

中图分类号：TP242.6

论文编号：10006SY1807103

北京航空航天大学  
硕士学位论文

基于稳定性的仿生四足机器人  
控制系统设计

作者姓名 于宪元

学科专业 机械设计及理论

指导教师 韩晓建 副教授

培养学院 机械工程及自动化学院



# **Control System Design of Bionic Quadruped Robot Based on Stability**

A Dissertation Submitted for the Degree of Master

**Candidate: Yu Xianyuan**

**Supervisor: Associate Prof. Han Xiaojian**

School of Mechanical Engineering & Automation  
Beihang University, Beijing, China



中图分类号：TP242.6

论文编号：10006SY1807103

## 硕 士 学 位 论 文

# 基于稳定性的仿生四足机器人 控制系统设计

作者姓名	于宪元	申请学位级别	工学硕士
指导教师姓名	韩晓建	职 称	副教授
学科专业	机械设计及理论	研究方向	机器人技术
学习时间自	2018年9月1日	起至	2021年5月30日止
论文提交日期	2021年5月25日	论文答辩日期	2021年5月15日
学位授予单位	北京航空航天大学	学位授予日期	2021年6月2日



## 关于学位论文的独创性声明

本人郑重声明：所呈交的论文是本人在指导教师指导下独立进行研究工作所取得的成果，论文中有关资料和数据是实事求是的。尽我所知，除文中已经加以标注和致谢外，本论文不包含其他人已经发表或撰写的研究成果，也不包含本人或他人为获得北京航空航天大学或其它教育机构的学位或学历证书而使用过的材料。与我一同工作的同志对研究所做的任何贡献均已在论文中做出了明确的说明。

若有不实之处，本人愿意承担相关法律责任。

学位论文作者签名：\_\_\_\_\_ 日期： 年 月 日

## 学位论文使用授权书

本人完全同意北京航空航天大学有权使用本学位论文（包括但不限于其印刷版和电子版），使用方式包括但不限于：保留学位论文，按规定向国家有关部门（机构）送交学位论文，以学术交流为目的赠送和交换学位论文，允许学位论文被查阅、借阅和复印，将学位论文的全部或部分内容编入有关数据库进行检索，采用影印、缩印或其他复制手段保存学位论文。

保密学位论文在解密后的使用授权同上。

学位论文作者签名：\_\_\_\_\_ 日期： 年 月 日

指导教师签名：\_\_\_\_\_ 日期： 年 月 日



## 摘 要

在自然界，足式动物具有非凡的运动能力。无论是在陡崖上灵活移动的山羊，还是在沙滩上奔跑的蜥蜴，都展现了足式动物对地形的高度适应能力。受这些自然选择结果的启发，机器人工程师们几十年来一直在尝试实现具有类似动态能力的足式机器人。

与其他形式的移动机器人相比，足式机器人有其特有的控制难点：对于整个机器人来说，在某些运动时刻是过驱动的，在另一些时刻是欠驱动甚至完全没有驱动的，同时这种不同的驱动模式之又经常快速切换，因此很难完美实现足式机器人的动态控制。

随着计算机技术的发展，CPU 算力逐步提高，一些需要大规模计算的控制理论部署到物理机器人上也逐渐成为现实。近年来，基于优化方法的预测控制已经成为足式机器人控制方法的研究热点。本文以模型预测控制（MPC）与全身控制（WBC）为核心，在四足机器人的控制理论、控制系统软硬件设计、结构设计、样机测试等方面进行了深入的研究，主要包括如下内容：

（1）躯干位姿的状态估计。为获取质心在世界系下的位置与速度，利用卡尔曼滤波器设计了一个足底里程计。假设支撑足在地面上不打滑，通过建立本体系下足底速度与世界系下足底速度之间的映射关系，来求解世界系下质心的位置与速度。

（2）基于单刚体模型的模型预测控制。将整个机器人简化为一个单刚体模型并分析其动力学，建立状态方程、预测方程，然后将控制问题转化为一个优化问题，最终求出足底反力。

（3）带优先级的全身控制。将整体的任务按照重要程度分为四个子任务，利用雅可比矩阵的零空间特性，保证低优先级的任务无法影响高优先级任务的控制，从而将因欠驱性而导致的不可控性转移到低优先级任务上。

（4）搭建仿生四足机器人多刚体动力学。在世界系与机器人躯干之间添加虚拟的 6 自由度关节，从而将仿生四足机器人抽象成由 13 个关节连接的 13 个刚体的组合，并建立其动力学方程。利用组合刚体法和牛顿欧拉法求解动力学方程中的相关参数。

（5）在仿真环境下搭建虚拟样机，仿真验证控制系统的稳定性。在 webots 平台上搭建了仿生四足机器人虚拟样机，并将控制系统算法转化为具体控制系统程序，实现了仿生四足机器人全方向行走、非结构化路面行走、侧向冲击后恢复稳定等功能。

（6）搭建并调试物理样机，验证控制系统稳定性。首先选择传动方式，设计躯干及腿部结构，对关节零部件进行有限元仿真；然后装配仿生四足机器人，并布置控制系统

硬件电路；最后将控制系统程序烧录至机器人并实验测试。在物理样机上实现了虚拟样机的所有功能，验证了控制系统的稳定性。

**关键词：**四足机器人，模型预测控制，全身控制，刚体动力学算法

## Abstract

Animals in nature are extremely expert in movement, and either goats that move agilely on the abrupt cliff or lizards that run on the beach showcase high adaption to various terrains. Inspired by those animals in nature, robot engineers have been attempting for decades to develop a legged robot with similar dynamic capabilities.

Compared with mobile robots in other forms, legged robots have their own unique problems related to control: the entire robot is over-driven at some moments, or under-driven or even not driven at other moments; additionally, a fast shift is required between drive modes, increasing the difficulty in controlling the legged robots in a dynamic and perfect manner.

Along with the advancement of computer techniques and the gradually enhanced CPU computing power, it has gradually come true that part of control theories that require large-scale computing have been applied to physical robots. In recent years, optimization-methods-based predictive control has become the focus of studies on control methods of legged robots. In this paper, Model Predictive Control (MPC) and Whole Body Control (WBC) are centered on conducting an in-depth study on control theories of quadruped robots, control system software and hardware design, structural design, prototype testing, etc. The main contents are as follows:

(1) State estimation of robot body position and posture. To acquire the position and the velocity of the center of mass in the world coordinate system, a plantar milemeter is designed with a Kalman filter. Assuming that the supporting feet do not slip on the ground, the mapping relation in foot speed between the present coordinate system and the world coordinate system is established to find out the position and velocity under the world system.

(2) Model Predictive Control based on a single rigid body model. By simplifying the entire robot into a single rigid body model and analyzing its dynamics, state equations and prediction equations are constructed. Then, the control problem is converted into an optimization one to finally solve the plantar reaction.

(3) Whole Body Control with priority. The overall tasks were divided into four sub-tasks according to their degrees of importance. The null space of the Jacobian matrix was utilized to ensure that sub-task with low priority should not influence the control of the sub-task with high priority. This way made it sure that uncontrollability arisen from underactuation could be transferred to the sub-task with low priority.

(4) Multi-rigid-body dynamics for building a bionic quadruped robot. Virtual joints with six DOFs was added between the World Coordinate System (WCS) and robot body, so that the bionic quadruped robot can be considered as a combination of 13 rigid bodies with 13 connected

joints in the abstract. Meanwhile, its dynamic equation was set up. The rigid-body assembling and the Newton-Euler formulation were used to obtain the relevant parameters of the dynamic equation.

(5) Setting up a virtual prototype in simulation environment to verify the simulation results concerning the stability of the control system. A virtual prototype of the bionic quadruped robot was set up on webots platform, and then the robot's control system algorithm was transferred into specific control system procedures. This way helped the bionic quadruped robot achieve the all-direction walking, unstructured pavement walking, and stability recovery after a lateral impact and other functions.

(6) Establishing and debugging the physical prototype, and verifying its control system stability. Firstly, a driving mode was chosen to design its body and leg structures, and finite element simulation was made with its joint-related components. Secondly, the bionic quadruped robot was assembled and the hardware circuit of its control system was arranged. Lastly, the control system program was burned into the bionic quadruped robot and tested. All functions of the virtual prototype were realized on the physical equal, proving the stability of the robot's control system.

**Key words:** Quadruped Robot, MPC, WBC, Rigid Body Dynamics Algorithm

# 目 录

第一章 绪论.....	1
1.1 研究意义.....	1
1.2 足式机器人控制理论发展简史.....	3
1.3 本文主要研究内容及章节安排.....	8
1.4 本章小节.....	9
第二章 仿生四足机器人控制系统总体介绍.....	10
2.1 有限状态机.....	11
2.2 坐标系及刚体的位姿描述.....	12
2.2.1 刚体坐标系.....	12
2.2.2 刚体平动的位置与速度描述.....	14
2.2.3 刚体转动的位置与速度描述.....	15
2.3 步态调度器.....	20
2.4 状态估计器.....	22
2.4.1 状态方程.....	24
2.4.2 观测方程.....	25
2.5 质心轨迹生成器.....	27
2.5.1 基于水平速度命令的质心轨迹规划.....	27
2.5.2 地形坡度估计.....	28
2.6 摆动相规划.....	30
2.6.1 落足点规划.....	30
2.6.2 足底轨迹规划.....	33
2.7 本章小节.....	35
第三章 仿生四足机器人模型预测控制 (MPC) .....	36
3.1 模型预测控制在间歇欠驱系统中的必要性.....	36
3.2 单刚体动力学建模.....	37
3.2.1 平动动力学与转动动力学.....	38
3.2.2 惯性张量的转动变换.....	38
3.2.3 足底反力约束.....	39
3.2.4 近似欧拉方程.....	40
3.2.5 角速度到 ZYX 欧拉角微分的映射.....	41

3.2.6 单刚体动力学模型及其离散化.....	42
3.3 预测方程.....	43
3.4 带约束的二次凸优化.....	44
3.5 本章小节.....	46
<b>第四章 仿生四足机器人全身控制 (WBC) .....</b>	<b>47</b>
4.1 带优先级全身控制基本原理.....	49
4.1.1 带优先级的双任务全身控制.....	49
4.1.2 带优先级的多任务全身控制.....	49
4.1.3 位置级全身控制.....	50
4.1.4 加速度级全身控制.....	51
4.2 全身控制在四足机器人控制系统中的应用.....	52
4.2.1 浮动基座机器人简介.....	52
4.2.2 子任务划分及其具体实现.....	53
4.3 松弛优化.....	55
4.4 本章小节.....	58
<b>第五章 多刚体系统动力学算法.....</b>	<b>59</b>
5.1 六维空间向量.....	59
5.1.1 运动向量空间与动力向量空间简介.....	59
5.1.2 Plücker 坐标系下的坐标变换.....	61
5.1.3 叉乘与微分.....	63
5.1.4 空间加速度.....	64
5.1.5 空间惯量与运动方程.....	65
5.1.6 六维空间向量的性质.....	67
5.2 多刚体系统的运动学建模.....	67
5.2.1 运动学树.....	67
5.2.2 关节模型.....	68
5.2.3 足底接触点.....	70
5.2.4 递推正运动学.....	71
5.2.5 全身控制雅可比矩阵 $J_1$ 、 $J_4$ 及各级任务 $J_i \dot{q}$ 项.....	71
5.3 多刚体系统的动力学建模.....	74
5.3.1 递推牛顿欧拉法.....	75

5.3.2 组合刚体法求 $M$ 矩阵.....	77
5.4 本章小节.....	79
<b>第六章 控制系统软硬件设计及仿真.....</b>	<b>80</b>
6.1 控制系统硬件设计.....	80
6.1.1 电气连接方案.....	80
6.1.2 主要硬件设备介绍.....	81
6.2 控制系统软件设计.....	84
6.2.1 控制系统软件各模块介绍.....	84
6.2.2 第三方库.....	87
6.2.3 CPU 算力分配.....	89
6.3 控制系统仿真实验验证.....	90
6.3.1 webots 仿真平台介绍.....	90
6.3.2 峰值关节扭矩实验.....	90
6.3.3 对角步态踏步稳定性实验.....	92
6.3.4 对角步态侧向冲击实验.....	92
6.3.5 多种步态下侧向冲击实验.....	93
6.4 本章小节.....	94
<b>第七章 物理样机设计与测试.....</b>	<b>95</b>
7.1 物理样机结构设计.....	95
7.1.1 腿型布置形式.....	95
7.1.2 传动方式选型.....	96
7.1.3 躯干设计.....	97
7.1.4 足底结构设计.....	97
7.2 关键零部件仿真.....	98
7.2.1 电机座设计与仿真验证.....	98
7.2.2 内曲柄与外曲柄仿真验证.....	98
7.2.3 大腿壳仿真验证.....	99
7.2.4 连杆仿真验证.....	99
7.3 整体装配图.....	99
7.4 零件加工与装配.....	100
7.5 物理样机测试.....	100
7.5.1 对角步态原地踏步实验.....	100

7.5.2 直线行走实验.....	100
7.5.3 非结构化路面行走实验.....	101
7.5.4 侧向冲击实验.....	101
7.5.5 顶倒立水瓶实验.....	102
7.5.6 四脚共线实验.....	103
7.6 本章小结.....	103
总结与展望.....	104
参考文献.....	106
攻读硕士学位期间取得的学术成果.....	109
致 谢.....	110

# 第一章 绪论

## 1.1 研究意义

移动机器人的移动能力通常受其物理构造所限制，在许多场景下无法取代人类进行工作。比如，普通的轮式机器人无法爬楼梯、穿越非常不平坦的地形；四旋翼无人机续航能力、负载能力一般不会太好；足式机器人往往在保持平衡与快速响应扰动方面表现不佳。如果相关研究人员能够攻克这些难题，机器人技术将对世界产生巨大的影响。随着机器人业务能力的增强，它们可以完成非常危险或者人类根本不可能完成的工作<sup>[1]</sup>。

因此，面临着复杂、未知、多变的非结构环境，移动机器人应具有良好的适应性<sup>[2]</sup>。受科幻作品影响，当普通大众想到机器人时，脑海里浮现的通常是功能齐全的机器人在现实世界里流畅运动的画面，它们能够非常智能地理解人类的指令、与物理世界交互，在任何场景下自主完成任务而无需人工操纵。尽管这一目标必将在未来的某天成为现实，但在那之前还有很多工作要做。在早期的机器人领域中，绝大多研究人员都专注于高精度、单任务的机器，用于代替人类做重复性工作，如车辆装配生产线。这些机器人通常会执行预先编排好的动作并尽可能避免与环境的交互作用。这种机器人与自主移动机器人形成了鲜明的对比：自主移动机器人往往不受事先编排好的动作约束，具有主动探索环境、主动定位、导航的能力，并且积极主动与物理环境接触<sup>[3]</sup>。

目前，已经有各式各样的机器人应用于汽车制造、机械加工、焊接、上下料、磨削抛光、搬运码垛、装配、喷涂等工作场景<sup>[4]</sup>。但在大多数情况下，机器人依然被用于运行重复动作、执行单一任务，而不是具有通用用途，有些甚至需要提前部署它们的工作环境，如轨道、UWB 系统、导航线等<sup>[5]</sup>。但是在抢险救灾情景下，比如地震、核电站事故，地形和环境将是高度非结构化、未知且危险的。在核电站事故这种高度非结构化环境下，如果能够部署机器人来调查事故或者展开搜救任务，而不是直接派人类前去，是实现全自主移动机器人之前的一个小目标，但在那之前还有技术难题需要解决。

相较于轮式、翼式等移动机器人，研究四足机器人具有更加深远的意义，具体表现为如下几个原因。

第一个原因是四足机器人具有灵活的机动性。在特殊应用场景下，需要能够在一般车辆无法通行的复杂路面上行驶的机器人。由轮子驱动的车辆在轨道、公路或其他预先铺设好的路面上具有出色的表现，但地球表面的大多数陆地是没有铺设道路的。在松软地面或崎岖不平的地形上行驶时，车轮的移动效率大大降低甚至无法移动，而足式机器

人可以在非结构化和恶劣的环境中工作<sup>[6]</sup>。因此，需要研发由腿驱动的交通工具，以便能够探索更多的地方。

四足机器人在崎岖地形中比轮式车辆和履带式车辆具有更好的机动性，原因是它们可以使用互相独立的落脚点来提供支撑力和牵引力<sup>[7]</sup>。而轮式机器人需要连续的支撑路径。因此，足式机器人的灵活性通常受限于可达地形中最佳立足点，而轮式机器人则受到最差的地形限制。如图 1，爬梯子就是一个很好的例证：车轮无法在不连续的支撑路径上移动，而有限的支撑点就可以满足腿足式系统的移动。



图1 足式系统可以轻而易举地处理非连续支撑路径

四足机器人的第二个优势是躯干的轨迹与足底的轨迹是可以解耦的，从而可以实现类似主动悬挂的避振效果<sup>[8]</sup>。在行走过程中，腿通常不会伸到最长，即有一定的长度储备。当踩到石头或者踩到坑里时，机器人可以通过调整足底相对于机身高度方向的长度，来实现机身与负载自由而平稳地移动。而轮式机器人使用被动悬挂，通常只能尽可能减小这种因地面崎岖而产生的振动。因此，作为一种运载工具，四足机器人相比于轮式机器人在稳定性上限上具有明显优势。

研究四足机器人的第三个原因，是可以借此理解人类和动物的运动机制。理解自然、解释自然、改造自然是人类自古以来的天性。对于普通大众来说，习以为常的走、跑、跳等动作似乎是轻而易举就能实现的。但是对于机器人领域，尤其是足式机器人领域的研究人员来说，只需要从机械、感知、运动控制的角度稍加思索，就会被人类奔跑、舞蹈、投掷、游泳等控制身体移动行为的复杂性所震撼。

尽管人类非常擅长跑步，但是在理解跑步的控制原理方面，人类还处于初级阶段。亲自制作一个能够由腿驱动的机器人，是深入理解足式运动原理，并验证理论的不二选择。当四足动物和四足机器人执行类似的运动任务时，它们的控制系统和机械结构必然

会遇到相似的问题。通过实际制作足式机器人，可以发现新的问题，对这些问题有新的认知，并研究出可能的解决方案。

制作四足机器人离不开工程和科学领域的共同进步。足式机器人需要大扭矩、高精度、高带宽的驱动模组<sup>[9-11]</sup>，需要低惯量且有一定强度的腿部结构<sup>[12, 13]</sup>，需要有一定感知能力并通过感知地形来寻找良好的落足点，需要欠驱时变系统的控制理论。这些工作中的大部分还没有被很好地解决，但是相关研究正在稳步推进。如果这些研究取得成功，将极大地推进足式机器人发展。这种足式运载工具将在工业、农业和军事应用中发挥巨大作用。

## 1.2 足式机器人控制理论发展简史

一般来说，足式机器人同一时刻支撑腿的数目越多越容易控制。因此从难易程度上来说，四足机器人对控制系统的鲁棒性需求要远低于两足机器人，同时两足机器人可以使用的控制理论，稍加修改就可以部署在四足机器人上。对于足式机器人来说，无论是单足机器人还是两足或四足机器人，其控制难点都是类似的，因此其控制方法本质上都是相同的，只不过在腿的数量以及控制难易程度上有所不同。在足式机器人发展史上，两足机器人与四足机器人的控制理论并没有特别的泾渭分明。

不同于传统具有固定基座的工业机器人，足式机器人具有一个六自由度的欠驱浮动基座，而该基座却是控制系统的主要被控对象<sup>[14]</sup>。因此，足式机器人通常通过控制各个关节的扭矩，来调整与环境的交互作用，进而间接维持欠驱浮动基座的姿态和速度<sup>[15-18]</sup>。不同时期的研究学者对于这类特殊的机器人系统的控制策略也有多种思路，下面将简略介绍足式机器人发展史上几个具有里程碑意义的控制理念。

**具有固定运动模式的连杆机构。**关于足式运动的研究，可以追溯到两个多世纪以前，当时美国加利福尼亚州州长利兰·斯坦福委托埃德沃德·迈布里研究马在小跑时是否存在四只脚同时离开地面的时刻。1878年，迈布里奇用当时的高速相机，抓拍到了四足腾空相瞬间。之后，迈布里奇继续记录了包括人类在内的40多种哺乳动物的行走和奔跑行为。他的摄影资料至今仍有相当大的价值，是足式运动研究的里程碑。对足式机器人的研究也起源于迈布里奇那个时代<sup>[8]</sup>。

19世纪70年代俄罗斯数学家切比雪夫设计了一种巧妙的行走机构，如图2(a)所示，它使用一套连杆机构，在连续转动的动力源驱使下，连杆末端划出一个直线加弧线的连续轨迹，可以作为足式机器人的足底轨迹。在接下来的八九十年间，相关研究学者把设

计足式机器人的工作重心放在设计不同形式的连杆上, 试图通过改变连杆的构造, 来改变足底轨迹, 进而改变足式机器人的运动特性, 图2(b)展示了一种典型的连杆驱动机器马。但是, 这些机器人的性能受到其固定运动模式的限制, 无法在机身受到干扰时做出响应, 也无法适应地形的变化。到 20 世纪 50 年代, 研究学者已经认识到, 提供重复运动的连杆机构无法针对外界环境的变化做出反应, 真正的足式机器人必须有反馈系统<sup>[8]</sup>。

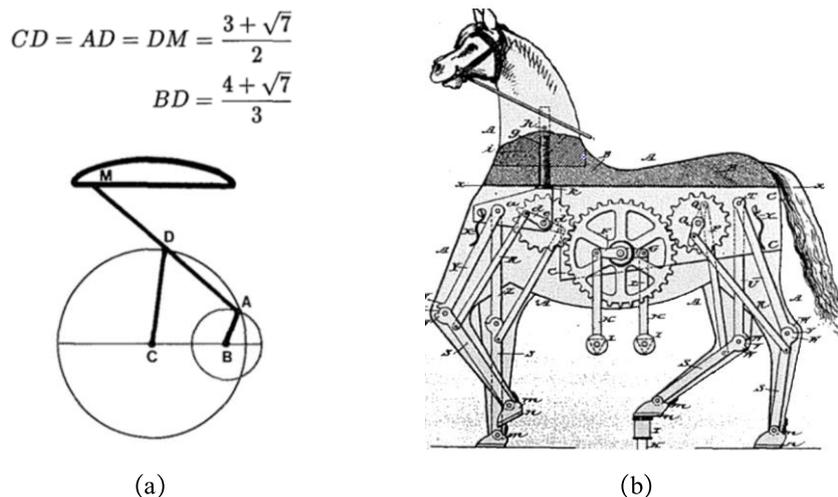


图2 可以实现抬腿与前进的切比雪夫机构(a), 早期机器马(b)

**可自由设计足底轨迹的位置控制。**部分研究学者<sup>[19, 20]</sup>采用高减速比大扭矩电机, 或者液压驱动元件设计串联的机械臂, 用来作为四足机器人的腿。通过规划足底轨迹, 采用逆运动学得到关节轨迹, 最后发送给电机执行。相对于具有固定运动模式的连杆机构, 这种方式大大加快了验证不同足底轨迹可用性的效率。然而, 尽管尝试了各种足底轨迹来试图减少足底与地面的冲击, 但是其控制理念与 19 世纪固定运动模式的连杆机器人并无本质区别。在当前工业界, 比如使用了机械臂的汽车装配流水线, 几乎都在使用位置控制: 事先规划好机器人末端的轨迹, 通过逆动力学得到实时的关节位置, 然后发送给电机位置指令, 以驱动机器人在封闭的、规则的、已知的环境下运动。有些时候也会加入视觉系统, 让机器人对外界环境具有一定的适应性。

这种控制方式要求机器人尽可能少的与外部环境进行接触, 当机器人在完成一些与环境存在力的作用的任务时, 比如零件打磨、汽车外壳抛光, 单纯的位置控制就会因位置误差而产生较大的不可控作用力。在足式机器人控制上, 控制系统更注重机器人与外部环境的作用力, 这是因为除重力外, 地面反力是浮动基体机器人所受到的唯一的外力。

从电机用户来看, 采用位置控制, 仅仅需要对电机发送位置指令, 就可以控制电机移动到期望位置。然而在电机内部, 当接收到位置指令后, 实际上也存在一个含反馈的

位置环，用于控制电机的位置。该位置环通常不对电机用户开放，这意味着电机内部的位置环对于机器人控制系统来说是一个黑箱。在机器人与外部有作用力的时刻，难以保证该黑箱的行为对整体的机器人控制是有利的。

由于在机器人控制系统和电机本体之间存在一个电机位置环，就必须给电机位置环一定的响应时间，这意味着发送至电机的位置信号频率不能太高。因此，位置控制的足式机器人往往不会有较高的控制带宽。

**由训练有素的人类驾驶。**一种颇为取巧的控制方法是由人类驾驶机器人。上个世纪60年代中期，美国通用电气公司制作了一辆四足卡车（如图3所示），这台机器高11英尺，重3000磅，采用液压驱动<sup>[8]</sup>。司机的四肢与卡车四条腿的控制把手相连，同时也存在由卡车传向司机的力反馈传感器。每当司机用卡车的腿推动障碍物时，力反馈系统让司机觉得就像他自己在推障碍物一样。经过不到一天的训练操作，司机就可以流畅地操作这台机器。尽管这台机器需要经过培训的人类来控制，但这台足式机器是足式机器人控制系统研究史上一个重要里程碑。



图3 通用电气设计的四足卡车

**Raibert 的三通道解耦控制方法<sup>[8]</sup>**。随着计算机技术与自动控制技术的发展，将控制系统集成到计算机软件当中，控制系统接受传感器的信号，并发送指令给执行器，这种闭环控制的思想逐步在足式机器人领域中扩散开来。上个世纪八十年代，Marc Raibert 在计算机上搭建控制系统，分别实现了单足跳跃机器人、双足机器人和四足机器人稳定运动控制，如图5所示。为了简化工程任务量，并放大足式机器人的动态特性，Raibert 首先制作了一台单足跳跃机器人，它像袋鼠一样通过跳跃来维持系统的移动和平衡。该机器人主要包括本体和腿两个部分，本体上固定着传感器、计算机以及仪表等部件，腿则可以通过伸缩来改变长度。传感器可以测量本体的俯仰角以及臀部的角度。

在控制上，Raibert 开创性地提出了三通道解耦控制思想。对机器人的高度、姿态和

移动速度分别设计了控制系统，并尽可能减小控制系统之间的干扰。他将跳跃过程分为支撑和腾空两个环节。在支撑环节足底触地，此时腿可以从地面借力，从而控制机器人的跳跃高度和姿态。在腾空环节，机器人近似沿着抛物线运动，此时通过控制落足点的位置，来调整机器人运动速度。这种三通道解耦控制法不仅在单足机器人上适用，只要通过简单的扩展，就很容易应用到双足和四足机器人上。

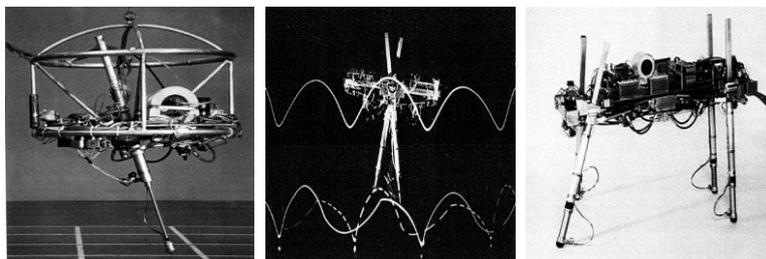


图4 Marc Raibert 的单足、两足、四足机器人

根据 Raibert 的理论，无论是单足机器人，还是两足、四足机器人，都可以抽象为一个弹簧负载倒立摆模型（SLIP: Spring Loaded Inverted Pendulum），使用简单的模型可以降低问题的复杂性，以便于分析与优化。对机器人的这种简化至今仍被广泛使用<sup>[21, 22]</sup>，并取得了不错的控制效果。Raibert 早期实验的成功，证明了足式机器人无需进行大规模优化或感知环境，就可以在不同的地形上运动，而精准的动力学模型不是绝对必要的。

**Jerry Pratt 的虚拟模型控制（VMC）**<sup>[23, 24]</sup>。上世纪 90 年代，同为 MIT 腿足式机器人实验室的 Jerry Pratt 在 Raibert 的基础上首次提出了虚拟模型控制（VMC: Virtual Model Control），这是一种影响深远的控制思想<sup>[17, 18, 25, 26]</sup>，图5为使用两组虚拟“弹簧阻尼”保持平衡的两足机器人。

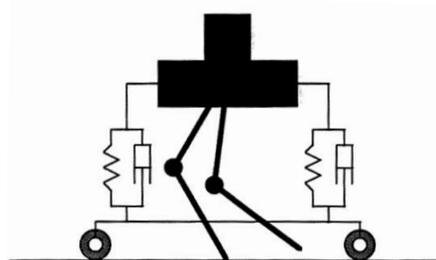


图5 使用两组虚拟“弹簧阻尼”模型保持平衡的两足机器人

虚拟模型控制是一种运动控制思想，它通过模拟虚拟组件来产生所需的关节力矩。这些关节力矩产生的效果与虚拟组件产生的效果相同，因此实现了模拟组件与真实机器人连接的控制效果。这些组件可以包括弹簧、阻尼器、缓冲器、质量以及其他任何可以想象的组件。许多用传统技术难以描述的复杂任务可以用简单的虚拟组件集轻松地描述出来。例如，设想一个机器人希望在一个未知的平面上施加冲击(比如敲门)，通常这将

是一个很难用数学语言描述的任务。然而，在虚拟模型控制中，只需要将虚拟弹簧和虚拟阻尼器以及一个具有给定动能的虚拟质量附着在机器人的手上，机器人的手将在虚拟组件集合的作用下移动，在对未知平面产生预期的冲击后，虚拟质量在虚拟弹簧阻尼器作用下又反弹回来。相比于其他控制方法，虚拟模型控制计算量小，控制效果显著，可以根据需求自由设置虚拟组件。

**全身控制 (WBC)** [27-30]。WBC 的英文全称为 Whole Body Control。为了降低运动控制算法的复杂性，控制任务通常会被分解为一组较为简单的任务，如保持平衡的躯干姿态控制，摆动相的足底轨迹跟随，或者用于防止滑动的接触力约束。每一个任务往往都需要多个驱动关节配合完成，同时每一个驱动关节又同时服务于多个任务，适应这种任务分解的理论框架是全身控制 (WBC)。一般来说，为实现一组任务，所有驱动关节都能得到控制信号的系统都可以称作全身控制，但是在本文所述全身控制至少实现了浮动基逆动力学[31-35]。任务可以通过加权平均的方式在同一个优先级上解决，也可以按照重要程度分成不同的优先级。比如要用机械臂操作一个望远镜观察远处的星星，几十米的位置误差可能都不会对观测造成影响，但是不到  $1^\circ$  的角度误差就会有巨大的视野变化，因此在这种控制任务下，姿态任务的优先级要高于位置任务的优先级。这种带有优先级的全身控制方法在足式机器人上具有广泛的应用[15, 36]。

**模型预测控制 (MPC)** [37, 38]。MPC 的英文全称为 Model Predictive Control。虽然 WBC 使用了浮动基逆动力学，考虑了完整的机器人动力系统模型，但它只找到瞬时的关节力矩和力。这意味着 WBC 无法“理解”系统未来的状态。在具有飞行相或者长时间驱动不足的步态下，单纯使用 WBC 效果不理想。考虑一种极端情况，四足机器人以具有长时间飞行相的步态 (图6) 跳跃前进。在支撑相后半程，单纯使用 WBC 或其他不具有预测特性的控制算法，控制系统不会为没有地面反力的飞行相做准备；在飞行相后半程，控制系统也不会为即将到来的地面冲击做准备。



图6 猫在跳跃时有长时间飞行相

长期以来,相关研究学者一直认为短时间范围内的预测在足式机器人运动控制中起着重要作用<sup>[39]</sup>。近年来,基于优化的模型预测控制逐渐成为足式机器人领域的研究热点,模型预测控制(MPC)试图在未来的一个滚动时域内,使用优化算法来跟踪给定的期望轨迹。已经有众多研究学者在四足机器人上实现了MPC算法,取得了不错的效果<sup>[3, 15, 16, 39, 40]</sup>。然而,在实物机器人上实现MPC算法需要在线解决大规模优化算法,对计算力要求高,往往无法实现较高的控制带宽。

综上所述,足式机器人控制系统经历了从无到有,从简化模型到全动力学模型,从瞬时控制到预测控制的发展,逐步形成了将控制问题转化为优化问题,充分发挥硬件算力,充分利用足式机器人特有的动力学特性的新局面。本文所述控制系统集各家之所长,以模型预测控制为核心,使用MPC求得足底反力;使用WBC计算关节空间的期望位置、期望速度与加速度;最后,通过多刚体动力学求得各个关节扭矩;同时也借鉴了VMC算法与SLIP模型的部分理念,最终实现了四足机器人在非结构化路面上全方位稳定行走,并能在一定的侧向冲击扰动下恢复稳定。

### 1.3 本文主要研究内容及章节安排

本文以模型预测控制(MPC)与全身控制(WBC)为核心,在四足机器人的控制理论、控制系统软硬件设计、结构设计、样机测试等方面进行了深入的研究。

机器人控制系统就是要解决机器人在哪儿、去哪儿和怎么去的问题。为了解决机器人在哪儿的问题,设计了四足机器人状态估计器。四足机器人状态估计器主要包含对躯干旋转的估计和对躯干平动的估计。对躯干旋转的估计由IMU实现,只需对IMU的原始数据稍作旋转坐标变换就可得到躯干的姿态与角速度。对躯干平动的估计方面,利用卡尔曼滤波器设计了一个足底里程计:假设支撑足在地面上不打滑,通过建立本体系下足底速度与世界系下足底速度之间的映射关系,来求解世界系下质心的位置与速度。

为了解决机器人去哪儿的问题,设计了质心轨迹规划器和足底轨迹规划器。利用手柄传来的上层指令,解算接下来一段时间质心的期望轨迹;根据实际质心水平速度和期望质心水平速度的偏差,选择合适的落脚点;在足底离地点和落脚点之间规划一条合适的足底轨迹。

为了解决机器人怎么去的问题,设计了基于模型预测控制与全身控制的控制器。在模型预测控制中,将整个机器人简化为一个单刚体模型并分析其动力学,建立状态方程、预测方程,将控制问题转化为一个优化问题,最终求出足底反力。在全身控制中,采用

了多刚体动力学。四足机器人全身控制将整体的任务按照重要程度分为四个子任务，利用雅可比矩阵的零空间特性，保证低优先级的任务无法影响高优先级任务的控制。

为了验证控制系统的可靠性，在仿真环境下搭建四足机器人虚拟样机，仿真验证了控制系统足以将不同步态下受到侧向冲击的四足机器人稳定下来。

最后，设计了四足机器人主体结构，主要包括布置驱动模块、选择传动方式、设计躯干及腿部结构、对关键零部件进行有限元仿真。将零件交付加工后，装配并搭建物理样机，在物理样机上再次验证了控制系统的稳定性。

本文各章节具体安排如下：

第一章为绪论，介绍了课题意义以及不同时期的研究学者的不同控制策略，着重介绍了几个在四足机器人控制系统发展史上具有里程碑意义的控制思想。

第二章为四足机器人控制系统概述，介绍了四足机器人控制系统整体框架、有限状态机、步态调度器、状态估计器、轨迹规划器。

第三章介绍四足机器人的模型预测控制。首先介绍了模型预测控制在间歇欠驱系统中的必要性，然后将四足机器人简化为一个单刚体模型并分析其动力学，搭建动力学方程及预测方程，最终将控制问题转化为优化问题。

第四章介绍四足机器人全身控制。首先介绍了带优先级的全身控制的基本原理，推导了双任务系统和多任务系统的递推方程，然后将全身控制方法应用到四足机器人上，最后对模型预测控制与全身控制的结果做松弛优化，得到关节扭矩指令。

第五章介绍了多刚体动力学算法，将四足机器人简化为由 13 个关节连接的 13 个刚体组成的刚体组合，并在 Plücker 坐标系下建立四足机器人多刚体动力学方程。

第六章介绍了控制系统的软硬件设计及仿真。首先介绍了控制系统硬件电气连接方案以及硬件设备，然后介绍了控制系统软件的各个模块及第三方库，最后在仿真平台上验证了控制系统的可靠性。

第七章介绍机械结构及部分零部件的有限元仿真。首先介绍了腿型布置形式，然后分析了不同传动方式的优缺点，然后对重点零部件进行有限元仿真分析。最后介绍了四足机器人物理样机实验，主要包括对角步态原地踏步实验，非结构化路面行走实验、侧向冲击实验，最后对实验结果做了简要分析。

## 1.4 本章小节

本章首先介绍了课题意义，然后概述了足式机器人控制理论发展史，最后概述了论文主要研究内容及章节安排。

## 第二章 仿生四足机器人控制系统总体介绍

四足机器人控制系统包含控制系统算法、控制系统软件与控制系统硬件。其中控制系统算法是控制系统的核心，也是本文的研究重点。

图7为四足机器人控制系统框图，其中实线箭头表示的是控制信号流，虚线箭头表示的是状态信号流。按照不同的功能，控制系统算法可以分为若干个模块，分别是：用于四足机器人行为管理的有限状态机；接收手柄指令，生成质心轨迹的质心轨迹生成器；通过融合多种传感器数据，估算机器人内部状态的状态估计器，其中包含了一个基于支撑足不打滑假设的足底里程计；用于切换支撑腿与摆动腿的步态调度器；用于在支撑相求足底反力的模型预测控制；包含了落足点规划、足底轨迹规划的摆动相规划；带优先级的多任务全身控制器；一组用于计算浮动基体逆运动学的刚体动力学算法。在控制系统中，除模型预测控制算法的运行频率为100Hz外，其他算法的运行频率都是500Hz。模型预测控制的控制周期用 $\Delta T$ 表示， $\Delta T = 0.01s$ ；其他控制周期用 $\Delta t$ 表示， $\Delta t = 0.002s$ 。

本章将从机器人学基本概念出发，逐个讲解控制系统中基本的算法。在第三章将介绍模型预测控制（MPC），第四章介绍全身控制（WBC），第五章介绍基于 Plücker 坐标系的刚体动力学算法。

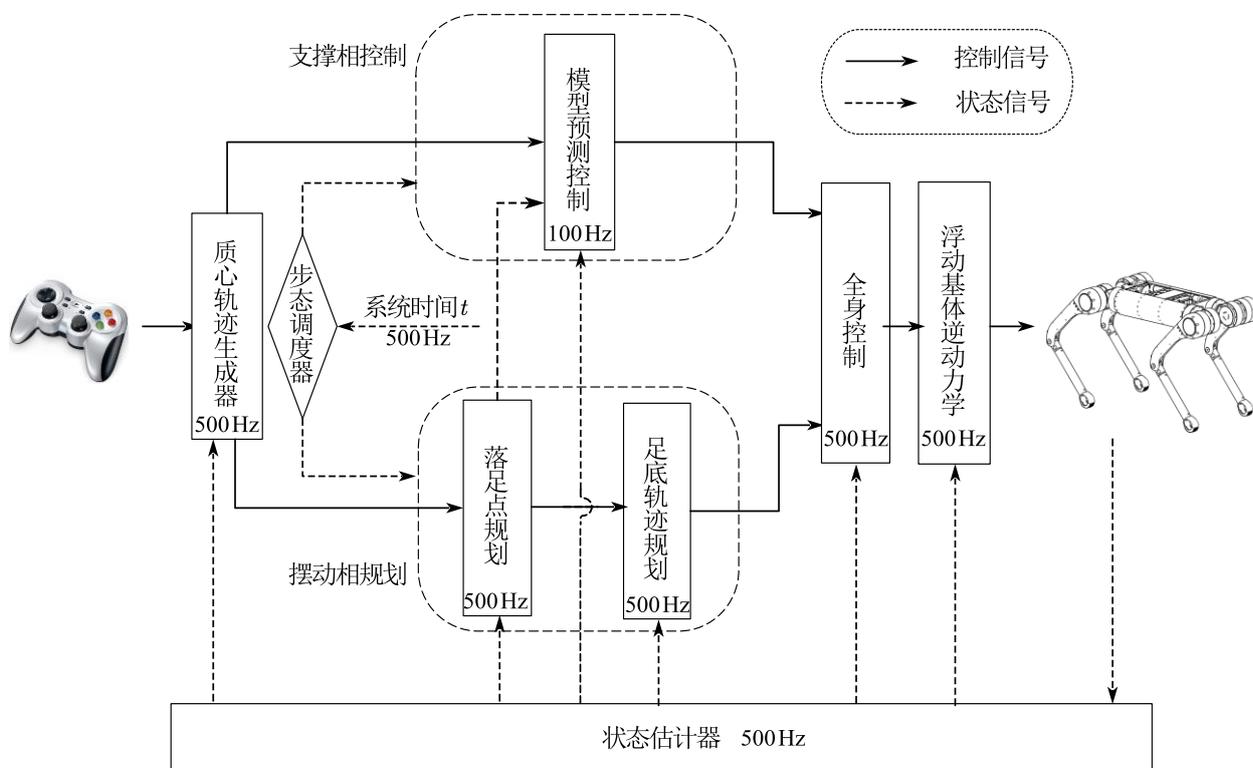


图7 四足机器人控制系统框图

## 2.1 有限状态机

在手柄的操控下，机器人将表现出不同的行为，如起立，下蹲，行走，平衡站立等等。一方面，某些状态无法直接切换到另一些状态，比如下蹲动作完成后，无法切换到行走状态；另一方面，一些状态存在“忙碌时间”，如起立动作执行到一半，尚未完全站起时，无法切换到行走状态，否则会发生危险。为了实现上述状态之间的调度，设计了一个有限状态机。

图 8 为状态机切换关系图，该图以箭头的形式表明了不同状态之间所允许的切换关系。例如，起立之后可以下蹲，下蹲之后也可以起立，但是下蹲后不能行走。图中虚线框起来的五个状态，都是控制系统有明确控制信号的状态。当机器人处于虚线框中的状态时，状态机中存在危险检测机制，检测到危险时状态机自动调度到阻尼保护状态，随后可手动切换到电机掉电状态或者关闭机器人电源。

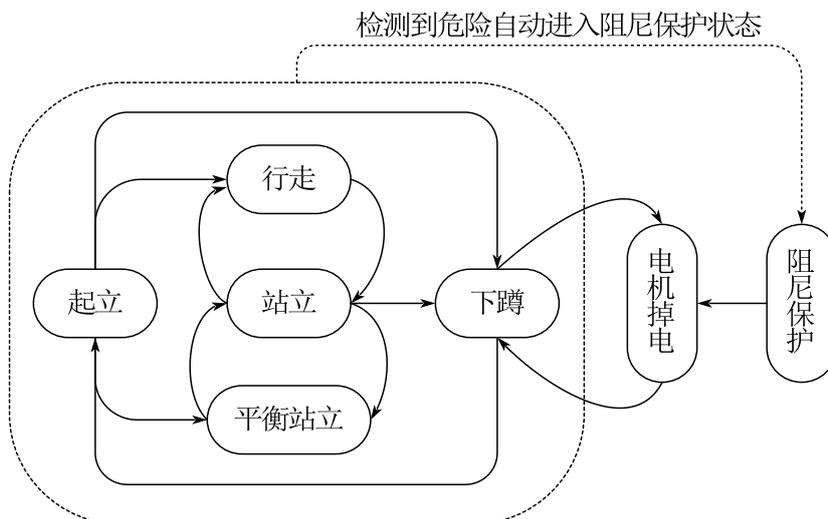


图8 状态机切换关系图

图 9 为状态机调度流程图。状态机收到状态切换指令后，首先检查状态机切换关系图，看此指令是否被允许执行，若不被允许执行，直接退出，等待下一个指令。若可以执行，判断当前状态是否处于忙碌时间，若处于忙碌时间，则一直等待到当前状态退出忙碌时间，然后进行状态切换。其中，行走状态若执行对角步态，非忙碌时间为四只脚同时在地面的瞬间；平衡站立状态的非忙碌时间为手柄指令皆为 0；下蹲、站立、起立的忙碌时间由一个计时器决定，当进入该状态时计时器开始计时，计时到一个固定时间后这些状态进入非忙碌状态。

在阻尼保护状态下，机器人的 12 个电机分别独立控制，其控制律为：

$$\tau_{motor} = -k_{damping} \cdot v_{motor} \quad (2.1)$$

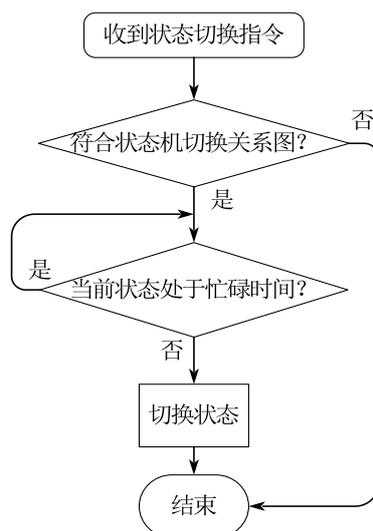


图9 状态机调度流程图

公式(2.1)中 $\tau_{motor}$ 为电机扭矩指令， $k_{damping}$ 是阻尼参数， $v_{motor}$ 是电机的转速。电机转动速度越大，电机对外的扭矩越大。通过实际调试，得到合理的参数 $k_{damping}$ ，可以实现机器人在危险情况下快速平稳地静止下来。表 1 展示了进入阻尼保护状态的各种触发条件。

表 1 机器人进入阻尼保护状态的条件

进入阻尼保护的原因	阈值
机身俯仰角与横滚角之和过大	$35^\circ$
机身水平速度过大	$5.5m/s$
足底速度过大	$5.5m/s$
关节速度过大	$18^\circ/s$
关节控制跟踪误差过大	$20^\circ$
足底控制跟踪误差过大	$0.2m$

## 2.2 坐标系及刚体的位姿描述

### 2.2.1 刚体坐标系

四足机器人每条腿都有 3 个连杆与 3 个关节，从靠近躯干开始，分别为 3 个连杆和 3 个关节编号 1, 2, 3。在本文中，依照右前腿、左前腿、右后腿、左后腿的顺序，为 4 条腿编号 1, 2, 3, 4。算上躯干，整个四足机器人包含 13 个刚体。从躯干开始，依照 4 条腿的顺序，为躯干、1 号腿 1 号连杆、1 号腿 2 号连杆、...、4 号腿 3 号连杆，分别编号 1, 2, ..., 13。

图10为四足机器人各个关节位置都为 0 时的状态。为了描述组成机器人的各个刚体在空间中的位置及姿态，首先定义一个刚接到外部物理环境的世界坐标系 $\mathcal{O}$ ，简称世界系。若机器人在水平地面上开机，在启动控制器的瞬间， $\mathcal{O}$ 的 $z$ 轴垂直向上且经过机器人

躯干的几何中心， $x$ 轴水平向前， $y$ 轴水平向左， $O$ 的原点在 $z$ 轴与地面的交点上。

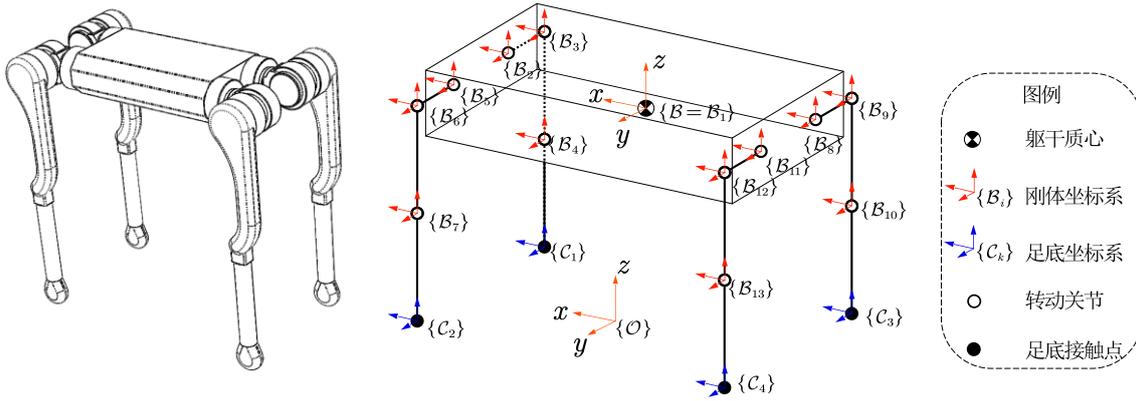


图10 各关节位置为0时，四足机器人坐标系

随后，需要为组成机器人的每一个刚体附加一个刚接其上的坐标系，称为刚体坐标系，用符号 $B_i$ 来表示，其中 $i$ 为刚体的编号。特别地，称刚接于躯干上的坐标系为本体系，用符号 $B$ 来表示，即 $B = B_1$ 。本体系的坐标原点为躯干的几何中心，在控制器启动瞬间，本体系与世界系的朝向相同。在结构设计上，组成躯干的各个零部件已经尽可能对称分布，因此可近似认为躯干质心在躯干的几何中心上，即躯干质心与本体系的原点重合。

机器人每一个关节都连接了两个刚体，称编号较小的刚体为父刚体，编号较大的刚体为子刚体。如图10，在每一个刚体上建立一个刚体坐标系，其方向与原点的选取采用以下三条简单的规则：

- 当所有关节位置为0时，所有刚体坐标系的朝向与本体系相同。
- 本体系的坐标原点，与躯干的几何中心重合。
- 所有关节位置为0时，一个刚体的刚体坐标系，应沿着尽可能少的坐标轴就可以平移至其父刚体的刚体坐标系。

足底接触点是机器人与外部环境交互作用的主要位置，亦非常重要，因此依照4条腿的顺序，在四个足底接触点上建立足底坐标系 $C_1 \sim C_4$ ，其原点与足底接触点重合，方向与足底接触点所在的3号连杆的刚体坐标系相同，且刚接到3号连杆上。

子刚体坐标系与父刚体坐标系之间的相对位置关系主要由两方面组成：机器人躯干宽度、长度、连杆长度等机器人物理尺寸；因关节转动而产生的相对位置关系。前者是常数，后者是关节位置的函数。图11展示了与坐标系之间相对位置关系有关的机器人物理尺寸，其具体参数为：

$$[h_x \ h_y \ l_1 \ l_2 \ l_3]^T = [0.275m \ 0.0625m \ 0.088m \ 0.27m \ 0.27m]^T \quad (2.2)$$

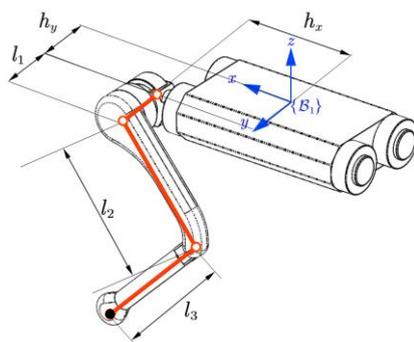


图11 与坐标系之间相对位置关系有关的机器人物理尺寸

## 2.2.2 刚体平动的位置与速度描述

### (1) 刚体平动的位置描述

运动学是对点、刚体、多刚体系统运动的描述，它仅仅用来描述物体是怎么运动的，而不关心物体为什么运动。为了描述一个点的位置与速度，引入了定义在 $\mathbb{R}^3$ 上的位置向量及其对时间的微分。如图12所示，点B相对于点A的位置可以用三维向量表示为：

$$\mathbf{r}_{AB} \quad (2.3)$$

在本文中，通常使用加粗小写字母表示向量，使用不加粗大写字母表示点，使用诸如A、B、C字符表示笛卡尔坐标系。向量的字符本身表明该向量的类型，右下角标表明该向量所表示的具体物理量，左上角标表示其所在的坐标系，右上角标与左下角标用于标识向量的其他信息。

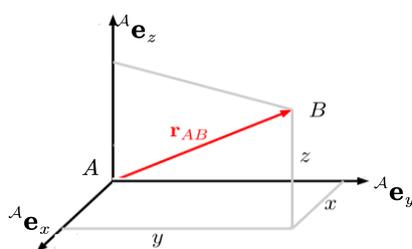


图12 点的位置描述

为了用具体的数字来描述向量的大小，在A点定义一个坐标系A，使用一组单位正交基 $(\mathbf{e}_x, \mathbf{e}_y, \mathbf{e}_z)$ 来描述坐标系A中向量的大小：

$${}^A\mathbf{r}_{AB} = x \cdot \mathbf{e}_x + y \cdot \mathbf{e}_y + z \cdot \mathbf{e}_z = [x \ y \ z]^T \quad (2.4)$$

### (2) 刚体平动的速度描述

定义了三维空间的点位置之后，点在三维空间的速度就是点位置对时间的一阶微分，图12中点B相对于点A的速度为：

$${}^A\dot{\mathbf{r}}_{AB} = [\dot{x} \ \dot{y} \ \dot{z}]^T \quad (2.5)$$

### 2.2.3 刚体转动的位置与速度描述

#### (1) 刚体转动的位置描述

如图13所示，两个坐标系 $\mathcal{A}$ 与 $\mathcal{B}$ ，其原点 $A$ 与 $B$ 在空间上重合。考虑一点 $P$ ，在坐标系 $\mathcal{A}$ 下的描述为：

$${}^A\mathbf{r}_{AP} = [{}^A r_{AP}^x \quad {}^A r_{AP}^y \quad {}^A r_{AP}^z]^T \quad (2.6)$$

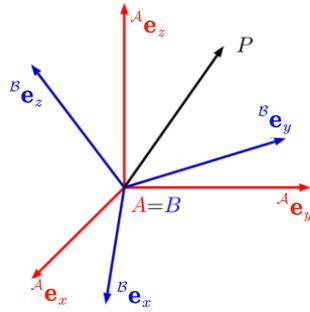


图13 两个坐标系之间的旋转

同理，点 $P$ 在坐标系 $\mathcal{B}$ 下的描述为：

$${}^B\mathbf{r}_{AP} = [{}^B r_{AP}^x \quad {}^B r_{AP}^y \quad {}^B r_{AP}^z]^T \quad (2.7)$$

在图12中，使用了一组单位正交基来参数化描述坐标系 $\mathcal{A}$ 中向量的大小。与之类似，记坐标系 $\mathcal{B}$ 中用于参数化描述向量的一组单位正交基在坐标系 $\mathcal{A}$ 中的描述为 $({}^A\mathbf{e}_x^B, {}^A\mathbf{e}_y^B, {}^A\mathbf{e}_z^B)$ ，则可以写出坐标向量 ${}^A\mathbf{r}_{AP}$ 和 ${}^B\mathbf{r}_{AP}$ 之间的映射关系：

$${}^A\mathbf{r}_{AP} = {}^A\mathbf{e}_x^B \cdot {}^B r_{AP}^x + {}^A\mathbf{e}_y^B \cdot {}^B r_{AP}^y + {}^A\mathbf{e}_z^B \cdot {}^B r_{AP}^z \quad (2.8)$$

将式(2.8)以紧凑的方式重写为：

$${}^A\mathbf{r}_{AP} = [{}^A\mathbf{e}_x^B \quad {}^A\mathbf{e}_y^B \quad {}^A\mathbf{e}_z^B] {}^B\mathbf{r}_{AP} = {}^A R_B^B \mathbf{r}_{AP} \quad (2.9)$$

其中， ${}^A R_B$ 是被称为旋转矩阵的3行3列矩阵，它可以将同一个向量（无论是坐标系 $\mathcal{B}$ 下还是坐标系 $\mathcal{A}$ 下，向量 $\overrightarrow{AP}$ 始终是同一个向量）从一个坐标系映射到另一个坐标系。由于其列向量为一组单位正交向量，则 ${}^A R_B$ 是正交矩阵，根据正交矩阵的性质：

$${}^A R_B^T {}^A R_B = \mathbb{I}_3 \quad (2.10)$$

其中 $\mathbb{I}_3$ 为3行3列单位矩阵，结合式(2.9)和式(2.10)不难得到以下关于旋转矩阵的性质：

$${}^B R_A = {}^A R_B^{-1} = {}^A R_B^T \quad (2.11)$$

最简单同时也是最常见的旋转为绕三个坐标轴的基本旋转，图14展示了分别绕 $x$ 轴、 $y$ 轴、 $z$ 轴旋转的基本旋转，其旋转矩阵分别为：

$${}^A R_B = R_x(\varphi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c_\varphi & -s_\varphi \\ 0 & s_\varphi & c_\varphi \end{bmatrix}, {}^A R_B = R_y(\varphi) = \begin{bmatrix} c_\varphi & 0 & s_\varphi \\ 0 & 1 & 0 \\ -s_\varphi & 0 & c_\varphi \end{bmatrix}, {}^A R_B = R_z(\varphi) = \begin{bmatrix} c_\varphi & -s_\varphi & 0 \\ s_\varphi & c_\varphi & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.12)$$

为提高公式紧凑性， $c_\varphi$ 与 $s_\varphi$ 分别为 $\cos(\varphi)$ 和 $\sin(\varphi)$ 的简写。

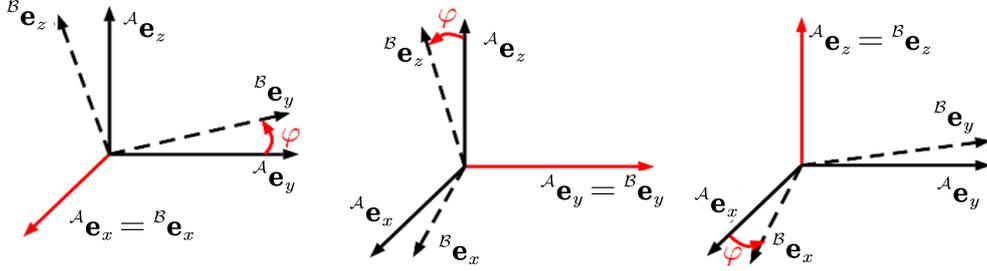


图14 分别绕  $x$  轴、 $y$  轴、 $z$  轴旋转的基本旋转

考虑三个坐标系  $A$ 、 $B$ 、 $C$ ，向量  $\mathbf{r}$  从坐标系  $B$  映射到坐标系  $A$  为：

$${}^A \mathbf{r} = {}^A R_B \cdot {}^B \mathbf{r} \quad (2.13)$$

向量  $\mathbf{r}$  从坐标系  $C$  映射到坐标系  $B$  为：

$${}^B \mathbf{r} = {}^B R_C \cdot {}^C \mathbf{r} \quad (2.14)$$

结合式(2.13)与式(2.14)，向量  $\mathbf{r}$  从坐标系  $C$  映射到坐标系  $B$ ，然后再从坐标系  $B$  映射到坐标系  $A$  为：

$${}^A \mathbf{r} = {}^A R_B \cdot {}^B R_C \cdot {}^C \mathbf{r} = {}^A R_C \cdot {}^C \mathbf{r} \quad (2.15)$$

由式(2.15)可以看出，多个连续的转动过程组合起来依然是个转动，这一连续的多个转动过程可由旋转矩阵连续左乘来描述：

$${}^A R_C = {}^A R_B \cdot {}^B R_C \quad (2.16)$$

如前文所述，三维空间中的旋转通常用  $3 \times 3$  旋转矩阵，即通过 9 个参数来描述。然而，由于公式(2.10)中存在正交约束条件，这 9 个参数并不是独立的。在工程上，参数越多，计算量与存储空间越大，而且不够直观也不利于调试。根据公式(2.16)，空间旋转可以理解为公式(2.12)中定义的三个基本旋转的组合。为了能完全描述所有可能的方向，至少需要绕坐标轴进行三次旋转，且连续两次旋转的坐标轴不能相同，这种用绕坐标轴旋转三次来描述空间旋转的方法称为欧拉角。依据坐标轴旋转次序的不同，欧拉角可分为 12 种 (XYZ, XZY, YXZ, YZX, ZXY, ZYX, XYX, XZX, YXY, YZY, ZXZ, ZYZ)。如图15所示，本文所用欧拉角为机器人领域较为常用的 ZYX 欧拉角，也被称为 Tait-Bryan 角，其坐标轴旋转次序为 ZYX。

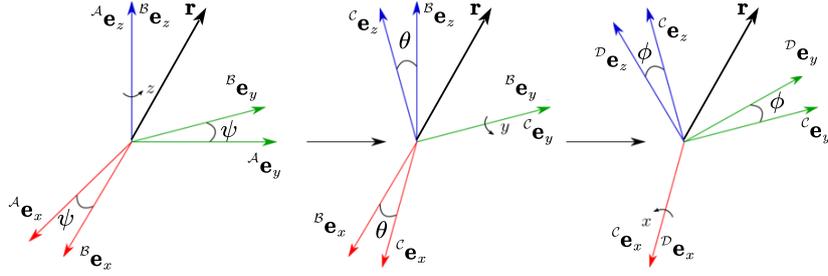


图15 ZYX 欧拉角的转动顺序

以向量的形式描述 ZYX 欧拉角：

$$\Theta = [\phi \ \theta \ \psi]^T \quad (2.17)$$

其中， $\phi$  为横滚角， $\theta$  为俯仰角， $\psi$  为偏航角。如图15所示，假设空间中有一向量  $\mathbf{r}$ ，在坐标系  $D$  下的描述为  ${}^D\mathbf{r}$ 。由于坐标系  $D$  是坐标系  $C$  绕  $x$  轴旋转  $\phi$  得到的，结合图14与公式(2.12)，可知  $\mathbf{r}$  在坐标系  $C$  下的描述  ${}^C\mathbf{r}$  为：

$${}^C\mathbf{r} = R_x(\phi) {}^D\mathbf{r} \quad (2.18)$$

类似的，依次回溯向量  $\mathbf{r}$  在坐标系  $C$ 、 $B$ 、 $A$  内的描述，最终得到  $\mathbf{r}$  在坐标系  $A$  下的描述为：

$${}^A\mathbf{r} = R_z(\psi) R_y(\theta) R_x(\phi) {}^D\mathbf{r} \quad (2.19)$$

因此，ZYX 欧拉角到旋转矩阵的变换为：

$$\begin{aligned} R(\Theta) &= R_z(\psi) R_y(\theta) R_x(\phi) \\ &= \begin{bmatrix} c_\psi & -s_\psi & 0 \\ s_\psi & c_\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_\theta & 0 & s_\theta \\ 0 & 1 & 0 \\ -s_\theta & 0 & c_\theta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & c_\phi & -s_\phi \\ 0 & s_\phi & c_\phi \end{bmatrix} \\ &= \begin{bmatrix} c_\theta c_\psi & c_\psi s_\phi s_\theta - c_\phi s_\psi & s_\phi s_\psi + c_\phi c_\psi s_\theta \\ c_\theta s_\psi & c_\phi c_\psi + s_\phi s_\theta s_\psi & c_\phi s_\theta s_\psi - c_\psi s_\phi \\ -s_\theta & c_\theta s_\phi & c_\phi c_\theta \end{bmatrix} \end{aligned} \quad (2.20)$$

通过分析式(2.20)，可以得到其逆变换，给定一个旋转矩阵为：

$$R = \begin{bmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{32} & c_{33} \end{bmatrix} \quad (2.21)$$

可以得到式(2.20)的逆变换为：

$$\Theta = \begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix} = \begin{bmatrix} \text{atan}(c_{32}/c_{33}) \\ -\text{atan}(c_{31}/\sqrt{c_{32}^2 + c_{33}^2}) \\ \text{atan}(c_{21}/c_{11}) \end{bmatrix} \quad (2.22)$$

显然式(2.22)并不是唯一的逆变换。在实际使用时，需要对分母极小甚至为 0 的情况做特殊处理。

## (2) 刚体转动的速度描述

如前文中公式(2.5)所述，在三维空间中定义一组正交基后，点的位置就可以参数化描述，点的平动速度就是点位置对时间的一阶微分。但是在涉及到转动时，问题就颇为复杂。根据牛顿第一定律，力是改变物体运动状态的原因，在不受外力作用时，物体的速度将保持不变。为了使刚体的转动也满足这一规律，不能将角速度简单定义为欧拉角对时间的一阶微分。因为欧拉角三个参数 $\phi$ 、 $\theta$ 、 $\psi$ 的旋转轴分别是图15中 ${}^c\mathbf{e}_x$ 、 ${}^B\mathbf{e}_y$ 、 ${}^A\mathbf{e}_z$ ，这三个单位向量中， ${}^B\mathbf{e}_y$ 是 $\psi$ 的函数， ${}^c\mathbf{e}_x$ 是 $\theta$ 、 $\psi$ 的函数，即： ${}^B\mathbf{e}_y = {}^B\mathbf{e}_y(\psi)$ ， ${}^c\mathbf{e}_x = {}^c\mathbf{e}_x(\psi, \theta)$ 。因此这三个单位向量既不满足正交性，又随着欧拉角的变化而变化，若以这三个单位向量为基来描述角速度，角速度的改变不仅仅和外力有关，也和刚体的姿态改变有关，这违背了牛顿第一定律。

图16为平面刚体绕定轴转动的示意图，图中箭头表示箭头起点处刚体的线速度。可见，在转动的作用下，刚体上的点都存在一个线速度，从而在平面上形成一个不随时间变化的稳定速度场。在该稳定速度场中，一个点的线速度矢量，仅取决于其位置矢量。从这个例子可以看出，角速度表征了在刚体转动过程中，刚体上点的位置矢量到线速度矢量的映射关系。换句话说，对角速度的描述，等效于对稳定速度场的描述。

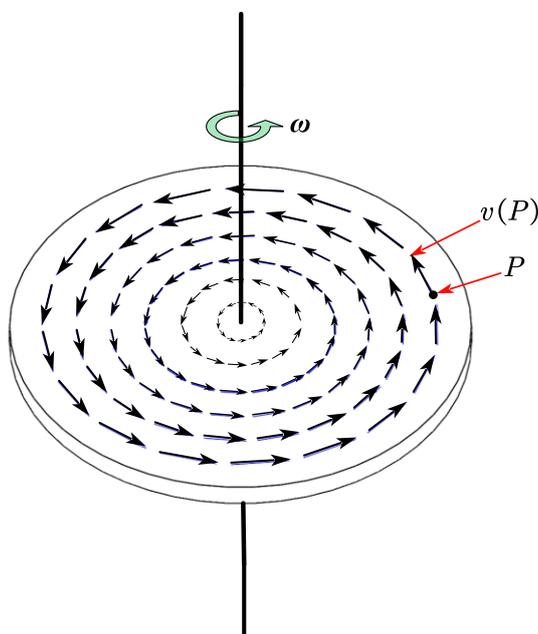


图16 平面刚体绕定轴转动形成速度场

现在，分析三维空间下转动刚体上点的位置矢量到线速度矢量的映射关系。如图13所示，坐标系 $B$ 相对于坐标系 $A$ 旋转，若 $P$ 为刚体上一点，坐标系 $B$ 为该刚体的刚体坐标系，对公式(2.9)等号两边微分：

$$\begin{aligned}
{}^A\dot{\mathbf{r}}_{AP} &= \frac{d({}^A\mathbf{r}_{AP})}{dt} \\
&= {}^A\dot{R}_B {}^B\mathbf{r}_{AP} + {}^A R_B \frac{d({}^B\mathbf{r}_{AP})}{dt} \\
&= {}^A\dot{R}_B {}^B\mathbf{r}_{AP} + {}^A R_B \cdot \mathbf{0} \\
&= {}^A\dot{R}_B {}^A R_B^T {}^A\mathbf{r}_{AP} \\
&= {}^A\boldsymbol{\Omega}_B {}^A\mathbf{r}_{AP}
\end{aligned} \tag{2.23}$$

在此处  $d({}^A\mathbf{r}_{AP})/dt$  表示  $\mathbf{r}_{AP}$  在坐标系  $\mathcal{A}$  下描述的分,  ${}^A\dot{\mathbf{r}}_{AP}$  表示  $\mathbf{r}_{AP}$  的分在坐标系  $\mathcal{A}$  下的描述, 在图 13 中坐标系  $\mathcal{A}$  为空间上的不动系, 因此二者相等。公式(2.23)中矩阵  ${}^A\boldsymbol{\Omega}_B = {}^A\dot{R}_B {}^A R_B^T$  为点  $P$  的位置矢量到其速度矢量的映射矩阵, 称该映射矩阵为角速度张量。分析该矩阵发现:

$$\begin{aligned}
{}^A\boldsymbol{\Omega}_B + {}^A\boldsymbol{\Omega}_B^T &= {}^A\dot{R}_B {}^A R_B^T + ({}^A\dot{R}_B {}^A R_B^T)^T \\
&= {}^A\dot{R}_B {}^A R_B^T + {}^A R_B {}^A\dot{R}_B^T \\
&= \frac{d}{dt} ({}^A R_B {}^A R_B^T) \\
&= \frac{d}{dt} (\mathbb{I}_3) \\
&= \mathbf{0}
\end{aligned} \tag{2.24}$$

因此  ${}^A\boldsymbol{\Omega}_B = -{}^A\boldsymbol{\Omega}_B^T$ , 在这个约束条件下, 角速度张量可以写成:

$${}^A\boldsymbol{\Omega}_B = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix} \tag{2.25}$$

据此, 定义角速度矢量为:

$${}^A\boldsymbol{\omega}_{AB} = [\omega_x \ \omega_y \ \omega_z]^T \tag{2.26}$$

根据叉乘的定义, 不难验证  ${}^A\boldsymbol{\omega}_{AB} \times {}^A\mathbf{r}_{AP} = {}^A\boldsymbol{\Omega}_B {}^A\mathbf{r}_{AP}$ , 为了方便描述角速度矢量与角速度张量之间的变换关系, 定义反对称矩阵运算符<sup>[41]</sup>:

$$\mathbf{a}_x = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix}_x = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix} \tag{2.27}$$

则有:

$${}^A\boldsymbol{\omega}_{AB \times} = {}^A\boldsymbol{\Omega}_B \tag{2.28}$$

其中  ${}^A\boldsymbol{\omega}_{AB}$  的物理意义为坐标系  $\mathcal{B}$  相对于坐标系  $\mathcal{A}$  的角速度矢量在坐标系  $\mathcal{A}$  中的描述。在公式(2.23)等号两边左乘  ${}^B R_A$ , 可在坐标系  $\mathcal{B}$  下分析速度矢量与位置矢量的关系:

$${}^B\dot{\mathbf{r}}_{AP} = {}^B R_A {}^A\dot{\mathbf{r}}_{AP} = \underbrace{({}^B R_A \cdot {}^A\boldsymbol{\Omega}_B \cdot {}^A R_B)}_{{}^B\boldsymbol{\omega}_{AB \times}} \cdot {}^B\mathbf{r}_{AP} \tag{2.29}$$

注意 ${}^B\dot{\mathbf{r}}_{AP}$ 为 $\mathbf{r}_{AP}$ 的微分在坐标系 $B$ 下的描述，而不是 $\mathbf{r}_{AP}$ 在坐标系 $B$ 下的描述的微分，即：

$${}^B\dot{\mathbf{r}}_{AP} = {}^B\left(\frac{d\mathbf{r}_{AP}}{dt}\right) \neq \frac{d({}^B\mathbf{r}_{AP})}{dt} = \mathbf{0} \quad (2.30)$$

根据公式(2.29)，角速度有以下性质：

$${}^B\boldsymbol{\omega}_{AB\times} = {}^B R_A \cdot {}^A\boldsymbol{\omega}_{AB\times} \cdot {}^A R_B \quad (2.31)$$

及其推论：

$${}^B\boldsymbol{\omega}_{AB\times} = {}^B R_A \cdot ({}^A\dot{R}_B^T \cdot {}^A R_B) \cdot {}^A R_B = {}^B R_A \cdot {}^A\dot{R}_B \quad (2.32)$$

## 2.3 步态调度器

在本文中，使用步态这一概念来描述机器人每条腿支撑相与摆动相的周期性调度行为。支撑相与摆动相的切换方式，通常有基于时间的切换方式和基于事件的切换方式。

基于时间的切换方式是指没有内部或外部传感器做辅助，仅仅依靠系统时钟强行切换。当使用这种切换方式时，控制程序内部的切换与物理世界的实际切换往往会有时差，尤其是在崎岖路面上。这就导致当机器人踩到坑里时，尽管足底还未真实触地，但控制程序已经切换到支撑相，从而足底会有一个较大的垂直向下加速度；当机器人踩到石头上时，尽管足底已经真实触地，但控制程序仍然在摆动相，导致足底被迫紧急减速。因此，无论是踩到坑里还是踩到石头上，机器人都会因切换有时差而受到地面传来的不正常冲击。

基于事件的切换方式是指在机器人足底加触碰传感器或者压力传感器，当真实物理触碰发生时，传感器能明确将触碰信号传回控制系统。理论上来说基于事件的切换方式要优于基于时间的切换方式，其稳定性上限更高，也必然是未来足式机器人的发展方向。但是基于事件的切换方式也存在诸多不可忽视的问题：

**足底传感器将导致腿部惯量加大。**由于传感器处在腿部的末端，且转动惯量与距离的平方成正比。因此在足底稍稍增加质量，都会急剧增加整条腿的转动惯量。同时，此处也是整个机器人上绝对速度最快的部位，增加质量会加大控制难度。

**足底传感器需要保证一定的使用寿命。**足底是机器人与外部世界接触的主要部位，不仅仅要反复承受落地瞬间的冲击，还要考虑到机器人在不同路况上行走，要具有一定的耐磨性，具有防水、防酸碱性能。

**足底传感器要有足够的可靠性。**一旦足底发生了真实物理触碰，而足底传感器错过了这一触碰或发生其他问题导致触碰信号未能传回控制器，控制器将控制足底一直向下

探，直到腿伸至最长，这往往会引发危险。

**信号与供电回路应精心设计。**足底传感器在小腿末端，而控制器往往在机器人躯干内部，足底传感器的供电线与信号线要从小腿到膝关节再到大腿，一路回传到躯干内部。这一系列连杆相互之间都有运动，因此导线一方面要能承受反复弯折，另一方面要保证不被金属零件夹伤。

**足底传感器应该有较大的测量角度。**部分研究学者<sup>[42]</sup>使用基于应变片的一维压力传感器作为触地检测手段，但这种方式不适合高速运动。如图17所示，红色箭头为实际地面反力方向，蓝色箭头为传感器最佳受力方向。当机器人迈步距离较大时，触地瞬间小腿与地面角度极小，此时地面反力在压力传感器方向的分量将很小，难以测量。



图17 高速奔跑下触地瞬间小腿与地面几乎平行

综上所述，本文中控制系统采用了较为保守的基于时间的步态调度器。图18为walk步态与trot步态时序图，*FR*、*FL*、*BR*、*BL*分别是右前腿、左前腿、右后腿、左后腿，黄色方块表示对应的腿处于支撑相，可见一个完整的步态由占空比和偏移量完整描述。

**占空比：**占空比是指支撑相时间占支撑相与摆动相总时间的比例。

**偏移量：**偏移量是指在一个步态周期中，每条腿支撑相开始时刻占总时间的比例。

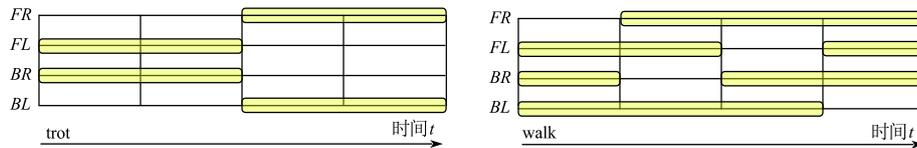


图18 trot步态（左）与walk步态（右）

本文采用“占空比-偏移量”描述法，用字符 $\mathbf{G}$ 表示步态，如对角步态可以描述为：

$$\mathbf{G}(\text{trot}) = \begin{bmatrix} \mathbf{G}_d \\ \mathbf{G}_o \end{bmatrix} = \begin{bmatrix} [0.5 & 0.5 & 0.5 & 0.5]^T \\ [0.5 & 0 & 0 & 0.5]^T \end{bmatrix} \quad (2.33)$$

$\mathbf{G}_d$ 表示占空比向量，对角步态的占空比均为0.5； $\mathbf{G}_o$ 表示偏移量向量，在一个步态周期中左前、右后腿首先进入支撑相，0.5个步态周期后，支撑相切换为右前、左后腿。

步态的频率可以通过改变周期长度来改变，因为步态是由“占空比-偏移量”形式来

定义的，所以即使步态频率发生改变，步态类型也会保持不变。表 2 给出了四足机器人常用步态的“占空比-偏移量”表示。

表 2 四足机器人常用步态的“占空比-偏移量”描述

步态名称	英文名称	占空比 $\mathbf{G}_d$	偏移量 $\mathbf{G}_o$
爬行步态	<i>walk</i>	$[0.75 \ 0.75 \ 0.75 \ 0.75]^T$	$[0.25 \ 0.75 \ 0.5 \ 0]^T$
对角步态	<i>trot</i>	$[0.5 \ 0.5 \ 0.5 \ 0.5]^T$	$[0.5 \ 0 \ 0 \ 0.5]^T$
跳跃步态	<i>pronk</i>	$[0.5 \ 0.5 \ 0.5 \ 0.5]^T$	$[0 \ 0 \ 0 \ 0]^T$
踱步步态	<i>pace</i>	$[0.5 \ 0.5 \ 0.5 \ 0.5]^T$	$[0 \ 0.5 \ 0 \ 0.5]^T$
奔跑步态	<i>bound</i>	$[0.5 \ 0.5 \ 0.5 \ 0.5]^T$	$[0 \ 0 \ 0.5 \ 0.5]^T$

设 $t$ 为从机器人开始踏步计时的系统时间， $t_{sw}$ 为名义摆动相时间， $t_{st}$ 为名义支撑相时间， $T_{gait} = t_{sw} + t_{st}$ 为一个完整的迈步周期， $n$ 为满足不等式 $t > (n-1)T_{gait}$ 的最大正整数，即当前的迈步周期为第 $n$ 个迈步周期， $t_{ng} = t - (n-1)T_{gait}$ 为当前步态周期的时间进度， $\tilde{t}_{ng} = t_{ng}/T_{gait}$ 为归一化后的时间进度，称为相位进度，显然有 $\tilde{t}_{ng} \in [0, 1)$ 。利用相位进度，为每条腿都构造一变量：

$$\tilde{i}_{\phi} = \begin{cases} \tilde{t}_{ng} - \mathbf{G}_o(i) & , \tilde{t}_{ng} \geq \mathbf{G}_o(i) \\ \tilde{t}_{ng} - \mathbf{G}_o(i) + 1, & \tilde{t}_{ng} < \mathbf{G}_o(i) \end{cases} \quad (2.34)$$

用布尔变量 $i s_{\phi}$ 来表示第 $i$ 号腿是否在支撑相：

$$i s_{\phi} = \begin{cases} 1, & \tilde{i}_{\phi} \leq \mathbf{G}_d(i) \\ 0, & \tilde{i}_{\phi} > \mathbf{G}_d(i) \end{cases} \quad (2.35)$$

定义支撑相相位进度：

$$\tilde{i}_{st} = \begin{cases} \tilde{i}_{\phi}/\mathbf{G}_d(i), & i s_{\phi} = 1 \\ 0, & i s_{\phi} = 0 \end{cases} \quad (2.36)$$

定义摆动相相位进度：

$$\tilde{i}_{sw} = \begin{cases} \frac{\tilde{i}_{\phi} - \mathbf{G}_d(i)}{1 - \mathbf{G}_d(i)}, & i s_{\phi} = 0 \\ 0, & i s_{\phi} = 1 \end{cases} \quad (2.37)$$

在本文中，用带有上波浪的 $\tilde{t}$ 表示归一化后，某个过程的进度，其取值范围为 $[0, 1)$ 。

## 2.4 状态估计器

机器人控制系统，就是要解决机器人在哪儿、去哪儿和怎么去的问题，分别对应了机器人控制系统中的状态估计、轨迹规划以及控制器。对被控系统输入输出进行测量得到的数据，只能反应系统的外部特性，而系统的动态规律则需要用内部状态变量来描述。

这些状态变量往往是无法直接测量的，因此需要用可直接获取的传感器数据对其估计，这个过程就叫状态估计。

本文所述四足机器人使用的传感器，包括测量关节角度 $q^j$ 与转速 $\dot{q}^j$ 的电机编码器，其中 $i$ 为腿的序号， $j$ 为关节序号；以及测量质心加速度 ${}^B a_{com}$ 、躯干旋转矩阵 ${}^O R_B$ 、躯干角速度 ${}^B \omega_{OB}$ 的IMU。机器人躯干的转动状态（旋转矩阵与角速度）可以直接从IMU读取，还需要融合多种传感器数据来获取质心在世界系下的位置 ${}^O p_{com}$ 与速度 ${}^O v_{com}$ 。

由于IMU可能安装到机器人躯干的任何一个地方，因此IMU输出的原始数据并不能直接表示躯干的旋转状态。为IMU定义两个坐标系：一个为IMU本体系 $\mathcal{B}_{imu}$ ，该坐标系与IMU刚接，一般厂家会在IMU设备上标出；另一个为IMU零系 $\mathcal{B}_{imu0}$ ，该坐标系与世界系刚接，当 $\mathcal{B}_{imu}$ 与 $\mathcal{B}_{imu0}$ 方向相同时，IMU输出的旋转矩阵为单位矩阵。图19展示了本文中控制器启动瞬间世界系 $\mathcal{O}$ 、本体系 $\mathcal{B}$ 、IMU本体系 $\mathcal{B}_{imu}$ 、IMU零系 $\mathcal{B}_{imu0}$ 的关系。

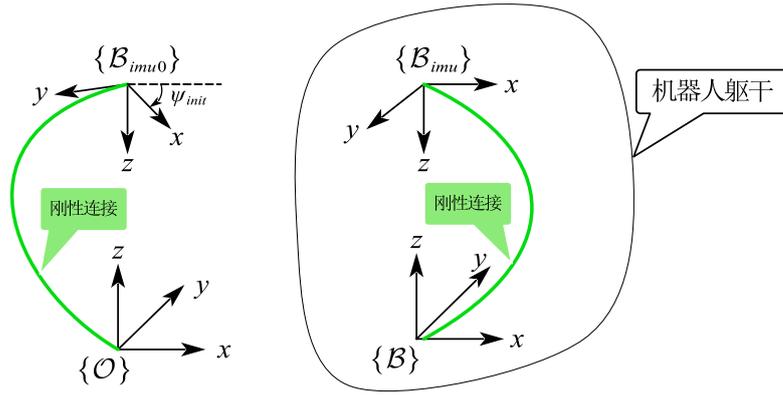


图19 世界系 $\mathcal{O}$ 、本体系 $\mathcal{B}$ 、IMU本体系 $\mathcal{B}_{imu}$ 、IMU零系 $\mathcal{B}_{imu0}$ 的关系

本体系 $\mathcal{B}$ 与IMU本体系 $\mathcal{B}_{imu}$ 的旋转变换为常值：

$${}^{imu}R_B = R_x(\pi) \quad (2.38)$$

世界系 $\mathcal{O}$ 与IMU零系 $\mathcal{B}_{imu0}$ 的旋转变换关系也为常值：

$${}^O R_{imu0} = R_z(\psi_{imu}) R_x(\pi) \quad (2.39)$$

其中 $\psi_{imu}$ 是由于机器人启动时朝向不确定而在IMU本体系 $\mathcal{B}_{imu}$ 和IMU零系 $\mathcal{B}_{imu0}$ 之间产生的角度。设其中 $R_z^{-1}(\cdot)$ 是 $R_z(\cdot)$ 的反函数，在状态估计器运行的第一个周期，通过读取陀螺仪原始旋转矩阵数据 ${}^{imu0}R_{imu}^{t=0}$ 可以计算该角度的值：

$$\psi_{imu} = -R_z^{-1}({}^{imu0}R_{imu}^{t=0}) \quad (2.40)$$

根据组合旋转的计算规则(2.16)，陀螺仪原始旋转矩阵数据 ${}^{imu0}R_{imu}$ 与躯干旋转矩阵

${}^{\circ}R_B$ 之间的关系为:

$${}^{\circ}R_B = {}^{\circ}R_{imu0} {}^{imu0}R_{imu} {}^{imu}R_B \quad (2.41)$$

加速度与角速度仅仅考虑从IMU本体系统 $\mathcal{B}_{imu}$ 到本体系统 $\mathcal{B}$ 的变换即可:

$$\begin{cases} {}^{\mathcal{B}}\boldsymbol{\omega}_{OB} = {}^{\mathcal{B}}R_{imu} \boldsymbol{\omega}_{original} \\ {}^{\mathcal{B}}\boldsymbol{a}_{com} = {}^{\mathcal{B}}R_{imu} \boldsymbol{a}_{original} \end{cases} \quad (2.42)$$

其中 $\boldsymbol{a}_{original}$ 和 $\boldsymbol{\omega}_{original}$ 分别是IMU输出的原始加速度数据和原始角速度数据。

为获取质心的平动状态,设计了一个足底里程计。足底里程计是指假设支撑足在地面上不打滑(即世界系下足底速度为0),通过建立本体系统下足底速度与世界系下足底速度之间的映射关系,来求解世界系下质心速度与位置的方法。

从传感器获取的信号一般含有一些噪音信号,为了过滤掉噪音信号,提取信号中的有用成分,需要使用滤波器对原始信号进行处理。四足机器人上使用了IMU和编码器两种传感器,为了融合这两种传感器的信息,充分利用系统状态之间的线性关系,本文采用了卡尔曼滤波器作为足底里程计的滤波器。

### 2.4.1 状态方程

注意到公式(2.42)中 ${}^{\mathcal{B}}\boldsymbol{a}_{com}$ 与 ${}^{\mathcal{B}}\boldsymbol{\omega}_{OB}$ 都是在本体系统下的描述,使用旋转矩阵求得其在世界系下的描述:

$$\begin{cases} {}^{\circ}\boldsymbol{a}_{com} = {}^{\circ}R_B \cdot {}^{\mathcal{B}}\boldsymbol{a}_{com} \\ {}^{\circ}\boldsymbol{\omega}_{OB} = {}^{\circ}R_B \cdot {}^{\mathcal{B}}\boldsymbol{\omega}_{OB} \end{cases} \quad (2.43)$$

构建卡尔曼滤波器连续状态方程:

$$\begin{cases} {}^{\circ}\dot{\boldsymbol{p}}_{com} = {}^{\circ}\boldsymbol{v}_{com} \\ {}^{\circ}\dot{\boldsymbol{v}}_{com} = {}^{\circ}\boldsymbol{a}_{com} + {}^{\circ}\boldsymbol{g} \\ {}^{\circ}_i\dot{\boldsymbol{p}} = 0 \quad \forall i \in \{1, 2, 3, 4\} \end{cases} \quad (2.44)$$

其中 ${}^{\circ}\boldsymbol{g} = [0 \ 0 \ -9.8m/s^2]^T$ 为重力加速度,  ${}^{\circ}_i\dot{\boldsymbol{p}}$ 为第*i*条腿的足底坐标在世界系下的速度。为提高公式紧凑性,定义组合向量 ${}^{\circ}\boldsymbol{p}_{foot} = [{}^{\circ}_1\boldsymbol{p}^T \ {}^{\circ}_2\boldsymbol{p}^T \ {}^{\circ}_3\boldsymbol{p}^T \ {}^{\circ}_4\boldsymbol{p}^T]^T$ 。将公式(2.44)离散化并合并到一个方程中:

$$\underbrace{\begin{bmatrix} {}^{\circ}\boldsymbol{p}_{com} \\ {}^{\circ}\boldsymbol{v}_{com} \\ {}^{\circ}\boldsymbol{p}_{foot} \end{bmatrix}}_{\boldsymbol{x}_{k+1}} = \underbrace{\begin{bmatrix} \mathbb{I}_3 & \Delta t \cdot \mathbb{I}_3 & \mathbf{0}_{3 \times 12} \\ \mathbf{0}_{3 \times 3} & \mathbb{I}_3 & \mathbf{0}_{3 \times 12} \\ \mathbf{0}_{12 \times 3} & \mathbf{0}_{12 \times 3} & \mathbb{I}_{12} \end{bmatrix}}_A \underbrace{\begin{bmatrix} {}^{\circ}\boldsymbol{p}_{com} \\ {}^{\circ}\boldsymbol{v}_{com} \\ {}^{\circ}\boldsymbol{p}_{foot} \end{bmatrix}}_{\boldsymbol{x}_k} + \underbrace{\begin{bmatrix} \mathbf{0}_{3 \times 3} \\ \Delta t \cdot \mathbb{I}_3 \\ \mathbf{0}_{12 \times 3} \end{bmatrix}}_B \underbrace{{}^{\circ}\boldsymbol{g}}_{\boldsymbol{u}_k} \quad (2.45)$$

其中,  $\Delta t$ 为控制周期,  $\mathbf{0}_{i \times j}$ 为*i*行*j*列零矩阵,  $\mathbb{I}_{12}$ 为12行12列单位矩阵。在本文中,使用字符*k*表示第*k*个控制周期。因此,忽略高斯白噪声项,公式(2.45)即卡尔曼滤波器

的离散状态方程:

$$\mathbf{x}_{k+1} = A\mathbf{x}_k + B\mathbf{u}_k \quad (2.46)$$

#### 2.4.2 观测方程

如图20所示, 通过电机编码器获取关节角度 $q^j$ 后, 结合公式(2.2), 利用空间几何关系可以求得本体系下足底位置为:

$${}^B_i\mathbf{p} = \begin{bmatrix} {}^B_i p^x \\ {}^B_i p^y \\ {}^B_i p^z \end{bmatrix} = \begin{bmatrix} -l_2 s_2 - l_3 s_{23} + \delta h_x \\ \zeta l_1 c_1 + l_3 s_1 c_{23} + l_2 c_2 s_1 + \zeta h_y \\ \zeta l_1 s_1 - l_3 c_1 c_{23} - l_2 c_1 c_2 \end{bmatrix} \quad (2.47)$$

其中,  $c_j = \cos(q^j)$ ,  $s_j = \sin(q^j)$ ,  $c_{jk} = \cos(q^j + q^k)$ ,  $s_{jk} = \sin(q^j + q^k)$ ,  $\zeta$ 与 $\delta$ 为符号变量, 定义如下:

$$\zeta = \begin{cases} 1, & i \text{ 为左腿} \\ -1, & i \text{ 为右腿} \end{cases}, \quad \delta = \begin{cases} 1, & i \text{ 为前腿} \\ -1, & i \text{ 为后腿} \end{cases} \quad (2.48)$$

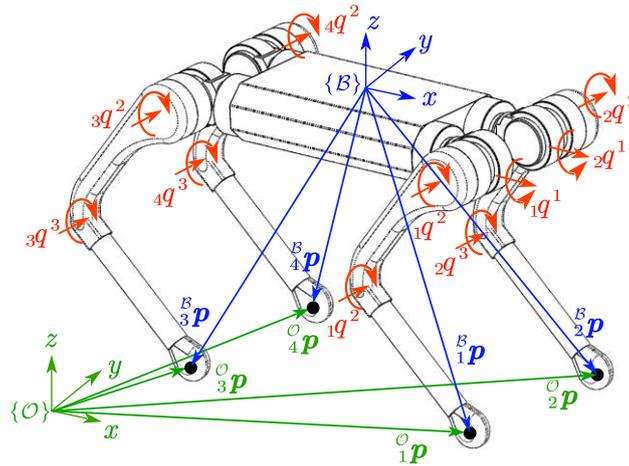


图20 关节与足底的运动学关系

对公式(2.47)求偏微分, 可以得到本体系下的足底速度与关节速度之间的关系:

$${}^B_i\dot{\mathbf{p}} = \begin{bmatrix} 0 & -l_2 c_2 - l_3 c_{23} & -l_3 c_{23} \\ -\zeta l_1 s_1 + l_3 c_1 c_{23} + l_2 c_2 c_1 & -l_3 s_1 s_{23} - l_2 s_2 s_1 & -l_3 s_1 s_{23} \\ \zeta l_1 c_1 + l_3 s_1 c_{23} + l_2 c_1 c_2 & l_3 c_1 s_{23} + l_2 c_1 s_2 & l_3 c_1 s_{23} \end{bmatrix} \begin{bmatrix} \dot{q}^1 \\ \dot{q}^2 \\ \dot{q}^3 \end{bmatrix} \quad (2.49)$$

根据向量加法法则, 足底坐标从本体系到世界系的映射关系为:

$${}^O_i\mathbf{p} = {}^O\mathbf{p}_{com} + {}^O R_B \cdot {}^B_i\mathbf{p} \quad (2.50)$$

为求足底速度从本体系到世界系的映射关系, 结合公式(2.32), 对公式(2.50)微分:

$$\begin{aligned} {}^O_i\dot{\mathbf{p}} &= {}^O\mathbf{v}_{com} + {}^O\dot{R}_B \cdot {}^B_i\mathbf{p} + {}^O R_B \cdot {}^B_i\dot{\mathbf{p}} \\ &= {}^O\mathbf{v}_{com} + {}^O R_B \cdot {}^B\boldsymbol{\omega}_{OB \times} \cdot {}^B_i\mathbf{p} + {}^O R_B \cdot {}^B_i\dot{\mathbf{p}} \\ &= {}^O\mathbf{v}_{com} + {}^O R_B \cdot ({}^B\boldsymbol{\omega}_{OB \times} \cdot {}^B_i\mathbf{p} + {}^B_i\dot{\mathbf{p}}) \end{aligned} \quad (2.51)$$

根据支撑足不打滑假设，足底在世界系下静止，即  ${}^0\dot{\boldsymbol{p}} = \mathbf{0}$ ，从而有：

$${}^0\boldsymbol{v}_{com} = -{}^0R_B \cdot ({}^B\boldsymbol{\omega}_{OBx} \cdot {}^B\boldsymbol{p} + {}^B\dot{\boldsymbol{p}}) \quad (2.52)$$

另外，足底高度  ${}^0p^z$  与足底坐标  ${}^0\boldsymbol{p}$  有如下关系：

$${}^0p^z = [0 \ 0 \ 1] \cdot {}^0\boldsymbol{p} \quad (2.53)$$

将公式(2.50)、(2.52)、(2.53)整合到一个方程中，可得卡尔曼滤波器观测方程为：

$$\underbrace{\begin{bmatrix} -{}^0R_B \cdot {}^B\boldsymbol{p} \\ -{}^0R_B \cdot {}^B\boldsymbol{p} \\ -{}^0R_B \cdot {}^B\boldsymbol{p} \\ -{}^0R_B \cdot {}^B\boldsymbol{p} \\ \hline -{}^0R_B \cdot ({}^B\boldsymbol{\omega}_{OBx} \cdot {}^B\boldsymbol{p} + {}^B\dot{\boldsymbol{p}}) \\ \hline {}^0p^z \\ {}^0p^z \\ {}^0p^z \\ {}^0p^z \end{bmatrix}}_{\boldsymbol{z}_k} = \underbrace{\begin{bmatrix} \mathbb{I}_3 & \mathbf{0}_{3 \times 3} \\ \mathbb{I}_3 & \mathbf{0}_{3 \times 3} & -\mathbb{I}_{12} \\ \mathbb{I}_3 & \mathbf{0}_{3 \times 3} \\ \mathbb{I}_3 & \mathbf{0}_{3 \times 3} \\ \hline \mathbf{0}_{3 \times 3} & \mathbb{I}_3 \\ \mathbf{0}_{3 \times 3} & \mathbb{I}_3 & \mathbf{0}_{12 \times 12} \\ \mathbf{0}_{3 \times 3} & \mathbb{I}_3 \\ \hline \mathbf{0}_{1 \times 6} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0}_{1 \times 9} \\ \mathbf{0}_{1 \times 9} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0}_{1 \times 6} \\ \mathbf{0}_{1 \times 12} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0}_{1 \times 3} \\ \mathbf{0}_{1 \times 15} & \mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0}_{1 \times 0} \end{bmatrix}}_H \underbrace{\begin{bmatrix} {}^0\boldsymbol{p}_{com} \\ {}^0\boldsymbol{v}_{com} \\ {}^0\boldsymbol{p} \\ {}^0\boldsymbol{p} \\ {}^0\boldsymbol{p} \\ {}^0\boldsymbol{p} \\ \hline {}^0p^z \end{bmatrix}}_{\boldsymbol{x}_k} \quad (2.54)$$

注意  $\mathbf{0}_{1 \times 0}$  为空矩阵，将公式(2.54)写成紧凑形式为：

$$\boldsymbol{z}_k = H \cdot \boldsymbol{x}_k \quad (2.55)$$

由于足底里程计求解质心位置是个积分过程，必然存在累计误差。在  $x$  方向与  $y$  方向的累计位置误差不会对控制器造成大的影响；但是在高度方向若有误差，就会直接体现在机器人行走时质心高度与抬腿高度的控制上。因此假设所有支撑足在世界系下高度  ${}^0p^z$  名义上为 0。在这一假设下，可以通过足底在本体系下的位置，反过来求解质心在世界系下的高度。即观测值中的足底高度项：

$${}^0p^z = 0 \quad (2.56)$$

尽管这一假设会导致机器人无法感知自己踩在了多高的平台上，但这并不会对最终的控制效果带来明显影响。至此，卡尔曼滤波器的状态方程与观测方程都已经给出，可以使用标准的卡尔曼滤波递归方程对系统进行估计：

$$\begin{cases} \hat{\boldsymbol{x}}_k = A\hat{\boldsymbol{x}}_{k-1} + B\boldsymbol{u}_{k-1} \\ \underline{P}_k = A\underline{P}_{k-1}A^T + Q \\ K_k = \frac{\underline{P}_k H^T}{H\underline{P}_k H^T + R} \\ \hat{\boldsymbol{x}}_k = \hat{\boldsymbol{x}}_k + K_k(\boldsymbol{z}_k - H\hat{\boldsymbol{x}}_k) \\ \underline{P}_k = (\mathbb{I} - K_k H)\underline{P}_k \end{cases} \quad (2.57)$$

其中,  $P_k$ 、 $P_k$ 、 $Q$ 、 $R$ 、 $K_k$ 、 $\mathbb{I}$ 分别为先验估计协方差矩阵、后验估计协方差矩阵、状态转移协方差矩阵、测量噪声协方差矩阵、卡尔曼增益、单位矩阵。

足底里程计假设了支撑足在地面上不打滑,而上述卡尔曼滤波器同时融合了支撑足与摆动足的运动学信息。为了降低摆动足运动学信息的贡献,需要在 $Q$ 矩阵和 $R$ 矩阵中将摆动足对应的协方差设置为极大的值。比如,若 $i$ 号腿是摆动腿,则有:

$$\begin{cases} Q_{block}(7+3(i-1), 7+3(i-1), 3, 3) = n_{big\_number} \cdot \mathbb{I}_3 \\ R_{block}(1+3(i-1), 1+3(i-1), 3, 3) = n_{big\_number} \cdot \mathbb{I}_3 \\ R_{block}(13+3(i-1), 13+3(i-1), 3, 3) = n_{big\_number} \cdot \mathbb{I}_3 \\ R_{block}(25+(i-1), 25+(i-1), 1, 1) = n_{big\_number} \end{cases} \quad (2.58)$$

公式(2.58)中,  $X_{block}(x, y, \Delta x, \Delta y)$ 是 $X$ 矩阵中从 $x$ 行 $y$ 列开始算的一个 $\Delta x$ 行 $\Delta y$ 列矩阵,在本文所述四足机器人上 $n_{big\_number} = 100$ 。

## 2.5 质心轨迹生成器

四足机器人控制算法多种多样,很难用单一的结构去概括所有的控制方案。尽管没有办法统一所有的控制系统,但总得来说,一个完整的控制系统总是会包含一个上层的运动规划和一个下层的运动跟随。

四足机器人简化为单刚体模型后,在空间中有6个自由度。为了完整描述机器人状态,每个自由度都需要描述位置信息与速度信息,因此完整描述某时刻机器人期望的状态需要12个变量,本节将从两个方面依次求解这12个变量。

### 2.5.1 基于水平速度命令的质心轨迹规划

本小节以手柄控制机器人水平运动为例,介绍水平速度指令如何转换为质心轨迹。如图21所示,使用Logitech F710上的摇杆控制机器人水平速度。操作者通过推动摇杆,可以指定机器人的期望速度 ${}^B_d\mathbf{v}_{handle} = [{}^B_dv_x \quad {}^B_dv_y \quad 0 \quad 0 \quad 0 \quad {}^B_d\omega_z]^T$ 。本文中,使用角标 $d$ 表示某一变量为期望值。 ${}^B_d\mathbf{v}_{handle}$ 是本体系下的期望速度,这是因为使用有头模式操作四足机器人更加符合人们的思维习惯。结合状态估计器与公式(2.22),不难得到此时的偏航角 $\psi_0$ 。设当前时刻为第0个控制周期,使用递推方法求第 $k$ 个控制周期的期望偏航角与期望角速度:

$$\begin{cases} d\psi^k = d\psi^{k-1} + {}^O_d\omega_z^{k-1} \cdot \Delta T \\ {}^O_d\omega_z^k = {}^B_d\omega_z \\ d\psi^0 = \psi_0 \\ {}^O_d\omega_z^0 = {}^B_d\omega_z \end{cases} \quad (2.59)$$

在本小节中仅考虑水平移动，忽略俯仰角与横滚角，因此将本体系下的偏航角速度近似为世界系下的偏航角速度。 $\Delta T$ 为MPC的控制周期。

本体系下期望质心速度为：

$${}^B \mathbf{v}_{com}^k = [ {}^B v_x \quad {}^B v_y \quad 0 ]^T \quad (2.60)$$

世界系下期望质心速度为：

$${}^O \mathbf{v}_{com}^k = R_z({}^d \psi^k) \cdot {}^B \mathbf{v}_{com}^k \quad (2.61)$$



图21 Logitech F710 上提供水平速度的摇杆

在上一控制周期已经求得了当前控制周期的期望位置  ${}^O \mathbf{p}_{com}^0$ ，结合此时的实际位置  ${}^O \mathbf{p}_{com}^0$ ，求  $k=1$  时刻的期望位置：

$${}^O \mathbf{p}_{com}^1 = \varpi \cdot {}^O \mathbf{p}_{com}^0 + (1 - \varpi) \cdot {}^O \mathbf{p}_{com}^0 + {}^O \mathbf{v}_{com}^0 \cdot \Delta T \quad (2.62)$$

其中  $\varpi$  是质心的“堵转保护”系数。设想若操作机器人撞向一堵墙，期望位置应该在一定程度上“跟随”实际位置，否则机器人会逐步加大推墙的力度，直到失控。 $\varpi$  越大，质心抵抗外力改变其位置的能力越强。

从  $k=1$  时刻的期望位置  ${}^O \mathbf{p}_{com}^1$  开始，递推后面的期望质心位置：

$${}^O \mathbf{p}_{com}^k = {}^O \mathbf{p}_{com}^{k-1} + {}^O \mathbf{v}_{com}^{k-1} \cdot \Delta T \quad (2.63)$$

另外，由于  ${}^O \mathbf{v}_{com}^k$  只会在  ${}^O \mathbf{p}_{com}^k$  的  $x$  方向和  $y$  方向积分， $z$  方向一般不设置期望速度，因此需要将  ${}^O \mathbf{p}_{com}^k$  的第 3 项用期望机身高度  $d h_{body}$  填充，即：

$${}^O \mathbf{p}_{com}^k(3) = d h_{body} \quad (2.64)$$

在本文中  $n(i)$  表示向量  $n$  的第  $i$  个元素。

### 2.5.2 地形坡度估计

上一小节根据手柄速度指令，求得了质心轨迹中的期望偏航速度  ${}^O \omega_z^k$ 、期望偏航角  ${}^d \psi^k$ 、期望质心速度  ${}^O \mathbf{v}_{com}^k$ 、期望质心位置  ${}^O \mathbf{p}_{com}^k$ 。在行走过程中，未设置由操作手主动操控的俯仰动作和横滚动作，因此期望俯仰速度与期望横滚速度设置为 0，即：

$${}^O \boldsymbol{\omega}^k = [0 \quad 0 \quad {}^O \omega_z^k]^T \quad (2.65)$$

当机器人在斜坡上行走时，摆动相的落足点在 $z$ 方向坐标不能简单规划为0。否则，在坡下的腿就会表现出如同踩到坑里，在坡上的腿就会表现出如同踩到石头上（如图22所示）这对机身控制的干扰非常明显。因此，有必要估算机器人所在地面的坡度，从而合理规划机器人的落足点高度坐标。

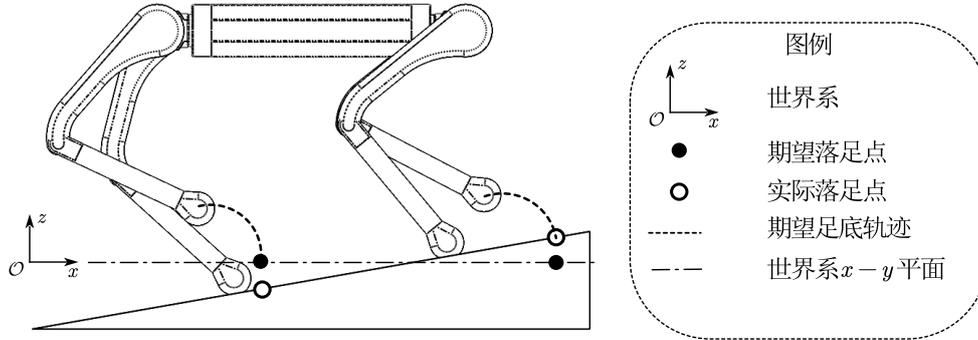


图22 机器人在斜坡上行走，期望落足点与实际落足点偏差较大

机器人不会在垂直于地面的墙面上行走，因此可以设地面的平面方程为：

$$z = a + bx + cy = [1 \ x \ y] A_{pla} \quad (2.66)$$

机器人每条腿都会交替成为支撑相和摆动相。记  ${}^o_i \mathbf{p}_{st \cdot end} = [x \ y \ z]^T$  为  $i$  号腿最后一次成为支撑腿时的足底坐标。即：若  $i$  号腿是支撑腿，则  ${}^o_i \mathbf{p}_{st \cdot end}$  为当前足底坐标；若  $i$  号腿是摆动腿，则  ${}^o_i \mathbf{p}_{st \cdot end}$  为最后一次支撑相切换为摆动相瞬间的足底坐标。 ${}^o_i \mathbf{p}_{st \cdot end}$  将由以下递推方法获得：

$$\begin{cases} {}^o_i \mathbf{p}_{st \cdot end}^0 = {}^o_i \mathbf{p}^{t=0} \\ {}^o_i \mathbf{p}_{st \cdot end}^k = {}^o_i \mathbf{p}^k, \quad \rho S_{\phi} = 1 \\ {}^o_i \mathbf{p}_{st \cdot end}^k = {}^o_i \mathbf{p}_{st \cdot end}^{k-1}, \quad \rho S_{\phi} = 0 \end{cases} \quad (2.67)$$

将  ${}^o_i \mathbf{p}_{st \cdot end}$  的  $z$  坐标与  $xy$  坐标分离，并分别填充到两个矩阵中：

$${}^o \mathbf{z}_f = [{}^o_1 \mathbf{p}_{st \cdot end}(3) \quad {}^o_2 \mathbf{p}_{st \cdot end}(3) \quad {}^o_3 \mathbf{p}_{st \cdot end}(3) \quad {}^o_4 \mathbf{p}_{st \cdot end}(3)]^T \quad (2.68)$$

$$W_{pla} = \begin{bmatrix} 1 & {}^o_1 \mathbf{p}_{st \cdot end}(1) & {}^o_1 \mathbf{p}_{st \cdot end}(2) \\ 1 & {}^o_2 \mathbf{p}_{st \cdot end}(1) & {}^o_2 \mathbf{p}_{st \cdot end}(2) \\ 1 & {}^o_3 \mathbf{p}_{st \cdot end}(1) & {}^o_3 \mathbf{p}_{st \cdot end}(2) \\ 1 & {}^o_4 \mathbf{p}_{st \cdot end}(1) & {}^o_4 \mathbf{p}_{st \cdot end}(2) \end{bmatrix} \quad (2.69)$$

利用最小二乘法计算地面的平面方程：

$$A_{pla} = W_{pla}^+ \cdot {}^o \mathbf{z}_f \quad (2.70)$$

其中  $W_{pla}^+$  表示  $W_{pla}$  的伪逆矩阵，为  $A_{pla}$  设计一个低通滤波器：

$$\hat{A}_{pla}^k = [\hat{a} \ \hat{b} \ \hat{c}]^T = \eta A_{pla} + (1 - \eta) \hat{A}_{pla}^{k-1} \quad (2.71)$$

其中  $\eta$  为低通滤波系数，本文中  $\eta = 0.2$ 。根据平面方程的性质，地面的法向量为

$\mathbf{n} = [-\hat{b} \ -\hat{c} \ 1]^T$ ，将其单位化：

$$\mathbf{n}_e = \frac{\mathbf{n}}{|\mathbf{n}|} \quad (2.72)$$

根据公式(2.9)， $\mathbf{n}_e$  是机体期望旋转矩阵  ${}^dR_B$  的第三列，结合公式(2.20)：

$$\mathbf{n}_e = \begin{bmatrix} S_\phi S_\psi + C_\phi C_\psi S_\theta \\ C_\phi S_\theta S_\psi - C_\psi S_\phi \\ C_\phi C_\theta \end{bmatrix} = \underbrace{\begin{bmatrix} C_\psi & S_\psi & 0 \\ S_\psi & -C_\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}}_Y \underbrace{\begin{bmatrix} C_\phi S_\theta \\ S_\phi \\ C_\phi C_\theta \end{bmatrix}}_\xi = Y\xi \quad (2.73)$$

在上一小节中，期望偏航角  ${}^d\psi^1$  已经求出，因此矩阵  $Y$  已知，可解方程 (2.73)：

$$\xi = Y^+ \mathbf{n}_e \quad (2.74)$$

根据公式(2.73)中  $\xi$  的定义，可得机器人期望俯仰角与期望横滚角：

$$\begin{cases} {}^d\phi^1 = \arcsin(\xi(2)) \\ {}^d\theta^1 = \arctan(\xi(1)/\xi(3)) \end{cases} \quad (2.75)$$

此处认为期望俯仰角和期望横滚角不时变，将期望欧拉角整合到一个向量  ${}^d\Theta^k = [{}^d\phi^1 \ {}^d\theta^1 \ {}^d\psi^k]^T$ ，则第  $k$  个控制周期机器人期望状态为：

$${}^d\mathbf{x}^k = \begin{bmatrix} {}^d\Theta^k \\ {}^d\mathbf{p}_{com}^k \\ {}^d\boldsymbol{\omega}^k \\ {}^d\mathbf{v}_{com}^k \\ {}^d\mathbf{g}(3) \end{bmatrix} \quad (2.76)$$

公式(2.76)用 13 个参数描述了第  $k$  个控制周期机器人的期望状态，其中最后一个参数  ${}^d\mathbf{g}(3) = -9.8m/s^2$  为常数，将在下一章介绍状态向量中加入该常数的原因。将未来  $h$  个控制周期的状态组合称一个向量：

$$D = [({}^d\mathbf{x}^1)^T \ ({}^d\mathbf{x}^2)^T \ \dots \ ({}^d\mathbf{x}^h)^T]^T \quad (2.77)$$

则称向量  $D$  为机器人的期望轨迹，其大小为  $13 \times h$  行 1 列。

## 2.6 摆动相规划

摆动相规划主要有两方面的约束条件：一方面要保证足够的通过性，在崎岖不平的路面上，摆动足底平均高度要足够高，否则会碰到障碍物；另一方面要保证动作轻盈，不能对整体的控制带来大的干扰。本文将摆动相规划分为落足点规划和足底轨迹规划两个环节，接下将来用两个小节分别介绍。

### 2.6.1 落足点规划

落足点是指摆动腿足底轨迹规划的终点。如图23所示，若机器人水平移动速度不变，

机器人的摆动足将跟随一个有对称性的足底轨迹，对称点的 $x$ 、 $y$ 坐标为：

$${}^B_i \mathbf{p}_{hip} = \begin{bmatrix} \delta h_x \\ \zeta(h_y + l_1) \\ 0 \end{bmatrix} + \begin{bmatrix} p_{offset}^x \\ 0 \\ 0 \end{bmatrix} \quad (2.78)$$

该值包含两项，第一项是在第一关节角度为0时，第二连杆刚体坐标系的坐标原点在本体系下的水平坐标；第二项是在 $x$ 方向的偏移量，由于四足机器人是全肘式结构，实际机器人质心偏后，因此落足点应稍稍靠后，否则在踏步时机器人会在重力作用下有后退趋势。本文中 $p_{offset}^x = -0.018m$ 。从公式(2.78)可见，对称点刚接到本体系。

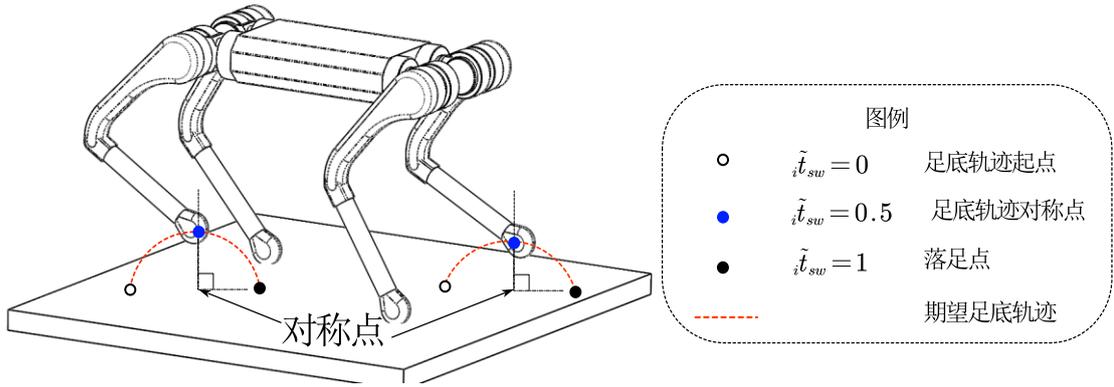


图23 对称点为 $\tilde{t}_{sw} = 0.5$ 时刻的期望足底坐标（仅考虑 $x$ 、 $y$ 坐标）

此时系统时间是 $t$ ，则 $i$ 号摆动腿名义触地时刻为 $t_{touch} = t + (1 - \tilde{t}_{sw})t_{sw}$ ，可以通过公式(2.63)的递推方法估算 $t_{touch}$ 时刻质心坐标，但是需要大量计算矩阵乘法，计算量较大。在精度允许的情况下，可以采用一种近似的方法求触地时刻躯干的位姿：

$$\begin{cases} {}^O \mathbf{p}_{com}^{touch} = {}^O \mathbf{p}_{com}^t + {}^O R_B^t \cdot {}^B \mathbf{v}_{com}^t \cdot (1 - \tilde{t}_{sw})t_{sw} \\ \psi^{touch} = \psi^t + {}_d \omega_z^t \cdot (1 - \tilde{t}_{sw})t_{sw} \end{cases} \quad (2.79)$$

其中 ${}^O \mathbf{p}_{com}^t$ 、 ${}^O R_B^t$ 、 ${}^B \mathbf{v}_{com}^t$ 、 $\psi^t$ 、 ${}_d \omega_z^t$ 分别为当前时刻的质心坐标、旋转矩阵、期望质心速度、偏航角、期望偏航角速度。本文中，使用右上标 $t$ 表示某一变量为当前值，使用右上标 $touch$ 表示名义触地时刻的值。尽管这是一种近似方法，但随着 $\tilde{t}_{sw}$ 逐渐逼近1， ${}^O \mathbf{p}_{com}^t$ 也将逐渐逼近 ${}^O \mathbf{p}_{com}^{touch}$ ，因此其误差将逐渐减少，并在名义触地时刻消失。名义触地时刻对称点坐标为：

$${}^O_i \mathbf{p}_{hip}^{touch} = {}^O \mathbf{p}_{com}^{touch} + R_z(\psi^{touch}) {}^B_i \mathbf{p}_{hip} \quad (2.80)$$

如图24所示，为了让本体系下支撑足开始时刻和结束时刻坐标的平均值为对称点，需要将落足点设置在实际速度方向且距离对称点半个步长的位置，即：

$$\begin{cases} \Delta p_1 = {}^0 v_{com}^t \cdot t_{st}/2 \\ \Delta p_2 = R_z(\psi^{touch}) [R_z({}^d \omega_z^t \cdot t_{st}/2) \cdot {}^B p_{hip} - {}^B p_{hip}] \end{cases} \quad (2.81)$$

其中 $\Delta p_1$ 为平动项， $\Delta p_2$ 为转动项。

根据Raibert的理论<sup>[8]</sup>，在一个方向增加落足点的距离，将自然引发反方向的加速度，这是足式机器人系统的内在特性。利用这一规律可以修正质心的速度跟踪误差：

$$\Delta p_3 = k_p ({}^0 v_{com}^t - {}^0 v_{com}^t) \quad (2.82)$$

其中 $k_p$ 为比例系数，本文中 $k_p = 0.15$ 。

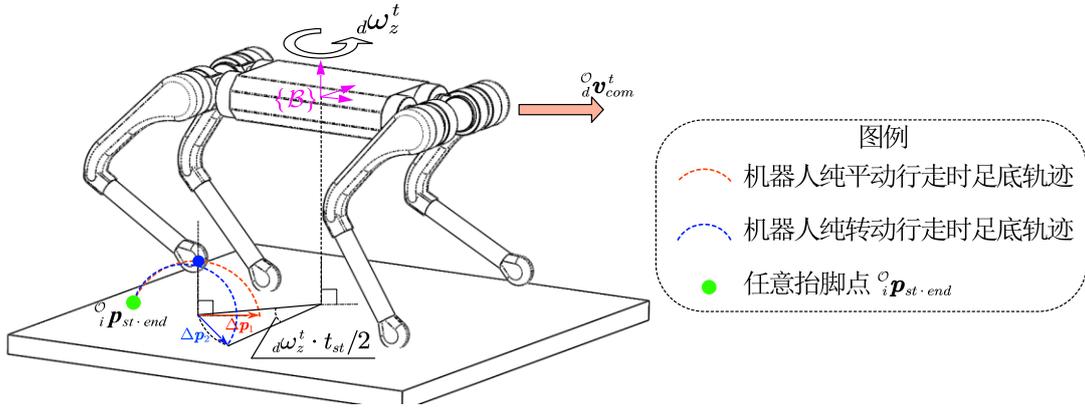


图24 落足点设置在实际速度方向，距离对称点半个步长的位置

如图25所示，可以简化为倒立摆的移动物体，在绕垂直地面的轴做匀速圆周运动时，地面支撑点相对于质心总是在远离圆心的方向。这一特性同样要体现在落足点规划上：

$$\Delta p_4 = \frac{{}^0 p_{com}^t(3)}{-{}^0 g(3)} {}^0 v_{com}^t \times {}^0 \omega_{OB}^t \quad (2.83)$$

其中 ${}^0 g = [0 \ 0 \ -9.8m/s^2]^T$ 为重力加速度， $a = -{}^0 v_{com}^t \times {}^0 \omega_{OB}^t$ 为空间刚体匀速圆周运动的向心加速度，对公式(2.23)两边同时微分，忽略角加速度，可得到该结论：

$$\begin{aligned} {}^A \ddot{\mathbf{r}}_{AP} &= {}^A \dot{\Omega}_B {}^A \mathbf{r}_{AP} + {}^A \Omega_B {}^A \dot{\mathbf{r}}_{AP} \\ &= \mathbf{0}_{3 \times 3} {}^A \mathbf{r}_{AP} + {}^A \Omega_B {}^A \dot{\mathbf{r}}_{AP} \\ &= {}^A \omega_{AB} \times {}^A \dot{\mathbf{r}}_{AP} \\ &= -{}^A \dot{\mathbf{r}}_{AP} \times {}^A \omega_{AB} \end{aligned} \quad (2.84)$$

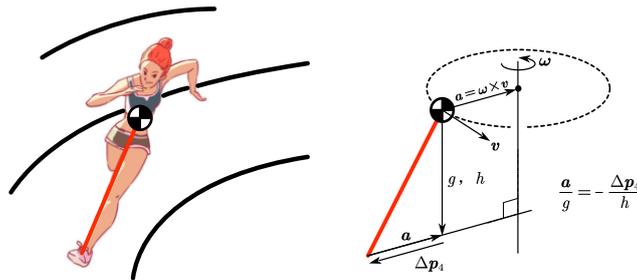


图25 匀速圆周运动的倒立摆，地面支撑点相对于质心总是在远离圆心的方向

综上分析，落足点在 $x$ 、 $y$ 方向的坐标为：

$${}^0\mathbf{p}_{sw \cdot end} = {}^0\mathbf{p}_{hip}^{touch} + \Delta\mathbf{p}_1 + \Delta\mathbf{p}_2 + \Delta\mathbf{p}_3 + \Delta\mathbf{p}_4 \quad (2.85)$$

将 $x$ 坐标与 $y$ 坐标代入(2.66)可得 $z$ 方向坐标：

$${}^0\mathbf{p}_{sw \cdot end}(3) = a + b \cdot {}^0\mathbf{p}_{sw \cdot end}(1) + c \cdot {}^0\mathbf{p}_{sw \cdot end}(2) \quad (2.86)$$

## 2.6.2 足底轨迹规划

在前文地形坡度估计中，定义了 ${}^0\mathbf{p}_{st \cdot end}$ 为 $i$ 号腿最后一次成为支撑腿时的足底坐标，对于摆动腿来说， ${}^0\mathbf{p}_{st \cdot end}$ 是 $\tilde{t}_{sw} = 0$ 时刻足底坐标，也是足底轨迹的起点，在前文中已经求得了落足点 ${}^0\mathbf{p}_{sw \cdot end}$ 作为足底轨迹的终点，因此足底轨迹的规划是点到点的规划问题。

### (1) 点到点的最优轨迹

考虑如图26(a)所示物理系统，一个单位质量的滑块放置在光滑地面上，受到一个水平方向的作用力 $u$ 。初始时刻滑块位置与速度都为0，要求在 $u$ 的作用下，在单位时间内走过单位距离并停止。这个问题存在着无数种可能的运动轨迹（如图26(b)所示），现在希望从中选出最优的一条。

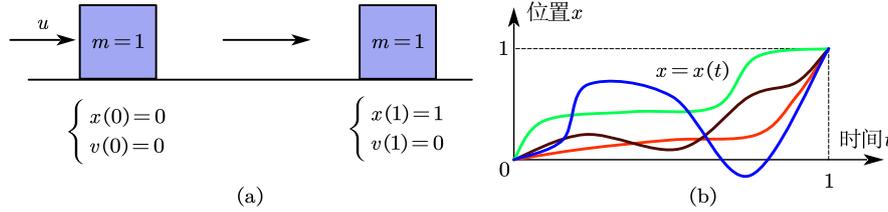


图26 在 $u$ 的作用下，单位质量的滑块在单位时间内走过单位距离并停止

该系统动力学方程为：

$$\dot{x} = v, \quad \ddot{x} = \dot{v} = u \quad (2.87)$$

用如下指标评价轨迹优劣：

$$J = \int_0^1 u^2 dt = \int_0^1 \ddot{x}^2 dt \quad (2.88)$$

该评价指标表明，若 $J$ 足够小，意味着实现轨迹 $x=x(t)$ 的过程不需要太大的力。可以采用变分法求解满足 $J$ 最小的函数 $x(t)$ ，构造一个函数：

$$\mathcal{L}(t, x, \dot{x}, \ddot{x}) = \ddot{x}^2 \quad (2.89)$$

由于 $\mathcal{L}$ 是待求函数 $x(t)$ 二阶导的函数，而传统的欧拉-拉格朗日方程只能处理最多一阶导，因此应用推广的欧拉-拉格朗日方程<sup>[43]</sup>：

$$\frac{\partial \mathcal{L}}{\partial x} - \frac{d}{dt} \frac{\partial \mathcal{L}}{\partial \dot{x}} - \frac{d^2}{dt^2} \frac{\partial \mathcal{L}}{\partial \ddot{x}} = 0 \quad (2.90)$$

即：

$$(0) - (0) - \frac{d^2}{dt^2} (2\ddot{x}) = 0 \quad (2.91)$$

从而有：

$$\frac{d^4}{dt^4} x = 0 \quad (2.92)$$

由公式(2.92)可知 $x(t)$ 是 $t$ 的三次函数，令：

$$\begin{cases} x(t) = at^3 + bt^2 + ct + d \\ v(t) = 3at^2 + 2bt + c \end{cases} \quad (2.93)$$

将如下边界条件代入(2.93)：

$$\begin{cases} x(0) = 0 \\ v(0) = 0 \end{cases}, \begin{cases} x(1) = 1 \\ v(1) = 0 \end{cases} \quad (2.94)$$

可得：

$$\begin{cases} x(t) = 3t^2 - 2t^3 \\ v(t) = 6t - 6t^2 \end{cases} \quad (2.95)$$

## (2) 足底轨迹规划

如图27所示，红色曲线为期望足底轨迹，蓝色箭头为从轨迹起点到轨迹终点的向量：

$${}^O_i \mathbf{p}_{st \cdot sw} = {}^O_i \mathbf{p}_{sw \cdot end} - {}^O_i \mathbf{p}_{st \cdot end} \quad (2.96)$$

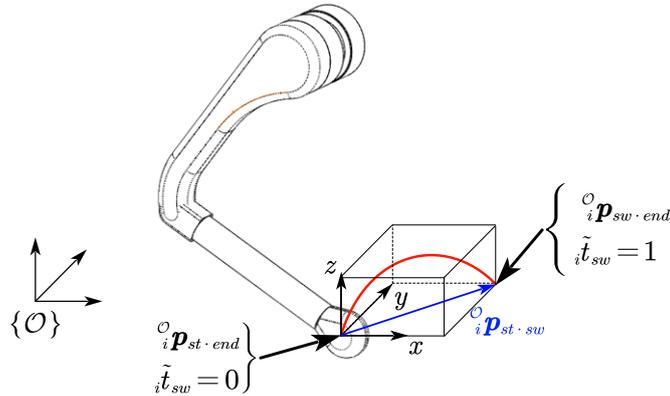


图27 足底轨迹

根据上一小节点到点的最优轨迹曲线，可分别对 $x$ 、 $y$ 两个维度做规划：

$$\begin{cases} dx(\tilde{t}_{sw}) = {}^O_i \mathbf{p}_{st \cdot end}(1) + {}^O_i \mathbf{p}_{st \cdot sw}(1) \cdot (3 \cdot \tilde{t}_{sw}^2 - 2 \cdot \tilde{t}_{sw}^3) \\ dy(\tilde{t}_{sw}) = {}^O_i \mathbf{p}_{st \cdot end}(2) + {}^O_i \mathbf{p}_{st \cdot sw}(2) \cdot (3 \cdot \tilde{t}_{sw}^2 - 2 \cdot \tilde{t}_{sw}^3) \\ d\dot{x}(\tilde{t}_{sw}) = {}^O_i \mathbf{p}_{st \cdot sw}(1) \cdot (6 \cdot \tilde{t}_{sw} - 6 \cdot \tilde{t}_{sw}^2) \\ d\dot{y}(\tilde{t}_{sw}) = {}^O_i \mathbf{p}_{st \cdot sw}(2) \cdot (6 \cdot \tilde{t}_{sw} - 6 \cdot \tilde{t}_{sw}^2) \end{cases} \quad (2.97)$$

$z$ 方向的轨迹规划分为抬腿部分与落腿部分，因此是一个分段函数：

$$\begin{cases} dz(\tilde{t}_{sw}) = {}^o_i\mathbf{p}_{st \cdot end}(3) + d h_{foot} \cdot (3\sigma^2 - 2\sigma^3), & \sigma = 2 \cdot \tilde{t}_{sw} \leq 1 \\ dz(\tilde{t}_{sw}) = dz(0.5) + [{}^o_i\mathbf{p}_{sw \cdot end}(3) - dz(0.5)] \cdot (3\sigma^2 - 2\sigma^3), & \sigma = 2(\tilde{t}_{sw} - 0.5) > 0 \\ d\dot{z}(\tilde{t}_{sw}) = d h_{foot} \cdot (6\sigma - 6\sigma^2), & \sigma = 2 \cdot \tilde{t}_{sw} \leq 1 \\ d\dot{z}(\tilde{t}_{sw}) = [{}^o_i\mathbf{p}_{sw \cdot end}(3) - dz(0.5)] \cdot (6\sigma - 6\sigma^2), & \sigma = 2(\tilde{t}_{sw} - 0.5) > 0 \end{cases} \quad (2.98)$$

其中  $d h_{foot}$  为期望抬腿高度。结合(2.97)与(2.98)可得摆动足期望足底轨迹:

$$\begin{cases} {}^o_i\mathbf{p}^d = [dx(\tilde{t}_{sw}) \quad dy(\tilde{t}_{sw}) \quad dz(\tilde{t}_{sw})]^T \\ {}^o_i\dot{\mathbf{p}}^d = [d\dot{x}(\tilde{t}_{sw}) \quad d\dot{y}(\tilde{t}_{sw}) \quad d\dot{z}(\tilde{t}_{sw})]^T \end{cases} \quad (2.99)$$

图28为  ${}^o_i\mathbf{p}^d$  在  $x$ 、 $y$ 、 $z$  三个方向的曲线。

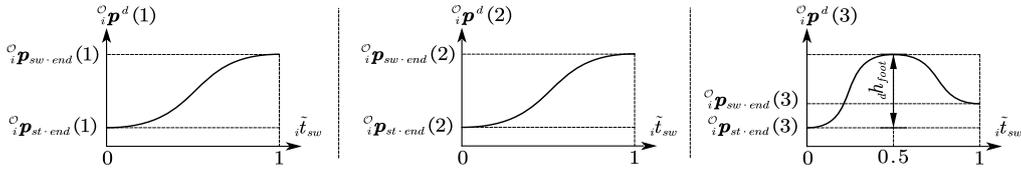


图28  ${}^o_i\mathbf{p}^d$  在  $x$ 、 $y$ 、 $z$  三个方向的曲线

## 2.7 本章小节

本章主要介绍四足机器人控制系统基本算法。首先介绍了用于机器人行为管理的有限状态机；然后介绍了刚体坐标系、世界系、本体系 and 足底坐标系，以及刚体的位姿描述方法；随后介绍了用于支撑相与摆动相切换的步态调度器；然后介绍了状态估计器，其中包含一个基于足底不打滑假设的足底里程计；然后介绍了由手柄速度命令转化为质心轨迹的质心轨迹规划器，以及一个估计机器人所踩地平面倾斜程度的地面坡度估计算法；最后介绍了摆动相规划算法，包含落足点规划和足底轨迹规划。

### 第三章 仿生四足机器人模型预测控制 (MPC)

机器人控制系统，就是要解决机器人在哪儿、去哪儿、怎么去的问题，分别对应了机器人控制系统的状态估计器、轨迹规划器和控制器。在第二章中已经详细阐述了状态估计和轨迹规划，本章主要介绍控制器中的模型预测控制器。本文采用了一种将模型预测控制 (MPC) 与全身控制 (WBC) 相结合的控制方案：MPC 使用简化的单刚体模型，在较长时间范围内求解最优的足底反力；WBC 采用多刚体模型，将整体的控制任务按照优先级分为 4 个小的控制任务，以前文中规划的轨迹作为跟踪目标，在每一个控制周期解算出电机指令。

#### 3.1 模型预测控制在间歇欠驱系统中的必要性

图29为一名运动员跃过障碍物示意图，该过程可简单分为三个阶段：起跳阶段，飞行阶段，落地阶段。若将运动员看作一个系统，忽略空气阻力的影响，该系统所受外力仅包含地面作用到脚上的足底反力和重力。在腾空阶段，系统仅受重力的作用，因此质心在腾空阶段的移动轨迹是一条抛物线，无论运动员的肌肉如何发力，都无法改变质心的移动轨迹，这是典型的欠驱系统。在起跳阶段与落地阶段，运动员可以通过控制肌肉发力来控制足底反力，此时质心的位置又是可控的，称这种某些自由度间歇性进入欠驱状态的系统为间歇欠驱系统。

为了越过障碍物，运动员跳的是否足够高、足够远，完全取决于在起跳阶段脚底受到地面的作用力有多大。而足底反力又取决于运动员肌肉如何发力，这就要求运动员在起跳阶段就得考虑好未来一段时间自己质心位置的变化。类似的，尽管在腾空阶段运动员无法改变质心轨迹，但可以通过调整四肢位置，为即将到来的落地做准备。



图29 运动员越过障碍

机器人的控制器同样应该有类似的效果：在当前控制周期，根据反馈与期望值求控制量时，不仅要实现机器人跟踪当前控制周期的期望位置，还要求出未来一段时间的多个控制量，以实现机器人跟踪未来一段时间的轨迹，这就要求控制器根据控制量和系统模型预测未来一段时间机器人的状态。

足式机器人运动往往与跳跃这种间歇欠驱系统类似。以最常见的四足机器人对角步态为例，如图30(a)所示，机器人右前腿和左后腿为摆动腿，其足端与地面无相互作用。支撑足可以简化为一个球，将两个支撑足的球心连一条线，称为支撑线。由于足底反力的作用点都在支撑线上，因此无论足底反力有多大，都不可能产生绕支撑线的扭矩，即此时机器人绕支撑线旋转的方向上是欠驱的。在半个迈步周期后，支撑腿与摆动腿切换，支撑线变为右前腿和左后腿足端球心的连线，在绕该支撑线旋转的方向上，机器人依然是欠驱的。

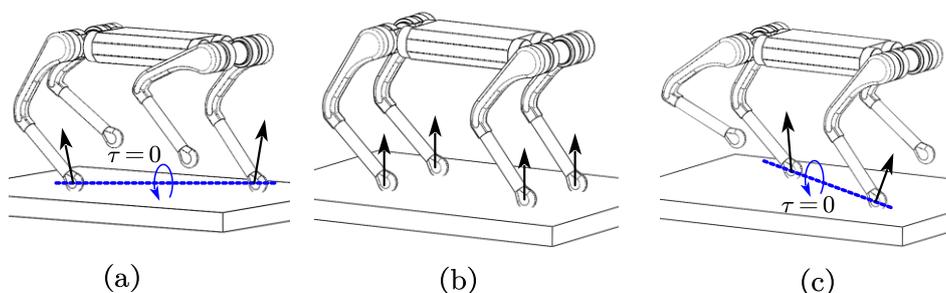


图30 足底反力作用点在支撑线上，无法产生绕支撑线的扭矩

为了保证机器人在欠驱方向上不失稳，在某一方向成为欠驱方向之前，控制器应该提前为即将到来的欠驱状态做准备；在某一方向即将脱离欠驱状态之前，控制器亦应该为即将到来的可驱状态做准备。因此，四足机器人对角步态的控制器需要有一定的预测特性。

### 3.2 单刚体动力学建模

在每一个控制周期，MPC 控制器都对未来一段时间进行预测，并选出其中最优的控制量序列。这相当于对系统做了多次仿真，因此计算量非常大。为了简化计算，忽略因关节位置变化导致的腿部质量分布变化，将四足机器人建模为一个单刚体模型。考虑单刚体在空间中 6 个自由度的位置与速度，仅用 12 个变量就可以完整描述四足机器人的状态。相关研究<sup>[44]</sup>表明，这种简化足以将 MPC 的计算量降低到可接受的水平，同时不会损失太多控制精度。

### 3.2.1 平动动力学与转动动力学

如图31所示, 忽略因关节位置变换导致的腿部质量分布变化, 将四足机器人建模为单刚体并对其受力分析, 其中黑色实心点为足底位置, 黑色箭头为质心指向足底位置的向量, 红色箭头为足底反力, 品红色箭头为刚体所受重力。根据牛顿第二定律, 在世界系下刚体平动加速度为:

$${}^o\ddot{\mathbf{p}}_{com} = \frac{\sum_{i=1}^4 {}^o\mathbf{f}_i}{m} + {}^o\mathbf{g} \quad (3.1)$$

根据角动量定理, 刚体在质心处的角动量对时间的微分, 等于作用于该刚体的所有外力在质心处产生的力矩的矢量和, 即:

$$\frac{d}{dt} ({}^P I \cdot {}^o\boldsymbol{\omega}_{OB}) = \sum_{i=1}^4 {}^o\mathbf{r}_i \times {}^o\mathbf{f}_i + \mathbf{0}_{3 \times 1} \times (m \cdot {}^o\mathbf{g}) \quad (3.2)$$

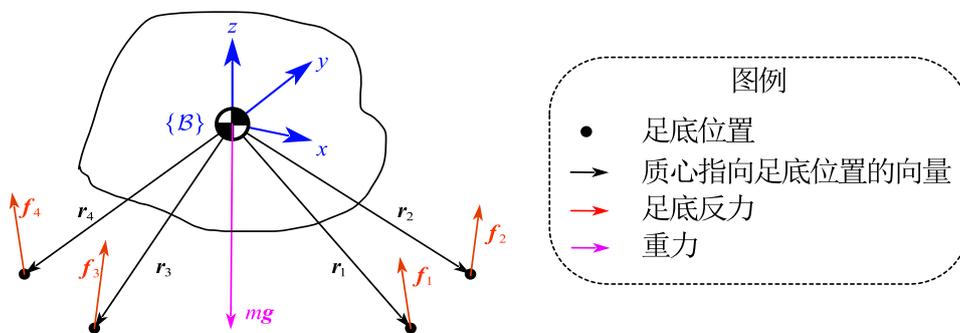


图31 单刚体模型受力分析

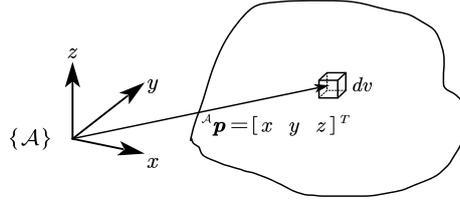
### 3.2.2 惯性张量的转动变换

其中 ${}^P I$ 为刚体的惯性张量在定向本体系 $\mathcal{P}$ 下的描述, 定向本体系的方向与世界系相同, 坐标原点与本体系重合。如图32所示, 在坐标系 $\mathcal{A}$ 中有一刚体, 体积为 $V$ , 可由无数个体积为 $dv$ , 密度为 $\rho$ 的体积元组成, 每个体积元的位置由矢量 ${}^A\mathbf{p} = [x \ y \ z]^T$ 指定。则刚体的惯性张量定义为:

$${}^A I = \begin{bmatrix} I_{xx} & -I_{xy} & -I_{xz} \\ -I_{xy} & I_{yy} & -I_{yz} \\ -I_{xz} & -I_{yz} & I_{zz} \end{bmatrix} \quad (3.3)$$

${}^A I$ 矩阵中各元素为:

$$\begin{cases} I_{xx} = \iiint_V (y^2 + z^2) \rho dv \\ I_{yy} = \iiint_V (x^2 + z^2) \rho dv \\ I_{zz} = \iiint_V (x^2 + y^2) \rho dv \end{cases}, \begin{cases} I_{xy} = \iiint_V (xy) \rho dv \\ I_{xz} = \iiint_V (xz) \rho dv \\ I_{yz} = \iiint_V (yz) \rho dv \end{cases} \quad (3.4)$$

图32 刚体可由无数个体积为 $dv$ 的体积元组成

由公式(3.4)可以看出，惯性张量描述了质量在刚体中的分布情况。考虑到在定向本体系下，刚体的惯性张量是其姿态的函数，而本体系下的惯性张量是常值，因此需要推导惯性张量从本体系到定向本体系的映射关系。设刚体的角动量与角速度分别为 $l$ 、 $\omega_{OB}$ ，由于角动量的微分是三维力矩，因此角动量是三维矢量。考虑定向本体系下的角动量为：

$$\begin{aligned}
 {}^P I &= {}^P R_B \cdot {}^B I \\
 &= {}^P R_B \cdot {}^B I \cdot {}^B \omega_{OB} \\
 &= {}^P R_B \cdot {}^B I \cdot {}^P R_B^T \cdot {}^P R_B \cdot {}^B \omega_{OB} \\
 &= ({}^P R_B \cdot {}^B I \cdot {}^P R_B^T) ({}^P R_B \cdot {}^B \omega_{OB}) \\
 &= {}^P I \cdot {}^P \omega_{OB}
 \end{aligned} \tag{3.5}$$

由于定向本体系与世界系平行，即 ${}^P R_B = {}^O R_B$ ，从而有：

$${}^P I = {}^O R_B \cdot {}^B I \cdot {}^O R_B^T \tag{3.6}$$

在三维建模软件中，设置好装配体各个零部件的密度，可直接导出其惯性张量，本文所述四足机器人惯性张量为：

$${}^B I = \begin{bmatrix} 0.31 & 0 & -0.05 \\ 0 & 1.09 & 0 \\ -0.05 & 0 & 1.12 \end{bmatrix} \text{kg} \cdot \text{m}^2 \tag{3.7}$$

公式(3.2)中 ${}^O r_i$ 为质心指向足底坐标的向量，可由状态估计器求出的质心位置以及公式(2.50)求出的足底坐标直接做差求解：

$${}^O r_i = {}^O_i p - {}^O_i p_{com} \tag{3.8}$$

### 3.2.3 足底反力约束

公式(3.2)中 ${}^O f_i$ 为足底反力，当第 $i$ 条腿为摆动腿时，足底反力为0：

$${}^O f_i = \mathbf{0}_{3 \times 1}, \quad \forall_i s_\phi = 0 \tag{3.9}$$

不能有无穷大的足底反力作用到机器人上，因此限制足底反力竖直方向幅值：

$${}^O f_i^z \leq f_{\max} \tag{3.10}$$

如图33所示，为保证足底与地面不发生相对滑动，足底反力的水平分量不能大于其竖直分量与滑动摩擦系数 $\mu$ 的乘积，即满足摩擦锥条件：

$$\mu \cdot {}^o f_i^z \geq \sqrt{({}^o f_i^x)^2 + ({}^o f_i^y)^2} \quad (3.11)$$

由于公式(3.11)是非线性约束,不利于计算。在满足使用的情况下,可以将其拆分为4个线性约束,结合公式(3.10),将足底反力约束整合到一个矩阵中:

$$\underbrace{\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}}_{\underline{c}_i} \leq \underbrace{\begin{bmatrix} -1 & 0 & \mu \\ 0 & -1 & \mu \\ 1 & 0 & \mu \\ 0 & 1 & \mu \\ 0 & 0 & 1 \end{bmatrix}}_{C_i} \underbrace{\begin{bmatrix} {}^o f_i^x \\ {}^o f_i^y \\ {}^o f_i^z \end{bmatrix}}_{\mathbf{f}_i} \leq \underbrace{\begin{bmatrix} +\infty \\ +\infty \\ +\infty \\ +\infty \\ f_{\max} \end{bmatrix}}_{\bar{c}_i} \quad (3.12)$$

将公式(3.12)写称紧凑形式:

$$\underline{c}_i \leq C_i \cdot {}^o \mathbf{f}_i \leq \bar{c}_i \quad (3.13)$$

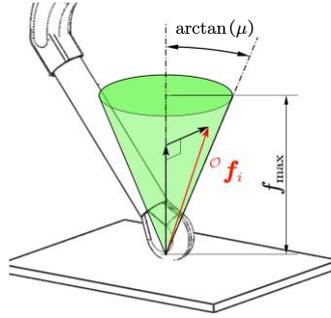


图33 足底反力约束条件

从公式(3.12)可以看出,  $\underline{c}_i$ 、 $C_i$ 、 $\bar{c}_i$ 中具体内容与 $i$ 无关。

### 3.2.4 近似欧拉方程

由于 ${}^B I$ 是常数,结合公式(2.23)对公式(3.6)两边微分:

$$\begin{aligned} {}^P \dot{J} &= {}^o \dot{R}_B \cdot {}^B I \cdot {}^o R_B^T + {}^o R_B \cdot {}^B I \cdot {}^o \dot{R}_B^T \\ &= ({}^o \boldsymbol{\omega}_{OB \times} \cdot {}^o R_B) \cdot {}^B I \cdot {}^o R_B^T + {}^o R_B \cdot {}^B I \cdot ({}^o R_B^T \cdot ({}^o \boldsymbol{\omega}_{OB \times})^T) \\ &= {}^o \boldsymbol{\omega}_{OB \times} \cdot {}^P I + {}^P I \cdot ({}^o \boldsymbol{\omega}_{OB \times})^T \\ &= {}^o \boldsymbol{\omega}_{OB \times} \cdot {}^P I - {}^P I \cdot {}^o \boldsymbol{\omega}_{OB \times} \end{aligned} \quad (3.14)$$

因此公式(3.2)左边角动量对时间的微分为:

$$\begin{aligned} \frac{d}{dt} ({}^P I \cdot {}^o \boldsymbol{\omega}_{OB}) &= {}^P I \cdot {}^o \dot{\boldsymbol{\omega}}_{OB} + {}^P \dot{J} \cdot {}^o \boldsymbol{\omega}_{OB} \\ &= {}^P I \cdot {}^o \dot{\boldsymbol{\omega}}_{OB} + ({}^o \boldsymbol{\omega}_{OB \times} \cdot {}^P I - {}^P I \cdot {}^o \boldsymbol{\omega}_{OB \times}) \cdot {}^o \boldsymbol{\omega}_{OB} \\ &= {}^P I \cdot {}^o \dot{\boldsymbol{\omega}}_{OB} + {}^o \boldsymbol{\omega}_{OB \times} \cdot {}^P I \cdot {}^o \boldsymbol{\omega}_{OB} - {}^P I \cdot {}^o \boldsymbol{\omega}_{OB \times} \cdot {}^o \boldsymbol{\omega}_{OB} \\ &= {}^P I \cdot {}^o \dot{\boldsymbol{\omega}}_{OB} + {}^o \boldsymbol{\omega}_{OB \times} \cdot {}^P I \cdot {}^o \boldsymbol{\omega}_{OB} \\ &\approx {}^P I \cdot {}^o \dot{\boldsymbol{\omega}}_{OB} \end{aligned} \quad (3.15)$$

在刚体角速度很小的情况下,公式(3.15)中 ${}^o \boldsymbol{\omega}_{OB \times} \cdot {}^P I \cdot {}^o \boldsymbol{\omega}_{OB}$ 项可以忽略,从而得到线性的欧拉方程,实践<sup>[40, 45]</sup>表明这种近似足以满足控制需求。

### 3.2.5 角速度到 ZYX 欧拉角微分的映射

定义  $\mathbf{e}_1$ 、 $\mathbf{e}_2$ 、 $\mathbf{e}_3$  为  ${}^oR_B$  的三列，结合公式(2.20)有：

$${}^oR_B(\Theta) = R_z(\psi)R_y(\theta)R_x(\phi) = [\mathbf{e}_1 \ \mathbf{e}_2 \ \mathbf{e}_3] \quad (3.16)$$

根据公式(2.23)，角速度与旋转矩阵之间存在如下关系：

$${}^o\omega_{OB \times} = {}^o\dot{R}_B {}^oR_B^T \quad (3.17)$$

将公式(3.16)代入公式(3.17)：

$$\begin{aligned} {}^o\omega_{OB \times} &= {}^o\dot{R}_B {}^oR_B^T \\ &= (\dot{R}_z R_y R_x + R_z \dot{R}_y R_x + R_z R_y \dot{R}_x) (R_z R_y R_x)^T \\ &= (\dot{R}_z R_y R_x + R_z \dot{R}_y R_x + R_z R_y \dot{R}_x) R_x^T R_y^T R_z^T \\ &= \dot{R}_z R_y R_x R_x^T R_y^T R_z^T + R_z \dot{R}_y R_x R_x^T R_y^T R_z^T + R_z R_y \dot{R}_x R_x^T R_y^T R_z^T \\ &= \dot{R}_z R_z^T + R_z \dot{R}_y R_y^T R_z^T + R_z R_y \dot{R}_x R_x^T R_y^T R_z^T \\ &= (\mathbf{e}_3 \dot{\psi})_{\times} + R_z (\mathbf{e}_2 \dot{\theta})_{\times} R_z^T + R_z R_y (\mathbf{e}_1 \dot{\phi})_{\times} R_y^T R_z^T \end{aligned} \quad (3.18)$$

公式(3.18)中  $R_z$ 、 $R_y$ 、 $R_x$  分别为  $R_z(\psi)$ 、 $R_y(\theta)$ 、 $R_x(\phi)$  的简写。结合公式(2.31)中角速度在不同坐标系下旋转变换的性质，公式(3.18)可继续推演为：

$$\begin{aligned} {}^o\omega_{OB \times} &= (\mathbf{e}_3 \dot{\psi})_{\times} + R_z (\mathbf{e}_2 \dot{\theta})_{\times} R_z^T + R_z R_y (\mathbf{e}_1 \dot{\phi})_{\times} R_y^T R_z^T \\ &= (\mathbf{e}_3 \dot{\psi})_{\times} + (R_z \mathbf{e}_2 \dot{\theta})_{\times} + (R_z R_y \mathbf{e}_1 \dot{\phi})_{\times} \\ &= (\mathbf{e}_3 \dot{\psi} + R_z \mathbf{e}_2 \dot{\theta} + R_z R_y \mathbf{e}_1 \dot{\phi})_{\times} \end{aligned} \quad (3.19)$$

因此，ZYX 欧拉角的微分到角速度的变换关系为：

$$\begin{aligned} {}^o\omega_{OB} &= \mathbf{e}_3 \dot{\psi} + R_z \mathbf{e}_2 \dot{\theta} + R_z R_y \mathbf{e}_1 \dot{\phi} \\ &= [R_z R_y \mathbf{e}_1 \ R_z \mathbf{e}_2 \ \mathbf{e}_3] [\dot{\phi} \ \dot{\theta} \ \dot{\psi}]^T \\ &= \begin{bmatrix} c_{\theta} c_{\psi} & -s_{\psi} & 0 \\ c_{\theta} s_{\psi} & c_{\psi} & 0 \\ -s_{\theta} & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \end{aligned} \quad (3.20)$$

由于：

$$\begin{bmatrix} c_{\psi}/c_{\theta} & s_{\psi}/c_{\theta} & 0 \\ -s_{\psi} & c_{\psi} & 0 \\ c_{\psi} s_{\theta}/c_{\theta} & s_{\psi} s_{\theta}/c_{\theta} & 1 \end{bmatrix} \begin{bmatrix} c_{\theta} c_{\psi} & -s_{\psi} & 0 \\ c_{\theta} s_{\psi} & c_{\psi} & 0 \\ -s_{\theta} & 0 & 1 \end{bmatrix} = \mathbb{I}_3 \quad (3.21)$$

因此角速度到 ZYX 欧拉角微分的映射为：

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} c_{\psi}/c_{\theta} & s_{\psi}/c_{\theta} & 0 \\ -s_{\psi} & c_{\psi} & 0 \\ c_{\psi} s_{\theta}/c_{\theta} & s_{\psi} s_{\theta}/c_{\theta} & 1 \end{bmatrix} {}^o\omega_{OB} \quad (3.22)$$

当横滚角  $\phi$  与俯仰角  $\theta$  很小时，公式(3.22)可近似为：

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} c_\psi & s_\psi & 0 \\ -s_\psi & c_\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} {}^O\omega_{OB} = R_z^T(\psi) \cdot {}^O\omega_{OB} \quad (3.23)$$

尽管这里做了近似处理，但实践<sup>[15, 16]</sup>表明这种近似足以满足控制需求。

### 3.2.6 单刚体动力学模型及其离散化

将公式(3.23)、(3.15)、(3.2)、(3.1)整合到一个方程下：

$$\frac{d}{dt} \begin{bmatrix} \Theta \\ {}^O\mathbf{p}_{com} \\ {}^O\omega_{OB} \\ {}^O\dot{\mathbf{p}}_{com} \end{bmatrix} = \begin{bmatrix} \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & R_z^T(\psi) & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbb{I}_3 \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \end{bmatrix} \begin{bmatrix} \Theta \\ {}^O\mathbf{p}_{com} \\ {}^O\omega_{OB} \\ {}^O\dot{\mathbf{p}}_{com} \end{bmatrix} + \begin{bmatrix} \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ {}^P I^{-1} \mathbf{r}_{1 \times} & {}^P I^{-1} \mathbf{r}_{2 \times} & {}^P I^{-1} \mathbf{r}_{3 \times} & {}^P I^{-1} \mathbf{r}_{4 \times} \\ \mathbb{I}_3/m & \mathbb{I}_3/m & \mathbb{I}_3/m & \mathbb{I}_3/m \end{bmatrix} \begin{bmatrix} {}^O\mathbf{f}_1 \\ {}^O\mathbf{f}_2 \\ {}^O\mathbf{f}_3 \\ {}^O\mathbf{f}_4 \end{bmatrix} + \begin{bmatrix} \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{3 \times 1} \end{bmatrix} \quad (3.24)$$

公式(3.24)中 ${}^P I^{-1}$ 为 ${}^P I$ 的逆矩阵。由于公式(3.24)不能直接作为标准的状态方程，将该公式稍做改动，将第三项重力加速度项整合到状态中：

$$\underbrace{\frac{d}{dt} \begin{bmatrix} \Theta \\ {}^O\mathbf{p}_{com} \\ {}^O\omega_{OB} \\ {}^O\dot{\mathbf{p}}_{com} \\ {}^O\mathbf{g}(3) \end{bmatrix}}_{\mathbf{x}} = \underbrace{\begin{bmatrix} \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & R_z^T(\psi) & \mathbf{0}_{3 \times 3} & 0 \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbb{I}_3 & \vdots \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & 0 \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & 1 \\ 0 & \dots & 0 & \dots & 0 \end{bmatrix}}_{A_c} \underbrace{\begin{bmatrix} \Theta \\ {}^O\mathbf{p}_{com} \\ {}^O\omega_{OB} \\ {}^O\dot{\mathbf{p}}_{com} \\ {}^O\mathbf{g}(3) \end{bmatrix}}_{\mathbf{x}} + \underbrace{\begin{bmatrix} \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ {}^P I^{-1} \mathbf{r}_{1 \times} & {}^P I^{-1} \mathbf{r}_{2 \times} & {}^P I^{-1} \mathbf{r}_{3 \times} & {}^P I^{-1} \mathbf{r}_{4 \times} \\ \mathbb{I}_3/m & \mathbb{I}_3/m & \mathbb{I}_3/m & \mathbb{I}_3/m \\ \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} \end{bmatrix}}_{B_c} \underbrace{\begin{bmatrix} {}^O\mathbf{f}_1 \\ {}^O\mathbf{f}_2 \\ {}^O\mathbf{f}_3 \\ {}^O\mathbf{f}_4 \end{bmatrix}}_{\mathbf{u}} \quad (3.25)$$

以紧凑形式重写公式(3.25)，可得单刚体系统的连续状态方程：

$$\dot{\mathbf{x}}(t) = A_c(\psi) \mathbf{x}(t) + B_c(\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3, \mathbf{r}_4, \psi) \mathbf{u}(t) \quad (3.26)$$

用 $\Delta T$ 来表示模型预测控制的周期，在一个 $\Delta T$ 中，可以近似认为系统状态的微分 $\dot{\mathbf{x}}$ 未发生变化：

$$\dot{\mathbf{x}} = \frac{\mathbf{x}(k+1) - \mathbf{x}(k)}{\Delta T} = A\mathbf{x}(k) + B\mathbf{u}(k) \quad (3.27)$$

因此可以采用如下方法将(3.26)离散化：

$$\mathbf{x}(k+1) = (\mathbb{I} + \Delta T \cdot A) \mathbf{x}(k) + \Delta T \cdot B\mathbf{u}(k) = A_k \mathbf{x}(k) + B_k \mathbf{u}(k) \quad (3.28)$$

其中， $A_k$ 、 $B_k$ 的值分别为：

$$A_k = \begin{bmatrix} \mathbb{I}_3 & \mathbf{0}_{3 \times 3} & R_z^T({}_d\psi^k) \Delta T & \mathbf{0}_{3 \times 3} & 0 \\ \mathbf{0}_{3 \times 3} & \mathbb{I}_3 & \mathbf{0}_{3 \times 3} & \mathbb{I}_3 \Delta T & \vdots \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbb{I}_3 & \mathbf{0}_{3 \times 3} & 0 \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbb{I}_3 & \Delta T \\ 0 & \dots & 0 & \dots & 1 \end{bmatrix}, B_k = \begin{bmatrix} \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ {}^P I_k^{-1} \cdot \mathbf{r}_{1 \times}^k \Delta T & {}^P I_k^{-1} \cdot \mathbf{r}_{2 \times}^k \Delta T & {}^P I_k^{-1} \cdot \mathbf{r}_{3 \times}^k \Delta T & {}^P I_k^{-1} \cdot \mathbf{r}_{4 \times}^k \Delta T \\ \mathbb{I}_3 \Delta T / m & \mathbb{I}_3 \Delta T / m & \mathbb{I}_3 \Delta T / m & \mathbb{I}_3 \Delta T / m \\ \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} & \mathbf{0}_{1 \times 3} \end{bmatrix} \quad (3.29)$$

在公式(3.29)中,  $A_k$ 是一个 13 行 13 列的矩阵,  $B_k$ 是一个 13 行 12 列的矩阵。 ${}_d\psi^k$ 为质心轨迹规划器生成的期望偏航角; 若忽略俯仰角与横滚角, 将第  $k$  个控制周期的旋转矩阵 ${}^oR_B^k$ 近似为 ${}^oR_B^k = R_z({}_d\psi^k)$ , 代入公式(3.6)可求第  $k$  个控制周期的惯性张量 ${}^pI^k$ ;  $r_{ix}^k$ 为第  $k$  个控制周期第  $i$  条腿的名义落足点, 若第  $i$  条腿在当前时刻是支撑腿, 则 $r_{ix}^k$ 为当前的足底坐标, 若第  $i$  条腿在当前时刻是摆动腿, 则 $r_{ix}^k$ 为公式(2.85)和公式(2.86)求解的落足点, 即:

$$r_{ix}^k = \begin{cases} {}^o_i p^t & , \quad {}_i S_\phi^t = 1 \\ {}^o_i p_{sw \cdot end}^t & , \quad {}_i S_\phi^t = 0 \end{cases} \quad (3.30)$$

这里假设了在模型预测控制的预测时域中, 每条腿作为支撑腿的时间范围内, 足底坐标不能变化。因此, 预测时域的长度不能超过半个迈步周期。否则, 若某条腿当前时刻是支撑腿, 即将切换为摆动腿, 而在预测时域长度超过半个迈步周期的情况下, 在预测时域内该腿有可能从支撑腿切换为摆动腿再切换为支撑腿, 第二次成为支撑腿时的足底坐标, 依然用第一次支撑腿的足底坐标, 就会产生较大的预测误差。

### 3.3 预测方程

在上一小节中推导出了单刚体系统离散化状态方程(3.28)。在该方程中,  $A_k$ 与 $B_k$ 仅和期望轨迹、落足点坐标以及当前系统状态有关, 是已知量; 假设第  $k$  个控制周期的控制量 $\mathbf{u}(k)$ 已知, 就可以由 $\mathbf{x}(k)$ 递推出 $\mathbf{x}(k+1)$ ; 假设 $\mathbf{u}(k+1)$ 已知, 也可以由 $\mathbf{x}(k+1)$ 递推出 $\mathbf{x}(k+2)$ 。

可以利用单刚体系统离散化状态方程(3.28)递推出未来  $h$  个控制周期的状态:

$$\begin{aligned} \mathbf{x}(1) &= A_0 \mathbf{x}(0) + B_0 \mathbf{u}(0) \\ \mathbf{x}(2) &= A_1 A_0 \mathbf{x}(0) + A_1 B_0 \mathbf{u}(0) + B_1 \mathbf{u}(1) \\ \mathbf{x}(3) &= A_2 A_1 A_0 \mathbf{x}(0) + A_2 A_1 B_0 \mathbf{u}(0) + A_2 B_1 \mathbf{u}(1) + B_2 \mathbf{u}(2) \\ \mathbf{x}(4) &= A_3 A_2 A_1 A_0 \mathbf{x}(0) + A_3 A_2 A_1 B_0 \mathbf{u}(0) + A_3 A_2 B_1 \mathbf{u}(1) + A_3 B_2 \mathbf{u}(2) + B_3 \mathbf{u}(3) \\ &\vdots \\ \mathbf{x}(h) &= \left( \prod_{i=h-1}^0 A_i \right) \mathbf{x}(0) + \sum_{i=0}^{h-2} \left[ \left( \prod_{j=h-1}^{i+1} A_j \right) B_i \mathbf{u}(i) \right] + B_{h-1} \mathbf{u}(h-1) \end{aligned} \quad (3.31)$$

将未来  $h$  个控制周期的控制量组合成一个向量:

$$U = [\mathbf{u}^T(0) \quad \mathbf{u}^T(1) \quad \cdots \quad \mathbf{u}^T(h-1)]^T \quad (3.32)$$

将公式(3.31)的递推过程整合成更紧凑的矩阵形式:

$$X = A_{qp} \mathbf{x}(0) + B_{qp} U \quad (3.33)$$

公式(3.33)称为模型预测控制的预测方程。 $\mathbf{x}(0)$ 是当前时刻的状态, 其数据来自状态

估计器； $U$  是预测时域  $h$  内的控制量，大小为  $12h$  行 1 列； $X$  是系统未来  $h$  个控制周期的状态，大小为  $13h$  行 1 列； $A_{qp}$  是状态系数矩阵，大小为  $13h$  行  $13$  列； $B_{qp}$  是控制系数矩阵，大小为  $13h$  行  $12h$  列。 $X$ 、 $A_{qp}$ 、 $B_{qp}$  的值分别为：

$$X = \begin{bmatrix} \mathbf{x}(1) \\ \mathbf{x}(2) \\ \vdots \\ \mathbf{x}(h) \end{bmatrix}, A_{qp} = \begin{bmatrix} A_0 \\ A_1 A_0 \\ A_2 A_1 A_0 \\ \vdots \\ \prod_{i=h-1}^0 A_i \end{bmatrix}, B_{qp} = \begin{bmatrix} B_0 & \mathbf{0} & \cdots & \mathbf{0} \\ A_1 B_0 & B_1 & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \mathbf{0} \\ (\prod_{j=h-1}^1 A_j) B_0 & (\prod_{j=h-1}^2 A_j) B_1 & \cdots & B_{h-1} \end{bmatrix} \quad (3.34)$$

### 3.4 带约束的二次凸优化

在上一章公式(2.77)中，已经得到了未来  $h$  个控制周期的期望轨迹  $D$ ，而公式(3.33)中系统未来的状态  $X$  仅取决于系统未来的控制量  $U$ 。因此，现在希望找到一个  $U$ ，使得  $D$  与  $X$  的差尽可能为  $\mathbf{0}$ ，那么  $U$  中的第一组控制量  $\mathbf{u}(0)$  就可以作为当前控制周期的控制量。结合公式(3.9)与(3.13)，可以构建如下优化问题：

$$\begin{aligned} \min_U J(U) &= (X - D)^T Q (X - D) + U^T R U \\ \text{s.t. } \circ \mathbf{f}_i &= \mathbf{0}_{3 \times 1}, \quad \forall_i s_{\phi} = 0 \\ \underline{c}_i &\leq C_i \cdot \circ \mathbf{f}_i \leq \bar{c}_i \end{aligned} \quad (3.35)$$

其中  $J(U)$  为优化目标函数，该优化问题含义为：在存在约束的情况下，求  $U$ ，使得  $J(U)$  最小。 $J(U)$  由两项组成，第一项  $(X - D)^T Q (X - D)$  是为了让  $X$  与  $D$  的差尽可能为  $\mathbf{0}$ ；其中  $Q$  为反馈权重矩阵，是一个  $13h$  行  $13h$  列的对角矩阵，对角线上第  $i$  个元素  $Q(i, i)$  对应了  $X(i)$  与  $D(i)$  的权重， $Q(i, i)$  越大， $U$  就会保证  $X(i)$  与  $D(i)$  的差距越小。符号“s.t.”用于标记该优化问题的约束条件。在本文所述四足机器人的工程实现上，不同控制周期的同一个控制闭环权重相等，即：

$$Q(i, i) = Q(i \% 13, i \% 13) \quad (3.36)$$

公式(3.36)中“%”是取余数符号，在工程实现上  $Q$  的前 13 个对角元素为：

$$Q.\text{diagonal}(0, 13) = [25 \ 25 \ 10 \ 1 \ 1 \ 100 \ 0 \ 0 \ 0.3 \ 0.2 \ 0.2 \ 20 \ 0]^T \quad (3.37)$$

$J(U)$  的第二项  $U^T R U$  是为了保证控制量  $U$  不能太大，否则，持续运行在较高水平扭矩下，会导致电机发热、电机精度下降、机构寿命缩短、噪音大等问题。其中  $R$  为控制量权重矩阵，是一个  $12h$  行  $12h$  列的对角矩阵，其值为：

$$R = 0.00005 \cdot \mathbb{I}_{12h \times 12h} \quad (3.38)$$

为了将公式(3.35)转换为二次规划的一般形式，将公式(3.33)代入公式(3.35)：

$$\begin{aligned}
J(U) &= (X - D)^T Q (X - D) + U^T R U \\
&= \left( \underbrace{A_{qp} \mathbf{x}_0 - D}_E + B_{qp} U \right)^T Q (A_{qp} \mathbf{x}_0 - D + B_{qp} U) + U^T R U \\
&= (E + B_{qp} U)^T Q (E + B_{qp} U) + U^T R U \\
&= E^T Q E + (B_{qp} U)^T Q (B_{qp} U) + 2E^T Q (B_{qp} U) + U^T R U \\
&= U^T B_{qp}^T Q B_{qp} U + U^T R U + 2E^T Q B_{qp} U + E^T Q E \\
&= U^T (B_{qp}^T Q B_{qp} + R) U + U^T (2E^T Q B_{qp})^T + \underbrace{E^T Q E}_{const}
\end{aligned} \tag{3.39}$$

令：

$$\begin{cases} H = 2(B_{qp}^T Q B_{qp} + R) \\ g = (2E^T Q B_{qp})^T = 2B_{qp}^T Q (A_{qp} \mathbf{x}_0 - D) \end{cases} \tag{3.40}$$

公式(3.39)中  $E^T Q E$  不是  $U$  的函数，对于“ $U$  取何值时  $J(U)$  最小”这一问题没有影响，因此直接舍去； $\mathbf{x}_0$  是  $\mathbf{x}(0)$  的简写；另外， $Q$  是对角矩阵，因此  $Q = Q^T$ 。则公式(3.39)可继续推演为：

$$J(U) = \frac{1}{2} U^T H U + U^T g \tag{3.41}$$

将公式(3.13)对足底力的约束整合成对  $U$  的约束：

$$\underline{c} \leq C U \leq \bar{c} \tag{3.42}$$

其中  $\underline{c}$ 、 $C$ 、 $\bar{c}$  分别为：

$$\underline{c} = \mathbf{0}_{20h \times 1}, \quad C = \begin{bmatrix} C_i & \cdots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{0} & \cdots & C_i \end{bmatrix}, \quad \bar{c} = \begin{bmatrix} \bar{c}_i \\ \bar{c}_i \\ \vdots \\ \bar{c}_i \end{bmatrix} \tag{3.43}$$

尽管  $U$  为何值时  $J(U)$  最小这一问题尚未解决，但是根据公式(3.9)，已经可以确定  $U$  中摆动腿对应的足底力为  $\mathbf{0}$ ，对于最常用的对角步态，摆动腿与支撑腿分别为 2 个，因此可以确定  $U$  中有一半的元素为  $\mathbf{0}$ ，若这些元素也参与优化求解，显然会降低求解效率。在实际工程实现时，可以采用如下技巧降低运算规模：若第  $k$  个控制周期的第  $i$  条腿为摆动腿，则将  $U$  和  $g$  中第  $n_{ki} \sim n_{ki} + 2$  个元素剔除，将  $H$  中第  $n_{ki} \sim n_{ki} + 2$  行和第  $n_{ki} \sim n_{ki} + 2$  列剔除，将  $C$  中的  $n_{ki} \sim n_{ki} + 2$  列剔除，此时，再将  $C$  中全为  $\mathbf{0}$  的行以及  $\underline{c}$  和  $\bar{c}$  中对应的元素剔除。其中  $n_{ki}$  的值为：

$$n_{ki} = 12(k - 1) + 3(i - 1) + 1 \tag{3.44}$$

本文不再另设一套符号来表示这些剔除对应元素后的矩阵。至此，公式(3.35)所描述的优化问题，化简为二次规划的一般形式：

$$\begin{aligned} \min_U J(U) &= \frac{1}{2} U^T H U + U^T g \\ \text{s.t. } \underline{c} &\leq C U \leq \bar{c} \end{aligned} \quad (3.45)$$

该问题可用通用的二次规划求解库计算，在工程实现上，本文使用了基于 C++ 语言的 qpOASES 库。求解出来的  $U$  中的第一组控制量  $\mathbf{u}(0)$  就可以作为当前控制周期的控制量。由于使用了剔除元素的方式来降低运算规模， $\mathbf{u}(0)$  的大小不再是 12 行 1 列，而是  $3n_{st}$  行 1 列， $n_{st}$  为此时支撑腿的数量。在后文中，使用更直观的字符  ${}^o f^{MPC}$  来描述 MPC 求解的控制量：

$${}^o f^{MPC} = \mathbf{u}(0) \quad (3.46)$$

### 3.5 本章小节

本章主要介绍了四足机器人模型预测控制的数学原理。首先分析了模型预测控制在间歇欠驱系统中的必要性；然后将四足机器人简化为一个单刚体模型，并分析其受力情况，并列出了其状态方程；随后根据状态方程搭建了预测方程，利用预测方程将控制问题转化为优化问题，最终求出了足底反力。

## 第四章 仿生四足机器人全身控制（WBC）

一个固定基座串联机器人总是由若干个关节和连杆组成，若关节数目为 $N_j$ ，将这 $N_j$ 个关节参数按照一定顺序组成一个 $N_j \times 1$ 的向量 $\mathbf{q}$ ，则称 $\mathbf{q}$ 为关节空间向量， $\mathbf{q} \in R^{N_j}$ 。人们创造机器人，是希望机器人能够完成一定的任务，比如搬箱子、拧螺栓、分拣货物等等。总的来说，控制系统至少有一个轨迹生成器，用来生成一个或多个期望位置点 $\mathbf{x}_d$ ，比如机器人捡地上一个矿泉水瓶，就需要以矿泉水瓶的位置作为期望位置。为了实现闭环控制，控制系统中至少有一个状态估计器，用来获取机器人当前的位置 $\mathbf{x}$ 。称期望位置 $\mathbf{x}_d$ 和当前位置 $\mathbf{x}$ 为工作空间向量，若该位置需要 $M$ 个维度来描述，则 $\mathbf{x} \in R^M$ 。工作空间向量与关节空间向量存在映射关系：

$$\mathbf{x} = f(\mathbf{q}) \quad (4.1)$$

对公式(4.1)微分可得工作空间速度与关节空间速度之间的映射关系：

$$\dot{\mathbf{x}} = J(\mathbf{q})\dot{\mathbf{q}} \quad (4.2)$$

其中 $J(\mathbf{q})$ 是 $M$ 行 $N_j$ 列的矩阵，称为雅可比矩阵，是关节空间 $\mathbf{q}$ 的函数。

雅可比矩阵 $J(\mathbf{q})$ 是关节空间速度到工作空间速度的映射。图34展示了关节空间速度与工作空间速度的映射关系。在某一机器人可达构型 $\mathbf{q}$ 上，对于一个明确的关节空间速度 $\dot{\mathbf{q}}$ ，必然存在一个明确的工作空间速度 $\dot{\mathbf{x}}$ 。但反过来却不一样，在一些特殊情况下，无法找到能满足工作空间速度 $\dot{\mathbf{x}}$ 的关节空间速度 $\dot{\mathbf{q}}$ 。定义 $R(J)$ 为机器人在雅可比矩阵为 $J$ 时所能达到的工作空间速度的集合，它是 $R^M$ 的一个子集。

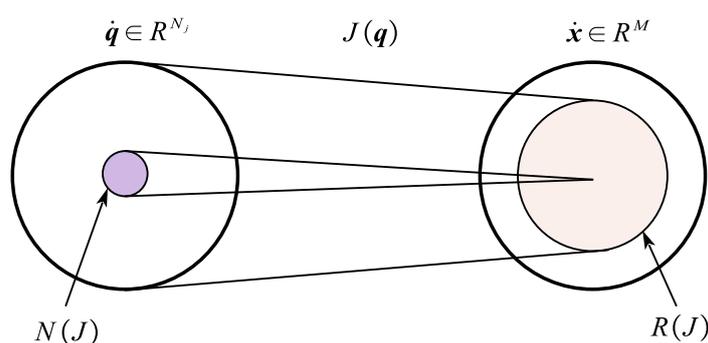


图34 关节空间与工作空间的映射关系

在另外一些情况下，工作空间维度较少而关节空间维度较大，对于某些工作空间速度 $\dot{\mathbf{x}}$ ，可以算出多个甚至无数个 $\dot{\mathbf{q}}$ ，称这种机器人称为冗余机器人。图35(a)展示了一个平面四自由度机器人，其中 $q_1$ 、 $q_2$ 、 $q_3$ 为转动关节， $q_4$ 为直线关节，第一第二连杆长度都为 $l$ 。当保证 $q_1 = q_3$ 时，若 $q_2$ 与 $q_4$ 存在如下关系，则末端执行器速度为0：

$$\sin(\pi - \dot{q}_2) = -\frac{\dot{q}_4}{l^2} \quad (4.3)$$

为描述这种冗余特性，定义  $N(J)$  为雅可比矩阵  $J$  的零空间投影矩阵，简称零空间矩阵。 $N(J)$  与  $J$  满足如下特性：

$$\dot{x} = J(q) \dot{q} \Rightarrow \dot{q} = J^+ \dot{x} + N \dot{q}_v \quad (4.4)$$

$$J \cdot N = 0$$

为保证紧凑性，公式(4.4)中省略描述了零空间矩阵对雅可比矩阵的依赖性，即  $N = N(J)$ 。定义  $J^+$  为  $J$  的右逆矩阵，满足  $JJ^+ = \mathbb{I}$ ，由于：

$$0 = J - \mathbb{I}J = J - JJ^+J = J(\mathbb{I} - J^+J) \quad (4.5)$$

因此零空间矩阵通常按如下公式求解：

$$N = \mathbb{I} - J^+J \quad (4.6)$$

右逆矩阵按照如下方法求解：

$$J^+ = J^T(JJ^T)^{-1} \quad (4.7)$$

通过零空间矩阵，可求工作空间速度到关节空间速度的映射关系，即(4.2)的逆变换：

$$\dot{q} = J^+ \dot{x} + N \dot{q}_v \quad (4.8)$$

其中  $\dot{q}_v$  为任意满足  $\dot{q}_v \in R^{N_j}$  的关节空间速度向量。对公式(4.8)等号两边同时乘以  $J$  可以验证该公式的正确性：

$$\begin{aligned} J\dot{q} &= JJ^+ \dot{x} + JN \dot{q}_v \\ &= \mathbb{I} \dot{x} + 0 \dot{q}_v \\ &= \dot{x} \end{aligned} \quad (4.9)$$

至此，在尽可能保证工作空间任务  $\dot{x}$  完成的前提下，关节可以按照  $\dot{q}_v$  的速度任意运动，来实现其他额外任务，如图35(b)所示关节避障任务。

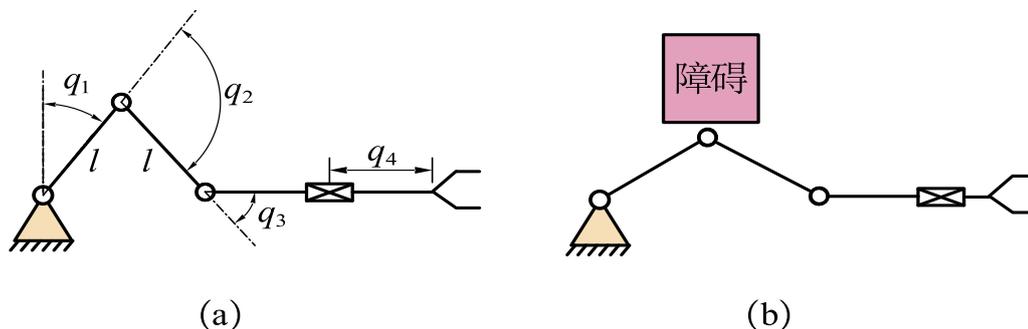


图35 四自由度平面机器人

像这种充分利用关节冗余特性，同时保证多个控制任务的控制方法称为全身控制 (WBC)。

## 4.1 带优先级全身控制基本原理

在设计机器人控制系统时，有时会发现末端执行器位置要比方向更重要，例如在焊接、激光切割等工作中，必须尽可能保证末端执行器位置精度；有时也会发现末端执行器方向要比位置更重要，比如，机械臂末端是一台天文望远镜，当观测非常远的天文现象时，机械臂位置误差并不重要，但方向上稍有误差，就可能无法对准观测目标。

因此，一个综合的机器人控制任务，可以依据不同的意义分解成不同的子任务，并根据重要程度划分优先级，低优先级任务不应该影响高优先级任务的执行。

### 4.1.1 带优先级的双任务全身控制

设某一机器人有两个工作空间任务  $\dot{\mathbf{x}}_1$  与  $\dot{\mathbf{x}}_2$ ，其雅可比矩阵分别为  $J_1$ 、 $J_2$ ，对应的零空间分别为  $N_1$ 、 $N_2$ 。若任务 1 优先级较高，为尽可能完成任务 1，使用公式(4.8)：

$$\dot{\mathbf{q}} = J_1^+ \dot{\mathbf{x}}_1 + N_1 \dot{\mathbf{q}}_v \quad (4.10)$$

现在尝试改变  $\dot{\mathbf{q}}_v$ ，让  $\dot{\mathbf{q}}$  也尽可能实现任务 2 的控制，将公式(4.10)的  $\dot{\mathbf{q}}$  代入公式(4.2)：

$$\begin{aligned} \dot{\mathbf{x}}_2 &= J_2 \dot{\mathbf{q}} \\ &= J_2 (J_1^+ \dot{\mathbf{x}}_1 + N_1 \dot{\mathbf{q}}_v) \\ &= J_2 J_1^+ \dot{\mathbf{x}}_1 + J_2 N_1 \dot{\mathbf{q}}_v \end{aligned} \quad (4.11)$$

因此  $\dot{\mathbf{q}}_v$  为：

$$\dot{\mathbf{q}}_v = (J_2 N_1)^+ (\dot{\mathbf{x}}_2 - J_2 J_1^+ \dot{\mathbf{x}}_1) \quad (4.12)$$

将  $\dot{\mathbf{q}}_v$  代入公式(4.10)可得：

$$\dot{\mathbf{q}} = J_1^+ \dot{\mathbf{x}}_1 + N_1 (J_2 N_1)^+ (\dot{\mathbf{x}}_2 - J_2 J_1^+ \dot{\mathbf{x}}_1) \quad (4.13)$$

在上述过程中，首先尽可能满足了任务 1，然后尽可能满足了任务 2，最终公式(4.13)中关节速度  $\dot{\mathbf{q}}$  可以实现带优先级的双任务控制。

另外，有文献<sup>[28]</sup>证明  $N_1 (J_2 N_1)^+ = (J_2 N_1)^+$ ，因此公式(4.13)可化简为：

$$\dot{\mathbf{q}} = J_1^+ \dot{\mathbf{x}}_1 + (J_2 N_1)^+ (\dot{\mathbf{x}}_2 - J_2 J_1^+ \dot{\mathbf{x}}_1) \quad (4.14)$$

### 4.1.2 带优先级的多任务全身控制

若有  $n_i$  个控制任务，将第  $i$  个任务的工作空间位置用  $\mathbf{x}_i$  表示，其雅可比矩阵与零空间矩阵分别为  $J_i$ 、 $N_i$ 。一般  $i$  从 1 开始计数， $i$  越小优先级越高。

设  $\dot{\mathbf{q}}_i$  为按优先级实现前  $i$  个任务的关节空间速度。当考虑第  $i$  个任务时，可以将前  $i-1$  个任务组合成一个大任务  $A_{i-1}$ ，即：

$$\underbrace{\begin{bmatrix} \dot{\mathbf{x}}_1 \\ \dot{\mathbf{x}}_2 \\ \vdots \\ \dot{\mathbf{x}}_{i-1} \end{bmatrix}}_{\dot{\mathbf{x}}_{i-1}^A} = \underbrace{\begin{bmatrix} J_1 \\ J_2 \\ \vdots \\ J_{i-1} \end{bmatrix}}_{J_{i-1}^A} \dot{\mathbf{q}}_{i-1} \quad (4.15)$$

即:

$$\dot{\mathbf{x}}_{i-1}^A = J_{i-1}^A \dot{\mathbf{q}}_{i-1} \quad (4.16)$$

其零空间矩阵为:

$$N_{i-1}^A = \mathbb{I} - (J_{i-1}^A)^+ \cdot J_{i-1}^A \quad (4.17)$$

假设此时  $\dot{\mathbf{q}}_{i-1}$  已知, 参照公式(4.10),  $\dot{\mathbf{q}}_i$  可写为:

$$\dot{\mathbf{q}}_i = \dot{\mathbf{q}}_{i-1} + N_{i-1}^A \dot{\mathbf{q}}_v \quad (4.18)$$

对公式(4.18)左右两边同时左乘  $J_{i-1}^A$  可以验证  $\dot{\mathbf{q}}_i$  和  $\dot{\mathbf{q}}_{i-1}$  在任务  $A_{i-1}$  所在的工作空间等效。现在尝试改变  $\dot{\mathbf{q}}_v$ , 让  $\dot{\mathbf{q}}_i$  实现  $\dot{\mathbf{x}}_i$  的控制:

$$\begin{aligned} \dot{\mathbf{x}}_i &= J_i \dot{\mathbf{q}}_i \\ &= J_i (\dot{\mathbf{q}}_{i-1} + N_{i-1}^A \dot{\mathbf{q}}_v) \\ &= J_i \dot{\mathbf{q}}_{i-1} + J_i N_{i-1}^A \dot{\mathbf{q}}_v \end{aligned} \quad (4.19)$$

因此  $\dot{\mathbf{q}}_v$  为:

$$\dot{\mathbf{q}}_v = (J_i N_{i-1}^A)^+ (\dot{\mathbf{x}}_i - J_i \dot{\mathbf{q}}_{i-1}) \quad (4.20)$$

将  $\dot{\mathbf{q}}_v$  代入公式(4.18)可得:

$$\begin{aligned} \dot{\mathbf{q}}_i &= \dot{\mathbf{q}}_{i-1} + N_{i-1}^A (J_i N_{i-1}^A)^+ \cdot (\dot{\mathbf{x}}_i - J_i \dot{\mathbf{q}}_{i-1}) \\ &= \dot{\mathbf{q}}_{i-1} + (J_i N_{i-1}^A)^+ \cdot (\dot{\mathbf{x}}_i - J_i \dot{\mathbf{q}}_{i-1}) \end{aligned} \quad (4.21)$$

公式(4.21)中  $\dot{\mathbf{q}}_i = J_i^+ \cdot \dot{\mathbf{x}}_i$ , 通过该递推公式, 可得到满足所有任务的关节空间速度  $\dot{\mathbf{q}}_m$ 。

### 4.1.3 位置级全身控制

公式(4.21)提供了工作空间速度到关节空间速度的映射, 若第  $i$  个任务有期望位置  $\mathbf{x}_i^d$ , 且当前位置  $\mathbf{x}_i$  已知, 定义工作空间位置跟踪误差:

$$\mathbf{e}_i = \mathbf{x}_i^d - \mathbf{x}_i \quad (4.22)$$

以及关节空间位置跟踪误差:

$$\Delta \mathbf{q}_i = \mathbf{q}_i^d - \mathbf{q}_i \quad (4.23)$$

则  $\mathbf{e}_i/T$  与  $\Delta \mathbf{q}_i/T$  可以近似认为机器人从当前位置调整到期望位置的工作空间速度和关节空间速度, 将其代入公式(4.21)可得:

$$\frac{\Delta \mathbf{q}_i}{T} = \frac{\Delta \mathbf{q}_{i-1}}{T} + (J_i N_{i-1}^A)^+ \cdot \left( \frac{\mathbf{e}_i}{T} - J_i \frac{\Delta \mathbf{q}_{i-1}}{T} \right) \quad (4.24)$$

公式(4.24)两边同时乘 $T$ 可得:

$$\Delta \mathbf{q}_i = \Delta \mathbf{q}_{i-1} + (J_i N_{i-1}^A)^+ \cdot (\mathbf{e}_i - J_i \Delta \mathbf{q}_{i-1}) \quad (4.25)$$

公式(4.25)是工作空间位置到关节空间位置的映射, 通过该公式得到 $\Delta \mathbf{q}_m$ 后, 代入公式(4.23)可得期望关节空间位置 $\mathbf{q}_m^d$ 。

#### 4.1.4 加速度级全身控制

将公式(4.2)等式两边同时微分:

$$\ddot{\mathbf{x}} = \dot{J}\dot{\mathbf{q}} + J\ddot{\mathbf{q}} \quad (4.26)$$

对于多任务系统, 第 $i$ 个任务的关节空间加速度与工作空间加速度关系为:

$$\ddot{\mathbf{x}}_i = \dot{J}_i \dot{\mathbf{q}}_i + J_i \ddot{\mathbf{q}}_i \quad (4.27)$$

注意, 公式(4.27)描述的是不同加速度之间的映射关系,  $\dot{\mathbf{q}}$ 是从传感器读取的结果, 作为已知量参与计算。类似于速度级多任务全身控制的处理, 将前 $i-1$ 个任务组合成一个大任务 $A_{i-1}$ ,  $\ddot{\mathbf{q}}_{i-1}$ 为实现前 $i-1$ 个工作空间加速度任务的关节空间加速度, 若 $\ddot{\mathbf{q}}_{i-1}$ 已知, 则依照(4.10),  $\ddot{\mathbf{q}}_i$ 可写为:

$$\ddot{\mathbf{q}}_i = \ddot{\mathbf{q}}_{i-1} + N_{i-1}^A \ddot{\mathbf{q}}_v \quad (4.28)$$

对公式(4.28)左右两边同时左乘 $J_i^A$ , 再同时加 $\dot{J}_{i-1}\dot{\mathbf{q}}$ , 可以验证 $\ddot{\mathbf{q}}_i$ 和 $\ddot{\mathbf{q}}_{i-1}$ 在任务 $A_{i-1}$ 所在的工作空间上效果相同。现在尝试改变 $\ddot{\mathbf{q}}_v$ , 让 $\ddot{\mathbf{q}}_i$ 实现 $\ddot{\mathbf{x}}_i$ 的控制, 将公式(4.28)代入(4.27):

$$\begin{aligned} \ddot{\mathbf{x}}_i &= \dot{J}_i \dot{\mathbf{q}}_i + J_i \ddot{\mathbf{q}}_i \\ &= \dot{J}_i \dot{\mathbf{q}}_i + J_i (\ddot{\mathbf{q}}_{i-1} + N_{i-1}^A \ddot{\mathbf{q}}_v) \\ &= \dot{J}_i \dot{\mathbf{q}}_i + J_i \ddot{\mathbf{q}}_{i-1} + J_i N_{i-1}^A \ddot{\mathbf{q}}_v \end{aligned} \quad (4.29)$$

因此 $\ddot{\mathbf{q}}_v$ 为:

$$\ddot{\mathbf{q}}_v = (J_i N_{i-1}^A)^+ \cdot (\ddot{\mathbf{x}}_i - \dot{J}_i \dot{\mathbf{q}}_i - J_i \ddot{\mathbf{q}}_{i-1}) \quad (4.30)$$

再将公式(4.30)代入(4.28)可得:

$$\begin{aligned} \ddot{\mathbf{q}}_i &= \ddot{\mathbf{q}}_{i-1} + N_{i-1}^A \ddot{\mathbf{q}}_v \\ &= \ddot{\mathbf{q}}_{i-1} + N_{i-1}^A (J_i N_{i-1}^A)^+ \cdot (\ddot{\mathbf{x}}_i - \dot{J}_i \dot{\mathbf{q}}_i - J_i \ddot{\mathbf{q}}_{i-1}) \\ &= \ddot{\mathbf{q}}_{i-1} + (J_i N_{i-1}^A)^+ \cdot (\ddot{\mathbf{x}}_i - \dot{J}_i \dot{\mathbf{q}}_i - J_i \ddot{\mathbf{q}}_{i-1}) \end{aligned} \quad (4.31)$$

公式(4.31)中 $\dot{\mathbf{q}}_i = J_i^+ \cdot (\dot{\mathbf{x}}_i - \dot{J}_i \dot{\mathbf{q}}_i)$ , 通过该递推公式, 可最终得到满足所有任务的关节空间加速度 $\ddot{\mathbf{q}}_m$ 。

## 4.2 全身控制在四足机器人控制系统中的应用

### 4.2.1 浮动基座机器人简介

在前文中已经充分讨论了位置级、速度级、加速度级多任务全身控制在固定基座串联机器人上的递推公式，然而这些结论无法直接应用到四足机器人上。尽管四足机器人可以等效为连接于躯干上的 4 个 3 自由度机械臂，但是躯干本身在世界系下是可移动的，因此仅通过 12 个关节参数无法直接求解组成机器人的刚体在世界系下的位姿，更不可能对其实现控制。如图36所示，从基座可移动性上来分，机器人可分为固定基座机器人与浮动基座机器人，四足机器人是一种典型的浮动基座机器人。



图36 固定基座机器人（左），浮动基座机器人（右）

为解决浮动基座机器人无法使用固定基座机器人算法的问题，如图37所示，在世界系与浮动基座机器人本体系之间建立一个没有实体的 6 自由度关节<sup>[31]</sup>。算上其他 12 个实体关节，四足机器人总共包含 13 个关节，其关节空间向量为：

$$\mathbf{q} = \begin{bmatrix} \mathbf{q}_b \\ \mathbf{q}_j \end{bmatrix} \quad (4.32)$$

其中， $\mathbf{q}_b$  是浮动基体关节， $\mathbf{q}_b = [\Theta^T \quad {}^0\mathbf{p}_{com}^T]^T$ ， $\mathbf{q}_j$  为 12 个实体关节。

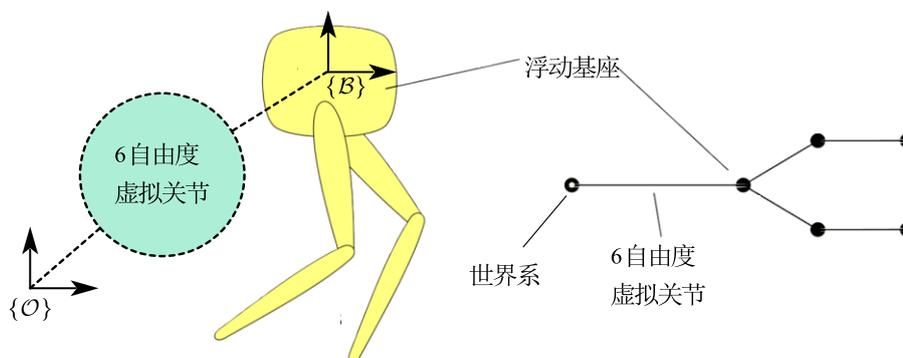


图37 在浮动基座与世界系之间加一个 6 自由度虚拟关节

由于 $q$ 中含有欧拉角项,不适合直接微分处理,因此定义广义速度与广义加速度为:

$$\dot{q} = \begin{bmatrix} {}^B \omega_{OB} \\ {}^B \dot{p}_{com} \\ \dot{q}_j \end{bmatrix}, \quad \ddot{q} = \begin{bmatrix} {}^B \dot{\omega}_{OB} \\ {}^B \ddot{p}_{com} \\ \ddot{q}_j \end{bmatrix} \quad (4.33)$$

注意,公式(4.33)的 $\dot{q}$ 与 $\ddot{q}$ 并不是 $q$ 的一阶微分和二阶微分,而是重新定义的两个符号。

浮动基座机器人在关节空间的动力学方程通常具有以下标准的形式<sup>[31]</sup>:

$$M(q)\ddot{q} + C(q, \dot{q}) = S_j \tau + J_c^{T^O} f_c \quad (4.34)$$

其中 $q$ 为广义关节空间矢量, $q$ 中前6个参数描述了世界系与本体系之间的虚拟6自由度关节, $q$ 的其他参数是其余实体关节的角度(转动关节)或者长度(移动关节)。 $M(q)$ 是关节空间质量矩阵。 $C(q, \dot{q})$ 是偏置力矩阵,包含了重力、科氏力等与关节加速度无关的力。 $J_c$ 和 $f_c$ 分别为接触雅可比和接触力。 $\tau$ 是广义关节扭矩,由于虚拟6自由度关节无法产生扭矩,因此采用选择矩阵 $S_j$ 来过滤掉 $\tau$ 中前六个元素中可能的非零项:

$$S_j = \begin{bmatrix} \mathbf{0}_{6 \times 6} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \quad (4.35)$$

总的来说,公式(4.34)右边是系统受到的力,包含外部环境提供的接触力和系统内部的关节力;左边是系统当前状态以及在这些力的作用下实现的运动。

#### 4.2.2 子任务划分及其具体实现

在本文所述四足机器人全身控制中,将整体控制任务分为四个小任务,按照优先级从高到底分别为: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。由于支撑腿与地面稳定可靠接触是使用其他控制算法的前提条件,因此支撑腿轨迹跟随任务有最高的优先级。在行走过程中,机身保持平稳是一个重要的控制目标,因此机身转动控制与平动控制分别排在第2、第3优先级。机器人与外界接触主要依赖于支撑腿,因此摆动腿的轨迹跟随控制并不太过重要,放在最低优先级。

四足机器人递推全身控制伪代码如公式(4.36)所示。其中 $K_p^i$ 与 $K_d^i$ 为对应控制任务的位置与速度反馈增益,所有控制任务的工作空间都是在世界系下描述。 $J_1$ 是支撑腿轨迹跟随任务的雅可比,由于支撑腿本身就是接触腿,因此 $J_1$ 与公式(4.34)中的 $J_c$ 有相同的含义,即 $J_1 = J_c$ 。通过该算法,可以求得关节空间期望位置 $q^{cmd}$ 、期望速度 $\dot{q}^{cmd}$ 、期望加速度 $\ddot{q}_i^{cmd}$ 。

在第二章中,无论是质心轨迹规划还是足底轨迹规划,都未涉及对加速度的规划,而支撑腿控制是保证足底在世界系下不动,更不会有加速度的规划,因此 $\ddot{x}_c^d = 0$ 。

$$\begin{aligned}
 \Delta \mathbf{q}_1^{cmd} &= \mathbf{0}, \quad \dot{\mathbf{q}}_1^{cmd} = \mathbf{0} \\
 \ddot{\mathbf{q}}_1^{cmd} &= J_1^+ (-J_1 \dot{\mathbf{q}}) \\
 \text{for } i &= 2 \text{ to } 4 \text{ do} \\
 \mathbf{e}_i &= \mathbf{x}_i^d - \mathbf{x}_i \\
 \ddot{\mathbf{x}}_i^{cmd} &= \ddot{\mathbf{x}}_i^d + K_p^i (\mathbf{x}_i^d - \mathbf{x}_i) + K_d^i (\dot{\mathbf{x}}_i^d - \dot{\mathbf{x}}_i) \\
 J_{i-1}^A &= [J_1^T \cdots J_{i-1}^T]^T \\
 N_{i-1}^A &= \mathbb{I} - (J_{i-1}^A)^+ \cdot J_{i-1}^A \\
 \Delta \mathbf{q}_i^{cmd} &= \Delta \mathbf{q}_{i-1}^{cmd} + (J_i N_{i-1}^A)^+ \cdot (\mathbf{e}_i - J_i \Delta \mathbf{q}_{i-1}^{cmd}) \\
 \dot{\mathbf{q}}_i^{cmd} &= \dot{\mathbf{q}}_{i-1}^{cmd} + (J_i N_{i-1}^A)^+ \cdot (\dot{\mathbf{x}}_i^d - J_i \dot{\mathbf{q}}_{i-1}^{cmd}) \\
 \ddot{\mathbf{q}}_i^{cmd} &= \ddot{\mathbf{q}}_{i-1}^{cmd} + (J_i N_{i-1}^A)^+ \cdot (\ddot{\mathbf{x}}_i^{cmd} - \dot{J}_i \dot{\mathbf{q}} - J_i \ddot{\mathbf{q}}_{i-1}^{cmd}) \\
 \text{end} \\
 \mathbf{q}^{cmd} &= \mathbf{q} + \Delta \mathbf{q}_4^{cmd} \\
 \dot{\mathbf{q}}^{cmd} &= \dot{\mathbf{q}}_4^{cmd} \\
 \ddot{\mathbf{q}}^{cmd} &= \ddot{\mathbf{q}}_4^{cmd}
 \end{aligned} \tag{4.36}$$

在考虑任务 2，也就是机身转动控制任务时，不可以简单使用  $\mathbf{e}_2 = {}_d\Theta - \Theta$  来求跟随误差。在之前关于角速度的讨论中已经阐明了欧拉角的基是随着欧拉角变化而变化的，这里需要将欧拉角的误差变换到与世界系下角速度相同的基下来描述，对公式(3.20)在  $\Theta$  附近线性化近似，有：

$$\frac{\mathbf{e}_2}{\Delta T} = \begin{bmatrix} c_\theta c_\psi & -s_\psi & 0 \\ c_\theta s_\psi & c_\psi & 0 \\ -s_\theta & 0 & 1 \end{bmatrix} \frac{\Delta \Theta}{\Delta T} \tag{4.37}$$

从而有：

$$\mathbf{e}_2 = \begin{bmatrix} c_\theta c_\psi & -s_\psi & 0 \\ c_\theta s_\psi & c_\psi & 0 \\ -s_\theta & 0 & 1 \end{bmatrix} ({}_d\Theta - \Theta) \tag{4.38}$$

对于任务 1，由于  $\Delta \mathbf{q}_i^{cmd}$ 、 $\dot{\mathbf{q}}_i^{cmd}$ 、 $\ddot{\mathbf{q}}_i^{cmd}$  都已经直接指定，因此  $\mathbf{x}_i$ 、 $\mathbf{x}_i^d$ 、 $\dot{\mathbf{x}}_i$ 、 $\dot{\mathbf{x}}_i^d$  无需特别指定，任务 1 可以理解为纯粹的运动学逆解，不存在 PD 控制器。或者也可以理解为  $\mathbf{e}_i$ 、 $\dot{\mathbf{x}}_i^d$ 、 $\ddot{\mathbf{x}}_i^{cmd}$  三个与期望值相关的参数都为 0。

对于任务 2， $\dot{\mathbf{x}}_2^d$ 、 $\dot{\mathbf{x}}_2$  分别为  ${}^o\omega_{OB}$  和  ${}^o\omega_{OB}$ 。由于工作空间定义在世界系，而  $\dot{\mathbf{q}}$  的角速度项定义在本体系，因此任务 2 的雅可比矩阵为：

$$J_2 = [{}^oR_B \quad \mathbf{0}_{3 \times 15}] \tag{4.39}$$

对于任务 3， $\mathbf{x}_3$ 、 $\dot{\mathbf{x}}_3$  分别为躯干质心在世界系下的位置  ${}^o\mathbf{p}_{com}$  与速度  ${}^o\mathbf{v}_{com}$ ， $\mathbf{x}_3^d$ 、 $\dot{\mathbf{x}}_3^d$  分别为其期望值。由于工作空间定义在世界系，而  $\dot{\mathbf{q}}$  中机身平动速度项定义在本体系，因此任务 3 的雅可比矩阵为：

$$J_3 = [\mathbf{0}_{3 \times 3} \quad {}^0R_B \quad \mathbf{0}_{3 \times 12}] \quad (4.40)$$

对于任务 4,  $\mathbf{x}_i$ 、 $\mathbf{x}_i^d$ 、 $\dot{\mathbf{x}}_i$ 、 $\dot{\mathbf{x}}_i^d$  分别是世界系下各个摆动足的足底位置、期望足底位置、足底速度、期望足底速度依照腿的顺序组成的向量, 这些值都可以从前文的状态估计器和足底轨迹规划器中获取。至此, 算法(4.36)中尚未得到的量有第 1、第 4 任务的雅可比矩阵  $J_1$ 、 $J_4$ , 所有任务的  $J_i \dot{\mathbf{q}}$  项。另外, 公式(4.34)中的  $M$ 、 $C$  也是未知, 这些变量将在下一章逐一求解。

### 4.3 松弛优化

浮动基座机器人动力学方程(4.34)右边是系统受到的力, 包含外部环境提供的接触力和系统内部的关节力; 左边是系统的状态以及在这些力的作用下实现的运动。若只考虑 6 自由度的浮动基体, 即只考虑公式(4.34)的前六行, 就可以分析机器人躯干的受力与运动情况。因此定义浮动基座选择矩阵:

$$S_f = \begin{bmatrix} \mathbb{I}_{6 \times 6} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \quad (4.41)$$

将公式(4.34)左右两边同时乘以  $S_f$  可得:

$$S_f M \dot{\mathbf{q}} + S_f C = S_f J_c^T {}^0 \mathbf{f}_c \quad (4.42)$$

公式(4.42)就是机器人躯干的动力学方程, 其中  $M$ 、 $C$ 、 $J_c$  的具体求解算法将在下一章给出, 这里假设已知。现在, 若已知足底反力  ${}^0 \mathbf{f}_c$ , 就可以求躯干在  ${}^0 \mathbf{f}_c$  作用下产生的广义加速度  $S_f \ddot{\mathbf{q}}$ , 这是正动力学问题; 若已知期望的广义加速度  $S_f \ddot{\mathbf{q}}$ , 也可以求满足  $S_f \ddot{\mathbf{q}}$  的足底反力  ${}^0 \mathbf{f}_c$ , 这是逆动力学问题。

在上一章中, 已经采用模型控制得到了足底反力  ${}^0 \mathbf{f}^{MPC}$ , 而本章又通过全身控制得到了广义加速度  $\ddot{\mathbf{q}}^{cmd}$ , 若将这两个量直接代入公式(4.42), 显然左右两边不会相等。因为  ${}^0 \mathbf{f}^{MPC}$  更加侧重于未来一段时间的控制, 而  $\ddot{\mathbf{q}}^{cmd}$  更加侧重保证不同任务之间的优先级。为了让足底反力与广义加速度满足躯干的动力学方程(4.42), 为  ${}^0 \mathbf{f}^{MPC}$  和  $\ddot{\mathbf{q}}^{cmd}$  分别添加一个松弛变量:

$$\begin{cases} \ddot{\mathbf{q}} = \ddot{\mathbf{q}}^{cmd} + \begin{bmatrix} \delta_{\ddot{\mathbf{q}}} \\ \mathbf{0} \end{bmatrix} \\ {}^0 \mathbf{f}_c = {}^0 \mathbf{f}^{MPC} + \delta_f \end{cases} \quad (4.43)$$

其中  $\delta_{\ddot{\mathbf{q}}}$  与  $\delta_f$  是在模型预测控制与全身控制的推导结果上添加的一个松弛变量, 这两个变量当然越小越好, 因此构造如下带约束优化问题:

$$\begin{aligned}
 \min_{\delta_q, \delta_f} J(\delta_q, \delta_f) &= \delta_q^T Q_1 \delta_q + \delta_f^T Q_2 \delta_f \\
 s.t. \quad S_f M \ddot{\mathbf{q}} + S_f C &= S_f J_c^T \mathbf{f}_c \\
 \ddot{\mathbf{q}} &= \ddot{\mathbf{q}}^{cmd} + \begin{bmatrix} \delta_q \\ \mathbf{0} \end{bmatrix} \\
 {}^o \mathbf{f}_c &= {}^o \mathbf{f}^{MPC} + \delta_f \\
 c_i &\leq C_i \cdot {}^o \mathbf{f}_c^i \leq \bar{c}_i
 \end{aligned} \tag{4.44}$$

其中 ${}^o \mathbf{f}_c^i$ 为第 $i$ 个支撑腿的足底反力,  $Q_1$ 与 $Q_2$ 是权重矩阵, 在此处的含义为控制系统信任全身控制多一些, 还是信任模型预测控制多一些, 在本文中,  $Q_1 = \mathbf{I}$ ,  $Q_2 = 0.005 \cdot \mathbf{I}$ 。

由于(4.44)所描述的优化问题规模较小, 因此采用了基于 C++ 的二次型优化库 QuadProg++, 现在将公式(4.44)变换为 QuadProg++ 所能接受的标准格式, 构造优化变量:

$$\mathbf{x} = \begin{bmatrix} \delta_q \\ \delta_f \end{bmatrix} \tag{4.45}$$

则(4.44)中优化函数表达式可转换为:

$$\begin{aligned}
 J(\delta_q, \delta_f) &= \delta_q^T Q_1 \delta_q + \delta_f^T Q_2 \delta_f \\
 &= \mathbf{x}^T \underbrace{\begin{bmatrix} Q_1 & \mathbf{0} \\ \mathbf{0} & Q_2 \end{bmatrix}}_{0.5G} \mathbf{x} \\
 &= 0.5 \mathbf{x}^T G \mathbf{x}
 \end{aligned} \tag{4.46}$$

将公式(4.43)代入(4.42)可构造等式约束:

$$S_f M \left( \ddot{\mathbf{q}}^{cmd} + \begin{bmatrix} \delta_q \\ \mathbf{0} \end{bmatrix} \right) + S_f C = S_f J_c^T ({}^o \mathbf{f}^{MPC} + \delta_f) \tag{4.47}$$

令 $M_f$ 、 $\ddot{\mathbf{q}}_f^{cmd}$ 、 $C_f$ 、 $J_{cf}^T$ 分别为 $M$ 、 $\ddot{\mathbf{q}}^{cmd}$ 、 $C$ 、 $J_c^T$ 的前六行组成的矩阵, 则公式(4.47)可化简为:

$$M_f (\ddot{\mathbf{q}}_f^{cmd} + \delta_q) + C_f = J_{cf}^T ({}^o \mathbf{f}^{MPC} + \delta_f) \tag{4.48}$$

整理成关于 $\mathbf{x}$ 的等式约束方程:

$$\underbrace{\begin{bmatrix} M_f & -J_{cf}^T \\ C_f & M_f \end{bmatrix}}_{C_e^T} \underbrace{\begin{bmatrix} \delta_q \\ \delta_f \end{bmatrix}}_{\mathbf{x}} - \underbrace{J_{cf}^T \cdot {}^o \mathbf{f}^{MPC} + C_f + M_f \ddot{\mathbf{q}}_f^{cmd}}_{c_e} = \mathbf{0} \tag{4.49}$$

即:

$$C_E^T \mathbf{x} + c_e = \mathbf{0} \tag{4.50}$$

假设此时有 $n_c$ 条支撑腿, 则 ${}^o \mathbf{f}_c$ 是一个 $3n_c$ 行 1 列的列向量:

$${}^o \mathbf{f}_c = \begin{bmatrix} {}^o \mathbf{f}_c^1 \\ \vdots \\ {}^o \mathbf{f}_c^{n_c} \end{bmatrix} \tag{4.51}$$

应满足不等式约束:

$$\underbrace{\begin{bmatrix} \underline{c}_1 \\ \vdots \\ \underline{c}_{n_c} \end{bmatrix}}_{\underline{c}_A} \leq \underbrace{\begin{bmatrix} C_1 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \ddots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & C_{n_c} \end{bmatrix}}_{C_A} \cdot {}^o \mathbf{f}_c \leq \underbrace{\begin{bmatrix} \bar{c}_1 \\ \vdots \\ \bar{c}_{n_c} \end{bmatrix}}_{\bar{c}_A} \quad (4.52)$$

即:

$$\underline{c}_A \leq C_A \cdot {}^o \mathbf{f}_c \leq \bar{c}_A \quad (4.53)$$

又因为  ${}^o \mathbf{f}_c = {}^o \mathbf{f}^{MPC} + \delta_f$ , 代入(4.53)可得:

$$\underline{c}_A \leq C_A ({}^o \mathbf{f}^{MPC} + \delta_f) \leq \bar{c}_A \quad (4.54)$$

不等式约束(4.54)可以拆分成两个关于  $\delta_f$  的不等式约束:

$$\begin{cases} -C_A \delta_f + \bar{c}_A - C_A \cdot {}^o \mathbf{f}^{MPC} \geq \mathbf{0} \\ C_A \delta_f + C_A \cdot {}^o \mathbf{f}^{MPC} - \underline{c}_A \geq \mathbf{0} \end{cases} \quad (4.55)$$

将不等式约束(4.55)整理成关于  $\mathcal{X}$  的不等式约束方程:

$$\underbrace{\begin{bmatrix} \mathbf{0} & -C_A \\ \mathbf{0} & C_A \end{bmatrix}}_{C_f^T} \underbrace{\begin{bmatrix} \delta_{\ddot{q}} \\ \delta_f \end{bmatrix}}_{\mathcal{X}} + \underbrace{\begin{bmatrix} \bar{c}_A - C_A \cdot {}^o \mathbf{f}^{MPC} \\ C_A \cdot {}^o \mathbf{f}^{MPC} - \underline{c}_A \end{bmatrix}}_{c_i} \geq \mathbf{0} \quad (4.56)$$

即:

$$C_f^T \mathcal{X} + c_i \geq \mathbf{0} \quad (4.57)$$

结合(4.57)、(4.50)、(4.46), 原带约束优化问题(4.44)可变换为 QuadProg++所能接受的标准格式:

$$\begin{aligned} \min_{\mathcal{X}} J(\mathcal{X}) &= 0.5 \mathcal{X}^T G \mathcal{X} \\ s.t. \quad C_E^T \mathcal{X} + c_e &= \mathbf{0} \\ C_f^T \mathcal{X} + c_i &\geq \mathbf{0} \end{aligned} \quad (4.58)$$

使用 QuadProg++求解库将  $\delta_{\ddot{q}}$  与  $\delta_f$  求解出后, 代入公式(4.43)求出机身加速度  $\ddot{q}$  与足底反力  ${}^o \mathbf{f}_c$ , 将公式(4.34)稍作变形:

$$M(\mathbf{q}) \ddot{\mathbf{q}} + C(\mathbf{q}, \dot{\mathbf{q}}) = S_j \begin{bmatrix} \boldsymbol{\tau}_f \\ \boldsymbol{\tau}_j \end{bmatrix} + J_c^{T^o} \mathbf{f}_c \quad (4.59)$$

其中  $\boldsymbol{\tau}_f$  为虚拟 6 自由度关节的扭矩,  $\boldsymbol{\tau}_f = \mathbf{0}$ 。因此有:

$$S_j \begin{bmatrix} \boldsymbol{\tau}_f \\ \boldsymbol{\tau}_j \end{bmatrix} = \begin{bmatrix} \boldsymbol{\tau}_f \\ \boldsymbol{\tau}_j \end{bmatrix} \quad (4.60)$$

至此, 公式(4.59)中除了  $\boldsymbol{\tau}_j$  以外的所有变量都已求出, 因此可求  $\boldsymbol{\tau}_j$  为:

$$\begin{bmatrix} \mathbf{0}_{6 \times 1} \\ \boldsymbol{\tau}_j \end{bmatrix} = M(\mathbf{q}) \ddot{\mathbf{q}} + C(\mathbf{q}, \dot{\mathbf{q}}) - J_c^{T^o} \mathbf{f}_c \quad (4.61)$$

从  $\mathbf{q}^{cmd}$ 、 $\dot{\mathbf{q}}^{cmd}$  中剔除前 6 行,剩下的为实体关节位置指令  $\mathbf{q}_j^{cmd}$  与实体关节速度指令  $\dot{\mathbf{q}}_j^{cmd}$ , 最终发送给电机的扭矩指令为:

$$\boldsymbol{\tau}^{cmd} = \boldsymbol{\tau}_j + k_p(\mathbf{q}_j^{cmd} - \mathbf{q}_j) + k_d(\dot{\mathbf{q}}_j^{cmd} - \dot{\mathbf{q}}_j) \quad (4.62)$$

其中  $k_p$ 、 $k_d$  为关节级 PD 控制器的参数。

#### 4.4 本章小节

本章主要介绍了四足机器人全身控制的数学原理。首先利用雅可比矩阵的零空间特性,实现了带优先级的双任务全身控制;然后将双任务系统推广到多任务系统,再将速度级全身控制推广到位置级和加速度级全身控制;然后介绍了浮动基体的标准动力学方程,并在浮动基体和世界系之间添加了六自由度虚拟关节;随后介绍了带优先级的全身控制在四足机器人上的具体实现;最后对模型预测控制和全身控制的结果做松弛优化,得到最终的关节扭矩指令。

## 第五章 多刚体系统动力学算法

在第四章中介绍了全身控制的基本原理与具体实现，然后与第三章中模型预测控制的足底反力结果做松弛优化，最终得到了发送给电机的扭矩指令。但是在该过程中，还有几个矩阵尚未求出，包括公式(4.61)中的 $M$ 矩阵、 $C$ 向量，以及伪代码(4.36)中第1任务、第4任务的雅可比矩阵 $J_1$ 、 $J_4$ ，以及各级任务的 $J_i \dot{q}$ 项。本章将基于 Plücker 坐标系建立一套多刚体动力学算法体系，用于求解这些矩阵。

### 5.1 六维空间向量

#### 5.1.1 运动向量空间与动力向量空间简介

传统的刚体动力学算法将刚体运动分为3维平动与3维转动，试图用两套算法（牛顿第二定律与欧拉方程）分别处理。然而平动与转动并不能简单分离：刚体上的点会因刚体的转动而具有线速度；力施加到刚体上，即可能改变平动的运动状态，也可能改变转动的运动状态。平动与转动之间的耦合关系体现在刚体动力学算法的方方面面，导致传统刚体动力学算法既不简洁，又难以推导与使用。因此，本章将引入一套更加广义的概念来描述刚体的速度、加速度、力、动量等属性，从而使刚体的动力学算法变得简洁易用。

在前文关于角速度的讨论中，论证了对刚体速度的描述本质上是对刚体上的点的速度在空间中形成的速度场的描述。即给定刚体上任意一点的位置矢量，若能求得该点的速度，则说明对该刚体的速度已经完整描述。如图38(a)所示，点 $O$ 为空间中的不动点， $v_o$ 为当前时刻与点 $O$ 重合的刚体上一点的速度， $\omega$ 为刚体绕过 $O$ 点的一直线转动的角速度，根据公式(2.23)，刚体上任意一点 $P$ 的速度为：

$$v_P = v_o + \omega \times \overrightarrow{OP} \quad (5.1)$$

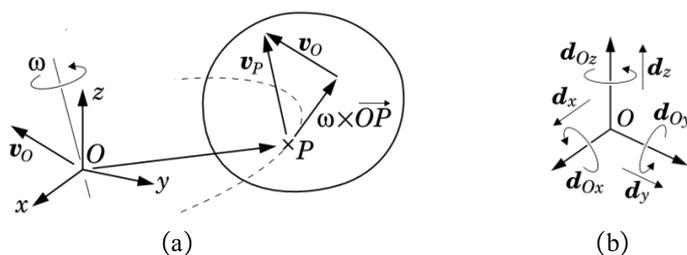


图38 刚体的速度描述(a)，运动向量空间 Plücker 坐标系(b)

因此，仅用角速度 $\omega$ 和平动速度 $v_o$ 就可以完整描述该刚体的速度。注意到 $\omega$ 的旋转轴规定为过 $O$ 的一直线，因此 $v_o$ 中没有因 $\omega$ 而产生的线速度分量，所以可以称 $v_o$ 为纯平

动速度， $\boldsymbol{\omega}$ 为纯转动速度。在 $O$ 点建立笛卡尔坐标系，并在该坐标系中建立一组单位正交基 $\{i, j, k\}$ ， $\boldsymbol{v}_o$ 和 $\boldsymbol{\omega}$ 可以分别用具体参数表示为：

$$\boldsymbol{v}_o = [i \ j \ k] \begin{bmatrix} v_{Ox} \\ v_{Oy} \\ v_{Oz} \end{bmatrix}, \quad \boldsymbol{\omega} = [i \ j \ k] \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \quad (5.2)$$

为了将 $\boldsymbol{v}_o$ 和 $\boldsymbol{\omega}$ 整合到一个向量下，定义一组6维正交基：

$$\mathcal{D}_O = \{d_{Ox}, d_{Oy}, d_{Oz}, d_x, d_y, d_z\} \subseteq M^6 \quad (5.3)$$

如图38(b)所示， $d_{Ox}$ 、 $d_{Oy}$ 、 $d_{Oz}$ 分别为绕 $x$ 、 $y$ 、 $z$ 轴的单位转动， $d_x$ 、 $d_y$ 、 $d_z$ 为沿 $x$ 、 $y$ 、 $z$ 轴的单位平动，则刚体的六维空间速度为：

$$\boldsymbol{v} = \omega_x d_{Ox} + \omega_y d_{Oy} + \omega_z d_{Oz} + v_{Ox} d_x + v_{Oy} d_y + v_{Oz} d_z \quad (5.4)$$

在定义了单位正交基 $\mathcal{D}_O$ 后，就可以将 $\boldsymbol{v}$ 参数化描述：

$$\boldsymbol{v} = [\omega_x \ \omega_y \ \omega_z \ v_{Ox} \ v_{Oy} \ v_{Oz}]^T = [\boldsymbol{\omega}^T \ \boldsymbol{v}_O^T]^T \quad (5.5)$$

这种用6个坐标轴组成的坐标系为Plücker坐标系，一个Plücker坐标系的定义依赖于一个笛卡尔坐标系，在Plücker坐标系中每个向量都需要用6个参数来描述。 $\boldsymbol{v}$ 是定义在 $M^6$ 上的6维空间向量，在本文中用前缀“空间”来描述某个向量是6维的，或者某个张量是6行6列的。 $M^6$ 是运动向量空间，在运动向量空间上可以定义空间速度、空间加速度等与刚体运动相关的空间向量。

如图39(a)所示，参考刚体上任意一点 $P$ ，刚体所受的力可以表示为两部分之和：一个作用在过点 $P$ 的某条直线上的力 $\boldsymbol{f}$ ，以及扭矩 $\boldsymbol{n}_P$ 。若参考另一点 $O$ ，刚体所受的力也可以表示为两部分之和：一个作用在过点 $O$ 的某条直线上的力 $\boldsymbol{f}$ ，以及力矩 $\boldsymbol{n}_O$ ，其中 $\boldsymbol{n}_O$ 与 $\boldsymbol{n}_P$ 的关系为：

$$\boldsymbol{n}_P = \boldsymbol{n}_O + \boldsymbol{f} \times \overrightarrow{OP} \quad (5.6)$$

与运动向量空间 $M^6$ 类似的另一个重要向量空间为动力向量空间 $F^6$ ，在动力向量空间上可以定义空间力、空间动量等与力相关的向量。在 $O$ 点建立笛卡尔坐标系，并在该坐标系中建立一组单位正交基 $\{i, j, k\}$ ， $\boldsymbol{n}_O$ 和 $\boldsymbol{f}$ 可以分别用具体参数表示为：

$$\boldsymbol{n}_O = [i \ j \ k] \begin{bmatrix} n_{Ox} \\ n_{Oy} \\ n_{Oz} \end{bmatrix}, \quad \boldsymbol{f} = [i \ j \ k] \begin{bmatrix} f_x \\ f_y \\ f_z \end{bmatrix} \quad (5.7)$$

与运动向量空间类似，定义一组6维正交基：

$$\mathcal{E}_O = \{e_x, e_y, e_z, e_{Ox}, e_{Oy}, e_{Oz}\} \subseteq F^6 \quad (5.8)$$

如图39(b)所示,  $e_x$ 、 $e_y$ 、 $e_z$ 分别为绕 $x$ 、 $y$ 、 $z$ 轴的单位力矩,  $e_{Ox}$ 、 $e_{Oy}$ 、 $e_{Oz}$ 分别为沿 $x$ 、 $y$ 、 $z$ 轴的单位力, 则刚体所受的六维空间力为:

$$\hat{f} = n_{Ox}e_x + n_{Oy}e_y + n_{Oz}e_z + f_x e_{Ox} + f_y e_{Oy} + f_z e_{Oz} \quad (5.9)$$

在本章中, 若六维空间向量与传统三维向量发生符号冲突, 则在六维空间向量上加“尖帽”。在定义了单位正交基 $\mathcal{E}_O$ 后, 就可以将 $\hat{f}$ 参数化描述:

$$\hat{f} = [n_{Ox} \ n_{Oy} \ n_{Oz} \ f_x \ f_y \ f_z]^T = [n_O^T \ f^T]^T \quad (5.10)$$

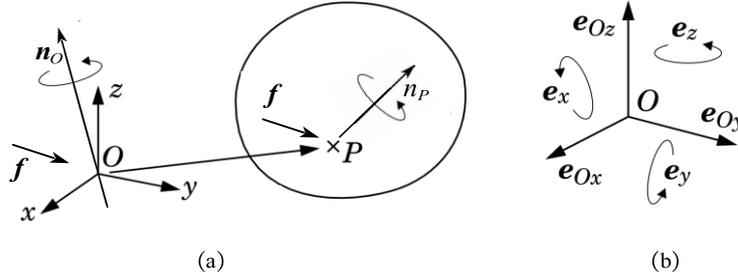


图39 刚体的受力描述(a), 动力向量空间 Plücker 坐标系(b)

尽管在三维的情况下, 力、力矩、速度、加速度这些向量都是定义在同一坐标系下, 且有相同的运算规则, 但是在六维情况下, 运动向量空间与动力向量空间的向量运算规则有些许不同。

### 5.1.2 Plücker 坐标系下的坐标变换

设 $\mathcal{A}$ 、 $\mathcal{B}$ 分别为两个笛卡尔坐标系, 在这两个笛卡尔坐标系上定义了两个 Plücker 坐标系, 也分别称为 $\mathcal{A}$ 和 $\mathcal{B}$ 。设 ${}^A m$ 、 ${}^B m$ 、 ${}^A f$ 、 ${}^B f$ 分别为六维运动向量 $m \in M^6$ 和六维动力向量 $f \in F^6$ 在 $\mathcal{A}$ 、 $\mathcal{B}$ 下的描述。则不同坐标系下空间向量的变换规则为:

$$\begin{cases} {}^B m = {}^B X_{\mathcal{A}} \cdot {}^A m \\ {}^B f = {}^B X_{\mathcal{A}}^* \cdot {}^A f \end{cases} \quad (5.11)$$

其中 ${}^B X_{\mathcal{A}}$ 是六维运动向量从坐标系 $\mathcal{A}$ 到坐标系 $\mathcal{B}$ 的坐标变换矩阵,  ${}^B X_{\mathcal{A}}^*$ 是六维动力向量对应的坐标变换矩阵。定义六维运动向量和六维动力向量之间的数积为:

$$m \cdot f = m^T f \quad (5.12)$$

特别地, 若 $f$ 是空间力且 $m$ 是该空间力作用点的空间速度, 则该数积为空间力 $f$ 传递的功率。为了保证数积的值在不同坐标系下相等, 有:

$$\begin{aligned} {}^A m^T {}^A f &= {}^B m^T {}^B f = ({}^B X_{\mathcal{A}} \cdot {}^A m)^T ({}^B X_{\mathcal{A}}^* \cdot {}^A f) \\ &= {}^A m^T ({}^B X_{\mathcal{A}}^T {}^B X_{\mathcal{A}}^*) {}^A f \end{aligned} \quad (5.13)$$

从而 ${}^B X_{\mathcal{A}}$ 与 ${}^B X_{\mathcal{A}}^*$ 存在如下关系:

$${}^B X_A^* = {}^B X_A^{-T} \quad (5.14)$$

坐标变换矩阵仅依赖于坐标系  $B$  相对于坐标系  $A$  的平移与旋转，因此坐标变换矩阵可以分解为两个简单坐标变换矩阵：平移变换矩阵和旋转变换矩阵。

### (1) 旋转变换

若  $A$  与  $B$  具有相同的坐标原点  $O$ ，六维运动向量  $\hat{m}$  由两个三维向量  $m$  和  $m_o$  组成，设  $E$  为从  $A$  到  $B$  的旋转矩阵，即  $E = {}^B R_A$ ，因此：

$${}^B \hat{m} = \begin{bmatrix} {}^B m \\ {}^B m_o \end{bmatrix} = \begin{bmatrix} E^A m \\ E^A m_o \end{bmatrix} = \begin{bmatrix} E & \mathbf{0} \\ \mathbf{0} & E \end{bmatrix} {}^A \hat{m} \quad (5.15)$$

因此纯旋转变换的  ${}^B X_A$  为：

$${}^B X_A = \begin{bmatrix} E & \mathbf{0} \\ \mathbf{0} & E \end{bmatrix} \quad (5.16)$$

对应的六维动力向量的纯旋转变换为：

$${}^B X_A^* = {}^B X_A^{-T} = \begin{bmatrix} E & \mathbf{0} \\ \mathbf{0} & E \end{bmatrix} \quad (5.17)$$

### (2) 平移变换

若  $A$  与  $B$  的坐标原点分别为  $A$  和  $B$ ，且  $A$  与  $B$  朝向相同。在  $A$  点六维运动向量  $\hat{m}$  由两个三维向量  $m$  和  $m_A$  组成，在  $B$  点六维运动向量  $\hat{m}$  由两个三维向量  $m$  和  $m_B$  组成。则  $m_A$  与  $m_B$  的关系为：

$$m_B = m_A - \overrightarrow{AB} \times m \quad (5.18)$$

${}^A \hat{m}$ 、 ${}^B \hat{m}$  的关系为：

$${}^B \hat{m} = \begin{bmatrix} m \\ m_B \end{bmatrix} = \begin{bmatrix} m \\ m_A - \overrightarrow{AB} \times m \end{bmatrix} = \begin{bmatrix} \mathbb{I} & \mathbf{0} \\ -r_x & \mathbb{I} \end{bmatrix} {}^A \hat{m} \quad (5.19)$$

其中  $r = \overrightarrow{AB}$ ，因此纯平移变换的  ${}^B X_A$  为：

$${}^B X_A = \begin{bmatrix} \mathbb{I} & \mathbf{0} \\ -r_x & \mathbb{I} \end{bmatrix} \quad (5.20)$$

对应的六维动力向量的纯平移变换为：

$${}^B X_A^* = {}^B X_A^{-T} = \begin{bmatrix} \mathbb{I} & -r_x \\ \mathbf{0} & \mathbb{I} \end{bmatrix} \quad (5.21)$$

### (3) 组合变换

如图40所示，分析两坐标系  $A$  与  $B$  之间的组合变换，其中  $r$  为向量  $\overrightarrow{AB}$  在  $A$  坐标系下的描述， $E$  为  $A$  到  $B$  的旋转矩阵  $E = {}^B R_A$ ，空间向量在  $A$  与  $B$  之间的坐标变换为：

$$\begin{aligned}
{}^B X_A &= \begin{bmatrix} E & \mathbf{0} \\ \mathbf{0} & E \end{bmatrix} \begin{bmatrix} \mathbb{I} & \mathbf{0} \\ -r_{\times} & \mathbb{I} \end{bmatrix} = \begin{bmatrix} E & \mathbf{0} \\ -Er_{\times} & E \end{bmatrix} \\
{}^B X_A^* &= \begin{bmatrix} E & \mathbf{0} \\ \mathbf{0} & E \end{bmatrix} \begin{bmatrix} \mathbb{I} & -r_{\times} \\ \mathbf{0} & \mathbb{I} \end{bmatrix} = \begin{bmatrix} E & -Er_{\times} \\ \mathbf{0} & E \end{bmatrix} \\
{}^A X_B &= \begin{bmatrix} \mathbb{I} & \mathbf{0} \\ r_{\times} & \mathbb{I} \end{bmatrix} \begin{bmatrix} E^T & \mathbf{0} \\ \mathbf{0} & E^T \end{bmatrix} = \begin{bmatrix} E^T & \mathbf{0} \\ E^T r_{\times} & E^T \end{bmatrix} \\
{}^A X_B^* &= \begin{bmatrix} \mathbb{I} & r_{\times} \\ \mathbf{0} & \mathbb{I} \end{bmatrix} \begin{bmatrix} E^T & \mathbf{0} \\ \mathbf{0} & E^T \end{bmatrix} = \begin{bmatrix} E^T & E^T r_{\times} \\ \mathbf{0} & E^T \end{bmatrix}
\end{aligned} \tag{5.22}$$

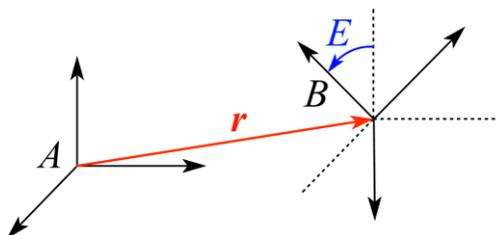


图40 坐标系A与B变关系

### 5.1.3 叉乘与微分

在三维空间下，公式(2.23)表明角速度可以将刚接于刚体上的常数向量微分：

$${}^o \dot{\boldsymbol{r}} = \boldsymbol{\omega} \times {}^o \boldsymbol{r} \tag{5.23}$$

这意味着符号“ $\boldsymbol{\omega} \times$ ”实现了向量 $\boldsymbol{r}$ 到其微分的映射。叉乘的这种性质也可以推广到Plücker坐标系下，但是对于六维运动向量和六维动力向量需要定义不同的叉乘规则。设六维运动向量 $\boldsymbol{m} \in M^6$ 和六维动力向量 $\boldsymbol{f} \in F^6$ 为某坐标系下的常数空间向量，该坐标系在世界系下以空间速度 $\boldsymbol{v}$ 移动，则Plücker坐标系下的叉乘满足以下等式：

$$\begin{cases} {}^o \dot{\boldsymbol{m}} = \boldsymbol{v} \times {}^o \boldsymbol{m} \\ {}^o \dot{\boldsymbol{f}} = \boldsymbol{v} \times {}^* {}^o \boldsymbol{f} \end{cases} \tag{5.24}$$

为了如(2.27)一样明确给出空间向量的叉乘规则，如图41，分析刚接于某坐标系上的空间向量的微分问题。如图41(a)，若某一刚体坐标系以速度 $\boldsymbol{d}_{Ox}$ 旋转，即旋转轴为 $Ox$ ，经过一段时间 $\delta t$ 之后，该动作对于旋转轴 $Ox$ 没有任何影响，即 $\boldsymbol{d}_{Ox}(t + \delta t) = \boldsymbol{d}_{Ox}(t)$ ；但是旋转轴 $Oy$ 在 $y-z$ 平面转过 $\delta t$ 弧度，在新的坐标系下若有单位速度 $\boldsymbol{d}_{Oy}(t + \delta t)$ ，相当于在原有坐标系下的速度 $\boldsymbol{d}_{Oy}(t) + \delta t \boldsymbol{d}_{Oz}$ ，因此 $\boldsymbol{d}_{Oy}(t + \delta t) = \boldsymbol{d}_{Oy}(t) + \delta t \boldsymbol{d}_{Oz}$ 。从而 $\boldsymbol{d}_{Ox}$ 和 $\boldsymbol{d}_{Oy}$ 的微分分别为：

$$\begin{aligned}
\frac{d}{dt} \boldsymbol{d}_{Ox} &= \lim_{\delta t \rightarrow 0} \frac{\boldsymbol{d}_{Ox}(t + \delta t) - \boldsymbol{d}_{Ox}(t)}{\delta t} = \mathbf{0} \\
\frac{d}{dt} \boldsymbol{d}_{Oy} &= \lim_{\delta t \rightarrow 0} \frac{\boldsymbol{d}_{Oy}(t + \delta t) - \boldsymbol{d}_{Oy}(t)}{\delta t} = \lim_{\delta t \rightarrow 0} \frac{\boldsymbol{d}_{Oy}(t) + \delta t \boldsymbol{d}_{Oz} - \boldsymbol{d}_{Oy}(t)}{\delta t} = \boldsymbol{d}_{Oz}
\end{aligned} \tag{5.25}$$

如图41(b), 若某一刚体坐标系以速度 $d_x$ 沿着 $x$ 方向移动, 经过时间 $\delta t$ 之后, 该动作对于旋转轴 $Ox$ 没有任何影响, 但是让 $Oy$ 在 $x$ 方向移动了 $\delta t$ 长度。因此 $d_{Ox}$ 在此过程中依然不变:  $d_{Ox}(t + \delta t) = d_{Ox}(t)$ ,  $\dot{d}_{Ox} = 0$ 。但在新的坐标系下若有单位速度 $d_{Oy}(t + \delta t)$ , 除了等效于原来的速度 $d_{Oy}(t)$ 外, 该角速度还将在 $O$ 点产生一个沿着 $z$ 方向的线速度 $\delta t d_z$ , 因此有:

$$\begin{aligned} d_{Oy}(t + \delta t) &= d_{Oy}(t) + \delta t d_z \\ \dot{d}_{Oy} &= d_z \end{aligned} \quad (5.26)$$

依照这种分析方法, 将所有6个轴的单位运动依次分析, 分别研究这6个单位运动对6个单位运动和6个单位力的影响, 排列组合可得72个微分结果, 如图41(c)(d)所示。其中, 前文对图41(a)的分析结果在图41(c)中被红色圈圈出, 对图41(b)的分析结果在图41(c)中由蓝色圈圈出。

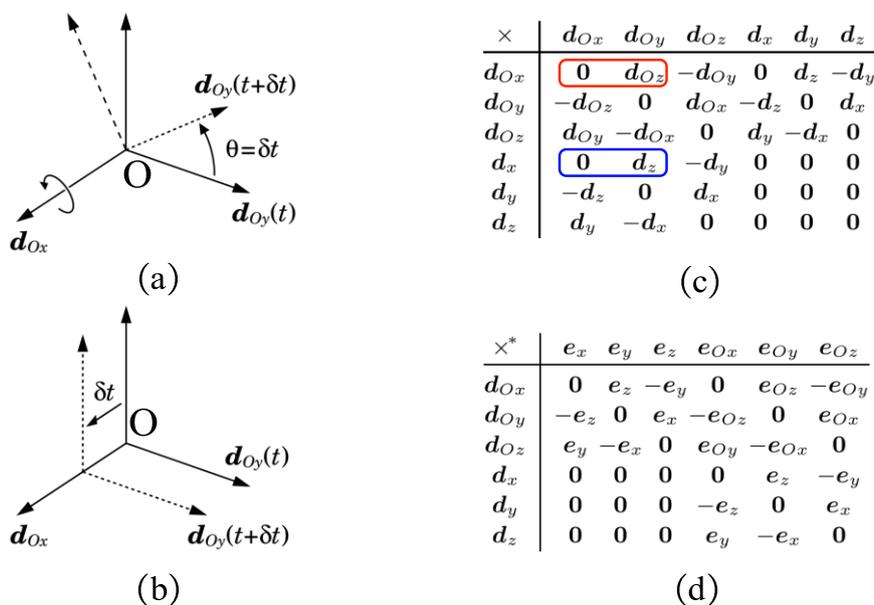


图41 空间向量叉乘

由图41(c)(d),  $v \times$ 与 $v \times^*$ 的定义为:

$$v \times = \begin{bmatrix} \boldsymbol{\omega} \\ \mathbf{v}_O \end{bmatrix} \times = \begin{bmatrix} \boldsymbol{\omega}_\times & \mathbf{0} \\ \mathbf{v}_{O \times} & \boldsymbol{\omega}_\times \end{bmatrix} \quad (5.27)$$

$$v \times^* = \begin{bmatrix} \boldsymbol{\omega} \\ \mathbf{v}_O \end{bmatrix} \times^* = \begin{bmatrix} \boldsymbol{\omega}_\times & \mathbf{v}_{O \times} \\ \mathbf{0} & \boldsymbol{\omega}_\times \end{bmatrix} \quad (5.28)$$

#### 5.1.4 空间加速度

空间加速度定义为空间速度对时间的微分, 即:

$$\mathbf{a} = \frac{d\mathbf{v}}{dt} = \frac{d}{dt} \begin{bmatrix} \boldsymbol{\omega} \\ \mathbf{v}_O \end{bmatrix} = \lim_{\delta t \rightarrow 0} \frac{1}{\delta t} \begin{bmatrix} \boldsymbol{\omega}(t + \delta t) - \boldsymbol{\omega}(t) \\ \mathbf{v}_O(t + \delta t) - \mathbf{v}_O(t) \end{bmatrix} = \begin{bmatrix} \dot{\boldsymbol{\omega}} \\ \dot{\mathbf{v}}_O \end{bmatrix} \quad (5.29)$$

该公式看起来显而易见，但与传统的加速度定义方法略有不同。在公式(5.1)中定义  $v_o$  为当前时刻与点  $O$  重合的刚体上一点的速度。然而，点  $O$  是空间中的不动点，由于刚体在运动，因此在  $t$  时刻和  $t + \delta t$  时刻，“与点  $O$  重合的刚体上一点”不是同一个点。如图 42 所示，设  $O'$  为  $t$  时刻与点  $O$  重合的刚体上一点， $r$  为  $O'$  的三维位置向量，因此  $\dot{r}$  是  $O'$  的速度， $\ddot{r}$  是  $O'$  的加速度。

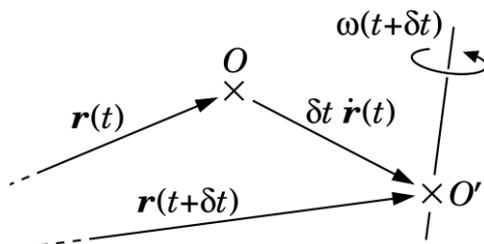


图42 由  $\ddot{r}$  计算  $\dot{v}_o$

在  $t$  时刻，由于  $O'$  与  $O$  重合，有  $v_o(t) = \dot{r}(t)$ 。但是在  $t + \delta t$  时刻， $O'$  与  $O$  已经不再重合，而是在  $\delta t$  时间内走过了  $\delta \dot{r}(t)$  的距离，因此  $v_o(t + \delta t)$  为：

$$v_o(t + \delta t) = \dot{r}(t + \delta t) - \omega \times \delta \dot{r}(t) \quad (5.30)$$

从而有：

$$\begin{aligned} \dot{v}_o &= \lim_{\delta t \rightarrow 0} \frac{1}{\delta t} [v_o(t + \delta t) - v_o(t)] \\ &= \lim_{\delta t \rightarrow 0} \frac{\dot{r}(t + \delta t) - \omega \times \delta \dot{r}(t) - \dot{r}(t)}{\delta t} \\ &= \ddot{r} - \omega \times \dot{r} \end{aligned} \quad (5.31)$$

因此空间加速度为：

$$a = \begin{bmatrix} \dot{\omega} \\ \dot{v}_o \end{bmatrix} = \begin{bmatrix} \dot{\omega} \\ \ddot{r} - \omega \times \dot{r} \end{bmatrix} \quad (5.32)$$

### 5.1.5 空间惯量与运动方程

#### (1) 空间动量

如图43所示，刚体的质心当前时刻与空间不动点  $C$  重合，其质量为  $m$ ，在质心处的惯性张量为  $\bar{I}_C$ ，当前时刻在  $C$  点的空间速度为  $\hat{v}_C = [\omega^T \ v_C^T]^T$ ，则该刚体的线动量为  $h = m v_C$ ，固有角动量为  $h_C = \bar{I}_C \omega$ ，对于空间上任意一点  $O$ ，刚体的动量矩为固有角动量与关于  $O$  的线动量的矩之和：

$$h_o = h_C + \overrightarrow{OC} \times h \quad (5.33)$$

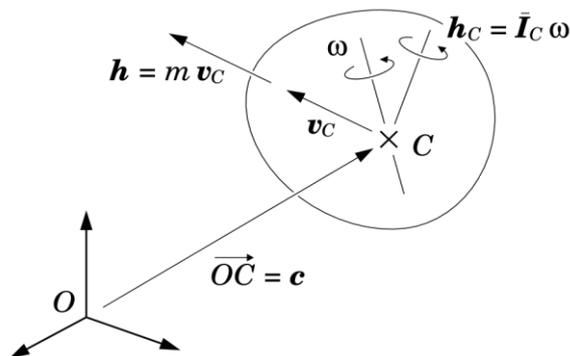


图43 刚体的动量

设  $\hat{\mathbf{h}}$  为刚体的空间动量,  $\hat{\mathbf{h}}_c$  与  $\hat{\mathbf{h}}_o$  分别为以  $C$  点和  $O$  点为原点的 Plücker 坐标系中描述的空间动量, 其值分别如下:

$$\hat{\mathbf{h}}_c = \begin{bmatrix} \mathbf{h}_c \\ \mathbf{h} \end{bmatrix} = \begin{bmatrix} \bar{\mathbf{I}}_C \boldsymbol{\omega} \\ m \mathbf{v}_C \end{bmatrix} \quad (5.34)$$

$$\hat{\mathbf{h}}_o = \begin{bmatrix} \mathbf{h}_o \\ \mathbf{h} \end{bmatrix} = \begin{bmatrix} \bar{\mathbf{I}}_C \boldsymbol{\omega} + \vec{OC} \times m \mathbf{v}_C \\ m \mathbf{v}_C \end{bmatrix} = \begin{bmatrix} \mathbb{I} & \vec{OC} \times \\ \mathbf{0} & \mathbb{I} \end{bmatrix} \hat{\mathbf{h}}_c \quad (5.35)$$

## (2) 空间惯量

类似于线动量, 在 Plücker 坐标系下构建空间速度到空间动量的映射关系:

$$\hat{\mathbf{h}} = \hat{\mathbf{I}} \hat{\mathbf{v}} \quad (5.36)$$

则 6 行 6 列的矩阵  $\hat{\mathbf{I}}$  称为空间惯量, 根据公式(5.34):

$$\hat{\mathbf{h}}_c = \begin{bmatrix} \mathbf{h}_c \\ \mathbf{h} \end{bmatrix} = \begin{bmatrix} \bar{\mathbf{I}}_C \boldsymbol{\omega} \\ m \mathbf{v}_C \end{bmatrix} = \begin{bmatrix} \bar{\mathbf{I}}_C & \mathbf{0} \\ \mathbf{0} & m \mathbb{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{\omega} \\ \mathbf{v}_C \end{bmatrix} = \begin{bmatrix} \bar{\mathbf{I}}_C & \mathbf{0} \\ \mathbf{0} & m \mathbb{I} \end{bmatrix} \hat{\mathbf{v}}_c \quad (5.37)$$

因此:

$$\hat{\mathbf{I}}_c = \begin{bmatrix} \bar{\mathbf{I}}_C & \mathbf{0} \\ \mathbf{0} & m \mathbb{I} \end{bmatrix} \quad (5.38)$$

根据公式(5.35), 有:

$$\hat{\mathbf{h}}_o = \begin{bmatrix} \mathbb{I} & \vec{OC} \times \\ \mathbf{0} & \mathbb{I} \end{bmatrix} \hat{\mathbf{h}}_c = \begin{bmatrix} \mathbb{I} & \vec{OC} \times \\ \mathbf{0} & \mathbb{I} \end{bmatrix} \hat{\mathbf{I}}_c \hat{\mathbf{v}}_c = \underbrace{\begin{bmatrix} \mathbb{I} & \vec{OC} \times \\ \mathbf{0} & \mathbb{I} \end{bmatrix} \hat{\mathbf{I}}_c}_{\hat{\mathbf{I}}_o} \begin{bmatrix} \mathbb{I} & \mathbf{0} \\ \mathbf{c}_x^T & \mathbb{I} \end{bmatrix} \hat{\mathbf{v}}_o \quad (5.39)$$

其中  $\mathbf{c} = \vec{OC}$ , 因此空间惯量  $\hat{\mathbf{I}}_o$  在 Plücker 坐标系下的一般形式为:

$$\hat{\mathbf{I}}_o = \begin{bmatrix} \mathbb{I} & \vec{OC} \times \\ \mathbf{0} & \mathbb{I} \end{bmatrix} \hat{\mathbf{I}}_c \begin{bmatrix} \mathbb{I} & \mathbf{0} \\ \mathbf{c}_x^T & \mathbb{I} \end{bmatrix} = \begin{bmatrix} \bar{\mathbf{I}}_C + m \mathbf{c}_x \mathbf{c}_x^T & m \mathbf{c}_x \\ m \mathbf{c}_x^T & m \mathbb{I} \end{bmatrix} \quad (5.40)$$

## (3) 运动方程

在 Plücker 坐标系下, 牛顿第二定律与欧拉方程被整合到一个方程中, 称为运动方

程。刚体受到合外力是刚体动量的变化率，因此：

$$\mathbf{f} = \frac{d}{dt}(\mathbf{I}\mathbf{v}) = \mathbf{I}\mathbf{a} + (\mathbf{v} \times \mathbf{I} - \mathbf{I}\mathbf{v} \times)\mathbf{v} = \mathbf{I}\mathbf{a} + \mathbf{v} \times \mathbf{I}\mathbf{v} \quad (5.41)$$

其中， $\mathbf{f}$  为刚体所受合外力， $\mathbf{I}$  为空间惯量， $\mathbf{v}$  为此时的空间速度， $\mathbf{a}$  为刚体在合外力  $\mathbf{f}$  的作用下所表现出来的空间加速度。

### 5.1.6 六维空间向量的性质

尽管 Plücker 坐标系下的一些六维空间向量的定义较传统方法颇为繁琐，但是一旦接受了这些定义与运算符，刚体的运动学性质与动力学性质就会简洁易用，本小节不加证明地提供一些常用六维空间向量的性质。

相对速度：若刚体 1 与刚体 2 的空间速度分别为  $\mathbf{v}_1$ 、 $\mathbf{v}_2$ ，则刚体 2 相对于刚体 1 的空间速度为  $\mathbf{v}_{rel} = \mathbf{v}_2 - \mathbf{v}_1$ ，加速度也符合这一简单相减的特性。

刚性连接：若两个刚体被刚性连接在一起，则他们的空间速度相同。

空间惯量可加性：刚体 1 与刚体 2 的空间惯量分别为  $I_1$ 、 $I_2$ ，若他们刚性连接在一起，则组合刚体的空间惯量为  $I_1 + I_2$ ，与两刚体的相对位置无关。

空间力可加性：若两个空间力  $\mathbf{f}_1$ 、 $\mathbf{f}_2$  同时作用于同一个刚体，这两个空间力的作用效果与  $\mathbf{f}_{tot} = \mathbf{f}_1 + \mathbf{f}_2$  相同。

作用力与反作用力：若刚体 1 在刚体 2 上施加空间力  $\mathbf{f}$ ，则刚体 2 在刚体 1 上施加空间力  $-\mathbf{f}$ 。

微分：六维运动向量的微分是六维运动向量，六维动力向量的微分是六维动力向量。

功率：若空间力  $\mathbf{f}$  作用于有空间速度  $\mathbf{v}$  的刚体上，则  $\mathbf{f}$  的功率为  $\mathbf{f} \cdot \mathbf{v}$ 。

动能：若一个空间惯量为  $I$  的刚体具有空间速度  $\mathbf{v}$ ，其动能为  $0.5\mathbf{v} \cdot \mathbf{I}\mathbf{v}$ 。

## 5.2 多刚体系统的运动学建模

多刚体系统由许多刚体以及刚体与刚体之间的关节组成，关节负责形成系统中的运动学约束。在这一小节，将介绍如何建立一个方便地被数学语言和程序语言所使用的刚体系统模型。

### 5.2.1 运动学树

机构的联通特性是指机构中刚体与关节连接的拓朴结构，可以用运动学树来表示。图44为四足机器人的运动学树示意图，其中：实心圆代表实际刚体，空心圆代表固定基座（虚拟的），线段代表关节；为每个刚体编号，固定基座编号为 0，浮动基座编号为 1，

依照右前腿、左前腿、右后腿、左后腿的顺序，依次为第 1、2、3 连杆编号；每个关节都有两个刚体，编号较小的那个为父刚体，较大的那个为子刚体；为每个关节编号，编号规则为刚体*i*通过关节*i*连接到其父刚体，注意，第 1 个关节为虚拟的 6 自由度关节。

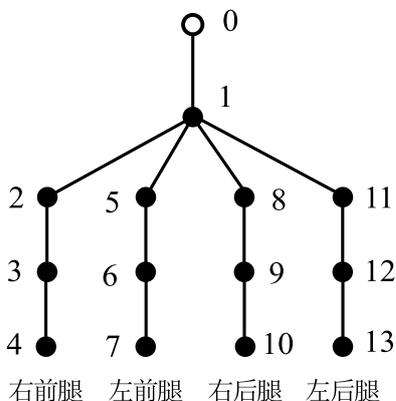


图44 四足机器人运动学树

设*N*为刚体的数量，则本文所述四足机器人*N* = 13，同时*N*也是关节的数量。定义λ(*i*)为刚体*i*的父刚体，若将λ(1)、λ(2)、…、λ(*N*)组成一个集合，称该集合为父节点集：

$$\lambda = \{\lambda(1), \lambda(2), \dots, \lambda(N)\} \tag{5.42}$$

图44中四足机器人的父节点集为：

$$\lambda = \{0, 1, 2, 3, 1, 5, 6, 1, 8, 9, 1, 11, 12\} \tag{5.43}$$

显然，通过父节点集，以及运动学树的构建规则，就能完整描述运动学树的拓扑结构。尽管如此，为了方便后续算法使用，如图45所示，再定义其他三个经常用到的集合。

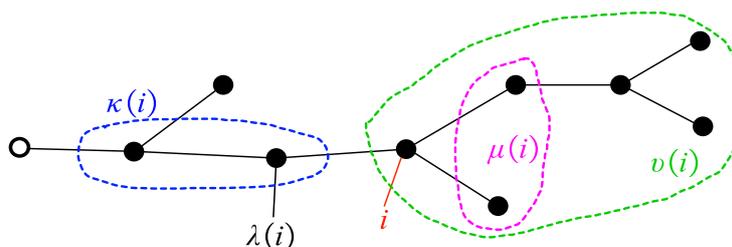


图45 κ、λ、μ、ν的定义示意图

其中，κ(*i*)是除了 0 号节点外，所有支撑起*i*节点的节点集合；μ(*i*)是以*i*节点为父节点的节点集合；ν(*i*)是以*i*节点为根节点的一个子树上的节点的集合。

### 5.2.2 关节模型

#### (1) 关节上的坐标变换

关节是两个刚体之间的运动约束，同时也因运动约束而产生了约束力。如图46所示，为了讨论刚体在关节上的运动与受力关系，为第*i*个关节上的子刚体与父刚体分别建立

刚体坐标系 $\mathcal{B}_i$ 和 $\mathcal{B}_{\lambda(i)}$ ，其中 $\mathcal{B}_0$ 为前文中所定义的世界系， $\mathcal{B}_1$ 为前文中所定义的本体系；另外，在父刚体上建立坐标系 $\mathcal{B}_{\lambda(i),i}$ ，满足当关节 $i$ 的参数为 $\mathbf{0}$ 时 $\mathcal{B}_{\lambda(i),i}$ 与 $\mathcal{B}_i$ 重合。

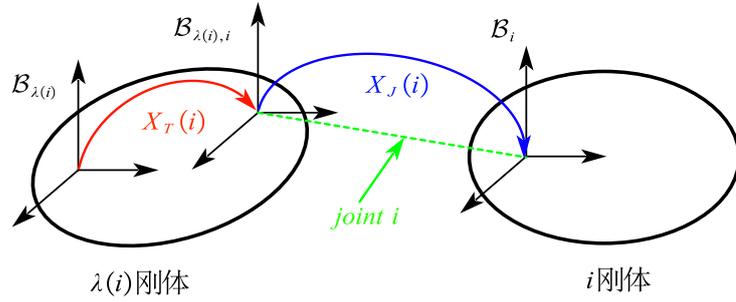


图46 关节上的坐标变换

定义 $X_T(i)$ 为从 $\mathcal{B}_{\lambda(i)}$ 到 $\mathcal{B}_{\lambda(i),i}$ 的坐标变换矩阵，由于 $\mathcal{B}_{\lambda(i)}$ 与 $\mathcal{B}_{\lambda(i),i}$ 都刚接到刚体 $\lambda(i)$ 上，因此 $X_T(i)$ 是个常值矩阵，其具体数值可结合公式(2.2)中四足机器人的尺寸信息，以及图10中刚体坐标系的位姿定义轻易获取，本文不做详细阐述。

定义 $X_J(i)$ 为 $\mathcal{B}_{\lambda(i),i}$ 到 $\mathcal{B}_i$ 的坐标变换矩阵，该矩阵是关节参数的函数。设第 $i$ 个关节的关节参数为 $\mathbf{q}_i$ ， $\mathbf{q}_i$ 的行数等于关节 $i$ 的自由度数，若关节 $i$ 是单自由度关节， $\mathbf{q}_i$ 是1行1列向量， $\mathbf{q}_1$ 是6行1列的向量。所有的关节参数按顺序写成一个列向量，组成了公式(4.32)中的关节空间向量。由于 $\mathbf{q}_1 = [\Theta^T \quad {}^0\mathbf{p}_{com}^T]^T$ ，结合公式(5.22)， $X_J(1)$ 为：

$$X_J(1) = \begin{bmatrix} {}^B R_O & \mathbf{0} \\ -{}^B R_O {}^O \mathbf{p}_{com \times} & {}^B R_O \end{bmatrix} \quad (5.44)$$

其中旋转矩阵 ${}^B R_O$ 由欧拉角 $\Theta$ 通过公式(2.20)求解。除了关节1是虚拟的6自由度关节外，四足机器人上其他关节都是单自由度旋转关节，且每条腿上的第一关节总是绕 $x$ 轴旋转，第二、三关节总是绕 $y$ 轴旋转。对于绕 $x$ 轴旋转的关节， $X_J(i)$ 为：

$$X_J(i) = \begin{bmatrix} R_x^T(q_i) & \mathbf{0} \\ \mathbf{0} & R_x^T(q_i) \end{bmatrix} \quad (5.45)$$

对于绕 $y$ 轴旋转的关节， $X_J(i)$ 为：

$$X_J(i) = \begin{bmatrix} R_y^T(q_i) & \mathbf{0} \\ \mathbf{0} & R_y^T(q_i) \end{bmatrix} \quad (5.46)$$

其中 $R_x$ 、 $R_y$ 是公式(2.12)中分别绕 $x$ 、 $y$ 轴的基本旋转矩阵。至此，从坐标系 $\mathcal{B}_{\lambda(i)}$ 到坐标系 $\mathcal{B}_i$ 的完整坐标变换矩阵为：

$${}^i X_{\lambda(i)} = X_J(i) X_T(i) \quad (5.47)$$

## (2) 关节速度与加速度

关节速度定义为子刚体与父刚体之间的相对速度：

$$\mathbf{v}_{J,i} = \mathbf{v}_i - \mathbf{v}_{\lambda(i)} \quad (5.48)$$

显然，关节速度参数 $\dot{\mathbf{q}}_i$ 直接决定了 $\mathbf{v}_J$ 的数值，由于 $\mathbf{v}_J$ 是6行1列的空间速度，而 $\dot{\mathbf{q}}_i$ 的行数取决于该关节的自由度数，因此 $\mathbf{v}_J \neq \dot{\mathbf{q}}_i$ 。构建 $\mathbf{v}_J$ 与 $\dot{\mathbf{q}}_i$ 的映射关系：

$$\mathbf{v}_{J,i} = S_i \dot{\mathbf{q}}_i \quad (5.49)$$

本质上关节是对刚体运动的约束，在不受约束的情况下，空间速度是运动向量空间 $M^6$ 的元素；但是在受到关约束后， $\mathbf{v}_{J,i}$ 被约束到一个 $M^6$ 的子空间中，称为关节运动子空间，因此 $S_i$ 被称为关节运动子空间映射矩阵，该矩阵的每一列都是六维运动向量。将(5.49)代入(5.48)：

$$\mathbf{v}_i = \mathbf{v}_{\lambda(i)} + S_i \dot{\mathbf{q}}_i \quad (5.50)$$

公式(5.50)提供了从父刚体到子刚体空间速度的递推公式。在 $B_i$ 中讨论该公式：

$${}^i \mathbf{v}_i = {}^i \mathbf{v}_{\lambda(i)} + {}^i S_i \dot{\mathbf{q}}_i \quad (5.51)$$

其中左上标 $i$ 代表变量在第 $i$ 个刚体的刚体坐标系 $B_i$ 下描述。在 $B_i$ 坐标系下，关节的旋转轴就是坐标系的坐标轴，因此若关节 $i$ 绕 $x$ 轴旋转，则：

$${}^i S_i = [1 \ 0 \ 0 \ 0 \ 0 \ 0]^T \quad (5.52)$$

若关节 $i$ 绕 $y$ 轴旋转：

$${}^i S_i = [0 \ 1 \ 0 \ 0 \ 0 \ 0]^T \quad (5.53)$$

关节绕 $z$ 轴旋转同理，只需将6行1列的零向量第三个元素改为1即可。若 $i=1$ ，即关节 $i$ 是六自由度关节，则：

$${}^1 S_1 = \mathbb{I}_{6 \times 6} \quad (5.54)$$

对公式(5.50)等号两边同时微分：

$$\begin{aligned} \mathbf{a}_i &= \mathbf{a}_{\lambda(i)} + S_i \ddot{\mathbf{q}}_i + \dot{S}_i \dot{\mathbf{q}}_i \\ &= \mathbf{a}_{\lambda(i)} + S_i \ddot{\mathbf{q}}_i + \mathbf{v}_i \times S_i \dot{\mathbf{q}}_i \end{aligned} \quad (5.55)$$

公式(5.50)提供了从父刚体空间加速度到子刚体空间加速度的递推公式。

### 5.2.3 足底接触点

设足底的数目为 $N_c$ ，对于四足机器人显然 $N_c = 4$ 。如图10所示，除了在每个刚体上建立一个刚体坐标系外，还在四个足底建立了足底坐标系 $C_i$ 。设足底坐标为 $C_i$ ，即 $C_i$ 为 $C_i$ 的坐标原点。设 $\rho(i)$ 为第 $i$ 个支撑足所刚接的刚体编号，从图10中不难看出：

$$\{\rho(1), \rho(2), \rho(3), \rho(4)\} = \{4, 7, 10, 13\} \quad (5.56)$$

定义  $X_C(i)$  为从  $B_{\rho(i)}$  到  $C_i$  的坐标变换, 显然  $X_C(i)$  是常值矩阵, 且四个  $X_C(i)$  都一样:

$$X_C(i) = \begin{bmatrix} \mathbb{I}_{3 \times 3} & \mathbf{0} \\ -([0 \ 0 \ -l_3]^T)_{\times} & \mathbb{I}_{3 \times 3} \end{bmatrix} \quad (5.57)$$

其中  $l_3$  为图11中机器人小腿长度。

### 5.2.4 递推正运动学

由公(5.47)(5.49)(5.50)(5.55)(5.57)联立可得机器人递推正运动学算法:

$$\begin{aligned} & \mathbf{v}_0 = \mathbf{0}, \quad \mathbf{a}_0 = \mathbf{0}, \quad {}^0X_0 = \mathbb{I}_{6 \times 6} \\ & \mathbf{for} \ i = 1 \ \mathbf{to} \ N \ \mathbf{do} \\ & \quad {}^iX_{\lambda(i)} = X_J(i) X_T(i) \\ & \quad {}^iX_0 = {}^iX_{\lambda(i)} {}^{\lambda(i)}X_0 \\ & \quad {}^i\mathbf{v}_{J,i} = {}^iS_i \dot{\mathbf{q}}_i \\ & \quad {}^i\mathbf{v}_i = {}^iX_{\lambda(i)} {}^{\lambda(i)}\mathbf{v}_{\lambda(i)} + {}^i\mathbf{v}_{J,i} \\ & \quad {}^i\mathbf{a}_i = {}^iX_{\lambda(i)} {}^{\lambda(i)}\mathbf{a}_{\lambda(i)} + {}^iS_i \ddot{\mathbf{q}}_i + {}^i\mathbf{v}_i \times {}^iS_i \dot{\mathbf{q}}_i \\ & \mathbf{end} \end{aligned} \quad (5.58)$$

该算法的输入为  $X_J(i)$ 、 $X_T(i)$ 、 ${}^iS_i$ 、 $\dot{\mathbf{q}}_i$ 、 $\ddot{\mathbf{q}}_i$ , 输出为每个刚体与其父刚体的坐标变换矩阵  ${}^iX_{\lambda(i)}$ , 固定基座坐标系到每个刚体坐标系的坐标变换矩阵  ${}^iX_0$ , 在  $B_i$  坐标系下的关节速度  ${}^i\mathbf{v}_{J,i}$ 、刚体空间速度  ${}^i\mathbf{v}_i$ 、刚体空间加速度  ${}^i\mathbf{a}_i$ 。

### 5.2.5 全身控制雅可比矩阵 $J_1$ 、 $J_4$ 及各级任务 $J_i \dot{\mathbf{q}}_i$ 项

在第四章全身控制伪代码(4.36)中, 假设了第 1、4 任务的雅可比矩阵  $J_1$ 、 $J_4$ , 以及各级任务的  $J_i \dot{\mathbf{q}}_i$  项已知, 本小节将在 Plücker 坐标系下求出这 6 个矩阵。

#### (1) 全身控制雅可比矩阵 $J_1$ 、 $J_4$

由公式(5.50)可得刚体  $i$  的空间速度的非递推公式:

$$\mathbf{v}_i = \sum_{j \in \kappa(i)} S_j \dot{\mathbf{q}}_j \quad (5.59)$$

若定义选择变量  $\epsilon_{ij}$  为:

$$\epsilon_{ij} = \begin{cases} 1, & j \in \kappa(i) \\ 0, & j \notin \kappa(i) \end{cases} \quad (5.60)$$

公式(5.59)可重写为:

$$\mathbf{v}_i = \sum_{j \in \kappa(i)} S_j \dot{\mathbf{q}}_j = \sum_{j=1}^N \epsilon_{ij} S_j \dot{\mathbf{q}}_j = \underbrace{[\epsilon_{i1} S_1 \ \cdots \ \epsilon_{iN} S_N]}_{b^{J_i}} \begin{bmatrix} \dot{\mathbf{q}}_1 \\ \vdots \\ \dot{\mathbf{q}}_{N_B} \end{bmatrix} \quad (5.61)$$

即:

$$\mathbf{v}_i = {}_b J_i \dot{\mathbf{q}} \quad (5.62)$$

其中 ${}_b J_i$ 称为刚体雅可比矩阵，是关节空间广义速度 $\dot{\mathbf{q}}$ 到刚体空间速度 $\mathbf{v}_i$ 的映射矩阵。公式(5.61)忽略了各个量所在的坐标系，在小腿的刚体坐标系求各个小腿的 ${}_b J_i$ 伪代码为：

```

for  $i = 1$  to  $N_C$  do
     $j = \rho(i)$ ,  ${}^{\rho(i)} X_j = \mathbb{I}_{6 \times 6}$ ,  ${}_b J_{\rho(i)} = \mathbf{0}$ ,  ${}_b J_{\rho(i), \rho(i)} = {}^{\rho(i)} S_{\rho(i)}$ 
    while  $\lambda(j) > 0$  do
         ${}^{\rho(i)} X_{\lambda(j)} = {}^{\rho(i)} X_j {}^j X_{\lambda(j)}$ 
         $j = \lambda(j)$ 
         ${}_b J_{\rho(i), j} = {}^{\rho(i)} X_j {}^j S_j$ 
    end
end
    
```

(5.63)

其中 ${}_b J_{\rho(i), j}$ 行数为6，列数为第 $j$ 关节的自由度数。

与定向本体类似，在第 $i$ 个足底上建立定向足底系 $Q_i$ ，其坐标原点与足底坐标系重合，方向与世界系相同。设 ${}^{\rho(i)} R_0$ 为从固定基座坐标系到第 $i$ 个小腿的刚体坐标系的旋转矩阵，在递推正运动学算法(5.58)中已经解算出了 ${}^{\rho(i)} X_0$ ，根据Plücker坐标系下坐标变换矩阵的定义，可知 ${}^{\rho(i)} R_0$ 是 ${}^{\rho(i)} X_0$ 右下角的三行三列矩阵。构建第 $i$ 个支撑足的足底坐标系到定向足底系的坐标变换矩阵：

$$X_{Q_i}(i) = \begin{bmatrix} {}^{\rho(i)} R_0^T & \mathbf{0} \\ \mathbf{0} & {}^{\rho(i)} R_0^T \end{bmatrix} \quad (5.64)$$

将刚体雅可比矩阵 ${}_b J_{\rho(i)}$ 从刚体坐标系 $\mathcal{B}_{\rho(i)}$ 转到定向足底系 $Q_i$ ：

$${}^Q J_i = X_{Q_i}(i) X_C(i) {}_b J_{\rho(i)} \quad (5.65)$$

${}^Q J_i$ 是关节空间广义速度 $\dot{\mathbf{q}}$ 到 $Q_i$ 下描述的第 $\rho(i)$ 个刚体的空间速度的映射矩阵，由于定向足底系方向与世界系相同，坐标原点是 $C_i$ ，根据空间速度的定义，在定向足底系下描述的空间速度的平动速度分量，就是 $C_i$ 在世界系下的速度。根据公式(5.62)， ${}^Q J_i$ 是6行18列的矩阵，前3行对应了角速度，后3行对应了平动速度。由于全身控制的第1、4个任务仅考虑对足底的平动控制，因此设 ${}^Q J'_i$ 为 ${}^Q J_i$ 的后三行，即：

$${}^Q J'_i = {}^Q J_{i, block}(4, 1, 3, 18) \quad (5.66)$$

则全身控制第1任务的雅可比矩阵 $J_1$ 是将所有支撑足的 ${}^Q J'_i$ 按列组合成一个大矩阵，第4任务的雅可比矩阵 $J_4$ 是将所有摆动足的 ${}^Q J'_i$ 按列组合成一个大矩阵，伪代码为：

$$\begin{aligned}
& J_1 = \mathbf{0}_{0 \times 0}, \quad J_4 = \mathbf{0}_{0 \times 0} \\
& \text{for } i = 1 \text{ to } N_C \text{ do} \\
& \quad \text{if } {}^i s_\phi = 1 \text{ then} \\
& \quad \quad J_1 = [J_1^T \quad ({}^Q J_i')^T]^T \\
& \quad \text{else} \\
& \quad \quad J_4 = [J_4^T \quad ({}^Q J_i')^T]^T \\
& \quad \text{end} \\
& \text{end}
\end{aligned} \tag{5.67}$$

(2) 全身控制各级任务  $J_i \dot{q}$  项

对公式(4.2)左右两边同时微分:

$$\ddot{\mathbf{x}} = \mathbf{J}\ddot{\mathbf{q}} + \dot{\mathbf{J}}\dot{\mathbf{q}} \tag{5.68}$$

可见, 全身控制各级任务  $J_i \dot{q}$  项, 可以理解为  $\ddot{\mathbf{q}} = 0$  时的工作空间加速度, 对应任务 1 和任务 4 来说, 就是足底坐标在  $\ddot{\mathbf{q}} = 0$  时世界系下的三维加速度。设  $\ddot{\mathbf{q}} = 0$  时刚体  $i$  的空间加速度为  $\mathbf{a}_i^{vp}$ , 根据公式(5.55),  $\mathbf{a}_i^{vp}$  的递推公式为:

$$\mathbf{a}_i^{vp} = \mathbf{a}_{\lambda(i)}^{vp} + \mathbf{v}_i \times S_i \dot{q}_i \tag{5.69}$$

可见,  $\mathbf{a}_i^{vp}$  是  $\kappa(i)$  中各个刚体的空间速度与对应的关节速度的叉乘的累积和, 称该加速度为空间  $VP$  加速度。根据公式(5.50), 对于浮动基体, 即刚体 1, 有:

$$\mathbf{a}_1^{vp} = \mathbf{a}_0^{vp} + \mathbf{v}_1 \times S_1 \dot{q}_1 = \mathbf{0} + \mathbf{v}_1 \times \mathbf{v}_1 = \mathbf{0} \tag{5.70}$$

因此全身控制的第 2、3 任务的  $J_i \dot{q}$  项为  $\mathbf{0}$ , 即:

$$\begin{cases} J_2 \dot{q} = \mathbf{0}_{3 \times 1} \\ J_3 \dot{q} = \mathbf{0}_{3 \times 1} \end{cases} \tag{5.71}$$

现在考虑第 1、4 任务的  $J_i \dot{q}$  项。将递推正运动学(5.58)稍做修改, 求各刚体的空间  $VP$  加速度  $\mathbf{a}_i^{vp}$ :

$$\begin{aligned}
& \mathbf{a}_0^{vp} = \mathbf{0} \\
& \text{for } i = 1 \text{ to } N \text{ do} \\
& \quad {}^i \mathbf{a}_i^{vp} = {}^i X_{\lambda(i)}^{\lambda(i)} \mathbf{a}_{\lambda(i)}^{vp} + {}^i \mathbf{v}_i \times {}^i S_i \dot{q}_i \\
& \text{end}
\end{aligned} \tag{5.72}$$

将第  $i$  个小腿的空间  $VP$  加速度  ${}^i \mathbf{a}_{\rho(i)}^{vp}$  及其空间速度  ${}^{\rho(i)} \mathbf{v}_{\rho(i)}$  转换到对应的平动足底系下:

$$\begin{cases} {}^Q \mathbf{a}_{\rho(i)}^{vp} = X_Q(i) X_C(i) {}^i \mathbf{a}_{\rho(i)}^{vp} \\ {}^Q \mathbf{v}_{\rho(i)} = X_Q(i) X_C(i) {}^{\rho(i)} \mathbf{v}_{\rho(i)} \end{cases} \tag{5.73}$$

设 ${}^o\mathbf{a}_{\rho(i)}^y$ 中后三列，即平动加速度项为 $\mathbf{a}_{\rho(i)}^{C_i}$ ， ${}^o\mathbf{v}_{\rho(i)}$ 中平动速度项与转动速度项分别为 $\mathbf{v}_{\rho(i)}^{C_i}$ 、 $\boldsymbol{\omega}_{\rho(i)}$ ，根据空间加速度的定义(5.32)，足底坐标点 $C_i$ 在世界系下的 $VP$ 加速度为：

$${}^o\mathbf{a}_{C_i} = \mathbf{a}_{\rho(i)}^{C_i} + \boldsymbol{\omega}_{\rho(i)} \times \mathbf{v}_{\rho(i)}^{C_i} \quad (5.74)$$

则全身控制第1任务的 $J_1\dot{\mathbf{q}}$ 项是将所有支撑足的 ${}^o\mathbf{a}_{C_i}$ 按列组合成一个向量，第4任务的 $J_4\dot{\mathbf{q}}$ 项是将所有摆动足的 ${}^o\mathbf{a}_{C_i}$ 按列组合成一个向量，伪代码如下：

$$\begin{aligned} & J_1\dot{\mathbf{q}} = \mathbf{0}_{0 \times 0}, \quad J_4\dot{\mathbf{q}} = \mathbf{0}_{0 \times 0} \\ & \text{for } i = 1 \text{ to } N_c \text{ do} \\ & \quad \text{if } {}_iS_{\phi} = 1 \text{ then} \\ & \quad \quad J_1\dot{\mathbf{q}} = \left[ \left( J_1\dot{\mathbf{q}} \right)^T \quad \left( {}^o\mathbf{a}_{C_i} \right)^T \right]^T \\ & \quad \text{else} \\ & \quad \quad J_4\dot{\mathbf{q}} = \left[ \left( J_4\dot{\mathbf{q}} \right)^T \quad \left( {}^o\mathbf{a}_{C_i} \right)^T \right]^T \\ & \quad \text{end} \\ & \text{end} \end{aligned} \quad (5.75)$$

### 5.3 多刚体系统的动力学建模

多刚体系统的动力学算法通常用来处理作用在系统上的力与这些力产生的加速度之间的关系。动力学算法是计算与动力学有关的量的具体数值的过程，浮动基座机器人在关节空间的动力学方程通常具有以下标准的形式：

$$M(\mathbf{q})\ddot{\mathbf{q}} + C(\mathbf{q}, \dot{\mathbf{q}}) = S_j\boldsymbol{\tau} + J_c^T \mathbf{f}_c \quad (5.76)$$

关于该动力学方程的介绍在第四章已经有详细阐述。使用该动力学方程，主要用来实现正动力学算法与逆动力学算法。

**正动力学算法：**给定系统受到的作用力，求系统的加速度。正运动学算法通常用于物理仿真引擎中。系统既可能受到内部关节扭矩 $\boldsymbol{\tau}$ 也可能受到外部接触力 ${}^o\mathbf{f}_c$ ，另外由于关节约束力在关节运动方向没有位移，因此关节约束力不考虑在内。总得来说，正运动学算法可以用如下方程描述：

$$\ddot{\mathbf{q}} = FD(model, \mathbf{q}, \dot{\mathbf{q}}, \boldsymbol{\tau}, \mathbf{f}_c) \quad (5.77)$$

在公式(5.77)中，*model*是指系统模型，在这里作为正动力学算法的一个参数。

**逆动力学算法：**给定系统的加速度，求需要施加系统怎样的作用力。逆动力学算法可以应用于运动控制、轨迹规划、机械设计等多种场景下。多刚体模型在关节上通常装有驱动电机，以实现关节扭矩 $\boldsymbol{\tau}$ ；但是系统与外部的作用力 $\mathbf{f}_c$ 无法直接使用驱动电机来实现，通常是被动产生的：如发生碰撞，或通过调整多个电机的扭矩来间接实现。因此

逆动力学算法中  $f_c$  一般作为已知量，由外部传感器或内部规划器给出。总得来说，逆动力学算法可以用如下方程描述：

$$\boldsymbol{\tau} = ID(\text{model}, \mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}, \mathbf{f}_c) \quad (5.78)$$

### 5.3.1 递推牛顿欧拉法

#### (1) 关节力

设  $\mathbf{f}_J$  为父刚体通过关节作用到子刚体上的力，该力分为两个部分：若在关节允许运动的方向上有驱动电机，驱动电机的扭矩  $\boldsymbol{\tau}$  所形成的空间力  $\mathbf{f}_\tau$ ；在关节不允许运动的方向，产生的关节约束力  $\mathbf{f}_b$ ，即：

$$\mathbf{f}_J = \mathbf{f}_\tau + \mathbf{f}_b \quad (5.79)$$

由于关节运动所产生的关节位移仅出现在关节允许运动的方向上，而  $\mathbf{f}_b$  只在关节不允许运动的方向上出现，因此  $\mathbf{f}_b$  不做功。设关节有虚位移  $\delta \mathbf{q}$ ，导致  $\mathbf{v}_J$  方向有虚位移  $\delta \mathbf{d}$ ，根据公式(5.49)，在极短的时间内有：

$$\delta \mathbf{d} = \mathbf{S} \delta \mathbf{q} \quad (5.80)$$

根据虚功原理，任意虚位移产生的虚功为 0：

$$\boldsymbol{\tau}^T \delta \mathbf{q} = \mathbf{f}_J^T \delta \mathbf{d} = \mathbf{f}_J^T \mathbf{S} \delta \mathbf{q} \quad (5.81)$$

从而有：

$$\boldsymbol{\tau} = \mathbf{S}^T \mathbf{f}_J \quad (5.82)$$

#### (2) 刚体受力分析

如图47所示，分析刚体  $i$  的受力情况。在串联型机器人中，一个刚体可以有多个子刚体，但是只能有一个父刚体。设  $i$  子刚体集为  $\mu(i) = \{j, k, l\}$ ，则刚体  $i$  的受力有：父刚体通过关节  $i$  传递来的关节力  $\mathbf{f}_i$ ，子刚体通过关节  $j$ 、 $k$ 、 $l$  传来的反作用力  $-\mathbf{f}_j$ 、 $-\mathbf{f}_k$ 、 $-\mathbf{f}_l$ ，以及外部环境作用力  $\mathbf{f}_i^c$ 。这里暂时不考虑重力，在后文中将用惯性力来实现重力。

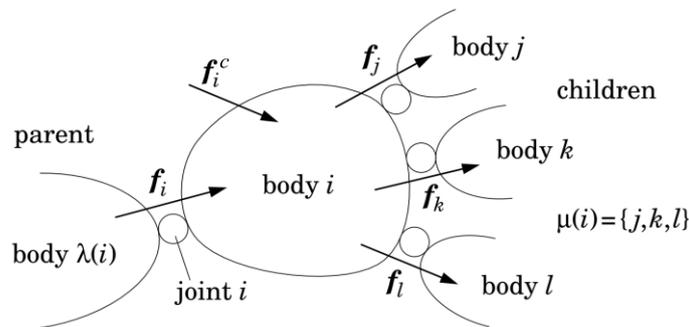


图47 刚体  $i$  受力分析

分析图47可知，刚体*i*所受合外力为：

$$\mathbf{f}_i^B = \mathbf{f}_i + \mathbf{f}_i^c - \sum_{j \in \mu(i)} \mathbf{f}_j \quad (5.83)$$

将刚体*i*所受合外力 $\mathbf{f}_i^B$ 代入运动方程(5.41)：

$$\mathbf{f}_i^B = I_i \mathbf{a}_i + \mathbf{v}_i \times {}^* I_i \mathbf{v}_i \quad (5.84)$$

若已知其他力，也可以通过公式(5.83)，来求关节力 $\mathbf{f}_i$ ：

$$\mathbf{f}_i = \mathbf{f}_i^B - \mathbf{f}_i^c + \sum_{j \in \mu(i)} \mathbf{f}_j \quad (5.85)$$

### (3) 递推牛顿欧拉法伪代码

结合公式(5.51)、(5.55)、(5.49)、(5.82)、(5.84)、(5.85)，可以实现逆动力学算法，即递推牛顿欧拉法：

```

1 :    $\mathbf{v}_0 = 0, \mathbf{a}_0 = -\mathbf{a}_g, {}^0X_0 = \mathbb{I}_{6 \times 6}$ 
2 :   for  $i = 1$  to  $N$  do
3 :      ${}^iX_{\lambda(i)} = X_J(i) X_T(i)$ 
4 :      ${}^iX_0 = {}^iX_{\lambda(i)} {}^{\lambda(i)}X_0$ 
5 :      ${}^iX_0^* = {}^iX_0^{-T}$ 
6 :      ${}^i\mathbf{v}_i = {}^iX_{\lambda(i)} {}^{\lambda(i)}\mathbf{v}_{\lambda(i)} + {}^iS_i \dot{\mathbf{q}}_i$ 
7 :      ${}^i\mathbf{a}_i = {}^iX_{\lambda(i)} {}^{\lambda(i)}\mathbf{a}_{\lambda(i)} + {}^iS_i \ddot{\mathbf{q}}_i + {}^i\mathbf{v}_i \times {}^iS_i \dot{\mathbf{q}}_i$ 
8 :      ${}^i\mathbf{f}_i = I_i^i \mathbf{a}_i + {}^i\mathbf{v}_i \times {}^* I_i^i \mathbf{v}_i - {}^iX_0^{*0} \mathbf{f}_i^c$ 
9 :   end
10:  for  $i = N$  to  $1$  do
11:     $\boldsymbol{\tau}_i = {}^iS_i^T \mathbf{f}_i$ 
12:    if  $\lambda(i) \neq 0$  then
13:       ${}^{\lambda(i)}\mathbf{f}_{\lambda(i)} = {}^{\lambda(i)}\mathbf{f}_{\lambda(i)} + {}^{\lambda(i)}X_0^{*i} \mathbf{f}_i$ 
14:    end
15:  end

```

在该算法中，尽管一般意义上固定基座是不动的，但是令 $\mathbf{a}_0 = -\mathbf{a}_g$ ，相当于给系统了一个沿着*z*轴向上的重力加速度，此时世界系变成了非惯性系，在该非惯性系下做受力分析，相当于所有刚体都受到一个沿着*z*轴向下的惯性力，从而模拟了重力的效果，但是，这样迭代出的空间加速度都不是真实空间加速度。

${}^iX_0^*$ 与 ${}^{\lambda(i)}X_0^*$ 是从世界系到 $\mathcal{B}_i$ 和 $\mathcal{B}_{\lambda(i)}$ 的坐标变换矩阵， ${}^{\lambda(i)}X_0^*$ 右上角的“\*”表明其作用于动力向量。 ${}^0\mathbf{f}_i^c$ 是世界系下的外部作用力， $\boldsymbol{\tau}_i$ 是第*i*个关节的关节扭矩。

在第二个**for**循环中，*i*从最后一个刚体开始处理，这样能保证每一个刚体都比其父刚体提前处理，从而保证在计算 $\boldsymbol{\tau}_i$ 时， ${}^i\mathbf{f}_i$ 中仅包含了刚体 $\lambda(i)$ 对刚体*i*的关节力，而不包

括子刚体集 $\mu(i)$ 中任何一个刚体对刚体 $i$ 的反作用力。

(4) 通过递推牛顿欧拉法求 $C$ 向量

通过分析公式(5.76)与(5.78)发现:

$$C = ID(model, \mathbf{q}, \dot{\mathbf{q}}, \mathbf{0}, \mathbf{0}) \quad (5.87)$$

即 $\ddot{\mathbf{q}} = \mathbf{0}$ 且 $\mathbf{f}_c = \mathbf{0}$ 时的关节扭矩, 而逆动力学算法 $ID$ 已经由递推牛顿欧拉法实现, 因此只需要将(5.86)中的第7行的 $\ddot{\mathbf{q}}$ 和第8行的 ${}^o\mathbf{f}_i^c$ 置零即可, 即:

$$\begin{aligned} 7: & \quad {}^i\mathbf{a}_i = {}^iX_{\lambda(i)}{}^{\lambda(i)}\mathbf{a}_{\lambda(i)} + {}^i\mathbf{v}_i \times {}^iS_i\dot{\mathbf{q}}_i \\ 8: & \quad {}^i\mathbf{f}_i = I_i^i\mathbf{a}_i + {}^i\mathbf{v}_i \times {}^*I_i^i\mathbf{v}_i \\ 11: & \quad C_i = {}^iS_i^T {}^i\mathbf{f}_i \end{aligned} \quad (5.88)$$

### 5.3.2 组合刚体法求 $M$ 矩阵

在公式(5.76)中,  $M(\mathbf{q})$ 是一个18行18列的对称正定矩阵, 其每一行都和关节扭矩 $\boldsymbol{\tau}$ 的每一行对应, 每一列都和关节空间加速度 $\ddot{\mathbf{q}}$ 的每一行对应, 设 $M_{ij}$ 为 $M$ 的分块矩阵, 对应了 $\boldsymbol{\tau}$ 中的第 $i$ 个关节和 $\ddot{\mathbf{q}}$ 中的第 $j$ 个关节。显然, 若 $i=1$ , 则矩阵 $M_{ij}$ 有6行, 否则 $M_{ij}$ 只有1行; 若 $j=1$ , 则 $M_{ij}$ 有6列, 否则 $M_{ij}$ 只有1列。

系统的总动能是系统中每个刚体动能之和, 因此:

$$T = \frac{1}{2} \sum_{k=1}^N \mathbf{v}_k^T I_k \mathbf{v}_k \quad (5.89)$$

将刚体 $k$ 的非递推空间速度公式(5.59)代入(5.89):

$$\begin{aligned} T &= \frac{1}{2} \sum_{k=1}^N \left( \sum_{i \in \kappa(k)} S_i \dot{\mathbf{q}}_i \right)^T I_k \left( \sum_{j \in \kappa(k)} S_j \dot{\mathbf{q}}_j \right) \\ &= \frac{1}{2} \sum_{k=1}^N \sum_{i \in \kappa(k)} \sum_{j \in \kappa(k)} \dot{\mathbf{q}}_i^T S_i^T I_k S_j \dot{\mathbf{q}}_j \\ &= \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \sum_{k \in v(i) \cap v(j)} \dot{\mathbf{q}}_i^T S_i^T I_k S_j \dot{\mathbf{q}}_j \end{aligned} \quad (5.90)$$

其中 $v(i) \cap v(j)$ 是集合 $v(i)$ 与集合 $v(j)$ 的交集,  $k \in v(i) \cap v(j)$ 意味着刚体 $k$ 同时由关节 $i$ 和关节 $j$ 支撑。不难想到, 对于串联机器人, 若刚体 $k$ 同时由关节 $i$ 和关节 $j$ 支撑, 要么刚体 $i$ 由关节 $j$ 支撑, 要么刚体 $j$ 由关节 $i$ 支撑, 不可能出现关节 $i$ 和关节 $j$ 在两条分支上, 又同时支撑刚体 $k$ , 因此:

$$v(i) \cap v(j) = \begin{cases} v(i) & , i \in v(j) \\ v(j) & , j \in v(i) \\ \emptyset & , \text{其他} \end{cases} \quad (5.91)$$

另一方面，系统的动能在关节空间可表示为：

$$T = \frac{1}{2} \dot{\mathbf{q}}^T \mathbf{M} \dot{\mathbf{q}} = \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \dot{\mathbf{q}}_i^T M_{ij} \dot{\mathbf{q}}_j \quad (5.92)$$

比较公式(5.90)与(5.92)，可得 $M_{ij}$ 的求解公式：

$$M_{ij} = \sum_{k \in v(i) \cap v(j)} S_i^T I_k S_j \quad (5.93)$$

将运动学树上以刚体 $i$ 为根节点的子树取出来，组合成一个刚体，称为组合刚体 $i$ ，该组合刚体的空间惯量为：

$$I_i^c = \sum_{j \in v(i)} I_j \quad (5.94)$$

公式(5.94)也可以写成递推形式：

$$I_i^c = I_i + \sum_{j \in \mu(i)} I_j^c \quad (5.95)$$

利用递推公式(5.95)，求解坐标系 $\mathcal{B}_i$ 中组合刚体 $i$ 的空间惯量 ${}^i I_i^c$ 的伪代码为：

```

for  $i = N$  to 1 do
   ${}^i I_i^c = I_i$ 
  for each  $j$  in  $\mu(i)$  do
     ${}^i I_i^c = {}^i I_i^c + {}^i X_j^* {}^j I_j^c {}^i X_j$ 
  end
end

```

其中 ${}^i X_j^* {}^j I_j^c {}^i X_j$ 用于将 ${}^j I_j^c$ 从坐标系 $\mathcal{B}_j$ 转换到坐标系 $\mathcal{B}_i$ 。利用组合刚体这一概念， $M_{ij}$ 的求解公式(5.93)可推演为：

$$M_{ij} = \begin{cases} S_i^T I_i^c S_j, & i \in v(j) \\ S_i^T I_j^c S_j, & j \in v(i) \\ \mathbf{0}, & \text{其他} \end{cases} \quad (5.97)$$

由于 $M$ 是对称矩阵，因此 $M_{ij} = M_{ji}^T$ ；将公式(5.97)中 $S_i^T$ 、 $I_i^c$ 、 $S_j$ 都转换到坐标系 $\mathcal{B}_i$ 中描述，公式(5.97)可推演为：

$$M_{ij} = \begin{cases} {}^i S_i^T I_i^c {}^i S_j = {}^i S_i^T I_i^c {}^i X_j^j S_j, & i \in v(j) \\ M_{ji}^T, & j \in v(i) \\ \mathbf{0}, & \text{其他} \end{cases} \quad (5.98)$$

根据公式(5.98)，可得组合刚体法求浮动基体动力学方程 $M$ 矩阵的算法为：

$$\begin{aligned}
& M = \mathbf{0}_{18 \times 18} \\
& \text{for } i = N \text{ to } 1 \text{ do} \\
& \quad M_{ii} = {}^i S_i^T I_i^{ci} S_i \\
& \quad j = i \\
& \quad \text{while } \lambda(j) > 0 \text{ do} \\
& \quad \quad {}^i X_{\lambda(j)} = {}^i X_j^j X_{\lambda(j)} \\
& \quad \quad M_{i\lambda(j)} = {}^i S_i^T I_i^{ci} X_{\lambda(j)}^{\lambda(j)} S_{\lambda(j)} \\
& \quad \quad M_{\lambda(j)i} = M_{i\lambda(j)}^T \\
& \quad \quad j = \lambda(j) \\
& \quad \text{end} \\
& \text{end}
\end{aligned} \tag{5.99}$$

## 5.4 本章小节

本章主要介绍了多刚体系统的动力学算法。首先介绍了六维空间向量及其性质，给出了空间速度、空间加速度、空间力、空间惯量等的定义，以及作用在这些空间向量上的运算，比如坐标变换、叉乘、运动方程等；随后介绍了多刚体系统的运动学建模，将四足机器人建模为运动学树，并给出了递推运动学算法；然后通过分析运动学，给出了全身控制雅可比矩阵  $J_1$ 、 $J_4$  及各级任务  $J_i \dot{q}$  项的求解算法；最后分析了多刚体系统的动力学，利用递推牛顿欧拉法求  $C$  矩阵，利用组合刚体法求  $M$  矩阵。

## 第六章 控制系统软硬件设计及仿真

实践是检验真理的唯一标准。本文的前五章，简略介绍了足式机器人控制系统的研究历史，详细介绍了本文所述四足机器人控制系统算法。本章将详细介绍这些理论的具体工程实现，主要包括控制系统硬件、控制系统软件以及相应的仿真与测试。

### 6.1 控制系统硬件设计

控制系统硬件是控制系统软件的物理载体，选择合适的硬件是实现稳健可靠的控制系统的前提，本节将介绍控制系统硬件的电气连接方案，以及所涉及的各种硬件设备。

#### 6.1.1 电气连接方案

四足机器人电气连接方案如图48所示。

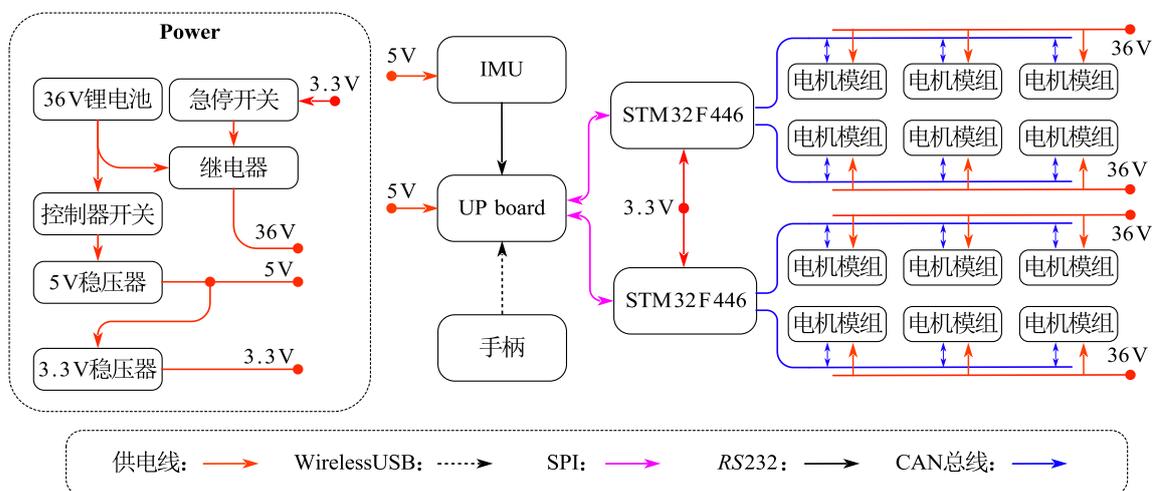


图48 四足机器人电气连接图

图48中红色导线为供电线，虚线为无线USB，紫色为SPI通信线，黑色为RS232串口通信线，蓝色为CAN总线。箭头指示了供电的方向或者信号的方向：对于信号线，双向箭头为双向通信，单向箭头为单向通信；对于供电线，箭头所指器件为被供电器件。

整个系统由一块36V锂电池供电。电流从控制器开关流出后，经5V稳压器稳压成5V电流，用于给UP board和IMU供电。5V电流再经过3.3V稳压器稳压成3.3V电流，用于给两块STM32F446供电。另一路36V电流流入大功率继电器，继电器导通信号为3.3V高电平信号，该信号由一个急停开关控制。从继电器流出的36V电流分别为12个电机模组供电。采用控制器开关与急停开关同时控制系统供电，比单个开关控制更加灵活：一方面，在调试初期测试通信回路时，不需要电机模组供电，可以断开急停开关，以免发生危险；另一方面，当机器人行走过程中遇到危险时，直接拍下急停开关，此时其他控制单元如UP board和STM32F446都还继续运行，可以及时查看系统日志排查问题。

由于电机模组的通信信号为 CAN 总线，而 UP board 没有 CAN 总线通信接口，若使用具有 CAN 总线接口的 STM32F446 作为主控制器，则又无法承担繁重的模型预测控制计算任务，因此采用 UP board 为主计算单元，两块 STM32F446 作为通信中继器，UP board 与 STM32F446 之间采用 SPI 通信，STM32F446 与电机模组之间采用 CAN 总线通信。

### 6.1.2 主要硬件设备介绍

四足机器人控制系统硬件设备主要包含 IMU、UP board、STM32F446、手柄、电机模组等，本小节将对这些硬件设备作简要介绍。

#### (1) IMU

本工程采用了美国 MicroStrain 公司生产的 3DM-GX3-25 型号航姿参考系统(AHRS)，图49展示了 3DM-GX3-25 的外观及尺寸。



图49 3DM-GX3-25 外观及长度

3DM-GX3-25 支持多种通信协议，如 USB、RS232、RS422 等方式，本文采用了 RS232 方法，信号输出频率可达 1kHz。内置两块用于自适应卡尔曼滤波的处理器，具有偏差跟踪、误差估计、自适应噪声建模等丰富的功能。它具有三轴加速度计、陀螺仪、磁强计和温度传感器，以实现测量的最佳组合，其中加速度计精度 0.02mg，陀螺仪精度 <math><0.003^\circ/\text{sec}</math>，姿态信号精度 <math><0.01^\circ</math>，重复定位精度 <math>0.2^\circ</math>，满足四足机器人控制要求。

#### (2) UP board

UP board 是四足机器人的主控制器，是前文几乎所有控制系统算法的载体。搭载了最高频率 1.92GHz 的 64 位英特尔®凌动™X5 Z8350 处理器，配备了 4GB 内存与 64GB 硬盘。如图 50 所示，UP board 是一个仅有信用卡大小的板卡电脑，板上集成了完全兼容树莓派的 40 针 GPIO 总线，有 4 个 USB2.0 端口、1 个 USB3.0OTG、1 个千兆以太网端口、1 个 HDMI 端口。其供电电压为 5V，最大需要 4A 供电电流。UP board 支持 Win10、安卓、Linux 等多种操作系统，本文所用操作系统为 Ubuntu16.04。

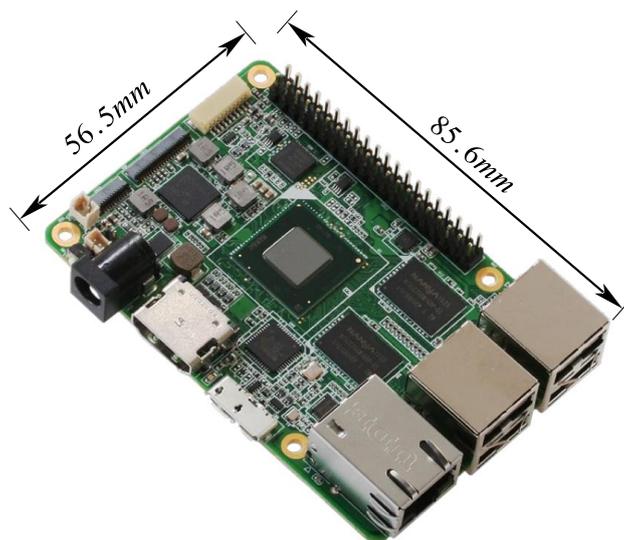


图50 UP board 外形及长宽尺寸

### (3) 手柄

为方便调试与控制，使用了图51所示的 Logitech F710 型号无线手柄。



图51 Logitech F710 无线手柄

Logitech F710 与UP board之间使用 2.4GHz 无线 USB 通信。Logitech F710 上有 11 个按键，6 个摇杆，其中按键 LB 用于起立，按键 RB 用于下蹲，按键 X 用于对角步态行走，按键 Y 用于平衡站立，按键 B 用于停止行走，左摇杆用于前后左右运动，LT 与 RT 摇杆用于偏航方向运动。

### (4) STM32F446

STM32F446 系列单片机是意法半导体推出的高度集成的节能产品，具有高性能和丰富的连接性，它利用 ST 的专有 ART Accelerator、智能架构、先进的 Flash 技术及其嵌入式 ARM Cortex-M4 内核。主频可达 180MHz，功耗仅仅 200  $\mu$ A/MHz。如图52所示，本文所用单片机具体型号为STM32F446RCT6，具有 64 个 GPIO，256kb 闪存容量。使用一路 SPI 与UP board 通信，两路 CAN 与电机模组通信。为保证一定的通信带宽，使用了

两块单片机同时驱动 12 个电机模组，每路 CAN 总线控制一条腿。



图52 STM32F446RCT6

### (5) 电机模组

传统伺服电机将电机本体、减速器、驱动器、编码器分离，尽管在使用时有一定的灵活性，但是对四足机器人来说，这种电机组合形式集成度低、成本高、质量密度低。鹏行智能团队专门针对 18kg 级四足机器人设计了高集成度电机模组，使用扁平式外转子 BLDC 电机，将 1:6 减速比的行星减速器集成到磁圈内部，在不增加轴向尺寸的情况下实现一定的减速能力。采用双磁编码器方案，在减速器输入端设置 14 位磁编码器，在输出端设置霍尔传感器，能实现在线多圈绝对式定位，掉电单圈绝对式定位。电机模组上集成了驱动电路，避免了驱动电路与电机不匹配的麻烦。经过不同厂家之间对比后，最终选取了图53所示的 ME19-A 型号电机模组。如图54，其额定扭矩 18Nm，峰值扭矩 48Nm，峰值扭矩 350RPM，质量 730g，工作电压 36V，完全满足四足机器人使用要求。



图53 ME19-A 电机模组

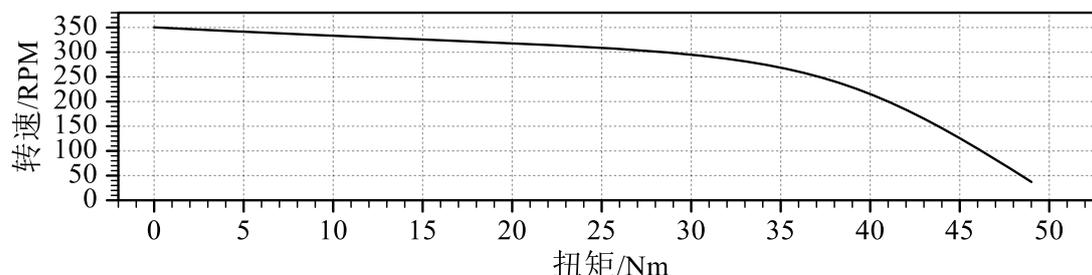


图54 ME19-A 电机模组最大功率时转速转矩特性曲线（厂家提供）

## 6.2 控制系统软件设计

控制系统软件是将控制系统算法用合适的编程语言实现，并编译成计算机可运行的可执行文件。搭建控制系统软件，基本要求是实现控制系统算法，关键是与不同硬件的交互，要充分考虑硬件上的计算能力、通信能力及易用性。

图55为控制系统函数调用图，控制系统程序从 `main` 函数开始运行，首先初始化系统的各种状态信息，包含参数初始化、硬件初始化、MPC 初始化、WBC 初始化、步态初始化；然后运行控制器，控制器包含状态估计、轨迹规划和状态机三个部分；其中状态估计首先使用 SPI 与下位机通讯，获得机器人传感器信息，然后利用卡尔曼滤波器估算机器人状态；轨迹规划器首先接收手柄指令，然后根据手柄指令计算质心轨迹；状态机用于调度机器人的各种具体行为，包括对角步态行走、起立、停止、下蹲、平衡站立、阻尼状态；对角步态行走状态中，依次计算步态调度器、MPC、WBC 以及刚体动力学算法，然后将计算出来的扭矩作为关节 PD 控制器的前馈；状态机中平衡站立状态也涉及 WBC 计算和刚体动力学算法；状态机其余状态直接将规划好的期望关节速度和期望关节角度传递给关节 PD 控制器中；关节 PD 控制器整合上游指令，将指令编码后利用 SPI 发送给各个电机，从而完成整个机器人控制流程。

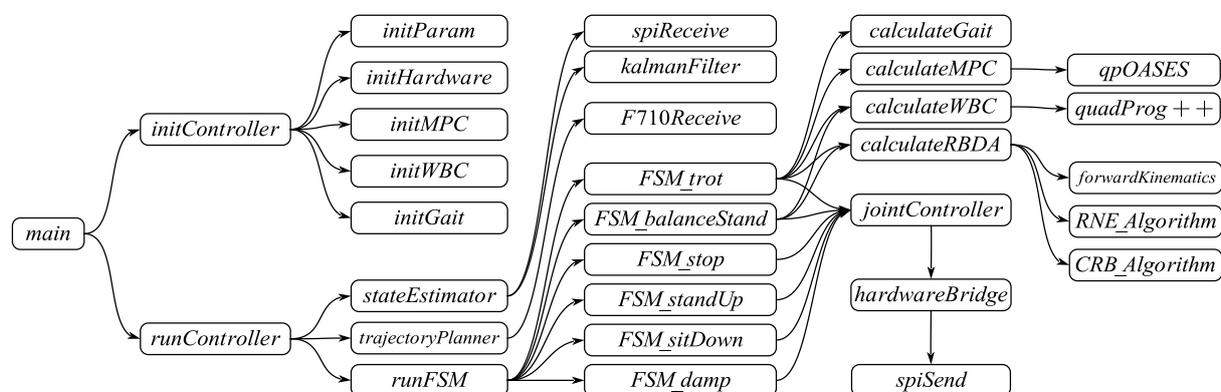


图55 控制系统函数调用图

### 6.2.1 控制系统软件各模块介绍

模块化设计是实现大型工程的必然途径，将一个大的任务分解成一个个独立的小模块，尽可能减少模块与模块之间的耦合性，既方便调试，又方便在不同工程之间移植。如图7所示，四足机器人控制系统被分成不同的功能模型，本小节将逐个介绍这些模块。

基本数学计算模块。尽管 C++ 语言中已经集成了标准的数学库 `math.h`，但是针对四足机器人控制系统，依然有诸多基本数学算法需要补充。点到点的轨迹规划插值算法：采用第二章公式(2.95)最优的点到点轨迹公式，传入参数为起始点坐标、终点坐标、总时

间以及当前时间，传出参数为轨迹在当前时刻的位置、速度、加速度。多种刚体旋转参数化方法的互相转换算法：刚体的旋转可以用旋转矩阵、四元数、欧拉角、轴角法等不同的方法来描述，尽管他们表示了相同的含义，但在不同应用场景下有不同的易用性，因此经常需要相互转换。反对称矩阵算法：实现了将三维向量转换成对应的 3 行 3 列反对称矩阵的算法，主要应用于角速度矢量与角速度张量的互相转换。

状态估计模块。机器人控制系统是用来解决机器人在哪儿、去哪儿、怎么去的问题，状态估计用于解决机器人在哪儿的问题。状态估计器的传入参数为传感器的原始数据，如编码器传回的各个关节角度、角速度，IMU 传回的躯干旋转矩阵、角速度、加速度。传出参数为标准化后的传感器数据，以及躯干在世界系下的位置与速度。由于状态估计模块是是对传感器信号的处理，其输出参数是后续控制器的输出参数，因此在每一个控制周期，取得传感器数据后立即运行状态估计算法。状态估计模块的核心是基于足底不打滑假设的足底里程计，其估计算法采用了公式(2.57)所介绍的卡尔曼滤波器。

状态机模块。状态机用于调度机器人的不同行为。在手柄或其他上层控制器的控制下，机器人将表现出不同的行为，如行走、下蹲、起立等动作。机器人不能完全听从操作者的指令去执行动作，比如在下蹲过程中，若机器人收到手柄传来的行走指令，是不能直接行走的，否则会出现危险。因此，机器人在接受指令和执行指令之间需要一个“思考”指令的过程。一个状态是否可以切换到另一个状态，被提前编码记录，采用 *switch* 语句进行不同状态间的调度。另外，有些状态也存在“忙碌”状态，比如在下蹲过程中若收到起立指令，二则发生冲突则继续下蹲，但在下蹲动作完成后再收到起立动作指令，则能正常运行起立动作，这说明在下蹲的过程中，下蹲状态处于“忙碌”状态，不再接受其他指令。若机器人出现明显的危险动作，比如足底速度过大、关节跟踪误差过大、机器人发生倾倒等等异常现象，机器人将立即主动进入阻尼保护状态，不再执行任何控制算法，直到操作者对系统重新启动。

步态调度模块。本文采用步态这一概念来描述支撑腿与摆动腿的周期性切换行为。支撑腿与摆动腿的周期性切换，主要有基于时间的方法和基于事件的方法两大类。由于四足机器人上未装足底传感器，因此采用了基于时间的占空比-偏移量步态描述法。在控制系统中有专门记录系统时间的计时器，步态调度模块首先查询系统时间，计算当前处于第几个迈步周期，然后根据迈步周期长度计算当前迈步周期的时间进度，再结合当前步态的占空比-偏移量步态描述，算得每条腿是处于支持相还是摆动相，以供后续控制器选择不同的控制算法。

模型预测控制器模块。作为控制系统的核心算法之一，模型预测控制算法占据了控制算法的大部分计算量，为了防止出现大的计算延迟，模型预测控制独占系统一块 CPU。为了实现模型预测控制器独占 CPU，首先将模型预测控制算法独立出一个线程，然后配置该线程的 CPU 亲和性，使操作系统在调度该线程时只选择 CPU0；同时，让其他所有线程的亲和性配置在 CPU1~CPU3，具体 CPU 的算力分配将在下一小节介绍。模型预测控制模块的输入参数为质心轨迹生成器传来的质心轨迹以及状态估计器传来的系统状态，输出参数为支撑足足底力。在模型预测控制模块中，集成了一个基于单刚体动力学状态方程的预测方程，将状态估计器传来的系统状态作为预测方程的初始值，将一个未知的控制量序列作为控制量，预测未来一段时间的状态。随后将预测的状态与期望质心轨迹对比做差，放入带约束的凸优化控制器中解算出使期望状态与预测状态差距最小的控制量序列，然后将控制量序列的第一组控制量作为当前时刻的支撑足足底力。

多刚体动力学算法模块。传统的刚体动力学算法将刚体的运动和受力分为平动和转动两个方面，使用两套动力学算法来分别求解。但是刚体在空间有六个自由度，平动与转动之间的耦合关系体现在刚体动力学算法的反方方面面，导致传统的刚体动力学算法既难以使用又容易出错。本文在文献<sup>[31]</sup>所提出算法的基础上，在 Plücker 坐标系下使用六维空间矢量，推导了专门适用于四足机器人的多刚体动力学算法，并用 C++ 实现，封装成刚体动力学算法库。该算法库实现了四足机器人的递推正运动学、递推牛顿欧拉法、组合刚体法等浮动基座动力学所必须的重要算法，为全身控制器提供了底层算法支撑。

全身控制器模块。为了解决模型预测控制计算量太大，只能使用单刚体模型的缺点，采用了基于多刚体动力学的全身控制器。全身控制本质上是一种工作空间 PD 控制器，它将总体的任务按照重要程度分成若干个小任务，利用各个任务雅可比矩阵的零空间的性质，使低优先级的任务无法干扰高优先级任务的控制，实现带优先级的逆运动学算法，再将逆运动学算法得到的关节空间加速度传给多刚体逆动力学算法，最终得到关节扭矩作为控制量。本文中四足机器人全身控制器将总体的控制任务分为四个子任务，优先级从高到低分别是支撑足足底控制、躯干转动控制、躯干平动控制，摆动足足底控制。支撑足与地面不打滑是状态估计器与控制器正常执行的前提，因此支撑足控制任务列在第一优先级，但是第一任务中并不存在 PD 控制器，因此第一任务是纯粹的运动学逆解。第二、三任务都是躯干控制任务，在行走过程中躯干保持平稳，是四足机器人整体的控制系统最终的控制目标，因此亦为重要。在机械结构设计环节，已经尽可能减小腿部质量占总质量的比重，因此摆动腿的摆动动作并不会对躯干造成大的扰动；另一方面，通

过落足点调整机器人水平加速度的控制方法对落足点的精度有较大的容忍度，因此在整个机器人控制环节，摆动腿的控制处于最低优先级。全身控制的输入参数为各级任务的实际位置、实际速度、期望位置、期望速度，与模型预测控制输出的支撑足足底力做松弛优化后，最终输出的是 12 个关节扭矩。

硬件接口模块与仿真接口模块。为了将控制系统与被控系统独立起来，搭建了硬件交互层，既硬件接口模块。硬件接口模块用于控制系统软件与控制系统硬件数据交互，包括从手柄、编码器、IMU 中读取数据以及给电机发送扭矩指令。为了让仿真效果尽可能与实物效果接近，实现仿真“所见即所得”，在控制器与仿真环境之间搭建仿真接口模块。仿真接口模块与硬件接口模块交互的数据完全一致，区别仅在于硬件接口接收到的编码器与 IMU 数据来自真实硬件，发送给电机的扭矩也是发给了真实电机，而仿真接口模块收到的传感器数据来自仿真环境，发送的电机扭矩是发给了仿真环境中的电机。

尽管控制系统中还有其他重要功能，如地形估计、质心轨迹规划、落足点规划等必要算法，但这些算法代码规模较小，不值得设立一个单独的模块，因此都被集中到了模型预测控制模块中。

## 6.2.2 第三方库

### (1) Eigen

Eigen 是一个基于 C++ 线性代数计算库，用于矩阵与向量计算。Eigen 支持所有尺寸的矩阵、动态尺寸的矩阵以及稀疏矩阵。支持整形、复数、浮点型和双精度浮点型等多种数据类型，支持各种矩阵分解算法。Eigen 支持多种硬件平台和编译器，图 56 展示了使用 Eigen 求矩阵转置的例子。

Example:	Output:
<pre>MatrixXf a(2,3); a &lt;&lt; 1, 2, 3, 4, 5, 6; cout &lt;&lt; "Here is the initial matrix a:\n" &lt;&lt; a &lt;&lt; endl;  a.transposeInPlace(); cout &lt;&lt; "and after being transposed:\n" &lt;&lt; a &lt;&lt; endl;</pre>	<pre>Here is the initial matrix a: 1 2 3 4 5 6 and after being transposed: 1 4 2 5 3 6</pre>

图 56 使用 Eigen 求矩阵转置

### (2) qpOASES

qpOASES 是一个基于 C++ 的二次型凸二次规划库。公式(6.1)为 qpOASES 所能求解的二次规划问题的数学形式，其中  $H$  矩阵是对称的正定或半正定矩阵。qpOASES 使

用了在线活动集策略：假设连续使用 qpOASES 求解多个优化问题时，其最优输出不会有太大变化，在这一启发式前提下，qpOASES 可以快速得求解连续的二次规划问题。使用在线活动集策略使得 qpOASES 非常适合模型预测控制应用。在实际使用中，qpOASES 用于求解公式(3.45)所示的优化问题。

$$\begin{aligned} \min_x \quad & \frac{1}{2} x^T H x + x^T g \\ \text{s.t.} \quad & lb_A \leq Ax \leq ub_A \\ & lb \leq x \leq ub \end{aligned} \quad (6.1)$$

### (3) QuadProg++

与 qpOASES 类似，QuadProg++是另一个基于 C++的二次型凸二次规划库，它在解决小规模二次规划问题上有极高的效率。在实际使用中，QuadProg++用于求解公式(4.58)所示的优化问题。实际测试发现，在公式(4.58)这种规模等级的二次规划问题上，QuadProg++的计算效率几乎是 qpOASES 的十倍。图57展示了 QuadProg++官方说明文档及其所能求解的二次规划问题的数学形式。

The quadprog\_solve() function implements the algorithm of Goldfarb and Idnani for the solution of a (convex) Quadratic Programming problem by means of an active-set dual method.

The problem is in the form:

$$\begin{aligned} \min \quad & 0.5 * x^T G x + g_0^T x \\ \text{s.t.} \quad & C^T x + ce_0 = 0 \\ & C^T x + ci_0 \geq 0 \end{aligned}$$

图57 QuadProg++的官方说明及其标准二次规划数学形式

### (4) LCM

Lightweight Communication and Marshalling (LCM)是一组用于消息传递和数据编组的库与工具，专门为高带宽低延时的实时系统设计。它为多种编程语言的应用程序提供了发布/订阅消息机制，以及自动编组/反编组代码生成。在四足机器人应用上，LCM 主要有两方面的功能。一方面，仿真环境与控制系统分属两个进程，C++不支持进程间的直接通信，因此使用 LCM 在仿真环境端与控制系统端分别设置发布/订阅接口，两端通信。另一方面，在实物机器人运行时，可使用 LCM 在机器人 Linux 主机和笔记本 Linux 主机之间通信，从而实现机器人状态实时回传，以及实现通过笔记本在正在运行的机器人上在线修改参数。如图58所示，使用机器人状态实时回传功能，在仿真环境下可实时观察并记录实物机器人的状态。

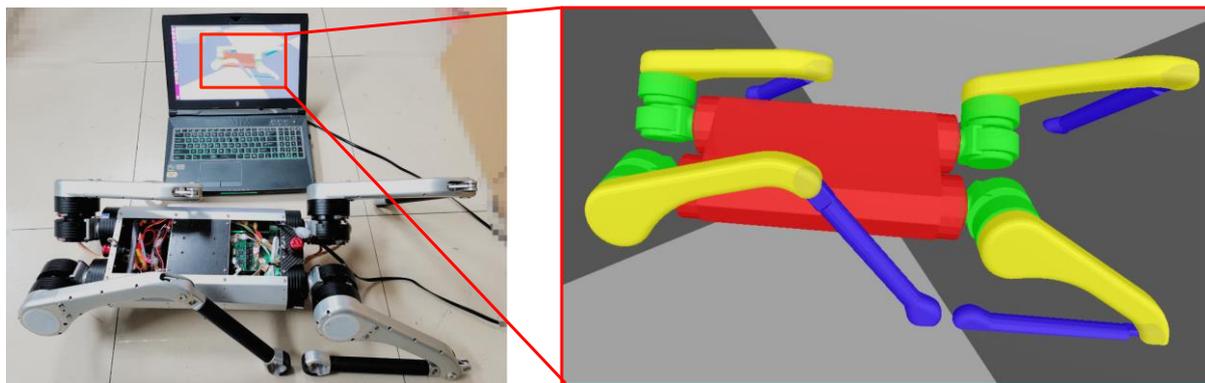


图58 使用 LCM 实现机器人状态实时回传

### 6.2.3 CPU 算力分配

控制系统在 UP board 上运行时，主要有 4 个进程，包括主控制环进程，MPC 计算进程，SPI 通信进程和陀螺仪通信进程。其中主控制环进程中运行了手柄驱动程序、地形估计、质心轨迹规划、足底轨迹、全身控制等大部分算法；MPC 计算进程专门用于求解公式(3.45)所示的优化问题；SPI 通信进程负责 UP board 与 STM32F446 的 SPI 通信，陀螺仪通信进程负责 UP board 与陀螺仪的通信工作。由于 UP board 的算力在 MPC 计算时依然紧张，如图59所示，设计了进程监视器，用于监视各种进程的超时情况。进程监视器以 2Hz 的频率将各个进程运行状况打印出来，并将超时的进程标红。

		TASKS					
		rt	rt-max	T-des	T-act	T-max	
	print-tasks	0.161	0.161	500.000	499.999	499.999	0
主控制环进程	robot-control	0.294	0.335	2.000	2.004	2.629	120
SPI通讯进程	spi	1.317	1.635	2.000	2.001	2.006	120
陀螺仪通信进程	IMU	0.001	0.004	2.000	2.009	2.213	120
MPC计算进程	mpc	0.001	0.001	10.000	10.002	10.004	24
	vmcTraj	0.001	0.001	50.000	50.001	50.002	4
	lcm-vis	0.032	0.056	16.700	16.616	17.276	16

图59 进程监视器

为了合理利用 UP board 的计算资源，采用 CPU 亲和性技术，将四个线程分别绑定到 UP board 的四个 CPU 上。经过实际测试发现，计算一次 MPC 大约需要 7.5ms，为保证一定的算力余量，设计 MPC 计算的频率为 100Hz；计算一次主控制器环约需 1.5ms，设计主控制器环的运行频率为 500Hz；至于 SPI 通信与陀螺仪通信所占用的时间，与前两个线程相比可以忽略不计。因此，将四个线程分别绑定到四个 CPU 上后，CPU 占有率分别为 75%、75%、13%、13%，UP board 的算力满足使用要求，四个 CPU 工作时序图如图60所示。

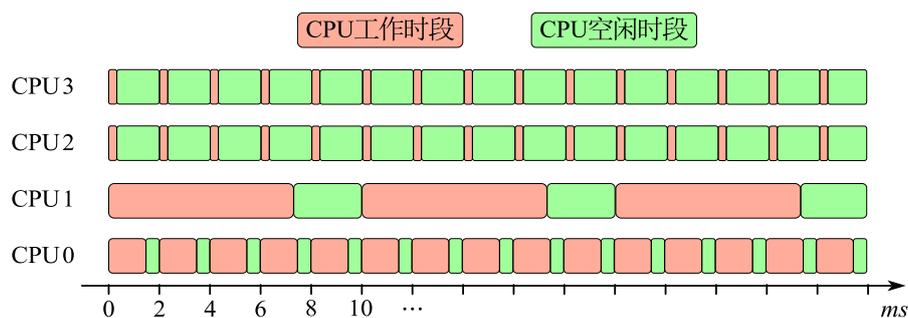


图60 CPU工作时序图

### 6.3 控制系统仿真实验验证

尽管已经在控制系统算法中从各方面保证了控制系统的可靠性，但在实际控制物理样机之前，还需要若干实验来进一步验证系统的可靠性。

#### 6.3.1 webots 仿真平台介绍

webots 是专门用于机器人仿真的开源跨平台仿真软件。它提供了快速搭建物理原型的环境，使用户可以创建具有物理特性（如质量、关节、摩擦等）的 3D 虚拟世界。在 webots 中有许多机器人上常见的传感器与执行器，如距离传感器、摄像头、激光雷达、陀螺仪、加速度计、转动电机、直线电机等等。在 webots 上，所有机器人都是可编程的，可以使用 C、C++、Java、python、matlab 等多种语言来实现机器人的仿真控制，可以方便地利用库函数接收 webots 中机器人上的传感器数据，并控制机器人上的执行器。webot 内置了接近 100 种机器人模型，包括轮式机器人、人形机器人、爬行移动机器人、单臂移动机器人、双臂移动机器人、无人机、大狗、飞艇等等，其中包括常见的 Boston Dynamics Atlas、DJI Mavic 2 PRO、Nao、PR2、YouBot、UR、Turtlebot3 Burger 等机器人，如图 61 为部分 webots 内置的机器人模型。



图61 部分 webots 内置的机器人

#### 6.3.2 峰值关节扭矩实验

在前文关于硬件设备的介绍中，已经介绍了所选型电机模组的峰值扭矩为 48Nm，额定扭矩为 18Nm。为验证该电机满足四足机器人使用要求，在仿真环境下进行机器人行走测试，并打印关节扭矩。具体动作为：首先机器人以 0.5m/s 的速度前进，然后以

0.2m/s 的速度向右移动，然后以 0.5m/s 的速度后退。打印出来的关节扭矩、关节转速、关节转角如图62-64所示。

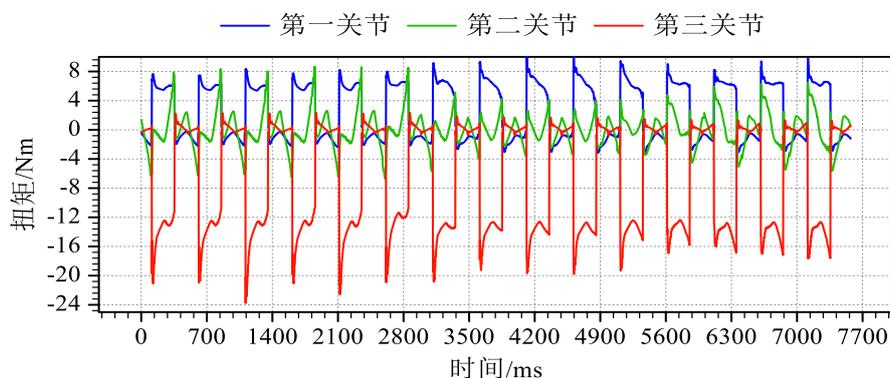


图62 右后腿三个关节的扭矩

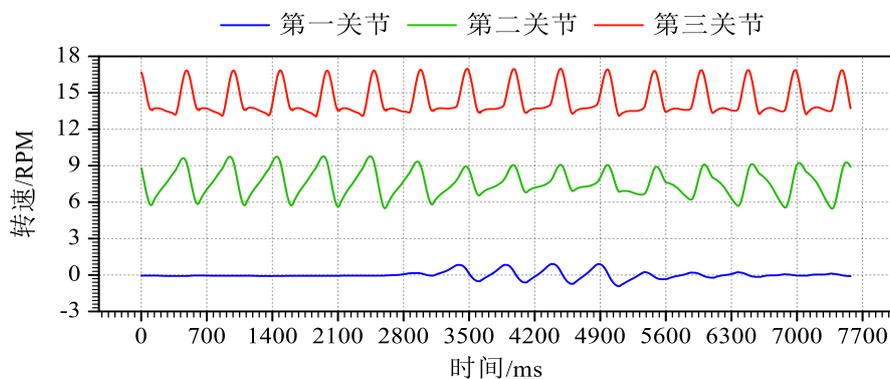


图63 右后腿三个关节的转速

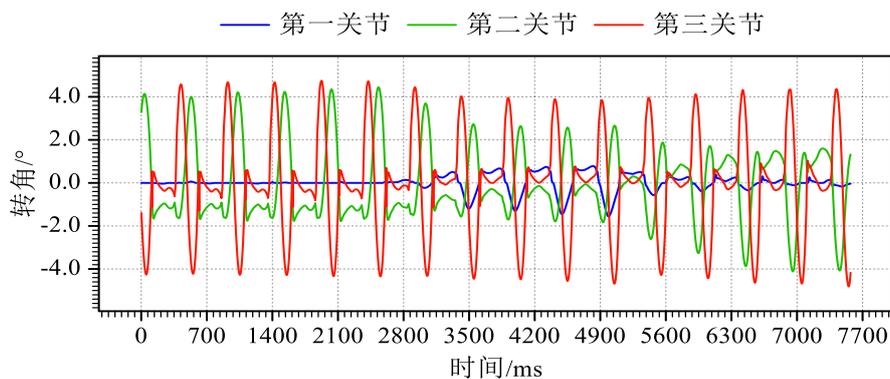


图64 右后腿三个关节的转角（减去均值后）

从图62中可以看出，第三关节扭矩最大，在大多数时间里，其扭矩不超过 16Nm，低于 ME19-A 电机模组的 18Nm 的额定扭矩；在一些特殊时刻关节扭矩会超过 18Nm，但也不会高于 25Nm，低于 ME19-A 电机模组的 48Nm 的峰值扭矩，因此所选电机模组扭矩完全满足四足机器人行走要求。

从图63中可见，第三关节转速普遍较大，但整个运动过程的转速不超过 17.5RPM，

参考图54可知所选电机模组转速完全满足四足机器人行走要求。

由于 ME19-A 电机模组没有运动范围限制，因此所选电机模组的转角也满足图64中四足机器人行走要求。表 3 中列举了仿真中电机的部分极限参数。

表 3 仿真过程中电机的部分极限参数

参数名称	值
最大扭矩绝对值	23.78Nm
最大转速绝对值	16.85RPM
最大转角	4.75°
最小转角	-4.81°
最大扭矩出现时的转速	13.65 RPM
最大转速出现时的扭矩	0.16Nm

### 6.3.3 对角步态踏步稳定性实验

对角步态是四足机器人行走最常见的步态，也是动步态中最稳定的步态。因此，当机器人原地踏步时，理论上对躯干的控制效果最为稳定。本小节将测试仿真环境下四足机器人原地踏步，并打印欧拉角，测试结果如图65所示。从图中可以看出，控制系统对躯干的旋转能实现一个较为稳定的控制效果。

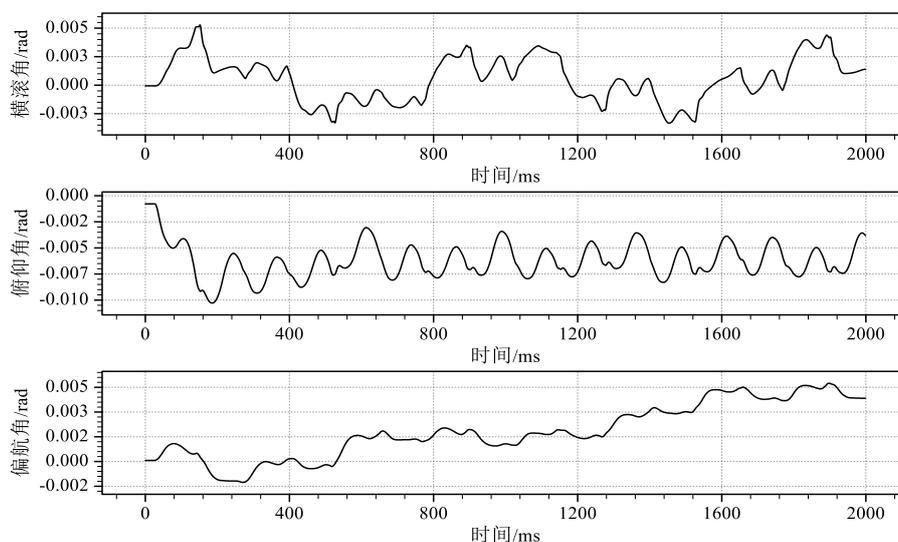


图65 原地踏步时欧拉角变化曲线

### 6.3.4 对角步态侧向冲击实验

在受到侧向冲击后恢复稳定的能力是四足机器人极为重要的能力，图66为四足机器人在对角踏步时，受到侧向 12kg 重物以 2m/s 撞击后欧拉角变化曲线图。

图67为四足机器人受侧向冲击后的时序图。在受到向右的冲击后，落足点右移，最终机器人可以停下来，这在一定程度上说明使用落足点来调整躯干水平速度是有效的。图下方的曲线为躯干角随时间变化曲线，躯干角定义为欧拉角的欧几里得范数，即：

$$\|\Theta\|_2 = \sqrt{\phi^2 + \theta^2 + \psi^2} \quad (6.2)$$

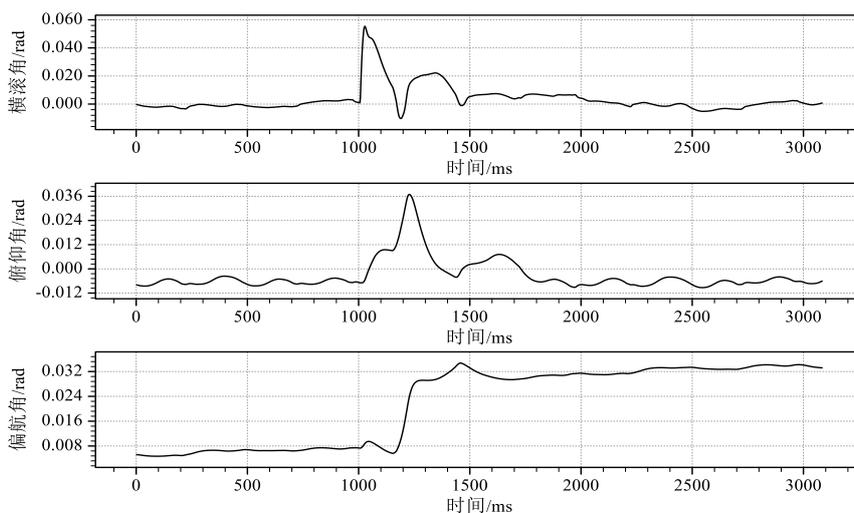


图66 侧向冲击后欧拉角变化曲线

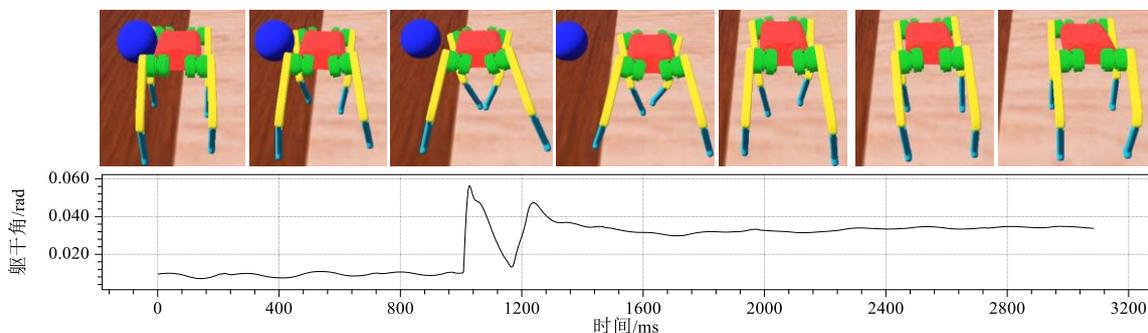


图67 对角步态 (trot) 受到侧向冲击后时序图

### 6.3.5 多种步态下侧向冲击实验

本文所述控制算法与具体的步态种类无关，只要某种步态存在一定合理性，理论上控制算法就能使其在四足机器人上实现。在仿真环境下，测试了奔跑步态、踱步步态、跳跃步态在受到侧向 12kg 重物以 2m/s 撞击后的表现，图68-70为时序图。

实验结果表明，在理想的仿真环境下，控制系统足以将不同步态下受到侧向冲击的四足机器人稳定下来。

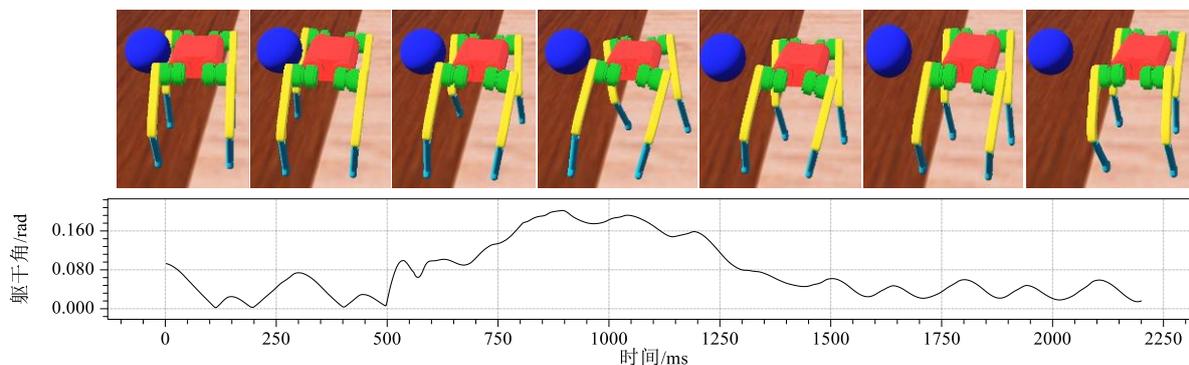


图68 奔跑步态 (bound) 受到侧向冲击后时序图

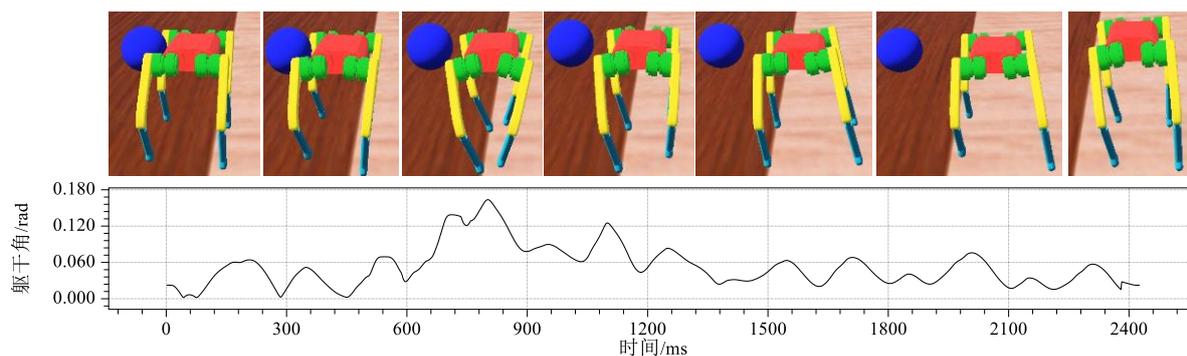


图69 踱步步态 (pace) 受到侧向冲击后时序图

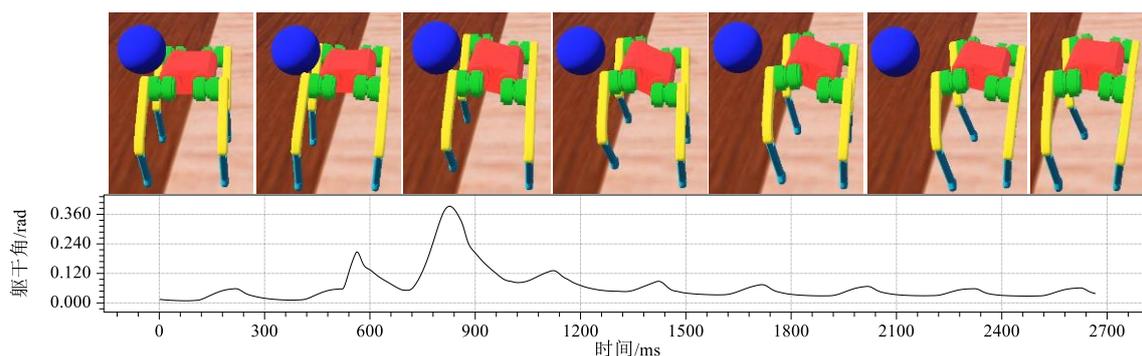


图70 跳跃步态 (pronk) 受到侧向冲击后时序图

## 6.4 本章小节

本章主要介绍了控制系统软硬件设计及其仿真。首先介绍了控制系统的硬件，包括硬件设备的电气连接方案，以及IMU、主控制器、手柄等硬件设备；然后介绍了控制系统软件，包括控制系统软件各个模块、第三方库以及CPU算力分配策略；最后介绍了控制系统的仿真验证，通过设计几个简单的测试，验证了控制系统的稳定性。

## 第七章 物理样机设计与测试

若说控制系统软件和硬件是机器人的灵魂与大脑，那么机械结构就是机器人的骨骼与肌肉。一台可靠的四足机器人固然离不开聪明的大脑，同样也需要一副强健的身躯。只有保证四足机器人本身硬件系统的可靠性，才能谈及控制系统的可靠性。假如机器人强度不够，在调试和正常运行过程中，就可能发生危险，轻则破坏机器，重则造成人员伤亡。本章将从腿型配置形式、传动方式、关键零部件设计及有限元仿真等方面详细介绍四足机器人的机械结构设计过程与仿真验证过程。最后，将控制系统应用到搭建完成的物理样机上，进行了若干实验，验证了物理样机具有极高的动态能力和稳定性。

### 7.1 物理样机结构设计

#### 7.1.1 腿型布置形式

四足机器人的腿型布置形式按照膝关节朝向可分为全肘式、内膝肘式、外膝肘式三种类型。图71展示了比较典型的几个四足机器人，其中 spot、laikago 和 A1 都是全肘式，ANYmal 为内膝肘式。

尽管有学者仿真分析发现内膝肘式四足机器人行走较为平稳<sup>[46]</sup>，但是他们的控制系统所用动力模型简化程度过高，控制系统也较为简单，说服力不足。麻省理工学院腿足机器人实验室的研究<sup>[12, 13]</sup>表明，其设计的控制系统能发挥良好效果的前提，是腿部质量占机器人总质量较小。对于使用了不考虑腿部质量的简化动力学模型的控制系统，腿部质量影响较大。而内膝肘式四足机器人四条腿的质量分布较为集中，距离机器人总质心较近，具有更小的转动惯量，因此腿部的移动对躯干的受力影响较小，有利于控制。然而，由于前后腿膝关节距离很近，内膝肘式四足机器人前后腿极易发生干涉。

外膝肘式四足机器人质量分布较为分散，尽管不易发生干涉，但是腿部移动更容易对躯干造成影响，难以控制。另外，由于外膝肘式四足机器人的膝关节皆冲外，在运动过程中容易碰到外界环境，不利于在狭窄空间中运动，也不易于包装运输。但是，外膝肘式四足机器人的躯干两侧不会受到腿的影响，在实际使用时可以用来驮货物。



图71 从左到右依次为 spot、laikago、A1、ANYmal

总的来说，外膝肘式与内膝肘式各有优缺点。本文采用了较为中庸的全肘式，避免了走上两个极端。但是，全肘式四足机器人腿部质量整体向后，从而导致机器人整机质心向后。在原地踏步时，需要额外调整落足点的偏移量，来保证机器人支撑中心与质心在地面上的投影重合，否则机器人会受重力影响不断后退。这是全肘式四足机器人特有的缺点，但可以从控制层面上轻易解决。

### 7.1.2 传动方式选型

在前文模型预测控制中采用了单刚体模型，因此需要腿部惯量较小，从而削弱因腿部移动导致的整机惯性张量变化。为了降低腿部惯性张量，将膝关节电机上移至第二关节处，使用机械结构将膝关节电机的动力传递给膝关节。

从膝关节电机到膝关节的传动形式通常有三种：同步带传动，链传动和平行四杆传动。

皮带是远距离传送动力的常见方式，可以不受传动距离限制，将动力源的动力传递到执行元件。而且皮带本身具有弹性，可以缓冲振动，提高系统的柔顺性。但是，皮带在传递较大扭矩时会打滑，尽管采取同步带传动，也难以实现本文中四足机器人所需的扭矩要求。经过校核后发现同步带传动所需皮带宽度较宽，设计之后结构不够紧凑，因此未采用皮带传动。

链传动需要的空间有所减小，但还是较为臃肿。链传动具有多边形效应，在不同的转动位置传动比不同，也不适合四足机器人这种角度敏感型应用场景。链传动通常适合自行车这种单向传动应用，在双向传动时，由于回程间隙较大，会引起较大的噪音。因此最终也未采用链传动。

结构可靠且简单的平行四杆传动已经在诸多四足机器人上得到验证，比如宇树科技的 laikago，浙江大学的绝影等。因此，如图72所示，最终选取了平行四杆结构作为从膝关节电机到膝关节的传动结构。



图72 平行四杆结构

### 7.1.3 躯干设计

躯干是机器人的受力主体，承载了大部分零部件与电子元器件，因此躯干需要保证一定的强度和刚度，同时还需要为机器人内部电路板预留出足够的空间。为提高机身刚度，减小机身质量，躯干整体采用框架式结构。使用四块镂空竖板保证左右方向和上下方向承力，四块竖板之间穿插6根碳纤维管，保证前后方向承力，竖板与碳纤维管有24处连接，保证了躯干其余三个旋转方向的承力。最后再加上外壳，再次提高整体强度，最终设计出的躯干结构如图73所示。

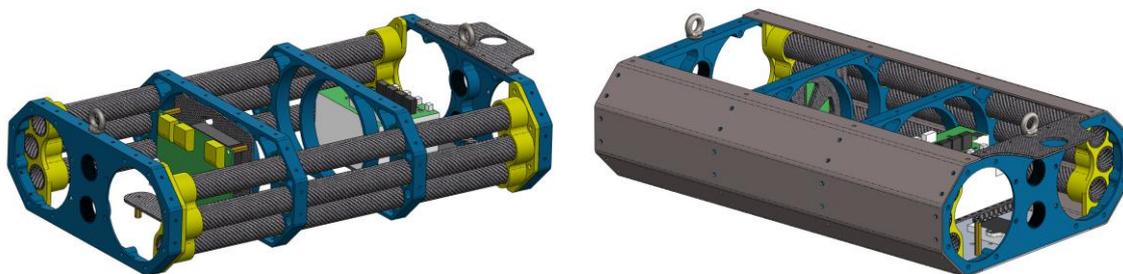


图73 躯干框架（左），躯干加壳（右）

### 7.1.4 足底结构设计

机器人的足底是机器人与外界环节唯一的接触位置。因此需要保证接触可靠，需要具有一定的强度、耐磨性、阻尼性以及与地面之间有较大的摩擦系数。

为了保证足够的强度，如图74足底零件内部采用了铝制结构，与小腿的碳纤维管之间使用两根螺纹连接。经过有限元仿真校核后，验证了该结构的可靠性。

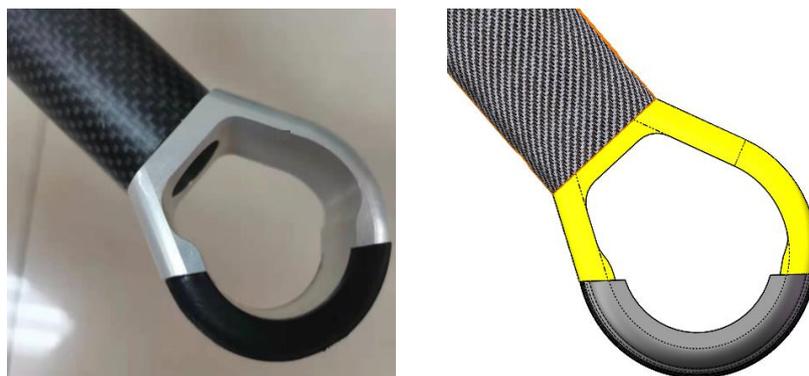


图74 足底实物结构（左）足底三维建模（右）

然而，经实际测试发现，铝制零件与实验室地面的摩擦系数仅0.05，在仿真环境下验证了该摩擦系数不足以满足机器人正常行走，因此足底与地面直接接触的材料不能是铝。在本项目中，选用了3D打印的软胶作为足底材料。如图75所示，足底材料经过两版迭代，最终保证了足底与地面的稳定可靠接触。足底软胶与足底内部铝制零件用胶水连接，在第二版足底软胶中设计了存胶槽，提高胶接的可靠性，防止在行走过程中脱落。

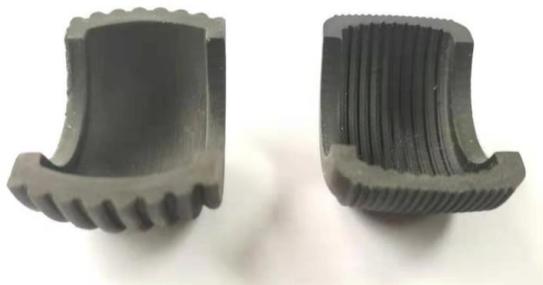


图75 初版足底软胶（左）终版足底软胶（右）

## 7.2 关键零部件仿真

### 7.2.1 电机座设计与仿真验证

如图76(a)所示，红色零件为电机座。电机座用于连接第一关节电机的转子和第二关节电机的定子。利用电机正反两面都有安装孔的优势，将电机座设计成叉架结构。

利用有限元仿真软件对电机座进行仿真，添加接触约束后，施加 48Nm 的载荷。仿真结果如图76(b)所示，最大应力为 105.28MPa，选用的材料为 7075 铝合金，其屈服强度可达 505MPa，完全满足使用要求。

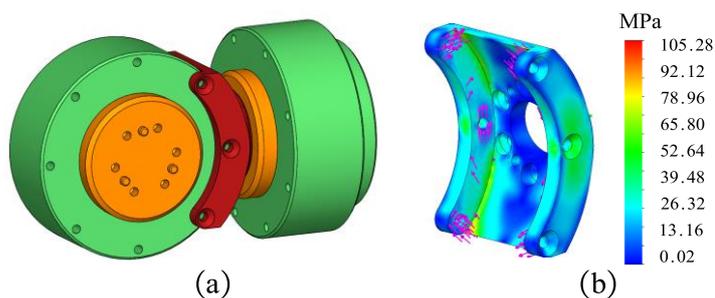


图76 电机座及其仿真

### 7.2.2 内曲柄与外曲柄仿真验证

如图77所示，将内曲柄与外曲柄用同样步骤进行仿真分析。

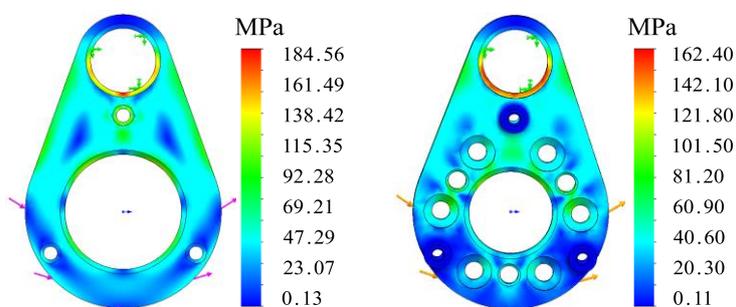


图77 内曲柄（左）与外曲柄（右）仿真验证

由仿真结果可见，在传递 48Nm 扭矩时，内曲柄最大应力为 184.56MPa，外曲柄最大应力为 162.40MPa，因此 7075 铝合金完全满足使用要求。

### 7.2.3 大腿壳仿真验证

如图78所示，为大腿内壳、大腿外壳、髌关节盖做联合有限元仿真。在传递 48Nm 扭矩时，最大应力为 114.24MPa，因此 7075 铝合金完全满足使用要求。

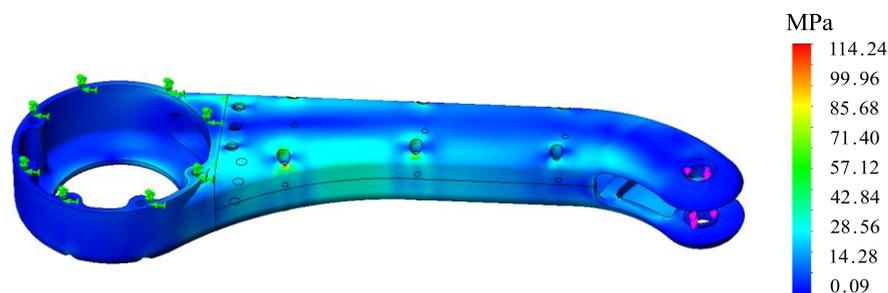


图78 大腿壳的有限元仿真

### 7.2.4 连杆仿真验证

如图79所示，为平行四杆机构中的连杆做有限元仿真。在传递 48Nm 扭矩时，最大应力为 71.44MPa，因此 7075 铝合金完全满足使用要求。

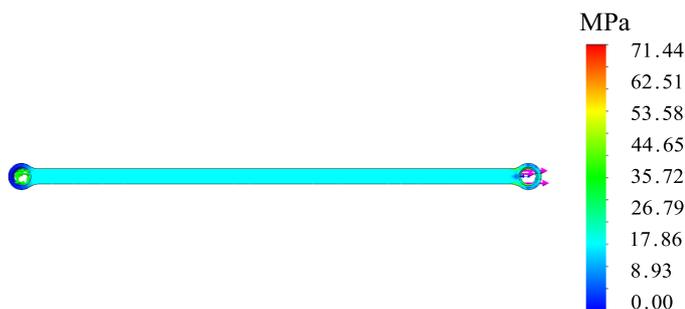


图79 连杆的有限元仿真

## 7.3 整体装配图

如图80所示，依照实际零部件的装配关系，按层次将所有零部件装配到一个总装配体中，最终得到完整的四足机器人装配。从三维设计软件中可测量该机器人长宽高分别为 0.78m、0.34m、0.5m，质量为 17.0kg。

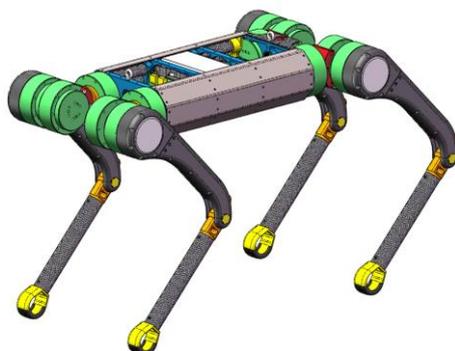


图80 四足机器人整体装配图

## 7.4 零件加工与装配

为了交付工厂加工，将三维设计软件设计的三维图转化为二维图。然后将工厂加工的零件装配成物理样机，再将第六章中介绍的控制系统硬件布置到物理样机上，最终搭建完成的物理样机如图81所示。



图81 搭建完成的物理样机

## 7.5 物理样机测试

### 7.5.1 对角步态原地踏步实验

对角步态是四足机器人行走最常见的步态，也是动步态中最稳定的步态。在前文中已经在仿真平台下测试了对角步态原地踏步的稳定性，本小节将测试物理样机以对角步态原地踏步，并打印欧拉角，测试结果如图82所示。与图65对比，可见物理样机的稳定性略差于仿真环境的虚拟样机，这是因为实际物理环境有各种各样的干扰和误差，但实际物理环境下控制系统对躯干的转动控制效果依然在可接受范围内。

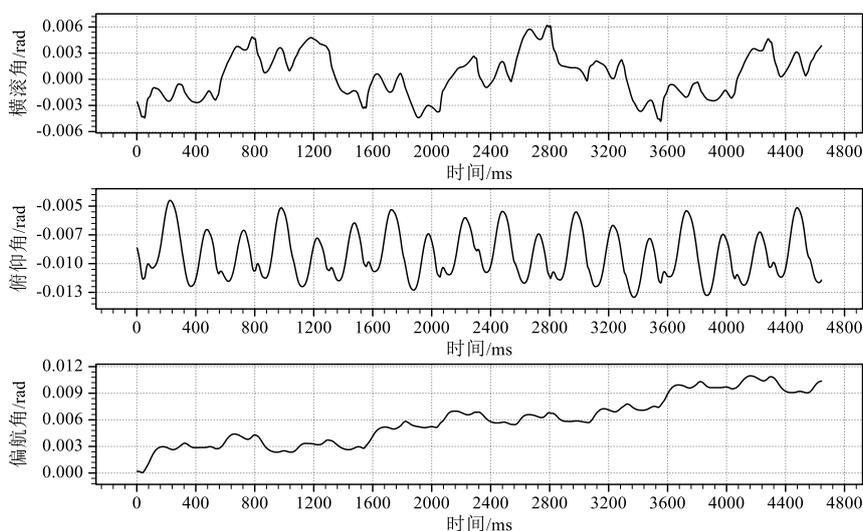


图82 物理样机原地踏步时欧拉角变化曲线

### 7.5.2 直线行走实验

四足机器人最常见的运动形式是向正前方行走。在物理样机上，控制机器人以0.5m/s的速度向前行走，并打印欧拉角，测试结果如图 X 所示。相对于图82中原地踏步

时欧拉角的变化，直线行走时稳定性略差。这是因为在动态移动过程中四足机器人的支撑中心不在质心正下方，在重力的作用下机器人物理系统有失稳的趋势，但控制系统又尽可能使机器人恢复稳定，在这两种作用下机器人进入一种相对稳定的动态平衡状态。从而既不失稳，又在稳定性上略差于原地踏步。

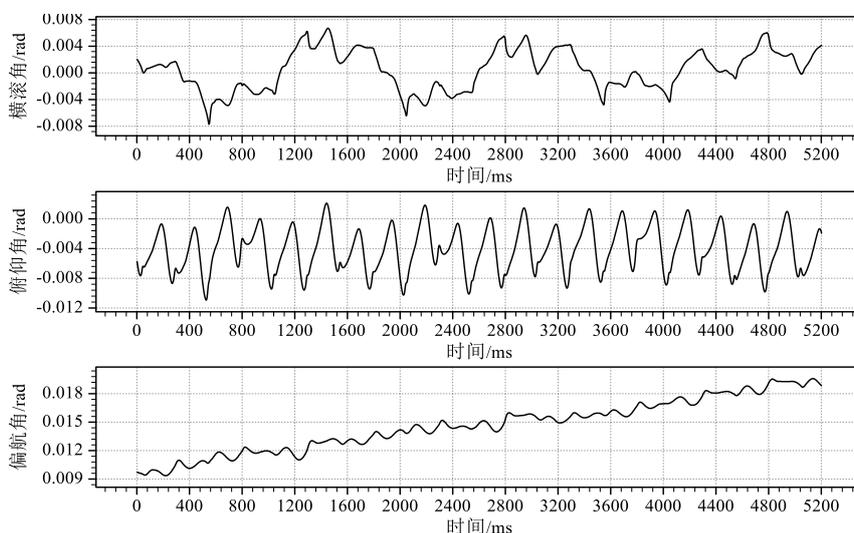


图83 物理样机原地踏步时欧拉角变化曲线

### 7.5.3 非结构化路面行走实验

能够在非结构化路面上行走是高稳定性四足机器人的一个重要标志。为了验证物理样机具有在非结构化路面上行走的能力，设计了非结构化路面行走实验。在地面上随机放置3块高度为40mm的木板，控制机器人依次从木板上走过，并实时打印其躯干角。为保证随机性，控制系统提前并不“知道”木板的高度。实验结果如图84所示。从实验结果来看，物理样机具有在非结构化路面上行走的能力。

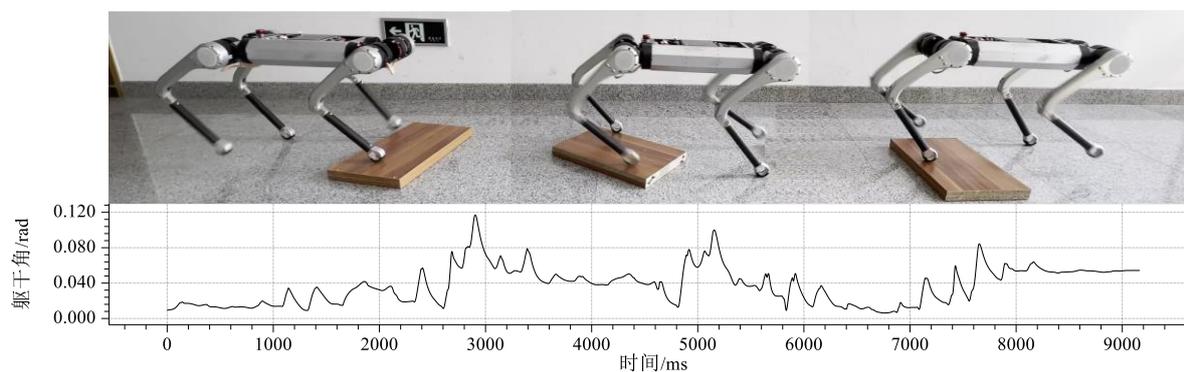


图84 非结构化路面行走实验结果

### 7.5.4 侧向冲击实验

在受到侧向冲击后恢复稳定的能力是四足机器人极为重要的能力，图85为四足机器人在对角踏步时，受到侧向冲击后恢复稳定的时序图及躯干角随时间变化曲线。从实验

结果可见，物理样机在受到侧向冲击后，具有一定恢复稳定的能力。

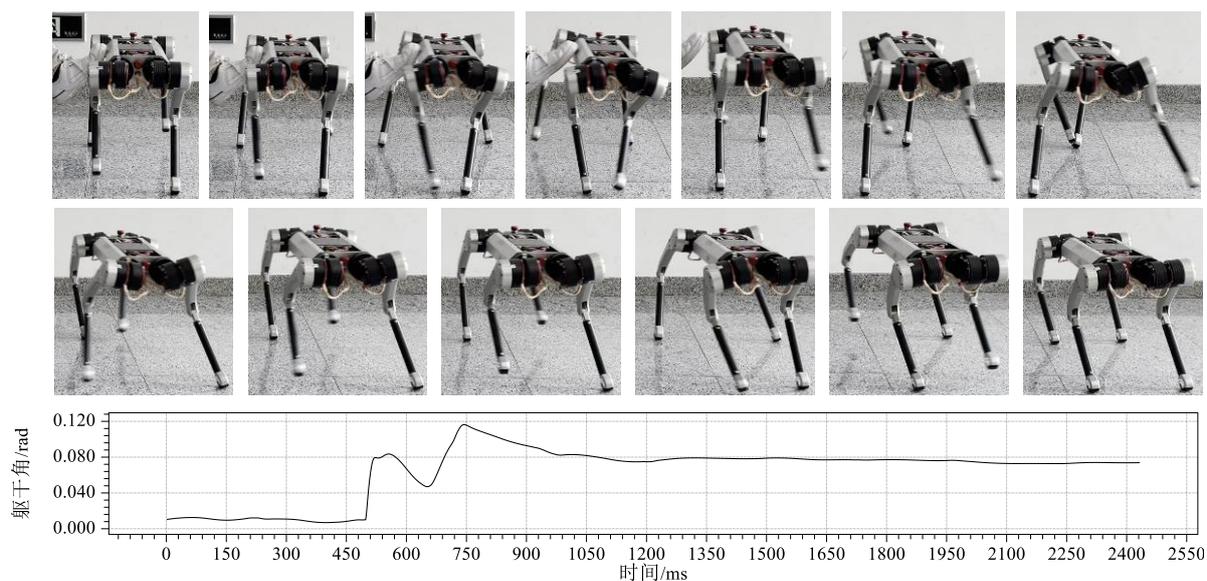


图85 侧向冲击实验结果

### 7.5.5 顶倒立水瓶实验

如图86所示，为了验证控制系统对躯干控制的稳定性，在机器人原地踏步时，在机器人背上放置一个倒立的装满水的水瓶。实验时机器人踏步的频率为2Hz，抬腿高度6cm。实验现象为：在摆动足与地面冲击的作用下，机器人躯干有轻微振动，但这种振动不足以将水瓶振倒，仅仅使水瓶在机器人背上轻微滑动；约15s后，水瓶从机器人背上的正中心滑到一个螺栓旁边，被螺栓挡住后不再移动。该实验验证了控制系统具有很强的维持躯干平稳的能力。



图86 顶倒立水瓶实验

### 7.5.6 四脚共线实验

如图87所示，为验证含有预测成分的控制系統对四足机器人欠驱性的处理能力，调整公式(2.78)中对称点坐标，使机器人在原地踏步时，四个脚的落脚点在同一条直线上，并设置踏步频率为 2Hz。在以对角步态为基础的四脚共线步态中，四足机器人实现了前进、后退、转弯、侧向移动等动作。该实验进一步验证了基于 MPC 与 WBC 的控制系統在控制四足机器人时的稳定能力。



图87 四脚共线实验

## 7.6 本章小结

本章首先介绍了不同腿型布置的优缺点，并最终选择了全肘式腿型；随后介绍了膝关节电机到膝关节的传动方式；然后介绍了关键零部件的设计与仿真，包括躯干、电机座、内曲柄、外曲柄、大腿壳和连杆。最后介绍了在物理样机上实现的几个实验，包括对角步态原地踏步实验、直线行走实验、非结构化路面行走实验、侧向冲击实验、顶倒立水瓶实验和四脚共线实验，验证了本文介绍的控制系統在物理样机上依然有较好的稳定性。

## 总结与展望

### 工作总结：

本文以模型预测控制（MPC）与全身控制（WBC）为核心，在四足机器人的控制理论、控制系统软硬件设计、结构设计、样机测试等方面进行了深入的研究。

(1) 利用卡尔曼滤波器设计了一个足底里程计：假设支撑足在地面上不打滑，通过建立本体系下足底速度与世界系下足底速度之间的映射关系，来求解世界系下质心的位置与速度。

(2) 设计了质心轨迹规划器和足底轨迹规划器：利用手柄传来的上层指令，解算接下来一段时间质心的期望平面轨迹；根据实际质心水平速度和期望质心水平速度的偏差，选择合适的落脚点；在足底离地点和落脚点之间规划一条合适的足底轨迹。

(3) 设计了基于模型预测控制与全身控制的控制器：在模型预测控制中，将整个机器人简化为一个单刚体模型并分析其动力学，建立状态方程、预测方程，将控制问题转化为一个优化问题，最终求出足底反力。在全身控制中，采用了多刚体动力学。四足机器人全身控制将整体的任务按照重要程度分为四个子任务，利用雅可比矩阵的零空间特性，保证低优先级的任务无法影响高优先级任务的控制。

(4) 为了验证控制系统的可靠性，在仿真环境下搭建四足机器人虚拟样机，仿真验证了控制系统足以将不同步态下受到侧向冲击的四足机器人稳定下来。

(5) 最后，设计了四足机器人主体结构，主要包括布置驱动模块、选择传动方式、设计躯干及腿部结构、对关键零部件进行有限元仿真。将零件交付加工后，装配并搭建物理样机，在物理样机上再次验证了控制系统的稳定性。

### 工作展望：

尽管在虚拟样机和物理样机上都实现了不错的稳定控制效果，但整个系统还有一定的进步空间，下面是对四足机器人后续研究工作的一些展望。

(1) 精准建模。在对四足机器人的数学建模过程中，为了简便做了很多近似与假设，比如认为躯干质心与躯干几何中心重合；比如为了线性化欧拉方程，忽略了欧拉第二项；比如递推牛顿欧拉法中未考虑关节摩擦力。若想得到更好的控制效果，应尽可能用更精准模型。

(2) 设计考虑动力学的轨迹规划器。本文的质心轨迹规划器仅考虑了手柄指令，未考虑生成的轨迹是否合理。在对角步态这种简单动步态上是可行的，但是若要探索更多

的步态，就必须从动力学角度设计稳定判据，用来辅助生成轨迹。

(3) 自主研发电机模组。作为四足机器人控制系统末端的执行部件，电机模组对系统的影响体现在方方面面：在结构设计时，大多数非标零件尺寸与电机模组尺寸相关联；在选择电池时，电压、电流主要取决于电机模组；在控制算法上，需要考虑电机模组的通信协议、功能以及控制带宽。可以说一旦选型了电机模组，机器人的大致形式也就确定了。因此，为了在研发四足机器人时有更高的灵活性，有必要自主研发电机模组。

(4) 设计更轻的腿。物理样机的壳体型大腿是研发团队首次采用的构型，设计较为保守，完全有可能进一步减轻腿部质量。

(5) 选择更好的主控制器。Up board 的算力勉强满足 MPC 的计算规模，但是为了在设计控制系统时有更好高的灵活性，应该选用更强大的主控制器。

(6) 加入足底传感器。扰动的主要来源是足底与地面的冲击，因此若落脚的过程中有一定的反馈，将对系统稳定性有极大改善。

(7) 尝试基于强化学习的控制算法。尽管基于 MPC+WBC 的四足机器人控制系统已经取得了不错的控制效果，但是与真实的四足动物相比，机器人的动作与行为明显简单、单调与僵硬。自古以来，但凡人造事物，总是简洁而明确的。控制系统的设计者总是倾向于将控制系统设计的清晰明白，所有环节皆可解释。但人力有时而穷，随着被控系统复杂性增加，这种直接设计控制系统的工作将变得越来越繁琐。因此，不妨设计一些简单的规则，就如同生命在自然界亿万年来进化一样，让四足机器人在仿真环境下不断试错，并迭代进化，最终收敛出一个可用的控制系统。这种无需设计控制系统具体细节，只需要指定最终指标的控制系统的搭建方法，必将在接下来的几年大放异彩。

## 参考文献

- [1] Craig John J. 机器人学导论[M]. 贞超, 译. 机械工业出版社, 2006.
- [2] 蔡自兴. 机器人学[M]. 清华大学出版社, 2000.
- [3] Bledt G. Policy Regularized Model Predictive Control Framework for Robust Legged Locomotion[D]. Massachusetts Institute of Technology, 2018.
- [4] 王田苗, 陶永. 我国工业机器人技术现状与产业化发展战略[J]. 机械工程学报, 2014,50(9):1-13.
- [5] 肖竹, 王勇超, 田斌, 等. 超宽带定位研究与应用:回顾和展望[J]. 电子学报, 2011,39(1):133-141.
- [6] 杨钧杰, 孙浩, 王常虹, 等. 四足机器人研究综述[J]. 导航定位与授时, 2019,6(5):61-73.
- [7] 刘静, 赵晓光, 谭民. 腿式机器人的研究综述[J]. 机器人, 2006(01):81-88.
- [8] Raibert M. Legged Robots That Balance[M]. MIT Press, 1986.
- [9] Seok S, Wang A, Otten D, et al. Actuator design for high force proprioceptive control in fast legged locomotion: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2012[C]. IEEE, 2012.
- [10] Wensing P M, Wang A, Seok S, et al. Proprioceptive Actuator Design in the MIT Cheetah: Impact Mitigation and High-Bandwidth Physical Interaction for Dynamic Legged Robots[J]. IEEE transactions on robotics, 2017,33(3):509-522.
- [11] Katz B G. A Low Cost Modular Actuator for Dynamic Robots[D]. Massachusetts Institute of Technology, 2018.
- [12] Seok S, Wang A, Chuah M Y, et al. Design Principles for Energy-Efficient Legged Locomotion and Implementation on the MIT Cheetah Robot[J]. IEEE/ASME Transactions on Mechatronics, 2015,20(3):1117-1129.
- [13] Seok S, Wang A, Chuah M Y, et al. Design principles for highly efficient quadrupeds and implementation on the MIT Cheetah robot: IEEE International Conference on Robotics and Automation (ICRA), 2013[C]. IEEE, 2013.
- [14] Mistry M, Buchli J, Schaal S. Inverse dynamics control of floating base systems using orthogonal decomposition: 2010 IEEE International Conference on Robotics and Automation, 2010[C].
- [15] Kim D, Di Carlo J, Katz B, et al. Highly Dynamic Quadruped Locomotion via Whole-Body Impulse Control and Model Predictive Control[J]. 2019.

- [16]Di Carlo J, Wensing P M, Katz B, et al. Dynamic Locomotion in the MIT Cheetah 3 Through Convex Model-Predictive Control: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2018[C]. IEEE, 2018.
- [17]Focchi M, Focchi M, Del Prete A, et al. High-slope terrain locomotion for torque-controlled quadruped robots[J]. *Autonomous robots*, 2017,41(1):259-272.
- [18]张国腾, 荣学文, 李贻斌, 等. 基于虚拟模型的四足机器人对角小跑步态控制方法[J]. *机器人*, 2016,38(01):64-74.
- [19]勾文浩, 袁立鹏, 宫赤坤, 等. 基于足端轨迹的仿生四足机器人运动学分析与步态规划[J]. *现代制造工程*, 2017(7):37-41.
- [20]李贻斌, 李彬, 荣学文, 等. 液压驱动四足仿生机器人的结构设计和步态规划[J]. *山东大学学报(工学版)*, 2011,41(05):32-36.
- [21]王兴兴. 新型电驱式四足机器人研制与测试[D]. 上海大学, 2016.
- [22]Hereid A, Powell M J, Ames A D. Embedding of SLIP dynamics on underactuated bipedal robots through multi-objective quadratic program based control, 2014[C]. IEEE, 2014.
- [23]Pratt, J., Dilworth, et al. Virtual model control of a bipedal walking robot: *Robotics and Automation*, 1997. Proceedings., 1997 IEEE International Conference on vol.1, 1997[C].
- [24]Jerry P, Chee-Meng C, Ann T, et al. Virtual Model Control: An Intuitive Approach for Bipedal Locomotion[J]. *The International Journal of Robotics Research*, 2001,20(2):129-143.
- [25]李满天, 蒋振宇, 王鹏飞, 等. 基于多虚拟元件的直腿四足机器人Trot步态控制[J]. *吉林大学学报(工学版)*, 2015,45(5):1502-1511.
- [26]李鑫. 基于虚拟模型和阻抗控制的四足液压驱动机器人行走研究[D]. 哈尔滨工业大学机械工程, 2018.
- [27]Yoshihiko N, Hideo H, Tsuneo Y. Task-Priority Based Redundancy Control of Robot Manipulators[J]. *The International Journal of Robotics Research*, 1987,6(2):3-15.
- [28]A. M A, A. K C. Obstacle Avoidance for Kinematically Redundant Manipulators in Dynamically Varying Environments[J]. *The International Journal of Robotics Research*, 1985,4(3):109-117.
- [29]Dietrich A, Ott C, Albu-Schaeffer A. An overview of 'null' space projections for redundant, torque-controlled robots[J]. *International Journal of Robotics Research*, 2015,34(11):1385-1400.
- [30]Siciliano, B., Slotine, et al. A general framework for managing multiple tasks in highly

- redundant robotic systems: *Advanced Robotics*, 1991. 'Robots in Unstructured Environments', 91 ICAR, Fifth International Conference on vol.2, 1991[C].
- [31] Featherstone R. *Rigid body dynamics algorithms*[M]. 2008.
- [32] Featherstone R. Plucker basis vectors: *IEEE International Conference on Robotics and Automation (ICRA)*, 2006[C].
- [33] Featherstone R, Orin D. *Robot dynamics: equations and algorithms*, 2000[C]. IEEE.
- [34] Featherstone R. A Beginner's Guide to 6-D Vectors (Part 1)[J]. *IEEE Robotics & Automation Magazine*, 2010,17(3):83-94.
- [35] Featherstone R. A Beginner's Guide to 6-D Vectors (Part 2)[J]. *IEEE Robotics & Automation Magazine*, 2010,17(4):88-99.
- [36] Bellicoso C D, Jenelten F, Fankhauser P, et al. Dynamic locomotion and whole-body control for quadrupedal robots: *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017[C].
- [37] 陈虹. *模型预测控制*[M]. 科学出版社, 2013.
- [38] 龚建伟, 姜岩, 徐威. *无人驾驶车辆模型预测控制*[M]. 北京理工大学出版社, 2014.
- [39] Bledt G. *Regularized Predictive Control Framework for Robust Dynamic Legged Locomotion*[D]. Massachusetts Institute of Technology, 2020.
- [40] Katz B, Di Carlo J, Kim S. Mini Cheetah: A Platform for Pushing the Limits of Dynamic Quadruped Control: *2019 International Conference on Robotics and Automation (ICRA)*, 2019[C].
- [41] Lynch K, Park F. *Modern Robotics: Mechanics, Planning, and Control*[M]. 2017.
- [42] 罗庆生, 罗霄. *仿生四足机器人技术*[M]. 北京理工大学出版社, 2016.
- [43] 老大中. *变分法基础*[M]. 第3版. 国防工业出版社, 2016.
- [44] Di Carlo J. *Software and Control Design for the MIT Cheetah Quadruped Robots*[D]. Massachusetts Institute of Technology, 2020.
- [45] Bledt G, Powell M J, Katz B, et al. MIT Cheetah 3: Design and Control of a Robust, Dynamic Quadruped Robot, 2018[C]. IEEE, 2018.
- [46] 辛玉红, 王翠红, 杨庆凤. 四足机器人最优腿型配置仿真研究[J]. *金陵科技学院学报*, 2018,v.34;No.115(01):47-50.

## 攻读硕士学位期间取得的学术成果

## 致 谢

时间如白驹过隙，忽然而已。在北航的科研生活即将画上圆满的句号，在这三年里我实践了一个完整的科研周期，收获良多。在这个过程中，有许许多多伴我成长的同学、老师和朋友，你们如雾海上的灯塔，指引着我前行的路。

首先我要感谢我的恩师，韩晓建老师。在韩老师的支持下，我与康磊师兄合作完成了并联臂机器人，与关贞成师兄合作完成了位控四足机器人，与高瑞师弟合作完成了力控四足机器人，积累了丰富的工程经验和学术素养，这将是未来工作中宝贵的财富。韩老师为人勤恳、平易近人，具有严谨的科研态度与精益求精的工作作风。本课题从选题到验收，无论是开题答辩还是论文布局，每一步都是在韩老师的指导下完成的，倾注了韩老师大量的心血，在此我向韩老师表示深切的谢意与祝福。

感谢实验室的所有同窗们。感谢带我走进足式机器人领域的周永康师兄和关贞成师兄，感谢厨艺了得的毛新月同学，感谢陪我一起装配、调试、录像以及勇拍急停开关的高瑞师弟、闫泽宇师弟、金泽凡师弟。

感谢我的姑姑王淑青女士，为我提供了无数跨领域奇思妙想的科研灵感。

感谢足式机器人领域的同行们。感谢宇树科技的王兴兴，鹏行智能的赵同阳、刘轶、杨平、张楠、潘志远，哈崎机器人的唐彬，感谢陪我一路同行的云星宇、郭栋、邢伯阳、张璿琛、刘松、黄杰、廖洽源、沈雅阁、杨太文、徐雄仕、吕秀林等等。板凳要坐十年冷，文章不写半句空，希望各位同行脚踏实地，摒弃浮躁，结硬寨打呆仗，稳步推进祖国足式机器人发展。你们是祖国足式机器人的脊梁，是战胜 Boston Dynamics 霸权的希望。

最后特别感谢养育我的父母，感谢你们一直以来的默默支持和付出，感谢你们一如既往的鼓励和无微不至的关爱，你们给了我坚实的后盾，让我可以勇敢地面对科研中的种种问题。