



本科生毕业设计(论文)

题 目: 基于 MPC 的四足机器人

运动控制

作 者: 张浩

学 号: U202143283

学 院: 机械工程学院

专 业: 机器人工程（科创班）

成 绩: _____

2025 年 05 月

本科生毕业设计(论文)

题 目: 基于 MPC 的四足机器人

运动控制

英文题目: Model Predictive Control-based

quadruped robot locomotion

学 院: 机械工程学院

班 级: 科创 21

学 生: 张浩

学 号: U202143283

指导教师: 李海泉 职称: 讲师

指导教师: 职称:

声 明

本人郑重声明：所呈交的论文是本人在指导教师的指导下进行的研究工作及取得研究结果。论文在引用他人已经发表或撰写的研究成果时，已经作了明确的标识；除此之外，论文中不包括其他人已经发表或撰写的研究成果，均为独立完成。其他同志对本文所做的任何贡献均已在论文中做了明确的说明并表达了谢意。

学生签名： 张浩 2025年6月9日

导师签名： 李海和 2025年6月9日

毕业设计(论文)任务书

- 一、学生姓名：张浩 学号：U202143283
二、题目：基于 MPC 的四足机器人运动控制
三、题目来源：真实
四、结业方式：论文
五、主要内容：
1. 了解四足机器人运动控制方法的发展现状与模型预测控制在四足机器人中的应用。
2. 建立四足机器人运动学动力学模型及状态估计器，实时获取机器人状态信息。
3. 构建 MPC 状态空间方程，分析行走约束条件；设计支撑腿力控制器。
4. 搭建仿真测试平台，验证控制算法有效性，完成实机部署与实验。

六、主要（技术）要求：

1. 最大前进速度不小于 1.5m/s，具备 2 种步态。
2. 四足机器人控制算法具备一定的地形适应能力，可以适应 10° 斜坡行走。
3. 完成机器人实机测试，优化控制参数，实现稳定可靠的步态切换。
4. 完成学校和学院统一规定的选题报告、中期检查和毕业论文等工作量。

七、日程安排：

- 第 1-4 周：调研国内外文献资料，翻译指定的外文文献，撰写选题报告。
第 5-6 周：建立系统动力学数学模型，规划步态等基本算法。
第 7-9 周：开展仿真实验并分析其运动效果。
第 10-12 周：完成实机控制算法的迁移部署。
第 13-15 周：撰写毕业论文，完成毕业答辩。

八、主要参考文献和书目：

1. 朱秋国.面向复杂场景的智能电力巡检四足机器人.浙江省,杭州云深处科技有限公司,2023-08-19.
2. 杭州宇树科技. 四足机器人公司介绍 - 宇树科技[EB/OL]. (2024-03-01) [2024-03-10]. <https://www.unitree.com/cn/about/>.
3. Bledt, G., Powell M.J., Katz B., et al. MIT Cheetah 3: Design and control of a robust, dynamic quadruped robot[C]. Institute of Electrical and Electronics Engineers. 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems(IROS), October 01-05, 2018, Madrid, Spain: IEEE, 2018: 2245-2252.
4. Di Carlo J., Wensing PM., Katz B., et al. Dynamic locomotion in the MIT Cheetah 3 through convex model-predictive control[C]. Institute of Electrical and Electronics Engineers. 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems(IROS), October 01-05, 2018, Madrid, Spain: IEEE, 2018: 1-9.
5. 谢惠祥.四足机器人对角小跑步态虚拟模型直觉控制方法研究[D].国防科学技术大学,2015.
6. 沈雅阁.针对复杂地形的四足机器人状态估计和运动控制算法研究与应用[D].电子科技大学,2022.
7. Raibert M. Legged Robots That Balance[M]. Cambridge: The MIT Press, 1986.
8. 王康.四足机器人复杂地形越障控制方法研究[D].北京交通大学,2023.
9. 龙俊峰, 于文烨, 李全意, 等. Learning H-Infinity Locomotion Control[EB/OL]. 2024[2025-02-26]. <https://arxiv.org/abs/2404.14405>.
10. 王旭.四足机器人运动及稳定控制关键技术综述[J/OL].吉林大学学报(工学版),2024[2025-02-26]. <https://doi.org/10.13229/j.cnki.jdxbgxb.20240722>.

指导教师签字: 李海泉 2025 年 2 月 22 日

学 生 签 字: 张浩 2025 年 2 月 22 日

系 (所) 负责人章: 李瑞 2025 年 2 月 27 日

摘要

四足机器人凭借其卓越的灵活性和复杂地形适应能力成为科研热点，但其高自由度和欠驱动特性使运动控制成为制约发展的关键。为此，本文以模型预测控制（MPC）为核心，围绕建模、控制理论、软硬件系统设计、仿真验证和实机测试展开深入研究，主要内容如下：

(1) 运动学建模。采用递推法求正运动学、几何法求逆运动学，建立足端位置与关节角度映射。正运动学求导获得雅可比矩阵，实现关节角速度与足端线速度的映射，并利用虚功原理推导出关节力矩与足端力关系。

(2) 状态估计器。设计基于离散卡尔曼滤波的状态估计器，融合 IMU 与关节编码器数据，估计机身位置和速度，为后续摆动腿和支撑腿控制提供状态信息。同时结合坡度估计，实现了自适应姿态调整。

(3) 常用步态与摆动腿控制器。介绍常用步态，构建基于时间的步态调度器生成步态相位与触地状态。在此基础上，规划了落足点和摆动腿的摆线轨迹。

(4) 动力学建模与 MPC 支撑腿控制器。分析支撑腿受力，建立单刚体模型和离散状态空间方程。设计期望轨迹与行走约束，分析了四足机器人行走约束条件，将 MPC 问题转化为二次规划（QP）问题，实时求解最优地面反力。

(5) 仿真验证与实机实验。在 MUJOCO 平台上，通过原地旋转、直线行走和斜坡运动验证控制逻辑；实际平台上进行状态估计、高速奔跑和台阶斜坡试验，结果表明控制方法具有良好的自适应性和鲁棒性。

总体而言，本文提出的控制系统实现了四足机器人的行走运动控制，为实际应用提供了坚实的理论依据和重要参考。

关键词： 四足机器人，运动学建模，状态估计，摆动腿规划，模型预测控制

Model Predictive Control-based quadruped robot locomotion

Abstract

Quadruped robots have emerged as a research hotspot thanks to their exceptional agility and ability to adapt to complex terrains. However, its high degrees of freedom and underactuated characteristics make motion control a critical challenge that hinders further development. To address this, this paper centers on Model Predictive Control (MPC) and conducts in-depth research on modeling, control theory, hardware and software system design, simulation validation, and practical testing. The main contents are as follows:

(1) Kinematic Modeling and Analysis. Recursive methods are used for forward kinematics and geometric methods for inverse kinematics to establish the mapping between foot position and joint angles. Differentiation of the forward kinematics yields the Jacobian matrix, realizing the mapping between joint angular velocities and foot linear velocities, while the virtual work principle is applied to derive the relationship between joint torques and foot forces.

(2) State Estimator. A state estimator based on discrete Kalman filtering is designed to fuse IMU and joint encoder data, thereby estimating the body position and velocity to provide state information for subsequent swing and support leg control. Additionally, slope estimation is integrated to achieve adaptive attitude adjustment.

(3) Common Gait and Swing Leg Controller. Common gaits are introduced, and a time-based gait scheduler is constructed to generate gait phases and ground contact states. On this basis, foot placement and the swing leg cycloid trajectory are planned.

(4) Dynamic Modeling and MPC Support Leg Controller. The forces acting on the support legs are analyzed, and a single rigid body model along with discrete state-space equations is established. Desired trajectories and walking constraints are designed, and the walking constraints of the quadruped robot are analyzed. The MPC problem is then transformed into a quadratic programming (QP) problem to solve for the optimal ground reaction force in real time.

(5) Simulation Validation and Practical Experiments. On the MUJOCO platform, the control logic is validated through in-place rotation, straight-line

walking, and slope motions; on the actual platform, state estimation, high-speed running, and stair/slope tests are conducted. The results indicate that the control method exhibits good adaptability and robustness.

Overall, the proposed control system achieves effective walking motion control for quadruped robots, providing a solid theoretical basis and an important reference for practical applications.

Key Words: **Quadruped Robot, Kinematic Modeling, State Estimation, Swing Leg Planning, Model Predictive Control**

目 录

摘要	III
Abstract	IV
1 引言	1
1.1 课题背景	1
1.2 研究意义	2
2 文献综述	3
2.1 四足机器人发展概述	3
2.1.1 国外研究现状	3
2.1.2 国内研究现状	5
2.2 四足机器人控制技术发展概述	7
3 系统建模及摆动腿控制器	10
3.1 整体结构	10
3.2 运动学建模	11
3.2.1 单腿正运动学	11
3.2.2 单腿逆运动学	12
3.2.3 一阶微分运动学	13
3.3 状态估计器	14
3.3.1 旋转矩阵	15
3.3.2 状态方程	16
3.3.3 观测方程	17
3.3.4 误差分析	19
3.3.5 状态估计	21
3.3.6 坡度估计	22
3.4 常用步态和摆动腿控制器	23
3.4.1 常用步态	23
3.4.2 步态调度器	25
3.4.3 落足点规划	25
3.4.4 摆动腿轨迹	28
4 基于 MPC 的支撑腿控制器	29
4.1 模型预测控制基本原理	29
4.2 单刚体动力学模型构建	30
4.2.1 单刚体动力学	31

4.2.2 近似角速度动力学	33
4.2.3 离散动力学状态空间方程	34
4.3 模型预测控制器	35
4.3.1 预测方程	35
4.3.2 期望轨迹	36
4.3.3 约束条件	38
4.3.4 二次规划	39
5 控制系统软件设计及仿真实验	41
5.1 控制系统软件设计	41
5.1.1 软件架构	41
5.1.2 参数设定	43
5.2 控制系统仿真实验验证	44
5.2.1 仿真平台简介	45
5.2.2 仿真原地旋转实验	45
5.2.3 仿真直线行走实验	46
5.2.4 仿真上下斜坡实验	47
6 控制系统硬件设计及实机实验	49
6.1 控制系统硬件设计	49
6.1.1 电气方案	49
6.1.2 硬件设备	50
6.2 控制系统实机实验验证	53
6.2.1 实机状态估计实验	53
6.2.2 实机高速奔跑实验	54
6.2.3 实机台阶斜坡实验	55
7 结 论	58
7.1 主要结论	58
7.2 工作展望	58
参考文献	59
在学取得成果	62
致 谢	64

1 引 言

1.1 课题背景

随着机器人技术的迅速发展，移动机器人在人类社会中扮演着日益重要的角色。在众多移动机器人中，四足机器人因其独特的运动方式和卓越的环境适应能力，成为当前机器人领域最具发展潜力的研究方向之一。与轮式和履带式机器人相比，四足机器人通过模仿自然界中四足动物的运动模式，能够在复杂、不规则的地形中灵活运动，展现出极强的越障能力和环境适应性。此外，四足机器人还具有出色的动态平衡性和运动稳定性，能够适应多种复杂的环境，这使得它在特殊工作场景中，尤其是高风险或困难的任务中，拥有独特的优势。

四足机器人在工业和民用领域的应用场景日益丰富。在工业应用方面，四足机器人可以执行如危险环境巡检、地下管道检查、矿山探测等任务，其灵活的运动能力使其能够替代人类工作者进入危险或狭窄的空间进行作业。在紧急救援方面，四足机器人的出色越障能力使其能够在地震、火灾等灾难现场进行搜救，能够通过配备的传感器和机械臂完成探测、救援等任务。在军事领域方面，四足机器人可以作为运输载具或侦察平台，为部队提供后勤支援和战场信息。在民用领域方面，四足机器人逐渐融入教育、娱乐及家庭服务等多个场景，四足机器人作为 STEAM 教育的理想平台，能够激发学生对机器人技术的兴趣。然而，尽管四足机器人在这些领域有广泛的应用前景，要在这些复杂场景中实现稳定可靠的运动控制，仍面临诸多技术挑战。

四足机器人的运动控制方法不断发展，其中常见策略包括弹簧倒立摆(Spring- Loaded Inverted Pendulum, SLIP)将腿部简化为弹簧 - 质量模型，便于能量分析，但难以处理多支撑切换和复杂地形；虚拟模型控制(Virtual Model Control, VMC)通过虚拟力映射，直观易调、计算量小，却忽略非线性与摩擦，难以支撑高速步态；基于二次规划(Quadratic Programming, QP)的方法可一体化优化运动与力控，灵活处理多种任务优先级，但在线求解开销大、实时性受限；多任务全身控制(Whole Body Control, WBC)通过层次化调度全身自由度，兼顾平衡、轨迹和力控，鲁棒性强，却高度依赖模型精度和参数调谐^[1]。尽管这些方法在计算效率和结构设计上各具优势，但在复杂地形或动态环境下，仍然存在适应性挑战。

近年来，随着强化学习(Reinforcement Learning, RL)技术的进步，基于学习的控制方法逐渐成为研究热点。通过在虚拟环境中进行大量训练，RL 使四足机器人能够自主学习复杂的运动策略，提升其对环境的适应能力。研究人员利用强化学习，使机器人掌握爬楼梯，斜坡等高难度动作。例如，Long 等^[2]提出混合内部模型方法，通过结合仿真机器人的响应与强化学习，实现灵活的运动控制，并进一步引入 H-Infinity 控制框架，增强机器人对外部扰动的抵抗能力。然而，尽管 RL 展现出强大潜力，其实际应用仍面临训练效率低、泛化能力受限以及安全性难以保障等挑战。

与此同时，随着计算硬件性能的提升和优化算法的进步，基于模型预测控制(Model Predictive Control, MPC)的方法已经相对成熟。MPC 通过在线优化求解，能够同时考虑系统动力学约束和环境约束，实现对未来状态的预测和优化控制，对复杂地形有较好的适应性^[3]。相比于强化学习方法，MPC 具有模型透明、可解释性强、约束处理自然等优势，且能够保证一定程度的控制安全性，因此更适合于四足机器人的实际应用。

1.2 研究意义

四足机器人作为仿生机器人领域的重要分支，不仅具有深远的科学价值，还在工程应用方面展现出巨大潜力。从学术角度来看，其运动控制涉及机械学、控制理论和生物力学等多个学科，涵盖了多体动力学建模、非线性系统控制以及生物运动机理等复杂问题。深入研究这些问题不仅能促进相关学科的发展，还能为其他类型的机器人控制提供有益参考。

在复杂地形适应性方面，四足机器人面临诸多挑战。运动规划需同时考虑动力学特性和环境约束，涉及高维非线性优化的实时求解。地形感知与建模则要求传感器数据与动力学模型紧密结合，为跨学科研究提供了新的探索方向。同时，控制策略的优化也推动了先进控制理论在实际系统中的应用。

本研究旨在设计基于 MPC 的四足机器人运动控制器，以验证机器人在复杂环境下实现稳定、灵活运动的可行性。相较于传统控制方法，MPC 能够综合考虑运动约束和环境信息，通过实时优化提前规划合理的控制策略，从而提升机器人在动态环境中的响应能力和安全性。

2 文献综述

2.1 四足机器人发展概述

四足机器人自 20 世纪 70 年代以来不断发展：1980 年代，东京工业大学的 TITAN 系列开创了静态步态控制；90 年代，通用电气的 ASV 首次在野外实现自主导航；进入 21 世纪后，MIT 的 Cheetah 系列在动态运动控制方面取得显著突破^[3]，波士顿动力于 2015 年发布的 Spot 机器人以其卓越性能成为工业领域标杆；国内则有宇树科技和云深处等公司不断推出性能优异的四足机器人，推动了技术进步与产业化^[4, 5]。

2.1.1 国外研究现状

Boston Dynamics 在四足机器人研究方面取得了重要突破，推出了 Big Dog 和 Spot Mini 等代表性机器人。这些机器人具备卓越的机动性和环境适应能力，在复杂地形中能够稳定行走，为四足机器人技术的发展提供了重要参考。

Big Dog: 如图 2-1(a)所示，BigDog 是一款液压驱动的四足机器人，尺寸为长 1.1m、宽 0.3m、高 1m，重 109kg，负载极限为 154kg，最高移动速度可达 6.4 公里/小时。BigDog 配备了 16 个液压执行器和多种传感器(如力传感器、位置传感器及陀螺仪)，能够实时监测足端力、关节角度及机身姿态。其采用 Trot 步态在砂石、山地及冰雪等恶劣路况下行进，并可通过 Bound 步态跳越沟壑。路径规划方面，BigDog 使用改进的 A*算法，并通过样条算法对规划路径进行平滑处理^[6]。

Spot Mini: 如图 2-1(b)所示，Boston Dynamics 于 2018 年推出的纯电动四足机器人 Spot Mini 搭载了激光雷达和视觉传感器，能够实现爬楼梯、开门等动作，并具备室内自主定位与导航能力^[7]。



图 2-1 Boston Dynamics 系列机器人

麻省理工学院(MIT)的 Sangbae Kim 团队在四足机器人领域做出了卓越的贡献，以下是该团队研究情况的简述：

Cheetah 2: 如图 2-2(a)所示，Cheetah 2 配备激光测距仪，能够以 Bound 步态达到 6.4m/s 的速度^[8]，并自主跳越 40cm 高的障碍物。其控制算法包括接近规划与跳跃优化，采用模型预测控制(MPC)调整水平速度，并通过非线性轨迹优化计算跳跃所需的地面反力和身体轨迹，以规划适应障碍物的跳跃动作^[9]。

Cheetah 3: 如图 2-2(b)所示，Cheetah 3 改进了腿部结构设计，增强了在复杂地形中的适应能力，即使在无视觉系统的情况下完成了上楼梯等动作^[10]。提出了一种结合轨迹优化、高频跟踪控制器及鲁棒着陆控制器的方法，使机器人能够跳上和跳下 0.76m 高的桌子^[11]。

Mini Cheetah: 如图 2-2(c)所示，Mini Cheetah 作为更小型化的版本，重量减轻至 9kg ，站立高度降至 0.3m 。结合模型预测控制与全身控制技术，实现了 2.45m/s 的前进速度，并首次完成了四足机器人的后空翻。此外，开发了动态空中运动在线规划与执行的控制框架，支持旋转跳跃和翻转等复杂空中动作^[12]。

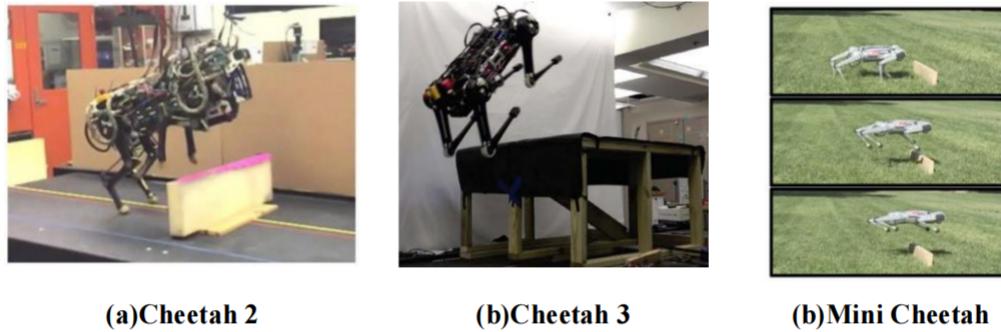


图 2-2 MIT 系列机器人

瑞士苏黎世联邦理工学院(ETH)的机器人系统实验室在 2016 年研制出 ANYmal 四足机器人，具有以下特点。

ANYmal B: 如图 2-3(a)所示，ANYmal B 站立高度 0.6m ，宽度 0.6m ，采用自研的关节模组，能够在奔跑或跳跃时承受较大冲击力，展现出优良的动态运动性能。搭载多种传感器，能够感知环境并创建三维地形地图，实现路径规划，支持爬楼梯和斜坡等功能，具有强大的抗干扰能力。ANYmal B 采用感知全身规划算法，能够在狭窄受限空间中运动，成功通过管道和低窄间隙等障碍物^[13]。

ANYmal C: 如图 2-3(b)所示，ETH 在强化学习方面进行了深入研究，

并将其成果应用于 ANYmal C，使机器人能够轻松应对具有挑战性的自然环境。该方法结合本体感受反馈，训练结果可直接泛化至真实机器人，保持在未见过的环境中的鲁棒性^[14]。



图 2-3 ETH 系列机器人

2.1.2 国内研究现状

宇树科技(Unitree)是国内专注于足式机器人研发的高科技企业，其产品如：GO2 和 B2 在消费级与工业级应用中展现出卓越的运动性能和智能感知能力。

GO2：如图 2-4(a)所示，GO2 配备自研 4D 激光雷达 L1，提供 $360^\circ \times 90^\circ$ 的半球形超广角感知能力，支持全地形感知，最小探测距离为 0.05m。能够执行水平跳远、垂直跳跃等高难度动作，具备越障能力，适用于家庭娱乐、教育研究和个人开发等领域。

B2：如图 2-4(b)所示，B2 专为工业应用设计，最大关节扭矩为 360N·m，能够轻松跳越 1.1m 的沟壑，并从 1m 高度安全下落。具备超强越障能力，能够跨越乱木堆、40cm 高台等各种障碍物。B2 搭载激光雷达、深度相机和高分辨率光学相机，适用于复杂多变的户外环境，如电力巡检和消防救援，展现出高耐用性和防护等级^[5]。



图 2-4 宇树科技(Unitree)系列机器人

杭州云深处科技(DEEPRobotics)自主研发的“绝影”系列机器人在电站、工厂、管廊巡检，以及应急救援、消防侦查、未来科研等多种应用环境中落地应用。

Lite 3：如图 2-5(a)所示，Lite 3 面向教育科研和科技爱好者，能够攀爬 15cm 台阶，完成水平跳远、垂直跳高及前空翻等高难度动作。搭载单目相机用于人体跟随识别，并可选配 3D 激光雷达和深度相机。

X30：如图 2-5(b)所示，X30 作为旗舰级产品，专为工厂巡检和消防应急救援设计^[4]，身高约 0.85m，体重约 40kg。前后各搭载 2 个 MID360 3D 激光雷达，具备先进的传感器系统和强大的融合感知能力，能够在多变环境中快速越过障碍物并自主导航复杂地形，即使在光线不足或完全黑暗的环境中，X30 亦能依靠其感知技术自主执行任务。



图 2-5 杭州云深处科技(DEEPRobotics)系列机器人

如图 2-6 所示，香港中文大学的宋志涛、乐林株等研究人员提出了一种能量最优的跳跃优化方法，该方法结合质心动力学和 kino-dynamic 约束。通过离线优化，他们预先生成并存储适用于不同障碍物的跳跃轨迹。机器人在执行跳跃任务时，轨迹选择器会根据用户指令或环境感知信息，自动匹配最优轨迹，并结合前馈优化力与关节空间 PD 反馈控制，实现精准跳跃。实验验证表明，该方法能够成功完成 30cm 高障碍物的跳跃^[15]。



图 2-6 香港中文大学四足机器人跳窗口

总体而言，国内外在四足机器人领域均取得了显著进展。国外的 Boston

Dynamics、MIT、ETH 等机构在动态控制上突出，国内 Unitree、DEEPRobotics 等企业在工业与消费市场表现突出，未来随着强化学习和 MPC 等技术进步，四足机器人智能化、自主性和环境适应能力将进一步提升并广泛应用于工业、服务和探索领域。

2.2 四足机器人控制技术发展概述

四足机器人作为仿生机器人研究的重要方向，其运动控制技术一直是学术界和工业界关注的核心问题。四足机器人的运动能力直接影响其在复杂地形中的适应性，尤其是在崎岖或不断变化的环境下，如何实现精准控制和稳定运动成为关键挑战。为提升其在不同地形条件下的性能，研究人员提出了多种控制策略和方法。根据控制策略的不同，当前主要的研究方法可归纳为以下几类：

(1) 基于弹簧倒立摆模型方法(SLIP)

1984 年，Raibert^[16]提出的 SLIP 模型将足式动物或机器人的腿部运动简化为带负载的无质量弹簧，适用于躯干质量远大于腿部质量的情境。该模型通过三个独立的控制方面实现跳跃：弹跳高度控制(调整虚拟腿推力)、前向速度控制(根据对称性原理规划落足点位置)及姿态控制(触地时通过髋关节扭矩调节)。由于其控制逻辑简洁、无需复杂建模且参数调整便捷，SLIP 模型被广泛应用于双足和四足机器人的设计中，支持多种步态模式。然而，该模型在静态或准静态运动控制以及质心位置与脚端轨迹的精确跟踪方面存在一定局限。

(2) 基于虚拟模型的控制方法(VMC)

1995 年，Pratt G^[17]提出的虚拟模型控制(VMC)足式机器人控制方法。VMC 是一种直观的运动控制方法，不依赖于精确的轨迹规划或强制性轨迹跟踪。VMC 通过比较机器人期望状态与实际状态，并运用简单规则调整机器人行为。其核心在于在机器人上附加虚拟机械构件(如阻尼器、弹簧)，以产生虚拟力。通过雅可比矩阵计算关节力矩，使机器人根据这些虚拟力进行运动。VMC 具有结构简单、直观性强、高稳定性、良好柔顺性及强抗干扰能力，已成功应用于 HYQ 等四足机器人中^[18]。

(3) 基于二次规划的控制方法(QP)

2017 年，Michele Focchi^[19]等人提出了一种准静态四足机器人行走控制框架，使机器人能够在最大 50° 坡度的 V 型槽内稳定行走。该框架的核心在于一种基于二次规划 (QP) 的足底反作用力分配方法。首先，通过 PD

控制对质心位置和机身姿态偏差进行校正，以生成所需的质心加速度和机身角加速度；接着，依据线性化的单刚体动力学模型构建 QP 问题，并在求解过程中纳入足底接触面摩擦锥约束及各足的支撑/离地状态。最终，通过优化计算出最优的足底反作用力分配，并利用雅可比矩阵推导所需的关节驱动力矩。这一平衡控制方法也在 MIT Cheetah 3 机器人上成功应用。

(4) 基于多任务全身控制的方法(WBC)

将四足机器人整体控制任务分为四个小任务，按照优先级从高到低分别为：支撑腿轨迹跟随任务、机身转动控制任务、机身平动控制任务、摆动腿足底轨迹跟随任务。多任务全身控制(WBC)利用雅可比矩阵及其零空间对多任务进行整合，确保在满足高优先级任务的同时尽可能地实现低优先级任务^[20]。在 WBC 的基础上，结合模型预测控制(MPC)进一步增强了四足机器人的动态能力。MPC 通过预测未来状态并优化控制输入，计算最优地面反作用力，而 WBC 依托这些反作用力，通过动力学模型求解关节的目标位置、速度、加速度及力矩指令。这种协同控制策略已在 MIT 的 Mini Cheetah 机器人上取得显著成效，使最高速度从 2018 年的 2.45m/s 提升至 2019 年的 3.7m/s。MPC 与 WBC 相结合的控制架构充分发挥了两者优势，使四足机器人在复杂环境中具备更稳定、更高效的运动能力。

(5) 基于模型预测控制的方法(MPC)

2018 年，麻省理工学院(MIT)的 Jared Di Carlo 等人提出了一种基于简化动力学的 MPC 方法，用于计算四足机器人的地面反作用力。该方法通过将复杂的机器人动力学简化为凸优化问题，同时保留系统的完整三维特性，使计算更加高效。在 MIT Cheetah 3 机器人上的实验表明，该方法使机器人实现了最高 3m/s 的前进速度、1m/s 的横向速度和 180° /s 的角速度，并成功执行站立、小跑、飞跑、快步和跳跃等多种运动模式。值得注意的是，该方法通用性强，仅需使用相同的增益和权重参数，即可在不同运动模式下保持稳定，展现出良好的适应性和鲁棒性。

该方法的实现包括三个主要步骤。首先，建立动力学模型并分析受力情况，推导出机器人各部件的状态方程。其次，基于状态方程构建预测模型，并将控制问题转化为优化求解，以获得最优地面反作用力。最后，为确保实时性，研究人员将优化问题构建为二次规划(QP)形式，以实现高效计算。然而，MPC 需要在线求解大规模优化问题，对计算资源要求较高，限制了控制带宽^[21]。为解决这一问题，研究者们持续优化算法，提高计算效率，以满足实时控制的需求。

(6) 基于强化学习的控制方法(RL)

基于强化学习(RL)的四足机器人控制方法，使机器人能够通过与环境交互，自主学习最优行为策略。在这一过程中，机器人执行特定动作，并根据环境反馈的奖励或惩罚不断调整策略，以最大化累积奖励。常见的 RL 算法，如深度 Q 网络(Deep Q-Network, DQN)、策略梯度(Policy Gradient)和近端策略优化(Proximal Policy Optimization, PPO)，使机器人能够从零开始学习复杂运动技能，具备较强的自适应性和泛化能力，即便在陌生环境中也能表现出良好的适应性。

然而，RL 控制方法也面临诸多挑战。首先，训练过程通常需要大量计算资源和时间，尤其在大规模训练或复杂环境中，训练效率较低。其次，尽管采用了域随机化等方法来增强策略的鲁棒性，仿真环境与真实环境的差异仍可能影响策略的稳定性和有效性。此外，RL 模型的结构和参数往往缺乏直观的物理解释，例如四足机器人的腿长、质量或电机特性等微小变化，可能导致训练好的模型在实际应用中完全失效。这使得强化学习的模型不具备良好的可解释性和通用性，限制了其在某些场景中的应用^[22]。

如表 2-1 所示，系统的比较各类四足机器人运动控制方法的优缺点及其应用现状。通过对不同控制方法的特性进行分析，可更直观地了解各方方法在实际应用中的适用性、优势及其局限性。

表 2-1 四足机器人控制方法的优缺点对比

控制方法	名称	优点	缺点
Spring loaded inverted pendulum	弹簧倒立摆(SLIP)	建模简单，对建模精度要求不高。	对腿部运动状态和受力一致性要求高。
Virtual model control	虚拟模型控制(VMC)	简单直观，不涉及动力学内容。	对虚拟模型的选择比较重要。
Quadratic Programming	二次规划(QP)	求解效率高，适用于实时系统。	性能依赖于准确的动力学建模。
Whole-body control	全身控制(WBC)	能够实现具有优先级的多任务运动。	计算量大，控制复杂。
Model predictive control	模型预测控制(MPC)	能够预测和优化，能适应复杂系统。	需要平衡模型准确度和计算能力。
Reinforcement Learning control	强化学习(RL)	自主学习能力强，无须标记大量数据，能够优化长期目标。	训练时间长，奖励设计困难，训练时面临稳定性和收敛性问题。

因此，即使基于强化学习的控制算法具有巨大的潜力，本文还是以研究基于数值优化的四足机器人模型预测控制(MPC)算法为主。

3 系统建模及摆动腿控制器

3.1 整体结构

本文的研究对象是由北京科技大学机器人实验室自主设计的四足机器人，如图 3-1 所示。



图 3-1 四足机器人

该四足机器人尺寸、质量等物理参数如表 3-1 所示。

表 3-1 四足机器人物理参数

符号	含义	数值
L_b	前后髋关节间距(m)	0.414m
W_b	左右髋关节间距(m)	0.1m
l_0	髋关节连杆(m)	0.08m
l_1	大腿(m)	0.22m
l_2	小腿(m)	0.20m
h	站立高度(m)	0.31m
τ	最大关节扭矩(Nm)	23.7Nm
m	总质量(kg)	12.5kg

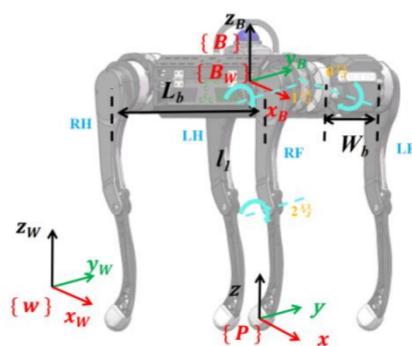


图 3-2 四足机器人坐标系

建立了如图 3-2 所示的坐标系和关节定义:

以机器人躯干质心为原点, 躯干正前方为 X 轴正方向, 坚直向上为 Z 轴正方向, 按右手定则建立了与躯干固连的机身坐标系 {B} 和与足端固连的足端坐标系 {P}。机器人四条腿分别记为左前腿 (LF)、右前腿 (RF)、左后腿 (LH) 和右后腿 (RH)。

为构建动力学模型, 还建立了固定于地面的世界坐标系 {W}。世界坐标系 X 轴水平朝前, Z 轴坚直向上, 其位置和姿态在运动过程中保持恒定。在机器人启动时刻, 机身坐标系 {B} 的初始姿态与世界坐标系 {W} 一致。

3.2 运动学建模

运动学建模的核心在于建立足端位置与关节角度以及足端速度与关节角速度之间的定量关系^[23]。

3.2.1 单腿正运动学

正运动学就是已知机器人 3 个关节角度 $\theta_0, \theta_1, \theta_2$, 求解足端 {P} 系相对于该腿 {0} 系的坐标的过程。如图 3-3 所示, 在此采用 T 矩阵递推法求解。以左前腿(LF)为例。

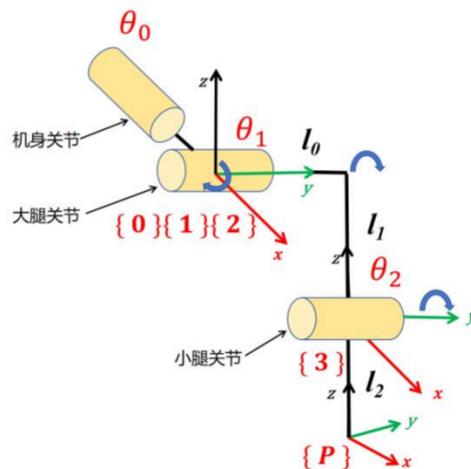


图 3-3 单腿正运动学坐标系

其中 {1} 系绕 {0} 系的 x 轴转动, 坐标变换矩阵 T_{01} 表示为:

$$T_{01} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta_0 & -\sin\theta_0 & 0 \\ 0 & \sin\theta_0 & \cos\theta_0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3-1)$$

{2} 系绕 {1} 系的 y 轴转动, 坐标变化矩阵 T_{12} 表示为:

$$T_{12} = \begin{bmatrix} \cos\theta_1 & 0 & \sin\theta_1 & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta_1 & 0 & \cos\theta_1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3-2)$$

{3} 系绕 {2} 系的 y 轴转动, 坐标变化矩阵 T_{23} 表示为:

$$T_{23} = \begin{bmatrix} \cos\theta_2 & 0 & \sin\theta_2 & 0 \\ 0 & 1 & 0 & l_0 \\ -\sin\theta_2 & 0 & \cos\theta_2 & -l_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3-3)$$

足端 {P} 系在 {3} 系中的坐标为常量, 坐标变化矩阵 T_{3P} 表示为:

$$T_{3P} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -l_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3-4)$$

所以足端 {P} 系到 {0} 系的坐标变化矩阵 T_{0P} 表示为:

$$T_{0P} = T_{01}T_{12}T_{23}T_{3P} = \begin{bmatrix} c_{12} & 0 & s_{12} & -l_1s_1 - l_2s_{12} \\ s_{12}s_0 & c_0 & -c_{12}s_0 & (l_1c_1 + l_2c_{12})s_0 + l_0c_0 \\ -c_0s_{12} & s_0 & c_{12}c_0 & -(l_1c_1 + l_2c_{12})c_0 + l_0s_0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3-5)$$

式中: $s_i = \sin\theta_i$, $c_i = \cos\theta_i$, $s_{ij} = \sin(\theta_i + \theta_j)$, $c_{ij} = \cos(\theta_i + \theta_j)$, $i, j = 0, 1, 2$; $i \neq j$, 后文中使用相同定义。

T 矩阵的第四列表示位置变换, 得到足端坐标系 {P} 相对于身体髋关节 {0} 系的位置, 即:

$$\begin{bmatrix} x_p \\ y_p \\ z_p \end{bmatrix} = \begin{bmatrix} -l_1\sin\theta_1 - l_2\sin(\theta_1 + \theta_2) \\ (l_1\cos\theta_1 + l_2\cos(\theta_1 + \theta_2))\sin\theta_0 + l_0\cos\theta_0 \\ -(l_1\cos\theta_1 + l_2\cos(\theta_1 + \theta_2))\cos\theta_0 + l_0\sin\theta_0 \end{bmatrix} \quad (3-6)$$

对于右前腿(RF)和右后腿(RH), 只需将式(3-6)中的 l_0 全部替换为 $-l_0$ 即可。

$$l_0 = \begin{cases} l_0, & i \text{ 为左腿} \\ -l_0, & i \text{ 为右腿} \end{cases} \quad (3-7)$$

至此即完成了正运动学的解算。

3.2.2 单腿逆运动学

逆运动学就是已知机器人足端 {P} 系相对于该腿 {0} 系的坐标, 求解 3 个关节角度 $\theta_0, \theta_1, \theta_2$ 的过程。在此采用几何法求解。以左前腿(LF)为例进行逆运动学建模。

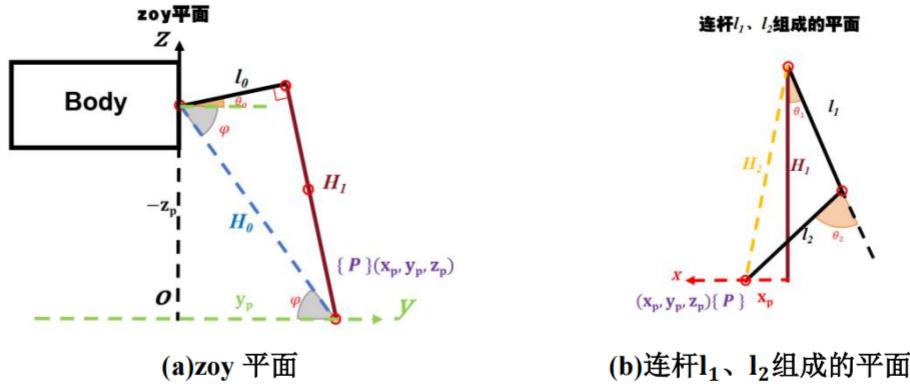


图 3-4 单腿逆运动学坐标系

如图 3-4(a)所示，在 zoy 平面内，由几何关系易得：

$$\begin{cases} H_0 = \sqrt{y_p^2 + z_p^2} \\ H_1 = \sqrt{H_0^2 - l_0^2} \end{cases} \quad (3-8)$$

$$\theta_0 = \arccos \frac{l_0}{H_0} - \text{atan}2(-z_p, y_p) \quad (3-9)$$

如图 3-4(b)所示，在由连杆 l_1 、 l_2 组成的平面内，由几何关系易得：

$$H_2 = \sqrt{H_1^2 + x_p^2} \quad (3-10)$$

求解 θ_2 ，由余弦定理得：

$$\theta_2 = \arccos \frac{l_1^2 + l_2^2 - H_2^2}{2l_1l_2} - \pi \quad (3-11)$$

求解 θ_1 ：

$$\theta_1 = \arccos \frac{l_1^2 + H_2^2 - l_2^2}{2l_1H_2} - \text{atan} \frac{x_p}{H_1} \quad (3-12)$$

对于右前腿(RF)和右后腿(RH)，只需将 l_0 全部替换为 $-l_0$ 即可。至此即完成了逆运动学的解算。

3.2.3 一阶微分运动学

运动学建立了关节角度与足端位置的关系，其一阶微分即为关节角速度与足端速度之间的关系。由于正向运动学公式较简单，本文采用直接对正运动学方程求导的方法，来求关节角速度与足端速度之间的雅可比矩阵。

对正向运动学的式(3-6)的左右两侧分别求时间 t 的一阶导数，整理后

可得:

$$\begin{cases} \dot{x}_P = J_{11} \cdot \dot{\theta}_1 + J_{12} \cdot \dot{\theta}_2 + J_{13} \cdot \dot{\theta}_3 \\ \dot{y}_P = J_{21} \cdot \dot{\theta}_1 + J_{22} \cdot \dot{\theta}_2 + J_{23} \cdot \dot{\theta}_3 \\ \dot{z}_P = J_{31} \cdot \dot{\theta}_1 + J_{32} \cdot \dot{\theta}_2 + J_{33} \cdot \dot{\theta}_3 \end{cases} \quad (3-13)$$

可以看出, 足端速度 \dot{x}_P 、 \dot{y}_P 、 \dot{z}_P , 和各个关节角速度 $\dot{\theta}_1$ 、 $\dot{\theta}_2$ 、 $\dot{\theta}_3$, 之间是线性关系, 因此整理为矩阵相乘的形式:

$$\begin{bmatrix} \dot{x}_P \\ \dot{y}_P \\ \dot{z}_P \end{bmatrix} = \begin{bmatrix} J_{11} & J_{12} & J_{13} \\ J_{21} & J_{22} & J_{23} \\ J_{31} & J_{32} & J_{33} \end{bmatrix} \cdot \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{bmatrix} = J \cdot \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{bmatrix} \quad (3-14)$$

左前腿和左后腿的雅可比矩阵结果:

$$J = \begin{bmatrix} 0 & -l_2 c_1 - l_3 c_{12} & -l_3 c_{12} \\ (l_2 c_1 + l_3 c_{12})c_0 - l_1 s_0 & -(l_2 s_1 + l_3 s_{12})s_0 & -l_3 s_{12}s_0 \\ (l_2 c_1 + l_3 c_{12})s_0 + l_1 c_0 & (l_2 s_1 + l_3 s_{12})c_0 & l_3 s_{23}c_0 \end{bmatrix} \quad (3-15)$$

对于右前腿(RF)和右后腿(RH)的雅可比矩阵, 只需将式(3-15)中的 l_0 全部替换为 $-l_0$ 即可。

在四足机器人的关节限位下, 雅可比矩阵 J 始终是可逆的。将式(3-14)的等式两侧同时左乘 J^{-1} , 得:

$$\begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{bmatrix} = J^{-1} \cdot \begin{bmatrix} \dot{x}_P \\ \dot{y}_P \\ \dot{z}_P \end{bmatrix} \quad (3-16)$$

通过雅可比矩阵, 还可以求解单腿静力学, 利用虚功原理可得:

$$\tau^T \delta q = F^T \delta x \quad (3-17)$$

其中, F 是作用在末端执行器上的空间作用力; δx 是产生的无穷小位移矢量; τ 为关节力矩; δq 为无穷小关节位移量。上式对时间微分, 可得到:

$$\tau^T \dot{q} = F^T v \quad (3-18)$$

将式(3-16)代入式(3-18)中, 并进一步化简可得到:

$$\tau = J^T F \quad (3-19)$$

由此, 通过雅可比矩阵建立了机器人静止状态下关节力矩和足端力之间的关系。

至此即完成了一阶微分运动学的解算。

3.3 状态估计器

由于无法直接测量四足机器人在世界坐标系下的位置和速度, 通常需要结合 IMU 读数和足端运动状态进行估计。假设支撑足与地面无相对滑动

(即足底速度为0)，便可在机体坐标系与世界坐标系之间建立速度映射，从而计算质心的速度和位置。为融合IMU与编码器提供的数据并充分利用系统状态间的线性关系，本文采用离散卡尔曼滤波器对足底里程计进行滤波与状态估计。

首先给出卡尔曼滤波器的状态方程和观测方程：

$$\begin{cases} x_k = Ax_{k-1} + Bu_{k-1} + w_{k-1} \\ y_k = Cx_k + v_k \end{cases} \quad (3-20)$$

其中的过程噪声 w 和测量噪声 v 都是零均值且不相关的白噪声，服从高斯分布，并且其协方差矩阵分别为 Q 和 R ：

$$w \sim (0, Q), v \sim (0, R) \quad (3-21)$$

3.3.1 旋转矩阵

状态估计器需要求解质心在世界坐标系{W}下的速度和位置，由于涉及多个坐标系之间的转换，因此首先需要推导从机体坐标系{B}到世界坐标系{W}的旋转矩阵 R_B^W 。

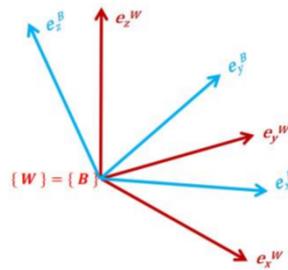


图 3-5 坐标系之间的旋转

如图3-5所示，两个坐标系W与B，其原点W与B在空间上重合。

考虑一点P，在坐标系W下的描述为：

$${}^W r_{WP} = [{}^W r_{WP}^x \quad {}^W r_{WP}^y \quad {}^W r_{WP}^z]^T \quad (3-22)$$

同理，点P在坐标系B下的描述为，

$${}^B r_{BP} = [{}^B r_{BP}^x \quad {}^B r_{BP}^y \quad {}^B r_{BP}^z]^T \quad (3-23)$$

记坐标系B中用于参数化描述向量的一组单位正交基在坐标系W中的描述为(${}^W e_x^B, {}^W e_y^B, {}^W e_z^B$)，则可以写出坐标向量 ${}^W r_{WP}$ 和 ${}^B r_{BP}$ 之间的映射关系：

$${}^W r_{WP} = {}^W e_x^B \cdot {}^B r_{BP}^x + {}^W e_y^B \cdot {}^B r_{BP}^y + {}^W e_z^B \cdot {}^B r_{BP}^z \quad (3-24)$$

将上式以紧凑的方式写为：

$${}^W r_{WP} = [{}^W e_x^B \quad {}^W e_y^B \quad {}^W e_z^B]^T r_{AP} = R_B^W {}^B r_{BP} \quad (3-25)$$

使用 ZYX 欧拉角 Θ 表示机器人的姿态, $\Theta = [\phi \theta \psi]^T$, 其中 ϕ 是横滚角, θ 是俯仰角, ψ 是偏航角。

绕 x 轴、y 轴、z 轴旋转的旋转矩阵分别为:

$$R_x(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c_\phi & -s_\phi \\ 0 & s_\phi & c_\phi \end{bmatrix}, \quad R_y(\theta) = \begin{bmatrix} c_\theta & 0 & s_\theta \\ 0 & 1 & 0 \\ -s_\theta & 0 & c_\theta \end{bmatrix}, \quad R_z(\psi) = \begin{bmatrix} c_\psi & -s_\psi & 0 \\ s_\psi & c_\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3-26)$$

为提高公式紧凑性, c_ψ 与 s_ψ 分别为 $\cos(\psi)$ 和 $\sin(\psi)$ 的简写。

当机器人从世界坐标系开始执行旋转变换时, 其旋转顺序为 ZYX: 首先绕 z 轴转动 ψ 角度, 然后绕经过第一次旋转后的 y 轴转动 θ 角度, 最后绕经过第二次旋转后的 x 轴转动 ϕ 角度。因此, 从机体坐标系 {B} 到世界坐标系 {W} 的旋转矩阵 R_B^W 可表示为:

$$\begin{aligned} R_B^W &= R_z(\psi)R_y(\theta)R_x(\phi) \\ &= \begin{bmatrix} c_\psi & -s_\psi & 0 \\ s_\psi & c_\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_\theta & 0 & s_\theta \\ 0 & 1 & 0 \\ -s_\theta & 0 & c_\theta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & c_\phi & -s_\phi \\ 0 & s_\phi & c_\phi \end{bmatrix} \\ &= \begin{bmatrix} c_\theta c_\psi & c_\psi s_\phi s_\theta - c_\phi s_\psi & s_\phi s_\psi + c_\phi c_\psi s_\theta \\ c_\theta s_\psi & c_\phi c_\psi + s_\phi s_\theta s_\psi & c_\phi s_\theta s_\psi - c_\psi s_\phi \\ -s_\theta & c_\theta s_\phi & c_\phi c_\theta \end{bmatrix} \end{aligned} \quad (3-27)$$

下面建立状态估计器需要的状态方程和观测方程。

3.3.2 状态方程

根据 Gerardo Bledt 等的研究, 此处使用的状态估计器能够估计机器人在世界坐标系 {W} 下的运动状态^[10]。即世界坐标系 {W} 下, 机身位置 ${}^W p_{com}$ 、机身速度 ${}^W v_{com}$, 以及四个足端的位置 ${}^W p_0$ 、 ${}^W p_1$ 、 ${}^W p_2$ 、 ${}^W p_3$ 。在本文中, 使用左上标 ${}^W x$ 来表示 x 变量处于世界坐标系 {W} 下, 左上标 ${}^B x$ 来表示 x 变量处于机身坐标系 {B} 下。

定义状态向量为: $x = [{}^W p_{com} \quad {}^W v_{com} \quad {}^W p_0 \quad {}^W p_1 \quad {}^W p_2 \quad {}^W p_3]^T$ 。

构建连续状态方程, 假设四个足端都与地面稳定接触, 则每个足端的速度都是 0。则状态向量的导数 \dot{x} 为:

$$\dot{x} = \begin{bmatrix} \dot{w_p}_{com} \\ \dot{w_v}_{com} \\ \dot{w_p}_0 \\ \dot{w_p}_1 \\ \dot{w_p}_2 \\ \dot{w_p}_3 \end{bmatrix} = \begin{bmatrix} w_{v_{com}} \\ R_B^W B a_{com} + g \\ 0_{3 \times 1} \\ 0_{3 \times 1} \\ 0_{3 \times 1} \\ 0_{3 \times 1} \end{bmatrix} \quad (3-28)$$

定义系统的输入为: $u = R_B^W B a_{com} + g$, 即世界坐标系下的机身加速度。则式(3-28)可以写为 $\dot{x} = A_c x + B_c u$ 的形式:

$$\dot{x} = \begin{bmatrix} 0_{3 \times 3} & I_3 & 0_{3 \times 12} \\ 0_{3 \times 18} & 0_{3 \times 18} & 0_{12 \times 18} \end{bmatrix} x + \begin{bmatrix} 0_{3 \times 3} \\ I_3 \\ 0_{12 \times 3} \end{bmatrix} [R_B^W B a_{com} + g] \quad (3-29)$$

对连续时间下的状态方程进行离散化。本文使用前向欧拉法, 方程如下:

$$\dot{x} = \frac{x(k+1) - x(k)}{\Delta t} = A_c x(k) + B_c u(k) \quad (3-30)$$

其中, Δt 表示控制周期, k 为当前时刻, $k+1$ 为下一时刻。

通过式(3-30)进一步推导出 $x(k+1)$ 的递推公式, 即 $k+1$ 时刻的状态 $x(k+1)$ 可以通过 k 时刻的状态 $x(k)$ 迭代求解, 迭代方程如下:

$$x(k+1) = (I + \Delta t \cdot A_c)x(k) + \Delta t \cdot B_c u(k) = A_k x(k) + B_k u(k) \quad (3-31)$$

所以离散状态空间模型的 A_k 、 B_k 分别为:

$$\begin{cases} A_k = I + \Delta t \cdot A_c \\ B_k = \Delta t \cdot B_c \end{cases} \quad (3-32)$$

因此可以写出形如 $x_k = Ax_{k-1} + Bu_{k-1} + w_{k-1}$ 的离散状态方程:

$$x_k = \begin{bmatrix} I_{3 \times 3} & \Delta t \times I_{3 \times 3} & 0_{3 \times 12} \\ 0_{3 \times 3} & I_{3 \times 3} & 0_{3 \times 12} \\ 0_{12 \times 3} & 0_{12 \times 3} & I_{12 \times 12} \end{bmatrix} x_{k-1} + \begin{bmatrix} 0_{3 \times 3} \\ \Delta t \times I_{3 \times 3} \\ 0_{12 \times 3} \end{bmatrix} [R_B^W B a_{com} + g] + w_{k-1} \quad (3-33)$$

其中, $A \in R^{18 \times 18}$ 表示状态矩阵; $B \in R^{18 \times 3}$ 表示控制矩阵; 状态矩阵和控制矩阵都是时不变的, 因此文中分别简写为 A 和 B 。 w_{k-1} 表示过程噪声, 服从高斯分布。

3.3.3 观测方程

足端位置从机身坐标系{B}到世界坐标系{W}的映射关系为:

$${}^W p_i = {}^W p_{com} + R_B^W \cdot {}^B p_{PBi} \quad (3-34)$$

足端速度从机身坐标系{B}到世界坐标系{W}的映射关系，对式(3-34)微分得：

$$\begin{aligned} {}^W v_i &= {}^W v_{com} + R_B^W \cdot {}^B p_{PBi} + R_B^W \cdot \dot{{}^B p}_{PBi} \\ &= {}^W v_{com} + [{}^B \omega_{com}]_x R_B^W \cdot {}^B p_{PBi} + R_B^W \cdot \dot{{}^B p}_{PBi} \\ &= {}^W v_{com} + R_B^W ([{}^B \omega_{com}]_x {}^B p_{PBi} + J_0 \dot{\theta}_0) \end{aligned} \quad (3-35)$$

旋转矩阵的导数为 $\dot{\mathbf{R}} = [\boldsymbol{\omega}]_{\times} \mathbf{R}$ 。 $[a]_{\times} \in R^{3 \times 3}$ 被定义为反对称矩阵，使得对于所有 $a, b \in R^{3 \times 1}$ ，都满足 $[a]_{\times} b = a \times b$ 。以三维向量 a 为例， a 的反对称矩阵形式表示如下：

$$[a]_{\times} = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix}_{\times} = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix} \quad (3-36)$$

定义系统的观测向量为：

$$y = \begin{bmatrix} {}^W p_{PB0} \\ {}^W p_{PB1} \\ {}^W p_{PB2} \\ {}^W p_{PB3} \\ {}^W v_{PB0} \\ {}^W v_{PB1} \\ {}^W v_{PB2} \\ {}^W v_{PB3} \\ {}^W p_{z0} \\ {}^W p_{z1} \\ {}^W p_{z2} \\ {}^W p_{z3} \end{bmatrix} = \begin{bmatrix} R_B^W {}^B p_{PB0} \\ R_B^W {}^B p_{PB1} \\ R_B^W {}^B p_{PB2} \\ R_B^W {}^B p_{PB3} \\ R_B^W ([{}^B \omega_{com}]_x {}^B p_{PB0} + J_0 \dot{\theta}_0) \\ R_B^W ([{}^B \omega_{com}]_x {}^B p_{PB1} + J_1 \dot{\theta}_1) \\ R_B^W ([{}^B \omega_{com}]_x {}^B p_{PB2} + J_2 \dot{\theta}_2) \\ R_B^W ([{}^B \omega_{com}]_x {}^B p_{PB3} + J_3 \dot{\theta}_3) \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (3-37)$$

其中， R_B^W 为机器人的机身坐标系{B}到世界坐标系{W}的旋转矩阵，由 IMU 传感器处理得到。在世界坐标系{W}下， ${}^W p_i$ 为足端位置， ${}^W v_i$ 为足端速度， ${}^W p_{PBi}$ 为足端相对于机身的位置， ${}^W v_{PBi}$ 为足端相对于机身的速度， ${}^W p_{zi}$ 为足端在空间中的高度，因为假设四个足端都与地面稳定接触，所以 ${}^W v_i$ 和 ${}^W p_{zi}$ 的值均为 0。

其次，在机身坐标系{B}下，加速度 ${}^B a_{com}$ ，角速度 ${}^B \omega_{com}$ ，均由 IMU 传感器直接测量得到。足端相对于机身的位置 ${}^B p_{PBi}$ 由机器人各个关节的编码器和机器人的正向运动学求得。 $J_i \dot{\theta}_i$ 为足端相对于机身的速度，由机器人一阶微分运动学求得。

将式(3-37)观测向量 y 中各变量和状态向量 x 建立联系,写为 $y = Cx$ 的形式:

$$y = \begin{bmatrix} {}^w p_{PB0} \\ {}^w p_{PB1} \\ {}^w p_{PB2} \\ {}^w p_{PB3} \\ {}^w v_{PB0} \\ {}^w v_{PB1} \\ {}^w v_{PB2} \\ {}^w v_{PB3} \\ {}^w p_{z0} \\ {}^w p_{z1} \\ {}^w p_{z2} \\ {}^w p_{z3} \end{bmatrix} = \begin{bmatrix} {}^w p_0 - {}^w p_{com} \\ {}^w p_1 - {}^w p_{com} \\ {}^w p_2 - {}^w p_{com} \\ {}^w p_3 - {}^w p_{com} \\ 0_{3 \times 1} - {}^w v_{com} \\ {}^w p_0 (2) \\ {}^w p_1 (2) \\ {}^w p_2 (2) \\ {}^w p_3 (2) \end{bmatrix} = Cx \quad (3-38)$$

其中， $w_{p_i}(2)$ 代表 w_{p_i} 向量的第三个元素，即 z 轴坐标。

对于系统的观测向量 y , 离散系统和连续系统并没有差别, 写成 $y_k = Cx_k + v_k$ 的形式:

$$y_k = \begin{bmatrix} -I_3 & I_3 & & & & \\ -I_3 & & I_3 & & & \\ -I_3 & & & I_3 & & \\ -I_3 & & & & I_3 & \\ -I_3 & & & & & I_3 \\ -I_3 & & & & & & \\ [001] & & [001] & & [001] & & [001] \\ & & & [001] & & & [001] \\ & & & & [001] & & [001] \end{bmatrix} x_k + v_k \quad (3-39)$$

其中， $C \in \mathbb{R}^{28 \times 18}$ 。 v_k 表示观测噪声，服从高斯分布。

3.3.4 误差分析

在上述状态估计器中，足端位置的计算基于其稳定触地的假设。当处于摆动腿时，基于该假设计算的结果不准确。状态估计器中的噪声协方差矩阵 Q 和 R 的对角线元素分别对应状态向量与观测向量的各分量，数值越大表示对该分量的滤波作用越强。噪声主要来自支撑足打滑、关节编码器

的噪声和 IMU 传感器的噪声。

对于支撑腿，足端触地时需考虑其稳定性。触地初期或准备腾空时，由于抖动较大，需采用较大的协方差；而支撑腿稳定时，抖动较小，协方差也相应减小。为此，通过引入窗口函数实现协方差的动态调整。

$$\sigma(x) = \begin{cases} \frac{x}{w_r}, & x < w_r \\ 1, & w_r \leq x \leq 1 - w_r \\ \frac{1-x}{w_r}, & x > 1 - w_r \end{cases} \quad (3-40)$$

假设一个足端在稳定触地时的噪声协方差矩阵为 C_{init} ，那么在整个触地周期中，它的噪声协方差矩阵 C_{stance} 应该为：

$$C_{stance} = [1 + (1 - \sigma)L]C_{init} \quad (3-41)$$

其中， σ 为可信度，由窗口函数 $\sigma(x)$ 计算得到。 L 用来代替 $+\infty$ 的极大数。

对于状态方程，机身的位置 $^w p_{com}$ 和速度 $^w v_{com}$ 的噪声主要受到 IMU 传感器中的加速度计影响，因传感器本身已经做过滤波，所以对应的协方差元素可设为较小值。

足端的位置 $^w p_i$ 依赖于触地假设。

当第 i 腿处于摆动腿时，需要将 Q 矩阵中 $^w p_i$ 的对应协方差设为 $+\infty$ 值。

当第 i 腿处于支撑腿时，需要将 Q 矩阵中 $^w p_i$ 的对应协方差设为较大的值，并且引入窗口函数 $\sigma(x)$ 使对应协方差可变。

对于观测方程，根据实际测试发现 $^w p_{PBi}$ 主要受到支撑足打滑和关节编码器角度噪声的影响，但幅度较小，因此对应协方差相对较低。

足端到机身的速度 $^w v_{PBi}$ 和足端高度 $^w p_{zi}$ 都依赖于触地假设。

当第 i 腿处于摆动腿时，需要将 R 矩阵中 $^w v_{PBi}$ 和 $^w p_{zi}$ 的对应协方差设为 $+\infty$ 值。

当第 i 腿处于支撑腿时， $^w v_{PBi}$ 的噪声则主要来自关节编码器转速噪声的影响，实测表明该噪声较大。而 $^w p_{zi}$ 噪声源自足端弹性引发的振动，幅度大，因此需设置较高的滤波强度。需要将 R 矩阵中 $^w v_{PBi}$ 和 $^w p_{zi}$ 的对应协方差设为较大的值，并且引入窗口函数 $\sigma(x)$ 使对应协方差可变。

综合上述分析及实测结果，最终确定的协方差参数见表 3-2。

表 3-2 状态估计器协方差参数

状态量/观测量	协方差
$w_{p_{com}}$	0.001
$w_{v_{com}}$	0.001
w_{p_i}	$+\infty/0.01$
$w_{p_{PBi}}$	0.02
$w_{v_{PBi}}$	$+\infty/3.0$
$w_{p_{zi}}$	$+\infty/1.0$

3.3.5 状态估计

在过程方程和观测方程的基础上，本文使用离散线性卡尔曼滤波器对其进行融合估计：

$$\begin{cases} \hat{x}_k^- = A\hat{x}_{k-1}^+ + Bu_{k-1} \\ P_k^- = AP_{k-1}^+A^T + Q \\ K_k = P_k^-C^T(R + CP_k^-C^T)^{-1} \\ \hat{x}_k^+ = \hat{x}_k^- + K_k(y_k - C\hat{x}_k^-) \\ P_k^+ = (I - K_kC)P_k^-(I - K_kC)^T + K_kRK_k^T \end{cases} \quad (3-42)$$

其中， \hat{x}_k^- 和 \hat{x}_k^+ 分别为状态向量的先验和后验估计； P_k^- 和 P_k^+ 分别为协方差矩阵的先验和后验估计； K_k 为卡尔曼增益； Q 、 R 分别为过程噪声和测量噪声协方差矩阵； \hat{x}_k^+ 即为状态估计器的最终估计结果。

计算卡尔曼增益 K_k 时，需要求解矩阵的逆，但在编程中直接求逆效率低且精度差。因此，本文采用以下方法将求逆转化为 LU 分解法：

令： $S = R + CP_k^-C^T$ ，则卡尔曼增益 K_k 化简为：

$$K_k = P_k^-C^TS^{-1} \quad (3-43)$$

所以 \hat{x}_k^+ 化简为：

$$\hat{x}_k^+ = \hat{x}_k^- + P_k^-C^TS^{-1}(y_k - C\hat{x}_k^-) \quad (3-44)$$

令： $S_y = S^{-1}(y_k - C\hat{x}_k^-)$ ，则有：

$$S \cdot S_y = y_k - C\hat{x}_k^- \quad (3-45)$$

对于式 (3-45)，可以用 Eigen 的 LU 分解法求出 S_y ：

$$S_y = (y_k - C\hat{x}_k^-)/S \quad (3-46)$$

同理， P_k^+ 化简为：

$$\begin{aligned} P_k^+ &= (I - P_k^- C^T S^{-1} C) P_k^- (I - P_k^- C^T S^{-1} C)^T + P_k^- C^T S^{-1} R (P_k^- C^T S^{-1})^T \\ &= (I - P_k^- C^T S^{-1} C) P_k^- (I - P_k^- C^T S^{-1} C)^T + P_k^- C^T S^{-1} R (S^T)^{-1} C (P_k^-)^T \end{aligned} \quad (3-47)$$

对 $S^{-1}C$, $S^{-1}R$, $(S^T)^{-1}C$ 三项, 使用同样的方法求解。

至此, 完成了四足机器人的状态估计。

3.3.6 坡度估计

四足机器人在复杂环境中运动时, 需要准确辨识起伏地面。尤其在长斜坡或楼梯上, 若不主动调整躯干姿态, 前腿工作空间变小, 易降低行走效率甚至导致摔倒, 如图 3-6(a)所示。为应对这些障碍, 机器人应将期望俯仰角调至与地面坡度一致, 如图 3-6(b)所示。本文采用最小二乘法拟合平面。

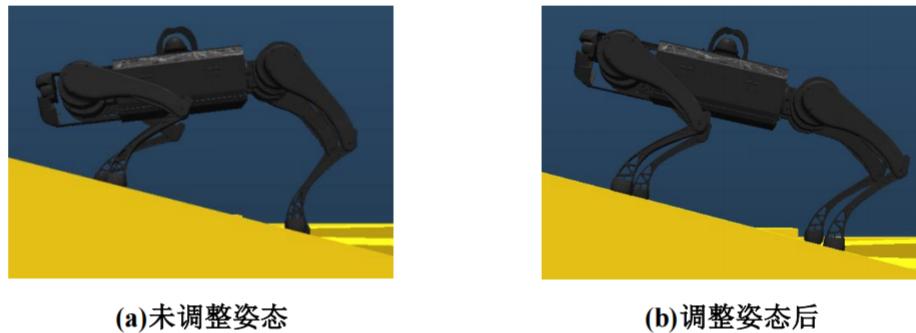


图 3-6 机器人上斜坡姿态对比

机器人不可能在垂直墙面上行走, 因此可将地面的平面方程表示为:

$$z = a + bx + cy = [1 \ x \ y][a \ b \ c]^T \quad (3-48)$$

令: $A_{pla} = [a \ b \ c]^T$, 则:

$$z = [1 \ x \ y]A_{pla} \quad (3-49)$$

机器人每条腿在运动过程中会交替切换为支撑腿和摆动腿状态。

定义 ${}^w_i p^k$ 为第 i 号腿最后成为支撑腿时的足端坐标, 由状态估计器获得。具体而言, 当第 i 号腿为支撑腿并且支撑腿相位 p 在 0.25 至 0.75 范围内 (即 $0.25 < p < 0.75$) 时, ${}^w_i p^k$ 刷新为当前的足端坐标 ${}^w_i p^k$ 。而当第 i 号腿处于摆动腿状态时, ${}^w_i p^k$ 不再刷新, 表示为 ${}^w_i p^{k-1}$ 。

这一坐标值将通过以下递推方法获得:

$$\begin{cases} {}^w_i p^0 = {}^w_i p_{i,t=0} \\ {}^w_i p^k = {}^w_i p^k, \quad S_i = 1 \\ {}^w_i p^k = {}^w_i p^{k-1}, \quad S_i = 0 \end{cases} \quad (3-50)$$

将 ${}^W_i p^k$ 的xy坐标与z坐标分离，并分别填充到下面两个矩阵中：

$${}^W z_f = [{}^W_0 p^k(2) \quad {}^W_1 p^k(2) \quad {}^W_2 p^k(2) \quad {}^W_3 p^k(2)]^T \quad (3-51)$$

$$W_{pla} = \begin{bmatrix} 1 & {}^W_0 p^k(0) & {}^W_0 p^k(1) \\ 1 & {}^W_1 p^k(0) & {}^W_1 p^k(1) \\ 1 & {}^W_2 p^k(0) & {}^W_2 p^k(1) \\ 1 & {}^W_3 p^k(0) & {}^W_3 p^k(1) \end{bmatrix} \quad (3-52)$$

对于平面方程式(3-49)，可利用最小二乘法计算：

$$A_{pla} = W_{pla}^+ \cdot {}^W z_f \quad (3-53)$$

其中 W_{pla}^+ 表示 W_{pla} 的伪逆矩阵，为了减少测量噪声的影响，设计一个低通滤波器，滤波系数 $\eta=0.2$ 。

$$A_{pla}^k = [a \quad b \quad c]^T = \eta A_{pla}^k + (1 - \eta) A_{pla}^{k-1} \quad (3-54)$$

对于平面方程 $Ax + By + Cz + D = 0$ ，其平面法向量 $n = [A \quad B \quad C]^T$ 。因此，对于平面方程式(3-48)，其平面法向量在世界坐标系{W}下的描述为 ${}^W n = [-b \quad -c \quad 1]^T$ ，将其单位化：

$${}^W n_e = \frac{{}^W n}{|{}^W n|} \quad (3-55)$$

由式(3-25)可得， ${}^W n_e$ 是机体旋转矩阵 R_B^W 的第三列，结合旋转矩阵 R_B^W 的公式得：

$${}^W n_e = \begin{bmatrix} s_\phi s_\psi + c_\phi c_\psi s_\theta \\ c_\phi s_\theta s_\psi - c_\psi s_\phi \\ c_\phi c_\theta \end{bmatrix} = \begin{bmatrix} c_\psi & s_\psi & 0 \\ s_\psi & -c_\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_\phi s_\theta \\ s_\phi \\ c_\phi c_\theta \end{bmatrix} = r_{pla} \xi \quad (3-56)$$

偏航角 ψ 由IMU获得，因此矩阵 r_{pla} 已知，可解出 ξ ：

$$\xi = r_{pla}^+ n_e \quad (3-57)$$

ξ 已经求解，因此可得机器人期望横滚角 ϕ 与期望俯仰角 θ 为：

$$\begin{cases} \phi = \arcsin(\xi(1)) \\ \theta = \arctan(\xi(0)/\xi(2)) \end{cases} \quad (3-58)$$

至此完成了四足机器人的地面坡度估计。

3.4 常用步态和摆动腿控制器

3.4.1 常用步态

在四足机器人的运动过程中，使用步态这一概念来描述机器人支撑腿(Stance Leg)与摆动腿(Swing Leg)的周期性调度行为。以下是描述步

态的几个参数：

步态周期 (P)：是指足端完成一次完整步态循环所需的时间。在步态周期内，足端依次经历触地和腾空状态的交替。

支撑腿占空比 (s_r)：是支撑腿在一个步态周期内处于触地状态的时间比例，取值范围为 (0, 1)。支撑腿占空比越大，支撑腿的接地时间越长，步态越稳定。

偏移时间 (b)：是指四条腿支撑状态起始时刻在步态周期中的相对位置。

常见的四足机器人步态按速度从慢到快通常包括以下几种：

站立 (Standing)：四足机器人站立时所有腿都支撑身体。无步态生成，通常用于静止或待命。

稳定行走 (Walk)：四足机器人正常的步态，支撑腿占空比 $s_r > 0.5$ ， $b = [0.25 \ 0.75 \ 0.5 \ 0.0]$ ，如图 3-7 所示。在任意时刻，至少三条腿处于支撑状态，适用于崎岖地形。步伐稳定，速度适中，具有较好的地形适应能力。

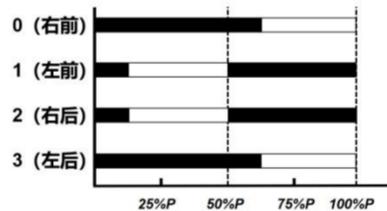


图 3-7 稳定行走 (Walk)

快速行走 (Trot)：四足机器人的快速步态，支撑腿占空比 $s_r = 0.5$ ， $b = [0.0 \ 0.5 \ 0.0 \ 0.5]$ ，如图 3-8 所示。通常是一对前后腿同时运动，步伐较快，兼顾速度和稳定性。

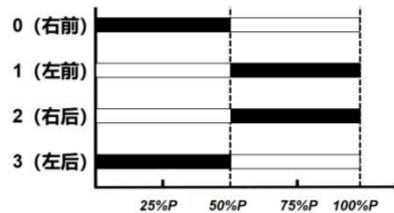


图 3-8 快速行走 (Trot)

奔跑 (Flying-trot)：四足机器人高速奔跑的步态，支撑腿占空比 $s_r < 0.5$ ， $b = [0.0 \ 0.5 \ 0.0 \ 0.5]$ ，如图 3-9 所示。因此会出现四个足端同时腾空的状态，控制难度较高，适用于较平坦地面上的快速移动。

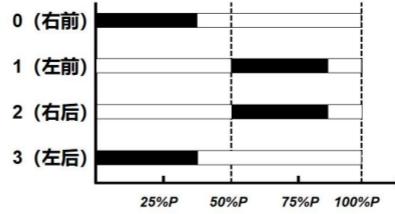


图 3-9 奔跑 (Flying-trot)

3.4.2 步态调度器

考虑到在足端安装传感器会引发以下问题：第一，足底传感器位于腿部末端，增加的质量会显著提升腿部的转动惯量；第二，对传感器的耐久性要求更高；第三，线路布置也面临较大挑战。因此，设计了基于时间步态调度器，按预设时序自动切换摆动腿与支撑腿状态。

设 T_{start} 为系统初始化完成的时间， T_{now} 为系统当前的时间。 P 为一个完整的迈步周期， $T_{swing} = (1 - s_r)P$ 为摆动腿时间， $T_{stance} = s_r P$ 为支撑腿时间。

定义系统经过的时间 t_{pass} ：

$$t_{pass} = T_{now} - T_{start} \quad (3-59)$$

计算第 i 条腿 ($i = 0, 1, 2, 3$) 在当前步态周期的时间进度，并将其归一化，得到 $|T_i| \in (0, 1)$ ：

$$|T_i| = \frac{\text{mod}(t_{pass} + P - P \cdot b_i, P)}{P} \quad (3-60)$$

计算第 i 条腿的接触状态与相位，支撑腿占空比为 s_r ，则：

$$S_i = \begin{cases} 1, & |T_i| < s_r, \\ 0, & |T_i| \geq s_r, \end{cases}$$

$$p_i = \begin{cases} \frac{|T_i|}{s_r}, & |T_i| < s_r, \\ \frac{|T_i| - s_r}{1 - s_r}, & |T_i| \geq s_r. \end{cases} \quad (3-61)$$

当 $|T_i| < s_r$ 时，第 i 条腿为支撑腿， $S_i = 1$ ，支撑腿相位为 $p_i = \frac{|T_i|}{s_r}$ 。

当 $|T_i| \geq s_r$ 时，第 i 条腿为摆动腿， $S_i = 0$ ，摆动腿相位为 $p_i = \frac{|T_i| - s_r}{1 - s_r}$ 。

3.4.3 落足点规划

本文采用基于 Raibert 启发落足点^[16]的规划方式。四足机器人的前进速

度控制是通过落足点的位置规划来控制的。落足点规划的控制原理用下图解释：

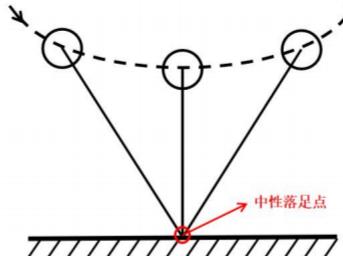


图 3-10 中性落足点示意图

如图 3-10 所示，中性落足点是理想的足端着地点。其特点是以纵轴为中心，左右运动轨迹和姿态均呈对称状态，并确保机器人腾空起步和触地时的前进速度相等。若落足点位于中性点之后，离地后质心速度增大，实现加速；反之则减速。

对于四足机器人，为简化运动控制模型。假设重心位于躯干形心附近，各足中性落足点设在大腿关节正下方，使重心投影落在对角足端中性落足点连线中点。

对于平移运动，如图 3-11 所示。具体分析如下：

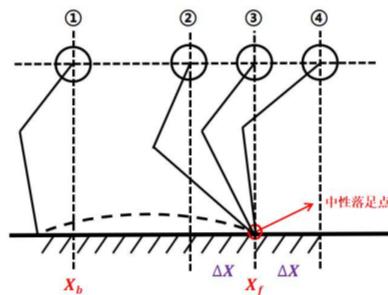


图 3-11 匀速平移落足点

上图为四足机器人的一条腿从抬起到落下的过程：

- (1) ③为中性落足点。
- (2) ①~②阶段为摆动相，机器人速度为 v_{xcom} ，摆动时间为 T_{swing} ，
则 $d_{12} = v_{xcom} t_{swing}$ 。
- (3) ②~④阶段为支撑相，机器人速度为 v_{xcom} ，触底时间为 T_{stance} ，
则 $d_{23} = \Delta x = \frac{1}{2} v_{xcom} t_{stance}$ 。
- (4) 若 $v_{xcom} < v_{xdcom}$ ，机器人加速，需要后移落足点，否则反之。
因此增加 $k_x(v_{xcom} - v_{xdcom})$ 项。

结合以上四点，可得平移运动的落足点计算公式为：

$$\begin{cases} {}^w x_i = {}^w x_b + {}^w v_{xcom}(1-p)T_{swing} + \frac{1}{2} {}^w v_{xcom} T_{stance} + k_x({}^w v_{xcom} - {}^w v_{xdcom}) \\ {}^w y_i = {}^w y_b + {}^w v_{ycom}(1-p)T_{swing} + \frac{1}{2} {}^w v_{ycom} T_{stance} + k_y({}^w v_{ycom} - {}^w v_{ydcom}) \end{cases} \quad (3-62)$$

其中，在世界坐标系{W}下， ${}^w x_i$ 、 ${}^w y_i$ 为第*i*条腿的目标落足点。 ${}^w x_b$ 、 ${}^w y_b$ 为大腿关节坐标，由状态估计器计算获得。 ${}^w v_{xcom}$ 、 ${}^w v_{ycom}$ 为机身当前的速度。 ${}^w v_{xdcom}$ 、 ${}^w v_{ydcom}$ 为期望的机身速度。

对于旋转运动，如图3-12所示。中性落足点会绕机器人中心以半径R旋转，R为机器人静止站立时设定的足端位置到机身中心的水平距离，为常数。

可得旋转运动的落足点计算公式为：

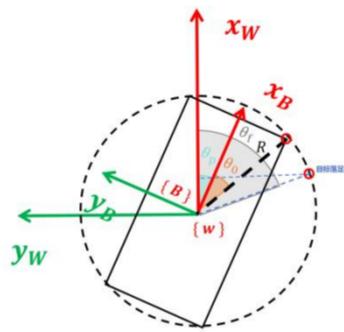


图3-12 匀速旋转落足点

$${}^w \theta_f = {}^w \theta_{yaw} + \theta_0 + {}^w \omega(1-p)T_{swing} + \frac{1}{2} {}^w \omega T_{stance} + k_\omega({}^w \omega - {}^w \omega_d) \quad (3-63)$$

其中，在世界坐标系{W}下， ${}^w \theta_f$ 为目标落足点与机身中心连线到世界坐标系 x_w 轴在xy平面的夹角。 ${}^w \theta_{yaw}$ 为机器人的偏航角。 θ_0 为大腿关节与机身中心连线到机身坐标系 x_B 轴的夹角，为常数。 p 为第*i*条腿的摆动腿的相位。 ${}^w \omega$ 为角速度。 ${}^w \omega_d$ 为期望的角速度。

则世界坐标系下的落足点为：

$$\begin{cases} {}^w x_i = R \cos {}^w \theta_f \\ {}^w y_i = R \sin {}^w \theta_f \end{cases} \quad (3-64)$$

结合平移与旋转，第*i*条腿落足点坐标为：

$$\begin{cases} {}^w x_i = {}^w x_b + R \cos {}^w \theta_f + {}^w v_{xcom}(1-p)T_{swing} + \frac{1}{2} {}^w v_{xcom} T_{stance} + k_x({}^w v_{xcom} - {}^w v_{xdcom}) \\ {}^w y_i = {}^w y_b + R \sin {}^w \theta_f + {}^w v_{ycom}(1-p)T_{swing} + \frac{1}{2} {}^w v_{ycom} T_{stance} + k_y({}^w v_{ycom} - {}^w v_{ydcom}) \end{cases} \quad (3-65)$$

可以看出，落足点的位置与质心速度、步态周期(*P*)、支撑腿占空比(*s_r*)有关。

3.4.4 摆动腿轨迹

本文采用基于摆线 (Cubic Spline) 的轨迹。实验验证表明，该方法在复杂地形下表现稳定。下文给出足端摆线轨迹的位置函数表达式：

$$\begin{cases} x_c = x_0 + \frac{x_1 - x_0}{2\pi} (2\pi p - \sin 2\pi p) \\ y_c = y_0 + \frac{y_1 - y_0}{2\pi} (2\pi p - \sin 2\pi p) \\ z_c = z_0 + \frac{h}{2} (1 - \cos 2\pi p) \end{cases} \quad (3-66)$$

同理，足端的摆线轨迹的速度函数表达式为：

$$\begin{cases} \dot{x}_c = \frac{x_1 - x_0}{T} (1 - \cos 2\pi p) \\ \dot{y}_c = \frac{y_1 - y_0}{T} (1 - \cos 2\pi p) \\ \dot{z}_c = \frac{\pi h}{T} \sin 2\pi p \end{cases} \quad (3-67)$$

其中，在世界坐标系{W}下， (x_c, y_c, z_c) 为足端的摆线轨迹位置。 (x_0, y_0, z_0) 为足端的腾空起点。 (x_1, y_1, z_1) 为足端的最终落足点。 $(\dot{x}_c, \dot{y}_c, \dot{z}_c)$ 为足端的速度。 h 为足端抬起的最大高度。 p 为摆动腿的相位。

得到世界系{W}下的期望的足端位置和速度后，首先将其变换到机身系{B}，然后通过逆运动学和微分运动学解算，得到期望的关节角度 q 和转速 \dot{q} ，并发送至关节电机驱动器，由关节电机驱动器内的 PD 控制器执行输出。

关节电机驱动器的控制规律为：

$$\tau = \tau_{ff} + K_p(q_d - q) + K_d(\dot{q}_d - \dot{q}) \quad (3-68)$$

其中， τ 为关节输出力矩； τ_{ff} 为前馈力矩(该项在支撑腿时用于直接输出期望力矩，摆动腿控制器中未使用)。 q 和 \dot{q} 为关节实际角度和转速； q_d 和 \dot{q}_d 为关节目标角度和转速； K_p 为角度刚度系数； K_d 为转速刚度系数；通过调节 K_p 和 K_d 可以实现对期望足端轨迹的良好跟踪。

4 基于 MPC 的支撑腿控制器

本文参考文献^[3]中提出的 Convex MPC 方法，实现对四足机器人支撑腿的精准控制，从而使机器人能够跟踪期望的身体位姿。为简化建模，本文将四足机器人视为单刚体模型，并利用 MPC 求解达到目标状态所需地面反作用力。

4.1 模型预测控制基本原理

模型预测控制基本原理，如图 4-1 所示。可归纳如下：

- (1) 预测：在 k 时刻估计或测量系统的状态值作为反馈，利用系统模型预测未来一段时间内的系统输出 $x_{[k|k]}, x_{[k+1|k]} \dots x_{[k+h|k]}$ （蓝色曲线）。
- (2) 优化：在预测区间 h 的时域内，通过优化算法求解出在 k 时刻的最优控制序列 $u_{[k|k]}, u_{[k+1|k]} \dots u_{[k+h|k]}$ （红色曲线）。使预测输出（蓝色曲线）尽可能接近预设的参考轨迹（绿色曲线）。
- (3) 执行：在每个控制周期中，仅将优化得到的控制序列中的第一组结果 $u_{[k|k]}$ ，应用到系统中。
- (4) 滚动优化：在每个新的控制周期，重复上述过程，基于最新的系统状态和输出反馈，更新预测和优化。

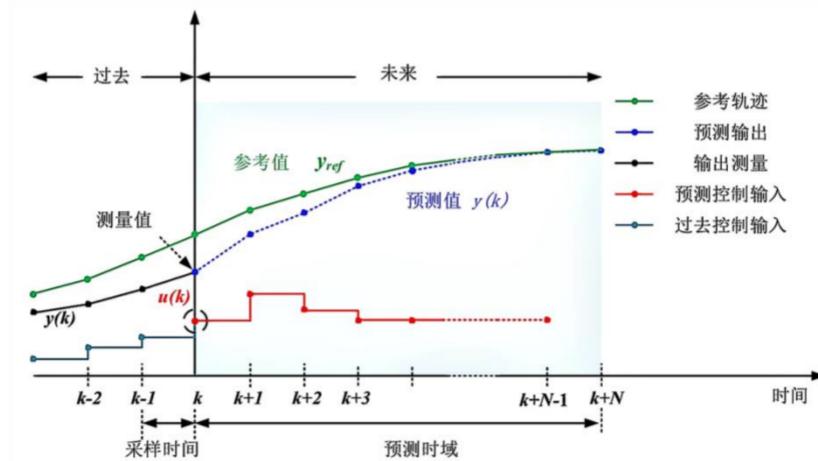


图 4-1 模型预测控制基本原理图

四足机器人的运动状态主要通过四条腿与地面的相互作用来改变。然而，在腾空阶段，机器人无法利用地面反作用力来控制自身状态，导致系统呈现间歇性欠驱动特性。因此，控制系统需要具备预测能力，即在欠驱

动状态发生前预见并采取相应措施。

相比于 SLIP、VMC 和 WBC 等瞬时控制器，模型预测控制（MPC）通过预测机器人的未来状态，能够更有效地控制具有腾空行为的动作，特别适用于具有复杂动力学模型和约束条件的系统。在四足机器人的欠驱动运动中，采用 MPC 控制具有以下优势：

(1) 处理欠驱动问题：四足机器人在高速运动中经常出现腾空阶段，此时无法通过地面反作用力来控制自身状态，导致系统处于欠驱动状态。MPC 能够在预测未来状态的基础上，提前调整控制策略，确保在欠驱动阶段维持机器人的稳定性。

(2) 满足摩擦约束：为了防止足端打滑，必须确保足端对地面的作用力与地面法线的夹角小于摩擦角。MPC 可以将这一摩擦力约束直接纳入优化问题中，在防止打滑的同时，充分利用摩擦力，提升机器人的动态性能。

(3) 应对环境不确定性：在复杂地形或未知环境中，机器人可能受到外界干扰或出现足端打滑等情况，导致偏离期望轨迹。MPC 能够根据实际运动情况重新进行预测，并及时调整控制输入，以适应不同的约束条件和环境变化，实现更好的轨迹跟踪。

(4) 优化运动控制能力：MPC 将机器人的运动控制问题转化为一个优化问题，通过不断地优化控制策略，实现更好的运动控制能力。相比其他控制方法，MPC 可以在考虑更多约束条件的情况下，实现更高效的运动控制。

因此，模型预测控制（MPC）在四足机器人运动控制中，通过预测未来状态，提前调整策略，解决了欠驱动和摩擦约束等问题，提升了机器人的动态性能和环境适应能力。

4.2 单刚体动力学模型构建

本文采用单刚体模型简化四足机器人的动力学分析，将机器人视为一个整体刚体，从而忽略部件间的相互作用，降低计算复杂性。该方法适用于四足机器人，因为电机集中于躯干，腿部质量较小，腿部在空中运动对整体影响可忽略。模型主要假设包括：

- (1) 零件刚体假设：所有零件均为刚体，在外力作用下不发生形变。
- (2) 忽略动量贡献：关节和腿部快速运动产生的动量可以忽略。
- (3) 忽略腿部空中影响：腿部空中运动对系统惯性和地面反力无显著影响。

(4) 惯性张量恒定：整体惯性张量不随肢体位置变化而改变。

基于这些假设，下文将建立四足机器人简化的单刚体动力学状态空间方程。

4.2.1 单刚体动力学

首先建立四足机器人单刚体动力学模型，如图 4-2 所示，其中红色箭头是世界坐标系下地面对机器人的反作用力 ${}^W f_i$ ，蓝色向量 ${}^W r_i$ 是足端在世界坐标中相对质心的位置矢量。下文中将建立单刚体模型的牛顿-欧拉方程。

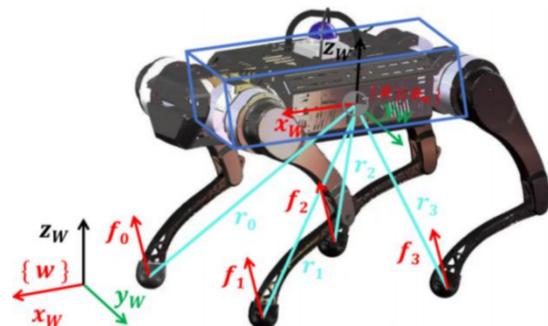


图 4-2 单刚体简化动力学模型

$$\begin{cases} m(\ddot{{}^W p_{com}} - g) = \sum_i {}^W f_i \\ \frac{d}{dt}(I \cdot {}^W \omega) = \sum_i ({}^W r_i \times {}^W f_i) \end{cases} \quad (4-1)$$

其中， m 为机器人的质量； n_i 为支撑腿的个数。

对于牛顿方程，在世界坐标系 $\{W\}$ 下， ${}^W \ddot{p}_{com}$ 为质心加速度， $g = [0 \ 0 \ -9.81]^T$ 为重力加速度， ${}^W f_i$ 为地面对机器人的反作用力。

对于欧拉方程， I 是机器人的惯性张量，其原点位于刚体质心、姿态与世界坐标系 $\{W\}$ 一致的定向机体坐标系 $\{Bw\}$ 下。在世界坐标系 $\{W\}$ 下， ${}^W \omega$ 为身体角速度， ${}^W r_i$ 是足端相对质心的位置矢量， ${}^W f_i$ 为地面对机器人的反作用力。

惯性张量描述了质量在刚体中的分布情况。考虑到在定向机体坐标系 $\{Bw\}$ 下，刚体的惯性张量是其姿态的函数，而在机体坐标系 $\{B\}$ 下的惯性张量 I^B 是常量，因此需要推导惯性张量从机体坐标系 $\{B\}$ 到定向机体坐标系 $\{Bw\}$ 的映射关系。

显然，在不同坐标系下，刚体的转动动能 K 都是相等的，因此对于定向机体坐标系 $\{B_w\}$ 和机体坐标系 $\{B\}$ 有：

$$\begin{aligned} {}^{Bw}K &= {}^BK \\ \frac{1}{2} {}^{Bw}\omega^T \cdot I \cdot {}^{Bw}\omega &= \frac{1}{2} {}^B\omega^T \cdot I^B \cdot {}^B\omega \\ {}^{Bw}\omega^T \cdot I \cdot {}^{Bw}\omega &= (R_W^B \cdot {}^{Bw}\omega)^T I^B (R_W^B \cdot {}^{Bw}\omega) \\ {}^{Bw}\omega^T \cdot I \cdot {}^{Bw}\omega &= {}^{Bw}\omega^T (R_B^W \cdot I^B \cdot R_B^{W^T}) {}^{Bw}\omega \end{aligned} \quad (4-2)$$

因此，将机身坐标系 $\{B\}$ 下的机器人惯性张量 I^B 转化为 $\{B_w\}$ 系下的机器人惯性张量 I ：

$$I = R_B^W \cdot I^B \cdot R_B^{W^T} \quad (4-3)$$

其中， R_B^W 为机器人的机身坐标系 $\{B\}$ 到世界坐标系 $\{W\}$ 的旋转矩阵。

在欧拉方程中角动量对时间的导数存在叉乘项 ${}^w\omega \times (I \cdot {}^w\omega)$ ，是非线性的。

$$\frac{d}{dt}(I \cdot {}^w\omega) = I \cdot {}^w\dot{\omega} + {}^w\omega \times (I \cdot {}^w\omega) \quad (4-4)$$

考虑到四足机器人的机体近似为一个对称的规则长方体，其空间转动惯量矩阵中的非对角项可以忽略，即：

$$I \approx \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \quad (4-5)$$

因此：

$${}^w\omega \times (I \cdot {}^w\omega) = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix} \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} = \begin{bmatrix} (I_{zz} - I_{yy})\omega_z\omega_y \\ (I_{xx} - I_{zz})\omega_x\omega_z \\ (I_{yy} - I_{xx})\omega_y\omega_x \end{bmatrix} \quad (4-6)$$

因为四足机器人一般不会快速旋转，因此关于角速度的平方项可以忽略不计，所以该叉乘项可以直接忽略掉，从而得到线性化方程：

$$\frac{d}{dt}(I \cdot {}^w\omega) \approx I \cdot {}^w\dot{\omega} \quad (4-7)$$

对式(4-1)化简：

$$\begin{cases} m({}^w\ddot{p}_{com} - g) = \sum_{i=0}^3 {}^w f_i \\ R_B^W \cdot I^B \cdot R_B^{W^T} \cdot {}^w\dot{\omega} = \sum_{i=0}^3 ({}^w r_i \times {}^w f_i) \end{cases} \quad (4-8)$$

写为矩阵形式：

$$\begin{bmatrix} 1 \\ {}^W r_0 & {}^W r_1 & {}^W r_2 & {}^W r_3 \end{bmatrix} \begin{bmatrix} {}^W f_0 \\ {}^W f_1 \\ {}^W f_2 \\ {}^W f_3 \end{bmatrix} = \begin{bmatrix} m({}^W \ddot{p}_{com} - g) \\ R_B^W \cdot I^B \cdot R_B^{W^T} \cdot {}^W \omega \end{bmatrix} \quad (4-9)$$

在控制器具体实现时，当某条腿处于腾空状态，未与地面接触时，只需将该条腿对应的足端力赋值为零向量即可。

至此，已完成了四足机器人单刚体动力学建模。将机器人的足端力与机器人的运动联系起来。

4.2.2 近似角速度动力学

使用 ZYX 欧拉角 θ 表示机器人的姿态， $\theta = [\phi \theta \psi]^T$ ，其中 ϕ 是横滚角， θ 是俯仰角， ψ 是偏航角。

ZYX 欧拉角 θ 和世界坐标系 {W} 下的角速度 ${}^W \omega$ 存在以下对应关系：

$${}^W \omega = \begin{bmatrix} \cos(\theta)\cos(\psi) & -\sin(\psi) & 0 \\ \cos(\theta)\sin(\psi) & \cos(\psi) & 0 \\ -\sin(\theta) & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \quad (4-10)$$

四足机器人处在正常运动状态时，机器人不可能垂直，俯仰角 $\theta \neq 90^\circ$ ，($\cos(\theta) \neq 0$)，式 (4-10) 可以被求逆为：

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} \cos(\psi)/\cos(\theta) & \sin(\psi)/\cos(\theta) & 0 \\ -\sin(\psi) & \cos(\psi) & 0 \\ \cos(\psi)\tan(\theta) & \sin(\psi)\tan(\theta) & 1 \end{bmatrix} {}^W \omega \quad (4-11)$$

假设四足机器人在运动过程中，姿态相对稳定，即俯仰角和横滚角(θ , ϕ)都为 0，式 (4-11) 可以近似为：

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \approx \begin{bmatrix} \cos(\psi) & \sin(\psi) & 0 \\ -\sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} {}^W \omega \quad (4-12)$$

等同于：

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \approx R_z^\top(\psi) {}^W \omega \quad (4-13)$$

至此，完成了四足机器人近似角速度动力学建模。将机器人世界系 {W} 下的角速度与机器人的欧拉角的微分联系起来。

4.2.3 离散动力学状态空间方程

将四足机器人在世界坐标系{W}下机身的欧拉角 θ 、质心位置坐标 ${}^w p_{com}$ 、角速度 ${}^w \omega$ 和质心线速度 ${}^w \dot{p}_{com}$ 作为状态向量 x , 整合为如式(4-14)所示的状态空间方程:

$$\begin{aligned} \frac{d}{dt} \begin{bmatrix} \theta \\ {}^w p_{com} \\ {}^w \omega \\ {}^w \dot{p}_{com} \end{bmatrix} &= \begin{bmatrix} 0_{3 \times 3} & 0_{3 \times 3} & R_z^\top(\psi) & 0_{3 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & 1_3 \\ 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} \end{bmatrix} \begin{bmatrix} \theta \\ {}^w p_{com} \\ {}^w \omega \\ {}^w \dot{p}_{com} \end{bmatrix} + \\ &\quad \begin{bmatrix} 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} \\ I^{-1}[{}^w r_0]_x & I^{-1}[{}^w r_1]_x & I^{-1}[{}^w r_2]_x & I^{-1}[{}^w r_3]_x \\ 1_3/m & 1_3/m & 1_3/m & 1_3/m \end{bmatrix} \begin{bmatrix} {}^w f_0 \\ {}^w f_1 \\ {}^w f_2 \\ {}^w f_3 \end{bmatrix} + \begin{bmatrix} 0_{11 \times 1} \\ g \end{bmatrix} \end{aligned} \quad (4-14)$$

其中, $1_3 \in R^{3 \times 3}$ 表示单位矩阵; I^{-1} 是惯性张量 I 的逆矩阵, I 在式(4-3)求得; $[{}^w r_i]_x$ 表示将三维向量 ${}^w r_i$ 转换为反对称矩阵形式。

将式(4-14)右侧的重力加速度项 g 整合到状态方程中, 写为 $\dot{x} = A_c x + B_c u$ 的形式, 即可得到连续时间下四足机器人的线性时变状态空间方程, 具体变换如下:

$$\begin{aligned} \frac{d}{dt} \begin{bmatrix} \theta \\ {}^w p_{com} \\ {}^w \omega \\ {}^w \dot{p}_{com} \\ g(3) \end{bmatrix} &= \begin{bmatrix} 0_{3 \times 3} & 0_{3 \times 3} & R_z^\top(\psi) & 0_{3 \times 3} & 0 \\ 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & 1_3 & 0 \\ 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & 0 \\ 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & 1 \\ 0_{1 \times 3} & 0_{1 \times 3} & 0_{1 \times 3} & 0_{1 \times 3} & 0 \end{bmatrix} \begin{bmatrix} \theta \\ {}^w p_{com} \\ {}^w \omega \\ {}^w \dot{p}_{com} \\ g(3) \end{bmatrix} + \\ &\quad \begin{bmatrix} 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} \\ I^{-1}[{}^w r_0]_x & I^{-1}[{}^w r_1]_x & I^{-1}[{}^w r_2]_x & I^{-1}[{}^w r_3]_x \\ 1_3/m & 1_3/m & 1_3/m & 1_3/m \\ 0_{1 \times 3} & 0_{1 \times 3} & 0_{1 \times 3} & 0_{1 \times 3} \end{bmatrix} \begin{bmatrix} {}^w f_0 \\ {}^w f_1 \\ {}^w f_2 \\ {}^w f_3 \end{bmatrix} \end{aligned} \quad (4-15)$$

其中, $A_c \in R^{13 \times 13}$ 矩阵是系统状态矩阵。 $B_c \in R^{13 \times 12}$ 矩阵是控制矩阵。 $x \in R^{13 \times 1}$ 是状态向量。 $u \in R^{12 \times 1}$ 是输入矩阵。将四足机器人足端所受的地面反作用力作为控制输入, 体现了四足机器人运动过程中, 主要依靠地面反力来改变自身的运动状态。

$$x = \begin{bmatrix} \theta \\ {}^w p_{com} \\ {}^w \omega \\ {}^w \dot{p}_{com} \\ g(3) \end{bmatrix} \quad u = \begin{bmatrix} {}^w f_0 \\ {}^w f_1 \\ {}^w f_2 \\ {}^w f_3 \end{bmatrix} \quad (4-16)$$

对式(4-15)采用上文提到的前向欧拉法进行离散，得到形如 $x_{k+1} = Ax_k + Bu_k$ 的离散时间状态空间表达式：

$$\begin{aligned} \begin{bmatrix} \theta \\ w_p \\ w_\omega \\ \dot{w}_p \\ g(3) \end{bmatrix}_{k+1} &= \begin{bmatrix} 1_3 & 0_{3 \times 3} & R_z^\top(\psi)\Delta t & 0_{3 \times 3} & 0 \\ 0_{3 \times 3} & 1_3 & 0_{3 \times 3} & 1_3\Delta t & 0 \\ 0_{3 \times 3} & 0_{3 \times 3} & 1_3 & 0_{3 \times 3} & 0 \\ 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & 1_3 & \Delta t \\ 0_{1 \times 3} & 0_{1 \times 3} & 0_{1 \times 3} & 0_{1 \times 3} & 1 \end{bmatrix} \begin{bmatrix} \theta \\ w_p \\ w_\omega \\ \dot{w}_p \\ g(3) \end{bmatrix}_k + \\ \Delta t \cdot \begin{bmatrix} 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} \\ I^{-1}[w_r_0]_x & I^{-1}[w_r_1]_x & I^{-1}[w_r_2]_x & I^{-1}[w_r_3]_x \\ 1_3/m & 1_3/m & 1_3/m & 1_3/m \\ 0_{1 \times 3} & 0_{1 \times 3} & 0_{1 \times 3} & 0_{1 \times 3} \end{bmatrix}_k &\begin{bmatrix} w_f_0 \\ w_f_1 \\ w_f_2 \\ w_f_3 \end{bmatrix}_k \end{aligned} \quad (4-17)$$

至此，已经完成了设计 MPC 控制器所需的离散时间下单纯刚体模型的线性时变状态方程的构建，为后续模型预测控制器的设计奠定了理论基础。

4.3 模型预测控制器

4.3.1 预测方程

从本质上讲，MPC 和其他基于优化方法的目标是一样的，都是找到最优的控制序列 $u_{[k|k]}, u_{[k+1|k]} \dots u_{[k+n|k]}$ 。在 MPC 中，控制序列被表示为一个紧凑的向量形式，整体求解预测区间内的最优控制序列。

设定预测区间长度为 h，在 k 时刻，系统将预测 h 以内的状态变量的变化趋势（从 $x_{[k|k]}$ 到 $x_{[k+h|k]}$ ）。具体地，由 k 时刻的状态变量 $x_{[k|k]}$ 递推，得到 k+1 时刻的状态变量 $x_{[k+1|k]}$ ，以此类推，递推出未来预测时域内的所有的状态量：

$$\begin{aligned} x(1) &= A_0 x(0) + B_0 u(0) \\ x(2) &= A_1 A_0 x(0) + A_1 B_0 u(0) + B_1 u(1) \\ x(3) &= A_2 A_1 A_0 x(0) + A_2 A_1 B_0 u(0) + A_2 B_1 u(1) + B_2 u(2) \\ &\vdots \\ x(h) &= \left(\prod_{i=h-1}^0 A_i \right) x(0) + \sum_{i=0}^{h-2} \left[\left(\prod_{j=h-1}^{i+1} A_j \right) B_i u(i) \right] + B_{h-1} u(h-1) \end{aligned} \quad (4-18)$$

由上式可以看出，系统 k+h 时刻的状态与初始状态 $x_{[k|k]}$ 以及预测区间内的控制序列 $u_{[k|k]}, u_{[k+1|k]} \dots u_{[k+n|k]}$ 相关。

为了简化系统分析，定义 $X \in R^{13h \times 1}$ ，包含了在 k 时刻预测的所有预测区间内的状态变量 x 。定义 $U \in R^{12h \times 1}$ 矩阵，表示 k 时刻计算得到的所有预

测区间内的控制序列 u 。表示为:

$$X = \begin{bmatrix} x(1) \\ x(2) \\ \vdots \\ x(h) \end{bmatrix} U = \begin{bmatrix} u(0) \\ u(1) \\ \vdots \\ u(h-1) \end{bmatrix} \quad (4-19)$$

借助 X 和 U 矩阵, 系统在预测区间内的预测方程可以紧凑地表示为:

$$X = A_{qp}x(0) + B_{qp}U \quad (4-20)$$

其中, $A_{qp} \in R^{13h \times 13}$ 是状态矩阵, $B_{qp} \in R^{13h \times 12h}$ 是控制矩阵。 A_{qp} 和 B_{qp} 具体形式如下:

$$A_{qp} = \begin{bmatrix} A_0 \\ A_1 A_0 \\ A_2 A_1 A_0 \\ \vdots \\ \prod_{i=h-1}^0 A_i \end{bmatrix} \quad B_{qp} = \begin{bmatrix} B_0 & 0 & \cdots & 0 \\ A_1 B_0 & B_1 & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ \left(\prod_{j=h-1}^1 A_j \right) B_0 & \left(\prod_{j=h-1}^2 A_j \right) B_1 & \cdots & B_{h-1} \end{bmatrix} \quad (4-21)$$

至此, 得到了模型预测控制的预测方程。

4.3.2 期望轨迹

四足机器人单刚体模型在空间中有 6 个自由度, 每个自由度描述位置和速度, 共 12 个变量来描述状态, 本节依次求解这 12 个变量。

本文中, 使用角标 $_d X$ 表示 X 变量的期望值。构建四足机器人的期望轨迹 $_d X$, 即建立状态向量 x , 从 k 时刻开始 $x_{[k|k]}$, 到 $k+h$ ($h>0$)时刻结束 $x_{[k+h|k]}$ 所有期望状态。期望轨迹 $_d X = \left[({}_d x_{[k|k]})^T \ ({}_d x_{[k+1|k]})^T \ \cdots \ ({}_d x_{[k+h|k]})^T \right]^T$ 。

(1) 求解机器人的期望速度。

操作者通过推动摇杆指定机器人的期望速度 ${}^B_d v_{handle} = [{}^B_d v_x \ {}^B_d v_y \ 0]^T$, ${}^B_d \omega_{handle} = [0 \ 0 \ {}^B_d \omega_z]^T$, 该速度是在机身坐标系 $\{B\}$ 下的期望速度, 这种操作方式更符合人们的思维习惯。在行走过程中, 未设置由操作者主动控制的俯仰、横滚和身高动作, 因此期望的俯仰速度、横滚速度以及 z 方向的期望速度均设为 0。

设当前时刻为 k , 则第 h 个控制周期机器人期望速度为:

机身坐标系 $\{B\}$ 下期望速度为:

$$\begin{cases} {}^B_d v_{com}^{k+h} = [{}^B_d v_x \ {}^B_d v_y \ 0]^T \\ {}^B_d \omega_{com}^{k+h} = [0 \ 0 \ {}^B_d \omega_z]^T \end{cases} \quad (4-22)$$

世界坐标系{W}下期望速度为:

$$\begin{cases} {}^W_d v_{com}^{k+h} = R_z({}_d \psi^k) \cdot {}^B_d v_{com}^k \\ {}^W_d \omega_{com}^{k+h} = [0 \quad 0 \quad {}^B_d \omega_z]^T \end{cases} \quad (4-23)$$

(2) 求解机器人的期望位姿。

当前时刻为 k, 由状态估计器获得此时的位置 ${}^W_d p_{com}^k$, 使用递推方法求第 h 个控制周期的期望的质心位置为:

$$\begin{cases} {}^W_d p_{com}^k = {}^W_d p_{com}^k \\ {}^W_d p_{com}^{k+h} = {}^W_d p_{com}^{k+h-1} + {}^W_d v_{com}^{k+h-1} \cdot \Delta t \end{cases} \quad (4-24)$$

另外, 由于 z 方向一般不设置期望速度, v_{com}^{k+h-1} 只会在 ${}^W_d p_{com}^{k+h}$ 的 x 方向和 y 方向积分, 因此需要将 ${}^W_d p_{com}^{k+h}$ 的第 3 项用期望机身高度 ${}_d h^k$ 填充, 即:

$${}^W_d p_{com}^{k+h}(3) = {}_d h^k \quad (4-25)$$

当前时刻为 k, 由 IMU 得到此时的偏航角 ψ_0 , 使用递推方法求第 h 个控制周期的期望偏航角为:

$$\begin{cases} {}^W_d \psi^k = {}^W \psi^k \\ {}^W_d \psi^{k+h} = {}^W_d \psi^{k+h-1} + {}^W_d \omega^{k+h-1} \cdot \Delta t \end{cases} \quad (4-26)$$

另外, 由于俯仰角速度 ω_x , 横滚角速度 ω_y 一般为 0, 因此需要将 ${}^W_d \phi^{k+h}$ 和 ${}^W_d \theta^{k+h}$ 分别用期望机身俯仰角与期望横滚角 ${}_d \phi^k$ 和 ${}_d \theta^k$ 填充, 即:

$$\begin{cases} {}^W_d \phi^{k+h} = {}_d \phi^k \\ {}^W_d \theta^{k+h} = {}_d \theta^k \end{cases} \quad (4-27)$$

本文中期望机身俯仰角与期望横滚角 ${}_d \phi^k$ 和 ${}_d \theta^k$ 由坡度估计给出。

将期望欧拉角整合到一个向量 ${}_d \Theta^{k+h} = [{}^W_d \phi^{k+h} \quad {}^W_d \theta^{k+h} \quad {}^W_d \psi^{k+h}]^T$, 则第 h 个控制周期机器人期望状态为:

$${}_d x^{k+h} = \begin{bmatrix} {}_d \theta^{k+h} \\ {}^W_d p_{com}^{k+h} \\ {}^W_d \omega_{com}^{k+h} \\ {}^W_d v_{com}^{k+h} \\ g(3) \end{bmatrix} \quad (4-28)$$

其中, $g(3) = -9.8m/s^2$, 为常数。

将未来 h 个控制周期的状态组合成一个向量, 这个向量即为期望轨迹 ${}_d X$:

$${}_d X = \begin{bmatrix} {}_d x^{k+1} \\ {}_d x^{k+2} \\ \vdots \\ {}_d x^{k+h} \end{bmatrix} \quad (4-29)$$

至此完成期望轨迹 ${}_dX$ 的求解。期望轨迹还用于确定动力学约束和未来的落足点位置。在实机中，期望轨迹很短，并且经常重新计算，以确保简化动力学在机器人受到干扰时保持准确。

4.3.3 约束条件

输入约束可以分为以下两个约束条件：

(1) 足底反力约束

S_i 为足端是否触地的标志变量。

当腿部处于摆动腿时，足底反力为 0， $S_i=0$ ， $f_i = 0_{3 \times 1}$ 。

当腿部处于支撑腿时，触地时最大足底力为 f_{max} ， $S_i=1$ ， ${}^Wf_i^z \leq f_{max}$ 。

(2) 摩擦锥约束

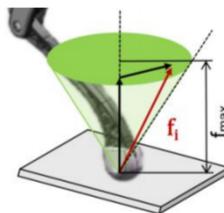


图 4-3 摩擦锥约束示意图

假设地面水平，为保证足底与地面不滑动，足底反力的水平分量 ${}^Wf_i^x$ 和 ${}^Wf_i^y$ ，应当满足如图 4-3 所示的摩擦锥约束，即：

$$\sqrt{{}^Wf_i^x{}^2 + {}^Wf_i^y{}^2} \leq {}^Wf_i^z \quad (4-30)$$

由于上式是非线性约束，不利于计算。在满足使用的情况下，将其拆分为线性约束：

$$\begin{aligned} |{}^Wf_i^x| &\leq \mu {}^Wf_i^z \\ |{}^Wf_i^y| &\leq \mu {}^Wf_i^z \\ 0 &\leq {}^Wf_i^z \end{aligned} \quad (4-31)$$

整理为矩阵形式的不等式约束，得到

$$\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \leq \begin{bmatrix} -1 & 0 & \mu \\ 0 & -1 & \mu \\ 1 & 0 & \mu \\ 0 & 1 & \mu \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} {}^Wf_i^x \\ {}^Wf_i^y \\ {}^Wf_i^z \end{bmatrix} \leq \begin{bmatrix} +\infty \\ +\infty \\ +\infty \\ +\infty \\ f_{max} \end{bmatrix} \quad (4-32)$$

因此四足机器人的输入约束可以表示为：

$$\begin{aligned} & s.t. f_i = 0_{3 \times 1}, \quad \forall S_i = 0 \\ & c_{i,min} \leq C_i \cdot f_i \leq c_{i,max} \end{aligned} \quad (4-33)$$

其中, 第一个约束条件中的 S_i 矩阵表示第*i*条腿处于支撑腿还是摆动腿, 当处于摆动腿时, S_i 赋值为零, 足端离开地面, 不再受到地面反作用力, 将地面反作用力赋值为零。第二个约束条件是摩擦椎约束, 保证足端在地面上不发生滑动。

接下来, 将构建四足机器人模型预测控制 (MPC) 的性能指标 J , 并将其转化为一个二次规划 (QP) 问题进行求解。

4.3.4 二次规划

四足机器人在运动过程中, 目标是最小化目标轨迹跟踪误差, 同时要使机器人在完成轨迹跟踪的条件下尽量降低能耗, 综合以上目标, 可以令代价函数 $J(U)$ 为:

$$\begin{aligned} \min_U J(U) &= (X - {}_dX)^T Q (X - {}_dX) + U^T R U \\ & s.t. f_i = 0_{3 \times 1}, \quad \forall S_i = 0 \\ & c_{i,min} \leq C_i \cdot f_i \leq c_{i,max} \end{aligned} \quad (4-34)$$

其中 $J(U)$ 是优化问题的代价函数, 矩阵 $Q \in R^{13h \times 13h}$ 为状态加权矩阵, 矩阵 $R \in R^{12h \times 12h}$ 为控制加权矩阵。

式(4-34)求解的目标就是找到最优的控制序列 $U_{[k]}$, 使得代价函数 J 最小。为了达成这样的目标, 需要将性能指标 J 用 $U_{[k]}$ 来表示, 并将其写为标准二次规划的形式。得到最优的控制序列 $U_{[k]}$ 之后, 对系统仅施加预测序列中的第一项 $u_{[k|k]}$ 便实现了 MPC 滚动优化控制。

由于现有的二次规划求解软件已经非常成熟, 在处理模型预测控制问题时, 我们的重点将放在如何将问题转化为标准的二次规划形式, 以便利用现有软件高效求解。

将式(4-20)代入式(4-34)中, 转换为二次规划的一般形式:

$$\begin{aligned} \min_U J(U) &= (X - {}_dX)^T Q (X - {}_dX) + U^T R U \\ &= (A_{qp}x(0) + B_{qp}U - {}_dX)^T Q (A_{qp}x(0) + B_{qp}U - {}_dX) + U^T R U \\ &= (A_{qp}x(0) - {}_dX + B_{qp}U)^T Q (A_{qp}x(0) - {}_dX + B_{qp}U) + U^T R U \end{aligned} \quad (4-35)$$

令: ,

$$E = A_{qp}x(0) - {}_dX \quad (4-36)$$

式 (4-35) 可化简为:

$$\begin{aligned} \min_U J(U) &= (X - {}_d X)^T Q (X - {}_d X) + U^T R U \\ &= U^T (B_{qp}^T Q B_{qp} + R) U + U^T (2 E^T Q B_{qp})^T + E^T Q E \end{aligned} \quad (4-37)$$

令:

$$\begin{cases} H = 2(B_{qp}^T Q B_{qp} + R) \in R^{12h \times 12h} \\ g = (2 E^T Q B_{qp})^T = 2 B_{qp}^T Q (A_{qp} x_0 - {}_d X) \in R^{12h \times 1} \end{cases} \quad (4-38)$$

将式 (4-37) 写为二次规划的形式:

$$\begin{aligned} \min_U J(U) &= \frac{1}{2} U^T H U + U^T g \\ \text{s.t. } c_{min} &\leq C U \leq c_{max} \end{aligned} \quad (4-39)$$

其中:

$$c_{min} = 0_{20h \times 1} \quad C = \begin{bmatrix} C_i & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & C_i \end{bmatrix} \quad c_{max} = \begin{bmatrix} c_{i,max} \\ c_{i,max} \\ \vdots \\ c_{i,max} \end{bmatrix} \quad (4-40)$$

构建的 MPC 优化目标函数是二次型凸优化形式, 凸优化问题中局部最优解同时也是全局最优解, 这个特性使凸优化问题在一定意义上更易于求解。

至此, 便可以使用通用的二次规划求解库 OsqpEigen 计算得到控制量矩阵, 即预测时域内所有的足底反力。应用求得的第一组控制量, $U(0) = [{}^W f_0^T \quad {}^W f_1^T \quad {}^W f_2^T \quad {}^W f_3^T]^T$, 即下一时刻最优足底反力。

对于支撑腿, 利用雅可比矩阵 J 将地面反力映射到关节空间。在此过程中, 未考虑腿部的重力因素。关节扭矩指令可通过以下公式计算:

$$\tau = [\tau_0 \quad \tau_1 \quad \tau_2]^T = -J^T \cdot R_B^{{}^W f^T} \cdot {}^W f \quad (4-41)$$

至此, 完成了基于模型预测控制的四足机器人支撑腿扭矩求解。

5 控制系统软件设计及仿真实验

5.1 控制系统软件设计

5.1.1 软件架构

四足机器人控制系统的软件架构如图 5-1 所示。划分为控制框架（ControlFrame）、I/O 接口、第三方库（Third Party Libraries）和 ROS 进程四个部分。

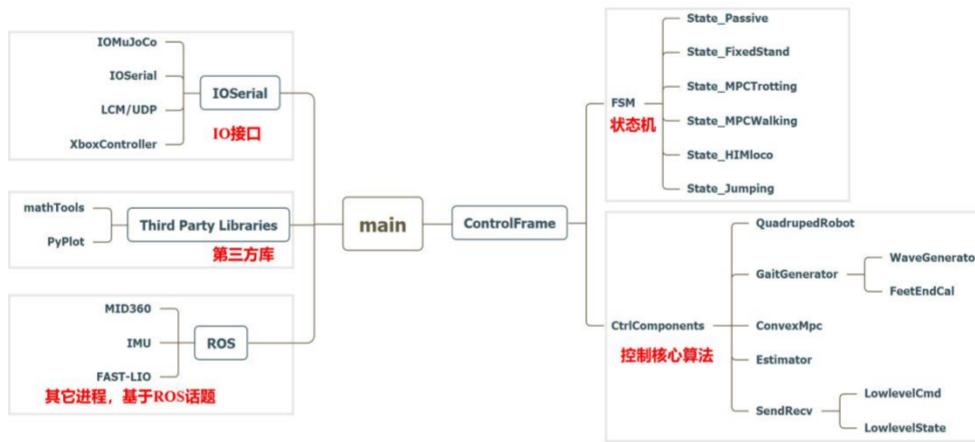


图 5-1 控制系统软件架构图

(1) 控制框架（ControlFrame）为机器人核心逻辑，包含有限状态机（FSM）和控制组件（CtrlComponents）两个子模块。

有限状态机（FSM）：FSM 负责管理机器人运动状态切换，保证各场景下稳定运行。主要包括：

`State_Passive`: 提供带阻尼的软保护状态，降低意外外力造成损伤风险。

`State_FixedStand`: 保持静止姿态与平衡，适用于初始化或待命状态。

`State_MPCTrotting`: 基于模型预测控制（MPC）的 Trot 小跑步态，兼顾稳定性与机动性。

`State_MPCWalking`: 使用相同 MPC 框架优化的 Walk 步态，强调低速稳定行走性能。

`State_HIMloco`: 由深度强化学习算法驱动的智能运动模式。

`State_Jumping`: 用于机器人实现原地跳跃动作。

控制组件（CtrlComponents）：CtrlComponents 集成机器人运动控制必备功能单元，包括：

QuadrupedRobot 模块：提供机器人本体的完整运动学与动力学模型，支撑各类控制算法。

GaitGenerator（步态生成器）：WaveGenerator（步态调度器）：确定各腿摆动和支撑的时间序列。FeetEndCal（落足点计算）：根据期望轨迹与当前状态计算精确的足端落地点。

ConvexMpc 控制器：实时进行凸优化模型预测，计算各支撑腿的最佳地面反作用力。

Estimator（状态估计器）：融合 IMU 和关节编码器数据，估算机器人位置、速度及地面坡度。

SendRecv 模块：完成控制系统与硬件平台的数据交互，其中：

LowlevelCmd：向 12 个电机发送实时控制指令。

LowlevelState：接收电机与 IMU 反馈。

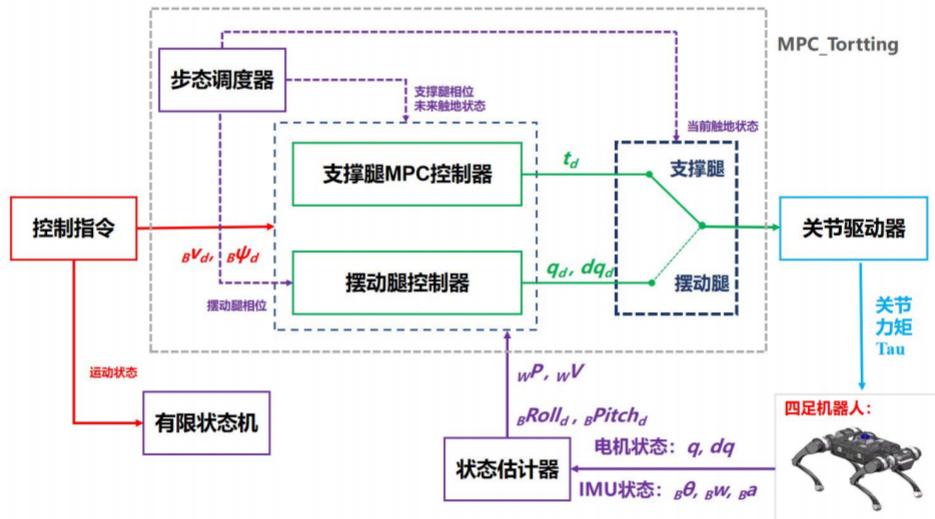


图 5-2 MPCTrotting 控制框架

如图 5-2 所示，MPCTrotting 控制框架，系统将手柄指令映射为角/平动速度，步态生成器输出腿部相位与触地状态；支撑腿 MPC 计算反作用力并执行，摆动腿生成轨迹由 PD 跟踪；状态估计更新位姿并估算坡度，调整姿态。

(2) I/O 通信模块 (IOSerial) 负责主控制系统与机器人硬件平台或仿真环境之间的数据交互。

仿真模式下用 IOMUJOCO 对象与 MUJOCO 交互，控制并反馈 12 个电机与 IMU 数据；硬件模式下用 IOserial 对象通过 USB 转串口与 STM32 通信，发送电机指令并接收状态。

根据需求，还可以创建 LCM 或 UDP 通信接口，用于与副控制板进行数据传输。

(3) 第三方库（Third Party Libraries）模块集成了 Eigen 矩阵运算、Xbox 手柄接口及 Matplotlib/PyPlot 日志与可视化功能。用于系统调试与运行状态分析。

(4) ROS 模块用于融合来自 ROS 系统的进程。该对象会订阅 ROS 中的 IMU_data 和 Odometry 两个话题，从而获取惯性测量单元（IMU）数据和里程计（Odometry）信息，为机器人提供姿态估计与定位辅助。

5.1.2 参数设定

在控制系统软件中，各模块参数设定是确保系统性能稳定、响应迅速以及抗干扰能力强的重要环节。

(1) 足端初始位置确定。通过大量实验发现，足端初始布局会影响四足机器人行走性能：髋关节正下方位置对斜坡、楼梯等复杂地形具有较好适应性，内八布局步态省力但稳定性不足，外八布局虽稳定性较高但能耗偏大。

根据实际的场景，以稳定性为首要目标，同时兼顾地形适应能力，最终确定足端位置如图 5-3 所示。采用髋关节正下方略呈外八的布局，以此提高机器人的整体稳定性。

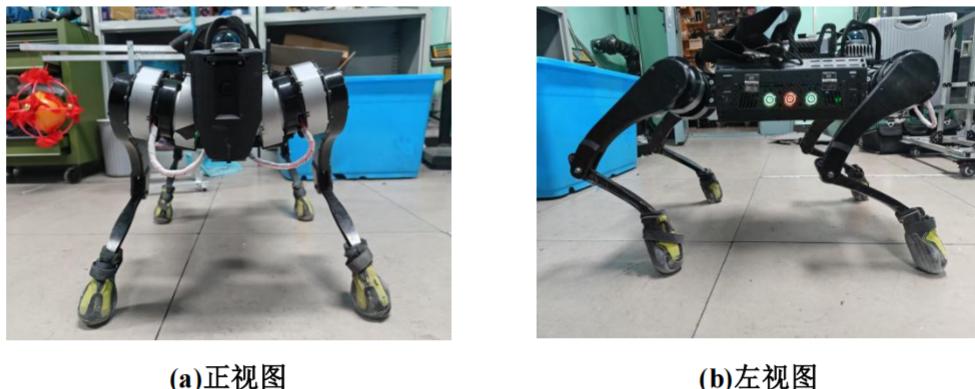


图 5-3 足端位置

(2) 在支撑腿控制器参数设定中，为兼顾系统响应速度与硬件算力，本设计将 MPC 控制器的运行频率设定为 100Hz。在两次 MPC 更新之间，最优反作用力保持不变，关节输出力矩则根据关节角度变化持续调整。

MPC 的预测时域长度需在精度与计算资源之间权衡。预测时域过短可能导致控制器过度关注短期误差而忽略整体轨迹，易引起系统超调。预测

时域过长则增加计算负担，同时简化模型的误差也会随时间积累而降低预测准确性。考虑到四足机器人的运动具有周期性特征，因此将预测时域设定为一个步态周期较为合理。考虑到平台算力，预测时域将被划分为 10 个预测点，即 MPC 的预测步长为步态周期的 1/10。

状态权重矩阵 Q 与控制权重矩阵 R 的设计遵循控制任务优先级原则。 Q 矩阵中较大的元素对应的状态变量将被优先考虑，以减小其输出误差； R 矩阵中较大的元素则会限制控制输入的幅度，从而减少过大的控制动作。最终参数通过仿真与实测反复调优确定，详见表 5-1。

表 5-1 模型预测控制器权重参数

状态量/控制量	值
欧拉角 $\theta = [\phi \theta \psi]$	[0.6 0.9 1.0]
质心坐标 ${}^w p_{com} = [X \ Y \ Z]$	[3.0 3.0 60.0]
角速度 ${}^w \omega = [\omega_x \ \omega_y \ \omega_z]$	[0.0 0.0 0.3]
质心速度 ${}^w \dot{p}_{com} = [v_x \ v_y \ v_z]$	[1.1 1.5 0.0]
反作用力	3×10^{-7}

(3) 在摆动腿控制器的参数设定中，四条腿采用一致的 PD 控制参数。具体而言，每条腿均分别沿机身坐标系{B}中的 X、Y、Z 方向进行轨迹跟踪，并对应设置独立的 PD 参数，详见表 5-2：

表 5-2 摆动腿控制器参数

机身{B}系坐标轴	KP/KD
X 方向	650/6
Y 方向	360/6
Z 方向	270/10

5.2 控制系统仿真实验验证

在 MUJOCO 仿真平台上，以统一参数（步态周期 0.42 秒、支撑腿占空比 0.52、足端高度 0.05m）设计三组实验：

- (1) 对角步态原地旋转实验，验证姿态控制逻辑。
- (2) 对角步态直线行走实验，验证直线行走稳定性。

(3) 对角步态上下斜坡实验，测试地形适应能力。

5.2.1 仿真平台简介

MUJOCO 是一款高性能开源 C++ 物理引擎，为主流的机器人仿真软件。

图 5-4 为 MUJOCO 中的机器人，红色方块为可视化的转动惯量。

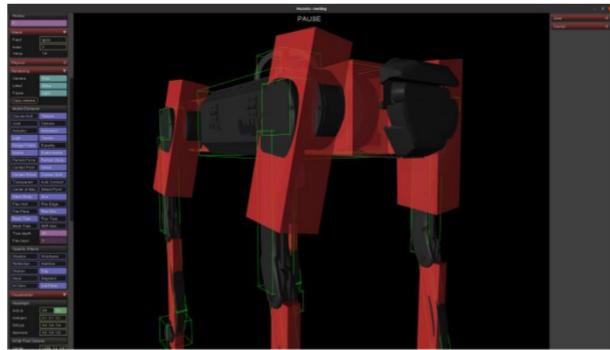


图 5-4 MUJOCO 中的机器人

MUJOCO 提供了完整的仿真生态系统，内置基于 OpenGL 的可视化渲染功能，并支持直接从 XML 格式文件加载模型，使其成为机器人仿真研究的理想平台。接下来将详细介绍在 MUJOCO 中导入机器人模型的具体步骤：

在 SolidWorks 中导出 URDF 文件。首先用 SolidWorks 进行三维建模，然后通过 SW_URDFExporter 插件导出 URDF 格式文件。URDF 描述了机器人的连杆、关节结构及其物理参数（质量、惯性矩阵等）。导出的 URDF 存在参数不完整或不准确的问题，需进行手动校正，确保模型符合实际物理特性。

URDF 转 XML 文件。通过自动化工具（./compile）将优化后的 URDF 转换为 MUJOCO 支持的 XML 格式。在 XML 中，机器人部件用`<body>`标签描述，关节用`<joint>`标签定义。后续完善，还需手动添加陀螺仪、加速度计等传感器配置，并为各关节配置`<actuator>`标签，设置驱动器参数（最大输出力矩、PID 参数等）。

5.2.2 仿真原地旋转实验

如图 5-5 所示，机器人在仿真环境中采用 Tort 步态原地旋转。如图 5-6(a) 所示，Roll 角在 -1.46° 至 1.61° 之间波动，左右倾斜控制良好；Pitch 角在 -2.97° 至 -0.26° 之间，存在轻微后倾；Yaw 角在 -179.99° 至 179.99° 交替变化，完成 360° 旋转。如图 5-6(b) 所示，X、Y 方向速度在 $\pm 0.12\text{m/s}$ 内对

称波动，无明显水平漂移；Z方向速度平均为0.02m/s，垂直运动稳定。整体而言，控制系统能在连续旋转中有效保持平稳姿态。

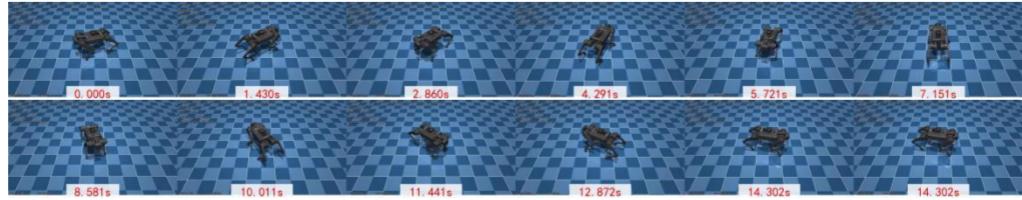


图 5-5 仿真对角步态原地旋转

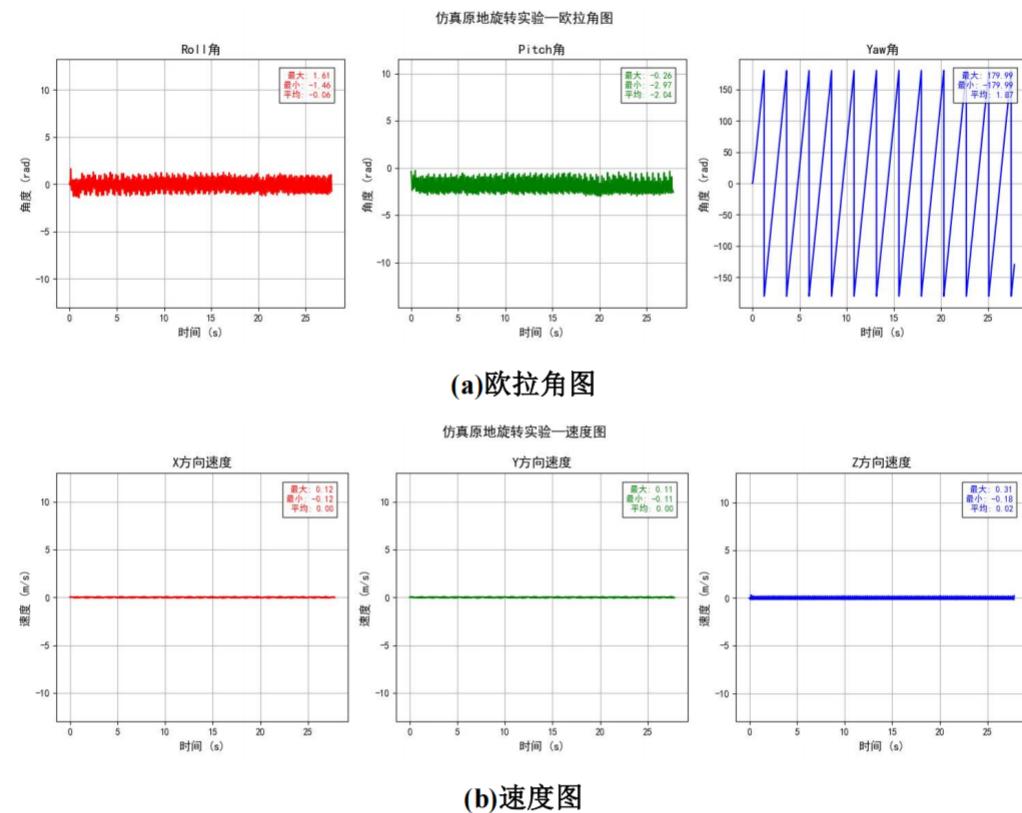


图 5-6 仿真对角步态原地旋转实验数据

5.2.3 仿真直线行走实验

如图5-7所示，机器人在仿真环境中采用Tort步态直线行走100m。如图5-8(a)所示，Roll角在 -2.21° 至 1.73° 之间波动，Pitch角在 -2.09° 至 6.06° 间有轻微后倾但整体影响不大，Yaw角波动范围为 $\pm 1.72^\circ$ ，方向控制精准。如图5-8(b)所示，X方向速度稳定在约1.12m/s，Y方向速度在 ± 0.17 m/s内小幅波动，无明显水平偏移；Z方向速度变化较小，平均0.02m/s。总体来看，控制系统在长距离连续运动中保持姿态平稳，轨迹精准。

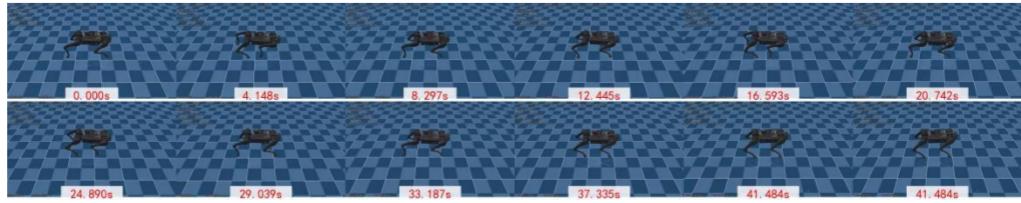


图 5-7 仿真对角步态直线行走

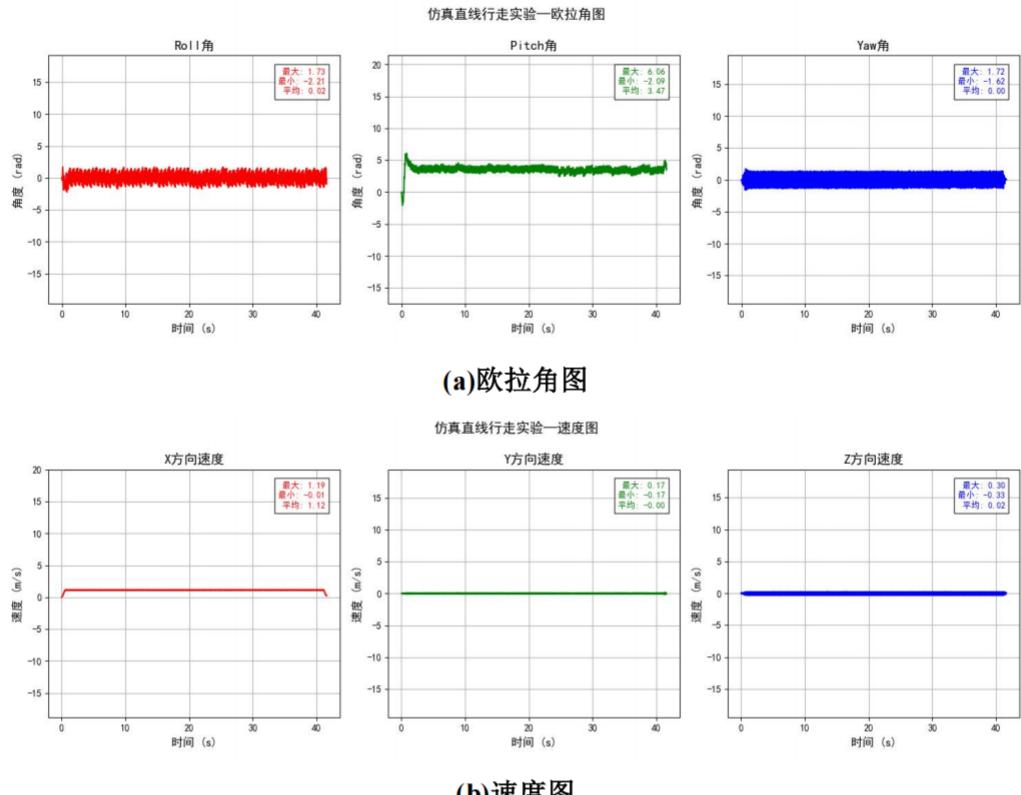


图 5-8 仿真对角步态直线行走实验数据

5.2.4 仿真上下斜坡实验

如图 5-9 所示，机器人在仿真环境中采用 Tort 步态上下斜坡 (15° 坡度)。如图 5-10(a)所示，Roll 角在 -1.90° 至 2.13° 之间波动，左右倾控制精准；Pitch 角在上坡、平地和下坡间变化，最大 15.06° ，最小 -15.15° ，平均 0.78° ，验证了坡度估计算法的有效性；Yaw 角波动在 -1.06° 至 0.65° 之间。如图 5-10(b)所示，X 方向平均速度为 0.18m/s ，前进速度几乎不受地形影响；Y 方向速度波动在 $\pm 0.14\text{m/s}$ 内，横向位移受控；Z 方向速度变化小，平均 0.02m/s ，垂直运动稳定。整体而言，控制算法在斜坡上展现出良好的鲁棒性和自适应能力，确保平稳过渡。

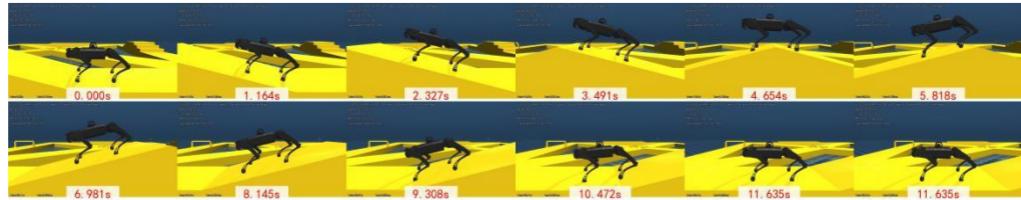
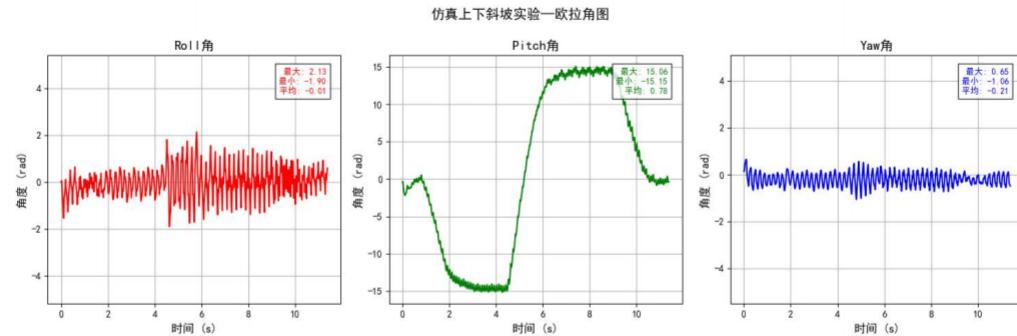
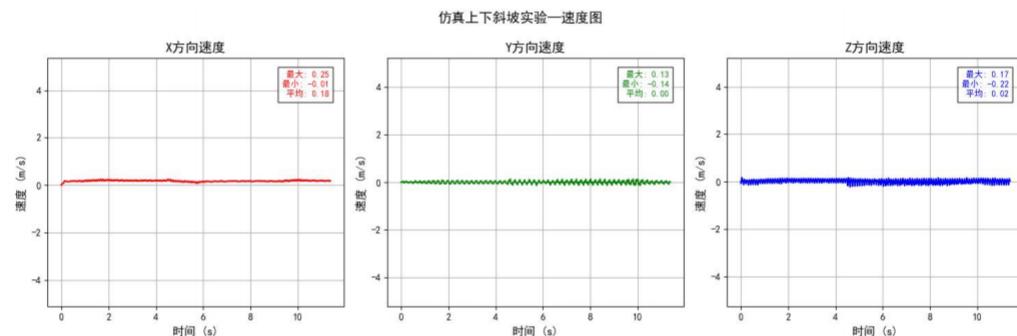


图 5-9 仿真对角步态上下斜坡



(a) 欧拉角图



(b) 速度图

图 5-10 仿真对角步态上下斜坡实验数据

6 控制系统硬件设计及实机实验

6.1 控制系统硬件设计

6.1.1 电气方案

四足机器人电气连接方案如图 6-1 所示。

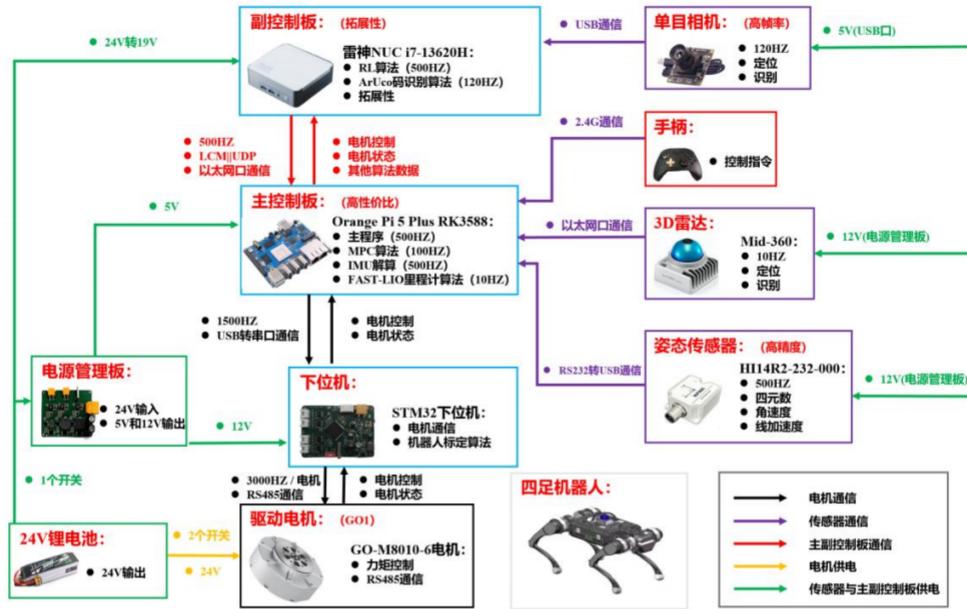


图 6-1 控制系统硬件架构图

(1) 通信设计

系统通信分为电机通信（黑色线）与传感器及主副控制板通信（紫色与红色线）。

电机通信方面：通信内容包括机器人 12 个电机的控制指令 (q 、 dq 、 τ_{au} 、 k_p 、 k_d) 下发与状态反馈 (q 、 dq 、 τ_{au}) 上传。为确保通信带宽充足，将 12 个电机划分成四路独立 RS-485 总线，每一路控制一条腿上的 3 个电机，减轻单路负载并优化机身线缆布局。具体实现方式为电机模组通过 RS-485 与 STM32 下位机数据交互，STM32 再通过 USB 转 Type-C 接口与 Orange Pi 主控板进行高速数据传输。此外，Orange Pi 主控板与雷神 NUC 副控板通过以太网实现高速数据交换。

传感器与主副控制板通信方面：IMU 传感器通过 RS232 转 USB 接入主控板，3D 激光雷达经网线直接连接主控板，二者均采用 ROS 话题与主程序交互；单目相机以 USB 连接副控板，通过 ROS 发布数据后，以 UDP

协议将信息传输至主控板。主副控制板之间通过网线，采用 LCM 与 UDP 协议实时交换数据。

(2) 供电设计

整个系统统一由 24V 锂电池供电，分为电机供电回路（黄色线）和传感器与主副控制板供电回路（绿色线）。

电机供电方面：为避免剧烈运动产生的瞬时大电流烧蚀单个 20A 圆形开关，电机供电分为前腿与后腿两个独立回路，每路单独设置一个 20A 开关进行控制，有效降低负载，提高系统稳定性与寿命。

传感器与主副控制板供电方面：24V 电源单独引出一路，通过 20A 开关分为两支：一支经电源管理板转换为 5V（Orange Pi 主控板）与 12V（IMU 传感器、STM32 下位机、激光雷达）输出；另一支通过 24V 转 19V 模块为雷神 NUC 副控板供电，实现集中管理。

6.1.2 硬件设备

(1) 驱动电机

如图 6-2 所示，四足机器人关节驱动选用 GO-M8010-6 永磁同步电机，集成矢量控制电路与磁编码器，最高扭矩达 23.7Nm，通过 4Mbps RS-485 总线进行高速数据传输，支持 MIT 力矩控制模式。



图 6-2 GO-M8010-6 关节电机

(2) Orange Pi 5 Plus 主控制板

如图 6-3 所示，采用 Orange Pi 5 Plus 作为主控板，配备 RK3588 八核处理器（四核 A76@2.4GHz 和四核 A55@1.8GHz），内置 Mali-G610 GPU 与 6TOPS NPU，搭载 16GB LPDDR4X 内存及 256GB 固态存储，运行带实时内核的 Ubuntu 22.04，稳定高效处理主程序、雷达定位与 IMU 数据采集任务。



图 6-3 Orange Pi 5 Plus 主控制板

(3) 雷神 NUC i7-13620H 副控制板

如图 6-4 所示, 选用雷神 NUC i7-13620H 迷你主机作为副控板, 内置 Intel 第 13 代 i7 处理器(10 核 16 线程, 睿频 4.9GHz)、16GB 内存及 512GB 固态硬盘, 运行 Ubuntu 20.04 系统, 承担强化学习与机器视觉算法计算, 通过 LCM 通信实时向主控板传输数据, 提升系统扩展性。



图 6-4 雷神 NUC i7-13620H 副控制板

(4) STM32 下位机

如图 6-5 所示, STM32H750VBT6 作为下位机, 主频 480MHz, 负责处理 12 个电机数据的角度换算, 具备四路 4Mbps RS-485 串口, 通过 12Mbps USB 虚拟串口与主控板连接, 独立通信线程设计实现 1500Hz 通信频率, 配合主控板完成复杂控制任务。



图 6-5 STM32 下位机实物图

(5) 超核电子 HI14R2-232-000 姿态传感器

如图 6-6 所示, 采用 HI14R2-232-000 姿态传感器, 基于工业级 MEMS 技术, 零偏不稳定为 $1.76^\circ /h$, 数据输出频率 500Hz, 工作温度-40°C至 85°C, 具有 IP68 防护等级。通过 ROS 实时向主控程序传输四元数、角速度及线加速度, Pitch、Roll 角为断电保持绝对角, Yaw 角上电自动归零, 提供精确姿态估计。



图 6-6 超核电子 HI14R2-232-000 姿态传感器

(6) 3D 激光雷达 Mid-360

如图 6-7 所示, Mid-360 激光雷达用于机器人全局定位, 测距范围 40-70m, 360° 水平视场与 59° 垂直视场, 10Hz 点云帧率, 通过 100Mbps 以太网传输数据。采用 FAST-LIO 算法, 并融合 IMU 数据实时插值定位信息, 达到室内 cm 级定位精度。



图 6-7 Mid-360 激光雷达

(7) 单目相机

如图 6-8 所示, 采用 Onsemi AR0234 单目相机, 分辨率 1080p@120Hz, 视场角 100°, 动态范围 83dB, 功耗 250mA@5V, 通过 USB 与 ROS 系统连接, 识别 ArUco 码并结合 IMU、激光雷达实现复杂环境下的精确导航与定位。



图 6-8 单目相机

(8) 电源管理板

如图 6-9 所示, 电源管理板将 24V 锂电池电压转换为多路稳定输出, 分别向雷神 NUC 副控板 (19V)、Orange Pi 主控板 (5V) 以及 STM32、IMU、激光雷达 (12V) 集中供电, 保障系统稳定运行。

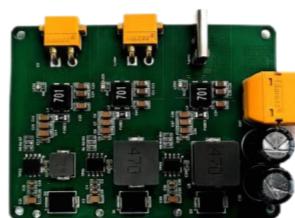


图 6-9 电源管理板

(9) 无线手柄

如图 6-10 所示，采用 2.4GHz 无线手柄进行机器人控制，通过独立通信线程与主控板连接。控制方案明确：左摇杆控制移动方向，右摇杆控制偏航；按键 A 为 Trot 步态、X 为平衡站立状态、Y 为 Walk 步态，B 为阻尼保护状态，提升操作便捷性和指令响应速度。



图 6-10 手柄

6.2 控制系统实机实验验证

为评估四足机器人控制系统性能，在实机平台上设计了三组实验：

- (1) 实机状态估计实验，验证状态估计器的有效性。
- (2) 实机高速度奔跑实验，测试机器人高速奔跑的能力。
- (3) 实机台阶斜坡实验，测试机器人复杂地形的适应性。

6.2.1 实机状态估计实验

如图 6-11 所示，机器人使用 Trot 步态在光滑地板上往返行走 20m，验证状态估计器的有效性。图中展示了状态估计器（红色曲线）与 FAST-LIO（蓝色曲线）估计的轨迹、位置和速度对比，后者接近真实值。如图 6-12(a) 所示，状态估计的轨迹与 FAST-LIO 接近，基本呈直线。如图 6-12(b) 所示，状态估计器在 X 和 Y 方向的速度估计稳定，误差较小。如图 6-12(c) 所示，在 X 方向上，状态估计器与 FAST-LIO 估计结果一致；而在 Y 方向上，存在一定偏差，最大偏差为 0.57m，平均偏差为 0.32m。综上所述，验证了状态估计器在真实运动中的有效性，其位置和速度估计误差较小，特别是在 X 方向上表现出较高的精度。



图 6-11 实机状态估计实验

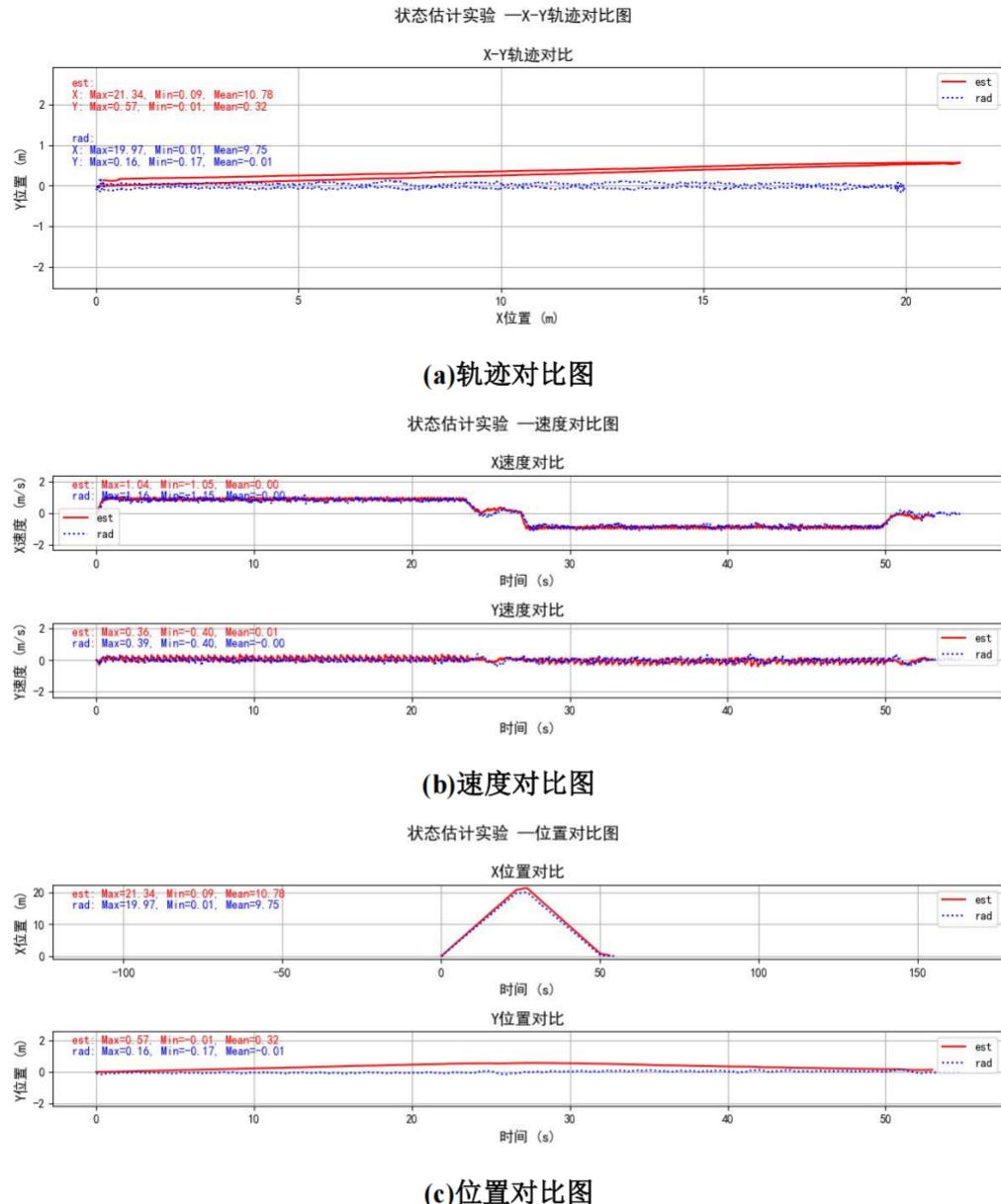


图 6-12 实机状态估计实验数据

6.2.2 实机高速奔跑实验

如图 6-13 所示，机器人使用 Flying-trot 步态以 3.0m/s 的速度，在平坦地面上沿直线奔跑 40m。图 6-14(a)中显示，欧拉角数据表明：Roll 角在约 -12° 至 3° 之间略偏负，Pitch 角波动在 $\pm 10^\circ$ 内且平均值接近 0，而 Yaw 角始终维持在 $\pm 6^\circ$ 内且平均近 0，从而确保了运动方向的精准控制。与此同时，图 6-14(b)所示速度数据表明，X 方向速度稳定在约 3.0m/s（2.9 – 3.5m/s），Y 方向仅有微幅波动（-0.4 至 0.7m/s），而 Z 方向虽因步态冲击呈现周期性变化，但平均值接近 0，显示垂直运动平稳。总体而言，实验

结果充分验证了机器人在高速奔跑过程中的速度稳定性和姿态控制能力。

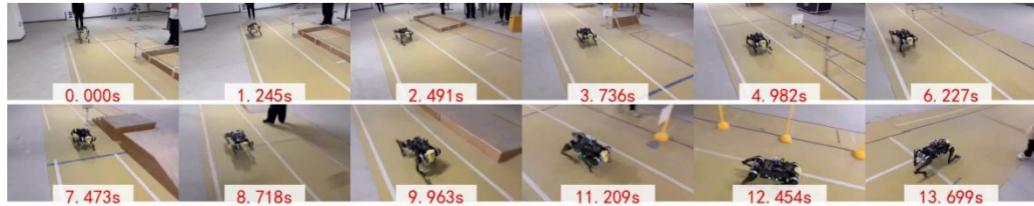


图 6-13 实机高速奔跑实验

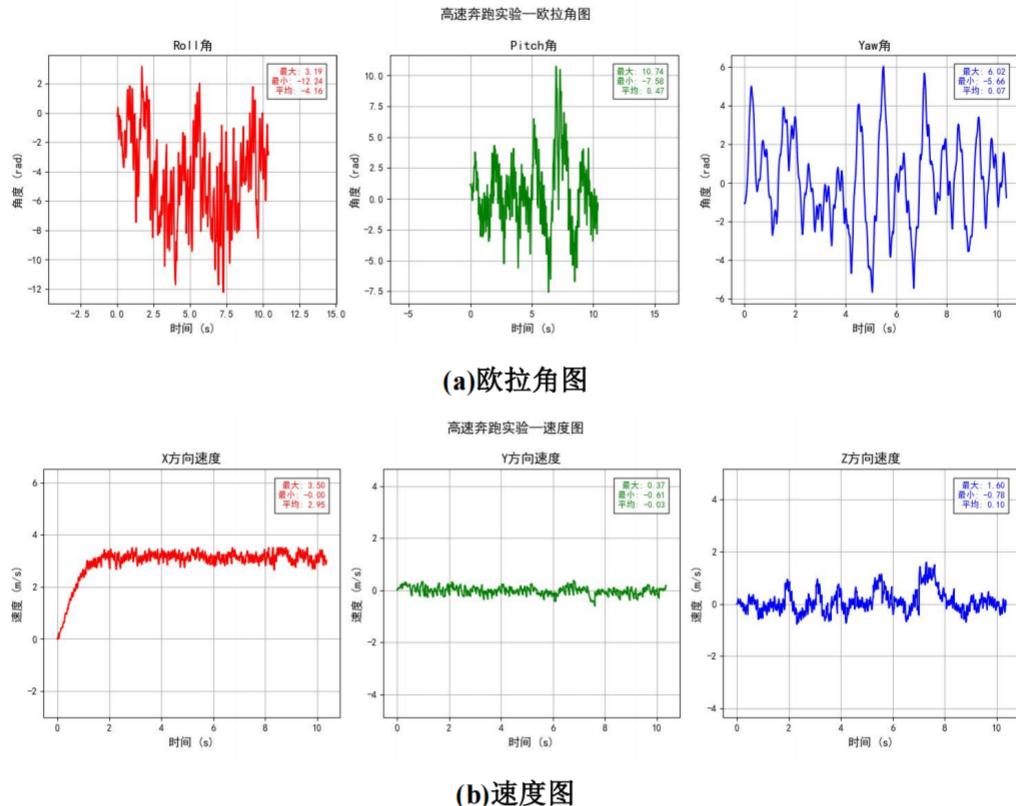


图 6-14 实机高速奔跑实验数据

6.2.3 实机台阶斜坡实验

如图 6-15 所示，四足机器人采用 Tort 步态完成上斜坡和下台阶任务。如图 6-17 所示，机器人使用 Walk 步态完成上台阶后切换 Tort 步态完成下斜坡任务。

从图 6-16(a)和 6-18(a)可以看出，Roll 角虽然存在瞬时波动，但保持在一个较窄的范围内，整体横向控制误差较小；Pitch 角在上坡、平地与下坡过程中呈现明显变化，显示出机器人对坡度变化的响应；Yaw 角波动较小，说明航向控制具备实时修正能力。从图 6-16(b)和 6-18(b)的速度数据来看，X 方向前进速度稳定，并且在步态从 Walk 切换到 Tort 时依然保持平稳；Y

方向速度波动受控，横向位移较小；Z方向速度变化微弱，体现出垂直运动的稳定性。总体而言，数据反映了机器人在复杂地形下的动态响应情况，尽管存在一定的瞬时波动，显示出控制算法的鲁棒性和自适应能力，同时步态切换平滑，有效保证了机器人在斜坡和台阶环境中的平稳过渡。



图 6-15 实机上斜坡下台阶实验

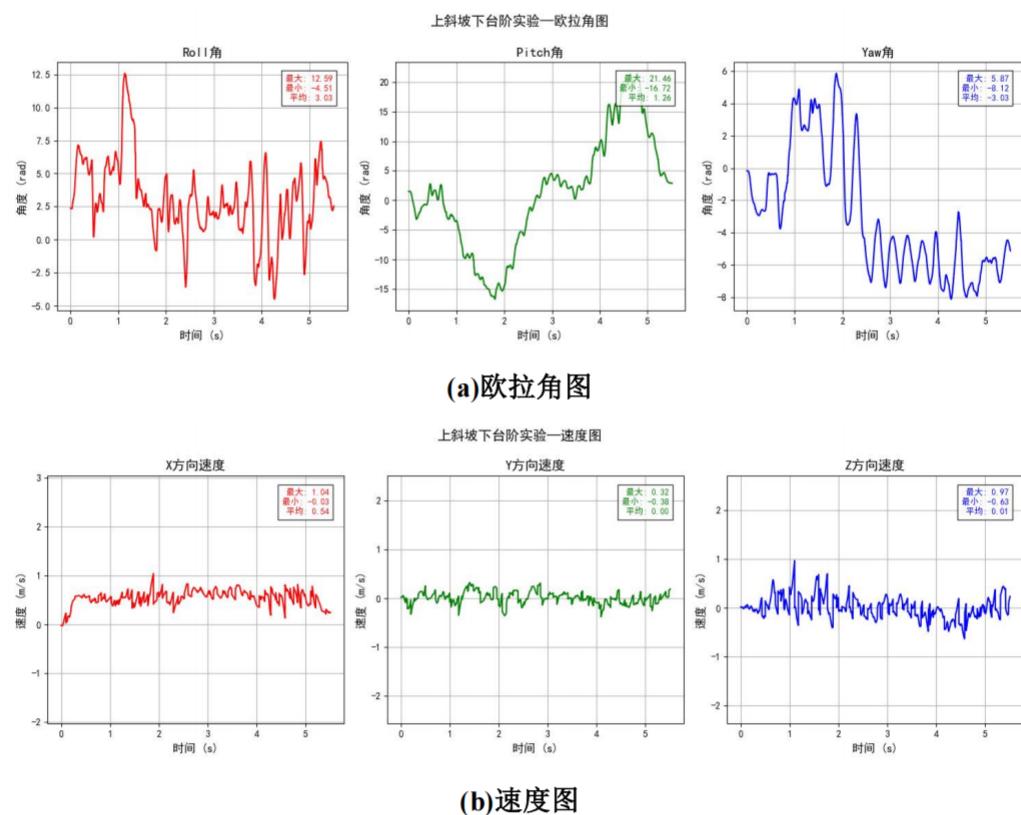


图 6-16 实机上斜坡下台阶实验数据

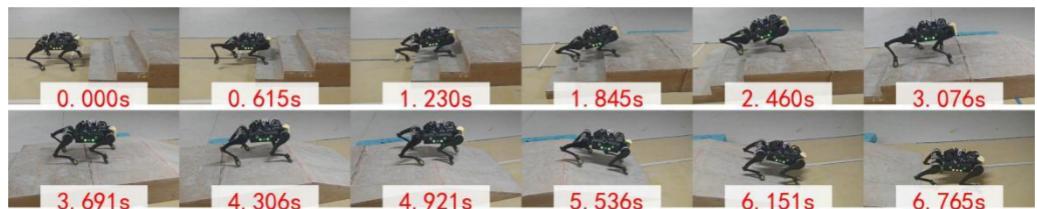


图 6-17 实机上台阶下斜坡实验

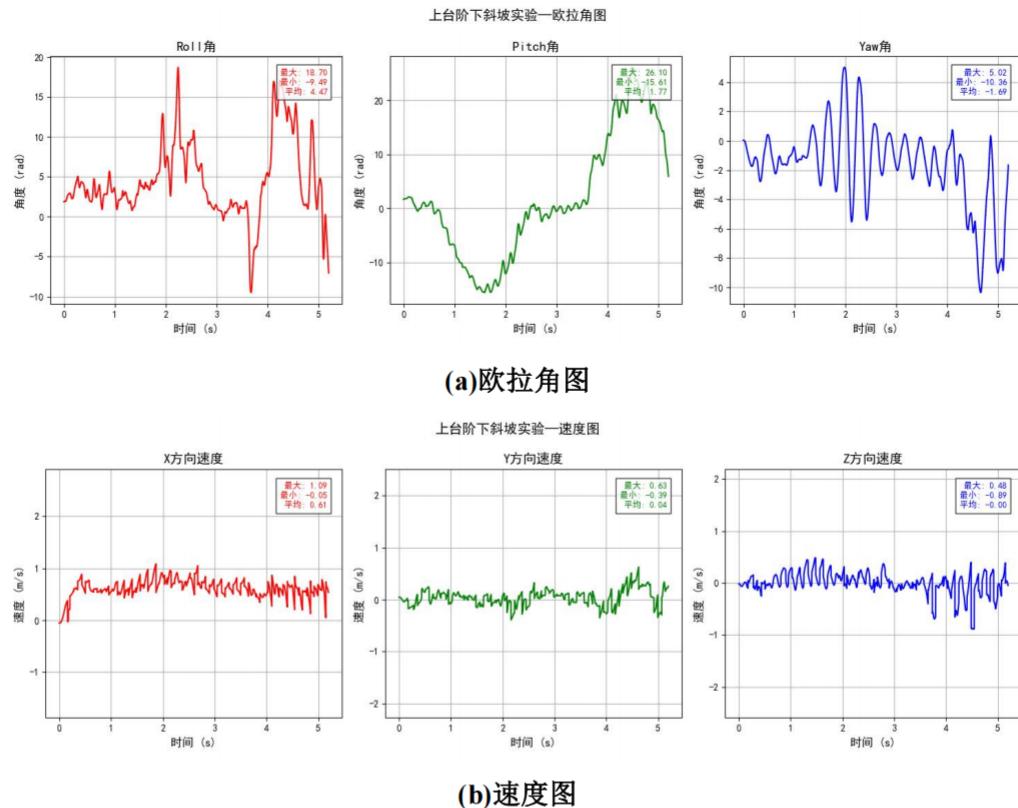


图 6-18 实机上台阶下斜坡实验数据

7 结 论

7.1 主要结论

本文以模型预测控制（MPC）为核心，围绕四足机器人建模、控制理论、软硬件系统设计、仿真验证和实机测试等方面进行了系统性研究。

- (1) 四足机器人建模：完成四足机器人的正、逆、微分运动学建模。
- (2) 状态估计器设计：设计了基于线性卡尔曼滤波的状态估计器，估计了机身位置和速度。并实现了坡度自适应调节。
- (3) 步态与摆动腿控制器设计：构建了基于时间的步态调度器，规划了落足点和摆动腿轨迹。
- (4) 动力学建模与 MPC 支撑腿控制器：建立了系统的动力学模型，设计了期望轨迹和约束方程，并通过 MPC 方法求解最优地面反作用力。
- (5) 仿真验证：在 MUJOCO 仿真平台上，验证了控制逻辑的正确性，包括原地旋转、直线行走和上下斜坡等运动。
- (6) 实机实验：在实际机器人平台上进行状态估计、高速奔跑、台阶斜坡等实验，实现了流畅切换步态，验证了控制方法的自适应性和鲁棒性。

7.2 工作展望

尽管系统在虚拟和实机测试中表现良好，但仍有较大优化空间，未来可从以下方面改进：

- (1) 动力学约束的轨迹规划：现有轨迹规划器仅依据手柄指令生成轨迹，未考虑动力学合理性，未来探索复杂步态时，应结合动力学稳定性约束。
- (2) 多任务全身控制（WBC）：WBC 通过任务优先级优化，引入全身动力学约束，从而提高机器人鲁棒性与响应速度。
- (3) 强化学习控制（RL）：传统控制依赖人工设计，面对复杂环境存在局限性；RL 通过仿真环境中的试错训练，端到端生成优化控制策略。通过融合 IMU 和电机反馈数据，已有强化学习方案实现优于传统控制效果；未来结合激光雷达与相机等多传感器数据进行训练，将进一步提升机器人地形感知能力与运动的灵活性、自然性。

参考文献

- [1] 王旭. 四足机器人运动及稳定控制关键技术综述[J/OL]. 吉林大学学报(工学版), 2024[2025-02-26]. <https://doi.org/10.13229/j.cnki.jdxbgxb.20240722>.
- [2] JUNFENG LONG, WENYE YU, QUANYI LI, et al. Learning H-Infinity Locomotion Control[EB/OL]. 2024[2025-02-26].<https://arxiv.org/abs/2404.14405>.
- [3] Di CARLO J, WENSING PM, KATZ B, et al. Dynamic locomotion in the MIT Cheetah 3 through convex model-predictive control[C]. Institute of Electrical and Electronics Engineers. 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems(IROS), October 01-05, 2018, Madrid, Spain: IEEE, 2018: 1-9.
- [4] 朱秋国. 面向复杂场景的智能电力巡检四足机器人. 浙江省, 杭州云深处科技有限公司, 2023-08-19.
- [5] 杭州宇树科技 . 四足机器人公司介绍 - 宇树科技 [EB/OL]. [2024-03-10]. <https://www.unitree.com/cn/about/>.
- [6] 丁良宏, 王润孝, 冯华山等. 浅析 BigDog 四足机器人[J]. 中国机械工程, 2012, 23(05): 505-514.
- [7] Boston Dynamics. The mobile robot designed for sensing, inspection, and remote operation[EB/OL]. <https://www.bostondynamics.com/spot>.
- [8] PARK H W, WENSING P M, KIM S. High-speed bounding with the MIT Cheetah 2: Control design and experiments[J]. The International Journal of Robotics Research, 2017, 36(2): 167–192.
- [9] PARK H W, WENSING P M, KIM S. Jumping over obstacles with MIT Cheetah 2[J]. Robotics and Autonomous Systems, 2021, 136: 1-12.
- [10] BLEDT G, POWELL M.J., KATZ B., et al. MIT Cheetah 3: Design and control of a robust, dynamic quadruped robot[C]. Institute of Electrical and Electronics Engineers. 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems(IROS), October 01-05, 2018, Madrid,

- Spain: IEEE, 2018: 2245-2252.
- [11] QUAN N, POWELL M J, KATZ B, et al. Optimized Jumping on the MIT Cheetah 3 Robot[C]//International Conference on Robotics and Automation (ICRA). Montreal, Canada: IEEE, 2019:7448-7454.
- [12] KATZ B, CARLO J D, KIM S. Mini Cheetah: A Platform for Pushing the Limits of Dynamic Quadruped Control[C]// 2019 International Conference on Robotics and Automation (ICRA).2019: 6295-6301.
- [13] BUCHANAN R, WELLHAUSEN L, BJELONIC M, et al. Perceptive Whole Body Planning for Multi-legged Robots in Confined Spaces[J]. Journal of Field Robotics, 2020, 38: 68-84.
- [14] LEE J, HWANGBO J, WELLHAUSEN L, et al. Learning quadrupedal locomotion over challenging terrain[J]. Science Robotics, 2020, 5(47): 1-13.
- [15] SONG Zhitao, YUE Linzhu, SUN Guangli, et al. An Optimal Motion Planning Framework for Quadruped Jumping[C]. //IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Kyoto, Japan, 2022: 11366-11373.
- [16] RAIBERT M. Legged Robots That Balance[M]. Cambridge: The MIT Press, 1986.
- [17] PRATT J E, PRATT G A, PRATT E. Virtual Model Control of a Biped Walking Robot[D]. Massachusetts Institute of Technology, 1995: 11-63.
- [18] 谢惠祥. 四足机器人对角小跑步态虚拟模型直觉控制方法研究 (博士学位论文) [D]. 尚建忠, 指导, 长沙: 国防科学技术大学, 2015.
- [19] FOCCHI M, DEL PRETE A, HAVOUTIS I, et al. High-slope terrain locomotion for torque-controlled quadruped robots[J]. Autonomous Robots, 2017, 41: 259-272.
- [20] KIM D, CARLO J D, KATZ B, et al. Highly Dynamic Quadruped Locomotion via Whole-Body Impulse Control and Model Predictive Control[J]. ArXiv, 2019: 1-8.
- [21] 王康. 四足机器人复杂地形越障控制方法研究 (硕士学位论文) [D]. 张秀丽, 指导, 北京: 北京交通大学, 2023.
- [22] 沈雅阁. 针对复杂地形的四足机器人状态估计和运动控制算法研究与

应用（硕士学位论文）[D]. 李学生, 指导, 成都: 电子科技大学,
2022.

- [23] 蔡自兴, 谢斌. 机器人学[M]. 第四版. 北京: 清华大学出版社, 2022:
42-86.

在学取得成果

一、在学期间所获的奖励

2024.10/北京科技大学国家奖学金/北京科技大学

2024.10/北京科技大学优秀三好学生/北京科技大学

2024.10/北京科技大学标兵宿舍长/北京科技大学

2023.10/北京科技大学人民三等奖学金/北京科技大学

2023.10/北京科技大学优秀共青团员/北京科技大学

2023.10/北京科技大学优秀宿舍长/北京科技大学

2022.10/北京科技大学国家励志奖学金/北京科技大学

2022.10/北京科技大学三好学生/北京科技大学

2022.05/北京科技大学人民(新生)奖学金/北京科技大学

2024.11/北航全球科创大赛仿生足式赛道冠军/北航全球科创大赛组委会

2024.07/全国大学生机器人竞赛 ROBOCON 2024 颗粒归仓编程技能挑战赛冠军/全国大学生机器人大赛组委会

2024.07/全国大学生机器人竞赛 ROBOCON 2024 颗粒归仓主赛季军/全国大学生机器人大赛组委会

2024.07/全国大学生机器人竞赛 ROBOCON 2024 颗粒归仓操作技能挑战赛一等奖/全国大学生机器人大赛组委会

2024.07/全国大学生机器人竞赛 ROBOCON 2024 机器马术竞速赛冠军/全国大学生机器人大赛组委会

2024.07/全国大学生机器人竞赛 ROBOCON 2024 机器马术障碍赛冠军/全国大学生机器人大赛组委会

2023.07/全国大学生机器人竞赛 ROBOCON 2023 吴哥之花一等奖/全国大学生机器人大赛组委会

2023.07/全国大学生机器人竞赛 ROBOCON 2023 机器马术竞速赛二等奖/全国大学生机器人大赛组委会

2023.07/全国大学生机器人竞赛 ROBOCON 2023 机器马术障碍赛二等奖

2023.07/全国大学生机器人竞赛 ROBOCON 2023 机器马术障碍赛二等奖

奖/全国大学生机器人大赛组委会

2023.07/2023 年中国大学生机械工程创新创意大赛“矿源杯”物流技术（起重机）创意赛二等奖/机械工程学会

二、在学期间发表的论文

- [1] 张浩, 孙鑫珩, 张希琛. 一种创新型模块化舵轮底盘的设计与控制[J]. 南京理工大学学报, 2024, 48(增刊): 81-88.

三、在学期间取得的科技成果

致 谢

感谢北京科技大学 MEI24 马术全体成员，感谢大家的陪伴与支持，帮助我顺利完成毕业设计。希望今年我们能够再次勇夺 ROBOCON2024 冠军！