环的系统。考虑本文的研究重点在于决策部分，因此更关注人在环上的深度强化学习算法。在这类算法中，机器可以从与人类的交互中不断学习，包括人类的演示、干预和评估奖励。这种利用机器与人类的交互来提供任务知识的方法，可以引导和约束机器做出更为安全的探索，减少了机器与环境高风险交互的次数。

对于人机混合智能系统，通过将人引进决策端，让机器做出鲁棒性更高且效果更好的决策，可以减少人的介入次数和人类与机器发生决策冲突的可能性。将人请入训练环的方法，在训练机器时引入人类的先验知识，一方面，可以提高训练时算法的采样效率，降低探索中出现危险行为的可能性。另一方面，人类引导能够使机器的决策更符合人类的思维形式，更有利于执行过程中的人机合作，减少人机冲突，提高系统的安全性。

1.2　研究现状

1.2.1　人机混合智能系统

人机混合智能系统的决策层是系统的关键部分，直接关系任务的执行情况，系统结构如图1.2所示。将人请入决策端后，如何混合人类和机器的决策的关键之一就是在人类和机器的控制权之间找到平衡。大量研究集中在人机交互、机器学习和控制理论中融合人机“决策”[13-14]。根据控制权的不同，目前不同控制方法下的决策方法有以下几种：

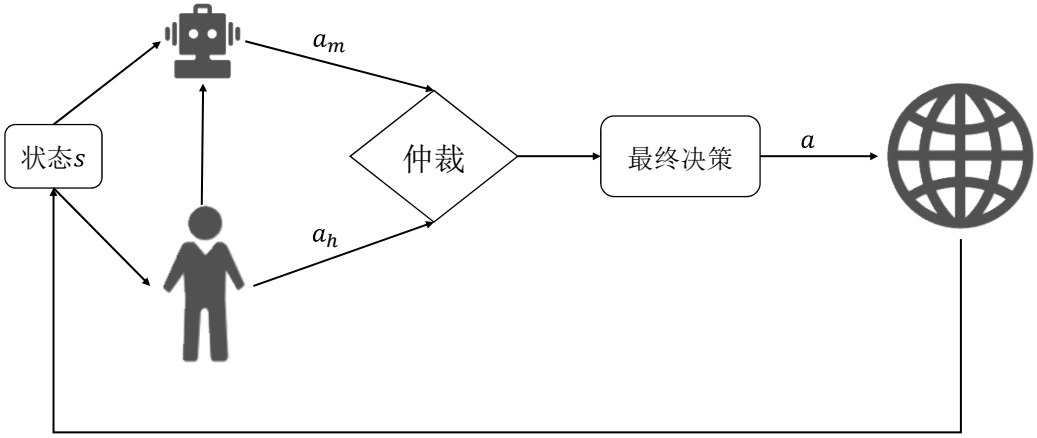


图 1.2　人机混合智能系统框架

1. 介入控制

介入控制，即当系统正常运行时，人类和机器中某一方或者双方同时发挥作用，但是人类或机器会在某个时刻拥有全部的控制权。比如当系统在触发某个事件时，会将全部权力转移给其中一方。在这种情况下，人类和机器有主次关系，根据介入方的角色，介入控制分为人介入机器和机器介入人两种。

人介入机器时，机器在此前是在自主运行，但是因为某种条件的触发，机器被剥夺了自主性。这种情况往往是为了避免机器做出的危险行为引发严重后果，多出现在需要保持人类绝对控制的场景，最典型的便是军事中的武器系统，人类需要保证对武器打击目标的绝对控制。更常见的一类模式是，当辅助机器人的用户命令等触发器被激活时[15]，当机械臂进入障碍物周围的关键区域时，或当目标预测值超过某个置信阈值[16]时，启动人的完全自主接管。由于机器缺乏鲁棒性，这种设置是为了出现某些事先未考虑且机器无法处理的情况时，利用人的能力来解决。

机器介入人时，机器和人此前在系统中共同发挥作用，此时因为特定条件触发，机器剥夺了人的决策权，避免人类做出危害系统性能和安全的行为。对于看重安全性的系统而言，典型的人类出错场景就是人类分心时，如驾驶汽车时人类打瞌睡，这时候机器的及时介入可以避免由于人类疲劳驾驶造成的交通事故。

Mcmullen 等人[17]提出基于模型预测控制的介入控制算法，通过在输入空间的均匀密集采样来评估人可能希望采取的潜在动作，而无需实现了解特定的目标。该方法使用基于模型的强化学习和模型预测控制的思想来预测采样出的动作在未来 N 步的系统轨迹，若人输入动作的未来轨迹没有违反安全约束则执行该动作；否则从安全轨迹的采样动作中选择与人的输入最接近的动作执行，允许用户在不依附单一目标的情况下更安全地在环境中移动。由于人的智能边界难以定量刻画，介入的触发条件需要随着系统目标和操控人员的变化而变化。

上述研究表明，介入控制对于一些简单的场景和任务具有很高的适用性，例如带延迟的遥操作[18]，但需要注意的是触发条件很难普遍适用于所有任务。因为无论是机器介入人还是人介入机器，都需要刻画相应的自主性边界，但是显然无论是对人还是机器而言，智能的边界都是难以量化的，需要设计者根据具体的场景和任务设计相应的度量指标，这正是这类系统的关键所在。

2. 共享控制

共享控制，又称共享自治，其定义较多，目前学术界尚未完全统一。Marion 等人[19]定义的共享控制为：将来自人类操作员的输入与自治系统的计算集成在一起，以产生系统的最终行为。共享控制要求人类和机器共同控制一个系统，同时行动以实现一个共同的目标。对于共享控制的任务，Abbink 等人[20]提出在理想情况下，机器或者人都可以单独的完成。Wang 等人[21]将共享控制分为直接共享控制和间接共享控制。间接共享控制中，操作人员只能通过控制器间接控制机器，系统接受高级输入并且自动转换为较为低级的命令，并最终生成机器的动作。直接共享控制中，将人类和机器视为独立的两个单元，并通过仲裁机制确定最终的控制命令。本文研究对象为人工智能赋能的人机系统，关心的机器是更为高级的，在决策层的权限等级应和人类持平，因此本文考虑的是直接共享控制。

仲裁是共享控制的关键机制，它关系着人类和机器之间控制权的划分。人机系统的相关研究中，仲裁规则的设计十分常见，其中最直接的方法是在多个相加和为一的权重被确定后，分别乘上对应动作值后线性相加求和得到最终的结果。大多数工作根据先验知识预先定义一种函数进行计算求得对应权重，也有研究者采用 hinge 损失函数[22]、概率[23]、概率分布熵[24]和预测不确定性[25]等性能指标来调整控制权限，用于确定远程操作过程中的抓取目标。指数函数族是获得连续和平滑权重的另一种方法，Muelling 等人[26]在计算权重时使用 Sigmoid 函数来考虑用户的最小控制贡献，以实现控制权限的平滑、无缝分配。但指数函数的导数值过大可能导致权限切换过于陡峭[24]。此外，Trautman 等人[27]提出了线性混合的扩展，根据用户统计适当地调节自治，其中共享控制被表述为随机过程，并描述了随机算子、自治和人群函数上的联合分布。局部加权回归[28]，高斯混合回归[29]，任务参数化半隐马尔可夫模型[30]和其他机器学习方法用于离线

编码和学习人类行为的轨迹分布，以生成自治系统的行为。Jarrass 等人[31]使用博弈论来分析人机系统中人机的混合决策问题，交互行为可以通过个体目标 (代价函数) 的不同组合和不同的优化标准来描述。利用纳什均衡解、斯塔克尔伯格均衡解或帕累托均衡解可以推导出人类和机器的策略类型。在纳什均衡中，每个机器都考虑自己的代价函数，使得各自的策略都是对其他策略的最佳响应。其中，纳什均衡可以通过 Li 等人[32-33]开发的最优控制和基于策略迭代的自适应最优控制以及评论员神经网络来实现，以使机器和人类能够同时对机器人施加控制。

深度强化学习的发展推动了共享自治系统的发展，Pellegrinelli 等人[34]使用部分可观测马尔可夫过程 (Partially Observable Markov Decision Process，POMDP) 对共享自治系统进行建模，并使用“后见之明”方法优化估计每个时间步机器人的最佳动作。Reddy 等人[11]使用人在环上强化学习训练端到端的神经网络，使其学会根据环境的观测和用户的输入来输出动作。

上述人机系统，另一个关键组件是机器对人目标的推理，大多数情况下，系统的目标会根据人的目标变化而变化，机器需要理解人类的意图才能更好的和人类一起控制系统。对于可进行逻辑编程的目标，系统便不再依赖人类，机器可以进行完全的自主操作。一个典型的例子便是，机械臂需要在一个物体集合中选择此时人类的目标进行抓取，故其需要根据人类的历史动作来计算所有物体对应为人类目标的概率，并在执行过程中，根据人类参与的控制行为实时计算概率分布，其中最大概率物体的为系统目标[35]。共享控制中人类和机器的控制权是平等的，二者的目标任务是相同的。

3. 分配控制

分配控制模式下整个任务分为两个独立的子任务，人类负责方向[36]、速度或操纵器的几个自由度[37]等特定的输入子集，而机器提供其余的输入或辅助[38]。例如，外部手持摄像头机器人方法的次优视点中，人类操作员可以访问由两个机械手组成的系统，其中一个配备抓手，另一个配备摄像头，以避免机械手本身的遮挡。自主算法负责调节抓手自由度的一个子集，以促使接近感兴趣的对象。此外，人类操作员能够通过对反馈装置施加力的方式，来引导夹持器沿着剩余的无效空间方向前进。此外，机器会修改或禁止手动命令的子空间 (包括速度、方向或运动轨迹)，以满足对状态的任意约束。目前学术界已经推出多种虚拟约束方法，比如势场[39]、虚拟夹具[40]和共享动态曲线[41]。例如，Rahal 等人[42]通过限制横向运动、原位旋转和急转弯的方式来帮助操作员完成切割任务，其中约束的设置与目标任务相关。Daniel 等人[41]构建了一个双手动作库，通过分析人类的双手操作，然后根据动作词汇表中推断出当前动作并采取相应的动作辅助，旨在有效提高特定双手动作的执行力。

1.2.2　人在环上强化学习

深度强化学习的发展使得机器有了足够的自主智能，能够独立完成多种复杂任务的决策。人在环上强化学习，即将人请入训练环，利用和人类交互的方法来提供任务知识和塑造强化学习训练中的奖励。如图1.3所示，根据人在训练环中作用的不同，人在环上强化学习方法主要有以下三种类型：

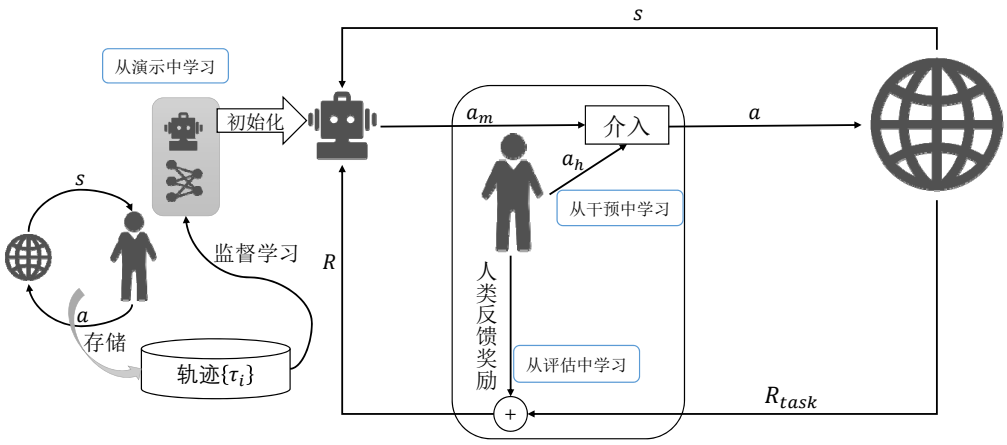


图 1.3　人在环上强化学习整体系统框架

1. 从演示中学习

从演示中学习利用人类执行任务的示例进行学习，直接让机器进行模仿，从而学会预期的行为。这种方法的优点就是机器的行为策略可以快速收敛，但该方法通常是离线执行的，因此当学习后的行为导致了不希望出现或者灾难性的结果时，它不能提供纠正或预防性输入的机制。

基于人类演示使用模仿学习进行训练的方法已经取得了丰富的成果[43-49]。早期关于从人类演示中学习的研究主要集中在教授更高层次的命令，比如控制机械臂的“挑选”、“移动”和“放置”[43-45]，后来这些研究转向了轨迹层的规划[46]。在自动驾驶领域，早期由 Pomerleau 等人[47]设计的用于自主陆地车辆的神经网络具有单个隐藏层，使用从演示中学习的方法训练后，实现了从图像到离散动作的映射。一些研究者使用大约 100 小时的人类驾驶数据用于机器从演示中学习[48]，成功地训练出一种从前置摄像头图像映射到方向盘指令的策略。类似的方法已经被用于小型无人机系统在混乱环境的导航，同时避开障碍物。Nair 等人[49]结合了自监督学习和行为克隆，采用图像作为输入，创建了一个机器人操纵绳子任务的像素级逆动力学模型。与导航相关的工作是利用人体演示的逆最优控制，在现实机器人中学习导航技能，并利用高斯过程检测故障状态和学习恢复策略。

Rahmatizadeh 等人[50]证明人类可以通过演示目标任务来帮助机器人学习。

长短期记忆网络 (Long Short-Term Memory，LSTM) 网络[51]可以学习模拟环境中的人类演示，并将学习到的策略传递给机器。当人类数据有限时，可使用高斯过程来提取更多关于人与机器之间每次交互的信息[52]，以减少机器学会完成任务所需的时间。显然，这些方法目前只能将机器人的行为限制在人类专家所演示的范围内。

另一种方法是从演示中推断执行任务所用的代价函数，当它推断奖励函数时，就被称作逆强化学习。相关研究[53-54]利用最大熵原理，在实际数据的特征约束下使模型分布的熵最大化，并利用学习奖励模型获得的最优策略的值函数与奖励的差值函数或期望值差[54]进行评估。Finn 等人[55]利用最大熵原理学习成本函数，同时为其优化策略，将该策略与已演示的轨迹进行比较，执行和评估每一步，并重复该过程，直到收敛或达到所需的性能。逆强化学习还可以选择将演示直接集成到强化学习算法中，以提高训练期间的样本效率。

2. 从干预中学习

从干预中学习，要求人类在机器执行任务过程中承担监督者的职责，并在必要时进行干预[56-57]，通过干预人类可以进一步调整和改进机器的策略，提高系统性能的同时避免灾难性后果的发生。相比于从演示中学习的方法，这种方法还可以减少与机器的直接交互次数，减轻人的负担。但是，这种方法需要减缓机器在训练过程中的执行速度，以便于操作人员可以在任何情况下进行干预，这就导致了算法的收敛速会变慢，行为也会不稳定。同时，由于需要大量的人类干预，算法无法很好地扩展到更复杂的任务。

大量研究使用的干预措施都建立在经过人类演示训练策略的基础上，Hilleli 和 El-Yaniv 等人[58]提出了使用人类交互来训练分类器，当其检测到不安全状态时，使用基于人类演示的方法训练得到的安全策略来干预系统决策。Saunders 等人[59]通过暂停任务，并训练模型模仿人类干预决策，训练好的干预模型能够取代人工，之后可以继续训练。研究结果表明，这种方法能够在简单的情况下工作得很好，但无法扩展到复杂的情况。此外，Grollman 等人[60]通过演示和混合主动控制相结合的方法训练机器人警察，以及 Hilleli 等人[58]使用将模仿学习与塑造交互式奖励相结合的方法训练赛车。上述方法将人类行为与机器人自主结合起来，以实现共同的目标。

基于干预的训练算法中与前述共享控制有重合之处。Javdani 等人[61]在机器人并不知道目标先验知识的前提下，利用逆最优控制和最大熵原理，根据历史输入估计人类目标的概率分布。不同于首先预测目标然后进行辅助的机器人传统方法，这种方法可以使用更少的数据样本更快地完成任务。意图推理是共享控制中重要的一部分，共享控制将人请入了决策端，但是人同时也参与训练端，目的是帮助机器学会推理出人类的意图，以便于更好的实现人机的合作。其他方

法[11]将强化学习预先训练好的策略与人类输入相结合，实验结果表明这种组合比单独人类或者机器决策的效果好。最主要的限制是它需要已经训练过的 Q 网络函数，这对于现实世界的任务而言可能不现实。

3. 从评估中学习

不同于需要人类操作者长时间的参与训练任务的方法，从评估中学习，利用的是人类领域知识，通过人类评估反馈的形式生成奖励来塑造机器的行为，奖励函数允许是近似的[62-63]。这种稀疏的交互减轻了人类操作者的负担，当训练出现机器无法执行的行为时，人类只需要理解任务目标并通过奖励进行引导即可。一个典型的例子，在受限空间中操纵拥有多个自由度的机械臂，此时由于障碍物和关节数量太多，人类无法提供完整的演示，但是可以很轻松的评价任务是否成功完成。当系统的时间域短于人类的反应时间时，自主系统就需要考虑信誉分配问题，即如何根据人类的一次反馈来评估多个不同行为的评估值。

由于机器只能通过不断地探索或者根据人类反馈的间接引导使得策略逐渐收敛，而不是更为明确的训练目标，这会导致算法的收敛速度变慢。同时，人类的评估信号通常是非平稳的，并且依赖于人类的策略，比如过去某些时刻好的行为在现在可能被视为不好的行为，如何让机器能学会区分两种情况和人类的评估设计密切相关。Knox 等人[62,64-65]基于“通过评估手动强化训练机器”的框架开发了将人类输入引入经典机器学习算法的方法。人类训练员根据机器过去的动作对其进行奖励，框架是沿着状态-动作对进行奖励的分配，以帮助机器形成最终的策略。这项工作后来扩展到使用深度神经网络来解决原始图像作为输入的 Atari 游戏[66]。León 等人[67]在任务执行过程中混合了人类演示和人类直接自然语言反馈，作为额外的动态奖励形成机制。但是到目前为止，这些方法只应用于低维强化学习问题。

不同于使用人类作为奖励函数生成器，MacGlashan 等人[63]提出了使用人类作为时序差分误差的演员-评论家算法 (Convergent Actor-Critic by Humans，COACH)。COACH 基于如下的假设：优势函数是人类反馈的一个很好的模型，以及评论网络的时序差分误差是对优势函数的无偏估计。Ibarz 等人[68]结合从专家演示中学习和从评估中学习，在不使用游戏分数的情况下训练机器通关 Atari 游戏。该方法首先训练了一种行为克隆策略，然后使用人类反馈和行为轨迹偏好学习奖励，实验表明这种结合的方法优于仅使用评估或仅使用演示。

1.3　本文工作和结构安排

本节针对人机混合智能系统的序贯决策方法的研究现状进行分析，总结现有研究存在的问题，然后阐述本文研究内容，最后介绍本文组织结构。

1.3.1　问题总结

本文主要研究人机智能混合系统的序贯决策问题，总结现有研究的问题如下：

- **人类辅助训练：**如前一节所述，现有研究成功使用了人工智能算法训练人机系统中的多个重要组件。例如，使用循环神经网络训练意图推理模块，使用深度强化学习训练机器策略，以及辅助驾驶中机器介入触发机制 (使用计算机视觉技术识别人类疲劳状态) 等。但是，人工智能算法具有不确定性、鲁棒性和泛化性差等特点，当环境中存在轻微扰动时机器算法将不再可靠。针对这些问题，大多数现有研究希望通过设计合理的人机混合决策方法，利用人类先验知识弥补机器本身自主能力的不足。然而，如果完全依赖执行阶段人机决策混合方法，当任务场景与训练环境存在一定差距时，由于算法的泛化性和鲁棒性差，机器决策将完全失效。此时，系统决策将完全依赖人类，极大地增加了人类的负担，人机系统失去了原本的价值。人工智能算法这种数据驱动的方法，其决策性能极大程度的依赖训练方法，如何利用人类先验知识辅助机器训练，提升其决策的泛化性和鲁棒性，是本文亟需解决的一个关键问题。
- **人机决策混合：**(1) 针对共享控制问题，仲裁机制能够根据人类和机器决策质量实时调整相应的决策权重。当人类决策性能较差时，控制权应当偏向机器，而当机器决策性能较差时，控制权应当更偏向人类，这意味着系统的仲裁机制能够同时评估人类和机器的决策。如何设计有效的仲裁机制，将直接影响共享控制系统的决策性能，是本文亟需解决的一个关键问题。(2) 针对介入控制问题，如何设定合理的触发机制，使得当出现机器或者人类某一方无法应对的场景时，另一方能够及时接手。介入控制主要分为人介入机器、机器介入人和人与机器相互介入三种情况。本文主要研究人介入机器的场景，此时人类默认绝对正确，如何设定合理的触发机制让人类能够及时介入的同时减轻人的操作负担，是本文亟需解决的一个关键问题。

1.3.2　本文研究内容

针对上述问题，本文同时从算法的训练端和执行端出发，通过引入人类智能的方式提高系统决策的鲁棒性和安全性，最终提高人机混合智能系统的决策性能。具体的研究内容主要包含：

- **提出人类策略限制下的共享控制算法：**训练阶段，针对算法训练效率低，泛化性和鲁棒性差的问题，本文使用了基于人类策略限制的深度强化学习算法训练机器，通过策略约束引导机器高效探索找到最优解，提高了算法的训练效率、安全性和泛化性。此外，由于策略限制的存在，机器决策具有

较强的鲁棒性。执行阶段，针对人类和机器决策失效的情况，设计了包含策略评估和策略限制的仲裁机制，仲裁机制能够滤除人类或者机器无效的动作；针对人类和机器控制权分配的问题，仲裁机制能够根据意图推理结果的置信程度动态调整可选动作空间，改善了系统的决策性能。

- **提出面向多机竞速的人机介入控制算法**：训练阶段，针对多机竞速场景的不确定性，本文设计包含人类反馈奖励的函数组，并通过深度强化学习算法训练无人机掌握自主决策能力。针对竞速规则复杂且难以编码的特性，这种基于人类评估的软性训练方法，能够有效引导机器做出符合规则的行动。执行阶段，针对竞速问题的强动态性，本文设计了人类的两级介入机制，给予了足够的人类响应时间。同时，将动作空间细分的机制减少了人类的介入负担。
- **搭建 Sim2Real 人机实验平台**：由于真实场景和模拟环境存在一定的差距，泛化性和安全性较差的算法难以在真实物理环境中做出有效决策。因此，仅仅依靠模拟环境的仿真实验，还不足以验证系统决策的泛化性。为了充分证明所提方法的有效性，本文以旋翼无人机为背景，搭建了一个具有一定通用性的 Sim2Real 人机仿真实验平台，并通过一系列的实验验证了所提方法在真实物理世界的性能。

1.3.3　论文组织结构

全文共六章，章节的组织结构如图1.4所示，具体描述如下所述：

第 1 章介绍了本文的研究背景和意义，以及人机混合智能系统和人在环上强化学习的相关研究现状，最后给出全文的组织结构和章节内容。

第 2 章阐述了全文研究所涉及到的相关基础知识和概念，以及后续研究所涉及到的基础算法。具体包括序贯决策与马尔可夫过程，强化学习与深度强化学习以及人在环上强化学习。

第 3 章提出了基于人类策略限制的人机共享控制决策方法。研究了在共享控制问题中，如何将人类的先验知识引入算法的训练端和执行端，以提升系统决策的性能，同时使得机器决策更符合人类的意图。具体包含训练端基于人类策略限制的强化学习算法的设计、共享控制结构和仲裁机制的设计、算法的仿真验证。该研究为后续在训练和执行端引入人类先验知识提供了基础性结构方案。

第 4 章面向多机竞速场景提出了基于人类反馈奖励的人机介入控制决策方法。以多无人机竞速问题为研究场景，研究了在介入控制问题中，如何将人类的先验知识引入算法的训练端和执行端，以提升系统决策的性能。具体包含人类反馈奖励塑造的方法、人介入机器的机制设计、相关的仿真实验。该研究为后续在真实物理平台进行仿真验证奠定基础。

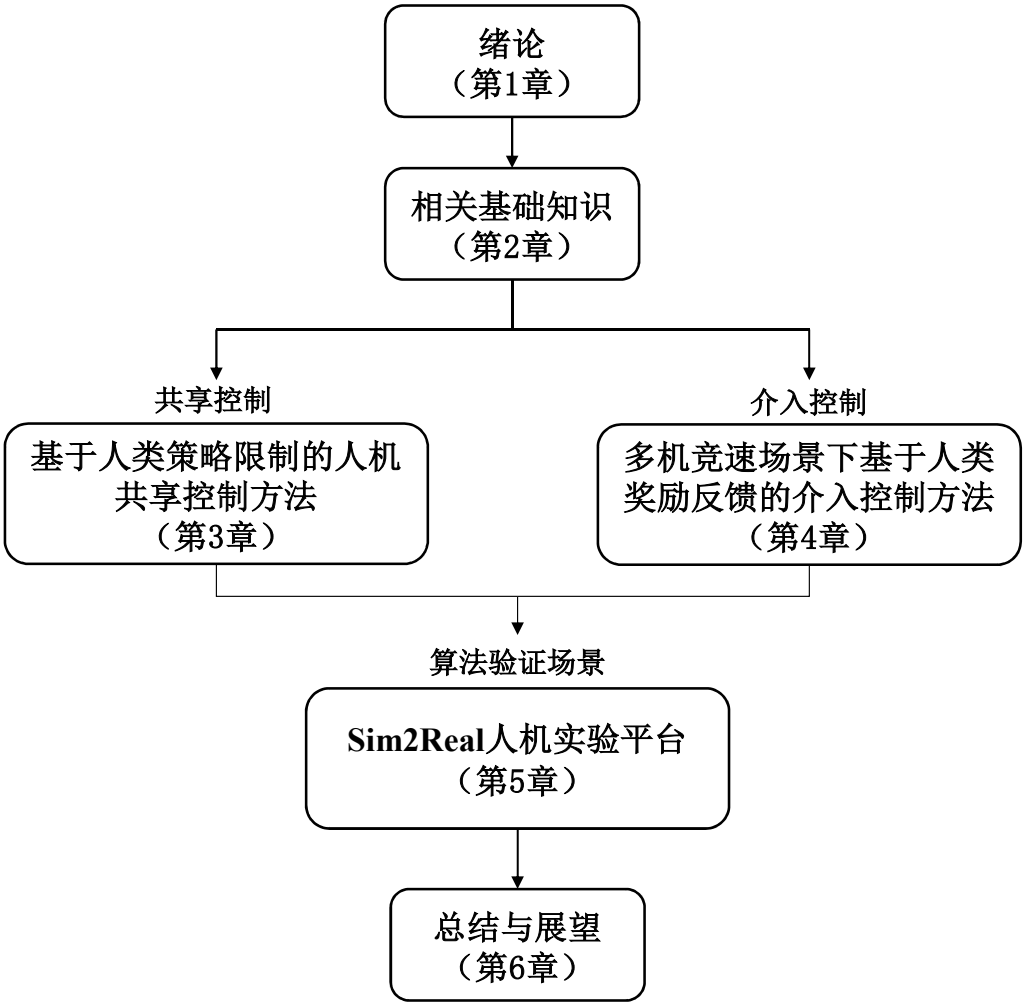


图 1.4　本文组织结构图

第 5 章以旋翼无人机为对象，搭建了从仿真到现实的人机实验平台，提出了算法部署到真实物理场景的整体流程和框架，并针对提出的多机竞速场景下强化学习算法驱动的人机介入控制方法，进行了现实场景下的算法验证。具体包括用于提高算法可迁移性的技巧设计、实验平台的整体结构设计、基于机器人操作系统的软件算法以及最终户外实验结果。

第 6 章总结全文的工作，阐述本文研究中存在的不足以及对未来研究的进一步展望。

第 2 章　相关基础知识

本章主要介绍本文研究将会涉及到的相关基本概念和基础知识，后续所有的研究将基于此进行展开，由于本文所提出的算法大都是基于相关领域典型算法所进行的改进，因此本章将进行基础性质的介绍，具体包括序贯决策与马尔可夫决策过程、强化学习、深度强化学习以及人在环上强化学习。

2.1　序贯决策与马尔可夫决策过程

2.1.1　序贯决策

序贯决策问题广泛存在于人机混合智能系统中，是本文主要的研究对象。序贯决策任务具有多阶段和时序的特点，需要机器在每个离散时间步与环境进行交互，即从环境中获取状态后执行相应决策并获得对应奖励，详细的交互过程如图2.1所示。在 t 时刻，机器从环境中获取当前状态 s_t，然后根据自身策略执行相应的动作 a_t，并从环境中获取反馈奖励 r_t，与此同时环境进行状态转移。值得注意的是，机器的策略可以是随时间变化的。根据系统状态转移的类型，序贯决策分为确定性序贯决策和随机序贯决策。当环境的状态转移是确定的，即在确定 t 时刻的状态和动作后下一时刻的系统状态是确定的，此时为确定性序贯决策；当状态转移函数为一个概率分布时，此时为随机序贯决策[69]。

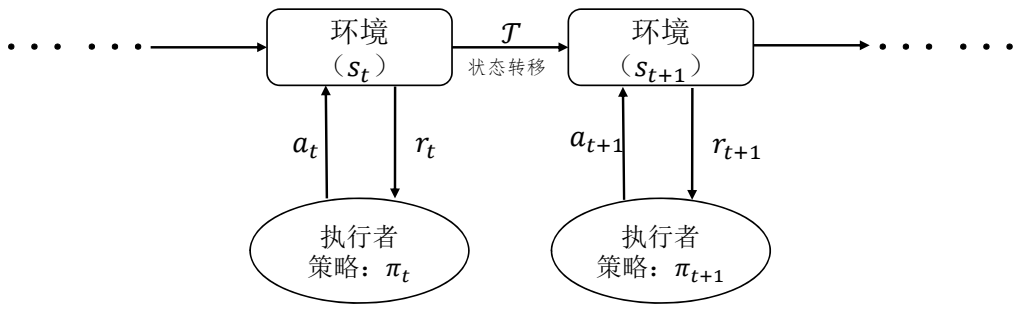


图 2.1　序贯决策状态转移示意图

如图2.1所示，序贯决策任务的奖励具有一定的延迟性，机器在完成最终的任务前，每次获取的奖励仅仅是单步的。序贯决策问题的求解目标为未来的累计收益，因此随着时间步的不断增加，显然决策求解空间也会指数增长。

对于序贯决策问题，当状态转移函数已知时，由于其满足最优子结构和重叠子问题这两个性质，可以通过动态规划方法进行求解。但是对于随机序贯决策问题，由于状态转移函数为概率分布，极大地增加了计算复杂度。动态规划要求准

确的状态转移函数，这对于大多数现实任务而言，是种苛刻的要求。考虑到这类任务的环境具有马尔可夫性质，因此本文将机器与环境交互过程建模成马尔可夫决策过程。

2.1.2　马尔可夫决策过程

马尔可夫性质指一个随机过程的下一时刻状态的条件概率分布，仅取决于当前时刻的状态，并不依赖过去所有的状态。马尔可夫过程是指一个随机过程中的状态转移具有马尔可夫性质。马尔科夫决策过程 (Markov Decision Process, MDP) 需要在马尔可夫过程中额外增加动作变量，即下一时刻的状态仅取决于当前时刻的状态以及动作。

一个马尔可夫决策过程可以定义为一个四元组 (S, A, T, R)，其中每个元素的含义如下：

- S 为系统的状态集合，其中包含起始状态的分布；
- A 为所有可能动作的集合；
- T 为状态转移分布，$T(s_{t+1}|s_t, a_t)$ 表示将 t 时刻的动作-状态映射到 $t+1$ 时刻的状态分布。值得注意的是，状态转移符合马尔可夫性质，即:

$$T(s_{t+1}|s_t, a_t, s_{t-1}, a_{t-1}, ..., s_0, a_0) = T(s_{t+1}|s_t, a_t) \tag{2.1}$$

- R 为奖励函数，$r_t(s_t, a_t)$ 表示在 t 时刻，机器在状态 s_t 下执行动作 a_t 所获得的即时奖励。

马尔可夫决策过程是一种用于建模顺序决策问题的工具，其中决策者以顺序方式与系统交互。有限马尔可夫决策过程指的是其状态和动作集都是有限的[70]，后续的多种理论都是基于有限马尔可夫决策过程建立的，交互过程的整体流程如图2.2所示。

上述的 MDP 模型是假设环境完全可观察的，即机器总能观察到自身的状态。但是机器不一定知道自身所处的状态，所以无法在按照策略 $\pi(s)$ 执行动作，此时的最优行动是取决于机器对于当前状态 s 的观测值。此时可以将问题建模成部分可观察马尔可夫过程 (Partially Observable Markov Decision Process, POMDP)。现实场景的状态往往是部分可观察的，所以 POMDP 问题的研究意义重大[71]。

一个 POMDP 由一个七元组 $(S, A, T, R, \omega, O, b)$ 所定义[72]，其中：

- S, A, R, T 和 MDP 所定义的相同；
- ω 是所有可能观察的集合；
- O 是观测函数，$O(a, s', o) = Pr(o_{t+1}|a_t, s_{t+1})$ 是当机器执行动作 a 后并且到达状态 s' 后观测到 o 的概率。
- 信念 $b = Pr(s|h)$，其中 $h_t =< a_0, o_1, a_1, o_2, \ldots, a_{t-1}, o_t >$ 表示从时间 0 开始

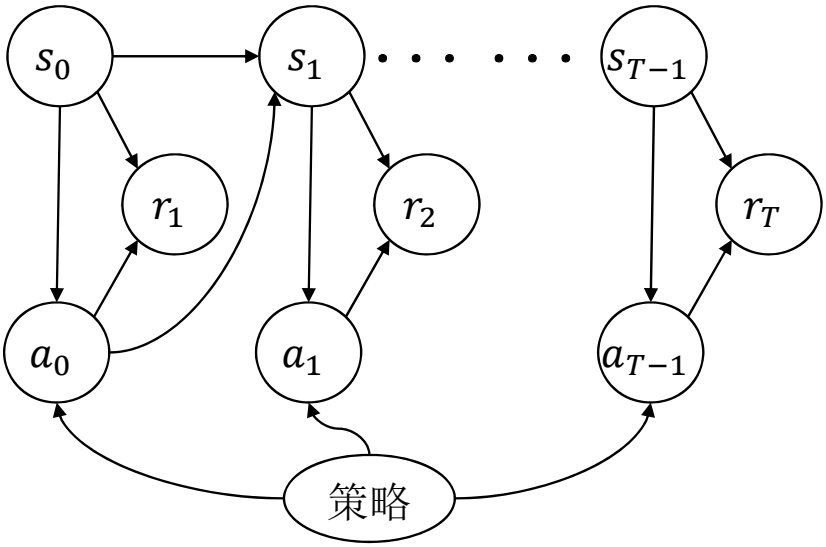


图 2.2　有限马尔可夫决策过程示意图

的所有机器和环境的历史交互信息组成，而信念 b 表示在观察历史 h 后处于状态 s 的概率。对于所有的 $s \in S$，$b(s) \in [0,1]$，且 $\sum_{s\in S} b(s) = 1$。每当机器收到新的观测并执行新的行动时，这种信念会发生改变。假设机器当前的信念状态为 b 时，执行相应的动作 a 并观察到 o 后，信念更新公式如下：

$$\begin{aligned} b'(s') &= Pr(s'|b,a,o) \\ &= \frac{Pr(s',b,a,o)}{Pr(b,a,o)} \\ &= \frac{Pr(o|s',b,a)\sum_{s\in S} T(s,a,s')b(s)}{Pr(o|b,a)} \end{aligned} \tag{2.2}$$

2.2　强化学习

人机混合智能系统中的决策任务，由于其典型的时序和多阶段特性通常被建模成 MDP 或 POMDP 模型，但是由于人、机器和环境之间的交互比较复杂，且现实场景中精确的环境模型通常是难以获得的。强化学习算法通过与环境交互的方式进行学习，数据驱动的形式使得算法在求解这类问题中有更强的通用性和灵活性。强化学习由机器和环境两个对象组成，其主要决策框架就是 MDP，机器主要有决策和学习两个功能，决策即是根据不同的状态执行相应的动作，学习则是根据与外界环境交互获得的奖励来调整自身的决策。强化学习由一个四元组所定义 (S, A, P, R)，在当前时刻，机器观察到状态 s，并执行动作 a，环境

以概率函数 $p(s'|s,a)$ 转移到下一时刻状态 s'，机器由此获得 r，以此循环直至到达最终状态，由此持续的整个过程成为一个回合 (episode)。

强化学习本质上是利用累计回报来学习，最终学会最优策略来最大化整体的回报，即：

$$G(\tau) = \sum_{t=0}^{T} \gamma^t r_{t+1} \tag{2.3}$$

其中，τ 表示机器的行动轨迹。当环境没有中止状态时，$T = \infty$。折扣系数 $\gamma \in [0,1]$ 用于控制长短程回报的权重比，同时避免因为系统没有终止状态，导致的累计回报不收敛。但是由于状态转移和机器策略都具有一定的随机性，所以算法优化的目标函数不应该是一条轨迹的累计折扣回报，而应该是期望回报，即：

$$\begin{aligned} \mathcal{J}(\theta) &= \mathbb{E}_{\tau\sim p_\theta}[G(\tau)] \\ &= \mathbb{E}_{\tau\sim p_\theta}[\sum_{t=0}^{t=T-1} \gamma^t r_{t+1}] \end{aligned} \tag{2.4}$$

其中，p_θ 表示机器的执行策略函数，θ 为其参数。强化学习引入了两个值函数来评估策略的期望回报，即状态值函数 $V(s)$ 和状态-动作值函数 $Q(s,a)$。

状态值函数 $V(s)$ 由期望回报分解而来，

$$\begin{aligned} \mathbb{E}_{\tau\sim p(\tau)}[G(\tau)] &= \mathbb{E}_{s\sim p_{s_0}}[\mathbb{E}_{\tau\sim p(\tau)}[\sum_{t=0}^{T-1} \gamma^t r_{t+1}|\tau_{s_0} = s]] \\ &= \mathbb{E}_{s\sim p_{s_0}}[V^\pi(s)] \end{aligned} \tag{2.5}$$

又由于 MDP 的马尔可夫性质，上述中，

$$V(s) = \mathbb{E}_{a\sim\pi(a|s)}\mathbb{E}_{s'\sim p(s'|s,a)}[r(s,a,s') + \gamma V^\pi(s')] \tag{2.6}$$

上式为 V 函数的贝尔曼方程，状态-动作值函数 Q 可以由 V 推导而来，

$$Q^\pi(s,a) = \mathbb{E}_{s'\sim\pi(s'|s,a)}[r(s,a,s') + \gamma V^\pi(s')] \tag{2.7}$$

状态值函数 $V^\pi(s)$ 是动作-状态值函数 $Q^\pi(s,a)$ 关于动作 a 的期望，

$$V^\pi(s) = \mathbb{E}_{a\sim\pi(a|s)}[Q^\pi(s',a')] \tag{2.8}$$

同时 Q 函数也能变换成相应的贝尔曼方程形式，

$$Q^\pi(s,a) = \mathbb{E}_{s'\sim p(s'|s,a)}[r(s,a,s') + \gamma\mathbb{E}_{a'\sim\pi(a'|s')}[Q^\pi(s',a')]] \tag{2.9}$$

值函数可以视作对策略的评估，也可以替代轨迹的累计收益期望作为机器的优化目标。强化学习求解的最优策略的值函数对应最优值函数，其中对应的最

优状态值函数记为 $V^*(s)$，以及最优的状态-动作值函数记为 $Q^*(s,a)$, 两者的对应关系为，

$$V^*(s) = \max_a Q^*(s,a) \tag{2.10}$$

同时，两个最优值函数对应的贝尔曼方程为最优贝尔曼方程，如下所示：

$$V^*(s) = \max_a \mathbb{E}_{s' \sim p(s'|s,a)}[r(s,a,s') + \gamma V^*(s')] \tag{2.11}$$

$$Q^*(s,a) = \mathbb{E}_{s' \sim p(s'|s,a)}[r(s,a,s') + \gamma \max_{a'} Q^*(s',a')] \tag{2.12}$$

2.2.1　基本解法

对于上述最优贝尔曼方程，强化学习有以下三种基本解法：

- 动态规划法，要求已知模型的状态转移概率和奖励函数，然后通过贝尔曼方程迭代求解计算最优值函数[73]。对于动态规划方法，主要有策略迭代和值迭代两种算法。策略迭代算法根据贝尔曼方程进行策略评估，并根据当前值函数进行策略改进，依次循环直至策略收敛。值迭代算法直接用贝尔曼方程更新值函数，策略为每次选择状态-动作值函数中最大值对应的动作。当值函数收敛得到最优的值函数时，对应策略便是最优策略。但是，上述算法要求模型已知，并且当状态的数量不断增加时，计算量指数增大，算法效率不高，适用范围较小。
- 蒙特卡洛法，并不需要状态转移函数和奖励函数已知，只需要机器与环境进行交互，收集数据，并根据这些数据求得 MDP 的最优策略解。机器经过 N 此试验后得到的 N 个轨迹和对应总汇报，Q 函数根据总回报近似计算得到，

  $$Q^\pi(s,a) \approx \hat{Q}^\pi(s,a) = \frac{1}{N}\sum_{n=1}^{N} G(\tau^{(n)}_{s_o=s,a_0=a}) \tag{2.13}$$

  在近似求得 Q 函数后进行策略改进，然后让机器在新的策略下进行试验，近似估计得到新的 Q 函数，并以此循环直至 Q 函数收敛。值得注意的是，如果策略 π 固定，经过多次试验后 Q 函数只能收敛到 $Q^\pi(s,\pi(s))$，这个过程称为利用。机器如果缺乏探索不能计算出其他策略下的 Q 函数，因此在该方法中需要平衡利用和探索。这类方法需要等到获得完整的一条轨迹时，机器才能进行学习，算法效率低且数据的方差较大，导致实际算法的效果不够理想。
- 时序差分法，结合动态规划和蒙特卡洛方法做出了一种改进。时序差分方法可以每隔一步或者几步，就利用贝尔曼方程进行价值评估。值函数的估

计值从累加平均变为增量计算的方式，即,

$$\hat{Q}_N^{\pi}(s,a)=\frac{1}{N}\sum_{n=1}^{N}G(\tau^{n}_{s_0=s,a_0=a})=\hat{Q}_{N-1}^{\pi}(s,a)+\frac{1}{N}(G(\tau^{N}_{s_0=s,a_0=a})-\hat{Q}_{N-1}^{\pi}(s,a)) \tag{2.14}$$

时序差分法根据行为策略 (进行探索) 和优化策略 (目标策略) 的同异，分为同策略算法和异策略算法，两者的典型算法代表分别为 SARSA 和 Q-learning，详见算法2.1和算法2.2。

算法 2.1　SARSA 算法流程

初始化参数步长 α 和折扣因子 γ ;
初始化状态空间 S、动作空间 A 和动作价值函数 $Q(s,a)$;
while *episode=1,2,...,M* **do**
    初始化状态 s ;
    根据一个基于动作价值 $Q(s,a)$ 的策略来选取当前状态 s 下的动作 a ;
    while *t=1,2,...,T* **do**
        执行动作 a，获得奖赏值 r 和下一状态 s' ;
        一个基于 $Q(s,a)$ 的策略来选取当前状态 s' 下的动作 a' ;
        $Q(s,a)\leftarrow Q(s,a)+\alpha[r+\gamma Q(s',a')-Q(s,a)]$;
        $s\leftarrow s'$，$a\leftarrow a'$
    end
end

算法 2.2　Q-learning 算法流程

初始化参数步长 α 和折扣因子 γ ;
初始化状态空间 S、动作空间 A 和动作价值函数 $Q(s,a)$;
while *episode=1,2,...,M* **do**
    初始化状态 s_0;
    while *t=1,2,...,T* **do**
        基于动作价值 $Q(s,a)$ 选取当前状态 s 下的动作 a ;
        执行动作 a，获得奖赏值 r 和下一状态 s' ;
        $Q(s,a)\leftarrow Q(s,a)+\alpha[r+\gamma Q(s',a')-Q(s,a)]$;
        $s\leftarrow s'$
    end
end

2.2.2　多臂赌博机

多臂老虎机对应一个学习问题：操作者需要在多个选项中进行选择，每次选择后会获得一定的奖励，奖励由一个平稳分布随机生成，目标是优化一段时间内的总收益。多臂老虎机是一个典型的需要平衡探索和利用的问题，当操作者对一个选项进行无限次的选择后，获得的多次收益会接近其真实的概率分布，但是在有限的次数内，操作者需要在多个选项中权衡，找到长期收益更高的选项。多臂赌博机是强化学习中的一类特殊问题，其一个回合的长度为 1 步。

每个选择 a 的价值函数可以通过计算均值获得，

$$Q(a) = \frac{\text{执行动作 } a \text{ 的总收益}}{\text{执行动作 } a \text{ 的总次数}} \tag{2.15}$$

最简单的方式是每次选择最大价值的动作进行探索，但是这种贪心算法只能看到眼前最大的收益。而 ϵ-贪心算法，可以平衡探索和利用，每次以一个很小的概率从动作空间中平等的选择一个执行，其他时刻依然使用贪心算法。

ϵ-贪心算法虽然可以进行探索，但不会考虑不确定性较大的选择，缺乏对不确定性的估计，而对于非贪心的动作，需要考虑其接近最大值的程度。基于置信度上界 (Upper Confidence Bound, UCB) 的方式显然更为有效，

$$A_t \triangleq argmax_a[Q_t(a) + c\sqrt{\frac{\ln t}{N_t(a)}}] \tag{2.16}$$

其中，$N_t(a)$ 表示到 t 时刻为止选择 a 的总次数，当 a 选择次数增多后不确定性就会变小。c 为大于 0 的一个正常数，决定了置信水平。值得注意的是，当 $N_t(a) = 0$ 时，动作 a 被视为满足最大化条件的动作。

2.3　深度强化学习

早期的强化学习算法，都是以表格的形式表示值函数，所面向场景的状态和动作都是离散的，这就导致算法的实用价值并不高。深度强化学习算法 (Deep Reinforcement Learning，DRL) 结合神经网络和强化学习算法，通过神经网络来表征强化学习的值函数和策略函数，能够将强化学习算法推广到动作和状态都是连续的场景。同时，神经网络反向梯度传播更新参数的方式，计算速度更快。神经网络拥有强大的表征能力，表格型强化学习算法难以解决 POMDP 问题，但是对于给定部分历史完全可观察和给定完整历史完全可观察这两类 POMDP 问题，通过循环神经网络和卷积神经网络进行特征提取的方式，可以获取隐藏的完整的状态信息，使得算法能用于解决这类问题。

根据策略迭代、值函数迭代的方式以及状态转移函数是否已知，深度强化学

习算法分为以下三类：基于值的深度强化学习算法，基于策略的深度强化学习算法，以及基于模型的深度强化学习算法。

2.3.1　基于值的深度强化学习算法

深度 Q 网络 (Deep Q Network, DQN) 是深度强化学习算法的经典开篇之作，通过神经网络和 Q-learning 相结合的方式，直接将图像作为神经网络的状态输入，输出所有动作的价值，贪心策略会从中选出最大值所对应的动作。但是两者结合还会带来一些问题：神经网络一般要求训练数据的分布固定且相互独立，但是强化学习的采样数据往往是稀疏且高度相关的，数据分布也会随着机器策略的变化而变化。

针对上述问题，DQN 使用经验复用池来存储 (s_t, a_t, r_t, s_{t+1}) 形式的采样数据，每次更新所用数据都是从经验池中随机抽取而来，这种方法打乱了数据之间的强关联性，更有利于网络的训练。DQN 中神经网络的优化目标为期望目标 Q 函数值和当前 Q 函数值的差值的平方。期望目标 Q 函数值、损失函数及其半梯度函数分别如下所示：

$$\begin{aligned} Q_{i-1}^{target} &= r + \gamma max_{a_{t+1}} Q(s_{t+1}, a_{t+1}; \theta_{i-1}) \\ L_i(\theta_i) &= \mathbb{E}_{s,a\sim\rho}[(Q_{i-1}^{target} - Q(s_t, a_t; \theta_i))^2] \\ \triangle_{\theta_i} L_i(\theta_i) &= \mathbb{E}_{s,a\sim\rho}[(Q_{i-1}^{target} - Q(s_t, a_t; \theta_i))\triangle_{\theta_i} Q(s_t, a_t; \theta_i)] \end{aligned} \tag{2.17}$$

参数每步更新时，依赖于此时 Q 函数值和目标 (预测)Q 值之间的差距，即当前时刻的参数会影响下一时刻的采样数据。例如，如果最大值对应的动作是向上，那么之后训练会受到更多上方数据的影响，网络参数很可能就会陷入局部最小值或者甚至不收敛。DQN 使用目标网络和在线网络进行训练，其中目标网络用于计算 Q 函数的目标值，会暂时冻结参数并且隔一段时间更新；在线网络则用于计算 Q 函数的估计值且会不断地更新参数。这种方式降低了两种网络之间的关联性，提高了算法的稳定性。DQN 算法综合了“双网络冻结参数更新”和“经验复用池”两种技巧，解决了神经网络和强化学习结合的一些问题，具体流程如算法2.3所示。

目前为止，出现了多种基于 DQN 算法的改进，其中 RAINBOW DQN 是本文后续研究所涉及的算法，其集成了关于 DQN 算法的六个改进，具体改进如下所示：

- DDQN：由于 DQN 算法中每次选择的都是当前认为价值最高的动作，这就导致了对 Q 值的过高估计。机器选择下一时刻的动作和计算下一时刻动作的 Q 值都用了目标网络，DDQN 算法则是将价值评估和动作选择分解开，其中 Q 值由目标网络计算得到，而动作值由在线网络进行选择。

算法 2.3　双网络 DQN 算法流程

超参数：回放缓存容量大小 N，奖励折扣稀疏 γ，Q 函数更新的延迟步长 C, ϵ-greedy 策略中的 ϵ；

初始化：经验回放缓存单元 D; 评估网络 Q，随机生成权重 θ; 目标网络 $\hat{Q}$，其权重 $\theta^{-}=\theta$；

while *episode=1,2,...,M* **do**

　初始化状态 s_0，其状态向量为 $\phi_0=\phi(s_0)$；

　while *t=1,2,...,T* **do**

　　以概率 ϵ 选取随机动作 a_t；

　　否则选取 Q 值最大的动作 $a_t=\max_a Q^*(\phi(s_t),a;\theta)$；

　　执行动作 a_t，获得奖赏值 r_t 和新状态 s_{t+1}，新状态向量 $\phi_{t+1}=\phi(s_{t+1})$；

　　将四元组 $(\phi_t,a_t,r_t,\phi_{t+1})$ 存入经验池 D；

　　从经验池 D 中采集 m 个样本 $(\phi_j,a_j,r_j,\phi_{j+1})$，j=1,2,...,m；

　　计算当前样本的目标 Q 值:

$$y_j=\begin{cases} r_j & \phi_{j+1}\text{为终止状态} \\ r_j+\gamma\max_{a'}\hat{Q}(\phi_{j+1},a';\theta^{-}) & \phi_{j+1}\text{非终止状态} \end{cases}$$

　　对损失函数 $(y_j-Q(\phi_j,a_j;\theta))^2$ 做梯度反向传播以更新评估网络参数 θ；

　　每 C 步更新目标网络 $\hat{Q}$ 参数 $\theta^{-}=\theta$；

　end

end

- 竞争网络：将 Q 值函数网络最后一层分解成两部分，即 Q=A+V，其中 A 是输出的优势函数，表明在当前状态下该动作相对于其他动作的优劣。V 是输出的价值函数，表明状态的优劣程度。通过同时评价状态和动作的方式提升学习的效率。
- 基于优先级的复用：将时序差分误差大的样本视作需要学习的对象，通过改变采样权重的方式调整样本抽取优先级，其中时序差分误差越大的优先级越高，这种方法提高了样本的训练效率。
- 多步学习：DQN 使用单步预测值和即时奖励来估计目标值，一旦训练时网络的输出偏差较大，目标值的偏差也会变得很大。使用多步的真实奖励的方法，可以解决偏差大的问题，当所选步数趋于回合步长时，算法退化成蒙特卡洛方法，带来了更多的方差，因此步长的选择可以用于平衡偏差和

方差。

- 分布式强化学习：DQN 的值函数输出为对应动作-状态对的期望值。但是，在当前状态下两个动作的期望值相同不等于两个动作等价。例如，当前状态下一个动作 50% 概率的价值为 10，50% 概率为-10，期望为 0。另一个动作 90% 概率的价值为 10，10% 概率为-90，期望为 0。只看期望，两个动作的优先级相同，但是从减少风险的角度来说，应该选择前一种动作。RAINBOW 将网络输出期望值改成输出直方图，来表示每个价值点的概率分布，通过优化当前分布与目标分布的差距来更新网络参数。
- 噪声网络：相较于采取 ϵ-贪婪策略来进行探索，RAINBOW 选择给网络参数添加噪声的方法来增强模型探索能力，噪声服从高斯分布且只添加在全连接层。

2.3.2　基于策略的深度强化学习算法

与上述先学习值函数然后再根据其选择动作的方法不同，基于策略的深度强化学习算法直接学习参数化的策略函数，动作并不直接依赖价值函数而是由策略函数计算得到，比较适合解决高维和连续的动作空间下的任务。同时学习策略函数和价值函数的这类算法称为演员-评论家 (Actor-Critic，AC) 算法。在 AC 算法中，“演员”指策略网络，用于计算策略，“评论家”指价值网络，用于评估某一状态-动作或状态的价值。由策略梯度定理可以得到策略的梯度更新公式如下：

$$\triangle \mathcal{J} \propto \sum_{s} \mu(s) \sum_{a} \triangle \pi(a|s) Q_{\pi}(s,a) \tag{2.18}$$

其中，μ 指的是在策略 π 下的同轨策略分布，详细证明可以见文献[74]。

在基于策略的深度强化学习算法中，比较典型的有 REINFORCE、信任域策略优化 (Trust Region Policy Optimization，TRPO)、软演员-评论家 (Soft Actor-Critic，SAC)、近端策略优化 (Proximal Policy Optimization，PPO) 等算法。策略参数化的方法使得每次策略迭代时所选择动作的变化较为平缓，相对于值方法在理论上有着更快的收敛保证。这类算法更为通用，可以适合任何动作类型，包括离散、连续或者是混合动作，且策略梯度定理给予了算法的局部最优收敛保证，但是这类算法的缺点是方差较高，样本效率低。

在基于策略的深度强化学习算法中，PPO 算法在多个领域得到了很好的应用。一方面，PPO 算法来源于 TRPO 算法，理论上保证了策略的预期回报随着每次更新单调递增；另一方面，PPO 算法使用了一阶优化算法，算法的部署更为轻量，计算和调试的复杂度较低。目前，PPO 算法已在多种问题中取得了很好的效果，也是本文后续研究所涉及到的主要算法之一，故在此进行详细的介绍。

PPO 的优化目标如下：

$$\max_{\pi_\theta'} \mathcal{L}_{\pi_\theta}(\pi_\theta') - \lambda \mathbb{E}_{s\sim\rho_{\pi_\theta}}[D_{KL}(\pi \| \pi_\theta')] \tag{2.19}$$

一类 PPO 算法根据 KL 散度的值，动态调整系数 λ，以此控制相邻策略的变化在合适范围内。但是这种方式需要计算 KL 散度，极大的增加了计算负担。另一种基于策略裁剪的 PPO 算法，控制策略的更新范围在一领域内，具体如下：

$$\mathcal{L}^{PPO-Clip}(\pi_\theta') = \mathbb{E}_{\pi_\theta}[\min(\ell_t(\theta')A^{\pi_\theta}(s_t,a_t)clip(l_t(\theta'),1-\epsilon,1+\epsilon)A^{\pi_\theta}(s_t,a_t))] \tag{2.20}$$

这种基于策略裁剪的方法更易于算法部署，但缺点是同策略的方式，使得每次策略更新后需要舍弃之前的数据，样本的采样效率较低。PPO 算法的具体流程如算法2.4所示。

2.3.3　基于模型的深度强化学习算法

这类算法要求环境模型的转移函数已知，机器可以根据转移函数想象未来时间步发生的事情 (状态-动作序列)。这类算法中最典型的是蒙特卡洛树搜索 (Monte Carlo Tree Search, MCTS)，MCTS 通过选择、扩展、模拟和反向传播四个步骤构建树，利用树的结构进行探索和利用以找到合适的策略。MCTS 对计算能力的要求很高，随着树的层数增加，探索轨迹数呈指数增长。这类算法虽然在棋类游戏中有很好的效果，但是很多现实任务中机器无法获取环境模型，环境中充满着随机噪声且信息不完全可知。

2.4　人在环上强化学习

现实中大部分的任务场景中都缺乏明确的奖励函数，但是训练中不够准确的奖励函数可能会导致策略学习的失败，人在环上强化学习就是利用人类提供奖励等引入人类先验知识的方法，来安全高效地引导机器学会如何决策。

一类常见的方法是修改损失函数，利用人类的轨迹数据去引导机器完成任务。Hester[75]提出了基于演示的 Q 学习 (Deep Q-learning from Demonstrations，DQfD) 算法，机器预先使用单步 Q 学习损失、n 步 Q 学习损失、大边际分类损失以及 l_2 正则化这四种组合的损失函数进行预训练。预训练之后，机器要使用专家数据，其经验回放单元结合了专家演示数据和自我探索生成数据，并且按照比例采样。与 DQfD 相类似，Pohlen 等人提出了另一种扩展方法：深度确定性策略梯度 (Deep Deterministic Policy Gradient from Demonstration，DDPGiD) 算法。DDPGiD 可以被用在连续动作空间的任务中，训练前它将专家演示数据存储在优先经验回放单元中，并在整个训练中保持。Gabriel 等人[76]在强化学习中也

算法 2.4　PPO-Clip 算法流程

超参数：截断因子 ϵ，子迭代次数 M,B；

初始化：策略网络 π，随机生成权重 θ; 初始价值函数网络 V，随机生成权重 ϕ；

for *episode: k=1,2,...,M* **do**

　在环境中执行策略 π_{θ_k}，并保存相应轨迹 $D_k = \{\tau_i\}$；

　计算将得到的奖励 $\hat{G}_t$，并基于当前的价值函数 V_{θ_k} 计算优势函数 $\hat{A}_t$；

　for $m \in \{1, 2, ..., M\}$ **do**

　　计算策略比值:

$$\ell_t(\theta') = \frac{\pi_\theta(A_t|S_t)}{\pi_{\theta_{old}}(A_t|S_t)} \tag{2.21}$$

　　采用 Adam 随机梯度上升优化算法来最大化 PPO-Clip 的目标函数并更新策略：

$$\theta_{k+1} = arg\max_\theta \frac{1}{|D_k|T}\sum_{\tau\in D_k}\sum_{t=0}^{T} min(\ell_t(\theta')A^{\pi_{\theta_{old}}}(S_t, A_t), clip(\ell_t(\theta'), 1-\epsilon, 1+\epsilon)A^{\pi_{\theta_{old}}}(S_t, A_t)) \tag{2.22}$$

　end

　for $b \in \{1, 2, ..., B\}$ **do**

　　采用梯度下降算法最小化均方误差来学习价值函数:

$$\theta_{k+1} = argmax_\phi \frac{1}{|D_k|T}\sum_{\tau\in D_k}\sum_{t=0}^{T}(V_\phi(S_t) - \hat{G}_t)^2 \tag{2.23}$$

　end

end

使用了预训练策略。对于基于视觉的操作任务，Schoettler 等人[77]直接将专家轨迹与强化学习混合，扩展了残量强化学习的概念，其中机器选择的动作被添加到专家控制器执行的动作中，这种方法可以直接用于在现实世界中训练机器。Peng 等人[78]使用专家轨迹作为参考，计算奖励函数，惩罚依赖于专家轨迹和执行轨迹之间的不匹配程度。构造这个奖励函数后，机器可以用任意一种强化学习算法进行训练，并且多种参考奖励可以集成到学习过程中，以便训练出能够执行多种组合技能的机器。

人类的自然语言也可以辅助机器学习。例如，Kaplan 等人[79]训练出了一台能够理解英语单词并将其翻译成游戏状态的机器，与此同时，机器还学会执行这

些命令。一方面，游戏图像作为输入用于训练策略和状态值函数；另一方面，使用带有语言指令的状态向量可以进一步帮助机器理解任务。Kaplan 根据机器是否完成指令，给予其一个额外的奖励或者惩罚。

还有一种方式直接从人类中获取奖励反馈，帮助强化学习算法训练机器。Daniel 等人[80]使用人类来评估机器的行为，不是让人类来设计奖励函数，而是让人类为观察到的轨迹分配奖励数值。由于人类的评价具有噪声且不可重复，Daniel 使用高斯过程和一个噪声超参数来学习人类奖励函数的概率模型。奖励模型学习所用轨迹分为两类：记忆缓冲区中选取的采样表现最好的轨迹，以及人类专家评估的轨迹。Su 等人[81]使用强化学习训练对话策略，根据人类反馈奖励训练模型，可以使用高斯过程来量化不确定性，以减少用户的查询数量。这种算法允许在真实世界的对话系统中在线学习，而无需手动注释数据。

第 3 章　人类策略限制下的人机共享控制方法设计

本章提出了一种基于人类策略限制的人机共享控制决策方法。研究了在共享控制问题中，如何在算法的训练端和执行端引入人类的先验知识，以提升系统决策的性能，并使得机器决策更符合人类意图。具体涉及训练端基于人类策略限制的强化学习算法的设计、共享控制结构和仲裁机制的设计、算法的仿真验证。该研究为后续在训练和执行端引入人类先验知识提供了基础性结构方案。

3.1　引言

在序贯决策问题的众多解决方法中，强化学习算法脱颖而出。强化学习驱动的机器可以在与环境的持续交互中获取训练数据，在解决无模型问题上有很大的优势。如今，强化学习已经在很多领域取得了很好的应用效果，比如游戏，金融，医药等。最近，谷歌利用强化学习成功地在聚变装置中自主控制等离子体[82]。

人机混合智能系统，结合人类直觉智能和机器决策智能，能够更好地完成目标任务。现实场景中，很多问题是人类或机器无法独自解决的。例如，人类很难实现对多臂机器人、四旋翼飞行器等具有高自由度物体的运动和姿态控制，但对机器来说却非常简单；人类能够很轻易地理解任务目标，但机器并不具备这种能力。

本章的研究对象为强化学习驱动的共享自治系统。这一类人机混合智能系统通过结合人类和机器的策略，来完成相关任务或者提高系统决策的性能。共享自治，即在人与机器交互后执行动作，已被证明可以比单个人或机器决策的性能更好，并且易于部署在真实场景中。共享自治系统使用强化学习进行训练，可以在没有环境模型的情况下解决序贯决策问题，扩大了系统的应用范围。

尽管强化学习帮助共享自治系统解决了一类序列决策问题，在该领域取得了很好的应用效果。但是，仅依靠强化学习算法解决许多现实的复杂任务是不可行的。在强化学习算法中，由于动作空间和状态空间的高维特性，机器需要使用大量的交互数据进行训练才能得到好的策略。然而，一方面，机器在现实环境中进行大量探索会带来高昂的成本压力和安全隐患，这阻碍了强化学习在现实世界中的发展。例如，在自动驾驶汽车的训练过程中，一旦系统执行出错容易造成交通事故和车辆损毁。另一方面，强化学习算法容易过拟合，这意味着一旦实际环境相比训练环境发生很大变化，原始策略就会变得无效。

为了解决这些问题，一些研究者提出利用人类先验知识来训练算法。例如，

在探索过程中引入专家策略以避免机器做出危险行为，设置特定的策略约束函数，以及让机器通过人类演示来学习目标策略。Siddharth 等人[11]提出了一种人在环上强化学习算法，该算法使用人类奖励反馈指导机器训练，加速了机器的学习速度，但忽视了强化学习训练过程中的问题: 机器探索行为缺乏安全保证。在探索过程中，机器容易做出危险的行为，从而可能造成巨大的损失。此外，一方面，引入人类奖励的方式虽然能帮助机器学习策略，但所得策略并不一定最优。事实上，由于人类奖励评估的非平稳性，最终算法可能会收敛到次优解。另一方面，上述算法在执行过程中默认人类的决策必定正确，但忽略了一些人类可能做出错误决策的情况。例如，人类在驾驶过程中过于疲劳以致操控方向盘时失误，此时辅助驾驶系统不应该依据人类错误的输入推理出人类意图。

人在环上的共享控制算法的设计重点在于人类如何参与训练过程，以及在执行过程中，人类和机器决策仲裁机制的设计。本章提出了一种策略约束下的人在环上强化学习方法 (Human-in-the-loop Reinforcement Learning with Policy Constraints，HRLPC) 来解决上述问题。针对强化学习算法缺乏安全保障、易收敛到次优解等问题，本章算法引入多个被包含人类先验知识的不同策略限制集合约束下的机器，然后在训练过程中，每次以多臂赌博机的模式选择其中一个机器进行探索，并共享探索数据，每次更新网络参数后进行策略消除以去除策略耦合的机器。这种方法加速训练的同时，通过设计策略限制集合可以控制机器探索行为的危险程度。其中，策略限制可以是人为设计的硬规则约束或者是通过其他学习算法训练得到的软约束策略。与训练过程中人类直接手动约束机器相比，这种方法降低了人类的训练负担。一方面，在训练过程中，本章算法添加了不同类型的策略来约束机器的探索行为，在保证安全的前提下，训练结束可以得到具有最优策略的机器。另一方面，在执行过程中，本章算法将策略评估和策略限制引入了仲裁机制，以限制人类和机器做出的不合理决策，使人机系统的决策更为安全和高效。

综上所述，本章主要的贡献点如下：

在训练阶段，提出一种策略限制下的人在环上强化学习算法训练人机系统中的机器。算法并不要求环境的模型已知，且通过人类策略限制使得机器的探索更高效安全；

在执行阶段，提出一种包含策略评估和策略限制的仲裁机制。算法能够滤除机器和人类不合理的决策，提高系统决策的安全性和鲁棒性。

本章结构安排如下，第 3.2 节给出了人机共享控制系统的建模，以及策略限制下强化学习算法的建模；第 3.3 节阐述了算法的具体设计和实现；第 3.4 节介绍了实验设置和仿真结果，并且讨论与分析了训练和执行两个阶段的结果；最终第 3.5 节总结了本章的研究内容。

3.2　问题建模

3.2.1　决策建模

已有的工作已经将人机共享控制建模成 POMDP 问题[61]，模型的奖励函数记为 R，转移函数记为 T，任务目标集合记为 $g \in G$，原始状态空间记为 S。由于目标集合的存在，状态空间可以拓展为 $S' = S \times G$，状态转移拓展为 $T'((s_t, g)|s_t, g, a_t) = T(s_{t+1}|s_t, a_t)$。人类策略记为 $\pi_h : S \times G \times A \to [0,1]$，人类动作为 a_h，观测概率函数为 $O(s, a_h|a, g) = \pi_h(a_h|s, g)$。当模型转移函数、目标空间和人类策略已知时，可以使用“后见之明”的优化方式求解[61]，但本章考虑的问题中环境转移模型未知，因此使用强化学习来求解，整个训练过程的流程如图3.1所示。模型由三部分组成：人类奖励模块、人类策略限制模块以及意图推理模块。

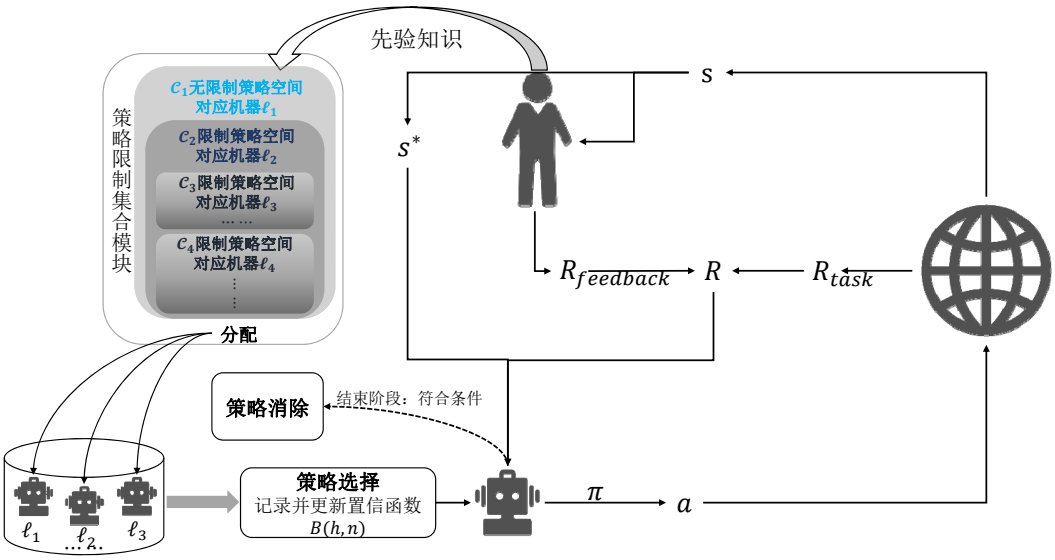


图 3.1　基于策略限制下深度强化学习的训练流程图

1、人类奖励相对于任务奖励函数而言是稀疏的，但是却可以很好的引导机器完成目标任务。任务奖励函数用于帮助机器学习相应的能力，当任务结束时，人类会根据其完成程度给与相应反馈奖励，使机器能够适应人类目标。

$$R = R_{task}(s, a, s') + R_{feedback}(s, a, s') \tag{3.1}$$

当目标空间已知时，可以训练神经网络根据人的输入动作输出预测目标，当目标已知时，反馈奖励则是可以直接通过比较预测目标和人类目标的差值得到。

2、人类策略限制 C 为一个将状态映射到可选动作集合的函数，即 $C(s) = \{a_i, a_k, \ldots\}$。给定一个由 k 个策略限制组成的集合 C，记 C_k 为第 k 个策略限制，

当任意状态下，策略 π 只采取 C_k 允许的动作，即 C_k ：$\forall(s,a),\pi(a|s)>0$ 仅当 $a\in C_k(s)$ 时，此时称策略 π 满足限制 C_k。

策略限制模块，包含一个基准强化学习算法 Alg，一组潜在的策略限制集合 C，以及一个置信界函数 $B(h,n)$。策略限制模块类似于一个多臂赌博机 (假设一共有 $|C|$ 个臂)，每个臂对应一个基于基准强化学习算法 Alg 训练的机器 l_i，每个机器 l_i 有着对应的策略限制 $C_i\in C$，每个机器 l_i 只会选择符合对应策略限制 C_i 的动作，记所有机器组成的集合为 $\mathcal{L}=\{1,\ldots,|C|\}$。每个回合由策略选择、策略更新和策略消除三步构成，详细算法介绍见下一节。

3、执行过程中，机器状态的输入为从环境观测的状态信息 s_t 以及人的相关信息 a_h，即

$$\tilde{s}_t^*=\begin{bmatrix}s_t\\a_h\end{bmatrix} \tag{3.2}$$

与训练过程中目标是固定的不同，在执行过程中，机器需要根据人的输入推理出目标。通过连接机器的原始状态向量和人类行为向量，可以将人类的意图传递给机器。显然，如果知道足够多的模型信息，机器也可以从人类的历史行为中直接推断出人类的目的，并串联机器的原始状态向量和目标位置向量。

3.2.2　仲裁建模

泛用性最强的仲裁函数为线性仲裁函数，因此本节先以此为例阐述其原理。假设 t 时刻系统状态为 s_t，人的动作为 a_h，机器的动作为 a_m，各自的策略分别记为 π_h 和 π_m，即：

$$\begin{aligned}a_h(t)&=\pi_h(s_t)\\a_m(t)&=\pi_m(s_t)\end{aligned} \tag{3.3}$$

仲裁机制的输入为人类和机器的共同输入，输出为最终的系统决策，记仲裁函数为 $\beta(\cdot)$，最终系统输出动作为：

$$a(t)=\beta(a_h(t),a_m(t)) \tag{3.4}$$

线性仲裁的具体表示如下：

$$\beta(a_h(t),a_m(t))=(1-\alpha)\cdot a_h(t)+\alpha\cdot a_m(t) \tag{3.5}$$

其中，$\alpha\in[0,1]$ 用于控制人类和机器的决策权重，当 α 为 1 时，系统完全由机器控制，α 为 0 时，系统完全由人类控制。值得注意的是 α 可以是参数化形式的函数，而并非固定的常数。线性仲裁函数的形式简单，但是如果能设计参数形式，能够使得系统的性能得到极大的提升。

上述仲裁函数 β 的本质其实是相似度函数，或者是距离度量函数。虽然线性仲裁具有简单好用的特性，但是局限性很大。一方面，对于非连续的动作，显然不能通过加权求和的方式获得最终动作。另一方面，加权求和的方式，本质上是在决策空间的两个点的欧氏距离上找折衷点，其度量方式限定在欧氏空间中。但是，目前研究领域存在多种度量方法，比如余弦相似度、汉明距离、曼哈顿距离、闵可夫斯基距离等。显然，对于不同问题每种度量方法的效果不尽相同。执行阶段整体的决策流程如图3.2所示。

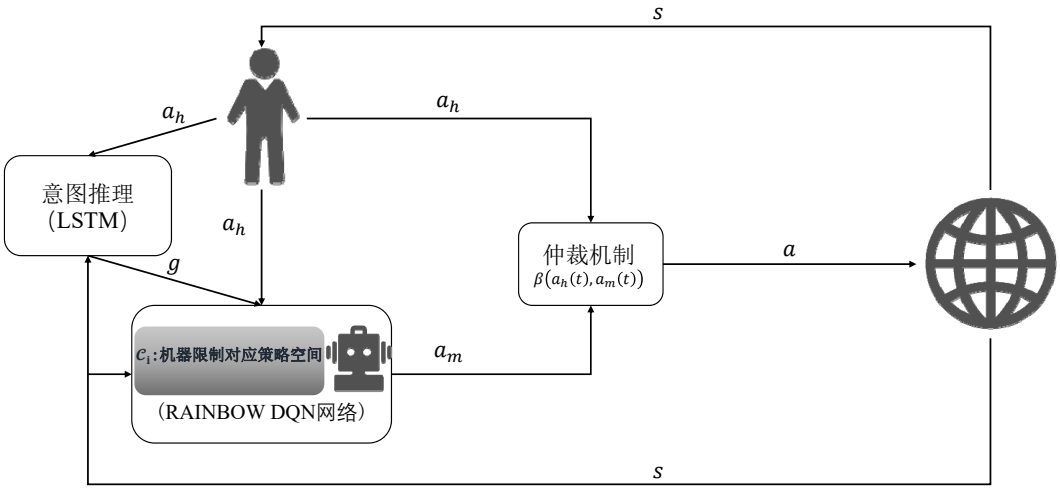


图 3.2　基于策略限制下深度强化学习的人机混合决策执行流程图

3.3　人类策略限制下的人机共享控制算法设计

基于 3.2 节的建模，本节将会详细阐述方法的设计与实现。整个算法分为训练阶段和执行阶段，训练阶段主体由策略限制下人在环上的强化学习算法构成，单回合训练过程需要经历策略选择、策略更新、策略消除三阶段；执行阶段主体为人机共享控制，由目标推理和仲裁机制两部分组成。

3.3.1　策略限制下的人在环上强化学习算法

本章提出的 HRLPC 算法使用 RAINBOW DQN[83]作为基准强化学习，算法具体的训练过程如下所示：

- 策略选择：从集合 $\mathcal{L}$ 中选择一个机器 l_i 进行探索。本章使用类似多臂赌博机中置信度上界 (Upper Confidence Bound，UCB) 方法，选择潜在估计回报上界最大的机器。UCB 机制根据过去机器 l_i 决策期间观测回报的平均先验

值 μ_i 以及输入置信函数选择动作。具体公式如下所示：

$$k = arg\max_{l\in\mathcal{L}}(\hat{\mu}_l + B(h, n_l)) = arg\max_{l\in\mathcal{L}}(\hat{\mu}_l + \frac{z(h)}{n_l^{\eta}}) \tag{3.6}$$

其中，置信函数方程中的 $z(h)$ 为一个非递减函数，超参数 $0 < \eta \leqslant \frac{1}{2}$，且 $O(z(h)^{\frac{1}{\eta}}) < O(h)$，$h$ 表示总的回合数，n_l 表示选取机器 l 进行探索的总次数。算法只依赖于每个机器的观测回报，而不需要准确预估值函数，每个机器的平均回报估计类似于赌博机每个臂的平稳回报。与传统多臂赌博机中假设每个臂的回报是平稳分布不同，这种方法下策略处于变化过程中，此时的估计上界并不代表真实的上界，其预期收益带有一定的随机性。但是，Tong 等人[84]的研究已经证明 UCB 能够以一个有效的方法，解决非平稳回报的多臂赌博机问题。Tong 提出一种蒙特卡洛树搜索方法，可以根据非平稳回报的上界，对树中的动作展开搜索并进行下游行动决策的优先级排序。如同 Tong 所证明的，当基准强化学习方法收敛时，即使 UCB 方法中每个臂的收益不服从稳态分布，这种约束采样方法也一样能保证收敛。

- 策略更新：收集 l_i 训练过程的状态、动作、奖励组成的轨迹数据，数据可以同时用来更新自身和所有相关机器的策略。Siddharth 等人[11]指出，由于任务界面的延迟，单步更新会影响人工控制，所以本章选择在每回合结束后再更新模型参数。
- 策略消除：使用启发式方法，当满足对应条件时从集合 $\mathcal{L}$ 中消除机器 l_i。该方法将会检查当机器 l_i 的值函数或状态-动作值函数是否已经稳定 (即算法是否收敛)。当机器 l_i 算法收敛时，如果平均表现比另一个机器 l_j 差，且机器 l_i 的限制集合 C_i 被机器 l_j 的限制集合 C_j 所包含，此时将消除机器 l_i。当机器 l_i 探索第 n 次时，对应生成了第 n 条轨迹 τ^n，用 δ_i^n 表示值函数的变化，可以通过测量状态-动作的平均差值得到绝对差值，即：

$$\delta_i^n = \frac{\|Q_i^{n-1} - Q_i^n\|}{|S||A|} \tag{3.7}$$

其中，Q_i^n 表示使用刚刚收集的数据更新后的状态-动作值函数。对于强化学习中值方法，可以直接使用轨迹的平均估计误差来近似表示值函数的变化值，即：

$$\delta_k^n = \sum_{\tau^n}(r_t + \max_{a\in C_k(s_{t+1})} Q_k^n(s_{t+1}, a)) - Q_k^n(s_t, a_t) \tag{3.8}$$

本章使用值变化程度来衡量算法是否已经收敛。当值变化在至少 T_n 个连续时间步中低于某个阈值 T_i 时，机器的学习过程已经稳定，如果满足该条件则会触发策略消除机制：当存在另一个约束更小的机器时，前一个机器将会被消除，算法的具体流程如算法3.1所示。

算法 3.1　基于策略限制的人在环上强化学习 (HRLPC) 算法训练流程

初始化 μ_k，$B(h,n)$，T_n，T_l 和机器集合 $\mathcal{L}$;
初始化在不同策略限制集合 C_i 的机器 l_i $\forall i \in \mathcal{L}$;
初始化机器的参数和超参数 ;
初始化值变化 $D_k = [], \forall k \in \mathcal{A}$ **while** *episode=1,2,...,M* **do**
    根据公式3.6选择机器 l_k 进行探索 ;
    生成轨迹 τ_h ;
    $R_{task} = \sum_{t=1}^{len(\tau_h)} r_t$;
    $R_h = R_{task} + R_{feedback}$;
    $n_k = n_k + 1$;
    $\hat{\mu}_k = \frac{(n_k-1)\hat{\mu}_k + R_h}{n_k}$;
    更新相关机器的网络 ;
    计算 δ_h 以及 $P_k = [D_k, \delta_h]$;
    if $\forall n \in \{n_k | n_k - T_n \leqslant 0\}$ 且 $D_k(n) \leqslant T_l$ **then**
        消除机器 l_k
    end
end

3.3.2　共享控制

共享控制方法中的两个核心组件为意图推理和仲裁机制。与训练过程中目标是固定的不同，执行阶段机器的观测信息为环境传入信息和人的信息的总和，即公式3.3中的 a_h 是由意图推理网络根据历史状态和人类输入动作输出的预测目标 g'。

意图推理：由于系统中人的决策具有典型的时序性，本章使用长短时记忆网络 (Long Short-Term Memory，LSTM) 进行目标推理的学习。LSTM 是一种特殊的递归神经网络 (Recurrent Neural Network，RNN)，能够解决长程序列训练过程中出现的梯度爆炸和梯度消失问题。LSTM 和 RNN 一样，具有短期记忆的能力，可以用于训练序列到类别模式、同步的序列到序列模式、异步的序列到序列模式的任务，在文本生成、机器翻译、语音识别、生成图像描述和视频标记等领域取得了很好的应用效果。

记一组目标集为 G，以无人机降落问题为例，G 就是所有可能目标着陆点的集合。机器需要在一组目标集中思考人类的目标是哪一个，本章以 LSTM 为意图推理网络，以人类的历史行为和系统的历史轨迹作为输入，输出预测目标的概率分布，每次选取概率最高的目标作为预测值。

意图推理的输出结果作为机器输入的重要一节，其精确程度影响着后续系

统的执行。但是，输出目标是存在不确定性的，这取决于 LSTM 输出目标概率分布的形状。为了避免错误的意图推理影响后续系统的正常运行，本章定义了意图推理的置信度，当置信度过低时，目标推理就很有可能是失败的，此时不该将预测目标作为人类真正的目标。意图推理置信度的具体公式如下所示：

$$c = \max_{g} p(g|a_h) - \min_{g} p(g|a_h) \tag{3.9}$$

此处使用分布中最大概率减去最小概率，是目前衡量分布不确定性最常见的一种方法。当 $c = 1$ 时，即置信度为 1，显然此时分布中只有一个目标的概率为 1 其余目标概率为 0，表示此时机器完全明白人类意图。当 $c = 0$ 时，即置信度为 0，此时概率分布为均匀分布，表明机器完全不知道人类的真实意图，LSTM 输出目标为均匀随机采样所得。

仲裁机制：系统需要一个能够根据机器和人类的输入确定最终动作的仲裁机制，由于本章使用的时 RAINBOW DQN 算法，可以使用 Q 值函数评估当前状态下动作的价值。如果推理的置信度非常低，那么人类没有必要相信系统，这将导致人类的输入信息变少，意图推理的输入变差导致其输出的准确性变低，系统进入恶性循环。如果推理的置信度很高，那么机器的预测目标准确度就很高，此时系统可以选择与机器最优决策最接近的动作。本章所设计的可选动作空间如下所示：

$$\Omega_c = \{a|Q(s,a) - \min_{a'} Q(s,a') \geqslant c(\max_{a'} Q(s,a') - \min_{a'} Q(s,a'))\} \tag{3.10}$$

可选动作空间 Ω_c 与置信度 c 相关，当置信度越高时，不等式的解越少，为 1 时只有最大 Q 值对应的动作，为 0 时，可选动作空间为整个动作空间。但是，如果只在 Ω_c 中选取动作，就会忽略人类误操作的可能。因此，本章引入判别动作空间 Ω_h：

$$\Omega_h = \{a|Q(s,a) - \min_{a'} Q(s,a') \geqslant \frac{c}{\alpha}(\max_{a'} Q(s,a') - \min_{a'} Q(s,a'))\} \tag{3.11}$$

其中，$\alpha > 1$ 为可调超参数，当人类动作不属于判别动作空间时 Ω_h，意味着此时 Q 函数认为人类的行为会带来不好的后果，此时需要忽略人类的动作。当机器对人类置信度越高，系统对人类行为的要求 (Q 函数值) 也越高。

当人类的行动是错误的时候，只使用可选动作空间 Ω_c 容易造成严重的损失。所以本章在原有仲裁的基础上引入了判别空间 Ω_h，当人的行为出错时，系统的动作由机器主导。最终系统的仲裁策略如下所示：

$$\pi(s) = \begin{cases} arg\max_{a \in \Omega_c} \rho(a, a_h) & if\, a_h \in \Omega_h \\ arg\max_{a \in \mathcal{A}} Q(s,a) \end{cases} \tag{3.12}$$

其中，$\rho(a, a_h)$ 是相似度度量函数，用于衡量两个动作的相似度，此处需要选择和人类动作相似度最高的动作。上述策略仲裁函数并不要求人类时刻参与决策，当人类没有动作输入 a_h 时，说明人类对机器的决策是满意的，此时完全由机器独自控制系统。

3.4　仿真实验

本章使用 OpenAI 所发布的 Lunar Lander 月球登陆器作为仿真环境，以验证算法的有效性，火箭着陆问题能够近似类比无人机着陆问题。如图3.3所示，这个游戏是用来模拟飞行器在月球上的降落，它的目标是在规定时间内，安全平稳地降落在某个目标点，一旦着陆器超时、坠落、飞出边界或者抵达着陆点之外的陆地时，任务视作失败。着陆器的运动空间是四维且离散的，分别对应于不同位置的引擎：空操作、着陆器的左引擎、右引擎以及主引擎。

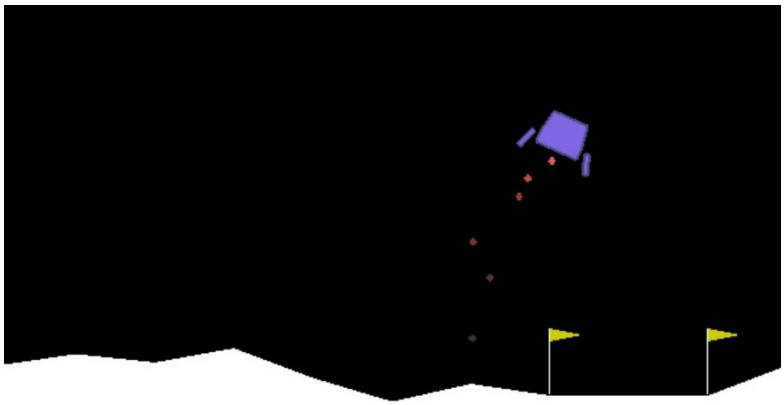


图 3.3　OpenAI GYM 月球登陆器场景示意图

原始状态空间是一个 8 维向量，包含着陆器的位置 (x, y)、水平速度 v_x、垂直速度 v_y，角度 θ、角速度 ω 和左右支撑器是否到达地面的指标 (I_l, I_r)。在这个基础上本章实验修改了游戏设置，在每一回合开始时，地面上会生成着陆器的目标着陆点：两个旗帜之间的水平区域组成目标着陆点，且每回合随机生成。目标着陆点对机器来说是不可识别的，但人类知道目标着陆点，因此每一回合机器都需要根据人类的输入预测目标。最终系统的状态空间是 9 维的，最后一维为有关目标位置的附加信息 g。

原始动作空间为 4 维的离散动作空间 $A = \{0, 1, 2, 3\}$，0 表示不执行任何操作，1 表示点燃左引擎，2 表示点燃主引擎，3 表示点燃右引擎。但是，使用该动作空间控制着陆器的方法过于简单，每个时间步只能控制其中一个引擎，控制程度不够精细。因此，本章重新编码了动作空间，将原始的动作空间重新编码为

$A' = \{0,1,2,3,4,5\}$，实现同一时间步可以控制多个引擎，动作值与控制规则的对应关系如表3.1所示。

表 3.1　动作值和发动机开关的对应关系

动作值	左发动机	主发动机	右发动机
0	关闭	关闭	开启
1	关闭	关闭	关闭
2	开启	关闭	关闭
3	关闭	开启	开启
4	关闭	开启	关闭
5	开启	开启	关闭

本节使用的动作的相似程度函数 ρ 为汉明距离。例如，动作值 0 对应的编码为“001”，动作 1 对应的编码为“000”，两个动作的距离 $\rho(0,1)=1$，类似的 $\rho(1,2)=3$。奖励设置中，任务奖励 R_{task} 与速度、倾斜角度和、目标距离以及最终支撑器与地面接触情况相关。反馈奖励 $R_{feedback}$ 与任务最终完成与否相关，任务成功为 $+200$ 失败则为 -200。

本节生成了 10 个不同的人类专业操作员水平的策略，最终的策略限制集合是其中单个或者多个策略的组合，这种方法生成的策略约束为软约束。集合中存在某个策略限制为空的可能，即意味着没有限制。本章使用的 Q 值函数是由一个具有两个隐藏层且每层拥有 128 个神经元的前馈神经网络构成。算法的学习率为 1e-4，目标网络参数的更新间隔为 4 个时间步，训练批次大小为 64。上置信界函数 $B(h,n)=c\frac{\sqrt{\log(h)}}{n^{1/2}}$，策略消除中判断算法稳定的值变化阈值 $T_i=0.01254$，时间阈值 $T_n=40$。有关上述一些超参数的影响在相关文献中已有讨论[84]，故不在本章实验的范围内。

首先，本节将会进行训练阶段算法决策性能的对比试验，以研究策略限制模块给算法训练带来的影响。然后，在执行阶段进行多组对比试验，以研究意图推理模块和策略限制模块对执行阶段算法决策性能的影响。最终，结合上述两阶段的实验结果，验证和分析本章所提出算法的性能。

3.4.1　训练阶段

本节进行算法训练的对比实验，研究人的策略限制对算法学习性能的影响。为了减轻人类的训练负担，意图推理模块和决策模块进行解耦训练：一方面，着陆器在训练时会被告知目标位置，但在执行阶段，机器需要根据人类输入推理出目的地信息。另一方面，采集人类飞行员的行动轨迹和目标数据，用于训练意图推理模块。最终实验结果如图3.4所示，本章绘制的所有奖励都具有 95% 的置信

区间。从训练回报曲线3.4中可以发现，与 RAINBOW 方法相比，本章所提方法 HRLPC 提高了收敛的速度和回报的稳定性。

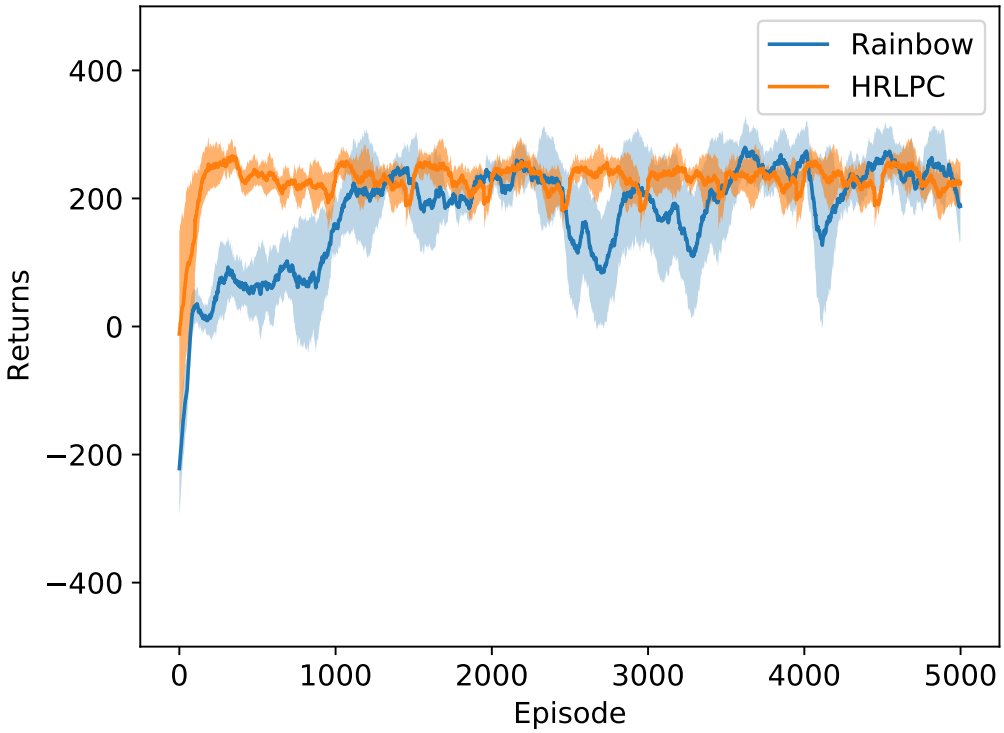


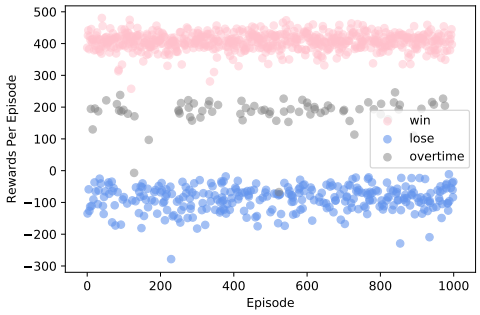
图 3.4　95% 置信区间下的奖励回报图

显然，在机器具有策略约束的情况下，机器可以快速找到最优策略，降低了机器训练所需的探索成本。当机器具有一个严格安全的策略约束时，可以避免训练过程中机器做出会造成严重损失的决策。虽然，实验结果表明使用该方法所获得的最终稳态奖励与 RAINBOW 算法相比没有太大的优势。但是，从图3.4中可以发现，RAINBOW 算法在训练阶段后半程的平均回报十分振荡。这是因为月球着陆器中的动作空间维度较低，在目标已知的情况下机器的探索并不困难，次优解与最优解之间的差异很小，长期的训练可以使得最终收敛所得策略的平均回报不低。但是，从回报曲线的振动程度来看，当不同回合之间目标的变化较大时，RAINBOW 算法的平均回报并不稳定，这意味着策略并未收敛到最优解。

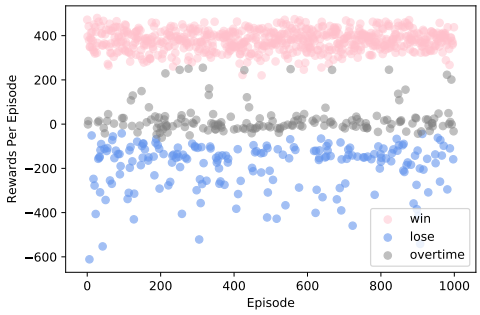
RAINBOW 算法经过长时间训练后的平均回报不低，但对于强化学习算法而言，长期训练会使得算法过拟合。在执行阶段，机器需要通过推理才能获取目标，意图推理模块一旦出现轻微的偏差就会导致算法的决策失误。因此，训练阶段机器并不需要训练 5000 回合之久。HRLPC 算法在策略约束机制的作用下，可以更快速稳定地进行探索，算法可以提前中止训练以获得合适的策略 (比如 1000 回合)，这种方法可以避免算法出现过拟合的情况。

3.4.2　执行阶段

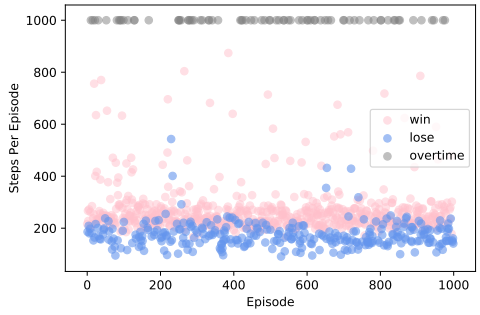
本节介绍人机共享控制系统执行阶段的仿真实验，研究策略限制模块对整个人机系统的决策性能的影响，以及嵌入意图推理后系统性能的变化。在执行过程中，此次实验邀请了 12 名志愿者 (5 名女性和 7 名男性) 参与。为了避免因为有人不认真参与而导致实验结果不公平，实验为每个参与者提供了奖金，成功率最高的人还会获得额外的奖金。每位志愿者都提前单独试用 30 回合以便熟悉界面和相关操作，随后志愿者和两种算法 (RAINBOW 与 HRLPC) 训练的机器分别进行 30 次协同操作月球着陆器完成任务 (此时目标需要根据人类输入进行推理得到)。在假设目标已知情况下，两种算法训练的机器分别进行 30 次任务。收集志愿者单独操作着陆器的数据，在进行分析后可以发现，即使是目标固定的情况下，所有人独自操作月球着陆器的任务成功率不到 10%。对于这类需要操作者迅速精确地操控高速运动物体的任务，非专业的人类很难完成，这是因为与机器相比，人类缺乏高频动态调整的能力。



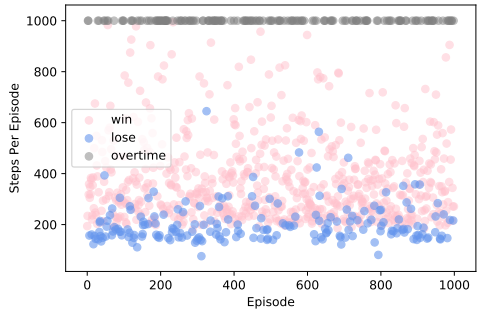
(a) RAINBOW 算法奖励分布



(b) HRLPC 算法奖励分布



(c) RAINBOW 算法步数分布

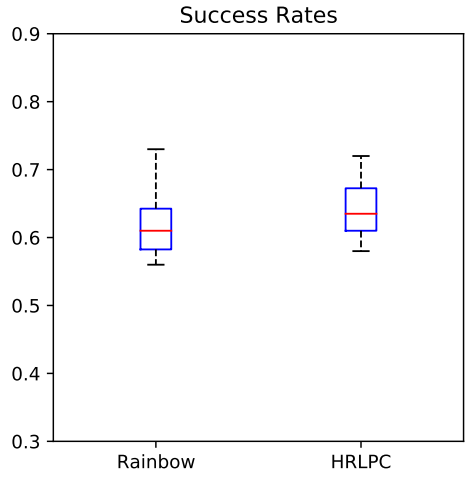


(d) HRLPC 算法步数分布

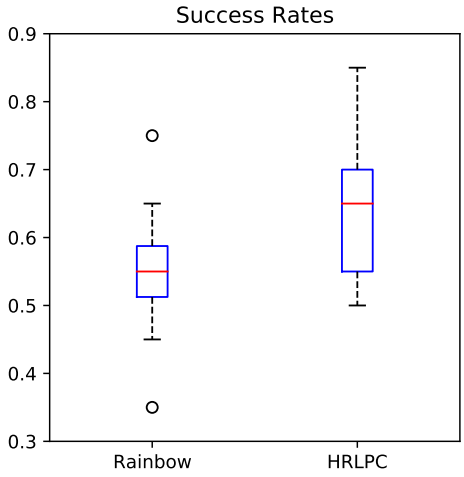
图 3.5　1000 回合内机器收到的回报和单回合步数的散点图：实验是在没有人类参与的情况下进行测试的，此时每回合机器会被直接告知目标位置。

由于执行阶段目标的分布较为随机，环境变得不稳定，机器策略会变得很不稳定。在训练阶段，可以将目标信息编码到状态向量前添加较小的高斯噪声，这种方法能够提高算法的泛化能力。最终，实验对比结果如图3.5和图3.6所示，可以发现在执行阶段，两种算法驱动的人机系统决策性能均比较稳定。

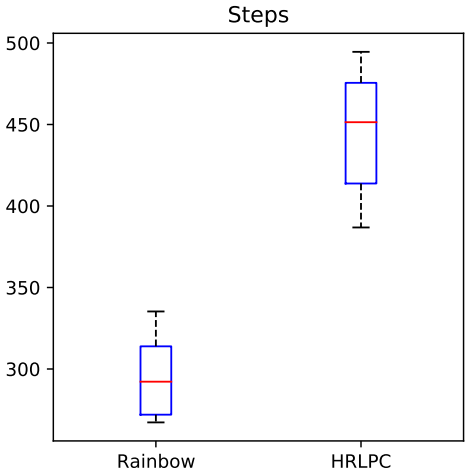
结果：从实验结果来看，本章所提出的 HRLPC 算法明显领先于 RAINBOW 算法。在散点图3.5中，HRLPC 算法驱动的机器坠毁率较低，但平均运行时间较高。RAINBOW 算法无论输赢平均运行时间较低，但坠毁率较高，这说明系统的安全性较差。从箱形图3.6中可以看到，当考虑更现实的共享控制场景，即机器无法直接获取目标时，人类参与下 HRLPC 的胜率得到了显著提高，而 RAINBOW 的胜率却显著下降。出现上述这种情况，正是因为虽然在训练阶段 RAINBOW 算法能够收敛，但是存在较大的震荡，说明此时策略并未收敛到最优解。这一点在执行过程的表现则是：目标不可直接获取时，推断误差会造成致命的失误，此时任务的成功率也会大大下降。



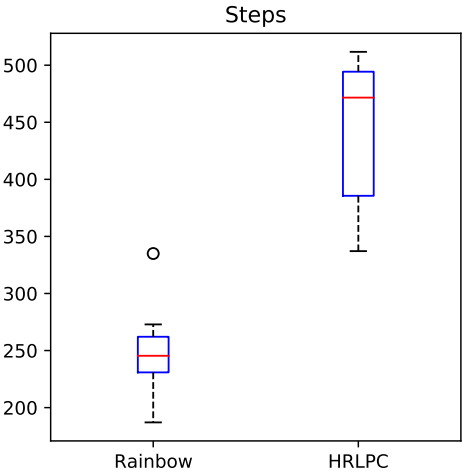
(a) 执行阶段成功率对比 (人类不参与)



(b) 执行阶段成功率对比 (人类参与)



(c) 执行阶段平均步数对比 (人类不参与)



(d) 执行阶段平均步数对比 (人类参与)

图 3.6　执行过程中，在人类参与和不参与的情况下，机器的胜率和步数的箱形图。

分析：如果仅从训练结果看，HRLPC 与 RAINBOW 相比，最终的平均回报相差不大，只是在收敛速度上有优势。执行阶段的实验结果也表明，当机器提前

知道目标时，RAINBOW 的成功率仅略低于 HRLPC。实际上，由于行动空间只有 6 维，机器的探索范围并不大，经过长期的探索，最终还是能找到近似最优解的次优解。但长时间不受约束的探索导致了很多冒险行为，这也是 RAINBOW 在训练和执行过程中都表现得非常激进的原因，不仅胜率较高，坠毁率也很高。从散点图中可以发现，没有策略约束的机器在执行过程中容易崩溃，只有当初始目标点恰好在正下方附近时，机器不需要太多调整就可以快速着陆。

人类参与后 RAINBOW 算法训练的机器最终平均运行时间下降，成功率也下降，这意味着机器的坠毁率变高，安全性变差。相反，HRLPC 算法训练的机器虽然平均步数略有增加，但是成功率反而有所增加，说明算法的鲁棒性和安全性更强，能够允许一定程度的意图推理和感知误差，对于目标随机生成的范围有一定的容忍度。当机器不被允许拥有额外的信息 (目标位置) 时，理论上整个系统的性能可能会下降。由于着陆器的着陆速度和自由度较高，仅靠 6 个离散动作很难使其平稳顺利地着陆。虽然经过强化学习算法训练的机器可以辅助人类，但是它们也需要根据人类的输入来预测目标位置。由于 LSTM 网络本身的误差，预测目标与实际目标之间存在一定的差距。这也就导致实验结果显示，HRLPC 的平均步数随着成功率一起提高，而 RAINBOW 的平均步长和成功率同步下降。这是因为在执行阶段，机器的策略约束仍然存在，机器会额外考虑人类输入行为的合理性。只要人类不总是传递错误的信息，系统就可以轻松完成任务。与此同时，机器会忽略不安全的人类输入行动，让系统尽可能安全地运行。对人类无效输入的舍弃会影响目标的推理，这也是 HRLPC 算法平均步长增加的原因。在失去目标时，着陆器会消耗更多的时间维持一定高度，直到成功确认目标后再进行着陆。

人类评估：为了评估志愿者对两类机器自主性程度的判断以及对整体系统的满意程度，本节设计了相关的调查问卷让志愿者进行评分，最终结果见表3.2。10 分表示强烈认同，0 分表示强烈反对，志愿者匿名填写分值。根据表格的结果可知，本章提出的 HRLPC 算法得分高于原本的 RAINBOW 算法，人类的满意程度更高，机器与人类的配合效果更好。

表 3.2　参与者对调查问题的回答 (对表述的同意程度)

调查表述	RAINBOW	HRLPC
智能机器的帮助对完成任务很有用	7.4	7.9
智能机器做了我想做的事.	7.2	8.2
在机器的帮助下，我更好地完成任务	8.1	8.6
机器的帮助使我感到困扰	3.8	1.3
我对这个决策系统感到满意	7.6	8.8

3.5　本章小结

针对基于深度强化学习的共享自治系统，本章提出了具有策略约束的人在环上强化学习算法，利用人类先验知识来设计策略约束集合。一方面，通过约束机器的探索可以帮助其快速安全地收敛到最优策略，降低了训练成本；另一方面，包含策略评估和策略限制的仲裁机制能够保证执行过程中系统决策的安全性和稳定性。最终实验结果表明，本章所提出的方法极大地提高了机器训练的采样效率和安全性，提升了执行过程中系统的性能和任务的成功率。这种将人类先验知识同时引入训练和执行两个阶段的形式，为后续进一步研究奠定了基础。

第 4 章　面向多机竞速的人机介入控制方法设计

第 3 章提出了基于人类策略限制的人机共享控制方法，利用人和机器智能互补的方式，提升系统的决策效果。虽然这种策略限制的方法能够让机器进行安全高效的探索，但是存在一定局限性：一方面，现实中存在很多任务，人类难以将所有可能的应对策略全编码成策略限制；另一方面，共享控制方法要求人类参与频率够高，机器才能够根据历史状态轨迹和人类的历史行动轨迹推测出目标，此时人类的操作负担较重。

本章面向多无人机竞速问题，提出了一种基于人类反馈的人机介入控制方法。多机竞速场景具有强动态性和不确定性，任务规则对机器而言比较复杂，能够作为很多现实任务的典型代表，具体体现在：一方面，该问题要求算法具有足够的鲁棒性，能够应对突发扰动等随机情况；另一方面，该问题要求无人机遵守竞速规则，不做出违反体育精神的行为。本章研究如何通过人类奖励反馈辅助机器训练，以引导机器学会人类的“竞速规则”，以及如何通过介入控制保证系统决策的安全。具体涉及人类反馈奖励塑造方法、人介入机器的机制设计、相关的仿真实验。该研究为后续算法在真实物理平台进行仿真验证奠定基础。

4.1　引言

序贯决策问题广泛存在于各种游戏中，比如 Atari、围棋、星际争霸、DOTA 等，深度强化学习早已证明其在这些任务中的潜力。研究人员如何将深度强化学习成功应用在机器人和自动化领域中，如何使用强化学习成功控制复杂的物理系统，对于设计能够完成复杂现实任务的人机混合智能系统而言，有着重要的指导意义。机器在完成任务的过程中需要与人交互，遵守没有精确化定义的规范。

在诸多现实任务中，无人机竞速问题正是一个非常具有挑战性的任务，比赛过程中所需要的感知、决策、规划和控制等技术，对机器人领域的发展具有重要意义。多无人机竞速中无人机之间的丰富互动，以及对无人机在复杂环境中能够快速灵巧且安全飞行的要求，都是对导航和控制技术的巨大挑战。人类选手也需要经过多年的训练才能在这样的比赛中取得不错的成绩。

无人机竞赛问题近年来吸引了学术界和工业界的大量关注，洛克希德•马丁公司以及一些会议，如 IROS 和 NIPS，举办了比赛来推广这一领域。现有的大部分研究都集中在如何使无人机在单无人机计时赛中达到或超过专业人类飞行员的飞行水平。在多无人机竞速比赛中，由于多架无人机同时参与比赛，无人机之间的交互会对比赛结果产生关键影响。比赛中选手之间频繁的交互会导致场上

局势迅速变化，选手必须根据场上的情况不断调整策略，比如阻挡、超车，有时甚至需要与其他无人机合作。除了考虑如何与其他玩家交互以外，选手还需要确保他们能够正确安全地通过赛道上所有的门。然而，到目前为止，针对多无人机竞速问题的研究还很少。Wang 等人[85]提供了一种多无人机竞速方法，该方法基于博弈论考虑了无人机之间的交互，其灵感来自传统的无人驾驶汽车自主竞速。这些游戏场景下的多无人机竞速方法，需要预先获取赛道的全局信息和很多的人为约束。首先，这类方法需要事预先了解整条赛道上的所有门的位置信息，然后根据赛道门拟合赛道的中心线和赛道边界。但是，在真正的比赛中无人机无法获得全局赛道信息，只能在其自身机载传感器的范围内获得有限的局部赛道信息。此外，算法有额外限制，要求无人机不能超过赛道边界，但这种限制在实际的无人机比赛中并不存在，而人为划定的赛道边界可能会导致无人机做出更保守的行为。最关键的是，基于博弈论的多无人机竞速算法在实际应用中要求无人机在线求解优化问题，且计算耗时随着预测步数和参与者数量的增加而增加。这种对计算能力的高要求，导致无人机需要携带重型高性能计算设备，或者在博弈算法求解出结果之前减速，但这些都与无人机竞速目标背道而驰，无人机需要减少负荷才能够更快速的机动。

解决这些问题的关键是赋予无人机在动态环境中对抗其他对手的自主能力，并减少其计算负担。强化学习可以赋予机器自主学习的能力，机器与环境交互后接收反馈并不断改进自身策略，最终达到甚至超越人类的专业水平。同时，强化学习算法的策略网络在单步时间内可以快速生成动作命令，无人机的计算负担更小，具有更高的机动性。因此，可以考虑采用深度强化学习方法，通过在比赛中不断学习来找到多无人机比赛中的最佳策略。

一些研究工作已经证明了将深度强化学习应用于无人机机动的潜力[86-87]。Ramos 等人[86]通过集成 DDPG 算法使无人机能够在移动平台上执行连续着陆任务。Song 等人[87]通过深度强化学习算法获取单无人机计时竞速赛的时间最优轨迹。但是，目前还没有研究工作将深度强化学习方法应用于多无人机自主竞速问题。现有研究还存在一些亟待解决的问题：1、现有的方法大多是研究单架无人机如何执行任务，无法直接转移到非平稳的多无人机比赛环境中。2、现有的方法没有考虑到真实环境的特点和无人机在现实环境中的运动，而是局限于特定的模拟游戏环境，导致训练结果无法迁移到真实的物理环境中。3、无人机难以理解全部的竞速规则，比赛规则一般都会要求双方选手遵守规范，规则会详细阐述哪些干扰行为是不被允许的，但往往这种规则带有一部分主观性且难以编码的。多机之间的互动需要保证安全性和规范性，而这仅靠机器自身是难以实现的。

为了解决上述问题，本章提出了一种基于人类奖励反馈的深度强化学习驱

动的人机介入控制方法，来解决多机竞速问题。将多无人机竞速问题建模为马尔可夫博弈，同时采用独立近端策略优化 (Independent Proximal Policy Optimization，IPPO) 算法[88]和多智能体近端策略优化 (Multi Agent Proximal Policy Optimization, MAPPO) 算法[89]进行训练以对比两种算法的效果。为了实现从仿真环境到真实场景的迁移，本章将无人机在真实物理系统中飞行的动态响应特性纳入训练环境中。同时，本章设计了人类多机介入机制，以便于在执行过程中，当机器做出不符合竞速规则的行为时，人能够及时介入，此时默认系统中人的决策是绝对正确的。在本章方法中，通过合理塑造人类奖励反馈，可以让人类介入机器的次数变少，使系统的行为更安全且平稳的同时，减轻了人类的负担。同时，为了让无人机学会某些竞速技能，本章将无人机比赛过程中的多个目标建模到整个奖励函数组中，具体包括对冲向终点线、穿越门和超车等行为的奖励，以及对追尾和越过边界等行为的惩罚。

本章主要的贡献点如下：

1、提出了基于深度强化学习的人机介入控制方法，实现了从仿真到现实场景的迁移 (关于现实场景的验证详见下一章)，证明了算法具有较强的泛化性。

2、提出的人介入机器的触发机制保证了系统的安全性和高效性，同时基于人类反馈奖励训练的方法减少了人的介入次数，保障了系统的安全性和决策的高效性。

3、提出的方法不依赖于赛道和赛道边界约束的全局信息，比博弈规划器[90]等规划算法更符合真实无人机比赛场景的要求。

4、提出的方法计算成本较低，能够满足实时竞速的要求，能够适应强动态的环境。

本章的结构安排如下，第 4.2 节给出了竞速场景的建模，主要涉及赛道的描述以及竞速问题的状态和动作空间设计；第 4.3 节进行了任务建模和方法设计，详细阐述了人类反馈奖励的塑造、本章强化学习训练所使用的技巧、人介入机器的机制设计；第 4.4 节进行了相关的仿真试验和结果分析，主要涉及方法的性能对比和关于人类反馈奖励的消融实验；最终在第 4.5 节总结了本章的工作内容。

4.2　场景建模

如图4.1所示，整条赛道由三维空间的多个门组成，没有具体的物理边界。在多机竞速比赛中，第一个依序穿越所有门到达终点线的无人机获得胜利。记 $p_i(t)$ 为无人机 i 在 t 时刻的位置。在比赛过程中，无人机之间需要保持安全距离以避免碰撞。

观测空间：主要由无人机对其他无人机相对位置的观测、对当前门相对位置

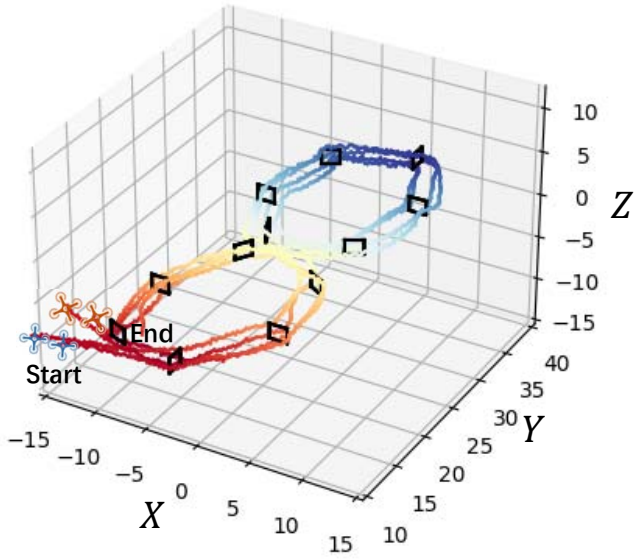


图 4.1　一个 3D 赛道案例

的观测这两部分组成，具体观测空间如图4.2所示。

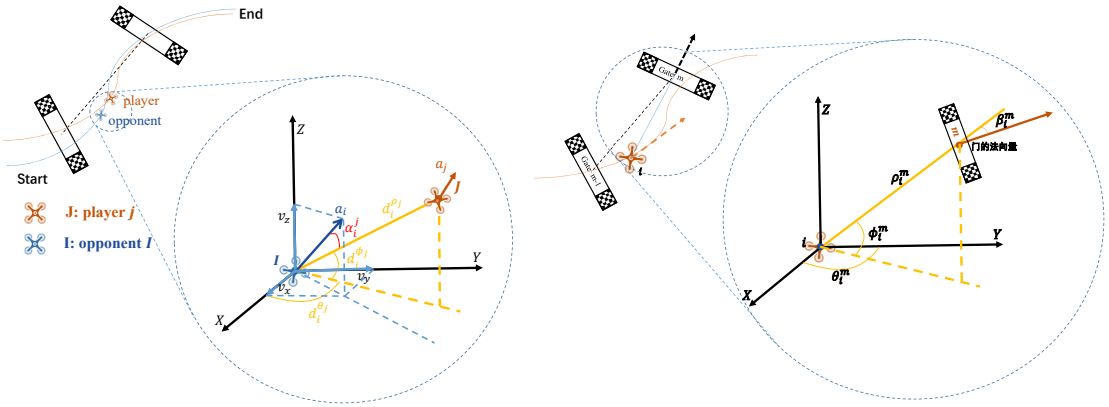
对于前者，本节定义无人机 i 对其他无人机的观测向量为 $o_i^{others} = [o_i^1, \alpha_i^1, \cdots, o_i^{i-1}, \alpha_i^{i-1}, o_i^{i+1}, \alpha_i^{i+1}, \cdots, o_i^n, \alpha_i^n]$，其中 n 表示所有无人机的数量，o_i^j 表示无人机 j 在以无人机 i 的球坐标系中的位置: $o_i^j = (d_i^{\rho_j}, d_i^{\theta_j}, d_i^{\phi_j})$。这里 α_i^j 用于表示从无人机 i 指向无人机 j 的向量与无人机 i 航向之间的夹角。当两架无人机的距离在合适范围内时，α 的变化情况直接反应了两架无人机是将要发生碰撞还是相互远离。

对于后者，本章假设无人机无法获取赛道上所有门的全局信息，在每一时刻无人机只能观测到当前门：$o_i^{gate_m} = [\rho_i^m, \theta_i^m, \phi_i^m, \beta_i^m]$, 其中 $(\rho_i^m, \theta_i^m, \phi_i^m)$ 表示当前门 m 在无人机 i 的球坐标系中的位置。β_i^m 表示门 m 的法向量与从无人机 i 指向门 m 中心的向量之间的夹角，结合球坐标系中的距离 ρ_i^m 可以用于表示无人机 i 相对于门 m 中心的偏离程度。

上述观测中不包含所有门的信息，这种状态设置显然更为合理，可以直接应用到需要考虑无人机视觉的场景中。赛道中门的数量不固定，同时允许无人机在多个具有不同数量门的场景中进行训练，从而提高策略的泛化性。本章所有的位置观测都使用了球坐标系，这是因为在球坐标系下无人机可以直接获得与其他无人机或门的相对距离。

行动空间：无人机在获得上述观测状态后，输出并执行动作命令。动作空间定义为 $a_t = [v_x, v_y, v_z]$，分别表示以无人机自身为中心的坐标系中各个方向上的

速度大小。本章在策略网络的最后一层引入 *tanh* 激活函数后，通过缩放可以将最终动作命令控制在预定义的范围内。



(a) 无人机 i 对无人机 j 的观测　　**(b) 无人机 i 对当前门 m 的观测**

图 4.2　无人机 i 的坐标系：以上观测信息均以无人机 i 自身坐标系为参考。

4.3　基于人类奖励反馈的人机共享控制算法设计

虽然 Song 等人[87]将竞速问题建模为马尔可夫决策过程，但是多机竞速中多台无人机的存在导致状态信息不完全，因此本章将其建模为部分可观测的马尔可夫博弈[91]。

一个涉及 N 个参与人的马尔可夫博弈，由所有可能的状态集合和所有动作组合的集合定义，下一时刻的状态可以根据状态转换函数 $\mathcal{T}: \mathcal{S}\times\mathcal{A}_1\times\cdots\times\mathcal{A}_k \to \mathcal{S}$ 计算获得。其中，$\mathcal{A}_i$ 表示机器 i 的动作空间，$\mathcal{S}$ 表示状态空间。每个机器的观测是通过对当前状态 $o_i: \mathcal{S} \to \mathcal{O}_i$ 的观测获得。其中，$\mathcal{O}_i$ 是机器 i 的观测空间。每个机器 i 根据策略 $\pi_{\theta_i}: \mathcal{O}_i \to [0,1]$ 选择一个动作，然后根据函数 $r_i: \mathcal{S}\times\mathcal{A}_i \to \mathbb{R}$ 获得各自的奖励。初始状态 $\rho: \mathcal{S} \to [0,1]$ 确定后，每个机器 i 的目标是最大化其累计回报，即：

$$R_i = \sum_{t=0}^{T} \gamma^t r_i^t, \quad r_i^t: \mathcal{S}\times\mathcal{A}_1\times\cdots\times\mathcal{A}_k \to \mathcal{R} \tag{4.1}$$

其中，γ 是用于平衡长短期奖励的折扣因子，T 是时域。

显然，将多无人机竞速问题建模为马尔可夫博弈更为合适。因为如果简单地将其建模为马尔可夫决策过程，其中每个机器的观测状态并不包含关于其他机器的任何信息。但是，每个机器的状态转移和奖励函数依赖于其他机器，这导致了环境的非平稳，不利于传统强化学习算法的应用。

4.3.1　奖励塑造

无人机飞完整条赛道所耗费的总时间与任务的主要目标直接相关。但是，无论是成功穿越门，还是完成最终目标，所获得的奖励都是稀疏的，这增加了机器在高维空间中策略搜索的难度。在多机竞速场景中，长期信用分配问题的一个主流的解决方案是将人类的先验知识引入奖励塑造中，使最终的奖励函数最大程度地接近真实目标，同时机器在每个时间步都能收到反馈。

在传统赛车比赛中，基于轨迹中心线过程投影的方法在应用于这类竞速问题时表现出了良好的性能。本节将基于投影的路径过程奖励扩展到无人机竞速问题中，将相邻门中心连接起来的线段作为赛道中心线。这种方法不需要额外的计算就可以获得参考轨迹，轨迹完全由放置在三维空间的一组门来确定，具体形式如图4.3所示。

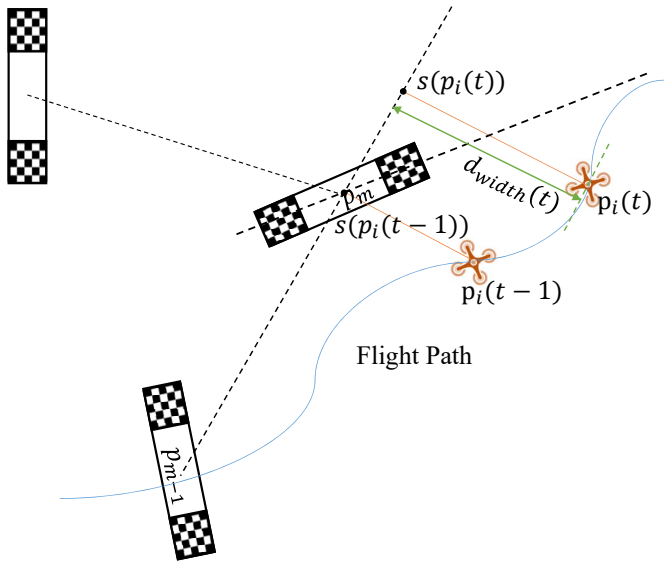


图 4.3　一个飞行轨迹片段

位置奖励：在任何给定的时刻，无人机可以根据它即将通过的下一个门与特定的线段相关联。为了计算一个时间步的路径进度，本节将无人机当前的位置投影到相邻门中心相连的线段上。根据无人机在相邻时间步上的位置，定义具体的位置奖励函数如下:

$$r_{pos}(t) = s(p_i(t)) - s(p_i(t-1)) \tag{4.2}$$

其中，$s(p_i) = (p_i - p_m)\cdot(p_m - p_{m-1})/\|(p_m - p_{m-1})\|$ 定义了无人机在相邻门中心线上的投影点与当前门中心之间的距离。在上面的等式中，p_m 表示门 m 的位置。因此，当 r_{pos} 为正时，无人机正在接近当前门；当 r_{pos} 为负时，无人机正在远离当

前门。

安全裕度奖励: 与之前的工作一样[87]，为了避免无人机穿越门时撞到边框，本节定义了一个安全裕度奖励来惩罚穿越门时过分靠近门框的无人机，以引导无人机从更靠近门中心的位置通过。安全裕度奖励的具体定义如下所示:

$$r_{safe}(t) = -f^2 \cdot (1 - exp(-\frac{0.3 \cdot d_w^2}{g})) + \begin{cases} r_c & if \text{ 与门框相撞} \\ 0 & else \end{cases} \tag{4.3}$$

其中，$f = max[1-(d_p/d_{max}), 0.0]$ 且 $v = max[(1-f)\cdot(w_g/d_{max}), 0.05]$。定义 d_w 为无人机到门的法向量的垂直距离，定义 d_g 为无人机到门平面的距离。无人机与门法向量的垂直距离需要使用该正方形门框的边长 w_g 来进行归一化操作，$d_{\max}$ 表示触发安全裕度奖励的距离阈值，奖励的图像化表示如图4.4所示。

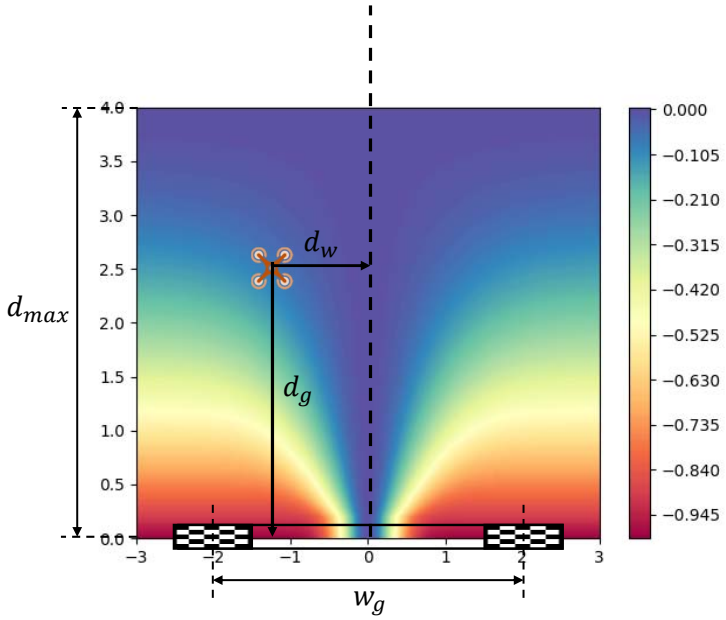


图 4.4　安全裕度奖励的说明

上述安全奖励由两部分组成，前者是用于降低撞击风险的软奖励，r_c 是用于惩罚与门碰撞行为的硬奖励，具体形式如公式4.4所示。其中，d_c 为碰撞点与门中心的欧氏距离，w_g 表示门的宽度。

$$r_c = -\min[(\frac{d_c}{w_g})^2, 10.0] \tag{4.4}$$

越界惩罚：由于本章的赛道没有物理实体边界，仅仅是由多个门组成，当赛道转弯部分足够弯曲时，实验发现机器在训练过程中会收敛到次优路径。如图4.3所示，当无人机沿着当前蓝色轨迹飞行时，任意两个相邻时刻的位置投影的进度为正，即无人机的 r_{pos} 为正，但此时无人机却正在逐渐远离赛道。因此，

本节引入了一个越界惩罚项，当无人机距离当前两个相邻门之间的中心线足够远时，即 $d_{width} > d_{bound}$，惩罚项会被激活。其中 α_d 是温度系数，d_{bound} 是一个常数，可以通过它来控制无人机的探索范围:

$$r_{out} = -e^{(\frac{d_{width}-d_{bound}}{\alpha_d})} + 1 \tag{4.5}$$

超车奖励：为了鼓励无人机学会超车，本节设置了一个正的常数超车奖励 r_o，同时为了避免训练过程中无人机之间出现相互配合刷取奖励的情况，即先超车再减速故意落后再超车，本节设置被超越的无人机将获得相应的负值奖励，奖励的具体形式如下所示：

$$r_{over} = \begin{cases} +r_o & if \text{ 超车} \\ -r_o & if \text{ 被超车} \\ 0 & else \end{cases} \tag{4.6}$$

人类反馈奖励：考虑到多无人机竞速场景中有大量的无人机交互的情况。一方面，确定无人机碰撞的责任分工尤为重要；另一方面，无人机仅靠自己完全学会竞速规则是困难的。虽然无法将竞速规则程序化，但是却可以将人类引入到训练环中，利用人类的先验知识帮助无人机分辨哪种行为是合适的。如果本节在训练中使用后续执行阶段的人介入机制，虽然可以避免无人机发生碰撞，但是却会加重人的负担。因此，人类更适合在训练中处于一个评论员的立场，通过将奖励反馈给无人机，以引导机器做出更规范的操作。传统的人在环上强化学习算法在每个训练回合结束阶段，也会引入人类的评价。但是，与本章所提出的方法相比，这种方法的人类奖励更为稀疏，导致机器需要对过去一段时间的动作轨迹自行进行奖励分配，使机器学习的速度变慢，训练的难度也随之增大。

为了让奖励更为丰富的同时，不增加人类的训练负担，本节设计了如下所示的人类反馈奖励函数：

$$r_{human} = -e^{(-\frac{|d*(|\beta|+a)*b|}{c})} - r_{crash} + r_{feedback} \tag{4.7}$$

其中，c 是决定奖励变化幅度的温度系数，d 表示无人机 i 与最近的无人机 j 之间的距离，对应的 β 表示无人机 i 与无人机 j 的连线与无人机 j 航向的夹角。$d_1 < d_2$，这两者是本节设置的两个表示距离的正常数值，$\beta_1 < \beta_2$，这两者是本节设置的两个表示弧度的正常数值。当 d 和 β 分别都小于指定阈值 d_2 和 β_2 时，系统会请人类进行判断此时是否该激活上述奖励函数；当 d 和 β 分别都小于指定阈值 d_1 和 β_1 时，奖励函数自动激活。$r_{feedback}$ 为人类的稀疏反馈，是一个常数值，与指数函数奖励相结合，用于引导无人机做出符合竞速规则的行为，同时避免追尾等事故的出现。为了避免无人机之间恶意碰撞，有一个基本的常数碰撞

惩罚项 r_{crash}，双方都遭受相同的惩罚。此外，如果所考虑的场景不允许模拟环境中无人机相撞，则可以额外设定最小距离阈值，当无人机之间的距离小于此阈值时，人类介入无人机进行操控。但是，一般而言为了减轻人类的负担，可以使用规则化的避碰策略替代人类进行操控，此时无人机依然会接收到碰撞惩罚项 r_{crash}。

总奖励：最终，每个时间步无人机收到的总奖励定义为:

$$\begin{aligned} r(t) = r_{pos}(t) + \lambda_1 \cdot r_{safe}(t) - \lambda_2 \cdot \|\omega_t\|^2 \\ +\lambda_3 \cdot r_{human}(t) + \lambda_4 \cdot r_{out} + \lambda_5 \cdot r_{over} + r_{cross} \end{aligned} \tag{4.8}$$

在角速度 ω_t 的二次项上加上一个惩罚，可以避免无人机出现大的振荡。r_{cross} 是无人机成功穿越门的奖励，成功通过门会有一个正奖励，错过则会有相应的惩罚。奖励的多个组成部分的作用将会在后面的实验部分进行详细分析。

$$r_{cross} = \begin{cases} r_0 & if \text{ 成功穿越门} \\ -r_1 & if \text{ 漏过门} \\ 0 & else \end{cases} \tag{4.9}$$

4.3.2　策略训练

对于马尔可夫博弈场景，Lowe 等人[92]提出的 MADDPG 算法采用了集中训练和分布式执行的框架，在训练过程中，所有机器的观测和动作都被传递到评论家网络中，此时环境变得平稳。但是，在这种情况下，上述方法的性能并不好。本节考虑了基于近端策略优化算法 (Proximal Policy Optimization，PPO) 的两种算法 (IPPO 算法和 MAPPO 算法) 来训练机器。与其他强化学习算法相比，PPO 算法对奖励塑造不敏感、超参数易于调优且性能非常好，所以在机器人领域广受欢迎。IPPO 算法是在多智能体环境下针对 PPO 算法的直接扩展，而 MAPPO 采用了 MADDPG 集中训练和分布式执行的框架，可以在训练过程中访问所有机器的状态和动作信息。

在多无人机竞速场景下，强化学习算法面临着一个巨大的挑战：高维的策略搜索空间、机器策略变化引起的动作复杂性和环境的非平稳性。此外，深度强化学习算法容易对模拟环境过拟合，这对实现从模拟环境到真实场景的迁移而言是一个巨大的挑战。在这里，本节将强调几个技巧，使得算法能够快速获得稳定的训练结果，并且很好地迁移到现实场景中。

1）PPO 训练技巧：如第二章中所介绍的那样，PPO 是一种求解信赖域优化的方法，如下所示：

$$\begin{aligned} &\max_{\theta} \ \mathbb{E}_t[\frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)} A_t^{\pi_{old}}] \\ &s.t. \ \mathbb{E}_t[KL(\pi_\theta(a_t|s_t)\|\pi_{\theta_{old}}(a_t|s_t))] \leqslant \delta \end{aligned} \tag{4.10}$$

PPO 的主要思想是保证更新后的新策略与旧策略的距离不会太远，并结合单调改进理论，使策略不会随着每一次迭代而变差。与其他信赖域优化算法相比，PPO-Clip 算法可以不计算 Kullback-Leibler 散度，大大提高了计算效率。因此，本节使用 PPO-Clip 算法对目标函数进行训练，具体优化目标如下所示:

$$\mathcal{J}^{CLIP}(\theta)=\mathbb{E}_t[\min(r_t(\theta)A_t,\mathrm{clip}(r_t(\theta),1-\epsilon,1+\epsilon)A_t)] \tag{4.11}$$

其中，$r_t(\theta)=\pi_\theta(a_t|s_t)/\pi_{\theta_{old}}(a_t|s_t)$。在训练过程中，本节使用优势函数归一化的技巧，具体为对一批数据进行广义优势估计后使用 Z-score 方法进行归一化，这种方法可以提高算法的性能。

2）并行训练框架：强化学习算法的采样效率仍然是一个非常关键的问题，算法的训练需要非常大量的数据。Deepmind 使用数百个 GPU 花费连续数月训练得到 AlphaGo[93]，在星际争霸中同时训练数千个机器来获得 AlphaStar[94]，其中每个机器学习所需的数据量相当于人类玩连续数百年的游戏。一般来说，训练更强的机器所需的数据量也是更多的。因此，如何提高机器在模拟环境中的采样效率是问题的关键所在。本章实验中，机器能够同时在 50 个并行环境中执行策略搜索，极大地加快了数据收集的速度。此外，可以使用环境并行化的方法来增加所收集数据的多样性。当赛道生成具有较大的随机性时，可以允许机器同时探索多个不同的赛道。当整条赛道的轨迹较长时，并行化允许机器在每次策略迭代时从多个环境中获取多条赛道的某段数据，而不是仅获取某个特定赛道的数据。这种方法避免了算法对特定赛道的过拟合，极大地提高了强化学习算法的性能。

3）随机初始化：为了避免基于强化学习的机器对模拟环境的过拟合，并增加训练数据的多样性，本节设计了多个赛道地图，每回合随机选择其中一个地图进行初始化，并且增加了门位置的不确定性。每次进行环境初始化后，赛道中的门都有一定程度的变化，这种变化体现在位置和方向 $[\triangle x,\triangle y,\triangle z,\triangle\alpha]$ 上，分别表示门的位置和法向量的变化量。每个无人机的初始位置在初始点固定大小的球形区域内随机生成，避免出现无人机在起飞时总是处于劣势的情况。

如果对手无人机 j 在早期就开始获胜，无人机 i 将因为没有得到积极的奖励而无法学会合适的策略。同时，对手无人机 j 将由于无人机 i 的羸弱，最终训练得到的很有可能是次优策略。通过上述多种随机初始化技巧，可以避免无人机对固定赛道和初始位置的过拟合，增加了策略探索的随机性，提高了模型的泛化性和稳定性。

4.3.3　介入控制

目前，机器在部分决策场景中存在较大的局限性，在很多情况下机器难以学得某些技能或者完成某些任务。在本章所研究的多无人机竞速问题中，虽然在某

些情况下，无人机可能被允许以较近的距离和其他无人机交互，但是在比赛规则中存在某些过近交互是不被允许的，是不符合竞速规则的。这种规则相对而言较为主观，判定规则与上下时序情况高度相关，这就导致很难以一定的方法对这些规则进行编码，以使无人机从中获取相应的信号进而学会理解规则。

在训练过程，一旦无人机之间距离过近就受到处罚，尤其是当无人机与其他无人机相撞时，会受到更大的惩罚。但是，这种软性机制存在一定的问题：一方面，如果训练过程中惩罚设置的非常大，机器虽然在执行过程中会时刻遵守安全准则，但与此同时无人机变得更为保守，一旦无人机落后被前机阻挡，就很容易因为过于保守的动作导致无人机再也难以超越前机。另一方面，如果距离过近的处罚被去除或者变轻，无人机之间很可能做出过于激进的行为，最终引发安全事故。最重要的是，上述的奖励设置不够全面，只是一种对无人机瞬时的奖励判定，而没有依赖上下文信息。显然，仅仅通过奖励塑造，想让无人机理解竞速规则是非常困难的。

考虑到上述机器的缺陷，人类独有的直觉智能可以帮助弥补机器的不足。基于此本节提出了人介入机器的机制，将人类的先验知识引入到系统的决策过程中。在人介入机器的系统中，如图4.5所示，人类并不时刻处于执行端，而是承担“监督员”的身份。当介入机制触发时人类进入决策端，避免机器做出违规行为的同时，引导其继续完成任务。在介入控制中，介入触发条件设置的合理程度会影响系统最终的执行效果，过于频繁的介入会加重人类负担，系统的决策也会丢失一定的平滑性，相邻时刻动作的突变可能会影响系统的安全。如果触发条件变得宽松，人的介入次数变少，但系统在真需要人介入时，人类没能够及时介入将会引发系统性能的下降，甚至可能造成安全事故。

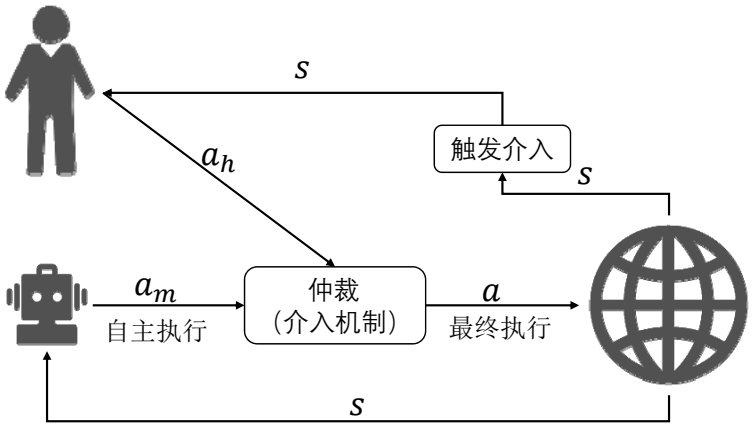


图 4.5　人介入机器框架

本节提出了两级介入机制：当另一架无人机 j 出现在无人机 i 的第一个锥形

区域内时，首先触发介入机制，仅让人类控制当前无人机的偏转，当无人机 j 更进一步出现在无人机 i 的第二个锥形区域内时，仅让人类控制无人机的速度。这种方法的好处在于，无人机竞速过程是强动态的，如果人类介入时需要同时控制多维的动作，对人类操作员而言是比较大的负担。一旦操作者顾此失彼，将会导致无人机失速或者是猛地震荡，此时人类的介入反而使得机器性能下降。多级制的好处是人类每次有足够的时间裕度进行调控，同时，每次控制的动作维度变少，减轻了人类的负担，且无人机动作变化的剧烈程度会有所下降。

4.4　仿真实验

本节使用 Flightmare 模拟器实现了一个可矢量化的类似 OpenAI 设计的多粒子环境 (Multiagent Particle Environmen，MPE) 格式的竞速环境。为了提高采样效率，本节使用了矢量化环境进行训练，允许无人机在多个环境中并行收集数据，并且在微软 Airsim 模拟赛道环境中进行了多无人机自主竞速飞行实验来验证算法的性能。

本节设置了如下几组实验，并根据实验结果回答以下的研究问题:(1) 本章最终选择 IPPO 算法的原因。(2) 塑造的奖励函数 (包含人类反馈奖励) 中多个组件的作用。(3) 本章算法相对于现有算法的优势。

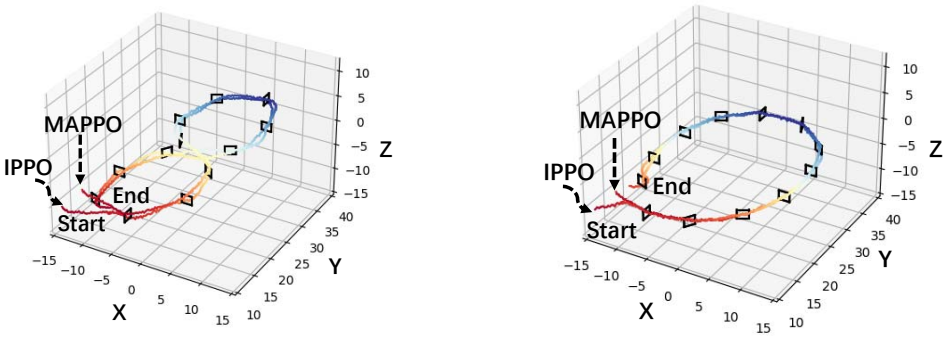
4.4.1　实验设置

为了给多无人机竞速提供丰富的博弈场景，比如无人机进行多次的超车和阻挡，本节在两条具有多个急转弯的赛道 (图4.6) 上测试了算法的性能。在训练过程中，为了提高算法的泛化性，避免过拟合，每一回合的赛道都是由两个固定赛道的几个片段组合而成的。

在实验过程中，半径为 $0.2m$ 的无人机之间的最小安全距离为 $0.5m$，单步决策时间为 0.05 秒，无人机最大飞行速度为 $2m/s$。本节算法中的网络参数基于 Adam 进行更新，学习率为 $2.4e-4$。每次策略迭代前需要从环境中收集 2048 步的数据，使用 8 个回合来优化机器的损失函数，每一批次数据的大小为 256。损失函数中的熵系数为 0.01，折扣因子 $\gamma = 0.99$。

4.4.2　强化学习算法性能比较

本节将 IPPO 算法与 MAPPO 算法进行比较，后者具有与 IPPO 算法相同的超参数。与 Yu 等人[89]提出的 MAPPO 算法不同，此处难以设计特定的机器全局状态。因此，本章 MAPPO 算法直接将所有机器各自的可观测信息连接起来作为全局信息。如前一节所介绍的，MAPPO 算法是多智能体强化学习领域中非常经



(a) 赛道 1　　**(b) 赛道 2**

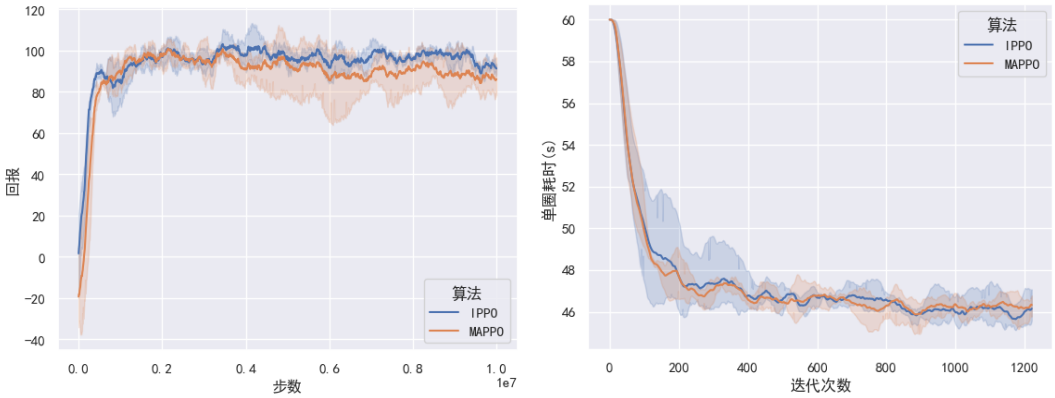
图 4.6　竞速赛道介绍：每条赛道的两条轨迹数据分别来自基于 MAPPO 和 IPPO 方法训练的无人机之间 1 对 1 竞速比赛。

典和先进的算法，它基于 PPO 算法，采用集中式训练和分布式执行的框架，用于克服多无人机竞速场景下环境的非平稳问题。显然，对于阵营对抗的问题，同一阵营的无人机可以更好地合作，MAPPO 算法更具有优势，因为其能够在训练过程中整合所有机器的信息。但是，在两阵营竞速问题中，随着每个阵营无人机数量的增加，由于人机介入机制的存在，介入人员操作水平的不同会给整个阵营的合作带来额外的影响。因此，本章研究的是两种算法在单机作战 (即没有团队) 的竞速场景下的性能差异，这也有利于后续衡量人的反馈奖励对于人机系统整体的性能影响。

每次训练的总步数为 1000 万步，本节一共进行了 10 次不同随机种子初始化下的实验。学习曲线和圈速如图4.7所示。图中的阴影区域代表 10 个不同的随机种子试验下平均评价的标准差的一半。

在图4.7中，为了便于查看，本章对曲线进行了滑动平均处理，滑动窗宽为 13 个时间步长。从 1 对 1 的竞速实验结果中可以发现，这种情况下，MAPPO 算法与 IPPO 算法相比并没有明显的优势，相反 IPPO 算法的平均回报略高于 MAPPO 算法。

为了进一步评估这两种算法，本节将基于 IPPO 算法的无人机与基于 MAPPO 算法的无人机在上述两种赛道上分别进行 1000 次比赛，得到的单圈时间箱线图如图4.8所示。需要强调的是，训练结果中的平均圈速与评估中的平均圈速不同，训练时每一回合的赛道是在两类赛道中随机生成。从比赛结果中可以发现，两种算法在性能上的差异非常小。但是，当参赛无人机的数量增加时，如图4.9所示，四架无人机参加个人竞速赛，其中两架基于 MAPPO 算法，另外两架基于 IPPO 算法，基于 IPPO 算法的无人机在竞速中具有更明显的优势。出现这种结果的原因是：MAPPO 算法在训练时需要将所有无人机的状态-动作对的信息输入到值



(a) 训练曲线　　**(b) 单圈耗时**

图 4.7　IPPO 和 MAPPO 算法在 1 对 1 无人机竞速赛中的训练结果比较

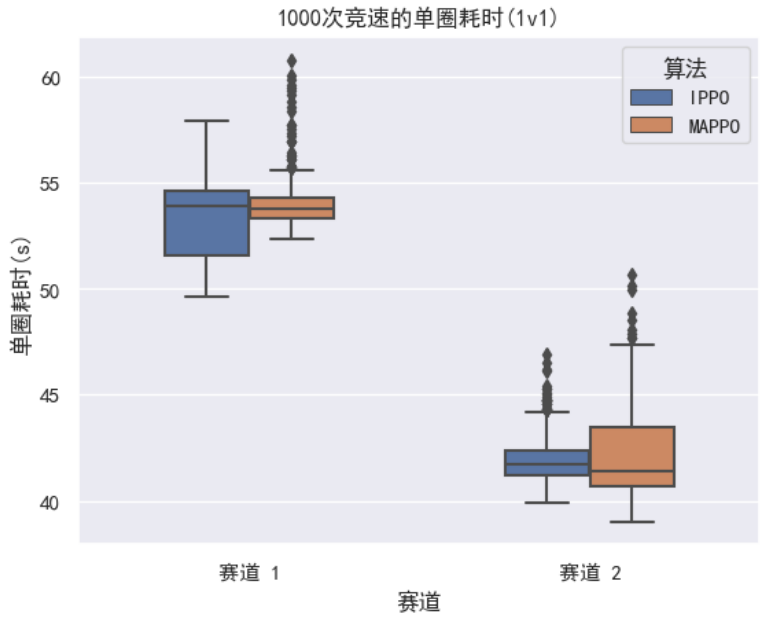


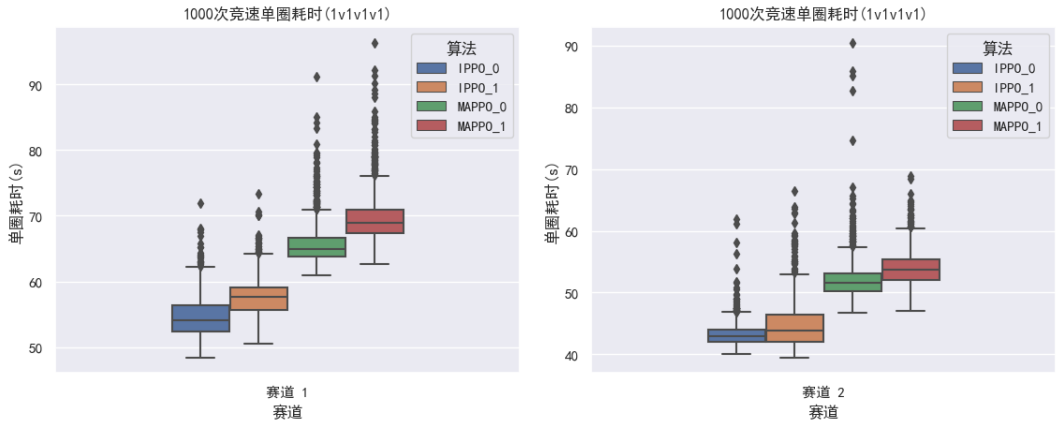
图 4.8　IPPO 和 MAPPO 算法进行 1000 次 1 对 1 竞速结果

函数网络中，无人机数量的增加导致了网络的输入维度增加，值函数的学习变得更为困难。因此，本章选择 IPPO 算法作为后续研究的基准算法。

4.4.3　消融实验

本节进行消融实验来分析所提奖励函数组中多个不同组成部分对算法性能的影响。

安全奖励：本节引入安全奖励，来鼓励无人机穿越门时更靠近中心，避免与门框架发生碰撞。显然，无人机可能为了取胜冒险超车，比如在弯道从更接近门框的位置穿越门，或者以激进的方式保持其领先地位。安全奖励的权重和大小设置会影响无人机决策的激进程度。本节设计了对比实验，以便于清楚地比较无人机在有安全奖励和没有安全奖励这两种情况下的训练表现。



(a) 赛道 1　　**(b) 赛道 2**

图 4.9　4 架无人机进行个人竞速赛的结果

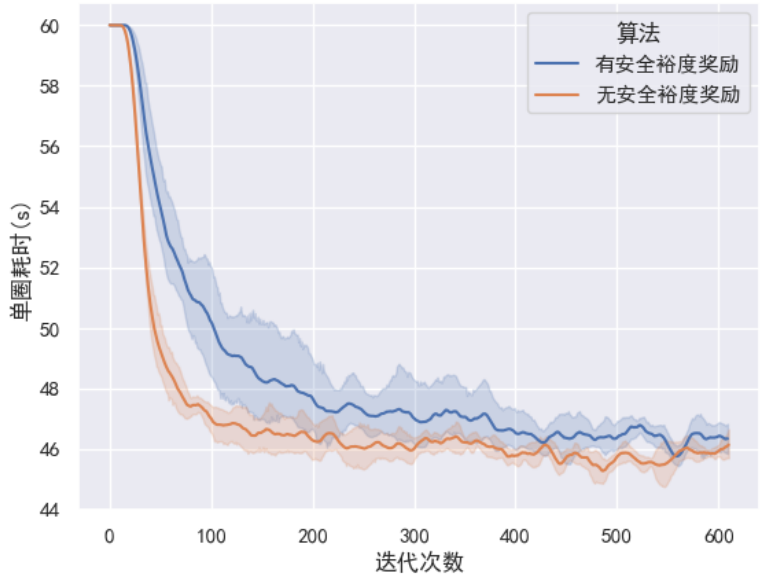
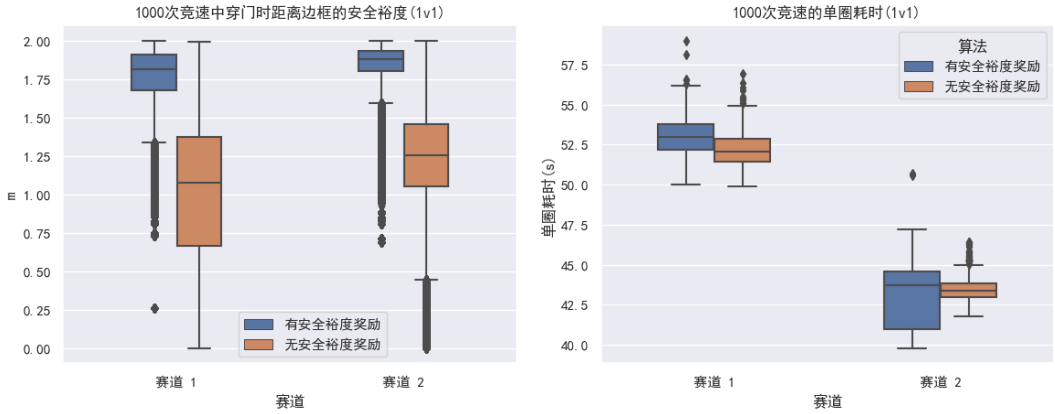


图 4.10　安全裕度奖励消融实验：训练曲线对比图

从图4.10可以发现，在取消安全奖励后，机器在训练过程中的最终平均耗时更短。但是，从图4.11a中可以看出，安全奖励的存在显著提高了无人机穿越门时的安全裕度，降低了无人机撞击门框的概率。虽然从 1 对 1 竞速圈速图4.11中可以发现，安全奖励的存在导致无人机变得更为谨慎，无人机的性能一定程度上有所下降。但是，考虑到无人机需要减少甚至避免碰撞，这类场景中安全至关重要，所以安全奖励是必须引入的。

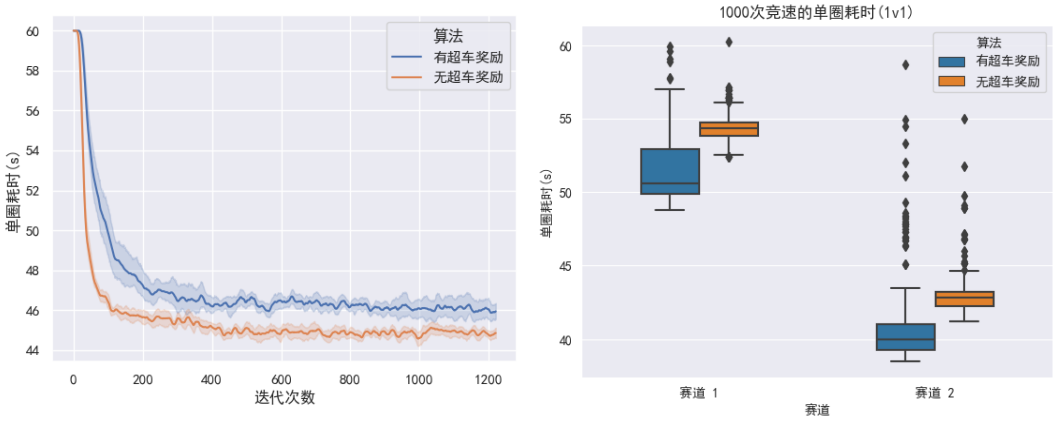
超车奖励：为了使无人机学会超车和阻挡这两个技能，本节设计了超车奖励，通过给予正奖励来鼓励无人机超车，同时设置了负奖励来鼓励无人机学会阻挡。

从图4.12a可以看出，在没有超车奖励的情况下，无人机在训练中表现更好，平均圈速更低。但是，将两架无人机分别放置在两条赛道上进行了 1000 场比赛



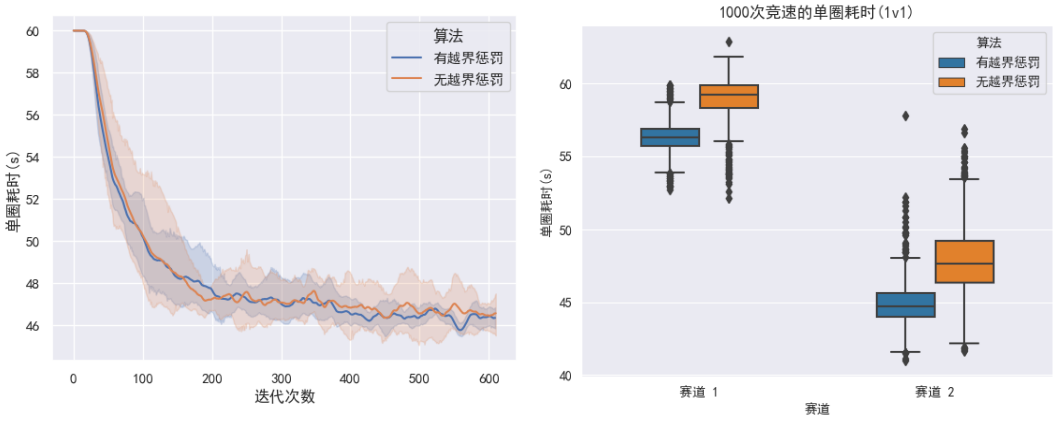
(a) 赛道 1　　**(b) 赛道 2**

图 4.11　安全裕度奖励消融实验：1 对 1 个人竞速赛的结果



(a) 训练曲线　　**(b) 1 对 1 个人竞速**

图 4.12　超车奖励消融实验



(a) 训练曲线　　**(b) 1 对 1 个人竞速**

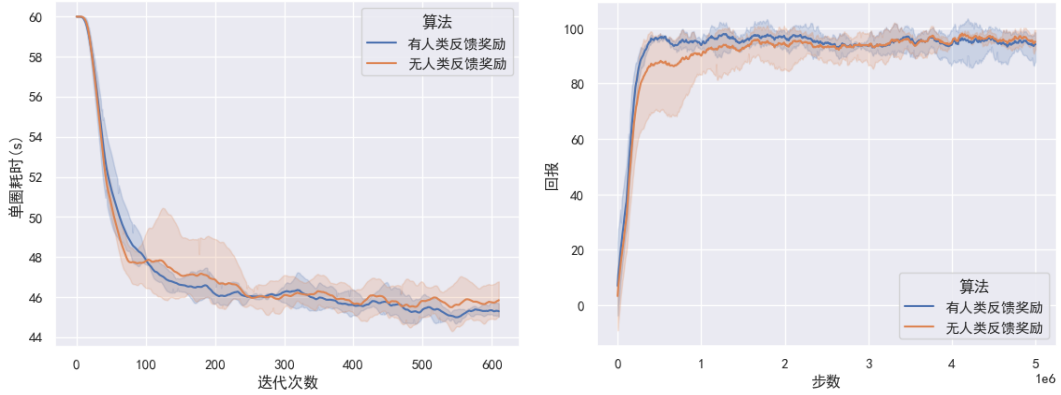
图 4.13　越界惩罚消融实验

后，从图4.12b中可以发现，没有超车奖励的无人机在比赛中表现得更弱。出现上述结果的原因是：在没有超车奖励的情况下，无人机在训练过程中进攻性较弱，最终无人机之间找到了一个和平有序的竞速方式，这导致每个人的平均单圈耗时都比较短，但这种结果与多无人机比赛的初衷相违背。

越界惩罚：由多个门组成的赛道没有特定的物理边界，此时无人机的探测范围过大。因此，本节设计了一个虚拟边界，并引入了越界惩罚以避免一些无效的探测。本节设计了对比试验来评估这项奖励对无人机训练性能的影响。

从圈速图4.13a和比赛结果图4.13b中可以发现，在舍弃越界惩罚后，无人机的学习曲线方差变大，在与基准算法进行 1 对 1 的比赛中处于明显的劣势。

人类反馈奖励：为了评估将人类请入训练环的实际影响，本节同时训练了一批拥有相同超参数的无人机，但是其中一半在训练中不引入人类反馈奖励机制，最终两者的训练对比结果如图4.14所示。其中，阴影部分表示在不同种子初始化下训练的同种机器的性能方差，为了视觉上的方便本节对曲线进行了滑动平均处理。可以发现，训练时没有引入人类反馈的机器单圈耗时的方差更大，系统性能稳定性更差。同时，从回报曲线来看，没有人类反馈的机器训练的收敛速度更慢，且训练前期的方差更大。因此，不难发现人类反馈奖励的引入能够加快机器的学习，同时提高机器策略学习过程中性能的稳定性。



(a) 训练单圈耗时曲线　　**(b) 训练奖励曲线**

图 4.14　人类反馈奖励消融实验：训练结果对比图

为了进一步的比较两种方法的性能，本节让两种无人机分别在两条赛道上进行了 1000 次的 1 对 1 个人竞速赛，结果如图4.15所示，可以发现无论是哪条赛道，训练时有人类奖励反馈的一方所表现的性能更好，单圈耗时更短，同时方差也更小。根据表4.1可知，在引入人类反馈奖励后，无论是何种赛道，人类介入的次数都显著减少。因此，训练时引入人的奖励反馈能够引导机器做出更符合人类规范的行为，并且减轻了执行阶段人类的操作负担。

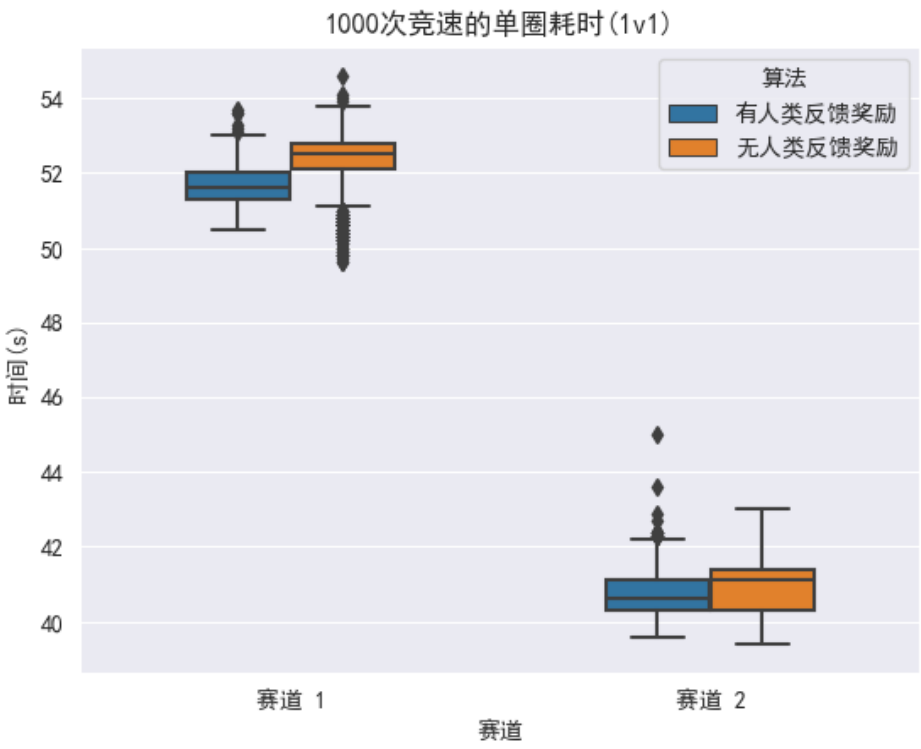


图 4.15　人类反馈奖励消融实验：1 对 1 个人竞速赛评估结果

表 4.1　1 对 1 个人竞速赛人类介入次数 (均值 ± 标准差)

算法	赛道 1	赛道 2
训练时引入人类反馈奖励	4.404 ± 6.805	8.087 ± 8.005
训练时没有人类反馈奖励	20.505 ± 8.797	11.918 ± 9.940

4.4.4　确定赛道下性能比较

本节将所提出的方法与博弈规划算法 (Game Theoretic Planner，GTP) 进行对比，此时赛道是确定的，并且 GTP 算法将会获取所有门的信息。GTP 算法会在轨迹空间中不断计算近似纳什均衡解，由于其所需的算力较大，原论文用了多个 CPU 并行计算，并且使用神经网络来近似预测纳什均衡。最终，两种算法的对比结果见表格4.2。可以发现，GTP 算法不但需要已知多个门的全局信息，而且单步的计算耗时非常大；从平均单圈耗时来看，本章所提算法的性能表现更为优异。

表 4.2　1 对 1 个人竞速赛结果

算法	单步计算耗时 (s)	赛道 1 单圈耗时 (s)	赛道 2 单圈耗时 (s)
本章算法	0.00217 ± 0.000332	51.214 ± 0.785	41.325 ± 0.892
GTP	1.193 ± 0.144	58.769 ± 3.741	45.265 ± 4.684

4.5　本章小结

本章以面向多无人机竞速问题，提出了基于人类反馈奖励的人机介入控制方法。在训练阶段，基于人类反馈奖励的强化学习算法能够引导机器更快速平稳的学会目标策略。在执行阶段，人类的两级介入机制能够避免因为机器的本质缺陷造成较大的损失。最后，本章通过多组对比实验验证所提出算法的有效性，进一步证明了人类奖励反馈不仅能加速机器的训练，还能够使机器的行动更符合规则，从而显著减少了人类的介入次数，减轻了人类的操作负担。

下一章将搭建 Sim2Real 人机实验平台，通过户外实验平台，进一步验证本章所提出的方法在真实物理环境中的性能表现。

第 5 章　Sim2Real 人机实验平台构建与算法验证

前两章进行了算法的设计，并在模拟环境中验证了算法的性能。本章将以多旋翼无人机为对象，搭建具有一定通用性的人机实验平台，便于验证算法从仿真环境迁移到真实物理系统后的性能。在具体的实验部分，本章将以第 4 章的无人机竞速问题为例，验证基于人类反馈的强化学习驱动的人机介入控制算法迁移到真实环境后的性能表现。

5.1　引言

在使用强化学习算法时，如果直接让机器在现实环境中进行探索，就会引发很多问题：训练时间过长造成的成本过高，探索过程中出现的危险动作造成了严重事故。虽然，探索过程中危险行为造成的高昂成本和安全负担这些问题，可以通过使用模拟环境训练机器来避免。但不幸的是，由于现实差距的存在，在大多数情况下，模拟环境中训练的机器无法直接适应现实世界。因此，在将算法从模拟环境迁移到现实场景之前，需要以一定的方式进行处理。

Sim2Real 全称是模拟到现实，属于迁移学习的一种，该领域的目的是使算法从模拟环境训练迁移到真实物理系统后仍有不错的性能表现。Sim2Real 这一领域中主要有四类研究工作：1) 领域适应，通过学习一个映射模型能够将模拟环境和现实环境中相同的状态映射到一个隐空间中，训练中则使用映射后的状态空间进行训练，这样在算法迁移时可以直接应用模型。2) 逆动力学模型，通过学习一个逆动力学模型来修正输出的动作，例如模拟环境中状态 s 下进行动作 a 到达 s'，则现实中得到 (s, s') 需要进行动作 a'，此时机器改进行动作 a' 而不是 a。3) 渐进网络，类似于课程学习，需要一类特殊的网络来让其从简单任务逐渐过渡到复杂的任务中。4) 领域随机化，将模拟环境中的部分参数信息比如图片、物理参数等随机化，比如让机器训练所用模拟环境中的摩擦力、空气阻力随机变化。

针对第 4 章提出的算法，本章在训练时引入了 Sim2Real 方法：一方面，使用了领域随机化的方法，在训练过程中引入了观测噪声；另一方面，构建并在模拟环境引入了无人机近似动态响应模型，模型模拟了真实物理系统下无人机的响应时滞特性、滤波特性。

本章使用的平台由三架搭载着机载树莓派、PX4 固件、电力绝缘子以及无人机和地面端所组成。无人机的驱动程序以机器人操作系统 (Robert Operating System，ROS) 为基础。本章构建的 Sim2Real 平台整体的方案如图5.1所示。首

先，在仿真环境平台算法训练中引入 Sim2Real 方法。然后，在物理模拟器平台进行算法测试。最终，基于中小型机平台搭建实验场景，并用于测试算法性能。上述三级方案，一方面，具有足够的通用性；另一方面，算法在物理模拟器平台上成功部署后，可以预先测试算法效果，同时还能测试无人机所订阅信息的准确性，为后续真实物理环境中的算法测试提供了额外的一轮检测，提高了测试效率和安全性。

图 5.1　实验平台示意图

5.2　软件介绍

平台涉及的 Airsim 和 MPE 这两种环境都在第 4 章提到，两者作为算法训练的模拟环境，本章不再赘述。

ROS 诞生于 2007 年的斯坦福 STAIR 项目，专门用于研究机器人，目前也涉及到人工智能相关的研发。ROS 提供了一套帮助软件开发者编写机器人应用软件的工具箱和程序库，其作为一种机器人框架，封装了多种机器人的硬件比如传感器等，使得上层的控制程序能在 ROS 中直接调用。ROS 最重要的设计目标是提供机器人研发领域的代码的重复利用率，ROS 本身是分布式进程，通过组合不同节点的代码就可以实现任一目标。整个 ROS 系统主要由计算图、文件系统和开源社区三个层次组成。计算图层最深，主要用于描述系统允许的方法，涉及节点、节点管理器、话题、通讯、服务等。文件系统层主要围绕元功能包展开，一个功能包主要分为功能清单包、消息类型、服务类型和代码。开源社区层为人分享资源提供了一个平台，主要是网站、博客之类。

PX4 由苏黎世联邦理工学院的软硬件项目 PIXHAWK 演变而来，是著名的开源自动驾驶软件，可以控制很多不同类型的设备，飞机方面比如多旋翼、固定翼等。PX4 在多旋翼无人机领域是性能非常优异的一个飞行控制器，能够快速平稳地自主控制无人机进行飞行，能够为无人机竞速提供有力的支持。PX4 作为一款大型无人机平台的核心部分，可以与 QGroundControl 等地面站、机载电脑和支持其协议的相机使用 MAVLink 协议进行通信。基于此通信协议，实验中 PX4 能够接收到来自地面站以及旋翼无人机所载电脑的位置、速度、角度、加速度、角加速度等信息，并进行相应的决策和控制。

树莓派是一款微型单板计算器，可基于多种操作系统比如 Linux 和 Windows，几乎能够做到像任何一台基于 Linux 的台式计算机那样运行。CUAV V5+ 是由 PX4 与 CUAV 团队合作制造的先进自动驾驶仪，能够提供数传、传感器等硬件接口。AT9S 遥控器可以远程遥控无人机，其中多旋翼模式中可以设置六种姿态模式以及能够进行可编程混控。

5.3　硬件平台配置

硬件平台主要分为无人机平台、地面端和 ZGET Homer 图数传通信系统。

图 5.2　无人机硬件配置示意图

如图5.2所示，无人机平台主要由 CUAV V5+ 和数传、GPS、数控遥控器、传感器、机载电脑、机架与动力组装而成。飞控系统将从机载电脑或者是从遥控器端接收的指令作为期望，通过扩展卡尔曼滤波器处理 GPS、惯性测量单元、罗盘和气压计等传感器中的信号，获得无人机的速度、姿态、位置等状态信息，将其作为非线性控制器的反馈输入，输出得到电机指令，并以脉冲调制带宽 (Pulse-Width Modulation，PWM) 信号的格式传入电调，由其控制电压、电流信号来驱动电机旋转。最终，桨叶旋转带动无人机进行飞行，以实现对期望的跟踪。整体的硬件连接如图5.3所示，其中飞控与机载电脑 (树莓派 4B) 通过 USB 数据线进行通信。

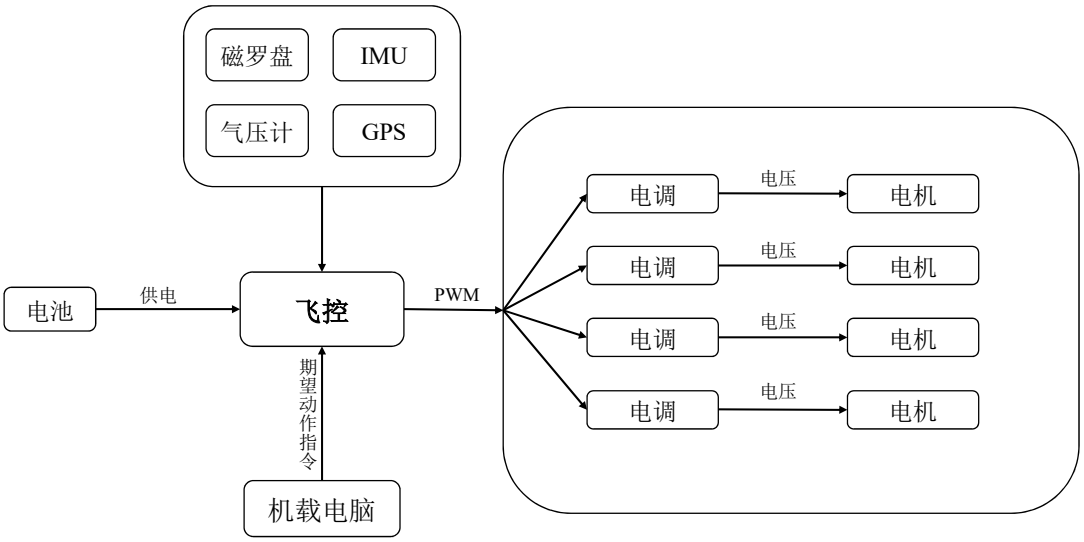


图 5.3　无人机硬件连接示意图

地面端的使用的是戴尔游匣 G15，15.6 英寸显示器，16GB 内存，i7-12700H 处理器，RTX3060 显卡。地面端装有 Ubuntu 18.04 以及相对应的 ROS Melodic 系统，主要作用是通信连接，通过运行 ROS 的节点管理器实现对无人机相关状态信息的订阅。

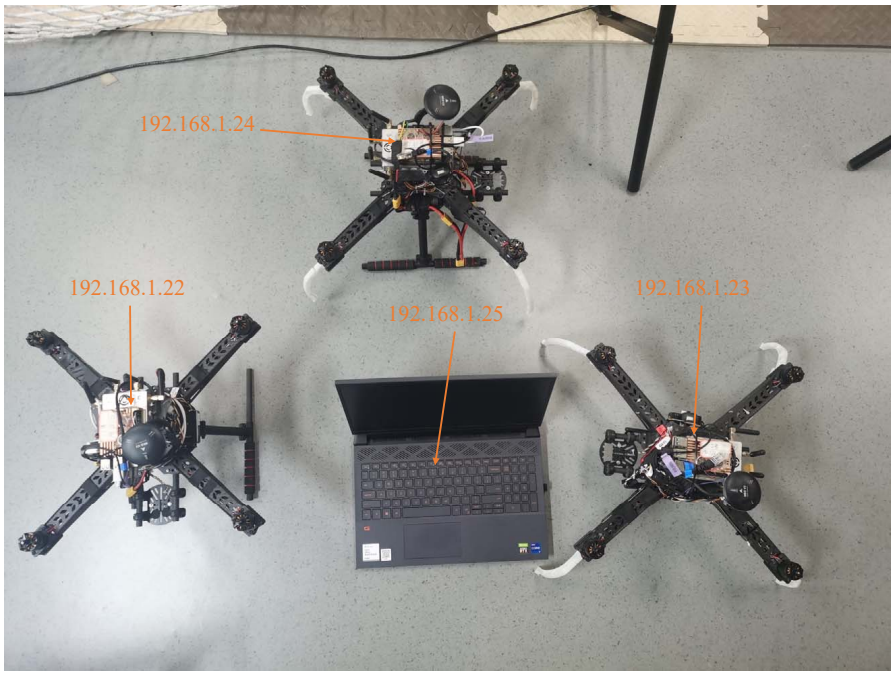


图 5.4　无人机控制平台

本章实验中，ZGET Homer 图数传通信系统由一个地面端和三个移动端组成，其中地面端通过网线与地面端进行连接，移动端通过网线与机载电脑 (树莓派 4B) 进行连接。ZGET Homer 图数传通信系统遵循 IEEE 802.11ac 标准，其数据传输和访问采用的是 TCP/IP 网络协议，工作频段为 5.1GHz~5.9GHz 频段，其具有功耗低、灵敏度高、体积小等优点，能够满足各个领域的无线通信需求。在

本章实验中，ZGET Homer 图数传通信系统基于无线组网通过 IP 访问的方式进行通信。如图5.4所示，地面端和移动端均有各自对应的 IP 地址，与地面端或移动端以网线形式连接的地面端或者机载电脑 (树莓派 4B) 都有对应的 IP 地址。然后，ZGET Homer 图数传以局域网形式进行通信。本章实验中，三个机载电脑 (树莓派 4B) 的 IP 地址分别为 192.168.1.22、192.168.1.23 和 192.168.1.24，对应无人机编号分别为 1，2，3，而地面端对应的 IP 地址为 192.168.1.25。实验中通过 ZGET Homer 图数传通信系统，地面端与机载电脑 (树莓派 4B) 之间能达到的最大传输带宽为 100Mb/s，通信存在一定的延迟，大约为 6ms。通过局域网，不同机载电脑 (树莓派 4B) 的传输带宽最大约为 90Mb/s，通信同样存在一定的延迟，约为 9ms。

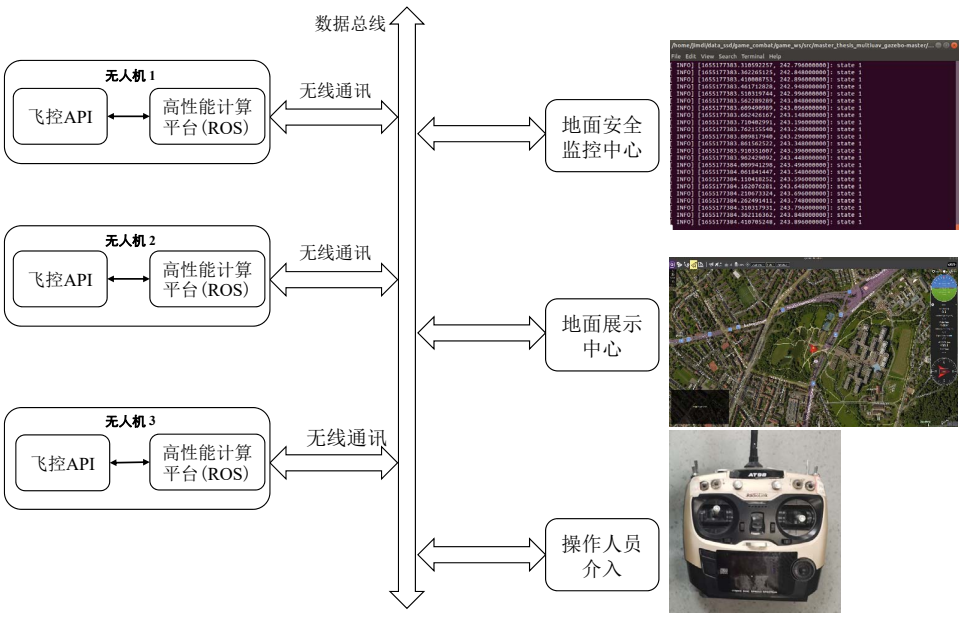


图 5.5　无人机实验平台示意图

硬件实验平台：如图5.5，本章实验一共由三台上述的装有 PX4 固件的无人机、一个地面端、遥控器以及 3 个 ZGET Homer 图数传通信系统组成。实验平台整体的架构冲分利用了 ROS 分布式运行的特点，其中 ROS MASTER(ROS 服务器) 运行在地面端进行监控，而无人机的 ROS 控制节点分别在三架无人机的机载电脑 (树莓派 4B) 上运行，其与地面端分别通过对应的图数传通信系统进行通信。

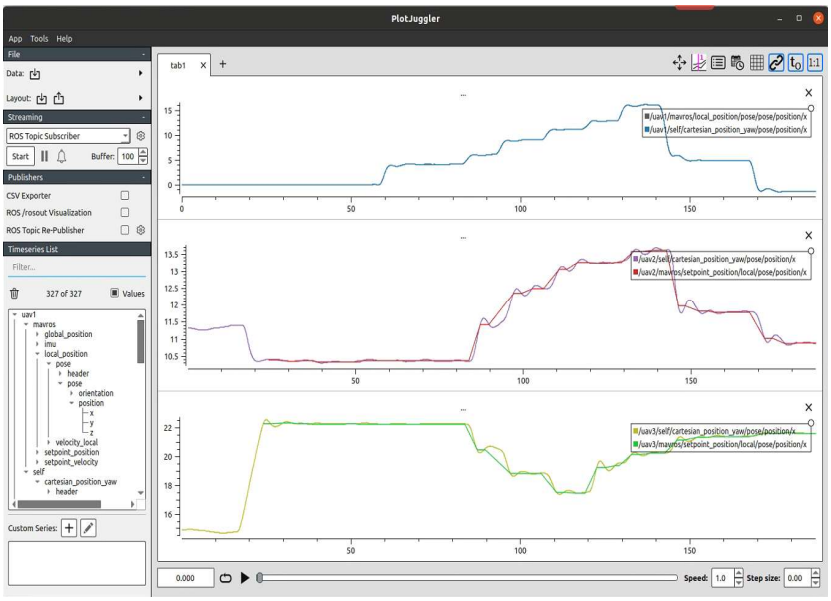
5.4　技术方案设计和实验结果

如图5.1所示，算法部署的流程主要分为训练、物理模拟器平台测试以及真实物理环境测试三部分，其中最为关键的主要是算法的训练和部署到真实物理环境

两个阶段。其中训练阶段能够直接影响算法的泛化性和鲁棒性，考虑到模拟环境和真实场景的差异，一般需要在人工智能算法的训练阶段引入额外的 Sim2Real 技巧；部署到真实环境时，机器需要通过通信系统获取策略函数输入所需的相关信息。考虑到机载算力有限，实验将在地面端进行计算并将策略函数输出的期望动作指令发送至对应无人机，无人机的飞控系统接收到指令后将会执行相应的动作。

接下来本节将会面向第 4 章中的多机竞速场景，验证所提出的基于人类反馈奖励的人机介入控制算法部署到真实物理环境后的性能。

(a) 处理前



(b) 处理后

图 5.6　无人机动作指令和实际动作的数据图

5.4.1　竞速算法 Sim2Real 训练技巧

如图5.6a为无人机在执行决策过程中，三维空间中各个方向的指令轨迹和实际动作数据曲线。其中，更为曲折的为指令曲线，无人机实际的动作曲线更为平滑。对比发现，无人机的实际动作中存在一定的噪声和时滞，并且对于高频变化的动作指令，真实系统中无人机会自动过滤并不执行。对于幅值较小的动作指令，无论如何变化无人机都不予响应。出现上述情况的主要原因是：真实的物理系统中存在着一定的噪声和惯性，无人机的电机并不会在短时间内迅速变化；并且，由于实机的控制精度有限，其无法执行较小幅值的动作。虽然，如果让无人机在现实环境中进行探索，能够让其学会适应环境。但是，这种方法会引发很多问题，比如训练时间过长、探索过程中会出现危险的动作。因此，算法在使用模拟环境训练时，需要从充分考虑现实差距。本节根据与图5.6a类似的真实无人机的飞行数据，在模拟环境中引入了无人机的动态响应模型：一方面，使用阈值函数过滤策略网络输出的低幅值动作指令；另一方面，将处理后的动作指令进行指数平滑处理，以此近似现实世界中的惯性和时滞。如图5.6b，可以发现在进行处理后，无人机的响应特征曲线吻合度更高，但是仍存在一定的时滞，这是因为算法选择的速度分量，对于无人机飞控而言是伪控制量，即将其输入给飞控系统后，需要控制实际底层电机去跟踪参考速度指令，因此时滞是无法完全避免的，这是自动控制原理层面造成的。

同时，无人机实际获取数据中存在一定的噪声。例如，无人机使用 GPS 定位，受到天气等因素的影响，定位不够精确，存在不同程度的误差。因此，本节在实际模拟环境中，针对所有无人机需要通过感知获取的数据，加入了高斯噪声。从神经网络训练的角度考虑，添加一定的噪声有利于提升网络的泛化性。

5.4.2　竞速软件算法的部署实现

如算法5.1所示，本节按照这种形式能够将将不同的无人机策略移植到真实物理环境中。首先，无人机需要手动起飞至一定的高度，无人机接收到初始的位置指令后，将会移动至预定位置等待竞速开始。然后，地面端实时订阅三架无人机的位置信息，分别进行处理以获取无人机 i 对于其他无人机的观测信息 o_i^{others} 和对于当前门 m 位置的观测信息 $o_i^{gate_m}$，拼接成状态向量后传入对应无人机 i 的策略网络，随后得到对应的动作指令。一方面，在计算观测信息时，需要考虑当前无人机 i 是否有发生碰撞的可能或者当前无人机是否违背竞速规则。如果出现上述任何一种情况，对应的操作人员或者预设的避碰规则将会被激活，以保障实验的安全。另一方面，对于策略函数输出的动作指令需要进行一定的平滑，避免因为相邻时刻指令的变化过大，导致无人机的飞行姿态不稳。最终，处理后的动作指令将会通过通信系统传至对应无人机。

算法 5.1　竞速算法实现流程

输入：三架无人机的策略函数 π_1，π_2，π_3；
初始化：赛道中所有门的位置和法向量；
初始化：三架无人机手动起飞至一定高度，并且移动至相应初始位置；
while 任务未结束 **do**
    订阅三架无人机的实时位置；
    for *i=1,2,3* **do**
        计算当前无人机 i 关于所有其他无人机的观测信息 o_i^{others}；
        if 无人机之间距离过近或者违背竞速规则 **then**
            操作人员或者避碰函数进行介入控制；
        end
        计算当前无人机 i 对当前门 m 的观测信息 $o_i^{gate_m}$；
        拼接上述两个观测信息后传入无人机 i 的策略函数 π_i 得到动作指令 a_i；
        对动作指令进行指数平滑处理得到最终的动作指令 a_i；
    end
    分别发送动作指令 a_1，a_2，a_3 至无人机 1，2，3
end

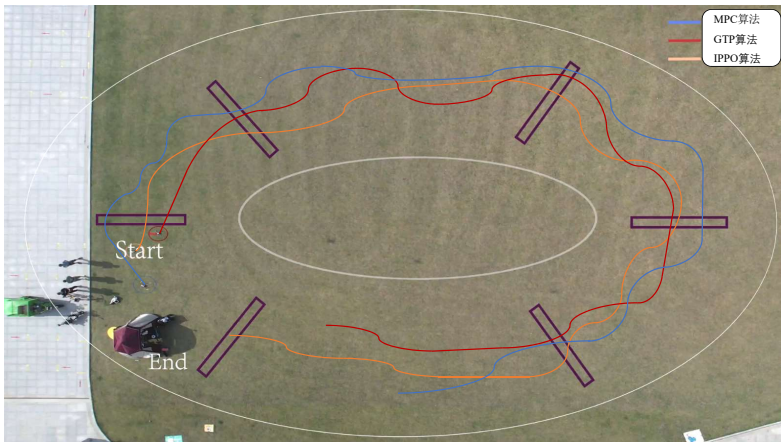


图 5.7　户外实验轨迹图

5.4.3　实验结果

本次户外实验的轨迹图如图5.7所示，由于场地原因实验使用的门为虚拟门(标注在图中)。其中一台为 GTP 算法驱动的无人机，一台为模型预测控制 (Model Predictive Control，MPC) 算法驱动的，还有一台无人机由第 4 章提出的基于 IPPO 的人机介入控制算法驱动。从实验轨迹图中可以发现，IPPO 算法在与 GTP 算法和 MPC 算法的竞速过程中表现出了更为优异的性能。

5.5　本章小结

本章搭建了 Sim2Real 的人机仿真实验平台，并在真实物理场景验证了第 4 章所提算法的性能水平。在第 4 章所提算法的训练中使用了 Sim2Real 方法：一方面，根据实际户外竞速场景引入相应的高斯噪声；另一方面，引入无人机近似动态响应模型模拟真实物理系统中的时滞特性。此外，本章基于三架无人机 (基于 PX4 开源固件) 和 ZGET Homer 图数传通信系统等设备，搭建了户外多无人机竞速实验环境，用于验证第 4 章所提算法迁移到真实物理环境后的有效性和泛化性。最终，实验表明相比于需要知道全局赛道信息的 GTP 算法和 MPC 算法，第 4 章所提算法在户外依然拥有很好的决策效果，并且只需要局部信息 (当前门的信息) 就可以表现得比 GTP 算法和 MPC 算法更优异。

第 6 章　总结与展望

6.1　全文工作总结

本文研究强化学习算法驱动的人机混合智能系统，研究如何将人类智能引入算法的训练端和执行端，以利用人类独有的直觉智能来弥补机器智能本身的局限性，从而提高决策的鲁棒性和安全性，改善了人机混合智能系统的性能表现。本文的主要研究工作总结如下：

(1) 针对深度强化学习算法驱动的人机混合智能系统，提出了一种将人类智能同时引入训练端和决策端的整体框架。该框架将这类人机混合智能系统的序贯决策任务建模成马尔可夫决策过程，并使用强化学习算法进行训练。同时，引入人类的先验知识辅助机器训练，提高学习效率并引导机器掌握相关的能力。最终，在执行端引入人类智能以提升系统整体的决策性能。

(2) 针对强化学习算法驱动的人机共享控制系统，在训练端，引入人类编码的策略限制集合，约束和引导机器探索，极大的提高了采样效率，避免机器做出危险行为，并引导机器找到最优解，从训练端提高了系统的决策性能；在执行端，设计了包含策略限制和策略评估的仲裁机制，舍弃了机器或人类的错误决策，提高系统性能和人类的满意度。

(3) 针对强化学习算法驱动的人机介入控制系统，本文面向多无人机竞速问题，在执行端，选择人类介入控制机制，能够保障系统的决策符合规则且安全。针对任务中算法训练存在的过拟合、机器难以理解部分规则等问题，考虑到该问题中难以将竞速规则编码成策略限制集合，在训练端，使用了软性的人类反馈奖励来引导机器掌握相应能力，同时减少了执行阶段人类的介入次数，降低了人类的操作负担以及系统控制权的争夺次数，保障了系统决策的安全性和稳定性。

(4) 针对以上人机混合智能系统的决策算法，本文以旋翼无人机为背景，搭建了 Sim2Real 人机实验平台，提出了算法部署的通用性框架。最后，验证了第五章提出的算法在迁移到真实物理系统后的性能水平。实验发现在户外多无人机竞速问题中，所提算法相对于 GTP 算法和 MPC 算法，在不需要赛道全局信息的情况下，单圈耗时更短，决策性能更强。

6.2　未来研究展望

本文研究了现有的强化学习算法驱动的人机混合智能系统决策中存在的问题，并从训练端和决策端出发，提出了相应的算法，进行了相关的仿真实验，并搭建了真实物理平台以便进一步验证算法的性能。但是，本文仍存在一些不足，

可以从以下几个方向进行深入研究：

(1) 本文共享控制部分评估人类和机器决策时使用的是价值函数，没有引入其余信息指标 (比如自动驾驶时人类的目光)，因此系统决策能力存在进一步提升的可能。此外，算法决策的准确性极大程度的依赖于目标推理的准确性，因此如果能计算出目标推理网络输出的确定性程度，忽略错误的推理结果，能够进一步提高系统的决策性能。

(2) 本文所提算法无论是在训练中引入人类策略限制还是引入人类反馈奖励，最终算法的性能和训练过程中策略限制、人类反馈奖励的具体形式密切相关。但是，本文并没有对如何设计最优的策略限制和最优的人类奖励反馈进行深入的研究，未来对于研究何种策略限制以及人类反馈能够最大限度地发挥人类先验知识的作用，对于提升系统整体性能有着重大意义。

(3) 本文的仿真实验平台虽然提出了算法部署的通用性框架，但由于实验条件限制仅验证了第 4 章所提算法。未来可以以该框架为基准，进一步扩展算法的验证场景。例如，针对第 3 章的算法，搭建户外无人机降落实验环境进行算法的性能验证。

参 考 文 献

[1] BAINBRIDGE L. Ironies of automation[M]//Analysis, design and evaluation of man–machine systems. Elsevier, 1983: 129-135.

[2] BIBBY K, MARGULIES F, RIJNSDORP J, et al. Man's role in control systems[J]. IFAC Proceedings Volumes, 1975, 8(1): 664-683.

[3] HOC J M. From human–machine interaction to human–machine cooperation[J]. Ergonomics, 2000, 43(7): 833-843.

[4] HOLLNAGEL E, MANCINI G, WOODS D D. Cognitive engineering in complex dynamic worlds[M]. Academic Press Professional, Inc., 1988.

[5] WOODS D D, ROTH E M. Symbolic ai computer simulations as tools for investigating the dynamics of joint cognitive systems[J]. Expertise and Technology: Cognition and Human-Computer Cooperation. Hillsdale: Lawrence Erlbaum, 1995: 75-90.

[6] HOC J M. Some dimensions of a cognitive typology of process control situations[J]. Ergonomics, 1993, 36(11): 1445-1455.

[7] 赵云波, 康宇, 朱进. 人机混合智能系统自主性理论和方法[M]. 北京: 科学出版社, 2021.

[8] ANDERSON S J, PETERS S C, PILUTTI T E, et al. An optimal-control-based framework for trajectory planning, threat assessment, and semi-autonomous control of passenger vehicles in hazard avoidance scenarios[J]. International Journal of Vehicle Autonomous Systems, 2010, 8(2-4): 190-216.

[9] LOSEY D P, MCDONALD C G, BATTAGLIA E, et al. A review of intent detection, arbitration, and communication aspects of shared control for physical human–robot interaction[J]. Applied Mechanics Reviews, 2018, 70(1).

[10] NIKOLAIDIS S, ZHU Y X, HSU D, et al. Human-robot mutual adaptation in shared autonomy[C]//Proceedings of the 2017 ACM/IEEE International Conference on Human-Robot Interaction. 2017: 294-302.

[11] REDDY S, DRAGAN A D, LEVINE S. Shared autonomy via deep reinforcement learning [A]. 2018.

[12] WU X, XIAO L, SUN Y, et al. A survey of human-in-the-loop for machine learning[J]. Future Generation Computer Systems, 2022.

[13] YANG C, ZHU Y, CHEN Y. A review of human–machine cooperation in the robotics domain [J]. IEEE Transactions on Human-Machine Systems, 2021, 52(1): 12-25.

[14] 钱大琳, 刘峰. 人机融合决策智能系统研究的多学科启示[J]. 系统工程理论与实践, 2003, 23(8): 130-135.

[15] KIM D J, HAZLETT-KNUDSEN R, CULVER-GODFREY H, et al. How autonomy impacts performance and satisfaction: Results from a study with spinal cord injured subjects using an assistive robot[J]. IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans, 2011, 42(1): 2-14.

[16] KATYAL K D, JOHANNES M S, KELLIS S, et al. A collaborative bci approach to autonomous control of a prosthetic limb system[C]//2014 IEEE International Conference on Systems, Man, and Cybernetics (SMC). IEEE, 2014: 1479-1482.

[17] MCMULLEN D P, HOTSON G, KATYAL K D, et al. Demonstration of a semi-autonomous hybrid brain–machine interface using human intracranial eeg, eye tracking, and computer vision to control a robotic upper limb prosthetic[J]. IEEE Transactions on Neural Systems and Rehabilitation Engineering, 2013, 22(4): 784-796.

[18] BIRK A, DOERNBACH T, MUELLER C, et al. Dexterous underwater manipulation from onshore locations: Streamlining efficiencies for remotely operated underwater vehicles[J]. IEEE Robotics & Automation Magazine, 2018, 25(4): 24-33.

[19] MARION P, FALLON M, DEITS R, et al. Director: A user interface designed for robot operation with shared autonomy[J]. The DARPA Robotics Challenge Finals: Humanoid Robots To The Rescue, 2018: 237-270.

[20] ABBINK D A, CARLSON T, MULDER M, et al. A topology of shared control systems—finding common ground in diversity[J]. IEEE Transactions on Human-Machine Systems, 2018, 48(5): 509-525.

[21] WANG W, NA X, CAO D, et al. Decision-making in driver-automation shared control: A review and perspectives[J]. IEEE/CAA Journal of Automatica Sinica, 2020, 7(5): 1289-1307.

[22] XU Y, DING C, SHU X, et al. Shared control of a robotic arm using non-invasive brain–computer interface and computer vision guidance[J]. Robotics and Autonomous Systems, 2019, 115: 121-129.

[23] DRAGAN A D, SRINIVASA S S. A policy-blending formalism for shared control[J]. The International Journal of Robotics Research, 2013, 32(7): 790-805.

[24] ZHU Y, YANG C, WEI Q, et al. Human–robot shared control for humanoid manipulator trajectory planning[J]. Industrial Robot: the International Journal of Robotics Research and Application, 2020, 47(3): 395-407.

[25] JAIN S, ARGALL B. Recursive bayesian human intent recognition in shared-control robotics [C]//2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2018: 3905-3912.

[26] MUELLING K, VENKATRAMAN A, VALOIS J S, et al. Autonomy infused teleoperation with application to brain computer interface controlled manipulation[J]. Autonomous Robots,

2017, 41: 1401-1422.

[27] TRAUTMAN P. Assistive planning in complex, dynamic environments: a probabilistic approach[C]//2015 IEEE International Conference on Systems, Man, and Cybernetics. IEEE, 2015: 3072-3078.

[28] ABI-FARRAJ F, OSA T, PETERS N P J, et al. A learning-based shared control architecture for interactive task execution[C]//2017 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2017: 329-335.

[29] ZEESTRATEN M J, HAVOUTIS I, CALINON S. Programming by demonstration for shared control with an application in teleoperation[J]. IEEE Robotics and Automation Letters, 2018, 3(3): 1848-1855.

[30] TANWANI A K, CALINON S. A generative model for intention recognition and manipulation assistance in teleoperation[C]//2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2017: 43-50.

[31] JARRASSÉ N, CHARALAMBOUS T, BURDET E. A framework to describe, analyze and generate interactive motor behaviors[J]. PloS one, 2012, 7(11): e49945.

[32] LI Y, TEE K P, CHAN W L, et al. Continuous role adaptation for human–robot shared control [J]. IEEE Transactions on Robotics, 2015, 31(3): 672-681.

[33] LI Y, TEE K P, YAN R, et al. A framework of human–robot coordination based on game theory and policy iteration[J]. IEEE Transactions on Robotics, 2016, 32(6): 1408-1418.

[34] PELLEGRINELLI S, ADMONI H, JAVDANI S, et al. Human-robot shared workspace collaboration via hindsight optimization[C]//2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2016: 831-838.

[35] NIKOLAIDIS S, ZHU Y X, HSU D, et al. Human-robot mutual adaptation in shared autonomy[C]//Proceedings of the 2017 ACM/IEEE International Conference on Human-Robot Interaction. 2017: 294-302.

[36] AJOUDANI A, TSAGARAKIS N, BICCHI A. Tele-impedance: Teleoperation with impedance regulation using a body–machine interface[J]. The International Journal of Robotics Research, 2012, 31(13): 1642-1656.

[37] RAKITA D, MUTLU B, GLEICHER M. An autonomous dynamic camera method for effective remote teleoperation[C]//Proceedings of the 2018 ACM/IEEE International Conference on Human-Robot Interaction. 2018: 325-333.

[38] ERDEN M S, MARIĆ B. Assisting manual welding with robot[J]. Robotics and Computer-Integrated Manufacturing, 2011, 27(4): 818-828.

[39] CRANDALL J W, GOODRICH M A. Characterizing efficiency of human robot interaction: A case study of shared-control teleoperation[C]//IEEE/RSJ International Conference on Intel-

ligent Robots and Systems: Vol. 2. IEEE, 2002: 1290-1295.

[40] YU W, ALQASEMI R, DUBEY R, et al. Telemanipulation assistance based on motion intention recognition[C]//Proceedings of the 2005 IEEE International Conference on Robotics and Automation. IEEE, 2005: 1121-1126.

[41] RAKITA D, MUTLU B, GLEICHER M, et al. Shared dynamic curves: A shared-control telemanipulation method for motor task training[C]//Proceedings of the 2018 ACM/IEEE International Conference on Human-Robot Interaction. 2018: 23-31.

[42] RAHAL R, ABI-FARRAJ F, GIORDANO P R, et al. Haptic shared-control methods for robotic cutting under nonholonomic constraints[C]//2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2019: 8151-8157.

[43] KANG S B, IKEUCHI K. Toward automatic robot instruction from perception-mapping human grasps to manipulator grasps[J]. IEEE Transactions on Robotics and Automation, 1997, 13(1): 81-95.

[44] KUNIYOSHI Y, INABA M, INOUE H. Learning by watching: Extracting reusable task knowledge from visual observation of human performance[J]. IEEE Transactions on Robotics and Automation, 1994, 10(6): 799-822.

[45] OSA T, PAJARINEN J, NEUMANN G, et al. An algorithmic perspective on imitation learning [J]. Foundations and Trends® in Robotics, 2018, 7(1-2): 1-179.

[46] LOSEY D P, MCDONALD C G, BATTAGLIA E, et al. A review of intent detection, arbitration, and communication aspects of shared control for physical human–robot interaction[J]. Applied Mechanics Reviews, 2018, 70(1).

[47] POMERLEAU D A. Alvinn: An autonomous land vehicle in a neural network[J]. Advances in Neural Information Processing Systems, 1988, 1.

[48] BOJARSKI M, DEL TESTA D, DWORAKOWSKI D, et al. End to end learning for self-driving cars[A]. 2016.

[49] NAIR A, CHEN D, AGRAWAL P, et al. Combining self-supervised learning and imitation for vision-based rope manipulation[C]//2017 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2017: 2146-2153.

[50] RAHMATIZADEH R, ABOLGHASEMI P, BÖLÖNI L. Learning manipulation trajectories using recurrent neural networks[A]. 2016.

[51] GRAVES A, GRAVES A. Long short-term memory[J]. Supervised Sequence Labelling with Recurrent Neural Networks, 2012: 37-45.

[52] DEISENROTH M P, FOX D, RASMUSSEN C E. Gaussian processes for data-efficient learning in robotics and control[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2013, 37(2): 408-423.

[53] ZIEBART B D, MAAS A L, BAGNELL J A, et al. Maximum entropy inverse reinforcement learning.[C]//Aaai: Vol. 8. Chicago, IL, USA, 2008: 1433-1438.

[54] WULFMEIER M, ONDRUSKA P, POSNER I. Maximum entropy deep inverse reinforcement learning[A]. 2015.

[55] FINN C, LEVINE S, ABBEEL P. Guided cost learning: Deep inverse optimal control via policy optimization[C]//International Conference on Machine Learning. PMLR, 2016: 49-58.

[56] AKGUN B, CAKMAK M, JIANG K, et al. Keyframe-based learning from demonstration: Method and evaluation[J]. International Journal of Social Robotics, 2012, 4: 343-355.

[57] AKGUN B, CAKMAK M, YOO J W, et al. Trajectories and keyframes for kinesthetic teaching: A human-robot interaction perspective[C]//Proceedings of the Seventh Annual ACM/IEEE International Conference on Human-Robot Interaction. 2012: 391-398.

[58] HILLELI B, EL-YANIV R. Toward deep reinforcement learning without a simulator: An autonomous steering example[C]//Proceedings of the AAAI Conference on Artificial Intelligence: Vol. 32. 2018.

[59] SAUNDERS W, SASTRY G, STUHLMUELLER A, et al. Trial without error: Towards safe reinforcement learning via human intervention[A]. 2017.

[60] GROLLMAN D H, JENKINS O C. Dogged learning for robots[C]//Proceedings 2007 IEEE International Conference on Robotics and Automation. Ieee, 2007: 2483-2488.

[61] JAVDANI S, ADMONI H, PELLEGRINELLI S, et al. Shared autonomy via hindsight optimization for teleoperation and teaming[J]. The International Journal of Robotics Research, 2018, 37(7): 717-742.

[62] KNOX W B, STONE P. Interactively shaping agents via human reinforcement: The tamer framework[C]//Proceedings of the Fifth International Conference on Knowledge Capture. 2009: 9-16.

[63] MACGLASHAN J, HO M K, LOFTIN R, et al. Interactive learning from policy-dependent human feedback[C]//International Conference on Machine Learning. PMLR, 2017: 2285-2294.

[64] KNOX W B, STONE P. Reinforcement learning from simultaneous human and mdp reward. [C]//AAMAS: Vol. 1004. Valencia, 2012: 475-482.

[65] KNOX W B, STONE P, BREAZEAL C. Training a robot via human feedback: A case study [C]//Social Robotics: 5th International Conference, ICSR 2013, Bristol, UK, October 27-29, 2013, Proceedings 5. Springer, 2013: 460-470.

[66] WARNELL G, WAYTOWICH N, LAWHERN V, et al. Deep tamer: Interactive agent shaping in high-dimensional state spaces[C]//Proceedings of the AAAI Conference on Artificial Intelligence: Vol. 32. 2018.

[67] LEÓN L A, TENORIO A C, MORALES E F. Human interaction for effective reinforcement learning[C]//European Conf. Mach. Learning and Principles and Practice of Knowledge Discovery in Databases (ECMLPKDD 2013): Vol. 3. Citeseer, 2013.

[68] IBARZ B, LEIKE J, POHLEN T, et al. Reward learning from human preferences and demonstrations in atari[J]. Advances in Neural Information Processing Systems, 2018, 31.

[69] 仵博. 动态不确定环境下的智能体序贯决策方法及应用研究[D]. 中南大学, 2013.

[70] SZEPESVÁRI C. Algorithms for reinforcement learning[J]. Synthesis Lectures on Artificial Intelligence and Machine Learning, 2010, 4(1): 1-103.

[71] 章宗长. 部分可观察马氏决策过程的复杂性理论及规划算法研究[D]. 中国科学技术大学, 2012.

[72] 姜哲. 人工智能: 一种现代方法[J]. (No Title), 2004.

[73] 范长杰. 基于马尔可夫决策理论的规划问题的研究[D]. 中国科学技术大学, 2008.

[74] 刘全, 翟建伟, 章宗长, 等. 深度强化学习综述[J]. 计算机学报, 2018, 41(1): 1-27.

[75] HESTER T, VECERIK M, PIETQUIN O, et al. Deep q-learning from demonstrations[C]// Proceedings of the AAAI Conference on Artificial Intelligence: Vol. 32. 2018.

[76] DE LA CRUZ JR G V, DU Y, TAYLOR M E. Jointly pre-training with supervised, autoencoder, and value losses for deep reinforcement learning[A]. 2019: arXiv-1904.

[77] SCHOETTLER G, NAIR A, LUO J, et al. Deep reinforcement learning for industrial insertion tasks with visual inputs and natural rewards[C]//2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2020: 5548-5555.

[78] PENG X B, ABBEEL P, LEVINE S, et al. Deepmimic: Example-guided deep reinforcement learning of physics-based character skills[J]. ACM Transactions on Graphics, 2018, 37(4): 1-14.

[79] KAPLAN R, SAUER C, SOSA A. Beating atari with natural language guided reinforcement learning[A]. 2017.

[80] DANIEL C, VIERING M, METZ J, et al. Active reward learning.[C]//Robotics: Science and Systems: Vol. 98. 2014.

[81] SU P H, GASIC M, MRKSIC N, et al. On-line active reward learning for policy optimisation in spoken dialogue systems[A]. 2016.

[82] DEGRAVE J, FELICI F, BUCHLI J, et al. Magnetic control of tokamak plasmas through deep reinforcement learning[J]. Nature, 2022, 602(7897): 414-419.

[83] HESSEL M, MODAYIL J, VAN HASSELT H, et al. Rainbow: Combining improvements in deep reinforcement learning[C]//Proceedings of the AAAI Conference on Artificial Intelligence: Vol. 32. 2018.

[84] MU T, THEOCHAROUS G, ARBOUR D, et al. Constraint sampling reinforcement learn-

ing: Incorporating expertise for faster learning[C]//Proceedings of the AAAI Conference on Artificial Intelligence: Vol. 36. 2022: 7841-7849.

[85] WANG Z, TAUBNER T, SCHWAGER M. Multi-agent sensitivity enhanced iterative best response: A real-time game theoretic planner for drone racing in 3d environments[J]. Robotics and Autonomous Systems, 2020, 125: 103410.

[86] RODRIGUEZ-RAMOS A, SAMPEDRO C, BAVLE H, et al. A deep reinforcement learning strategy for uav autonomous landing on a moving platform[J]. Journal of Intelligent & Robotic Systems, 2019, 93: 351-366.

[87] SONG Y, STEINWEG M, KAUFMANN E, et al. Autonomous drone racing with deep reinforcement learning[C]//2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2021: 1205-1212.

[88] DE WITT C S, GUPTA T, MAKOVIICHUK D, et al. Is independent learning all you need in the starcraft multi-agent challenge?[A]. 2020.

[89] YU C, VELU A, VINITSKY E, et al. The surprising effectiveness of ppo in cooperative, multi-agent games[A]. 2021.

[90] ATES U. Long-term planning with deep reinforcement learning on autonomous drones[C]// 2020 Innovations in Intelligent Systems and Applications Conference (ASYU). IEEE, 2020: 1-6.

[91] LITTMAN M L. Markov games as a framework for multi-agent reinforcement learning[M]// Machine Learning Proceedings 1994. Elsevier, 1994: 157-163.

[92] LOWE R, WU Y I, TAMAR A, et al. Multi-agent actor-critic for mixed cooperative-competitive environments[J]. Advances in Neural Information Processing Systems, 2017, 30.

[93] SILVER D, HUANG A, MADDISON C J, et al. Mastering the game of go with deep neural networks and tree search[J]. Nature, 2016, 529(7587): 484-489.

[94] VINYALS O, BABUSCHKIN I, CZARNECKI W M, et al. Grandmaster level in starcraft ii using multi-agent reinforcement learning[J]. Nature, 2019, 575(7782): 350-354.

致　谢

提笔至此，全文将尽，回首二十载求学生涯，心潮起伏。从最初的蒙昧无知，到后来的心有所向，不由感慨人生最难的就是明心见性，坚守本心。所幸我虽一路跌跌撞撞，一度心绪难平，但最终也算念念不忘，必有回响。三年研究生生活，收获科研知识的同时，我更确定自己心中所求，在此向所有关心和帮助过我的人们致以诚挚的谢意。

感谢我的导师康宇教授和赵云波教授。康老师无论是在科研工作还是生活方面，都做到了为人师表。康老师科研和工作时认真严谨，生活中待人随和且考虑事情周到，能够设身处地的为学生着想，使我受益良多。感谢赵老师对我科研方面的悉心指导，正是在赵老师的耐心引导下，我形成了科研思维体系和架构，明白了如何确定、研究和解决问题。赵老师有着严谨的逻辑、严格的科研态度以及一丝不苟的工作精神，是我学习的榜样。

感谢我的陈绍冯师姐和邸健师兄。小陈师姐细腻温和，对我十分照顾，给予了我很多的帮助，让我感受到了诸多温暖。邸健师兄认真细致、工作有条不紊且领导有方，在和师兄一起做项目时，师兄教我良多，并且在我之后的科研工作中还给予了很多指导。在师兄和师姐身上，我学会了很多，受益匪浅。感谢我的游诗艺师姐和李婧师姐。于我而言，也许科研最大的意义不是结果本身，而是过程，和诗艺师姐的科研交流让我收获颇丰，那种志同道合的感觉让我感受到科研的另一种快乐。李婧师姐温柔善良，细心体贴且才能兼备，师姐有很多值得我学习的优秀品质。让我非常开心的是能够在茶余饭后和师姐分享自己的喜乐悲欢，感觉枯燥的日子中多了很多的活力。

感谢我所有的朋友们。我常说旅途中最重要的不是风景，而是陪伴身边的人，生活中也是一样。很开心能和你们一起谈天说地，分享着彼此的快乐和悲伤，在你们的陪伴下我的生活显得如此丰富多彩。也许如今我们天南地北，各奔东西，也许我们身处不同的分岔路上，但希望我们在各自的领域熠熠生辉。

感谢我的父母，正是您们对我无私的关爱与支持，我度过了愉快的童年，从一个牙牙学语的孩童逐渐成长为一位顶天立地的男子汉。正是我的身后有你们，我才能放手追逐自己的理想。感谢我所有的亲人们，正是有你们的疼爱，我的精神世界无比强大，被爱和被关心的感觉，让我觉得自己是这世界上最幸福的人。

感谢党和国家，当今世界正处于百年未有之大变局，而我们能在和平安详的环境下健康成长，离不开党和国家的努力，正是这世间有人替我们负重前行，才有这岁月静好。

最后祝愿国泰民安，人民喜乐安康。

在读期间发表的学术论文与取得的研究成果

已发表论文

1. **Ming Li**, Yu Kang, Yun-Bo Zhao, Shiyi You, ”Shared Autonomy Based on Human-in-the-loop Reinforcement Learning with Policy Constraints”, 41st Chinese Control Conference (CCC), Hefei, China, 2022, pp. 7349-7354.

待发表论文

1. Jian Di, **Ming Li**, Yun-Bo Zhao, Yu Kang, ”Autonomous Multi-Player Drone Racing with Deep Reinforcement Learning”, IEEE Robotics and Automation Letters IEEE, under review. (共同一作)