一种模块化残差学习框架，以增强基于模型的方法实现稳健的运动能力

Min-Gyu Kim [1] , Dongyun Kang [1] , Hajun Kim [1] , and Hae-Won Park [1]

***Abstract*—本文提出了一种新颖的方法，将基于模型和基于学习的框架的优势结合起来，以实现稳健的运动能力。残差模块与基于模型的框架中的相应部分（使用启发式方法设计的步态规划器和动态模型）集成，以弥补模型不匹配导致的性能下降。通过利用模块化结构并为每个残差模块选择适当的基于学习的方法，我们的框架在高不确定性环境中表现出改进的控制性能，同时与基线方法相比，实现了更高的学习效率。此外，我们观察到，我们提出的方法不仅提高了控制性能，还提供了额外的好处，比如使名义控制器对参数调整更稳健。为了验证我们框架的可行性，我们在真实四足机器人上展示了与模型预测控制结合的残差模块。尽管存在超出模拟的不确定性，机器人仍成功保持平衡并跟踪指令速度。**

***Index Terms*—Legged Robots, Machine Learning for Robot Control, Optimization and Optimal Control**

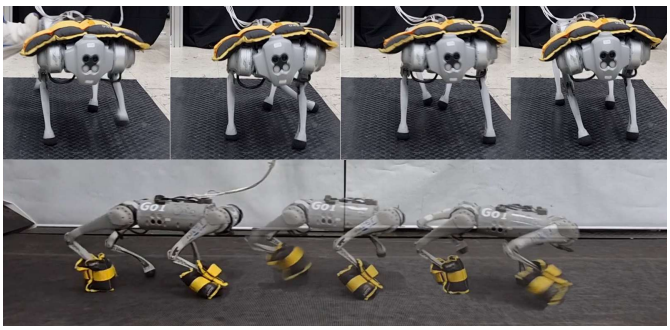

Fig. 1: 实验结果的快照。带有由残差模块调整的传统基于模型控制器的四足机器人能够克服诸如未知载荷和扰动等不确定性。

I. 介绍

L 煽动 系统因其在城市和恶劣环境中导航并有效执行分配任务的潜力而引起研究人员的兴趣。基于模型的方法（MBA），尤其是模型预测控制（MPC），因其能够处理系统动态和约束而被广泛研究。尽管基于学习的方法（LBA）有所兴起，MBA 仍是生成安全和一致运动的核心技术。

然而，MBA 面临着挑战，最为显著的是由于实际时间可行性要求的必要简化而导致的模型不匹配。由于腿式系统在接触时表现出复杂的混合动力学，因此用于控制设计的模型通常被简化，如单一刚体模型或线性倒立摆模型。虽然这降低了计算成本，但可能导致诸如接触动力学等关键信息的丢失，从而在未建模的场景中降低性能。

为了解决这些限制，LBA 可以提供一种有前景的补充，擅长通过数据建模复杂行为。MBA 和 LBA 的协同整合可以结合 MBA 的可靠性和 LBA 的适应性，即使在显著的不确定性下也能产生稳健的控制。

本文提出了一种混合方案，该方案利用 LBA 来弥补现有 MBA 的局限性。虽然 MBA 可以提供精细而可靠的运动，但由于模型的小误差或次优启发式方法，以及重载或外部干扰等重大不确定性，MBA 性能容易退化。

为了解决这个问题，我们在两个关键组件中引入了残差模块，即步态规划器和动态模型。每个残差模块都采用机器学习技术设计，以提供辅助动作。结果，混合控制器在面对不确定性时比传统的 MBA 具备更好的鲁棒性，同时相较于端到端的 LBA，亦能在训练域之外保留一致性。图 1 展示了一个鲁棒性的例子，其中机器人处理一个名义 MBA 无法单独应对的负载和干扰。

我们方法的一个关键优势是其模块化和简化的模块设计，使用强化学习（RL）和监督学习（SL），而不是完全依赖于 RL，这样可以在训练过程中保持名义控制器的同时实现高效学习。具体来说，我们提出了两个残差模块：(i) 基于 RL 的残差步态模块，用于自适应地校正脚落模式以应对干扰，(ii) 基于 SL 的残差动力学（RD）模块，用于处理系统模型中的差异。

我们通过增强学习（RL）解决复杂接触动力学中的足点选择挑战，同时通过监督学习（SL）补偿连续域动力学差异，将其与 RL 训练分离。这种方法减少了学习搜索空间，简化了 RL 过程。为了进一步降低计算负担，我们使用具有简化动态模型的凸模型预测控制（MPC）进行 3D 运动。此外，我们应用低通滤波器对残差进行处理，以隔离缓慢变化的不确定性，从而增强对噪声的鲁棒性。通过关注这些缓慢变化，我们假设残差项在 MPC 预测范围内保持不变，进一步简化了优化问题。与基线相比，我们的方法因其结构的简化、提高的训练效率以及在广泛的分布外（OOD）场景中的一致性能而显得突出。我们强调我们的贡献如下。

- 我们提出了一种混合方案，该方案利用基于学习的残差模块来补偿传统 MBA 中由于模型不准确和次优启发式算法导致的性能下降。
- 该框架采用轻量级模块化架构，通过 RL 进行步态适应和 SL 进行连续域动态校正，结合简化的名义动态

Manuscript received: March, 31, 2025; Revised June, 21, 2025; Accepted July, 13, 2025.

This paper was recommended for publication by Editor Abderrahmane Kheddar upon evaluation of the Associate Editor and Reviewers' comments. This work was supported by the Technology Innovation Program(or Industrial Strategic Technology Development Program-Robot Industry Technology Development)(RS-2024-00427719, Dexterous and Agile Humanoid Robots for Industrial Applications) funded By the Ministry of Trade Industry & Energy(MOTIE, Korea); in part by Korea Evaluation Institute of Industrial Technology (KEIT) funded by the Korea Government (MOTIE) under Grant No.20018216, Development of mobile intelligence SW for autonomous navigation of legged robots in dynamic and atypical environments for real application.

[1] The authors are with Humanoid Robot Research Center, Department of Mechanical Engineering, Korea Advanced Institute of Science and Technology (KAIST), Yuseong-gu 34141 Daejeon, Republic of Korea. haewonpark@kaist.ac.kr

Digital Object Identifier (DOI): see top of this page.

和基于过滤的 RD 解耦，实现高效学习。
- 大量实验表明，与基线相比，我们的方法在严重干扰和 OOD 条件下实现了稳健和可靠的任务执行，降低了参数调优的敏感性，并提高了学习效率，而不影响性能。

II. 相关工作

A. 基于模型和学习的方法

传统上，为了控制多足机器人，引入了一些基于启发式的方案，其中原始系统通过简化模型来表示，如线性倒立摆模型（LIPM）或单刚体模型（SRBM）[?], [?], [?] 。这一方案已通过一系列演示被验证为简单但实用。

在这些方法中，MPC 因其能够处理动态任务和系统约束而受到关注 [?], [?], [?], [?] 。然而，其计算成本通常需要简化，例如预定义步态序列或简化动力学如 SRBM，从而导致性能下降。

相比之下，数据驱动的方法在具有挑战性的任务中显示出令人印象深刻的结果 [?], [?], [?], [?] ，但也面临如次优收敛等问题。虽然系统识别 [?], [?] 和在线适应 [?], [?] 之类的技术提高了灵活性，端到端的 RL 方法仍然在样本效率低下和复杂的奖励设计过程等问题上苦苦挣扎。

为了解决由于模型差异导致的 MBA 性能下降问题，许多残差估计方法已在各种领域如腿部运动、无人机和车辆中被提出。这些方法的差异在于残差对实际差异的表示精确程度，以及残差是在线更新的还是离线设计的以分层方式使用。

具体而言，残差估计技术包括从自适应控制 [?] 到基于数据的方法，这些方法包括使用线性模型 [?] 进行在线回归、基于基函数学习 [?] 或高斯过程 [?] 的概率回归，以及离线训练的神经网络模型 [?], [?], [?] 。根据其复杂性，每种方法在精确性、实时能力以及与基于优化的控制器的集成易用性之间表现出权衡。

虽然通过残差估计来补偿动态差距在处理各种干扰和外部负载方面很有效，但仅靠这种方法可能不足以在非结构化环境中实现稳健控制，特别是对于不稳定且高度动态的系统，如多足机器人。

B. 混合方法

为了克服 MBA 和 LBA 的局限性，最近的研究探索了混合策略以结合它们的优势。有些工作使用 LBA 替换 MBA 中启发式定义的高级参考（例如，命令或步态序列），以便在不确定环境中更好地适应 [?], [?], [?], [?], [?] 。

额外的方法应用残余动作，例如力矩或关节参考，以直接微调标称输出 [?], [?], [?], [?] 。虽然这可以通过允许残余覆盖标称降级来提高反应性，但由于 MBA 中缺乏基于约束优化的可行性检查，面对不确定性时可能会带来不稳定性。

最后，若干研究利用 MBA 来加速 RL 的初始学习阶段，利用其生成精细动作的能力 [?], [?] 。同样，能够进行长期规划的 MBA，例如轨迹优化，也被用来解决稀疏奖励学习中 RL 的挑战 [?], [?] 。虽然这些方法提高了学习效率和收敛性，但由于在训练期间反复计算 MBA，它们也引入了计算开销。

III. 带名义控制器的残差模块

A. 系统概述

我们的方法将基于学习的残差模块集成到 MBA 中，如图 2 所示。每个模块都被设计为调节其对应的名义模块的输出，以提高特定任务的性能。

所提出方法的一个关键方面是根据每个组件的特征，利用 RL 和 SL 的顺序设计残差模块。在无人机的相关工作中，SL 已被用于从现实世界的数据 [?], [?], [?] 中建模空气动力学或执行器的动态。然而，由于需要选择落脚点，腿式机器人面临额外的挑战。

为了解决这个问题，我们首先引入一个基于 RL 的残差模块来补偿启发式步伐规划器，使其能够自适应调整脚步。在此过程中，我们利用模拟数据分析计算名义上连续动态中的差异。这些差异通过使用本体感受传感器数据历史进行 SL 重构，以确保在现实世界中的可靠性能。这种方法通过在 RL 过程中减少优化变量的数量来提高样本效率，同时在性能上不逊于其他混合方法。

B. 名义混合动力学

本节解释了一种足式系统的混合动力学模型及相应的残差模块。离散混合动力学可以表达如下 [?] 。

$$\dot{x} = f(x,u), \quad x \notin S \tag{1}$$

$$x^+ = \Delta(x^-), \quad x \in S \tag{2}$$

其中，x 是状态；u 是地面反作用力（GRF）；S 是包含每个接触时刻的切换集合；$f(x,u)$ 是连续动力学；而 $\Delta(x)$ 是切换动力学。在整篇论文中，矢量变量以粗体表示，而矩阵变量则用书法字体表示。当接触状态不变时，系统遵循如下的连续动力学。

$$x = [p,\ \boldsymbol{\phi},\ \dot{p},\ \boldsymbol{\omega}] \in \mathbb{R}^{12}, \tag{3}$$

$$\dot{x} = f_n(x,u) = \begin{bmatrix} \dot{p} \\ \mathcal{M}(\boldsymbol{\phi}_0)\boldsymbol{\omega} \\ \frac{1}{M}\sum_{i=1}^{4} u_i + g \\ \mathcal{I}_{\mathbb{W}}^{-1}\sum_{i=1}^{4}(p_{f,i} \times u_i) \end{bmatrix}, \tag{4}$$

其中，f_n 是标称的连续动力学；p 是质心（CoM）的位置；p_f 是落脚点；$\boldsymbol{\phi}$ 是欧拉角；$\boldsymbol{\omega}$ 是在主体框架中表达的质心的角速度 $\mathbb{B}$ ；$\mathcal{M}(\boldsymbol{\phi}_0)$ 是一个映射矩阵，用于在操作点 $\boldsymbol{\phi}_0$ 将角速度转换为欧拉角速度率；M 是机器人的集总质量；g 是重力；$\mathcal{I}_{\mathbb{W}}$ 是在世界框架 $\mathbb{W}$ 中的惯性矩阵，与以下关系 $\mathcal{I}_{\mathbb{W}} = \mathcal{R}(\boldsymbol{\phi}_0)\mathcal{I}_{\mathbb{B}}\mathcal{R}(\boldsymbol{\phi}_0)^T$ 相关；而 $\mathcal{R}(\cdot)$ 是旋转矩阵。

在这项工作中，我们选择了一种基于 SRBM 的凸 MPC 作为 [?] 中的名义控制器。该框架在许多研究中已被广泛应用，并具有一些良好的特性。例如，凸优化公式通过利用最先进的求解器确保快速计算时间，对大规模学习过程具有效益。

C. 残差步态模块

为了简化，我们假设每当接触状态改变时，着地点会瞬时变化。因此，我们按照 [?] 中的建议，在一个独立的基于启发式的规划器中分层定义步伐变量。i 条腿的着地点和步态模式定义如下。

$$p_{f,i}^{+} = p_{f,i}^{-} + \delta p_{f,heuristic,i}, \quad x \in S \tag{5}$$

$$\Phi_{k,i} = \Phi_{k-1,i} + \frac{\delta t}{T_{step,i}}, \tag{6}$$

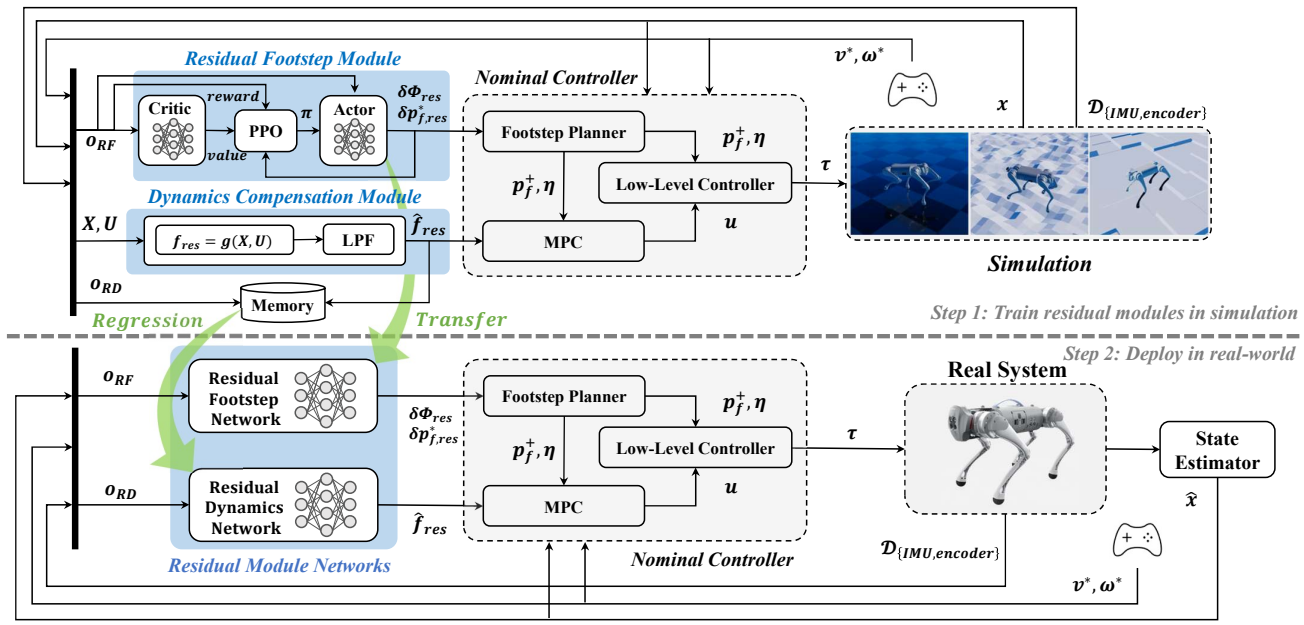

Fig. 2: 整体架构的示意图。每个残差模块通过寻找辅助动作来补偿相应标称模块中的模型不匹配。残差步态模块在仿真中使用强化学习进行学习，同时在学习过程中同时收集 RD 数据并将其输入标称控制器。然后，将 RD 模块重建为神经网络以用于实际部署。

TABLE I: 领域随机化列表

Parameter	Notation	Range
Command velocity	$[v_x^*, v_y^*, \omega_z^*]$	$\pm[2.0, 1.0, 1.0]\ m(rad)/s$
Friction coefficient	μ	$[0.4,\ 1.0]$
Payload	$M_{payload}$	$[-1.0,\ 5.0]\ kg$
Bumpiness	h_{env}	$[0, 0.1]\ m$
CoM position	$\delta p_{CoM,xyz}$	$\pm 0.05\ m$
Initial rotation	ϕ_{init}	$\pm 15\ deg$
Initial height *	δz_{init}	$[0, 0.1]\ m$

TABLE II:

* 偏离名义重心高度。

其中 $\delta p_{f,heuristic}$ 是来自启发式规划器的参考着地点；$\Phi \in [0,1]$ 是步态相位参数；δt 是控制时间步长；而 T_{step} 表示摆动和站立阶段的步伐时间。例如，初始化为站立状态的腿如果 Φ 从零开始，并在每个控制周期增加 $\delta t/T_{stance}$ 。当 Φ 达到 1 时，重置为零，并将 T_{step} 设置为 T_{swing} ，重复相同的过程。

由于启发式步态规划器包含固定步态模式和简化模型，无法完全反映实际动力学，例如惯性效应或不平坦地形，残余步态模块如下进行补偿。

$$p^{+}_{f,i} = p^{-}_{f,i} + \delta p_{f,heuristic,i} + \delta p_{f,res,i}, \quad x \in S \tag{7}$$

$$\Phi_{k,i} = \Phi_{k-1,i} + \frac{\delta t}{T_{step,i}} + \delta\Phi_{res,k-1,i}, \tag{8}$$

其中 $\delta p_{f,res}$，$\delta\Phi_{res}$ 为残余步态和相位。

残差步态模块是通过强化学习来学习的。名义控制器采用了一种启发式步态规划器，该规划器假定地形平坦且无滑动条件，无法完全处理更复杂的场景，例如具有不同摩擦系数的崎岖或台阶地形。为了解决这些局限性，我们在三种地形类型（平坦、崎岖、台阶）以及不同的摩擦条件下训练这些模块。此外，我们在每次重置时随机化系统和环境参数（如表 I 所详述），并将机器人初始化在略高于地面的随机身体角度和高度，以进一步增强鲁棒性。

奖励函数的设计类似于 MPC 的成本函数，以便该模块可以按照标称控制器的预期运作。由于标称控制器可以计算行走模式和所需的控制输入，它减少了学习的初始搜索空间，从而奖励函数可以设计得很简单，如下所示。

$$r_{total} = r_{alive} + c_1||x^* - x|| + c_2||\boldsymbol{\tau}||, \tag{9}$$

$$r_{alive} = \begin{cases} -10, & if robot fails \\ 0, & otherwise \end{cases} \tag{10}$$

，其中 r_{alive} 是在系统失败时的惩罚；x^* 是用户定义的期望状态；$\boldsymbol{\tau}$ 是关节扭矩；c_i 是每个奖励的权重。

训练使用近端策略优化（PPO）[?] 进行，产生 16 维输出，包括所有腿的残差落脚点和相位。PPO 设置的细节在表格 III 中提供。残差步态网络 $\boldsymbol{\pi}_{RF}$ 的观测 o_{RF} 构建如下。

$$[\delta p_{f,res,k}, \delta\Phi_{res,k}] = \boldsymbol{\pi}_{RF}(o_{RF,k}), \tag{11}$$

$$o_{RF,k} = [\dot{p}^*_k, \dot{\boldsymbol{\omega}}^*_k, \boldsymbol{\phi}_k, \boldsymbol{\omega}_k, \{\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}\}_{j,\{k,\cdots,k-h\}}, \delta p_{f,heuristic}, \Phi_{k,i}, \boldsymbol{\eta}, \boldsymbol{\tau}_k] \in \mathbb{R}^{113}, \tag{12}$$

，其中 $\dot{p}^*, \dot{\boldsymbol{\omega}}^*$ 是一个命令速度；$\boldsymbol{\theta}_{j,\{k,\cdots,k-h\}}$ 是具有窗口大小 h 的关节位置历史；$\boldsymbol{\eta}$ 是一个布尔向量，表示来自名义步态规划器的每条腿的计划接触序列；$\boldsymbol{\tau}$ 是一个关节力矩。注意，本文中使用了 $h = 2$ 。

D. 残差动态模块

$(f_n(x,u))$ 中的标称模型包含旋转动力学，旋转动力学在局部是与时间无关的，并基于欧拉角进行线性化。然后通过控制时间域使用数值前向欧拉积分预测状态。

$$x_{k+1} = x_k + \delta t f_n(x_k, u_k), \tag{13}$$

TABLE III: 用于残差步伐的 PPO 超参数

Parameter	Value
Network size (actor & critic)	[256, 128]
Activation function	LeakyReLU
# of environments	100
# of environment steps per update	200
# of batches	4
# of epochs	4
Learning rate	0.0005
Discount factor	0.996
GAE	0.95
Clip range	0.2

事实上，该模型并未完全考虑到实际的全身动力学或外部干扰，这主要是由于切换阶段的影响，可能导致性能下降。该模型差异 f_{res} ，可以使用状态历史和控制输入来确定，如下所示。

$$f_{res,k-1} = \delta t^{-1}(x_k - x_{k-1}) - f_n(x_{k-1}, u_{k-1}). \tag{14}$$

为了在实际应用中部署这一术语，必须解决几个挑战。首先，由于它被直接用作 MPC 的输入，其数值的过度波动会显著影响控制器的性能。此外，虽然将 RD 表示为 MPC 状态和控制输入的函数可以实现更准确的动态预测并可能允许其在 MPC 优化时间段 [?] 内使用，但这种方法将大大增加 RL 过程的计算负担。

为了缓解这些问题，我们设计了一个低通 IIR 滤波器：

$$\hat{f}_{res,k} = e^{\frac{-2\pi F_c}{F_s}} \hat{f}_{res,k-1} + (1 - e^{\frac{-2\pi F_c}{F_s}}) f_{res,k}, \tag{15}$$

，其中 F_c, F_s 是截止频率和采样频率。我们在这项工作中设定了 $F_c = 10\ Hz$, $F_s = 1\ kHz$ 。该滤波器被设计用于仅捕获不确定性的主要低频分量，例如有效载荷或外部干扰，这些可以在每个 MPC 控制回路中被视为准静态。此外，它还可以防止系统对噪声信号过于敏感。

尽管如此，(15) 中的解析 RD 仍然需要准确的信息，包括线速度、接触状态和 GRF。在现实环境中估计这些信息是有挑战性的，因此导致的 RD 测量不准确可能导致系统的灾难性故障。

一种可能的解决方案是，在收集仿真或实验数据后训练神经网络 [?] 。我们首先收集仿真数据，以便在训练残余足迹模块时准确标记 RD。训练完足迹模块后，我们将 RD 的数据集和相应的观测值重构到一个神经网络模型 $\boldsymbol{\pi}_{RD}$ 中，仅使用直接可测量的本体感受传感器数据（如 IMU 和关节编码器）来预测残余项。然后可以使用 SL 训练此模型。

$$\hat{f}_{res,k} = \boldsymbol{\pi}_{RD}(o_{RD,k}, \cdots, o_{RD,k-h}), \tag{16}$$

$$o_{RD,k} = [\boldsymbol{\phi}_k, \boldsymbol{\omega}_k, \boldsymbol{\theta}_{j,k}, \dot{\boldsymbol{\theta}}_{j,k}, \boldsymbol{\tau}_k, \hat{f}_{res,k-1}] \in \mathbb{R}^{54}. \tag{17}$$

该网络的设计结构与残差脚步模块相同。我们随机收集了总计一千万的数据来训练该模块。与仅依赖于分析设计相比，这种回归在现实世界中推断 RD 时提高了稳健性。

结合上述的动态性和步态规划，MPC 尝试解决以下优化问题以获得控制输入。

$$\underset{X,U}{\text{minimize}} \quad \sum_{i=0}^{N-1} l(x_{k+i}, u_{k+i}) \tag{18}$$

$$\text{subject to} \quad \dot{x}_{k+i+1} = f_n(x_{k+i}, u_{k+i}) + \hat{f}_{res,k}, \tag{19}$$

$$X \in \mathbb{X},\ U \in \mathbb{U} \tag{20}$$

TABLE IV: 鲁棒性测试的实验结果

Name	Normal (A)	With payload (B)	With disturbance (C)
vanilla-MPC	0.0216	Failed	Failed
res-dyn	0.0118	0.0538	Failed
res-all	0.0212	0.0197	0.0695

TABLE V:

值表示滚转角的均方根误差。

，其中 X, U 是一组贯穿预测视野 N 的状态和控制输入；$l(x, u)$ 是一个要最小化的代价函数；$\{\mathbb{X}, \mathbb{U}\}$ 是状态和控制输入的可行域。注意，当前的残差项 $\hat{f}_{res,k}$ 是由 RD 模块提供的，并且独立于状态和控制输入。因此，它在每次控制迭代中在整个预测视野中统一应用。这种基于滤波的解耦允许在学习和实际应用中快速计算 MPC。

代价函数被表述为一个最小二乘形式。

$$l(X, U) = X^T diag(w_x) X + U^T diag(w_u) U, \tag{21}$$

$$w_x = [w_p, w_{\boldsymbol{\phi}}, w_v, w_{\boldsymbol{\omega}}], \tag{22}$$

其中 $diag(\boldsymbol{\zeta})$ 是一个由对角向量 $\boldsymbol{\zeta}$ 组成的对角矩阵；而 $w_{(\cdot)}$ 是每个变量的权重。

前馈和反馈关节力矩通过腿部运动学计算，如下所示 [?] 。

$$\boldsymbol{\tau}_{ff,i} = \mathcal{J}(q)_i^T u_i, \tag{23}$$

$$\boldsymbol{\tau}_{fb,i} = \mathcal{J}(q)_i^T [\mathcal{K}_P(p^*_{f,i} - p_{f,i}) + \mathcal{K}_D(\dot{p}^*_{f,i} - \dot{p}_{f,i})], \tag{24}$$

$$\boldsymbol{\tau}_i = \boldsymbol{\tau}_{ff,i} + \boldsymbol{\tau}_{fb,i}, \tag{25}$$

，其中 $\mathcal{J}_i$ 是足部雅可比矩阵，$p^*_{f,i}$ 是来自步态规划器的参考足部位置，$\mathcal{K}_{P,D}$ 是笛卡尔 PD 增益。

IV. 结果

在本节中，我们进行对比实验，以评估所提出的框架在命令跟踪和抵御不确定性方面相对于基准控制器的优势。这些实验评估了系统可靠处理一系列干扰和模型误差（如踢击或重载荷）的能力。

此外，我们研究提出的方法是否能够减少标称 MBA 控制器的控制参数依赖性。我们还进行 OOD 测试，以评估在与仿真期间遇到的条件显著不同的情况下，我们的框架是否能保持一致的性能，并与现有的端到端 RL 方法进行比较。最后，我们评估与基线方法相比，我们的残差模块设计如何影响学习效率和收敛性。

A. 实验设置

我们通过模拟和使用 Unitree Go1 的真实实验验证了我们的框架，该框架是一个具有 12 自由度的 12 kg 四足机器人。模拟环境是使用物理模拟器 RAISIM [?] 构建的。MPC 和残差模块在 100 Hz 更新，而其余的低级部分运行在 1 kHz 。需要注意的是，MPC 在每次控制迭代中以 0.01 s 的时间步长预测下一个 0.1 s ，并涵盖 10 个预测视界。我们利用线性卡尔曼滤波器和基于动量的接触检测进行状态估计。名义上的站立和摆动时间设置为 0.3 s ，对应于一种小跑步态。

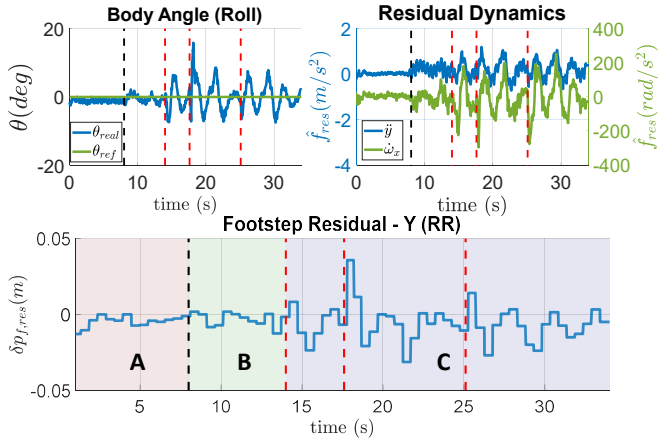

Fig. 3: 一个使用 res-all 进行鲁棒性测试的实验结果。每个图表表示机体框架中的滚转角（左上）、RD（右上）以及后右（RR）腿在机体框架中的 y 方向步态残差（底部）。黑色虚线标记了施加载荷的时刻，而红线表示扰动。

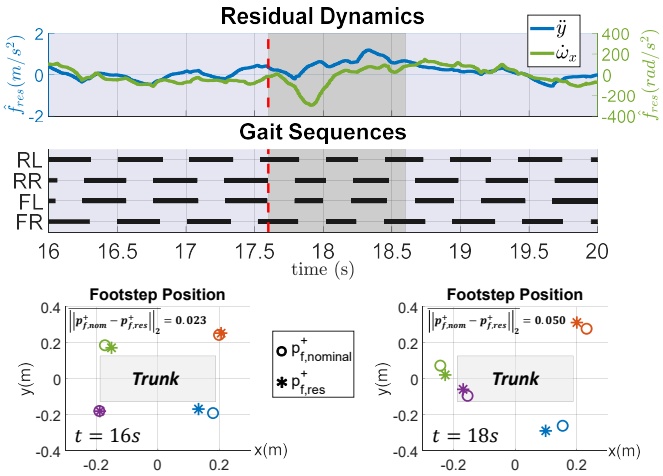

Fig. 4: 使用 res-all 进行鲁棒性测试的详细结果。在步态序列图中，黑线表示每条对应腿的支撑阶段，而灰色阴影区域表示在对机器人施加外部干扰后紧接着的 1 秒恢复期。足迹位置图显示了在机器人体框架内，自上而下视角下每个足迹的位置，其中图中的灰色矩形表示机器人的躯干。

B. 残差模块的效果

我们首先研究每个残差模块的效果。在这种情况下，机器人被命令保持其姿态，同时在其头部施加外部负载 6 kg，采用以下基线：标称的 MPC（vanilla-MPC），仅带有 RD 模块的 MPC（res-dyn），以及同时具有残差步态和动态模块的 MPC（res-all）。由于负载的施加距离质心较远，预期会有显著的模型误差。如果机器人成功适应给定的负载，那么它将受到外部扰动的影响。

表 V 和图 3 展示了结果。每个条件（正常、有负载、有干扰）的时间间隔在图 3 中标出。如表 V 所示，vanilla-MPC 在给定的负载下未能保持其姿态。

相反，res-dyn 允许机器人在负载施加后补偿模型的差异。然而，随着时间的推移，机器人表现出矢状面的摇摆运动。由于步态模式保持固定，它无法适应由此产生的运动，导致在施加额外干扰时失败。

另一方面，res-all 成功地控制了重载和外部扰动下的运动。如图 3 所示，每当施加扰动时，RD 模块都会捕获模型差异，而残差步伐模块则会调整落脚点以抵消引起的摇摆运动。

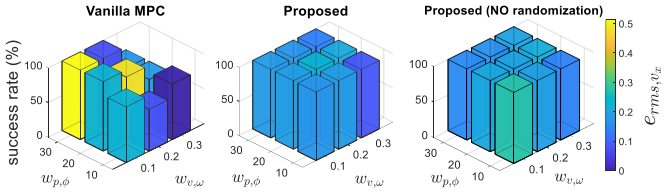

Fig. 5: 控制参数鲁棒性测试结果。每个图表显示了从对应的参数设置中速度跟踪的成功率（高度）和均方根误差（颜色），使用 (1) vanilla-MPC，(2) 提出的方案，和 (3) 模拟中无随机化的提出方案。每个权重标签表示所有对应的权重（例如，$w_{p,\phi} = [w_{p_{x,y,z}}, w_{\phi_{x,y,z}}]$）共享相同的值。

图 4 描述了来自 res-all 的详细结果。没有外部干扰时，残差步伐模块的落脚点和阶段保持在启发式算法附近。然而，当干扰产生了横滚方向上的角动量时，该模块会减少站立时间并调整落脚点以稳定运动。

所提出的方法还在相对高速运动下进行了评估，此时机器人在 x 方向上的加速度达到 $1m/s$。在该实验中，观测到在向前加速期间步态阶段的减少，如补充视频中所示。

C. 对控制参数的鲁棒性

众所周知，MPC 的性能对控制参数高度敏感，例如成本函数中的权重 [?]。尽管这些参数起着关键作用，但它们难以直接优化，并且通常通过经验选择来简化问题 [?]。类似于 RL 中的奖励工程，这导致了一个繁琐的调参过程。在本实验中，我们研究残差模块是否能够减少这种敏感性，从而在一系列参数配置中提高鲁棒性。

为了评估这个，机器人被分配在平坦地形上进行速度跟踪，并在不同的成本函数权重下进行。具体来说，在训练期间，（21）中的权重在 $w_{p,\phi} \in [10, 30]$ 和 $w_{v,\omega} \in [0.1, 0.3]$ 之间随机化。为了测试，我们设置 $w_{p,\phi} = [10, 20, 30]$、$w_{v,\omega} = [0.1, 0.2, 0.3]$。在所有情况下，$w_u$ 固定为 10^{-5}。其他系统参数按照表 I 中进行随机化。在每个设置中，我们运行 10 次模拟，每次 10 秒，并评估整个 10 s 时间窗口内成功率和 RMS 速度跟踪误差的平均值。

正如图 5 所示，vanilla-MPC 对权重表现出显著的性能敏感性，而我们的方法在所有配置中表现出一致的性能。这些结果表明，残差模块有效地降低了对手动调优的依赖，从而允许在更广泛的参数设置下成功执行任务。

此外，我们在训练过程中进行了一个没有权重随机化的实验。在这种情况下，权重固定为 $w_{p,\phi} = 10$，$w_{v,\omega} = 0.3$。图 5 显示，即使没有随机化，性能仍然保持相对一致。

D. 通过 MPC 实现一致性能

传统的端到端强化学习方法通常难以应对偏离其训练分布的情况 [?], [?]。相比之下，许多基于模型的控制器能够在广泛的状态范围内产生稳定的运动。我们旨在结合 MBA 的一致性与基于 RL 的方法的鲁棒性，并验证我们的框架是否比现有的方法提供了更好的一致性。

为了比较，我们设计了一个端到端的 RL 控制器（baseline-RL）作为基线 [?]。该框架同时学习一个状态估计器，以预测间接状态，例如平移速度和接触概率，以

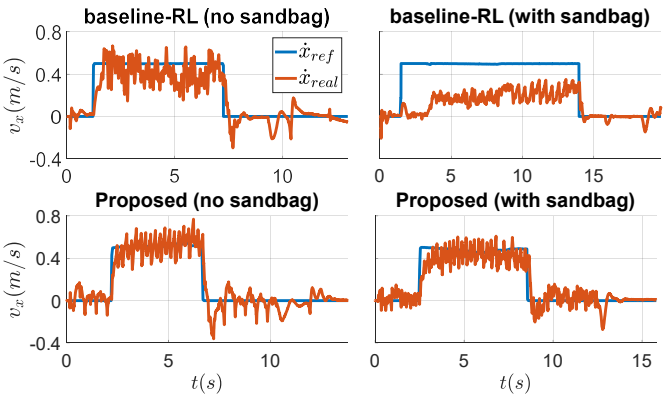

Fig. 6: 一个分布外测试的实验结果。

及控制策略。该策略在表格 III 中相同的条件和网络结构下使用 PPO 进行训练。

然后将基线与我们的方法进行比较。在这两种情况下，机器人都在如表 I 所示的领域内进行了训练。请注意，在模拟中，负载只施加在机器人的躯干上。在实验中，3 kg 负载被附加到右侧的前后腿上，创造了一个超出模拟的 OOD 情境。机器人被指令在保持姿态的同时遵循给定的速度指令。

实验结果如图 6 所示。在没有任何负载的情况下，baseline-RL 和 res-all 都成功地跟踪了给定的速度命令。然而，baseline-RL 的速度跟踪性能在有外部负载的情况下显著下降。相反，所提出的方法在没有明显退化的情况下保持了性能。这些结果表明，将 MBA 与剩余模块相结合可以增强鲁棒性，并在训练域之外的不确定性环境中实现可靠的运动。

E. 与基准的比较分析

我们提出的方法将不同类型的残差模块整合到名义控制器中。为了分析每个模块对性能的贡献，我们进行了基于基线方法的消融研究。

在模拟中，机器人被指令穿越地形，这些地形通过 Perlin 噪声程序生成，具有不同的粗糙程度。粗糙度级别 h_{env} 定义为地形的最大高度差。在每次模拟中，机器人被指令遵循给定的命令 $v_x^* = 1.0\ m/s$, $\omega_z^* = 1.0\ rad/s$ 。每次训练均按照表格 I 中的设置进行，除了 $M_{payload} = [0.9, 1.8]M$, $h_{env} = [0, 0.2]\ m$ 。测试环境分别由 $h_{env} = 0.2\ m$ 和 $M_{payload} = [1.75, 2, 2.25]M$ 组成。我们将成功率和速度跟踪误差的均方根作为标准，并对每种环境条件和控制器的组合进行 100 次模拟。

在消融研究中，我们评估了总共 8 种控制器配置，每种配置代表残差模块的不同组合。例如，fpos-phase 仅补偿步态位置和相位。基线包括 Chen 等人提供的方法 [?] ，该方法使用 RL 在关节空间（jpos）和动力学空间（dynRL）中寻找辅助动作；Yang 等人提供的方法 [?] ，他们提出了一个带有学习步态转换模块（phase）的分层框架；以及来自 [?] 的在线调整策略，该策略在滑动窗口内回归模型差异（resdyn-window）。

如表 VII 所总结，我们的方法在不同的环境条件下始终保持较高的成功率和稳定的跟踪性能，而大多数基线方法的性能则随着不确定性的程度而波动较大。

分析揭示了基线模型的特定弱点。例如，fpos-phase 的成功率在大不确定性下急剧下降，这突显了动态补偿的必要性。尽管 fpos-phase-dynRL 表现与我们的方法相当，但我们的框架展示了更高的学习效率，这一点将在后面讨论。相反，baseline-RL 采用了过于保守的策略，在高不确定性下拒绝执行命令以避免摔倒。此外，phase 和 resdyn-window 的差劲表现突显了在脚步和动态方面补偿不准确性对于稳健性的关键性。

最后，我们检查了每个残差模块对训练期间学习效率的影响。如图 7 所示，我们提出的方法不仅实现了最高的跟踪奖励，还表现出稳定和最快的收敛。这表明，将残差模块与标称 MBA 适当组合可以导致更精细的运动，从而实现更高效的样本学习。

TABLE VI: 比较分析结果

Name	$M_p = 1.75M$	$M_p = 2M$	$M_p = 2.25M$
res-all (proposed)	0.195 (76 %)	0.185 (74 %)	0.174 (58 %)
fpos-phase-dynRL	0.200 (88 %)	0.207 (72 %)	0.196 (57 %)
jpos-dynRL	0.185 (64 %)	0.189 (55 %)	0.194 (3 %)
fpos-phase	0.201 (71 %)	0.220 (72 %)	- (0 %)
phase	0.277 (39 %)	0.306 (11 %)	0.312 (1 %)
baseline-RL	0.805 (59 %)	0.766 (40 %)	0.800 (28 %)
vanilla-MPC	0.314 (41 %)	0.320 (25 %)	- (0 %)
resdyn-window	0.309 (39 %)	0.293 (30 %)	0.282 (14 %)

TABLE VII:

值表示 “RMS 速度误差（成功率）”。最低的 RMS 和最高的成功率以粗体标出。

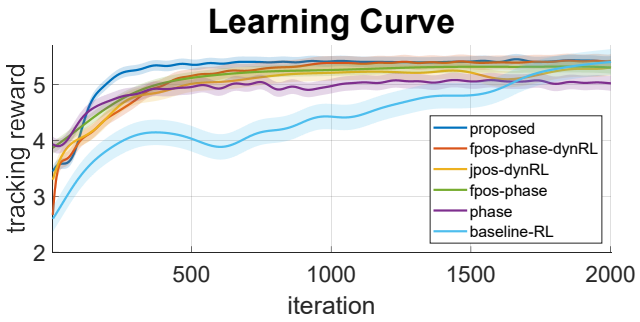

Fig. 7: 学习效率的对比结果。

V. 结论

总之，我们提出了一种新颖的混合框架，该框架结合了 MBAs 和 LBAs 的优势。设计残差模块是为了弥补传统 MBA 中每个部分启发式方法的局限性。通过模拟和硬件实验验证了所提出框架的优势。

我们观察到我们的框架可以产生更适应性的运动，同时仍然提供一致的性能，甚至超出训练域。此外，我们的方法可以缓解标称 MBA 的超参数敏感性。此外，我们的方法相比基线显示出更高的学习效率。

然而，为了完全验证我们方法的广泛适用性，有必要在各种条件下进行更广泛的评估。我们预计，通过在更多多样化领域中的进一步模块设计和训练，所提出的框架可以扩展以处理更广泛的不确定性，例如诸如柔软性、服从性和滑溜性等不确定的地形特性。