浙江大学

本科实验报告

姓名：_____

学院：_____

专业：_____

学号：_____

同组学员：_____

指导教师：_____

2026 年 5 月

一、摘要

本项目面向四足机器人在楼梯等复杂地形中的稳定运动问题，围绕“基于外感的步态规划与控制”开展研究与系统实现。项目最终形成了两条技术路线，其中以基于机器人运动学、动力学建模的 OCS2-NMPC 与 WBC 控制方法为主要研究方向，并以强化学习方法作为互补验证路线。第一条路线在模型预测控制框架中引入高程地图、凸平面分割、落足区域约束、摆动足高度规划和全身控制，实现基于外感地形的可解释优化控制；第二条路线基于深度强化学习，在 Isaac Gym 并行仿真环境中训练 Unitree Go2 策略网络，使机器人通过大量试错自主学习上下楼梯和复杂地形行走能力。

在模型预测控制路线中，项目首先在 `ocs2_ros2` 中的 ANYmal 机器人上进行离线楼梯地形验证，通过高程图、平面分割、符号距离场和 RViz 可视化验证感知 MPC 的核心链路；随后在 `quadruped_ros2_control` 中面向 Unitree Go2 搭建 ROS2/Gazebo 仿真系统，引入外部传感器、点云高程建图、离线/在线平面地形发布、感知型 OCS2 控制器和 WBC 全身控制；最后结合 `legged_perceptive` 的感知控制接口，完成基于 MPC 与 WBC 的 Go2 上下楼梯闭环控制。在强化学习路线中，项目基于 `legged_gym` 和 `rsl_rl` 修改训练任务，采用 PPO 算法、235 维观测空间和 12 维关节目标动作空间，通过平地预训练和复杂地形迁移两个阶段，使 Go2 在不依赖精确动力学建模的情况下学习到稳定上下楼梯策略。

实验结果表明，两种方法均能够实现基于外部感知信息的四足机器人上下楼梯。MPC 方法承担了本项目最大的系统实现工作量：从动力学建模、最优控制问题构造、SQP 求解、地形约束接入、摆动足规划，到 WBC 执行和 Gazebo 硬件接口适配，形成了完整的可解释控制闭环。该方法结构清晰、约束可解释、便于分析和安全边界明确，但对模型、约束、地形表达和求解器稳定性要求较高，工程实现复杂。强化学习方法实现路径更直接，不需要手工推导复杂动力学约束，只需在高并行仿真平台上不断训练即可获得较强地形适应能力，但其可解释性、稳定性证明、泛化边界和真实部署安全性相对较弱。两条路线的对比与融合，是项目后续继续推进的重点方向。

关键词：四足机器人；外部感知；模型预测控制；强化学习；OCS2；落足点规划；高程地图；全身控制；Isaac Gym；Unitree Go2

二、引言

四足机器人相比轮式和履带式机器人具有更强的地形通过能力，但在楼梯和台阶环境中，机器人必须同时解决感知、落足、姿态、力分配和动态平衡问题。平地运动中常用的周期步态和固定高度参考，在楼梯场景下容易出现足端撞击台阶、机体高度不足、支撑点落在边缘、MPC 约束不收敛或控制器输出抖动等问题。因此，本项目的核心目标不是简单让机器人“走起来”，而是研究如何将外部传感器得到的地形信息真正接入运动规划与控制过程，使机器人能够根据楼梯高度、支撑平面和局部地形变化主动调整机体和足端运动。

围绕这一目标，项目没有只选择单一技术路线，而是并行研究了两类当前四足机器人领域具有代表性的方案。基于模型的方法以 OCS2-MPC 为核心，它将机器人动力学、接触模式、摩擦约束、落足区域和地形高度显式写入优化问题，强调可解释性和约束安全性。基于强化学习的方法则使用无模型策略学习，机器人在大规模仿真环境中通过奖励函数和试错逐渐形成上下楼梯行为，强调实现效率、策略反应速度和复杂地形适应性。

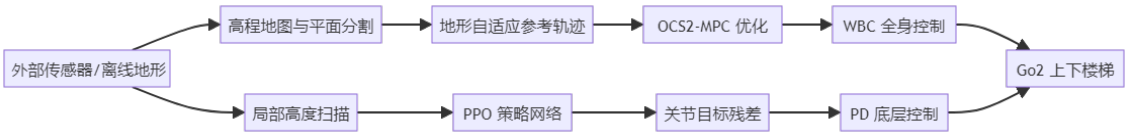
这两类方法代表了四足机器人控制中的两种典型思路。前者更像“把问题建模清楚后求解”，后者更像“在足够多的环境中让策略自己学会”。本项目的研究工作也围绕二者展开：先用 MPC 体系明确感知地形、落足约束和全身控制之间的关系，再用强化学习体系探索无需

复杂建模时能否通过训练获得可用策略，最后对两类方法进行对比并讨论融合方向。

三、系统路线与总体框架

本项目最终形成了“模型预测控制路线”和“强化学习路线”两套系统。二者都使用外部感知信息，但信息进入控制器的方式不同。MPC 路线将地形显式表示为高程图、平面区域和符号距离场，并在优化过程中使用这些地形结构；强化学习路线则将局部高度扫描作为策略网络观测的一部分，由神经网络自行学习地形与动作之间的映射关系。

两条路线的整体关系如图所示。



从工程上看，MPC 路线主要由 `ocs2_ros2`、`quadruped_ros2_control` 和 `legged_perceptive` 三部分构成。`ocs2_ros2` 提供底层最优控制框架、SQP 求解器和 ANYmal 感知验证示例；`quadruped_ros2_control` 负责 Go2 在 ROS2/Gazebo 中的控制器、硬件接口、点云建图和仿真环境；`legged_perceptive` 提供感知型腿足控制接口，包括平面地形接收、落足区域选择、足端碰撞约束和可视化。强化学习路线则基于 Isaac Gym、`legged_gym` 和 `rsl_rl`，围绕 Go2 的观测空间、动作空间、奖励函数和课程学习策略进行修改。

两条路线的方法对比如表 1 所示。

表 1: 两条技术路线的方法对比

维度	**基于模型的 MPC 方法**	**基于强化学习的方法**
核心思想	显式建立动力学、接触、地形和约束模型，并在线求解最优控制问题	通过大规模仿真试错学习状态到动作的策略映射
感知信息形式	高程地图、凸平面、符号距离场、可落足区域	局部地形高度扫描，作为神经网络输入
控制输出	NMPC 输出期望状态、接触力和关节速度，WBC 转换为力矩与低层关节命令	策略网络输出 12 个关节目标角度残差
优点	可解释、可加入明确安全约束、便于定位问题，适合作为主要控制框架	不需要复杂建模，训练成功后推理快，适合复杂非线性行为
缺点	建模和工程实现复杂，对地形质量和求解器稳定性敏感	可解释性差，训练依赖奖励设计，泛化和安全边界较难证明
本项目作用	主要研究方向，构建可解释的外感上下楼梯 NMPC-WBC 控制链路	互补研究方向，构建较简单直接的外感上下楼梯学习控制链路

四、基于模型预测控制的外感上下楼梯方法

4.1 方法原理

模型预测控制路线的基本思路是将楼梯地形转化为优化控制器可以理解的结构化信息。机器人并不是直接根据原始点云控制关节，而是先将点云或离线地形转为高程地图，再从高程地图中提取凸平面区域。平面区域用于判断足端是否能够落脚，地形高度用于修正机体参考高度和摆动足轨迹，符号距离场用于表示足端或腿部与地形之间的碰撞关系。

在 OCS2 框架中，机器人状态、输入、接触时序、动力学模型和约束共同组成一个有限时域非线性最优控制问题。每次控制循环中，控制输入或目标点先被转换为机体状态参考轨迹，再由 NMPC 根据当前状态和未来一段时间的目标轨迹求解最优状态与输入序列。随后，WBC 全身控制器根据 NMPC 输出的期望质心运动、接触力和足端运动，计算关节力矩、位置、速度以及 PD 增益命令，最终驱动 Go2 在仿真中运动。也就是说，NMPC 负责 “未来一段时间应该怎样走”，WBC 负责 “当前这一时刻怎样用全身动力学把这个计划执行出来”。

该方法的关键是地形信息不是以经验规则零散使用，而是进入 reference manager、precomputation、constraint 和 swing trajectory planner 等模块，成为优化问题的一部分。

MPC 控制链路如图所示。

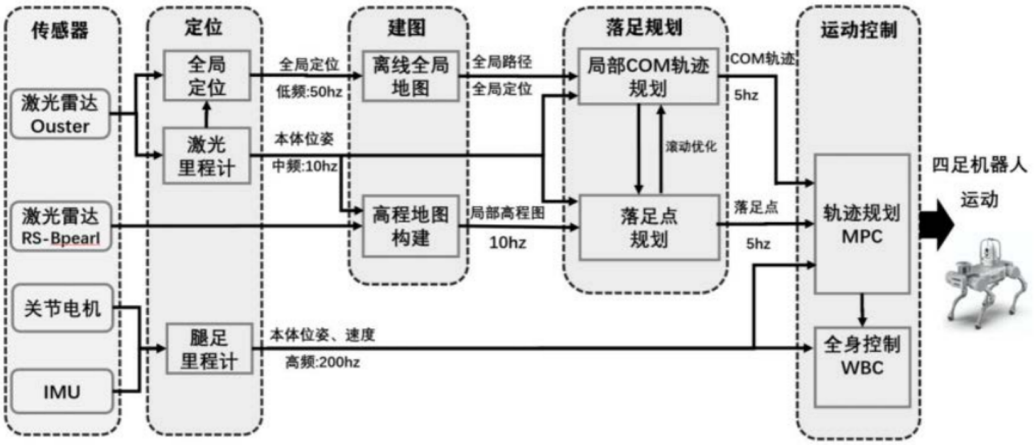


图 1: 四足机器人控制流程

4.2 NMPC 最优控制建模

本项目采用的 NMPC 方法可以写成一个连续时间有限时域最优控制问题。设机器人状态为 $\mathbf{x}(t)$，输入为 $\mathbf{u}(t)$，初始时刻为 t_0，预测终止时刻为 t_I，则 NMPC 在每次循环中求解：

$$
\begin{aligned}
\min_{\mathbf{u}(\cdot)} \quad & \phi(\mathbf{x}(t_I)) + \int_{t_0}^{t_I} l(\mathbf{x}(t), \mathbf{u}(t), t)\, dt \\
\text{s.t.} \quad & \mathbf{x}(t_0) = \mathbf{x}_0, \\
& \dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), t), \\
& \mathbf{g}_1(\mathbf{x}(t), \mathbf{u}(t), t) = 0, \\
& \mathbf{g}_2(\mathbf{x}(t), t) = 0, \\
& \mathbf{h}(\mathbf{x}(t), \mathbf{u}(t), t) \geq 0.
\end{aligned}
\tag{1}
$$

其中，ϕ 是终端代价，l 是运行代价，$\mathbf{f}$ 表示系统动力学，$\mathbf{g}_1$ 和 $\mathbf{g}_2$ 分别表示状态-输入等式约束和纯状态等式约束，$\mathbf{h}$ 表示不等式约束。对于四足机器人，这些约束并不是抽象数学项，而是具体对应到接触、摩擦、足端运动和地形安全条件。

在实现中，系统状态和输入采用质心动力学形式。状态包含归一化质心动量、机体位姿和关节位置；输入包含四个足端的接触力和关节速度。可以概括为：

$$\mathbf{x} = [\mathbf{h}_{\text{com}}^T, \mathbf{q}_b^T, \mathbf{q}_j^T]^T, \qquad \mathbf{u} = [\mathbf{f}_c^T, \mathbf{v}_j^T]^T. \tag{2}$$

这里 $\mathbf{h}_{\text{com}} \in \mathbb{R}^6$ 表示归一化质心线动量和角动量，$\mathbf{q}_b$ 是机体位置与姿态，$\mathbf{q}_j$ 是关节位置，$\mathbf{f}_c \in \mathbb{R}^{12}$ 是四个足端三维接触力，$\mathbf{v}_j$ 是关节速度。代价函数主要采用二次跟踪形式，惩罚当前状态、输入与目标轨迹之间的误差。对于上下楼梯任务，目标轨迹不再是固定高度平地轨迹，而是由地形高度、台阶方向和局部支撑面共同修正后的机体轨迹。

NMPC 中的主要约束总结如表 2 所示。

表 2: NMPC 主要约束及其在上下楼梯中的作用

约束类型	**数学意义**	**在上下楼梯中的作用**
质心动力学约束	保证质心动量变化与接触力一致	使机器人上台阶时的重力做功和支撑力分配符合动力学
摩擦锥约束	限制足端接触力不能超过摩擦极限	避免楼梯边缘或坡面支撑时出现打滑
支撑足零速度约束	支撑相足端相对地面不滑动	保证踩在台阶平面上的足端稳定支撑
摆动足高度约束	摆动足 z 方向跟随 gait 生成曲线	使足端能够跨过台阶边缘并在目标高度落足
落足区域约束	足端位置落在凸平面可支撑区域内	避免足端落在台阶边缘、空洞或不可支撑区域
碰撞距离约束	足端或腿部与地形保持安全距离	减少摆动腿踢到台阶立面或非平面障碍

求解方面，本项目沿用 OCS2 的多重射击离散化方式，将连续时间最优控制问题转写为非线性规划问题，再通过序列二次规划求解。每次 SQP 迭代会在线性化动力学和约束，并构造一个二次规划子问题。QP 子问题由高性能求解器求解，得到状态和输入的增量，再通过线搜索和约束评估更新轨迹。由于控制器需要在机器人运动中不断滚动运行，本项目还对 warm start、时间窗过期和轨迹扩展失败等情况做了保护，使 NMPC 能够在楼梯地形和步态切换下持续工作。

4.3 WBC 全身控制建模

NMPC 规划的是有限时域内的最优状态和输入，但它并不直接等价于电机命令。四足机器人真实执行时，必须同时满足全身刚体动力学、接触力、关节力矩限制、摆动足任务和机体运动任务。WBC 的作用就是在当前时刻根据 NMPC 给出的参考，求解一个全身动力学一致的关节命令。

本项目中的 WBC 采用 QP 形式，决策变量可以表示为：

$$\mathbf{x}_{\text{wbc}} = [\ddot{\mathbf{q}}^T, \mathbf{f}_c^T, \boldsymbol{\tau}^T]^T, \tag{3}$$

其中 $\ddot{\mathbf{q}}$ 是广义坐标加速度，$\mathbf{f}_c$ 是接触力，$\boldsymbol{\tau}$ 是关节力矩。WBC 在当前时刻求解这些变量，使其满足全身刚体动力学、接触一致性、摩擦和力矩等约束，同时尽量跟踪 NMPC 给出的机体和足端参考。

与 NMPC 的长时域规划不同，WBC 只关注当前控制周期，因此它更适合处理执行层的瞬时约束和任务优先级。例如支撑足不能移动、摆动足要跟踪期望轨迹、机体姿态要保持稳定、关节力矩不能超过限制，这些任务在 WBC 中可以被组织为等式或不等式约束。项目中的 `WeightedWbc` 和相关 QP 模块承担了这一功能，并最终输出 12 个关节力矩。

为了提高 Gazebo 和真实电机接口上的跟踪稳定性，执行层没有只发送力矩，而是将力矩作为前馈项，同时发送低增益的关节位置和速度 PD 命令。这种 “力矩前馈 + 低增益 PD” 的形式可以在足端接触台阶时降低冲击，并改善关节轨迹跟踪。项目在 Gazebo 硬件接口中扩展 `kp`、`kd` 命令通道，就是为了让 WBC 输出能够更接近真实腿足控制栈的执行方式。

NMPC 与 WBC 的分工对比如表 3 所示。

表 3: NMPC 与 WBC 的分工对比

模块	时间尺度	输入	输出	本项目中的作用
NMPC	未来一段预测时域	当前状态、目标轨迹、地形、步态	优化状态、输入、接触力和模式序列	规划如何上台阶、如何分配接触力、如何满足地形约束
WBC	当前控制周期	NMPC 输出、测量状态、刚体动力学模型	关节力矩、关节位置速度参考	将优化轨迹转化为可执行电机命令
低层 PD/硬件接口	电机控制周期	关节目标、速度、力矩前馈	电机实际输出	缓冲冲击、改善跟踪、稳定仿真或硬件执行

4.4 ANYmal 离线验证平台

项目首先在 `ocs2_ros2/advance examples/ocs2_anymal_loopshaping_mpc` 中改造 ANYmal 感知 MPC。ANYmal 已经具备较完整的模型和可视化基础，因此适合作为算法验证平台。项目新增了 `stair.png`、`complex_stair.png` 等离线楼梯高程图，并在 `PerceptiveMpcDemo.cpp` 中保留原有地形感知、平面分割、符号距离场和轨迹可视化功能，同时将固定脚本式演示改为键盘驱动的滚动 MPC。

在该示例中，高程图通过 `convex_plane_decomposition::loadGridmapFromImage()` 加载，并经过预处理、滑动窗口平面提取、RANSAC 细化、轮廓提取和后处理生成 `PlanarTerrain`。随后，平面地形被包装为 `SegmentedPlanesTerrainModel` 注册到参考管理器中。MPC 在滚动运行时，`generateRollingReference()` 每隔固定周期或在键盘输入变化时生成 2 秒前瞻参考轨迹，机体高度和姿态会根据地形变化调整。

为了支持长时间交互运行，项目还对 gait schedule 做了延展处理，避免机器人走过若干周期后 MPC 失去未来接触时序。可视化部分持续发布高程图、滤波地图、平面边界、内缩区域和符号距离场，使算法调试不再依赖日志，而是能够在 RViz 中直接观察落足区域和优化轨迹。

在 rviz 中对其进行测试，通过附件中的视频可以看到，我们通过键盘控制 ANYmal 机器人可以实现稳定的上下楼梯运动，MPC 预测出的可行路径点也会在 rviz 中可视化表示。证明了我们算法和思路的可行性与有效性。

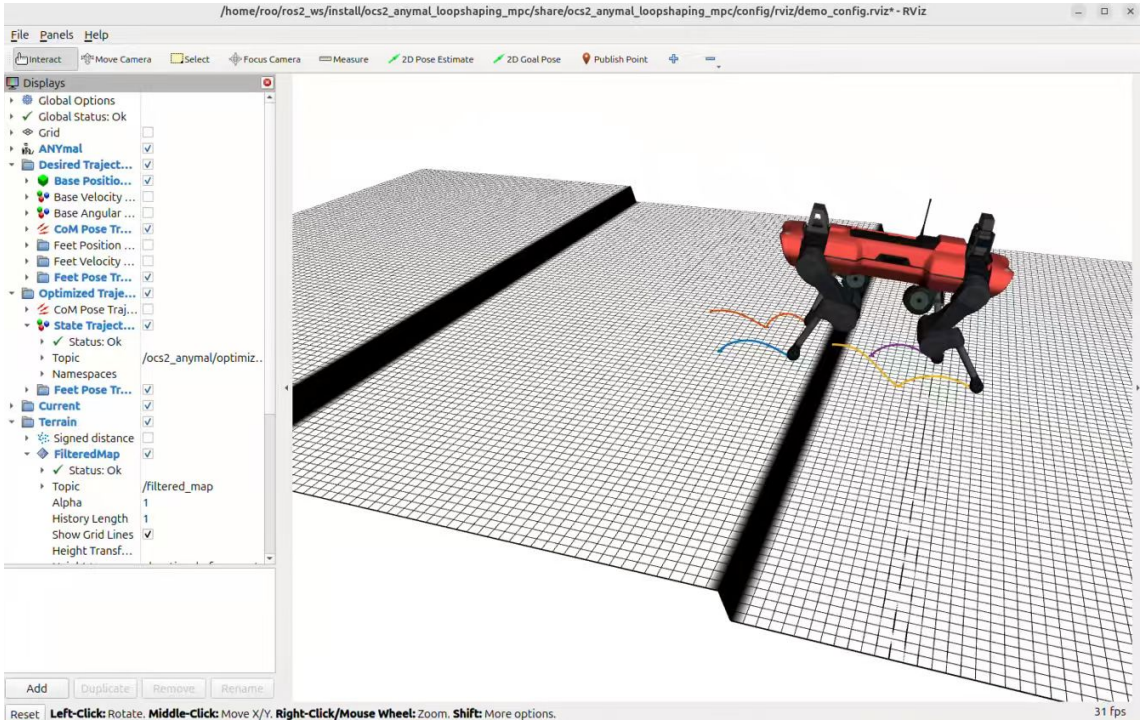


图 2: anymal 机器人离线感知上下楼梯

4.5 Go2 在线感知控制系统

在 ANYmal 验证完成后,项目将方法迁移到 Go2。Go2 系统位于 `quadruped_ros2_control`，重点工作是把感知地形、OCS2-MPC、WBC 和 Gazebo 硬件接口连接成闭环。项目在 `gz_quadruped_playground/worlds/stairs.sdf` 中建立了楼梯环境。

在感知部分,在线模式使用 `PointcloudElevationMappingNode.cpp` 从 `/rgbd_d435/points` 和 `/scan/points` 获取外部传感器点云。节点优先通过 TF 将点云转换到 `odom` 坐标系，若 TF 暂时不可用，则使用里程计作为补偿。随后，它以机器人当前位置为中心构建局部高程地图，并对小范围空洞进行邻域填补，发布 `/elevation_mapping/elevation_map_raw`。离线模式使用 `OfflinePlanarTerrainNode.cpp`，可以根据 Gazebo 楼梯几何直接生成与仿真世界一致的高程图和平面地形，也可以从 PNG 地形图加载。这一设计使项目能够分别验证“控制算法本身”和“在线感知链路”。

在控制器部分，`Ocs2QuadrupedController.cpp` 增加了 `enable_perceptive` 参数。启用后，控制器不再使用普通 `LeggedInterface`，而是创建 `PerceptiveLeggedInterface`，并将 `PlanarTerrainReceiver` 注册到 SQP 求解器的同步模块中。每次 MPC 求解前，接收到的平面地形会同步到控制器内部，参考管理器再根据地形修改机体参考和摆动足高度。

Go2 的目标管理由 `TargetManager.cpp` 完成。它不仅把键盘、手柄或 `/cmd_vel` 转换为目标速度，还会订阅离线或在线地图，对机器人当前位置、前方、后方、左右前方和前向预瞄区域进行高度采样。若前方台阶高度上升，目标管理器会提前提高机体高度，并按地形坡度修正机体俯仰角。这样机器人在接近台阶前已经开始调整姿态和高度，而不是等足端发生碰撞后才被动补偿。

最终，`StateOCS2.cpp` 中的 MPC 输出会交给 `WeightedWbc`。WBC 根据优化状态、优化输入和当前测量状态求解关节命令，并通过扩展后的 Gazebo 硬件接口发送力矩、位置、速度、刚度和阻尼。项目在 `gz_system.cpp` 中补充了 `kp`、`kd` 命令接口，并对力矩和增益做限

幅，提高了仿真控制稳定性。

4.6 `legged_perceptive` 中的感知控制模块

`legged_perceptive` 是项目中感知 MPC 模块的重要来源。它的核心贡献是把普通腿足 OCS2 接口扩展为感知型接口，使外部地形能够进入最优控制问题。保存平面地形和符号距离场，并添加足端落足约束、足端碰撞约束和腿部球模型碰撞约束。在每次 MPC 求解前根据地形修改参考轨迹，`ConvexRegionSelector` 则根据每条腿的接触相位寻找合适的可落足凸区域。

在 Go2 迁移中，项目并非简单复制这些模块，而是针对 ROS2 和在线楼梯环境做了增强。首先，地形接收端增加了有效性判断，只有在平面区域、地图层和多边形均合法时才更新内部地形，否则保留上一帧可用地形。其次，reference manager 中增加了安全高度查询和俯仰角限幅，避免地图边界外或高度为 NaN 时直接抛异常。再次，足端碰撞约束和 SDF 查询增加了初始化检查，使地图尚未准备好时约束不会破坏 MPC 求解。

这些改动反映了从算法示例到可运行系统的关键差别：理论上只要有地形就可以做落足规划，但实际在线系统中地形经常短时间无效，控制器必须在“感知不完美”的情况下保持稳定。通过测试，可以看到在连续的台阶地形下，四足机器人也可以稳定的运动，达到目标位置。

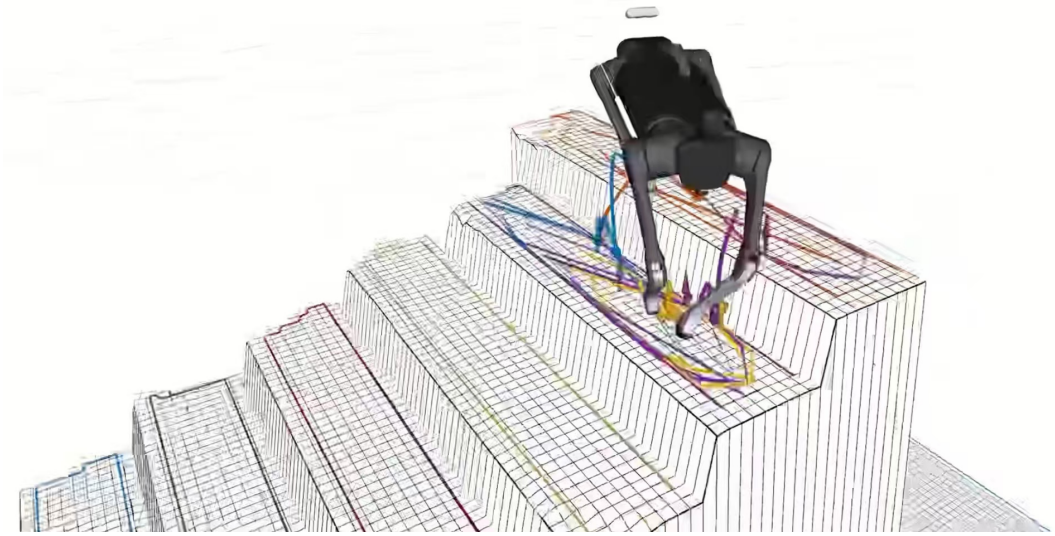


图 3: go2 机器人在线感知上下楼梯

4.7 结果分析

实验结果表明，基于外感的 MPC 控制方法能够成功实现 Go2 机器人在楼梯地形中的在线感知上下运动。系统通过深度点云与激光点云构建局部高程地图，并利用平面分割提取可支撑区域，将环境地形信息实时接入 MPC 与 WBC 控制链路。控制器能够根据前方台阶高度动态调整机体高度、俯仰角以及摆动足轨迹，使机器人在接近台阶前提前完成姿态调整与抬腿动作，从而较稳定地完成连续上下楼梯运动。在较低速度下，机器人整体运动较平稳，能够保持稳定支撑与对角步态，未出现明显侧翻或严重失稳现象。

在系统调试过程中，项目针对 MPC 权重、摆动足高度、地形参考增益以及 Gazebo 执行层 PD 参数进行了多轮调整与优化。初期实验中存在足端撞击台阶、摆动足抬升不足以及接触震荡等问题，后续通过增加 look-ahead 地形高度预测、提高 swing trajectory 高度、对机体 pitch 调整加入平滑与限幅处理，并扩展 Gazebo 中的 kp/kd 控制接口，逐步提高了系统稳定性。最终实验结果说明，感知型 MPC 能够有效利用外部地形信息，实现较为稳定和可解释的复杂地形运动控制，同时也验证了 MPC 与 WBC 在四足机器人外感运动中的工程可行性。

五、基于强化学习的外感上下楼梯方法

5.1 方法原理与训练平台

强化学习路线的目标是让机器人通过仿真试错自主学习上下楼梯策略，而不是由工程人员显式写出完整动力学约束和落足规划规则。项目使用 NVIDIA Isaac Gym Preview 4 作为仿真平台，基于 `legged_gym` 和 `rsl_rl` 开源库修改 Go2 训练任务。Isaac Gym 能够在 GPU 上并行运行大量仿真环境，本项目在单张 RTX 4060 显卡上同时运行 2048 个环境，大幅提升了采样效率。

训练算法采用 PPO（Proximal Policy Optimization），即近端策略优化。PPO 属于 Actor-Critic 结构的无模型强化学习算法，适合四足机器人这种高维连续动作空间问题。策略网络不需要显式知道机器人动力学方程，而是通过不断与仿真环境交互，根据奖励函数调整参数，逐渐形成能够稳定运动的控制策略。

强化学习系统的控制链路如图所示。

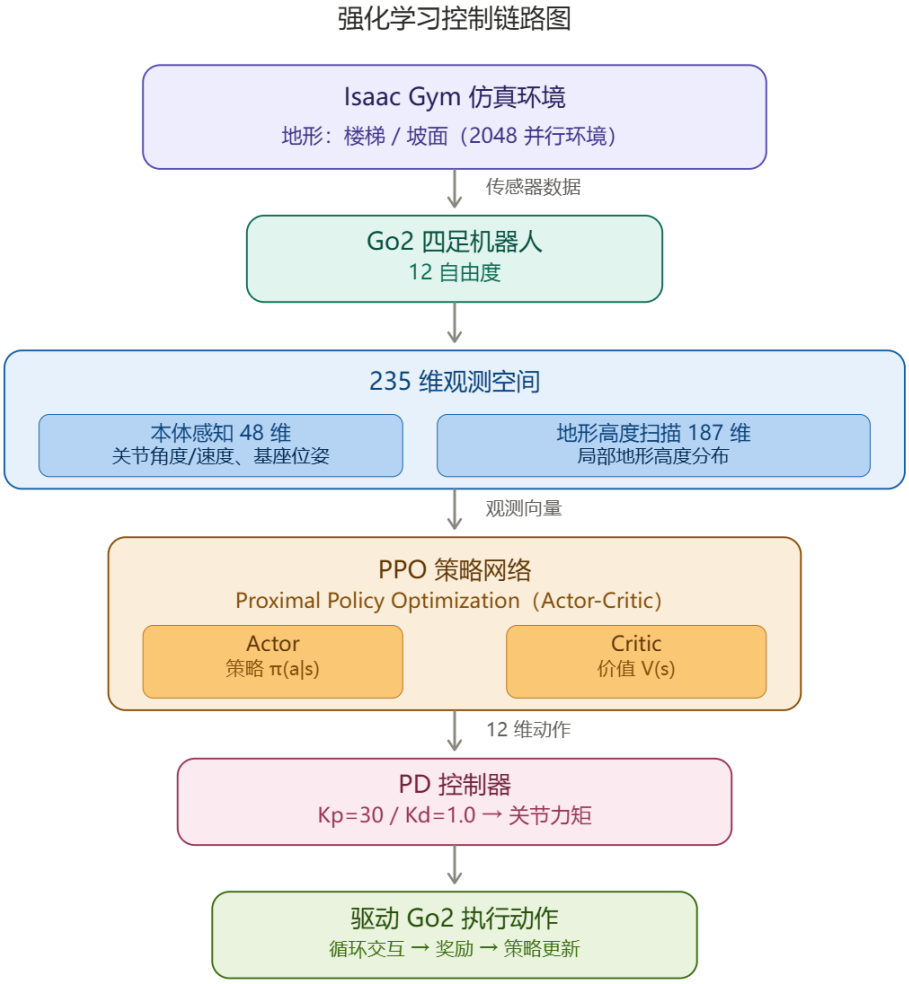


图 4: 强化学习系统的控制链路

5.2 状态空间、动作空间与底层控制

强化学习方法的观测空间总计 235 维，其中 48 维来自机器人本体感知，包括关节角度、关节速度、基座线速度、基座角速度和重力方向向量；187 维来自外部感知，即机器人周围局部地形的高度扫描数据。这里的外感信息没有被显式分割为平面区域，也没有转换为落足约束，而是直接作为神经网络输入。网络需要自己学习“看到怎样的高度分布时应该怎样抬腿、落脚和调整身体”。

动作空间为 12 维，对应 Go2 的 12 个关节。策略网络输出的不是最终力矩，而是关节目标角度残差。该残差与预先设置的默认关节姿态相加后，送入底层 PD 控制器。根据 Go2 轻量化、动态响应快的特点，项目将大腿关节默认姿态设置在约 0.8 rad 附近，PD 刚度设为 30.0，阻尼设为 1.0。这样的设计比直接输出力矩更容易训练，也能通过 PD 层过滤部分高频动作。

5.3 两阶段课程学习

直接让随机初始化的策略在楼梯和崎岖坡面上训练，容易出现无法站立、关节卡死、频繁跌倒等问题。因此项目采用两阶段课程学习。第一阶段为平地预训练，迭代 0 到 1500 次。在这一阶段，训练环境为平坦地形，不启用地形高度扫描和绊倒惩罚，主要让机器人学习姿态

保持、速度跟踪和平地行走。经过平地预训练后，模型已经能够稳定站立、行走和原地踏步。

第二阶段为复杂地形迁移，迭代 1500 到 4500 次。项目加载第一阶段得到的策略作为初始权重，将环境切换为包含上下台阶和崎岖坡面的复杂地形，同时开启高度图观测，并加入绊倒惩罚。当足端撞击台阶边缘或出现不合理接触时，奖励函数给予负反馈。经过后续约 3000 次迭代，策略逐渐学会提高摆动足、调整重心并跨越台阶，地形适应性明显增强。

这种课程学习的意义在于降低探索难度。平地阶段先学会基本动态平衡，复杂地形阶段再学习越障和高度适应。若从复杂地形直接开始训练，策略在早期几乎得不到有效正反馈，训练更容易陷入失败。

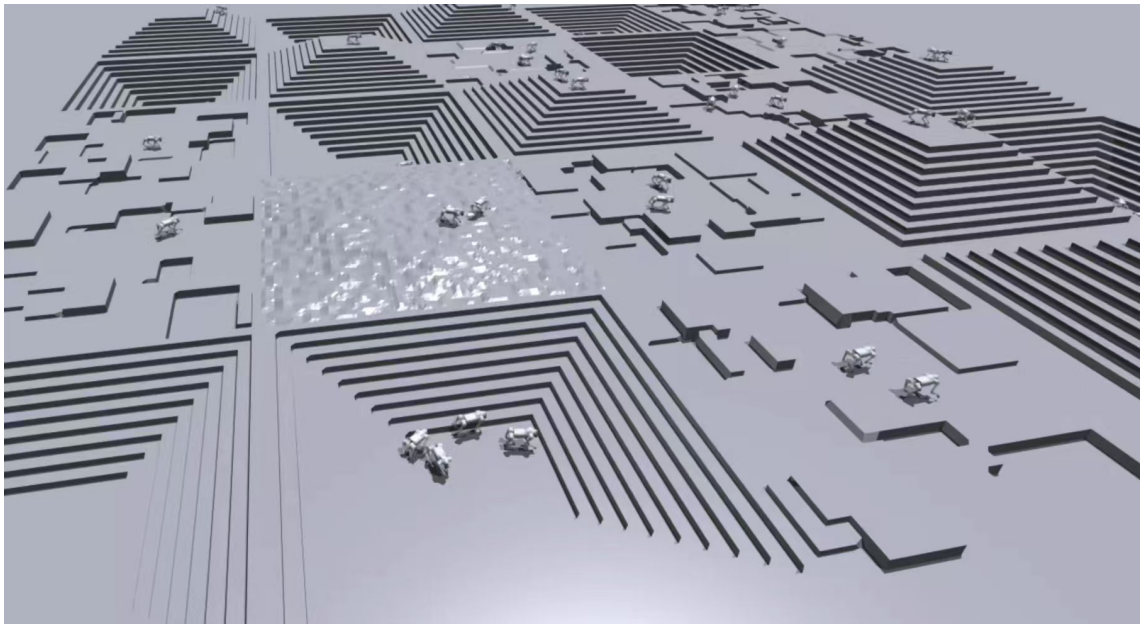


图 5: go2 基于 RL 在 IssacGym 中运动训练

5.4 实验结果与行为分析

训练至 4500 次迭代后，强化学习策略已经能够在台阶和坡面环境中实现较稳定的上下楼梯运动。从基座速度曲线看，偏航速度对指令具有较好的跟随趋势，虽然存在一定高频振荡，但整体能够完成航向维持和小幅转向。横向和纵向速度在零指令附近仍会出现波动，这并不完全是控制误差，而是策略为了维持质心平衡主动引入的动态补偿。垂直速度呈现规律周期性起伏，反映了机器人跨越台阶时重心周期性抬升和下降的过程。

足端接触力曲线显示，四条腿形成了清晰的交替支撑和摆动模式，支撑相与摆动相边界稳定。对角腿受力同步变化，说明策略自主形成了类似 Trot 的对角小跑步态。初始阶段可能出现较高接触力峰值，主要来自环境初始化或台阶高度突变带来的瞬态冲击，但策略能够较快恢复稳态。

关节轨迹和目标轨迹之间存在一定稳态偏差，这是强化学习策略中较典型的现象。策略并不是严格追踪人工设定的默认姿态，而是在奖励函数驱动下主动调整标称姿态，例如降低重心或改变关节零点附近的工作区间，以换取楼梯运动中的稳定裕度。扭矩和速度散点主要集中在安全区域，关节扭矩峰值未接近电机极限，说明策略在完成上楼梯做功的同时仍保留一定动力余量。

为了便于测试不同移动指令，项目还修改了推理脚本 `play.py`。复杂地形下原地转向容易导致失稳，因此推理阶段取消了偏航指令，主要保留前后和左右平移控制。同时加入低通

滤波器，平滑系数为 0.05，将键盘阶跃输入转为缓慢变化的速度指令，使起步、停止和侧移更平滑。

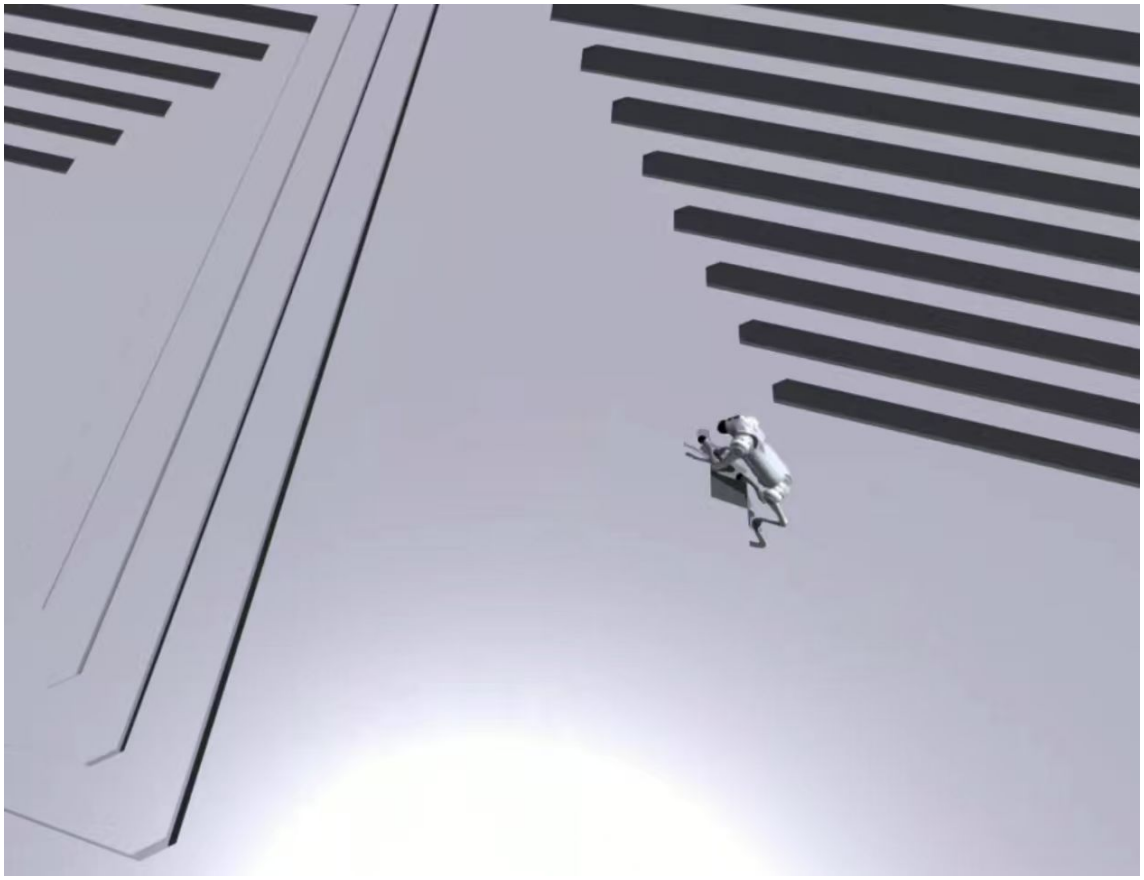


图 6: go2 机器人基于 RL 上下楼梯

5.5 强化学习路线中的问题

强化学习路线最明显的问题是灾难性遗忘。在长时间台阶地形训练中，由于楼梯上大幅转向容易摔倒并触发终止惩罚，策略为了最大化总奖励，会逐渐降低对转向指令的响应权重。最终模型具备较强的直线越障和侧移能力，但原地转向能力变弱。这说明强化学习在多目标优化中会自动做取舍：如果奖励函数中“生存”和“稳定”权重过高，而转向任务带来的失败风险较大，策略就会牺牲转向能力来保证不摔倒。

另一个问题是可解释性不足。MPC 中如果机器人抬腿不够，可以检查摆动足高度、地形高度、SDF 或约束权重；如果落足不合理，可以检查凸平面区域和落足约束。而强化学习中，策略网络的中间决策难以直接解释。它可能通过降低重心、改变步态、减小转向响应或形成某种补偿动作来获得高奖励，但这些行为背后的具体原因需要通过大量曲线、消融实验和可视化分析才能推断。

六、两种方法的对比分析

MPC 和强化学习都实现了基于外传感器的 Go2 上下楼梯，但二者适合解决的问题并不完全相同。MPC 方法强调“可建模、可约束、可解释”，强化学习方法强调“可训练、可泛化、可快速推理”。在本项目中，两条路线不是互相替代，而是构成互补关系。

两种方法的综合对比如表 4 所示。

表 4: MPC 方法与强化学习方法的综合对比

比较项	**MPC 与 WBC 方法**	**强化学习方法**
开发难度	需要建立机器人模型、接触模型、约束、参考管理器、WBC 和地形接口，前期工程复杂	不需要手写复杂动力学和约束，主要搭建训练环境、设计观测和奖励
调参方式	调整权重、约束、地形处理、MPC horizon、SQP 参数、WBC 任务优先级	调整奖励函数、课程学习、动作缩放、PD 参数、网络结构和随机化
可解释性	强。问题可定位到地图、落足区域、摆动足高度、动力学约束或 WBC	弱。策略行为来自神经网络，需要通过实验分析推断
安全约束	容易加入显式约束，如摩擦锥、落足区域、碰撞距离、关节力矩限制	通常通过奖励和终止条件间接实现，难以严格保证
实时性	在线求解 MPC，计算负担较重，求解失败需处理	训练后推理很快，适合高频控制
对模型依赖	高。模型误差会影响控制性能	低。可通过随机化提高对模型误差的鲁棒性
对数据依赖	不需要大量训练数据，但需要人工建模	需要大量仿真样本和训练时间
泛化能力	在模型和地形表达有效范围内可靠，超出假设时性能下降	可能对训练分布内地形很强，但训练外场景泛化不确定
工程可维护性	模块边界清楚，便于逐项调试	策略整体性强，问题定位更困难

从项目体验看，强化学习方法的确更“直接”。只要仿真平台、奖励函数和课程学习设计合理，机器人可以通过不断运行学会复杂行为，不需要开发者为每一种地形显式设计落足点、摆动足曲线和动力学约束。这对于台阶、坡面、碎石等复杂地形具有吸引力。但这种简单主要体现在建模层面，并不意味着强化学习没有成本。它的成本转移到了训练环境、奖励设计、并行仿真、随机化、训练稳定性和结果分析上。

MPC 方法则相反。它的前期实现较重，需要明确每个模块的输入输出，并处理求解器、地图、约束和 WBC 的耦合问题。但一旦系统运行起来，问题更容易解释。例如机器人在台阶前失稳，可以判断是机体高度参考不足、摆动足高度过低、平面地形无效、落足区域过小、MPC 权重不合适，还是 WBC 执行跟不上。对于需要安全性、可验证性和工程可维护性的机器人系统，这种可解释性非常重要。

因此，本项目更倾向于将两种方法理解为不同层级的工具。MPC 适合作为安全约束、全身控制和可解释规划框架；强化学习适合作为复杂地形下的快速策略、残差补偿或高层行为生成器。

七、问题与解决方案

项目中遇到的问题主要来自三类耦合：感知结果与控制器之间的耦合，MPC 求解器与滚动时域之间的耦合，以及仿真执行器与全身控制之间的耦合。问题与解决方案汇总如表 5 所示。

表 5: 问题分析与解决方案汇总

问题	原因分析	解决方法	效果
长时间滚动 MPC 后时间窗失效	当前时间超过求解器旧策略终止时间，MPC horizon 与当前 observation 不一致	在 `MPC_BASE.cpp` 中检测时间窗，超出后重置求解器并重新计算	支持键盘驱动的长时间滚动运行
SQP warm start 在模式变化时失效	地形和 gait 切换导致旧轨迹无法安全 spread 到新网格	在 `SqpSolver.cpp` 中检查 primal solution 形状，捕获 spread 异常，必要时清空旧解	避免求解器使用损坏 warm start
在线地形偶发为空或多边形非法	点云视野不足、TF 延迟、平面分割初期不稳定	在 `PlanarTerrainReceiver.cpp` 中过滤无效地形，保留上一帧可用地形	MPC 不会因瞬时感知失败丢失地形
SDF 查询出现 NaN 或越界	地图含空洞、尺寸过小、查询点越界或 SDF 未初始化	在 `grid_map_sdf` 和感知约束中加入初始化检查、NaN 填充和边界限制	足端碰撞约束更稳定
台阶前机体高度不足	只根据当前位置高度调整 base z，缺少前向预判	`TargetManager.cpp` 增加前方、左右前方和 lookahead 高度探测	机器人接近台阶前提前抬高机体
机体 pitch 在台阶边缘突变	高度差直接转姿态容易产生过大俯仰角	在 reference manager 和目标管理器中加入增益和限幅	姿态变化更平滑
Gazebo 执行器响应不稳定	只发送力矩或接口不足，难以匹配 WBC 输出	扩展 `kp`、`kd` 命令接口，并限制力矩与增益	WBC 输出更容易在仿真中稳定执行
步态命令被重复触发	高频循环反复读取同一个 command	在 `CtrlInterfaces.h` 中加入 command sequence/event	gait 切换更确定
强化学习转向能力退化	楼梯上转向容易摔倒，终止惩罚使策略主动降低转向响应	推理阶段取消偏航指令，并使用低通滤波平滑输入；后续需改进课程和奖励	直线越障稳定性提高，但转向能力仍需后续优化

其中，`legged_perceptive` 相关改动最能体现感知控制从“算法可行”到“系统可用”的过程。原始思想中，控制器假设收到的平面地形是有效的，SDF 可以正常查询，落足多边形可以正常生成。但在 Go2 在线楼梯场景中，这些假设经常被打破：高程图可能有空洞，平面分割可能短时间没有区域，足端查询点可能落到地图外，地形高度可能不是有限数值。项目通过有效性检查、fallback 高度、安全限幅、SDF 初始化判断和无效约束返回等方式，使控制器在不完美感知下仍能继续运行。

强化学习路线中的问题则更多来自训练目标之间的竞争。机器人为了不上台阶时摔倒，会形成保守策略；为了获得更高生存奖励，可能牺牲转向响应；为了降低冲击，可能主动改变默认姿态。这些现象不是代码 bug，而是奖励函数和训练分布共同塑造出的行为结果。因此，强化学习问题的解决方法通常不是修一个模块，而是重新设计课程、奖励权重、命令采样分布和失败条件。

八、实验结果与项目效果

MPC 路线在 ANYmal 离线验证中证明了高程图、平面分割、SDF、地形自适应参考轨迹和滚动 MPC 可以形成闭环。RViz 中能够观察到楼梯高程图、分割后的平面边界、符号距离场和机器人优化轨迹。机器人在键盘速度命令下可以持续生成与地形匹配的机体轨迹和摆动足轨迹，遇到地图边界或无效区域时能够停止并切换到站立状态。

Go2 离线楼梯实验进一步验证了方法在目标机器人上的适配效果。离线地形直接匹配 Gazebo 楼梯几何，因此可以排除点云噪声干扰，重点观察 MPC、WBC、Go2 模型和 Gazebo 硬件接口是否协调。实验表明，感知型控制器能够接收 `/planar_terrain`，目标管理器能够根据楼梯高度提高机体参考，WBC 能够将 MPC 输出转为关节力矩、位置和速度命令。在合适速度和步态下，Go2 可以完成较稳定的上下楼梯。

Go2 在线感知实验则验证了完整外感链路。深度点云和激光点云生成局部高程图，凸平面分割节点提取可支撑平面，控制器在运行过程中同步地形并修正参考轨迹。在线模式比离线模式更接近真实机器人部署，但也对 TF、点云质量、传感器视野和分割参数更敏感。当前实现已经形成可演示的在线感知上下楼梯闭环。

强化学习实验在 Isaac Gym 中完成。经过两阶段训练后，Go2 策略能够在台阶和坡面地形中保持动态平衡，并形成较规律的对角小跑步态。接触力曲线呈现稳定交替模式，基座垂向速度反映出跨越台阶时的周期性重心抬升，关节扭矩未接近电机极限，说明策略具备一定安全裕度。相比 MPC 路线，强化学习路线没有显式构造落足区域和动力学约束，但通过局部高度扫描和奖励函数学到了可用的外感运动策略。

九、代码库与工作贡献

`ocs2_ros2` 中的主要工作是建立 ANYmal 离线感知验证平台，并增强 OCS2 滚动 MPC 的工程稳定性。`PerceptiveMpcDemo.cpp` 被改造为键盘驱动的滚动演示，新增楼梯地形数据，并保留高程图、平面分割、SDF 和优化轨迹可视化。同时，项目对 `MPC_BASE.cpp`、`SqpSolver.cpp` 和 `grid_map_sdf` 做了鲁棒性修复，使长时间运行和地形突变场景更加稳定。

`quadruped_ros2_control` 中的主要工作是将感知 MPC 落地到 Go2。项目新增楼梯 Gazebo 世界、点云高程建图节点、离线平面地形节点、感知控制启动参数、感知型 OCS2 controller、地形自适应目标管理器、步态命令同步和 Gazebo 硬件接口扩展。该仓库最终承担 Go2 在线感知、MPC、WBC 和仿真执行的完整闭环。

`legged_perceptive` 中的主要工作是为感知型 MPC 提供算法模块基础，包括 `PerceptiveLeggedInterface`、`PerceptiveLeggedReferenceManager`、`ConvexRegionSelector`、`FootPlacementConstraint`、`FootCollisionConstraint`、`SphereSdfConstraint` 和地形可视化。Go2 仓库中的 perceptive 模块继承并改造了这些设计，使其适配 ROS2 和 Go2 楼梯场景。

强化学习部分的主要工作是搭建 Isaac Gym 训练环境，基于 `legged_gym` 和 `rsl_rl` 修改 Go2 任务，设计 235 维观测空间、12 维动作空间和两阶段课程学习流程，并通过 PPO 训练获得上下楼梯策略。项目还分析了基座速度、接触力、关节轨迹、扭矩速度分布和灾难性遗忘现象，并修改推理脚本以提升复杂地形下的操作稳定性。

十、结论与展望

本项目完成了两种基于外感的四足机器人上下楼梯方法。基于模型的 MPC 方法构建了从外部传感器到高程地图、平面分割、地形参考、落足约束、MPC 求解和 WBC 执行的完整链路，具有较强可解释性和工程可维护性。基于强化学习的方法通过 Isaac Gym 大规模并行训练，使 Go2 在不显式建立复杂动力学约束的情况下学会了上下楼梯策略，体现了数据驱动方法在复杂地形运动中的潜力。

后续研究可以重点考虑 MPC 与强化学习的结合。第一个方向是 “学习残差 + 模型控制”：MPC 继续负责动力学一致性、接触约束和安全边界，强化学习策略只输出对机体高度、足端轨迹或关节命令的残差补偿，用于弥补模型误差和复杂地形中的未建模效应。第二个方向是 “学习型高层规划 + MPC 低层执行”：强化学习根据地形高度图选择速度、步态、落足倾向或通过方向，MPC 负责将这些高层意图转化为满足约束的轨迹。第三个方向是 “MPC 生成专家数据，RL 蒸馏策略”：先用 MPC 在大量地形中生成可解释、安全的专家轨迹，再训练神经网络模仿 MPC，从而获得接近 MPC 的安全性和接近 RL 的推理速度。第四个方向是 “安全过滤器”：RL 负责快速给出动作，MPC 或控制屏障函数在动作执行前检查摩擦、碰撞、关节限制和稳定性，必要时修正策略输出。

在感知方面，未来可以将当前高程图扩展为概率地形图或时序融合地图，增强对点云噪声、遮挡和动态变化的鲁棒性。在强化学习方面，可以通过命令课程、地形课程、奖励重平衡和域随机化缓解灾难性遗忘，提高转向、侧移和上下楼梯之间的综合能力。在真实机器人部署方面，还需要进一步处理传感器标定、状态估计延迟、执行器带宽、地面摩擦变化和安全保护等问题。

总体而言，本项目已经形成了从可解释优化控制到数据驱动学习控制的完整研究基础。两条路线分别验证了模型方法和学习方法在外感四足机器人上下楼梯任务中的可行性，也为后续开展 MPC-RL 融合、复杂地形自主导航和真实 Go2 实验奠定了基础。

参考文献

1 Fankhauser P, Hutter M. A Universal Grid Map Library: Implementation and Use Case for Rough Terrain Navigation. Springer, 2018.

2 Grandia R, Jenelten F, Yang S, et al. Perceptive Whole-Body Planning for Multi-Contact Locomotion on Legged Robots. *arXiv preprint arXiv:2207.02643*, 2022.

3 Schulman J, Wolski F, Dhariwal P, et al. Proximal Policy Optimization Algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

4 Di Carlo J, Wensing P M, Katz B, et al. Dynamic Locomotion in the MIT Cheetah 3 Through Convex Model-Predictive Control. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2018: 1-6.

5 Rudin N, Hoeller D, Sakar M S, et al. Learning to Walk in Minutes Using Massively Parallel Deep Reinforcement Learning. *arXiv preprint arXiv:2109.11978*, 2021.

6 Lee J, Hwangbo J, Wellhausen L, et al. Learning Quadrupedal Locomotion over Challenging Terrain. *Nature*, 2020, 586(7830): 497-502.

7 Giftthaler M, Neunert M, Stäuble M, et al. A Family of MPC-Based Controllers for Dynamic Legged Robots. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2017: 2620-2625.

8 Mastalli C, Merkt W, Xin G, et al. MPC-Based Controller with Terrain Insight for Dynamic Legged Locomotion. *IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2020: 8284-8290.

9 Winkler A W, Mastalli C, Havoutis I, et al. Planning and Execution of Dynamic Whole-Body Locomotion for a Hydraulic Quadruped on Challenging Terrain. *IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2015: 5148-5154.

10 Katz B, Di Carlo J, Kim S. Mini Cheetah: A Platform for Pushing the Limits of Dynamic Quadruped Control. *IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2019: 6295-6301.

11 Bellicoso C D, Bjelonic M, Wellhausen L, et al. Advances in Real-World Applications for Legged Robots. *Journal of Field Robotics*, 2018, 35(8): 1311-1326.

12 Jenelten F, Farshidian F, Hutter M. Perceptive Locomotion Through Nonlinear Model Predictive Control. *IEEE Robotics and Automation Letters*, 2022, 7(3): 7684-7691.

13 Farshidian F, Buchli J, Hutter M. Efficient and Modular Implementation of Dynamic Locomotion Tasks for Quadrupedal Robots. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2017: 2000-2006.

14 Rudin N, Kolvenbach H, Tsounis V, et al. Learning High-Speed Navigation for Quadrupedal Robots with Sparse Rewards. *Robotics: Science and Systems (RSS)*, 2022.

15 Hwangbo J, Lee J, Dosovitskiy A, et al. Learning Agile and Dynamic Motor Skills for Legged Robots. *Science Robotics*, 2019, 4(26): eaau5872.