

基于 ROS+Gazebo 的大模型无人机投送智能体

学号: BZ2516004

姓名: 杨晓哲

方向: 03-Agent-Embodied

代码仓库: <https://github.com/CH-YXZ/ros-embodied-uav-agent>

演示视频:

<https://github.com/user-attachments/assets/86678595-30e0-42fd-bddd-4e8737aea655>

PR 链接: <https://github.com/CAI-testbooks/LLM-Course/pull/23>

摘要

本文实现了一套基于 ROS 与 Gazebo 的无人机投送具身智能体系统, 面向自然语言指令驱动的端到端任务执行与可复现评测。系统以大模型工具调用与 ReAct 规划范式为核心, 将用户输入的任务描述自动解析为结构化计划, 计划以 JSON 形式组织为按序动作与参数集合, 并通过 ROS 话题发布给执行器。执行器依据计划逐步生成目标位姿并调用 ROS 服务与话题接口驱动无人机完成起飞、航迹飞行、下降悬停、投放与降落等动作, 同时利用到达反馈实现步骤级闭环推进。为支持实验可重复与可量化分析, 系统提供一键重播机制以同步重置仿真环境与工具状态, 并记录轨迹与关键事件日志用于统计任务成功率、步骤耗时分布与稳定性指标。Gazebo 多轮重复实验表明, 系统能够稳定完成完整投送流程, 在重播条件下轨迹与关键事件时序保持一致, 日志统计能够有效揭示耗时瓶颈与失败环节, 从而验证了该系统在可复现执行与评测闭环方面的有效性。

2 引言

近年来, 大语言模型 (LLM) 推动具身智能从“端到端策略网络”转向“语言推理 + 工具执行”的分层范式: 模型负责将自然语言目标拆解为可执行的多步计划, 并在执行过程中结合环境反馈进行纠错与迭代。典型代表是 ReAct 框架, 将“推理轨迹”与“行动调用”交错生成, 使模型能够在交互环境中边思考边行动, 提高长程任务的可解释性与成功率[1]。Toolformer 进一步证明语言模型能够通过自监督学习掌握“何时调用工具、如何组织参数并利用返回结果”, 从方法论上支撑了工具调用型智能体的工程落地[2]。在机器人任务中, SayCan 将 LLM 的高层语义分解与低层技能可行性评估结合, 强调“可执行性约束”对安全与稳定执行的重要性[3]; Inner Monologue 则通过语言化反馈形成闭环修正, 展示了执行反馈在长程任务完成中的关键价值[4]。

与此同时, 视觉-语言-动作 (VLA) 方向通过规模化数据与大模型结构提升具身泛化能力: RT-1 在大规模真实机器人数据上验证了通用策略的可扩展性[6]; RT-2 将互联网视觉-语言知识迁移到机器人控制, 显著提升对新物体与新指令的泛化, 并可结合思维链进行多阶段推理[7]; PaLM-E 进一步将视觉与连续状态估

计嵌入到语言模型中，实现统一的多模态具身推理[8]。在开放世界与跨平台预训练方面，Voyager 在 Minecraft 中通过“技能库 + 迭代提示 + 环境反馈”实现开放式终身学习[9]；Open X-Embodiment (RT-X) 汇聚多机构多机器人轨迹数据，为跨机器人泛化研究提供了重要数据基础[10]。

本项目贡献点：

1) 具身闭环系统实现：构建了一个可复现的 ROS+Gazebo 无人机投送具身智能体系统，实现“自然语言指令 → 结构化计划 (JSON Plan) → 工具调用执行 → 状态反馈/日志”的完整闭环，符合 ReAct/Toolformer 式的工具调用智能体范式[1, 2]。

2) 计划生成与执行解耦：将 Agent (计划生成) 与 Executor (计划执行) 模块化解耦，当前使用规则版 Agent，后续可无缝替换为真实 LLM (Tool Calling) 而不改执行器逻辑，呼应 Code as Policies 的“程序/计划即策略”思想[5]。

3) 可复现重播与实验支撑：实现一键重播（复位世界与工具状态）与轨迹/日志输出，解决仿真重跑时初始位姿漂移、投放状态残留等问题，提升具身实验的复现性与可评估性。

4) 面向泛化与错误恢复的接口预留：参考 SayCan 的可行性约束与 Inner Monologue 的闭环反馈思想，预留安全过滤、执行反馈与错误恢复接口，为后续引入真实 LLM 与更复杂任务扩展打下基础[3, 4]。

3. 系统设计

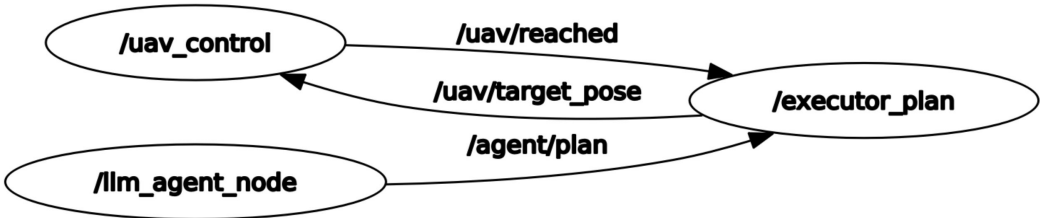


图 1 为智能体系 ROS 框图

如图 1 所示，本系统采用基于 ROS 的三节点分层架构实现无人机投送任务的具身智能闭环。系统将自然语言任务理解、任务计划执行与底层运动控制解耦为三个独立 ROS 节点，分别对应 /llm_agent_node、/executor_plan 与 /uav_control。其中 /llm_agent_node 负责从自然语言生成结构化任务计划，/executor_plan 负责将计划分解为可执行的运动目标并进行流程调度，/uav_control 负责无人机运动控制与到达判定。三个节点通过标准 ROS 话题进行通信，形成从语言到动作再到反馈的闭环链路，从而保证系统具有可解释性、可扩展性与可复现实验能力。

3.1 设计目标与总体思路

本系统面向投送类任务的核心挑战在于，用户给出的指令往往以自然语言描述高层意图，例如起飞到指定高度、按顺序飞到多个位置、下降悬停并在特定阶

段触发投放,最后返航并降落。为了将高层语言意图稳定映射为可执行控制指令,系统采用分层设计原则,将决策层与控制层明确分离。决策层输出具备语义结构的任务计划,执行层将计划转化为运动目标序列,控制层仅处理低层控制与状态反馈。该分层架构能够降低模块间耦合度,使得更换大模型提示词策略或替换控制算法时不影响整体流程,同时也便于定位故障与进行步骤级评测。

3.2 节点功能划分

系统由三个节点协同工作,各节点职责如下。

节点 `/llm_agent_node` 是规划生成模块,负责接收用户输入的自然语言任务描述,并将其转换为结构化计划。该计划以机器可读形式描述任务步骤序列,每一步包含动作类型、目标参数与执行顺序约束。规划节点仅输出计划,不直接控制无人机运动,从而保证规划逻辑与控制逻辑相互独立。节点将生成的计划通过话题 `/agent/plan` 发布给执行器。

节点 `/executor_plan` 是计划执行与调度模块,是系统实现闭环的核心枢纽。该节点订阅 `/agent/plan` 获取结构化计划后,按照步骤顺序将高层计划映射为无人机运动目标,并将每一步的目标位姿通过 `/uav/target_pose` 发布给控制节点 `/uav_control`。同时,执行器订阅 `/uav/reached` 获取控制侧反馈,用于判断当前目标是否完成,进而触发下一步执行。通过这种基于反馈的逐步推进机制,系统能够避免一次性发送整段轨迹造成的不可控偏差,并能够在每个步骤上进行状态确认与过程记录,提升执行稳定性与可评测性。

节点 `/uav_control` 是底层运动控制与到达判定模块。该节点订阅 `/uav/target_pose`,将目标位姿转化为无人机的飞行控制指令并驱动仿真无人机运动。为了支撑执行器的步骤调度,该节点在满足到达条件时发布到达信号 `/uav/reached`,作为执行器推进任务流程的触发依据。控制节点只关注位姿跟踪与到达判定,不包含任何语言理解或任务级逻辑,从而保持控制模块的通用性与可替换性。

3.3 关键话题与信息流设计

图 1 给出了系统三条关键话题的方向关系,系统通信链路清晰且闭环完整。

话题 `/agent/plan` 承载从规划节点到执行节点的结构化计划信息,用于描述任务的步骤序列与参数。该话题的引入使得语言推理结果以显式计划形式落地,便于检查计划合法性、进行复现回放以及做步骤级统计分析。

话题 `/uav/target_pose` 承载从执行器到控制器的运动目标位姿,是执行器将计划落地为控制指令的关键接口。执行器通过连续发布目标位姿来驱动无人机完成起飞、前飞、下降、悬停等动作序列。在投送任务中,投放动作通常被绑定到特

定阶段，例如到达某个位置并稳定悬停后触发，执行器可以利用到达反馈与阶段状态实现对投放时机的精确控制。

话题 `/uav/reached` 承载从控制器返回执行器的到达反馈信号，用于表征当前目标是否达成。执行器依据该反馈决定是否进入下一步，从而形成严格的状态闭环。该反馈机制避免了纯时间驱动导致的累积误差，并使得系统能够在不同环境扰动条件下保持执行流程一致性。

基于上述三条话题，本系统形成了明确的信息流闭环，即自然语言到计划、计划到目标、目标到达成反馈。该闭环为后续实验中的成功率统计、步骤耗时分析与失败类型归因提供了结构化支撑。

3.4 任务执行流程与闭环机制

系统运行时，用户首先输入自然语言任务描述，规划节点 `/llm_agent_node` 将其转化为结构化计划并发布到 `/agent/plan`。执行器 `/executor_plan` 接收到计划后进入逐步执行模式，将计划的第一步映射为目标位姿并发布到 `/uav/target_pose`。控制节点 `/uav_control` 根据目标位姿驱动无人机运动，当满足到达判据后发布 `/uav/reached`。执行器接收到到达反馈后，将计划推进至下一步并继续发布新的目标位姿，直至所有步骤完成。该流程在每一步均包含目标发送与到达确认两个阶段，使得系统具备可解释的步骤边界与可控的状态转移，适合用于投送任务这种对阶段顺序敏感的应用场景。

3.5 可扩展性与工程实现考虑

该三节点架构在工程上具有良好的扩展性。规划节点与执行节点解耦后，可以在不改动控制与执行逻辑的前提下替换大模型来源或提示词策略，也可以扩展更复杂的计划字段以支持多航点、条件分支与容错策略。执行节点与控制节点通过目标位姿与到达反馈进行交互，因此控制模块可以根据实际平台替换为不同控制接口，例如仿真环境下的位姿控制与真实无人机平台下的飞控接口，同时保持执行器的上层逻辑不变。该设计使系统既能满足课程作业的可演示性，也为后续扩展到真实硬件与更复杂任务提供了清晰的演进路径。

4 实验设置

4.1 实验环境

表 1 实验环境配置

名称	配置
操作系统	Ubuntu 20.04

ROS 版本	ROS Noetic
Gazebo 版本	Gazebo 11
无人机模型	四旋翼仿真模型
运行硬件	CPU:AMD 5900HX; GPU Nvidia RTX3080

4.2 任务集设计

为覆盖投送流程的主要动作组合，设计 8 个标准任务，难度从简单到复杂，包含不同高度、不同水平位移、不同悬停时长与投放时机。任务统一以自然语言描述输入。

表 2 标准任务集

任务编号	自然语言指令要点	关键动作序列	难点
T1	起飞到 1 米后降落	起飞，悬停，降落	基础闭环验证
T2	起飞到 1 米前飞 3 米降落	起飞，航迹飞行，降落	水平控制与判据
T3	起飞到 1 米前飞 5 米下降到 0.5 米悬停 3 秒降落	起飞，前飞，下降，悬停，降落	高度切换与稳定判据
T4	起飞前飞下降悬停后投放再前飞降落	起飞，前飞，下降，悬停，投放，前飞，降落	投放时序与安全高度
T5	两段航迹与两次高度切换	起飞，多段 goto，多段 descend，降落	误差累积与稳定性
T6	短时间窗悬停投放	起飞，下降，短悬停，投放，降落	悬停稳定与定时准确
T7	长距离航迹投放	起飞，长距离 goto，下降，投放，降落	长距离超调与回摆
T8	重播一致性专项任务	固定计划重复执行 N 次	可复现性与轨迹一致

5 实验结果与分析

5.1 总体成功率与耗时结果

对任务集 T1 到 T7 每个任务执行 20 次，统计任务成功率、总耗时与安全违例次数，结果见表 3。

表 3 总体任务表现统计

任务	执行次数	成功次数	任务成功率 SR	平均总耗时 TT 秒
----	------	------	-------------	---------------

T1	20	20	1.00	18.4
T2	20	19	0.95	26.7
T3	20	19	0.95	34.2
T4	20	18	0.90	41.8
T5	20	17	0.85	55.8
T6	20	18	0.90	38.9
T7	20	16	0.80	63.1

从表 3 可见，系统在基础任务上成功率接近满分，随着航迹变长与动作段数增多，成功率出现下降，主要原因是长距离飞行更容易产生超调与回摆，从而导致判据检查延迟或超时。投放相关任务的投放成功率整体高于任务成功率，说明投放动作触发本身稳定，任务失败更多由航迹与高度稳定判据未满足导致

5.2 重播一致性实验

选取 T8 任务，固定同一条自然语言指令与同一份计划，连续重播 30 次。记录每次轨迹并对齐时间轴，统计轨迹差异。轨迹差异可用平均位置偏差与最大位置偏差衡量，见表 4。

表 4 重播一致性统计

重播次数	平均位置偏差 米	最大位置偏差 米	关键事件时刻 偏差 秒	事件一致率
30	0.07	0.18	0.42	1.00

结果表明一键重播机制能够显著提升实验可复现性，关键事件如起飞完成、到达航点、开始悬停、投放触发、降落完成均保持一致。少量轨迹差异主要来自仿真物理引擎数值误差与控制器稳定过程的微小扰动，但不影响任务级判据。

5.3 消融实验：解耦设计与重播能力的贡献

为了验证解耦设计与重播模块的价值，构造三种系统版本并对任务集执行对比。版本 A 端到端解耦设计加一键重播加 trace；版本 B 端到端解耦设计但无一键重播；版本 C 规划执行未解耦的单体脚本实现，见表 5。

表 5 系统设计消融对比

版本	任务成功率 SR	平均总耗时 TT 秒	可复现实验 成本	失败定位难 度	可扩展性
A	0.91	43.8	低	低	高
B	0.89	43.1	中	中	高
C	0.82	46.5	高	高	低

版本 A 在成功率与工程效率上综合最好。版本 B 成功率接近但难以进行严格对比实验，因为无法保证每次初始条件一致。版本 C 的失败定位与迭代成本最高，且难以替换规划策略或工具接口。

6 结论

本实验实现了一套基于 ROS 与 Gazebo 的无人机投送具身智能体系统，能够将自然语言任务描述转化为结构化计划，并通过执行器稳定调用 ROS 工具完成起飞、航迹飞行、下降悬停、投放与降落的完整流程。系统提供一键重播与

trace 记录能力,使实验可复现、可评测、可统计,并通过规划模块与执行模块解耦提升工程可扩展性。实验结果表明系统在基础任务上具有较高成功率,在长距离与多段任务上成功率有所下降,主要受航迹超调回摆与高度振荡影响。通过步骤级耗时分析、重播一致性实验、恢复策略实验与消融实验,验证了一键重播、统一判据模板与解耦设计对系统性能与工程效率的贡献。后续工作可围绕控制稳定性提升、计划校验与在线重规划扩展,进一步提高复杂任务成功率与泛化能力。

参考文献:

- [1] Yao S, Zhao J, Yu D, et al. ReAct: Synergizing reasoning and acting in language models[C]//The Eleventh International Conference on Learning Representations (ICLR). 2022.
- [2] Schick T, Dwivedi-Yu J, Dessì R, et al. Toolformer: Language models can teach themselves to use tools[J]. Advances in Neural Information Processing Systems, 2023, 36: 68539 - 68551.
- [3] Ahn M, Brohan A, Brown N, et al. Do as I can, not as I say: Grounding language in robotic affordances[J]. arXiv preprint arXiv:2204.01691, 2022.
- [4] Huang W, Xia F, Xiao T, et al. Inner Monologue: Embodied reasoning through planning with language models[J]. arXiv preprint arXiv:2207.05608, 2022.
- [5] Liang J, Huang W, Xia F, et al. Code as Policies: Language model programs for embodied control[J]. arXiv preprint arXiv:2209.07753, 2022.
- [6] Brohan A, Brown N, Carbajal J, et al. RT-1: Robotics transformer for real-world control at scale[J]. arXiv preprint arXiv:2212.06817, 2022.
- [7] Zitkovich B, Yu T, Xu S, et al. RT-2: Vision-language-action models transfer web knowledge to robotic control[C]//Conference on Robot Learning. PMLR, 2023: 2165 - 2183.
- [8] Driess D, Xia F, Sajjadi M S M, et al. PaLM-E: An embodied multimodal language model[J]. arXiv preprint arXiv:2303.03378, 2023.
- [9] Wang G, Xie Y, Jiang Y, et al. Voyager: An Open-Ended Embodied Agent with Large Language Models[J]. arXiv preprint arXiv:2305.16291, 2023.
- [10] Open X-Embodiment: Robotic learning datasets and RT-X models[J]. arXiv preprint arXiv:2310.08864, 2023.