



華南農業大學

本科毕业论文

一种双五连杆构型的轮腿式平衡机器人控制器设计与仿真

曾子敬

201921710126

指导教师 金济 讲师

学 院 名 称

工程学院

专 业 名 称

自动化

论文提交日期

2023 年 5 月 10 日

论文答辩日期

2023 年 5 月 15 日

摘 要

近年来,随着自动化和智能技术的迅速发展,对于高效、多功能的移动机器人需求不断增加。单一的轮式或足式机器人往往在某些特定场景(如崎岖地形、楼梯等)中表现出局限性。两轮足式机器人结合了轮式和足式两种运动方式,能够提高移动机器人的适应性和灵活性以适应复杂多变的地形条件,从而在搜索救援、探测勘查、仓储物流及其他需要快速而灵活的机器人应用领域具有重要价值。

本文针对一种腿部为五连杆并联机构的两足轮腿式机器人展开理论研究。该种机器人可以划分为机体、五连杆腿部以及驱动轮三个部分,全身共含有六个电机。其中,四个电机负责提供腿部足端的输出力(矩),另外两个电机负责提供驱动轮的输出力矩。

本文依据虚拟模型控制思想,将腿部五连杆机构等效为具有一个旋转关节和一个移动关节的单杆模型。在此基础上对两足轮腿式机器人进行动力学建模,依据 LQR 方法设计平衡控制器,并基于 PID 控制思想,将平衡控制器拓展为全身运动控制器,使机器人具备自平衡、直线运动、偏航运动、翻滚运动以及高度控制能力。同时,针对常见的上下台阶场景设计分段控制策略,使机器人能够完成跳跃上台阶和自然下台阶的动作。最后,在 Matlab/Simulink 和 Webots 仿真环境中完成控制器的数值仿真与物理仿真。仿真验证了本文中所设计的全身运动控制器的控制效果,机器人能够在平坦地面上保持平衡,抵抗扰动,并且能够实现期望的运动功能,以及上下台阶的过程;对于单边桥和斜坡等特殊地形 also 具有良好的适应能力。

关键词: 轮足式机器人 虚拟模型控制 线性二次型调节器 建模仿真

Design and simulation of a wheel legged balancing robot with double five-link mechanism

Zeng Zijing

(College of Engineering, South China Agricultural University, Guangzhou 510642, China)

Abstract: In recent years, the demand for efficient and versatile mobile robots has increased with the rapid development of automation and intelligent technology. Single-wheeled or legged robots often exhibit limitations in certain specific scenarios (such as rough terrain, stairs, etc.). Two-wheeled legged robots combine both wheeled and legged locomotion modes, which can improve the adaptability and flexibility of mobile robots to cope with complex and changing terrain conditions, thus having important value in robot applications that require fast and flexible operation, such as search and rescue, exploration and surveying, warehousing and logistics.

This paper focuses on theoretical research on a two-legged wheeled legged robot with legs composed of a five-bar parallel mechanism. The robot can be divided into three parts: the body, the five-bar legs, and the drive wheels, totaling six motors. Four motors are responsible for providing output force (torque) to the foot end of the legs, while the other two motors provide output torque to the drive wheels.

Based on the virtual model control idea, this paper equivalent the leg's five-bar mechanism to a single bar model with one rotary joint and one moving joint. On this basis, the two-legged wheeled legged robot is dynamically modeled, and a balance controller is designed based on the LQR method. Then, based on the PID control idea, the balance controller is extended to a full-body motion controller, enabling the robot to have self-balancing, straight-line movement, yaw movement, rolling movement, and height control capabilities. At the same time, a segmented control strategy is designed for common scenarios such as going up and down stairs, allowing the robot to perform jumping up stairs and natural stair descent actions. Finally, numerical and physical simulations of the controller are performed in Matlab/Simulink and Webots simulation environments. The simulation verifies the control effect of the full-body motion controller designed in this paper. The robot can maintain balance on flat ground, resist

disturbances, realize desired motion functions, and complete the process of going up and down stairs. It also exhibits good adaptability to special terrain such as a one-sided bridge and slopes.

Key words: wheeled-legged robot VMC LQR modelling and simulation

目 录

1 前言	1
1.1 课题的研究背景及意义	1
1.2 国内外研究现状	1
1.2.1 国内研究现状	1
1.2.2 国外研究现状	4
1.3 主要研究内容	8
1.4 本章小结	9
2 五连杆机构数学模型的建立	10
2.1 五连杆机构运动学正解算	10
2.1.1 五连杆机构位置解算	10
2.1.2 五连杆机构速度解算	11
2.2 五连杆机构动力学逆解算	12
2.3 虚拟模型控制	13
2.3.1 虚拟模型控制方法引入	13
2.3.2 虚拟模型的动力学逆解算	13
2.3.3 虚拟模型的等效质量	14
2.4 本章小结	16
3 机器人平衡系统数学模型的建立与分析	16
3.1 自平衡系统动力学模型推导	16
3.2 自平衡系统分析	19
3.2.1 系统平衡点求解	19
3.2.2 系统线性化处理	20
3.2.3 系统稳定性、能控性和能观性分析	21
3.3 本章小结	21
4 基于 LQR 的平衡控制器设计	22
4.1 LQR 线性二次型最优调节问题引入	22
4.2 LQR 平衡控制器设计	23

4.3 LQR 平衡控制下系统稳定性数值验证	23
4.4 本章小结	25
5 全身运动控制器设计	25
5.1 位移与速度控制器设计	25
5.2 全身运动控制器设计	28
5.2.1 控制器参数对腿长的自适应	28
5.2.2 高度（腿长）与翻滚角控制	29
5.2.3 偏航角与两腿同步控制	30
5.3 上下台阶分段控制策略设计	30
5.4 系统控制框架结构	32
5.5 本章小结	34
6 基于 Webots 的物理仿真	34
6.1 Webots 仿真环境介绍	34
6.2 平衡系统物理仿真	34
6.3 平衡系统与腿部动力学控制物理仿真	35
6.4 全身运动控制物理仿真	36
6.5 本章小结	38
7 总结与展望	38
7.1 全文工作总结	38
7.2 未来工作展望	39
参考文献	41
附录	43
附录 1 系统非线性动力学数学模型	43
附录 2 全身运动仿真 Webots 主要代码	44
致谢	50

1 前言

1.1 课题的研究背景及意义

随着社会整体科技水平的提高，机器人的应用越来越广泛，几乎渗透到人类社会生产生活中的所有领域。其设计的初衷主要是为替代人类在工业、服务业、国防救灾和太空探索等领域更高效地完成目标任务，解放劳动力，同时帮助人类到达不便于触及的地域。比如，在汽车等生产工厂中，已经有不少成熟的工业机器人执行工艺流程，大大提高了产能；在线下餐饮、家居生活等场景中，也存在有智能配餐机器人、智能扫地机器人等帮助人们完成简单的工作；在战场以及部分灾难现场中，也可以看到部分搭载着特定传感器或作业机构的机器人自动或半自动地完成战场或灾后现场探索，人员搜救等任务；在月球探测任务中，主要由月球车及其搭载的必要设备帮助人类进行期望的采样与研究工作。

对于机器人来说，移动机器人是其中的一个重要分支。根据 GB/T39405-2020 可以将移动机器人分成以下几种类型：轮式机器人、足腿式机器人、履带式机器人、蠕动式机器人、浮游式机器人、潜游式机器人、飞行式机器人和其他运动方式机器人（国家市场监督管理总局，中国国家标准化管理委员会，2020）。其中，轮式机器人常常因为其结构与控制简单、低自重、高载重而被广泛应用于结构化场景中，但是在复杂地形条件下的适应能力不足，难以进一步拓展使用场景；足腿式机器人则因其具有良好的场地适应能力，如今也正是研究热点之一。由于其涉及技术复杂，多数研究成果目前仍处于实验室测试阶段，难以做到如波士顿动力公司的人形机器人和四足机器人等一样，在复杂的野外环境实现自主移动。另外，足腿式机器人大多在移动速度上，在结构场景中，相比轮式机器人会有一定劣势。

此时，轮腿式移动机器人，顾名思义是指机器人移动部分是由可独立运动的足腿和在足端所安装的轮子组成。它结合了轮式机器人和足腿式机器人的结构特点的移动机器人能够结合两者的优点，能很好的兼顾场地适应能力和移动速度两个方面。

其中，两足轮腿式移动机器人，由于能够凭借较少的执行器就具备较高的运动性能，也拥有着广阔的应用前景，是近年来一个研究热点。

1.2 国内外研究现状

1.2.1 国内研究现状

在 2018 年 10 月，哈尔滨工业大学机器人与系统国家重点实验室发布了一种以液压

驱动的轮腿式类人机器人（Wheel-legged hybrid robot, WLR）。WLR 在伸展状态下，身高 1.65 m，重 50 kg，全身共有 16 个自由度，如图 1 所示。WLR 能够完成如在平地上移动、蹲起、避障和拾起重物等类人行为。

该团队在机械结构设计方面，创新地采用了双气缸并联驱动连杆腰部关节，可以通过改变两气缸的输出位置让腰部实现俯仰和侧倾的两轴旋转运动，相比串联驱动结构，增加了腰部关节的输出扭矩，提高了刚度。同时，在小腿中集成了磁流变阻尼减震器，极大的提高了 WLR 在复杂地形下的适应能力。在控制算法方面，为保证机器人运动的鲁棒性和稳定性，采用了实时质心计算器、基于自平衡的增益调度控制器和基于质心调整的反馈控制器组成的控制框架，使 WLR 能够在水平地面上稳定的行驶和转弯。其最大移动速度可达 2 m/s，在保持平衡的条件下，能在 2 s 内从 0.35 m 蹲到 0.75 m，并能够完成拾起 2 kg 重物的任务（Li, Zhou and Feng et al., 2018）。

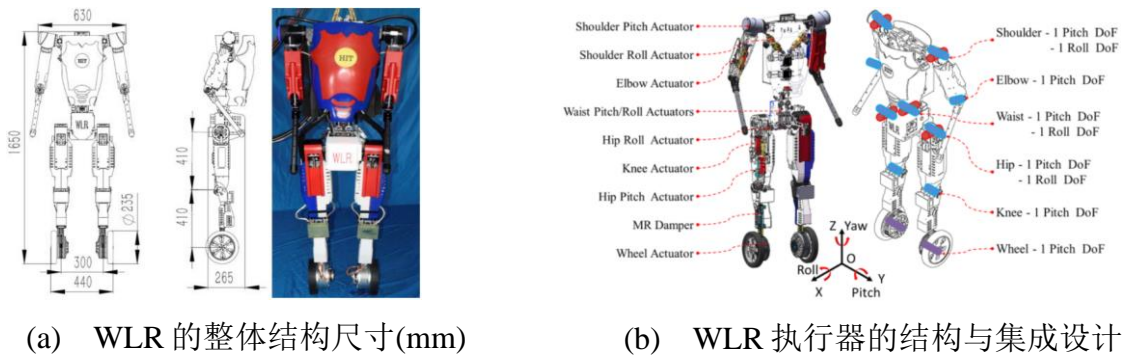


图 1 WLR 机器人尺寸和执行器的结构集成设计（Li et al., 2018）

在 2019 年 9 月，该团队发布了 WLR 二代机器人，如图 2 所示，在新一代 WLR 上，团队首次提出了无软管设计方法，克服了传统液压机器人由于受到移动关节间软管或连接液压驱动单元的限制，而降低机动性的缺点（Li, Zhou and Zhang et al., 2019）。

在 2019 年 12 月，哈尔滨工业大学（深圳）张绍设计了一种双足轮腿机器人，如图 3 所示。作者根据机器人所预期达到的基本平衡和移动功能，采用了 6 自由度的腿部设计方案。为实现类人的环境交互效果，参考了人体髌部和腿部的尺寸进行设计。为优化结构质量分布，膝关节驱动电机置于髌部，采用四连杆结构，将动力传递到膝关节；主动轮驱动电机直接内置于小腿，通过锥形齿轮传递动力，使结构更加紧凑。系统采用

结构可变的轮式倒立摆模型进行运动学建模，平衡和直线行驶速度综合控制采用了串级 PID 控制，将设定速度输入控制器，经过外环的计算得到内环的设定角度值，通过内环的计算得到主动轮驱动的 PWM 信号占空比输出。相对于单环平衡 PID 控制方案来说，增加的外环有利于增加系统的稳定性，产生一定的质心自适应能力。最后在实机的试验中，验证了此控制系统的可行性，取得了较好的效果（C., T. and S. et al., 2019）。

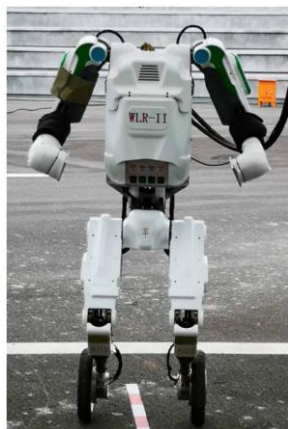


图2 无软管液压轮腿机器人
(Li et al., 2019)



图3 双足轮腿机器人 SR600
(C. et al., 2019)

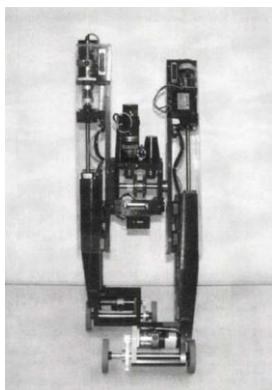
在 2021 年 6 月，腾讯 Robotics X 实验室公布了两足轮腿式机器人 Ollie，如图 4 所示。Ollie 采用了五连杆结构作为腿部构型，具有结构简单、刚性高、负载能力强等特点（S., L. and J. et al., 2021）。在控制层面，为了使 Ollie 保持站立状态而不剧烈向前或向后移动采用了线性输出调节技术和 LQR 技术来平衡机器人；针对初始状态远离平衡点启动到平衡状态过渡过程的控制问题，采用了非线性控制器 IDA-PBC 重新设计用于速度跟踪问题。得益于此，Ollie 具有双轮平衡、多模态移动、跳跃，在不平坦地形行驶等能力。另外，Ollie 机体上创新地加装了一条末端带有从动轮的支撑腿，一方面可以用于在完成需要大载重的任务时，增加机器人的支撑刚度，拓展应用场景；另外一方面，利用其提供额外的角动量，能够使其更好地完成一些高动态运动，如空翻。



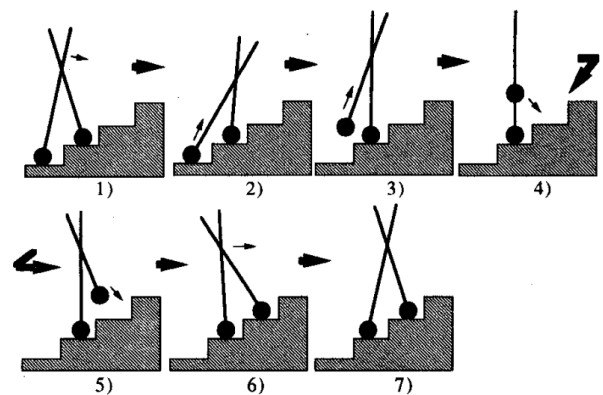
图4 Ollie 轮腿式机器人 (S. et al., 2021)

1.2.2 国外研究现状

在 1998 年间，日本岐阜大学 (Gifu University) 的 Osamu Matsumoto 等人研制出一种五自由度的双足轮机器人，如图 5-(a)所示。机器人高度范围为 0.575-0.74 m，宽 0.23 m，总重量 10 kg，载荷 2 kg。该机器人拥有两条末端装有主动轮的可伸缩机械腿，并分别通过一个旋转电机固定在腰部，以此使机器人实现两条腿分别做伸缩运动的同时前后摆动。该机器人共有轮式和爬楼梯两种运动模式：轮式模式下，机器人通过腿部的驱动轮运动，使机器人可以在平坦地形高速运动；在爬楼梯模式下，机器人以特定步态交替双腿运动并保持平衡，实现了机器人上楼梯的效果，该过程如图 5-(b)示 (Matsumoto, Kajita and Saigo et al., 1998)。



(a) Biped type 轮腿机器人



(b) Biped type 轮腿机器人爬楼梯过程

图5 Biped type 轮腿机器人及其爬楼梯过程 (Matsumoto et al., 1998)

在 2005 年，日本早稻田大学（Waseda University）团队研制出了两足轮腿式机器人 WS-2，如图 6 所示。WS-2 是基于世界上第一个能够载人的两足机器人 WL-16 改进而来的（Y., T. and Y. et al., 2004）。机器人腿部主要由 6 个伺服电动推杆构成的并联机构和脚掌组成，脚掌外侧分别安装了一个驱动轮。可通过改变电动推杆的位移来调整脚掌的内外翻角度，进而在足式和轮式之间切换。当 WS-2 在非平坦地形下，将采用足式行走模式，利用 ZMP（Zero Moment Point）的轨迹规划实现双足稳定动态行走功能；当在规则平坦地形下，机器人调整脚掌内翻，如图 7 所示，使脚掌外侧的轮子触底，内侧抬起，切换成轮式行驶模式。两种模式的兼容使得机器人的通过性和移动速度显著提升（K., T. and Y. et al., 2005）。



图 6 WS-2 样机

（Y., T. and Y. et al., 2004）



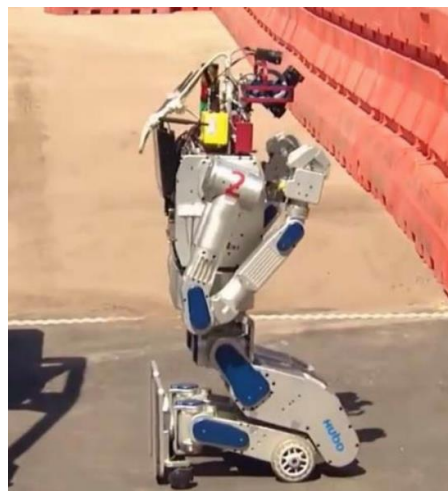
图 7 WS-2 轮式运动时脚掌状态

（K., T. and Y. et al., 2005）

韩国科学技术院（KAIST）研制的的双足机器人 DRC-Hubo 在 KAIST 的双足机器人 Hubo 基础上，两条腿的两侧分别在膝关节添加了驱动轮，在脚掌添加了万向轮改进而来，同样兼具足式运动模式和轮式运动模式。机器人采用站姿与跪姿的切换，来实现运动模式的切换。当机器人需要跨过障碍时，可以采用轮式运动模式，如图 8-b) 所示；当机器人在平台路面时，为加快移动速度或者提高底部刚性时，可以采用轮式运动模型，此时机器人等效为一个四轮差速移动平台（H., I. and T. et al., 2016；J., I. and O. et al., 2015；Oh, Sohn and Jang et al., 2017）。



(a) 足式运动模式



(b) 轮式运动模式

图 8 DRC-Hubo 机器人 (Oh et al., 2017)

2019 年 7 月，加州大学伯克利分校 (University of California, Berkeley) 展示了双足机器人 Cassie 驾驶电动平衡轮滑鞋的研究成果，如图 9 所示。Cassie 是由美国 Agility Robotics 公司所生产的双足机器人，如图 10 所示，可实现室内外稳定行走，具备良好的抗外力扰动能力 (Y., R. and X. et al., 2019)。研究人员建立了 Cassie 机器人和悬浮鞋的接触动力学模型，设计控制器实现了足式机器人驾驶滑轮鞋完成平衡、前后移动和转弯等功能，巧妙地提高了双足机器人的移动能力 (S., J. and B. et al., 2019)。



图 9 Cassie 机器人户外驾驶滑轮鞋
(S. et al., 2019)



图 10 Cassie 机器人非平坦地形站立
(Y. et al., 2019)

在 2017 年 2 月，波士顿动力公司发布了第一代轮腿式机器人 Handle，如图 11 所示。

根据该公司公布的资料显示，Handle 身高 1.9 m，体重 105 kg，移动速度达 24 km/h，有 10 个液压和电机混合驱动关节。它能够通过凹凸不平的平面，下台阶及跳跃，在室内外规则或非规则地形下灵活运动（Boston, 2022）。

在 2019 年 3 月，该公司公布了轮腿式 Handle 机器人二代，如图 12 所示。相较于第一代 Handle 机器人，该机器人更加注重抓取操作能力，为此搭载了一个机械臂。机械臂末端负载可达 15 kg，每小时可搬运码放 360 箱货物，最大移动速度 4 m/s（Boston, 2019）。更具体的机械设计与控制方案细节目前无相关论文发表。



图 11 第一代 Handle 轮腿式机器人
（Boston, 2022）



图 12 第二代 Handle 轮腿式机器人
（Boston, 2019）

2019 年，苏黎世联邦理工大学（ETH Zurich）的学生研发出了一种双轮足式跳跃机器人 Ascento，如图 13 所示，其能够在平面快速移动，能够通过跳跃来通过障碍物，并且具有摔倒自恢复能力。其机械结构设计了腿部三连杆的方案，通过最小均方差的方法优化设计，保证机器人质心的变化轨迹始终垂直于地面，实现机器人跳跃与行驶两种运动的解耦，即跳跃不会使机器人产生旋转运动。在平衡控制上采用线性二次调节器（Linear Quadratic Regulator, LQR）辅助的全身控制（Whole Body Control, WBC）方案，采用了带前馈的闭环反馈控制器来完成跳跃和倾倒恢复动作。其最大站立高度为 660 mm，重量为 10.4 kg，体积较为小巧，能够完成部分室内工作（Klemm, Morra and Salzmann et al., 2020）。

2021 年 12 月，苏黎世发布了 Ascento Pro，如图 14 所示。相较于一代，Ascento Pro 提升了在复杂环境持续跳跃和越障的能力，上楼梯时也能够连续跳跃，同时搭载摄像头

及激光雷达等传感器可以对环境进行数据采集和建模，实现自动驾驶和自主充电等功能（Ascento, 2021）。



图 13 Ascento 实物图
（Klemm et al., 2020）



图 14 Ascento Pro 实物图
（Ascento, 2021）

在 2021 年 1 月，布拉格捷克理工大学（Czech Technical University in Prague）发布了 Skateo 机器人，该机器人拥有和 Ascento 类似的结构，如图 4 所示。其体积更加小巧，主要通过欧拉-拉格朗日方程建立了机器人的非线性平面模型，使用 LQR 控制器实现平衡，用 PD 控制器实现伸缩腿的高度控制。同时实现了基于无迹卡尔曼滤波（Unscented Kalman Filter, UKF）的状态估计和基于优化的跳跃轨迹计算。最后通过 Gazebo 仿真平台进行验证和测试（Kollarcík and Gurtner, 2021）。



图 15 Skateo 实物图（Kollarcík and Gurtner, 2021）

1.3 主要研究内容

本课题的目标是对一种双五连杆构型的轮腿式平衡机器人进行数学与物理建模，并针对该机器人展开运动控制领域内的平衡、移动和姿态调整等相关运动控制方法的研究，

以期结合该种机器人的特点，设计出一套实用的运动控制算法框架，并完成数值与物理仿真验证。本文的研究内容主要包括以下五个方面：

（1）独立出机器人腿部的五连杆机构进行研究，借助运动学雅可比矩阵和虚功原理等方法，对机构进行运动学与动力学建模，以得到机构的两个关节电机的角度和力矩与末端位置和力（矩）之间的映射关系，从而可以将五连杆机构等效为可伸缩可摆动的虚拟连杆结构，为机器人的动力学模型的简化提供理论基础。该部分研究内容对应了论文的第二章。

（2）基于第二章对于五连杆机构的建模结果，简化系统的动力学模型，利用牛顿力学方法列写系统的动力学非线性微分方程组，求解系统的平衡点，并在平衡点处求解雅可比矩阵进行线性化，代入典型值，得到系统线性化后的状态空间表达式，并进行系统的能控能观性以及平衡点处的稳定性进行分析，为后续机器人控制器的设计打下基础。该部分研究内容对应了论文的第三章。

（3）基于第三章建立的机器人动力学模型，使用 LQR 方法进行平衡最优调节控制器设计，期望设计出机器人平衡部分的控制器，使机器人在平衡点附近能够维持稳定，抵抗外界扰动。对于加入闭环控制器后的闭环系统，使用李雅普诺夫第一法验证系统的稳定性，同时也使用 Matlab/Simulink 进行数值验证，确保设计出的控制器能够使得系统渐进稳定。该部分研究内容对应了论文的第四章。

（4）在第四章机器人具有自平衡能力的基础上，针对机器人的全身运动——前后移动、高度调整、偏航运动和翻滚运动进行控制器设计，以 LQR 平衡控制器为基础，补充 PD 或 PID 控制器构建复合控制结构，以期让机器人在保持平衡的前提下，能够具有良好的运动能力。该部分研究内容对应了论文的第五章。

（5）机器人控制算法框架的设计都是基于机器人模型在平衡点处线性化后的线性模型进行，控制算法框架本身具有一定的鲁棒性，对于远离平衡点处状态下也具有一定的控制能力。为测试控制算法框架对于原非线性系统的实际控制效果，使用 Webots 机器人物理仿真平台，搭建机器人物理模型，将控制算法框架通过编程实现，通过仿真验证控制算法框架有效性以及系统超参的合理性。该部分研究内容对应了论文的第六章。

1.4 本章小结

本章阐述了轮腿式机器人的研究背景与意义，突出了轮腿式机器人兼具轮式机器人

和腿式机器人的优点，分析了国内外两足轮腿式机器人的研究现状，并说明了本文的主要研究内容与文章结构。

2 五连杆机构数学模型的建立

本章将对机器人腿部五连杆机构进行数学建模，旨在得到关节电机到足端之间的运动学量与动力学量的映射关系，为后续机器人数学建模时提供模型简化的理论依据。

若无特殊说明，本章的数学建模将基于以下两个假设进行：

- (1) 所有连杆均为理想轻质连杆，不考虑质量的影响；
- (2) 所有关节均不考虑摩擦力影响。

2.1 五连杆机构运动学正解算

本小节将主要研究五连杆机构的运动学正解算，探究主动关节的角位置、角速度到足端从动关节之间的正映射关系。建立如图 16 所示的直角坐标系，并已标出涉及到的几何量。机构的主动关节为 A 和 E，从动关节为 B、C、D，关节角度为 q_1 、 q_2 、 q_3 、 q_4 ，连杆长度为 L_1 、 L_2 、 L_3 、 L_4 、 L_5 。

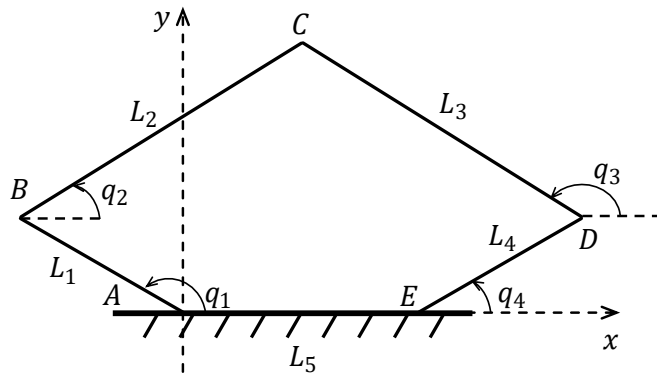


图 16 五连杆机构

下面，在已知五连杆长度 L_1 、 L_2 、 L_3 、 L_4 、 L_5 ，主动关节角度 q_1 、 q_4 和角速度 \dot{q}_1 、 \dot{q}_4 的条件下，求解五连杆主动关节 A 和 E 角度和角速度到从动关节 C 位置与速度映射关系。

2.1.1 五连杆机构位置解算

首先，由三角函数定义和坐标表示法，易知：

$$x_B = L_1 \cos(q_1) \quad (1)$$

$$y_B = L_1 \sin(q_1) \quad (2)$$

$$x_D = L_4 \cos(q_4) + L_5 \quad (3)$$

$$y_D = L_4 \sin(q_4) \quad (4)$$

利用向量闭链关系得到：

$$x_c = x_B + L_2 \cos(q_2) = x_D + L_3 \cos(q_3) \quad (5)$$

$$y_c = y_B + L_2 \sin(q_2) = y_D + L_3 \sin(q_3) \quad (6)$$

由式(5)、(6)选择性消去 q_3 ，得到方程：

$$A_0 \cos(q_2) + B_0 \sin(q_2) - C_0 = 0 \quad (7)$$

其中， $A_0 = 2L_2(x_D - x_B)$ ， $B_0 = 2L_2(y_D - y_B)$ ， $C_0 = L_2^2 + L_{BD}^2 - L_3^2$ ， $L_{BD} = \sqrt{(x_D - x_B)^2 + (y_D - y_B)^2}$ 。求解方程式(7)可得关节角度：

$$q_2 = 2 \tan^{-1} \left(\frac{B_0 \pm \sqrt{A_0^2 + B_0^2 - C_0^2}}{A_0 + C_0} \right) \quad (8)$$

本文所研究的五连杆机构不会出现如图 17 所示的工作状况，

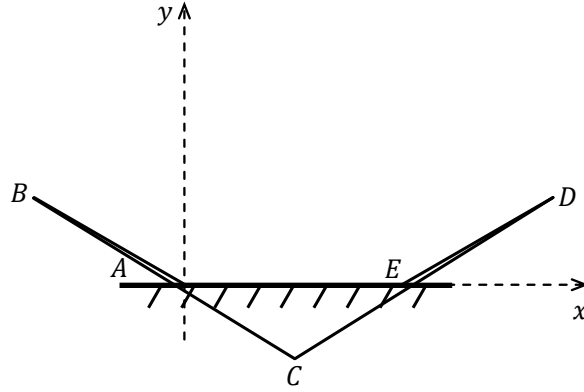


图 17 五连杆机构异常工况

故舍去符号取“-”号的情况。将式(8)带回式(5)、(6)可得到 C 点的位置分量 x_c 和 y_c ，而关节 D 的关节角度则可以通过式(9)求取。

$$q_3 = \tan^{-1} \left(\frac{y_c - y_D}{x_c - x_D} \right) \quad (9)$$

2.1.2 五连杆机构速度解算

式(5)、(6)左右两边分别对时间 t 求导，得：

$$\dot{x}_B - L_2 \dot{q}_2 \sin(q_2) = \dot{x}_D - L_3 \dot{q}_3 \sin(q_3) \quad (10)$$

$$\dot{y}_B + L_2 \dot{q}_2 \cos(q_2) = \dot{y}_D + L_3 \dot{q}_3 \cos(q_3) \quad (11)$$

由式(10)、(11)解得从动关节 B 和 D 的角速度为:

$$\dot{q}_2 = \frac{(\dot{x}_B - \dot{x}_D) \cos(q_3) + (\dot{y}_B - \dot{y}_D) \sin(q_3)}{L_2 \sin(q_2 - q_3)} \quad (12)$$

$$\dot{q}_3 = \frac{(\dot{x}_B - \dot{x}_D) \cos(q_2) + (\dot{y}_B - \dot{y}_D) \sin(q_2)}{L_3 \sin(q_2 - q_3)} \quad (13)$$

利用几何关系得到:

$$x_C = L_1 \cos(q_1) + L_2 \cos(q_2) \quad (14)$$

$$y_C = L_1 \sin(q_1) + L_2 \sin(q_2) \quad (15)$$

式(14)、(15)左右两边分别对 t 求导即可得到 C 点的速度表达式:

$$\dot{x}_C = -L_2 \dot{q}_2 \sin(q_2) - L_1 \dot{q}_1 \sin(q_1) \quad (16)$$

$$\dot{y}_C = L_2 \dot{q}_2 \cos(q_2) + L_1 \dot{q}_1 \cos(q_1) \quad (17)$$

先将式(1)~(4)分别对时间 t 求导得到 \dot{x}_B 、 \dot{y}_B 、 \dot{x}_D 、 \dot{y}_D ，代入式(12)得到 \dot{q}_2 ，再代入式(16)、(17)得:

$$\dot{x}_C = \frac{L_1 \dot{q}_1 \sin(q_1 - q_2) \sin(q_3) + L_4 \dot{q}_4 \sin(q_2) \sin(q_3 - q_4)}{\sin(q_2 - q_3)} \quad (18)$$

$$\begin{aligned} \dot{y}_C &= L_1 \dot{q}_1 \cos(q_1) \\ &\quad - \frac{\cos(q_2) (L_1 \dot{q}_1 \sin(q_1 - q_3) + L_4 \dot{q}_4 \sin(q_3 - q_4))}{\sin(q_2 - q_3)} \end{aligned} \quad (19)$$

2.2 五连杆机构动力学逆解算

本小节将主要研究五连杆机构的动力学逆解算规律，探究足端从动关节的输出力和力矩到主动关节输出力矩之间的逆映射关系。

设从动关节 C 点输出的线性力为 $\mathbf{F} = [F_x \ F_y]^T$ ，位移为 $\mathbf{x} = [x_C \ y_C]^T$ ；主动关节 A 和 E 的输出关节力矩为 $\boldsymbol{\tau} = [\tau_1 \ \tau_4]^T$ ，角位移 $\mathbf{q} = [q_1 \ q_4]^T$ 。其中， F_x 、 F_y 和 x_C 、 y_C 分别指关节 C 沿坐标轴的输出力和位移分量。

由虚功原理得到:

$$\mathbf{F}^T \delta \mathbf{x} - \boldsymbol{\tau}^T \delta \mathbf{q} = 0 \quad (20)$$

由式(18)、(19)可以看出，从动关节 C 的速度与主动关节 A 和 E 的角速度 \dot{q}_1 和 \dot{q}_4 相关，且 \dot{q}_1 和 \dot{q}_4 相互独立，构成二元线性方程组形式，转化成矩阵形式:

$$\begin{bmatrix} \dot{x}_C \\ \dot{y}_C \end{bmatrix} = J_{v\omega} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_4 \end{bmatrix} \quad (21)$$

即:

$$\delta \mathbf{x} = \mathbf{J}_{v\omega} \delta \mathbf{q} \quad (22)$$

其中, $\mathbf{J}_{v\omega} =$

$$\begin{bmatrix} \frac{L_1 \sin(q_1 - q_2) \sin(q_3)}{\sin(q_2 - q_3)} & \frac{L_4 \sin(q_2) \sin(q_3 - q_4)}{\sin(q_2 - q_3)} \\ L_1 \cos(q_1) - \frac{L_1 \sin(q_1 - q_3) \cos(q_2)}{\sin(q_2 - q_3)} & \frac{L_4 \sin(q_3 - q_4) (-\cos(q_2))}{\sin(q_2 - q_3)} \end{bmatrix} \quad (23)$$

由式(20)、(22)联立得到:

$$\boldsymbol{\tau} = \mathbf{J}_{v\omega}^T \mathbf{F} \quad (24)$$

2.3 虚拟模型控制

2.3.1 虚拟模型控制方法引入

传统的控制策略在处理具有复杂关节结构和多自由度的机器人时面临诸多挑战。这些方法通常需要对庞大的系统状态空间进行精确建模与优化, 计算复杂度较高。因此, 有必要探索一种更为直观、易于实现的控制策略。虚拟模型控制 (Virtual Model Control, VMC) 是一种先进的机器人控制策略, 旨在通过模拟虚拟组件和约束来生成合适的控制力矩。这一方法在 1990 年代由 Jerry Pratt 和 Gill Pratt 首次提出 (Pratt, Chew and Torres et al., 2001), 目的是解决高自由度机器人的运动控制问题。

虚拟模型控制的基本思想是在机器人系统中引入虚拟组件 (如弹簧、阻尼器等), 通过设定它们的作用方式来实现期望的运动特性。这些虚拟组件并不存在于实际的物理系统中, 但可以将复杂数学模型转化为简单的几何和力学关系。控制器会根据虚拟组件产生的虚拟力和力矩来调整实际关节驱动力矩, 以实现所需的运动行为 (谢惠祥, 2015)。

2.3.2 虚拟模型的动力学逆解算

为了简化轮腿机器人的动力学模型, 现于五连杆机构中引入虚拟关节 \mathbf{M} 和虚拟连杆 (合称“虚拟腿”), 并且对应引入虚拟力矩 T_b 与虚拟力 F_y 。连接关节 \mathbf{C} 与 \mathbf{AE} 中点处的虚拟关节 \mathbf{M} , 得到虚拟连杆 \mathbf{CM} 。虚拟连杆的长度与虚拟关节 \mathbf{M} 的角度分别为 L_0 和 q_0 , 如图 18 中红色虚线段所示。

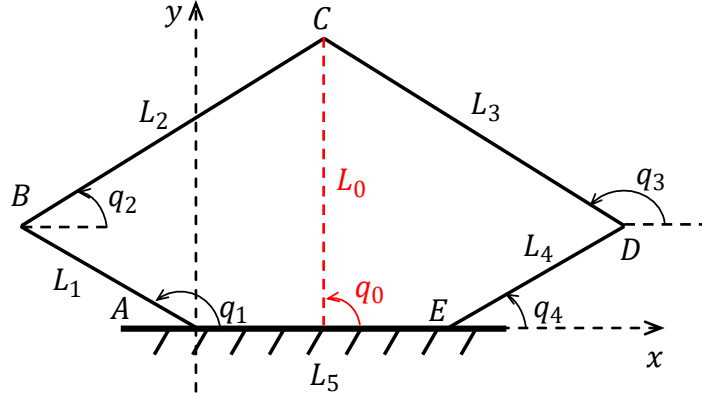


图 18 虚拟连杆和虚拟关节

在上一小节中已经得到了速度雅可比矩阵 $J_{v\omega}$ ，下面将通过转置、旋转变换 T_{rot} 和力到力矩的映射变换 T_{map} 。根据式(24)体现的原理，所得结果，即为从动关节 C 的输出力（矩）到主动关节输出力矩之间的逆映射关系：

$$J_{\tau F} = J_{v\omega}^T T_{rot} T_{map} \quad (25)$$

其中，

$$T_{rot} = \begin{bmatrix} \cos(q_0 - \frac{\pi}{2}) & -\sin(q_0 - \frac{\pi}{2}) \\ \sin(q_0 - \frac{\pi}{2}) & \cos(q_0 - \frac{\pi}{2}) \end{bmatrix} \quad (26)$$

$$T_{map} = \begin{bmatrix} -\frac{1}{L_0} & 0 \\ 0 & 1 \end{bmatrix} \quad (27)$$

$$J_{\tau F} = \begin{bmatrix} \frac{L_1 \sin(q_1 - q_2) \cos(q_0 - q_3)}{L_0 \sin(q_3 - q_2)} & \frac{L_1 \sin(q_1 - q_2) \sin(q_0 - q_3)}{\sin(q_3 - q_2)} \\ \frac{L_4 \sin(q_3 - q_4) \cos(q_0 - q_2)}{L_0 \sin(q_3 - q_2)} & \frac{L_4 \sin(q_3 - q_4) \sin(q_0 - q_2)}{\sin(q_3 - q_2)} \end{bmatrix} \quad (28)$$

最后可以得到：

$$\begin{bmatrix} \tau_1 \\ \tau_4 \end{bmatrix} = J_{\tau F} \begin{bmatrix} T_b \\ F_y \end{bmatrix} \quad (29)$$

2.3.3 虚拟模型的等效质量

本章节所有模型的推到过程均基于开始提出的两点假设，而本小节将为后续自平衡系统建模中所需要的虚拟腿的质量与转动惯量进行数学求解，以得到一个精确的腿部五连杆虚拟模型的等效质量与转动惯量，用于 Webots 物理仿真建模。

现在给五连杆机构引入质量与转动惯量，并在 C 点处固定一个关节电机。为便于计

算以及机械结构设计，做出两点假设：

- （1）连杆质量分布关于长轴呈对称分布；
- （2）五连杆左右两连杆对应对称。

基于如上假设搭建出如图 19 所示的五连杆机构质量分布图。定义变量见表 1。

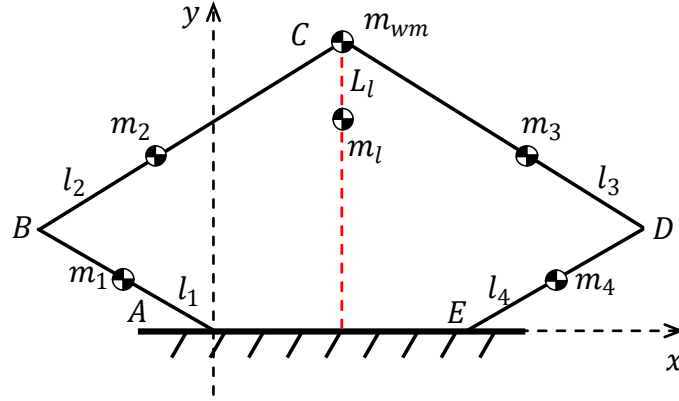


图 19 腿部质量分布

表 1 腿部质量分布变量定义

符号	含义	单位
m_1, m_2, m_3, m_4	连杆质量	kg
l_1, l_2, l_3, l_4	连杆质心位置	m
m_{wm}	足端电机质量	kg
m_l	腿部总质量	kg
L_l	腿部等效质心位置	m

易知五连杆机构的等效质量：

$$m_l = m_1 + m_2 + m_3 + m_4 + m_{wm} \quad (30)$$

在第 2.1.1 节中已经得到了五连杆机构的正运动学公式，结合各连杆质心在连杆上的相对位置，容易写出质心位矢分别为：

$$\mathbf{r}_1 = \begin{bmatrix} l_1 \cos(q_1) \\ l_1 \sin(q_1) \end{bmatrix} \quad (31)$$

$$\mathbf{r}_2 = \begin{bmatrix} L_1 \cos(q_1) + l_2 \cos(q_2) \\ L_1 \sin(q_1) + l_2 \sin(q_2) \end{bmatrix} \quad (32)$$

$$\mathbf{r}_3 = \begin{bmatrix} L_4 \cos(q_4) + l_3 \cos(q_3) + L_5 \\ L_4 \sin(q_4) + l_3 \sin(q_3) \end{bmatrix} \quad (33)$$

$$\mathbf{r}_4 = \begin{bmatrix} l_4 \cos(q_4) + L_5 \\ l_4 \sin(q_4) \end{bmatrix} \quad (34)$$

$$\mathbf{r}_{wm} = \begin{bmatrix} L_1 \cos(q_1) + L_2 \cos(q_2) \\ L_1 \sin(q_1) + L_2 \sin(q_2) \end{bmatrix} \quad (35)$$

则根据质心位置公式可以得到腿部等效质心的位矢：

$$\mathbf{r}_l = \frac{m_1 \mathbf{r}_1 + m_2 \mathbf{r}_2 + m_3 \mathbf{r}_3 + m_4 \mathbf{r}_4 + m_{wm} \mathbf{r}_{wm}}{m_l} \quad (36)$$

虚拟关节 M 的位置矢量为：

$$\mathbf{r}_M = \begin{bmatrix} L_5/2 \\ 0 \end{bmatrix} \quad (37)$$

等效质心关节 C 的相对距离为：

$$L_l = \|\mathbf{r}_w - \mathbf{r}_l\| \quad (38)$$

2.4 本章小结

本章针对腿部的五连杆机构进行运动学和动力学分析，得到了正运动学位置和速度解算和逆动力学解算规律，为后续机器人全身动力学模型简化提供了理论依据。

3 机器人平衡系统数学模型的建立与分析

本章将在上一章的基础上，将腿部五连杆机构抽象为一个虚拟连杆和虚拟关节，同时，考虑到机器人左右的对称性，将机器人切分为对称的单足虚拟连杆轮腿模型，并在此简化模型的基础上，进行动力学建模，为后续机器人运动控制器的设计打下基础。

3.1 自平衡系统动力学模型推导

为使得系统在运动过程中，上层机体结构尽可能保证水平，仅通过腿部的运动，改变机器人重心，完成运动控制。本章节系统模型，采用如图 20 所示的简化模型，模型具有 3 个实体——机体、摆杆和轮子，2 个驱动电机(从下到上，分别称为“驱动轮电机”和“虚拟关节电机”)。简化模型中，摆杆代表着某一工作状况下的五连杆结构中的虚拟连杆。

相较于常见的二阶轮式倒立摆模型，增加了一个主动关节，利于控制机体的俯仰角度，这也是五连杆机构轮腿式机器人相较于两轮平衡机器人的突出优势。

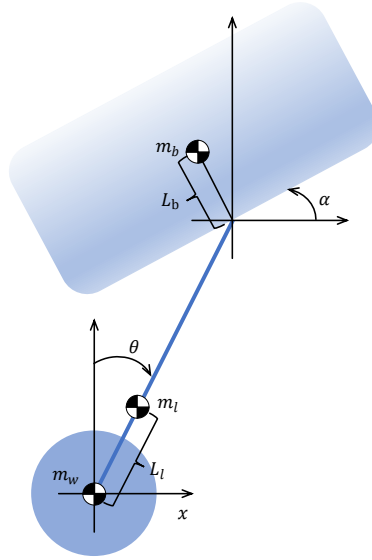


图 20 系统简化模型

现定义以下符号，如图 20 所示。其中，轮子中心的水平位移 x 的正方向为水平向右；摆角 θ 的正方向为顺时针方向；机体的俯仰角 α 的正方向为逆时针方向；而 T_w 和 T_b 以顺时针方向为正方向，参考值是依据实际机器人的物理参数，配合后续仿真进行设计的。

表 2 系统数学模型的符号说明

符号	含义	参考值	单位
L_l	等效腿连杆质心到驱动轮转轴的距离	0.063	m
L	等效腿长	0.210	m
L_b	机体质心到等效髋关节的距离	0.050	m
m_w	轮子质量	0.500	kg
m_l	腿部等效质量	1.600	kg
m_b	机体质量	6.000	kg
J_w	驱动轮绕转轴的转动惯量	0.00072	$kg \cdot m^2$
J_l	腿部绕质心的转动惯量	0.02503	$kg \cdot m^2$
J_b	机体绕质心的转动惯量	0.09005	$kg \cdot m^2$

续表 1

g	重力加速度大小	9.81	$m \cdot s^{-2}$
N_{wx}	驱动轮转轴对腿部的支持力 x 分量大小	-	N
N_{wy}	驱动轮转轴对腿部的支持力 y 分量大小	-	N
N_{bx}	机体对腿部的支持力 x 分量大小	-	N
N_{by}	机体对腿部的支持力 y 分量大小	-	N
x	驱动轮转轴的对地水平位移	-	m
θ	等效腿与竖直方向的夹角	-	rad
α	机体相对地面的俯仰角	-	rad
T_w	驱动轮电机的输出力矩	-	$N \cdot m$
T_b	腿部虚拟关节电机的输出力矩	-	$N \cdot m$

对驱动轮进行力学分析，得：

$$f - N_{wx} = m_w \ddot{x} \quad (39)$$

$$T_w - fR = I_w \frac{\ddot{x}}{R} \quad (40)$$

由式(39)、(40)联立，消去地面对驱动轮的摩擦力，得：

$$T_w - R(m_w \ddot{x} + N_{wx}) = I_w \frac{\ddot{x}}{R} \quad (41)$$

对等效腿部杆件进行力学分析，得：

$$N_{wx} - N_{bx} = m_l \frac{d^2}{dt^2} (L_l \sin \theta + x) \quad (42)$$

$$N_{by} - N_{wy} + m_l g = m_l \frac{d^2}{dt^2} (L_l \cos \theta) \quad (43)$$

$$\begin{aligned} & (N_{by}(L - L_l) + N_{wy}L_l) \sin \theta \\ & - (N_{bx}(L - L_l) + N_{wx}L_l) \cos \theta + T_b - T_w = I_l \ddot{\theta} \end{aligned} \quad (44)$$

对机体进行力学分析，得：

$$N_{bx} = m_b \frac{d^2}{dt^2} (x + L \sin \theta - L_b \sin \alpha) \quad (45)$$

$$N_{by} - m_b g = m_b \frac{d^2}{dt^2} [(L_b + L - L_l) \cos \theta + L_b \cos \alpha] \quad (46)$$

$$T_b + N_{by}L_b \sin \alpha + N_{bx}L_b \cos \alpha = I_b \ddot{\alpha} \quad (47)$$

联立式(42)、(45)解出中间变量 N_{bx} 、 N_{wx} ；联立式(43)、(46)解出中间变量 N_{by} 、 N_{wy} ，将解得的中间变量结果代入式(41)、(44)、(47)，化简并引入状态向量 $\mathbf{x} = [x_1 \ x_2 \ x_3 \ x_4 \ x_5 \ x_6]^T = [x \ \dot{x} \ \theta \ \dot{\theta} \ \alpha \ \dot{\alpha}]^T$ ，即可得到系统运动的非线性动力学模型（详见附录 1）。可利用 Mathematica 的 Solve 函数求解出如下形式的非线性方程组：

$$\begin{cases} \dot{x}_1 = f_1(x_1, x_2, x_3, x_4, x_5, x_6, T_w, T_b) \\ \dot{x}_2 = f_2(x_1, x_2, x_3, x_4, x_5, x_6, T_w, T_b) \\ \dot{x}_3 = f_3(x_1, x_2, x_3, x_4, x_5, x_6, T_w, T_b) \\ \dot{x}_4 = f_4(x_1, x_2, x_3, x_4, x_5, x_6, T_w, T_b) \\ \dot{x}_5 = f_5(x_1, x_2, x_3, x_4, x_5, x_6, T_w, T_b) \\ \dot{x}_6 = f_6(x_1, x_2, x_3, x_4, x_5, x_6, T_w, T_b) \end{cases} \quad (48)$$

使用向量表示为： $\dot{\mathbf{x}} = \mathbf{F}(\mathbf{x}, \mathbf{u})$ ，其中， $\mathbf{u} = [T_w \ T_b]^T$ 。

3.2 自平衡系统分析

3.2.1 系统平衡点求解

系统的平衡点定义在，当系统状态向量的变化率 $\dot{\mathbf{x}} = \mathbf{0}$ 、并且系统输入 $\mathbf{u} = \mathbf{0}$ 时，方程 $\mathbf{F}(\mathbf{x}) = \mathbf{0}$ 的解。利用 Mathematica 进行求解，得到系统平衡点为： $\mathbf{x}_{ef} = [x \ 0 \ 0 \ 0 \ 0 \ 0]^T$ 。

从平衡点结果分析，当系统处于平衡状态时，机构位型将如图 21 所示。此时，水平位移 x 不影响系统的平衡状态。

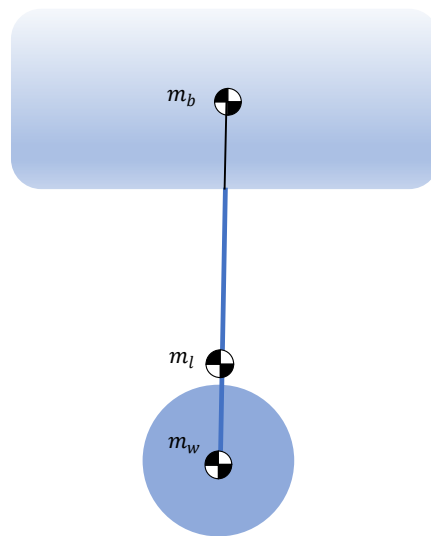


图 21 系统平衡位型示意图

3.2.2 系统线性化处理

为利于系统自平衡控制算法的实现，考虑到实际机器人的正常工作时主要位于平衡点附近，于是，本小节将在平衡点处求出雅可比矩阵，对系统非线性动力学模型完成线性化处理。

对状态变量进行线性化处理得到系统的状态矩阵：

$$\mathbf{A} = \left. \frac{\partial \mathbf{F}}{\partial \mathbf{x}} \right|_{\mathbf{x} = \mathbf{x}_{ef}} \quad (49)$$

对输入向量进行线性化处理得到系统的输入矩阵：

$$\mathbf{B} = \left. \frac{\partial \mathbf{F}}{\partial \mathbf{u}} \right|_{\mathbf{x} = \mathbf{x}_{ef}} \quad (50)$$

故此时系统的状态方程可以写为：

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \quad (51)$$

其中， \mathbf{A} 为方阵，阶数 $n = 6$ ； \mathbf{B} 为 6×2 的矩阵。矩阵形式见式(52)、(53)，元素的具体符号表达式限于篇幅，不具体展开。

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & a_{23} & 0 & a_{25} & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & a_{43} & 0 & a_{45} & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & a_{63} & 0 & a_{65} & 0 \end{bmatrix} \quad (52)$$

$$\mathbf{B} = \begin{bmatrix} 0 & 0 \\ b_{21} & b_{22} \\ 0 & 0 \\ b_{41} & b_{42} \\ 0 & 0 \\ b_{61} & b_{62} \end{bmatrix} \quad (53)$$

将表 2 中的参考值代入式(52)、(53)可以得到（保留三位小数）：

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -33.311 & 0 & 0.118 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 169.909 & 0 & 5.849 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & -29.931 & 0 & 33.127 & 0 \end{bmatrix} \quad (54)$$

$$\mathbf{B} = \begin{bmatrix} 0 & 0 \\ -2.172 & 11.232 \\ 0 & 0 \\ 13.2696 & -48.147 \\ 0 & 0 \\ 9.269 & 1.317 \end{bmatrix} \quad (55)$$

通过传感器检测或根据传感器检测的数据结合几何关系计算，系统的六个状态变量皆可得到，并且系统的输出——关节电机输出力矩与驱动轮电机输出力矩与系统输出无直接关系，故系统的输出方程为：

$$\mathbf{y} = \mathbf{C}\mathbf{x} + \mathbf{D}\mathbf{u} \quad (56)$$

其中， \mathbf{C} 为 6 阶单位矩阵， \mathbf{D} 为零矩阵。

3.2.3 系统稳定性、能控性和能观性分析

上一小节得到了该系统在平衡点处线性化后的状态矩阵 \mathbf{A} 和输入矩阵 \mathbf{B} 。现将对系统的李雅普诺夫稳定性、能控性与能观性进行分析。

根据李雅普诺夫第一法（王新生，曲延滨，2020），如果系统的非线性状态方程可以用其一次近似式线性化，则可以通过对线性化方程的稳定性分析给出原非线性系统在小范围内的稳定性。

由于状态矩阵的特征值为

$$\begin{aligned} r_{1,2} &= 0 \\ r_{3,4} &= \pm 5.86678 \\ r_{5,6} &= \pm 12.9853 \end{aligned} \quad (57)$$

由于此时系统状态矩阵(49)的特征值中含有非负实部，则平衡状态 \mathbf{x}_{ef} 总是不稳定。

下面对系统进行能控能观性的判断。系统的能控性秩判别矩阵：

$$\mathbf{Q}_c = [\mathbf{B} \quad \mathbf{A}\mathbf{B} \quad \mathbf{A}^2\mathbf{B} \quad \cdots \quad \mathbf{A}^{n-1}\mathbf{B}] \quad (58)$$

矩阵的秩 $\text{rank}(\mathbf{Q}_c) = 6 = n$ ，系统状态完全能控。

系统能观性秩判别矩阵：

$$\mathbf{Q}_o = [\mathbf{C} \quad \mathbf{C}\mathbf{A} \quad \cdots \quad \mathbf{C}\mathbf{A}^{n-1}] \quad (59)$$

矩阵的秩 $\text{rank}(\mathbf{Q}_o) = 6 = n$ ，系统可观。

综上所述，系统完全能控且可观，故可以设计一个全状态反馈控制器，自主配置系统的特征根，使得系统在平衡点处渐近稳定。

3.3 本章小结

本章基于第 2 章腿部五连杆的虚拟模型的动力学模型，对本文所研究的双五连杆构型轮腿式机器人腿部进行了简化，完成了机器人单侧的动力学建模；代入实际参考值，得到了系统的状态空间表达式，分析得到了系统的平衡点，完成了模型在系统平衡点附近的线性化近似工作，并分析得到系统是完全能控且可观但不稳定的，为后续系统运动

控制器设计，打下了基础。

4 基于 LQR 的平衡控制器设计

对于两足轮腿式机器人来说，要完成各种复杂运动的前提是机器人能够站立保持平衡并且抵抗一定程度的外界扰动。本章将对简化后的两足轮腿机器人进行动力学建模、平衡控制器参数设计，并利用 Matlab/Simulink 进行初步的数值验证，确保所设计出的控制器能够使机器人在平衡点附近渐进稳定。

4.1 LQR 线性二次型最优调节问题引入

对于某个动态系统的最优化设计，如果研究的系统是线性的，并且选取的性能指标是状态变量与控制变量的二次型函数，那么这一类最优化问题称为线性二次型问题。此类问题的最优解具有统一的解析形式，并且可以导出一个形式简单的线性状态反馈控制规律，便于实现系统的闭环反馈控制。

本文所研究的轮腿式机器人是复杂的多输入多输出系统，经过第 3 章的动力学模型建立与线性化处理，得到了一个线性定常系统。现期望当系统受到某外界扰动偏离平衡点后，给系统一个合理的输入，使系统能够以最小的代价回到平衡点，该问题可以归结为无限时间的线性定常状态调节器的设计问题，采用数学语言表述如下：

设线性定常系统的状态方程

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t), \quad \mathbf{x}(0) = \mathbf{x}_0 \quad (60)$$

选取性能指标

$$J = \frac{1}{2} \int_0^{\infty} [\mathbf{x}(t)^T \mathbf{Q}\mathbf{x}(t) + \mathbf{u}(t)^T \mathbf{R}\mathbf{u}(t)] dt \quad (61)$$

式中 $\mathbf{x}(t) \in R^n$; $\mathbf{u}(t) \in R^m$ ，且无约束； \mathbf{A} 、 \mathbf{B} 、 \mathbf{Q} 和 \mathbf{R} 是实矩阵，且 \mathbf{Q} 和 \mathbf{R} 分别为非负定和正定对称矩阵。要求确定最优控制 $\mathbf{u}^*(t)$ ，使性能指标(61)取得极小值。

对于此类问题，已有成熟的解法，具体结论如下：

对于无限时间的定常 LQ 调节问题， $\mathbf{u}^*(\cdot)$ 为最优控制的充要条件是具有形式：

$$\mathbf{u}^*(t) = -\mathbf{K}\mathbf{x}(t), \quad \mathbf{K} = \mathbf{R}^{-1}\mathbf{B}^T\mathbf{P} \quad (62)$$

式中 \mathbf{P} 为对称正定实矩阵，并且是黎卡提矩阵代数方程(63)的唯一解。

$$\mathbf{P}\mathbf{A} + \mathbf{A}^T\mathbf{P} - \mathbf{P}\mathbf{B}\mathbf{R}^{-1}\mathbf{B}^T\mathbf{P} + \mathbf{Q} = \mathbf{0} \quad (63)$$

最优轨线为下述方程的解：

$$\dot{\mathbf{x}}^*(t) = \mathbf{A}\mathbf{x}^*(t) + \mathbf{B}\mathbf{u}^*(t), \quad \mathbf{x}^*(0) = \mathbf{x}_0 \quad (64)$$

最优性能指标为：

$$J^* = \frac{1}{2} \mathbf{x}_0^T \mathbf{P} \mathbf{x}_0 \quad (65)$$

4.2 LQR 平衡控制器设计

设计 LQR 控制器时，在建立好系统模型之后，最重要的一步就是要进行权重矩阵的赋值。这个过程主要依据实际的需求，根据经验进行。

在机器人平衡控制的过程中，六个状态变量和两个输入量的重要程度不一。根据设计需求，状态变量中的水平位移 x_1 和机体俯仰角 x_5 ，在受到扰动后应尽快恢复到平衡位置，以保证机器人上层姿态良好，故这两个状态变量对应的权重系数要大于其他四个；两个输入量中，由于实际运行当中，可能存在对于底盘功率的限制条件，而对腿部关节电机的输出要求相对宽松，故驱动轮电机输出力矩的代价权重应当比虚拟关节电机输出力矩的权重大。

现根据经验给出权重矩阵：

$$Q = \begin{bmatrix} 2000 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 50 & 0 & 0 & 0 \\ 0 & 0 & 0 & 10 & 0 & 0 \\ 0 & 0 & 0 & 0 & 5000 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (66)$$

$$R = \begin{bmatrix} 0.25 & 0 \\ 0 & 1 \end{bmatrix} \quad (67)$$

借助 Mathematica 软件，式 (66)、(67) 结合式 (54)、(55)，使用 RiccatiSolve[{A,B},{Q,R}] 语句求解黎卡提矩阵代数方程得到矩阵解 \mathbf{P} 的值为：

$$\begin{bmatrix} 1003.037 & 251.159 & 384.431 & 58.790 & -112.237 & -24.444 \\ 251.159 & 82.078 & 134.052 & 19.365 & -33.491 & -8.082 \\ 384.431 & 134.052 & 253.067 & 31.950 & -55.105 & -13.485 \\ 58.790 & 19.365 & 31.950 & 4.632 & -8.339 & -1.931 \\ -112.237 & -33.491 & -55.105 & -8.339 & 232.589 & 7.566 \\ -24.444 & -8.082 & -13.485 & -1.931 & 7.566 & 0.979 \end{bmatrix} \quad (68)$$

进而得到状态反馈矩阵：

$$K = \begin{bmatrix} 32.203 & 15.171 & 31.286 & 6.041 & 128.821 & 4.044 \\ -41.722 & -21.113 & -50.388 & -8.048 & 35.314 & 3.470 \end{bmatrix} \quad (69)$$

4.3 LQR 平衡控制下系统稳定性数值验证

利用 LQR 方法得到了全状态反馈控制的增益矩阵 K 后构成闭环控制系统，此时系统状态方程可以写成

$$\dot{x} = Ax + Bu = Ax + B(-Kx) = (A - BK)x \triangleq A_{cl}x \quad (70)$$

闭环状态矩阵 A_{cl} 的特征值为

$$\begin{aligned} r_1 &= -4.296 \\ r_2 &= -175.326 \\ r_{3,4} &= -3.747 \pm 3.797i \\ r_{5,6} &= -26.247 \pm 22.557i \end{aligned} \quad (71)$$

此时所有的特征值的实部均小于 0，根据李雅普诺夫第一法可以知道，此时系统平衡点 x_{ef} 是李雅普诺夫渐进稳定的。

下面利用 Matlab/Simulink 环境进行系统数值仿真。仿真系统搭建见图 22。

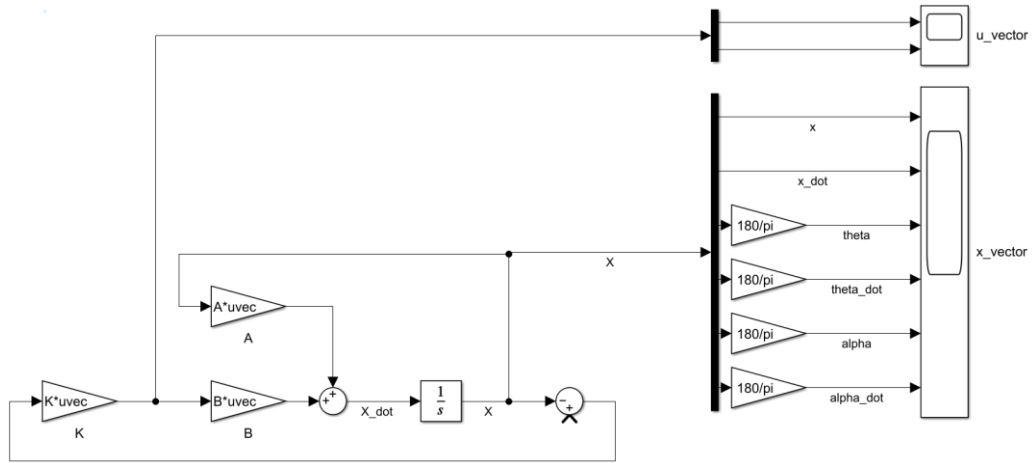


图 22 LQR 平衡控制仿真模型

在积分器中给状态变量 x 设定一个初始扰动量 1，见图 23。

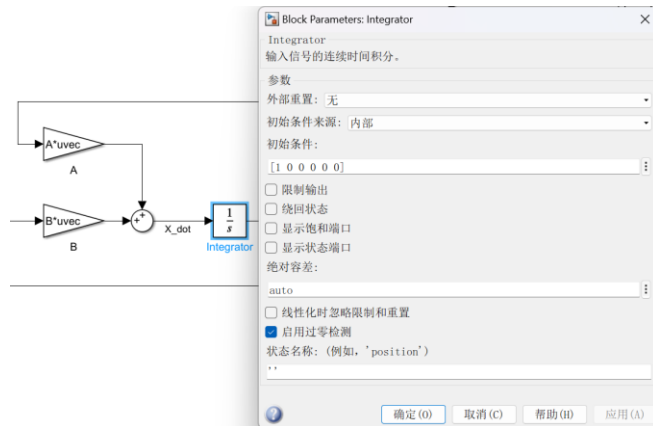


图 23 LQR 平衡控制仿真初值给定

最终系统六个状态变量的变化趋势见图 24。

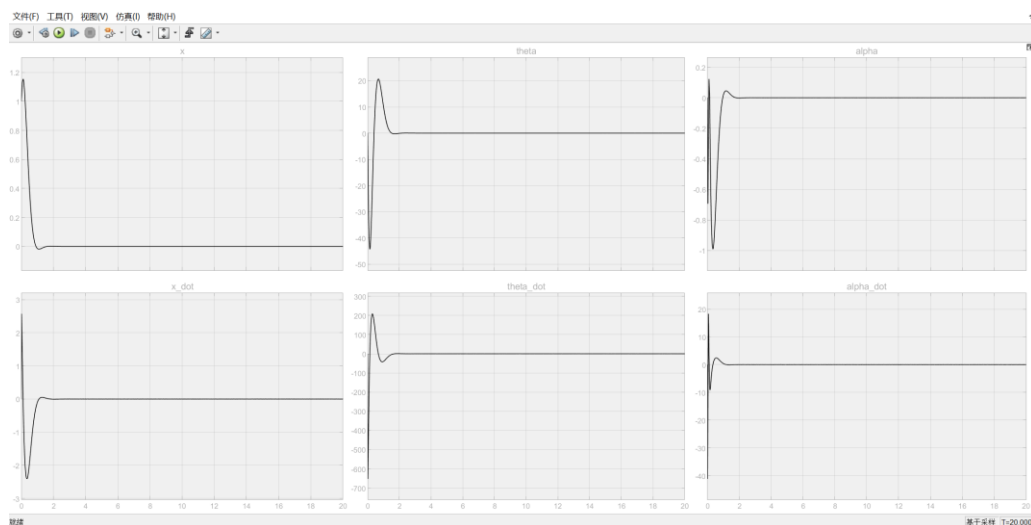


图 24 LQR 平衡控制仿真结果

可见，在首次依据经验进行 LQR 控制器参数设计后，系统在经受扰动之后，能够在一定时间内回到平衡点处，证明了此控制器设计思路与方法的可行性。

4.4 本章小结

本章基于第 3 章得到的系统动力学模型，使用 LQR 方法进行平衡控制器设计，利用拉雅普诺夫第一法与 Matlab/Simulink 搭建数值仿真模型，验证了 LQR 线性二次型最优调节器能够使系统在平衡点附近是渐进稳定的，能够抵抗外界扰动，恢复平衡状态。

5 全身运动控制器设计

经过第 2 到 4 章节的努力，目前已经完成了机器人在单一腿长工况下的平衡控制，而这离最终期望达到的机器人运动能力目标来说，还有一段不小的距离。在本章中，将基于前面的成果，进一步拓展机器人的运动能力，增加对机器人的速度、高度、偏航角以及翻滚角的控制维度，同时，设计专门的算法，实现机器人飞坡和上下台阶，极大增强机器人的地形适应与机动能力。

5.1 位移与速度控制器设计

对于轮腿式机器人来说，完成了自平衡控制以后，首先需要实现的，是让机器人能够沿着机器人的前后方向运动，这也是后续一切复杂运动的基础。

经过 3.2.1 小节中对于系统平衡点的分析，要在系统保持稳定的情况下，同时完成前后运动控制，考虑到系统处于平衡状态时，机器人水平位移的状态变量不影响平衡状态本身，而位移本身是速度的积分，进而可以得出，本小节控制机器人水平运动的主要思路就是通过改变位移变量，来间接控制速度变量。

于是，现在引入参考向量：

$$\mathbf{x}_d = \begin{bmatrix} x_{1d} \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (72)$$

对于原系统进行改进，修改系统控制率为：

$$\mathbf{u} = \mathbf{K}(\mathbf{x}_d - \mathbf{x}) \quad (73)$$

则原系统状态方程转化为：

$$\dot{\mathbf{x}} = (\mathbf{A} - \mathbf{BK})\mathbf{x} + \mathbf{BK}\mathbf{x}_d \quad (74)$$

要使状态变量最终能够稳定到参考量上，则意味着新的系统的稳定点在 $\mathbf{x}_{ef} = [x_d \ 0 \ 0 \ 0 \ 0 \ 0]^T$ 。现在，令 $\dot{\mathbf{x}} = \mathbf{0}$ ，利用 Mathematica 求解方程(75)可以得到 $\mathbf{x} = \mathbf{x}_{ef}$ ，说明此控制器的改进是合理的。

$$(\mathbf{A} - \mathbf{BK})\mathbf{x} + \mathbf{BK}\mathbf{x}_d = \mathbf{0} \quad (75)$$

使用 Matlab/Simulink 进行仿真以初步验证，基于 4.2 节的仿真模型的基础上，修改模型如图 25 所示。初始状态置 0，给位移状态变量一个阶跃参考信号。

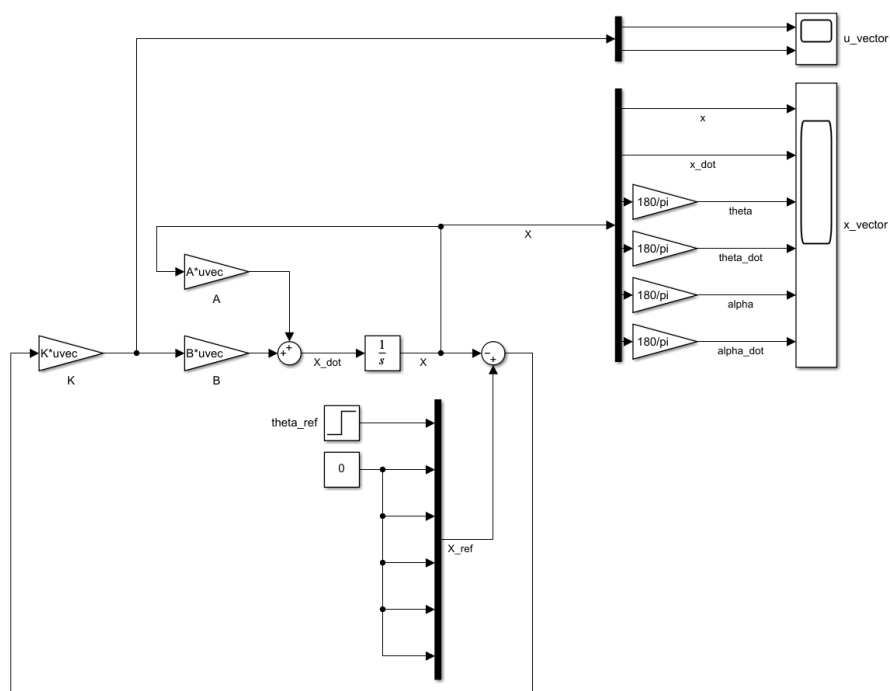


图 25 LQR 位移跟踪控制仿真模型

得到状态变量的变化趋势如下图，可以看出，位移状态变量趋近于参考值，而其他状态变量，在完成过渡过程之后，均恢复到平衡点处。实验也可验证此状态变量 LQR 跟踪控制器设计的合理性。

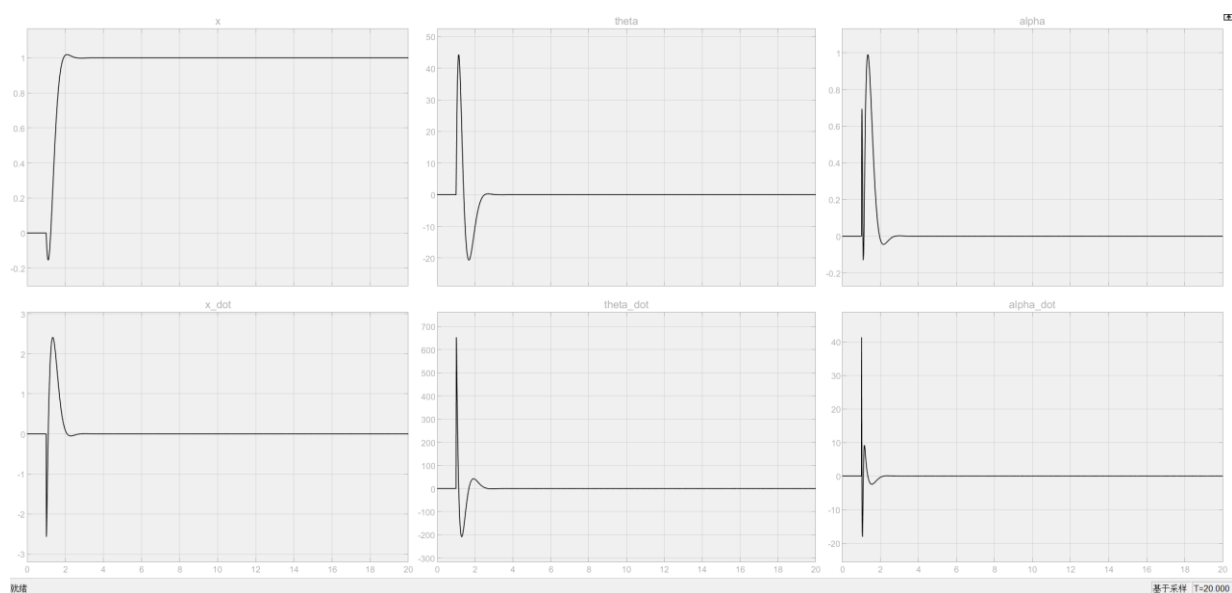


图 26 LQR 位置跟踪控制仿真结果

需要注意的是，在 3.1 节建模的过程中，使用的位移和速度状态向量为单侧轮子中

心的水平位移，在本小节中，需要利用如下公式转化为机体中心的位移与速度。

$$x_b = \frac{x_r + x_l}{2} \quad (76)$$

$$v_b = \frac{v_r + v_l}{2} \quad (77)$$

式中，下标 r 和 l 指示状态变量的来源，下标 b 指示该状态变量基于机体。

5.2 全身运动控制器设计

前文所做的工作，可以归结为机器人在前后方向上的一维运动控制。本小节，将进一步拓展机器人的运动能力，将为机器人赋予高度（腿长），偏航角与翻滚角控制能力，增加三个可控的自由度。

同时，考虑到实际场景中，由于模型的误差或者控制精度的问题，左右腿的控制往往是不对称的，这是为保证控制效果，需要进一步引入两腿同步控制，保证两腿的摆角在控制过程中始终相同。另外，为了拓展机器人的运动能力，对于前文得到的 **LQR** 平衡与跟踪控制器提出了新的要求，要求其对于机器人在不同腿长工况下都具有良好的控制效果，这也是全身运动控制的基础，本章也对控制器改进的方法进行阐述。

5.2.1 控制器参数对腿长的自适应

本小节前的理论分析过程均基于一种人为选取的腿长工况，而实际过程当中，如果时刻控制腿长为恒定值，那么机器人运动能力与传统的两轮式平衡机器人无异，五连杆腿部机构的设计将失去意义。本文所研究的五连杆构型的两足轮腿式机器人相比于两轮式平衡机器人的一大优势在于其腿部结构的设计可以使得机器人机体高度变为可控，极大地增强了地形适应能力，拓宽了机器人的应用场景。

本小节将在前文的基础上，进一步完善控制器，使之在机器人不同腿长工况条件下均能实现机器人的运动控制，确保机器人腿长控制时，系统是稳定且可控的。

前文中所能实现的运动控制效果可以总结为：在某一腿长工况下，机器人能够抵抗扰动，自动恢复到平衡点，并且对于人为给定参考值可以实现跟踪控制。根据经验，将采用多项式拟合的方法，连续等距离改变关节电机角度，间接改变腿长 L_0 ，将反馈矩阵 \mathbf{K} 的元素分别对腿长进行拟合，即可得到新的控制器形式：

$$\mathbf{u} = \mathbf{K}(L_0)(\mathbf{x}_d - \mathbf{x}) \quad (78)$$

下面将简要阐述上述方法的合理性：假设系统运动过程连续且光滑，当系统处在某一腿长工况并且偏离平衡点时，经过控制器(78)作用以后，系统状态将沿此时的最优轨

线 $\mathbf{x}^*(t)$ 趋近平衡点。一段时间后，腿长由于外力或者由于控制误差发生变化，此时，控制器更新为 $\mathbf{u} = \mathbf{K}(L'_0)(\mathbf{x}_d - \mathbf{x}')$ ，在更新后的控制器的作用下，系统将继续以新的最优轨线 $\mathbf{x}''(t)$ 趋近平衡点，如此以来，系统将能够最终恢复到平衡点，实现了控制器参数对于腿长的自适应。

5.2.2 高度（腿长）与翻滚角控制

本小节将基于前文得到的腿部动力学逆解算规律(29)与 LQR 平衡与跟踪控制规律(78)，并引入腿长 PID 控制器实现腿长控制，间接实现机器人的高度与翻滚角控制效果。

第 5.1 小节得到的 LQR 平衡与位移跟踪控制规律的物理意义在于，机器人可以根据自身的状态得到一个最优输入，使得自身完成平衡与位移跟踪控制。其中，系统的输入包含两个部分，一个是驱动轮电机输出力矩，另一个是虚拟关节电机输出力矩。在不考虑其他运动控制的情况下，前者可以直接由驱动轮电机提供，而后者，由于实际机器人并没有这个关节，并且也没有对应的关节电机，故无法直接由某个具体的电机提供。而第 2.3.2 小节中已经得到了腿部五连杆机构的动力学逆解算规律，其物理意义在于，当已知机构末端的期望输出力矩与力之后，逆解算得到关节电机的期望力矩。所以，当 LQR 控制器产生了虚拟关节电机期望力矩之后，可以由两个关节电机的共同作用实现。

然而，该腿部动力学逆解算规律并不是腿部虚拟力矩到关节电机力矩之间的一对一映射关系，关节电机的输出力矩除了产生让虚拟连杆旋转的虚拟力矩 τ_0 以外，还同时可以产生沿着虚拟连杆让其产生伸缩趋势的虚拟力 F_y 。对于此虚拟力的生成，本文将采用使用 PID 加前馈的控制方法。

腿长控制问题是典型的单输入单输出问题，输入量是 F_y ，输出量是 L_0 。实际控制需要机器人的腿长具有弹簧阻尼特性以增强其地形适应能力，故可以引入 PD 控制，而为了保证腿长控制的精度，可以引入积分控制。同时，考虑到机器人在地面上运动，其重力的影响是持续且可以由先验知识计算得到，故引入前馈控制，将系统重力内模进控制器中，以实现对于重力的抵消。综上所述，腿长控制规律可以写作如下形式：

$$e_L = L_d - L_0 \quad (79)$$

$$F_{yd1} = K_p e_L + K_i \int_0^t e_L dt + K_d(e_L - e'_L) + m_b g \quad (80)$$

其中， e_L 是腿长的偏差， e'_L 是上一次的腿长偏差； K_p 、 K_i 和 K_d 是控制器超参，由人根据经验整定； m_b 指机器人机体部分的质量，可以由机器人设计时的质量参数给出；

F_{yd1} 即为腿长 PID 加前馈控制器输出得到的期望虚拟力。

在分别完成两侧单腿的长度控制以后，可以通过同步控制两腿长度变化，实现机体的高度变化，高度的可控性由腿长 PID 加前馈闭环控制保证。除此之外，可以通过左右腿长差，来间接控制机体的翻滚角。而为保证翻滚角的控制闭环，可以引入一个新的 PD 控制器，用于对机体的翻滚角实现闭环控制。设机体的翻滚角为 γ ，则翻滚角 PD 控制器可以写成如下形式：

$$e_\gamma = \gamma_d - \gamma \quad (81)$$

$$F_{yd2} = K_p e_\gamma + K_d(e_\gamma - e'_\gamma) \quad (82)$$

故腿部虚拟力期望值为：

$$F_{yd} = F_{yd1} + F_{yd2} \quad (83)$$

5.2.3 偏航角与两腿同步控制

两足轮腿机器人在平面上的运动，可以看作是普通两轮差速底盘的运动。由于驱动轮电机采用的是力矩控制，则意味着，当给左右两个驱动轮施加大小相等方向相反的补偿期望力矩后，机器人即可完成偏航角的控制。为使偏航角控制闭环，引入 PD 控制器。设机体的偏航角为 β ，则偏航角 PD 控制器可以写作如下形式：

$$e_\beta = \beta_d - \beta \quad (84)$$

$$T_{w,offs} = K_p e_\beta + K_d(e_\beta - e'_\beta) \quad (85)$$

在机器人偏航角的控制过程中，会产生一个让两条腿分叉的趋势，对此，进一步引入一个 PD 控制器，对两腿摆角实现闭环控制，具体形式如下：

$$e_q = q_{0r} - q_{0l} \quad (86)$$

$$T_{b,offs} = K_p e_q + K_d(e_q - e'_q) \quad (87)$$

偏航角 PD 控制器的输出 $T_{w,offs}$ 将取相反符号，分别叠加到左右侧 LQR 平衡与运动控制器输出的 T_w 中；同理，摆角同步控制 PD 控制器的输出 $T_{b,offs}$ 将取相反符号，分别叠加到左右侧 LQR 平衡与运动控制器输出的 T_b 中。

5.3 上下台阶分段控制策略设计

机器人由于可编程控制，并且具有良好的抗疲劳性，所以常常被应用在一些结构化的场景中，比如室内的平地 and 台阶场景。对于地面上的运动经过前文的努力已经得到了较好的解决方案，而在台阶场景中，由于台阶的不连续性，若机器人想要完成下台阶的过程，则必然要经历一段腾空的时间，若想实现上台阶，则需要有一段起跳腾空的过程。

在机器人腾空的过程中，驱动轮地面的滚动约束消失，系统动力学模型发生突变，原有控制策略不适用。文献（Klemm et al., 2020）中针对跳跃上台阶情景提出了一种分段控制策略，即将跳跃上台阶的过程，划分为收腿、起跳检测、蹬腿、腾空和着陆共五个阶段，如图 27 所示。

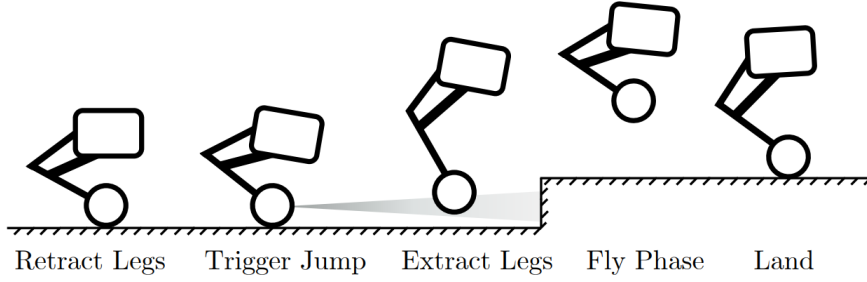


图 27 Ascento 跳台阶分段控制策略（Klemm et al., 2020）

参考上述过程，结合本课题具体情况，下面对于上台阶和下台阶情景分别进行控制方案设计。上台阶的收腿阶段，目的是增加腿部发力行程，提高跳跃高度，采用全身控制中的高度控制，给定机体高度设定值为最小值，即可完成控制。对于蹬腿控制也是类似，先后给定高度设定值为最大值和最小值，并针对性调整控制参数，提高腿部执行速度，即可达到快速蹬腿和收腿的效果。

在起跳检测时，本课题中将采用人为判断的方式，当操作员判断当前可以起跳，即通过遥控或键盘等设备输入指令，让机器人起跳。而下台阶时，起跳检测应变成离地检测，即检测当机器人驱动轮离开台阶，开始腾空的时刻。本课题采用支持力逆解算的方法，通过关节电机的力矩反馈，计算得到当前机器人受到的来自地面的支持力大小，与给定阈值比较，若支持力小于阈值，即可判断机器人离开台阶。

现对地面支持力解算算法原理进行详细阐述。在表 2 所示的符号系统基础上，加入地面对轮子的支持力 F_N 以及轮子中心的竖直位移 y_w 。对轮子在竖直方向上使用牛顿第二定律分析得到：

$$F_N - m_w g - N_{wy} = m_w \ddot{y}_w \quad (88)$$

其中，轮子中心的竖直加速度 \ddot{y}_w 无法直接通过安装传感器直接得到，需要利用机体的搭载的加速计和惯性测量单元解算出的机体在竖直方向加速度 \ddot{y}_b ，以及机体倾斜角度 θ 和腿长 L_0 及他们对时间的一、二阶导数，通过下式计算得到。

$$\begin{aligned}\ddot{y}_w &= \frac{d^2}{dt^2}(y_b - L_0 \cos(\theta)) \\ &= \ddot{y}_b - \ddot{L}_0 \cos(\theta) + L_0 \ddot{\theta} \sin(\theta) + L_0 \dot{\theta}^2 \cos(\theta) + 2\dot{L}_0 \dot{\theta} \sin(\theta)\end{aligned}\quad (89)$$

腿部虚拟连杆对于轮子的等效支持力竖直分量 N_{wy} 则可以通过腿部虚拟力反馈 F_{yf} 和虚拟反馈力矩 T_{bf} 的逆解算得到：

$$N_{wy} = F_{yf} \cos(\theta) + \frac{T_{bf}}{L_0} \sin(\theta) \quad (90)$$

轮子质量 m_w 可以通过设计参数或者实际测量得到，故可以解得地面支持力：

$$F_N = m_w(\ddot{y}_w + g) + N_{wy} \quad (91)$$

当机器人经过起跳腾空或者是离开台阶腾空后，机器人轮子与地面的无滑动滚动理想约束消失，原动力学模型(48)失效，系统处于部分不可控状态。考虑到机器人落地后的理想姿态应当是虚拟腿处于竖直状态，故在空中，将状态反馈矩阵中除了用于控制虚拟腿与地面倾角和倾角角速度之外的系数置零，如式(92)，即可实现效果。

$$\mathbf{K} = \begin{bmatrix} 0 & 0 & 31.286 & 6.041 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (92)$$

5.4 系统控制框架结构

至此，机器人运动控制框架已基本成型，系统控制结构如图 28 所示。整个系统共有四个给定输入，分别是机体偏航角设定值 y_d 、机体翻滚角设定值 r_d 、机体高度设定值 h_d 、系统状态向量设定值 \mathbf{x}_d ，其中，除了机体前进位移给定值为 x_d ，其他都是给定为平衡零点。

图 28 中橙色线路是传感器的反馈通路，利用机载惯性测量传感器和电机的编码器可以综合得到用于系统平衡反馈控制的机体俯仰角、虚拟连杆角和驱动轮水平位移以及它们对时间的一阶导数共六个状态变量 \mathbf{x} ，也可以得到用于全身运动控制的机体的偏航角 y_f 、翻滚角度 r_f 、高度 h_f 以及腿部虚拟模型中的关节角度 ${}^l q_f$ 、 ${}^r q_f$ 和腿长 ${}^l L_0$ 、 ${}^r L_0$ 。

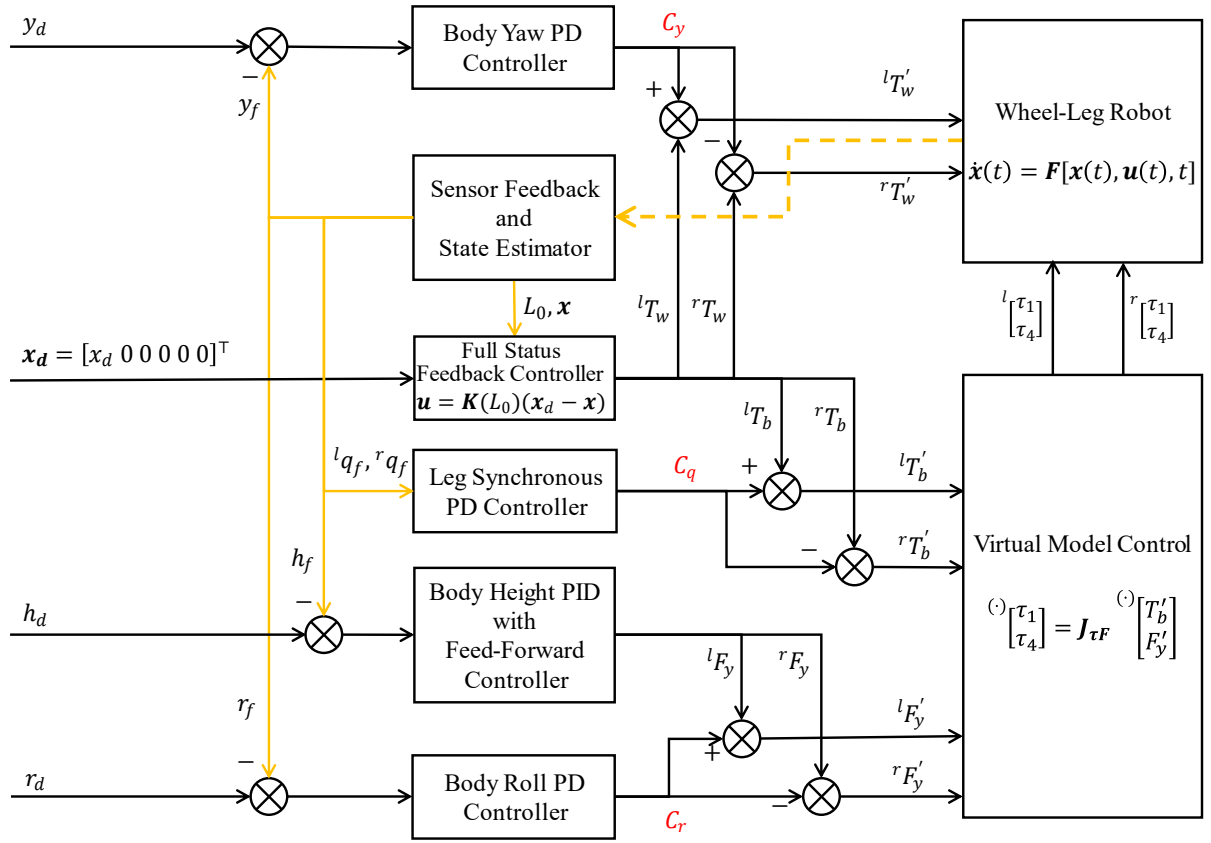


图 28 综合运动控制系统框图

控制对象是一个非线性时变系统。机器人的全身控制器的核心是反馈增益关于腿长自适应的全状态反馈控制器（Full Status Feedback Controller），主要负责机器人的平衡控制。在此基础上，增加了机体的偏航角 PD 控制器（Body Yaw PD Controller）、翻滚角 PD 控制器（Body Roll PD Controller）、高度 PID 与前馈控制器（Body Height PID with Feed-Forward Controller）以及两腿同步 PD 控制器（Leg Synchronous PD Controller）。在腿部的动力学控制方面，采用了虚拟模型控制（Virtual Model Control），避免了直接的五连杆机构动力学建模。

控制器周期执行过程中，根据反馈得到的左右侧虚拟腿长 lL_0 、 rL_0 ，更新状态反馈增益 \mathbf{K} ，然后由全状态反馈控制得到左右虚拟腿的参考力矩和驱动轮的参考力矩。

为实现全身运动控制，左右腿的驱动轮参考力矩 lT_w 、 rT_w 需要加上符号相反的机体偏航角控制器输出值 C_y 得到新的参考力矩 ${}^lT'_w$ 、 ${}^rT'_w$ ，交给驱动轮电机执行。

虚拟腿的参考力矩 lT_b 、 rT_b 需要加上符号相反的两腿同步控制器输出值 C_q 得到新的虚拟腿参考力矩 ${}^lT'_b$ 、 ${}^rT'_b$ 。

另外，通过机体高度控制器输出值 lF_y 、 rF_y 分别叠加上符号相反的翻滚角控制器输出值 C_r ，得到腿部虚拟力 ${}^lF'_y$ 、 ${}^rF'_y$ 。然后将左右腿部虚拟力和力矩 ${}^lF'_y$ 、 lT_b 和 ${}^rF'_y$ 、 rT_b 分别通过腿部虚拟模型，即力的雅可比矩阵 $J_{\tau F}$ 映射，得到关节电机真实的参考力矩 ${}^l\begin{bmatrix} \tau_1 \\ \tau_4 \end{bmatrix}$ 和 ${}^r\begin{bmatrix} \tau_1 \\ \tau_4 \end{bmatrix}$ 。

总的来说，该机器人控制系统，接受 4 个人为给定量 y_d 、 r_d 、 h_d 和 x_d ，经过全身运动控制器计算与虚拟模型的转化得到全身共 4 个关节电机和 2 个驱动轮电机的参考力矩 ${}^l\begin{bmatrix} \tau_1 \\ \tau_4 \end{bmatrix}$ 、 ${}^r\begin{bmatrix} \tau_1 \\ \tau_4 \end{bmatrix}$ 和 ${}^lT'_w$ 、 ${}^rT'_w$ ，然后由电机内置的力矩闭环完成参考力矩的实现。

5.5 本章小结

本章在第 4 章的基础上，进一步完善了机器人的控制器，将机器人的运动能力从一维直线运动拓展到了三维空间中的多自由度运动；针对上下台阶的典型场景设计了分段控制策略；综合平衡控制器和全身运动控制器构成了完整的机器人控制器结构。

6 基于 Webots 的物理仿真

本章将对前文设计的 LQR 平衡控制器、五连杆动力学模型以及全身运动控制器的控制效果进行仿真验证。

为保证仿真过程引入的变量可控，便于问题的定位，仿真过程将根据仿真模型的复杂度，分为三个阶段，不同阶段之间的区别在于腿部结构的不同。分阶段独立设计物理模型，有针对性，有递进性，逐步完成所有算法的验证工作。

6.1 Webots 仿真环境介绍

Webots 是由 Cyberbotics 公司开发的一款开源跨平台的机器人仿真软件，其提供了强大而完整的开发环境，基于 ODE 物理仿真引擎，提供较为真实的动力学仿真效果。软件具有机器人建模、编程和仿真功能，可用于开发、测试和验证各种机器人算法，并且该软件是免费开源的，拥有完备的技术文档，易于使用。

Webots 中的对象使用 OpenGL 渲染，呈现优美的视觉效果。此外，Webots 还支持将仿真过程导出为动画或交互式 HTML 文件，用户可以借此更好地学习和研究机器人领域相关知识，并优化各种机器人算法的性能。

6.2 平衡系统物理仿真

第一阶段的建模，主要侧重于验证在 LQR 平衡控制器的作用下，机器人能否维持

平衡态，并且在受到一定程度的外界扰动后能够自动回到平衡态。为突出重点，建模只保留机体、虚拟连杆和轮子三个部分，并且轮子的宽度设计为与机体等宽，尽可能保证机器人的运动在同于竖直平面内，同时，采用虚拟连杆进行仿真，也能够暂时排除五连杆动力学模型可能存在的问题。

仿真验证了，在没有外界扰动的情况下，LQR 平衡控制器能够保证机器人维持稳态，并且在机器人运动平面内，施加短暂或持续性外力干扰时，机器人都能够抵抗扰动，并最终回到稳态。

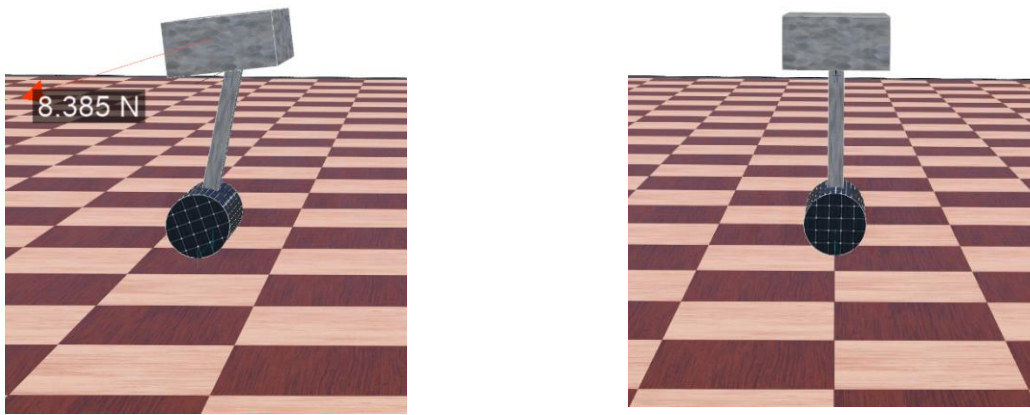


图 29 自平衡抗扰效果

6.3 平衡系统与腿部动力学控制物理仿真

第二阶段的仿真，侧重于对腿部五连杆动力学模型进行验证。建模方面，在第一阶段的模型基础上，将虚拟连杆替换为五连杆机构，其他参数均不改变；代码方面，在第一阶段的 LQR 平衡控制器的基础上，其输出的虚拟关节期望输出力矩，通过五连杆动力学模型，解算出关节电机期望输出力矩，再交由关节电机执行，并引入了腿长控制 PID 控制器。

从仿真结果可以看出，机器人能够抵抗扰动，恢复平衡态。在第一阶段验证了 LQR 平衡控制器有效性的前提下，本阶段仿真结果，验证了腿部五连杆动力学模型以及腿长控制 PID 器的有效性。

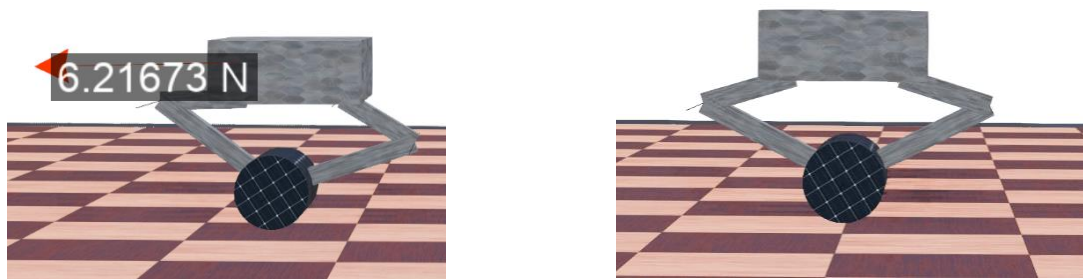


图 30 五连杆动力学模型验证

6.4 全身运动控制物理仿真

第三阶段的仿真，重点在于验证全身运动综合控制效果。建模按照机器人完整结构特点进行，机体和单侧腿部参数与第二阶段完全一致，但是轮子的宽度做了修改，为扁平的轮子，贴近真实情况。为便于测试，在仿真环境中引入了键盘输入，即可以通过键盘来控制机器人完成指定的运动。仿真的主要代码参考附录 2。

从下列仿真结果可以看出，如图 28 所示的全身运动综合控制器能够控制机器人完成直线运动、偏航运动、翻滚运动和蹲起运动，并且在运动的过程中，能够保证系统的稳定性。

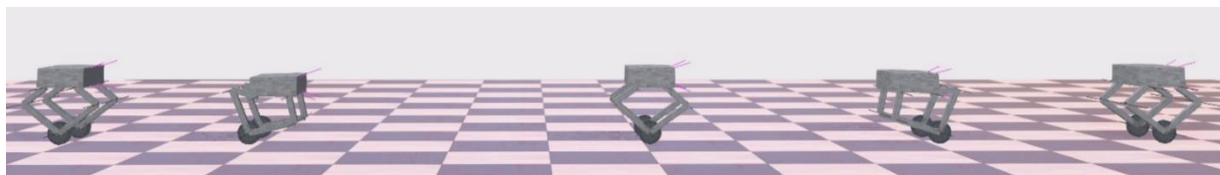


图 31 直线运动控制（静止、加速、匀速、减速和停止过程）

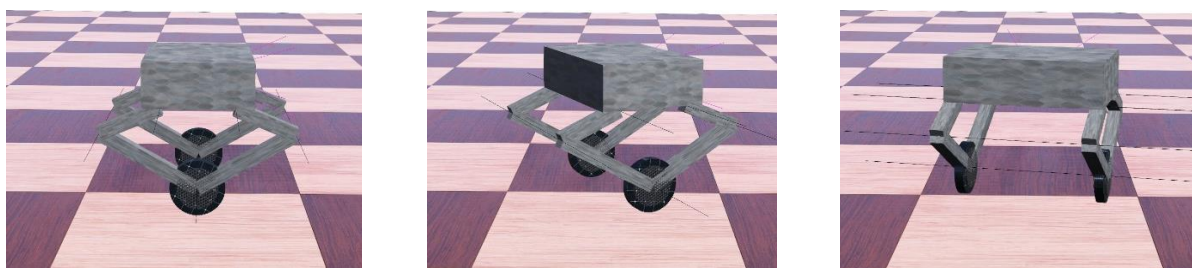


图 32 偏航角控制



图 33 翻滚角控制

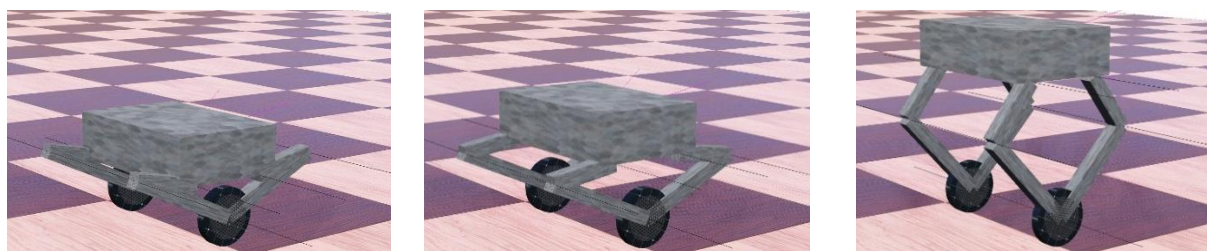


图 34 高度控制

仿真通过键盘进行过程控制，实现了跳跃上台阶，和下台阶（飞坡）的控制仿真，验证了分段控制策略的可行性。

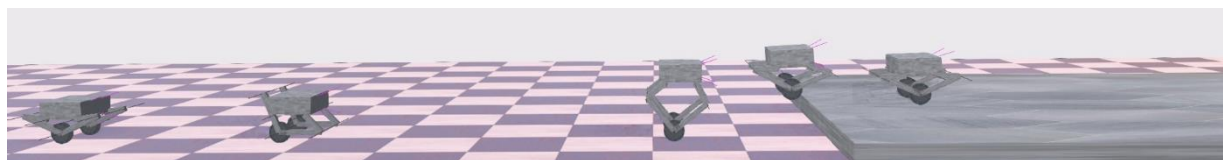


图 35 上台阶过程控制（静止、加速、蹬腿、收腿腾空、落地过程）

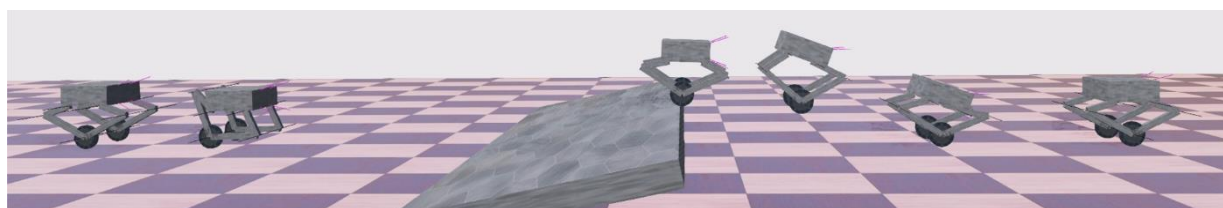


图 36 飞坡（下台阶）控制（静止、加速、腾空检测、腾空、落地缓冲、停止过程）

同时，仿真中还测试了单边桥和斜坡两种特殊地形下的自平衡效果，验证了控制器的鲁棒性。

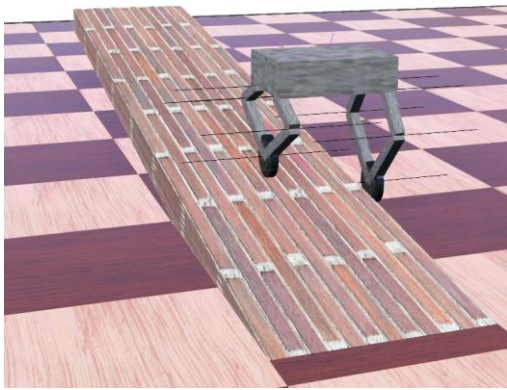


图 37 单边桥上自平衡

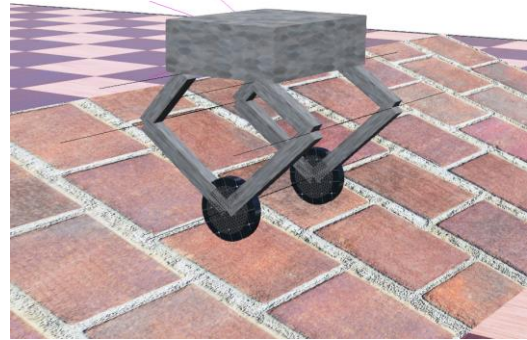


图 38 斜坡上自平衡

6.5 本章小结

本章通过三个阶段的针对性仿真，验证了 LQR 平衡控制器、五连杆动力学模型以及全身运动控制器的效果。机器人在全身运动控制器的作用下，能够较好地完成由键盘输入的运动指令，并且保证全程系统的稳定性；在机器人上下台阶的过程中，分段控制策略的效果也得到了验证。

7 总结与展望

7.1 全文工作总结

在机器人技术逐渐普及的今天，机器人已经逐渐应用于人类生产生活的方方面面，而移动机器人能够具有移动能力，代替人类完成许多的工作。轮腿式机器人结合了传统的轮式和足式机器人的优点，能够快速移动并具有良好的地形通过能力。

本文主要研究的对象是腿部为五连杆结构的双足轮腿式机器人，主要研究工作总结如下：

- (1) 国内外在轮腿式机器人领域的现状研究。
- (2) 腿部五连杆机构的运动学与动力学模型的建立。首先独立出了五连杆模型，建立了直角坐标系，随后依据几何关系建立三个从动关节坐标与关节角度之间的位置关系，然后利用等式求导，得到足端从动关节在坐标系中的线速度与关节角速度之间的雅可比矩阵关系；在此基础上，利用虚功原理，进一步推导出了足端力（矩）到关节电机力矩的映射关系。
- (3) 机器人动力学模型的简化、建立、分析以及线性化。首先根据五连杆的运动

学与动力学模型，将腿部等效为长度可变的虚拟连杆与虚拟关节。然后再对机器人的单侧使用牛顿-欧拉（Newton-Euler）矢量力学方法进行动力学建模，得到了系统的非线性动力学模型，再引入了状态变量，分析了系统的平衡点，最后，将模型在平衡点处利用雅可比矩阵完成线性化，得到系统的状态方程符号表达式。

（4）基于 LQR 的平衡控制器的设计与仿真验证。将机器人常见的几何尺寸、质量与转动惯量，代入到系统的状态空间符号表达式中，得到系统的状态矩阵和输入矩阵；根据经验，设置代价权重矩阵 Q 和 R ，通过求解黎卡提方程，进一步计算得到最优控制状态反馈增益矩阵 K ，即完成 LQR 平衡控制器设计。随后在 Webots 中搭建了简化模型，验证了控制器的有效性。

（5）机器人全身运动控制器的设计与仿真验证。在平衡控制的基础上，首先在原控制律中，引入参考状态向量，实现系统的位置跟随控制。然后通过连续修改腿长参数，计算不同腿长的情况下对应的反馈矩阵 K ，再利用多项式拟合，得到了各个矩阵元关于腿长的函数关系，实现平衡控制器对于腿长参数的自适应功能。随后加入辅助翻滚角、偏航角以及两腿同步的 PID 或 PD 控制器，进一步实现机器人的全身运动控制。最后，在 Webots 搭建了完整的模型，针对性的验证了全身运动控制器的有效性。

7.2 未来工作展望

受限于仿真环境与其他客观条件，本文的研究尚且还具有一些局限性，在未来仍然有许多可以优化或进一步研究的内容，主要有以下四个方面：

（1）在建模方面，是将机器人的单侧进行单独建模，全身运动控制只是考虑了在同一时间内，只执行一种运动的控制，如在直线行驶时，不考虑偏航角控制这类的耦合运动情况。所以，实际机器人在进行耦合运动时，效果并非最好，还有极大的优化空间，比如尝试对机器人全身进行动力学建模，再针对性设计控制器。

（2）在成果验证方面，将全身运动控制器（算法）与地面支持力解算等算法在实际机器人平台上进一步验证。仿真环境中的许多条件过于理想，比如，地面摩擦力设置为无穷大、各关节均为理想无摩擦关节，传感器的忽略了噪声的影响，控制器运行的速度仅受电脑性能的影响，远超实际控制器的性能等等，所以仿真结果不代表实际应用效果。

（3）在控制算法方面，本文提出的全身运动控制框架，无法对机器人在入弯时提

前进行向心加速度倾角补偿，仿真时，机器人在入弯体现出离心倾倒的趋势，机器人运动稳定性有待进一步优化。

（4）在应用方面，轮腿机器人本身只能作为一个移动载具，可以在其身上搭载其他执行机构或传感器，比如机械臂、激光雷达和双目摄像头等，以实现更加复杂的功能或者实现更好的地形自适应能力。

参 考 文 献

- 国家市场监督管理总局, 中国国家标准化管理委员会. GB/T 39405-2020 机器人分类[S]. 2020.
- 王新生, 曲延滨. 现代控制理论基础[M]. 第3版. 哈尔滨: 哈尔滨工业大学出版社, 2020.
- 谢惠祥. 四足机器人对角小跑步态虚拟模型直觉控制方法研究[D]. 国防科学技术大学, 2015.
- Ascento. This is ascento pro[EB/OL]. (2021-12-06)[2023/5/5]. <https://youtu.be/Uxt2wTI0m5o>.
- Boston. Introducing handle[EB/OL]. (2017-02-28)[2023]. <https://youtu.be/-7xvqQeoA8c>.
- Boston. Handle robot reimagined for logistics[EB/OL]. (2019-03-29)[2023/5/5]. https://youtu.be/5iV_hB08Uns.
- C., T., S., et al. System design and balance control of a bipedal leg-wheeled robot[A]2019 IEEE International Conference on Robotics and Biomimetics (ROBIO)[C], 2019
- H., I., T., et al. Walking-wheeling dual mode strategy for humanoid robot, drc-hubo+[A]2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)[C], 2016
- J., I., O., et al. Robotic software system for the disaster circumstances: system of team kaist in the darpa robotics challenge finals[A]2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)[C], 2015
- K., T., Y., et al. Realization by biped leg-wheeled robot of biped walking and wheel-driven locomotion[A]Proceedings of the 2005 IEEE International Conference on Robotics and Automation[C], 2005
- Klemm, Morra, Salzmann, et al. Ascento: a two-wheeled jumping robot[Z]. Ithaca: Cornell University Library, arXiv.org, 2020.
- Kollarčík, Gurtner. Modeling and control of two-legged wheeled robot[A], 2021
- Li, Zhou, Feng, et al. Design and experiments of a novel hydraulic wheel-legged robot (wlr)[A], 2018
- Li, Zhou, Zhang, et al. Wlr-ii, a hose-less hydraulic wheel-legged robot[A], 2019
- Matsumoto, Kajita, Saigo, et al. Dynamic trajectory control of passing over stairs by a biped type leg-wheeled robot with nominal reference of static gait[J]. 1998.

- Oh, Sohn, Jang, et al. Technical overview of team drc - hubo@unlv's approach to the 2015 darpa robotics challenge finals[J]. J. Field Robot., 2017,34(5):874-896.
- Pratt, Chew, Torres, et al. Virtual model control: an intuitive approach for bipedal locomotion[J]. The International Journal of Robotics Research, 2001,20(2):129-143.
- S., J., B., et al. Feedback control for autonomous riding of hovershoes by a cassie bipedal robot[A]2019 IEEE-RAS 19th International Conference on Humanoid Robots (Humanoids)[C], 2019
- S., L., J., et al. Balance control of a novel wheel-legged robot: design and experiments[A]2021 IEEE International Conference on Robotics and Automation (ICRA)[C], 2021
- Y., R., X., et al. Feedback control of a cassie bipedal robot: walking, standing, and riding a segway[A]2019 American Control Conference (ACC)[C], 2019
- Y., T., Y., et al. Realization of dynamic human-carrying walking by a biped locomotor[A]IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA '04. 2004[C], 2004

附录

附录 1 系统非线性动力学数学模型

$$\begin{aligned}
 & R x_4^2 \sin(x_3) (L m_b + L_l m_l) + R L_b m_b (x_6' \cos(x_5) - x_6^2 \sin(x_5)) + T_w \\
 & = R x_4' \cos(x_3) (L m_b + L_l m_l) \\
 & + \frac{x_2' (J_w + R^2 (m_l + m_w))}{R}
 \end{aligned} \tag{93}$$

$$\begin{aligned}
 & \sin(x_3) \left(L m_b \left((L_b - L_l + L) (-x_4' \sin(L_b \cos(x_5) + x_3) \right. \right. \\
 & + L_b (x_6' \sin(x_5) \sin(L_b \cos(x_5) + x_3) \\
 & + x_6^2 \cos(x_5) \sin(L_b \cos(x_5) + x_3) \\
 & + 2 x_4 x_6 \sin(x_5) \cos(L_b \cos(x_5) + x_3)) \\
 & + x_4^2 (-\cos(L_b \cos(x_5) + x_3)) \\
 & \left. - x_6^2 L_b^2 \sin^2(x_5) \cos(L_b \cos(x_5) + x_3)) + g \right) \\
 & + L_l m_l (g + L_l (x_4' \sin(x_3) + x_4^2 \cos(x_3))) \Big) \\
 & + \cos(x_3) (-x_4' \cos(x_3) (L^2 m_b + L_l^2 m_l) \\
 & + x_4^2 \sin(x_3) (L^2 m_b + L_l^2 m_l) \\
 & + L L_b m_b (x_6' \cos(x_5) - x_6^2 \sin(x_5)) - L_l m_l x_2') + T_b \\
 & = J_l x_4' + T_w
 \end{aligned} \tag{94}$$

$$\begin{aligned}
 & L_b m_b \sin(x_5) \left((L_b - L_l + L) (-x_4' \sin(L_b \cos(x_5) + x_3) \right. \\
 & + L_b (x_6' \sin(x_5) \sin(L_b \cos(x_5) + x_3) \\
 & + x_6^2 \cos(x_5) \sin(L_b \cos(x_5) + x_3) \\
 & + 2 x_4 x_6 \sin(x_5) \cos(L_b \cos(x_5) + x_3)) \\
 & + x_4^2 (-\cos(L_b \cos(x_5) + x_3)) \\
 & \left. - x_6^2 L_b^2 \sin^2(x_5) \cos(L_b \cos(x_5) + x_3)) + g \right) \\
 & + L_b m_b \cos(x_5) (L_b (x_6^2 \sin(x_5) - x_6' \cos(x_5)) \\
 & + L x_4' \cos(x_3) - L x_4^2 \sin(x_3)) + T_b - J_b x_6' = 0
 \end{aligned} \tag{95}$$

$$x_2 = x_1' \tag{96}$$

$$x_4 = x_3' \tag{97}$$

$$x_6 = x_5' \tag{98}$$

附录 2 全身运动仿真 Webots 主要代码

```
int main(int argc, char** argv) {  
    wb_robot_init(); //仿真系统初始化  
  
    .....  
  
    while ( wb_robot_step(TIME_STEP) != -1 ) { //主循环  
        a.stable_cnt++;  
  
        //机体传感器更新  
  
        const double* rpy = wb_inertial_unit_get_roll_pitch_yaw(imu); //读取机体俯仰角  
        const double* bodyOmega = wb_gyro_get_values(gyroBody); //读取机体俯仰角速度  
        const double* bodyAcc = wb_accelerometer_get_values(accelerometerBody);  
  
        .....  
  
        //键盘遥控器输入  
  
        if ( a.stable_cnt > STABLE_TIME ) {  
            int key = wb_keyboard_get_key();  
            switch ( key ) {  
                //case 'W': ADD(a.x_set, 0.0022, 10000); break;  
                //case 'S': SUB(a.x_set, 0.0022, -10000); break;  
                case 'W': ADD(a.v_set, 0.004, 2); break;  
                case 'S': SUB(a.v_set, 0.004, -2); break;  
  
                case 'A': ADD(a.omega_set, 0.008, PI); break;  
                case 'D': SUB(a.omega_set, 0.008, -PI); break;  
                case 'Q': SUB(a.high_set, 0.0003, 0.07); break;  
                case 'E': ADD(a.high_set, 0.0003, 0.33); break;  
                case 'Z': SUB(a.roll_set, 0.0005, -PI / 9); break;  
                case 'C': ADD(a.roll_set, 0.0005, PI / 9); break;  
                case 'F': a.down_stair = 1; break; //手动判断开始下台阶-将腾空
```

```

case 'G': a.down_stair = 0; break; //手动判断完成下台阶-已落地
case WB_KEYBOARD_UP: a.jump_flag = 1; break; //起跳
case WB_KEYBOARD_DOWN: a.fall_flag = 1; break; //判断落地， 功能同 G
case WB_KEYBOARD_LEFT: a.test_flag = 1; break;
case WB_KEYBOARD_RIGHT: a.test_flag = 0; break;
case 'R':
    //RTN(a.x_set, 0, 0.004);
    RTN(a.v_set, 0, 0.004);
    RTN(a.omega_set, 0, 0.2);
    RTN(a.high_set, legLengthRef, 0.0003);
    RTN(a.roll_set, 0, 0.0005);
    break;
default:
    //RTN(a.x_set, s[0].x1, 0.002);
    RTN(a.v_set, 0, 0.005);
    RTN(a.omega_set, 0, 0.01);
    break;
}
}

//全状态反馈控制
for ( int i = 0; i < 2; ++i ) {
    //腿部 VMC 解算
    s[i].q1 = wb_position_sensor_get_value(s[i].posSensor1) + PI;
    s[i].q4 = wb_position_sensor_get_value(s[i].posSensor4);
    vmc_calculate(&s[i].vmc, s[i].q1, s[i].q4, 0, 0);

    //读取轮子的角速度

```

```

s[i].wheelOmega = wb_gyro_get_values(s[i].gyroWheel);

//状态观测
s[i].x2 = -s[i].wheelOmega[2] * wheelRadius;
s[i].x1 += (s[0].x2 + s[1].x2) / 2 * TIME_STEP * 0.001; //位移需要转化成机体中心
的位移
s[i].x5 = -rpy[1];
s[i].x3 = -(s[i].x5 - s[i].vmc.q0 + PI / 2);
s[i].x4 = (s[i].x3 - s[i].last_x3) / TIME_STEP * 1000;
s[i].last_x3 = s[i].x3;
s[i].x6 = -bodyOmega[1];

//整合状态变量（与期望）得到状态向量
if ( a.stable_cnt > STABLE_TIME ) { //待系统稳定
    //起跳状态机
    switch ( a.jump_flag ) {
        case 0:
        {
            a.fall_flag = 0;
            break;
        }
        case 1: //蹬腿
        {
            a.high_set = 0.3;
            if ( a.high_fdb > 0.28 ) {
                a.jump_flag++;
            }
            break;
        }
    }
}

```

```

    }
    case 2: //收腿
    {
        a.fall_flag = 0;
        a.high_set = 0.1;
        if ( a.high_fdb < 0.12 ) {
            a.jump_flag++;
        }
        break;
    }
    case 3: //落地检测
    {
        if ( a.fall_flag ) {
            a.jump_flag = 0;
        }
        break;
    }
    default: break;
}

//控制器参数自适应
kUpdateBaseOnLeg(s[i].vmc.L0, s[i].K, KL); //腿长适应
if ( a.down_stair || a.jump_flag == 2 || a.jump_flag == 3 ) { //滞空适应
    double K0_array[2 * 6] = {
        0, 0, matrix_get(s[i].K, 1, 3), matrix_get(s[i].K, 1, 4), 0, 0,
        0, 0, 0, 0, 0, 0
    };
    matrix_cover(s[i].K, K0_array);
}

```

```

    }
} else { //状态调节
    double X_array[6 * 1] = {s[i].x1, s[i].x2, s[i].x3, s[i].x4, s[i].x5, s[i].x6};
    matrix_cover(s[i].X, X_array);
}

//全状态反馈
matrix_nmul(-1, s[i].K, NULL);
matrix_mul(s[i].K, s[i].X, s[i].U);
}

//全身运动控制
a.q0_offs = pid_calc(&pid_q0, s[1].vmc.q0, s[0].vmc.q0); //双腿摆角同步控制
a.roll_offs = -pid_calc(&pid_roll, rpy[0], a.roll_set); //双腿摆角同步控制
a.yaw_offs = pid_calc(&pid_yaw, bodyOmega[2], a.omega_set); //Yaw 控制

for ( int i = 0; i < 2; ++i ) {
    //起跳腿部力控制
    a.high_fdb = s[0].vmc.L0;
    if ( a.jump_flag == 1 ) {
        s[i].Fy = pid_calc(&s[i].legLengthPid, s[i].vmc.L0, a.high_set) + mb*g/2 + 150; //跳
        跃的蹬腿阶段，快速起跳
    } else if ( a.jump_flag == 2 ) {
        s[i].Fy = pid_calc(&s[i].legLengthJumpPid, s[i].vmc.L0, a.high_set) - ml*g; //跳跃
        的收腿阶段：PD+前馈，抵消腿的重力
    } else {
        s[i].Fy = pid_calc(&s[i].legLengthPid, s[i].vmc.L0, a.high_set) + mb*g/2; //普通平
        衡时：腿长 PID+前馈，抵消机体重
    }
}

```

```

    }

    //五连杆动力学逆解算
    double F_array[2 * 1] = {(matrix_get(s[i].U, 1, 1) + SIGN(i) * a.q0_offs)/2, s[i].Fy +
SIGN(i) * a.roll_offs}; //两条腿时，LQR 出力矩需要除以 2
    matrix_cover(s[i].F, F_array);
    matrix_mul(s[i].vmc.J, s[i].F, s[i].T);

    //系统输出
    wb_motor_set_torque(s[i].jointMotor1, matrix_get(s[i].T, 1, 1)); //输出关节电机力矩
    wb_motor_set_torque(s[i].jointMotor4, matrix_get(s[i].T, 2, 1)); //输出关节电机力矩
    wb_motor_set_torque(s[i].wheelMotor, matrix_get(s[i].U, 2, 1) + SIGN(i) *
a.yaw_offs); //输出轮子关节力矩
    }
}
wb_robot_cleanup();
return 0;
}

```

致 谢

时光荏苒，四季轮转，四年光阴也将于本文结束之际而步入尾声。蓦然回首，这一路上受到了太多的鼓励与帮助，也收获了无数宝贵的回忆。

感谢金济老师，他广博的知识与严谨的治学态度使我受益匪浅。在他的支持鼓励和悉心指导下，我才能够有机会完成本文。

感谢父母一直以来的支持与肯定，让我在求学路上能够乘风破浪，勇往直前。

特别感谢我们华南农业大学 Taurus 机器人赛队的全体成员，因为遇见你们我才能够有机会不断地突破自我，领会到热爱的真谛，收获一段深刻的体验，相信这段纯粹极致的经历将会一直是我前进道路上的力量源泉！

感谢所有曾经帮助过我的良师益友，以及可能仅有一面之缘的好心人，正因为有你们，才使这条成长的道路上洒满阳光。

最后，愿未来自己始终能够坚持初心，保持热爱，追求极致！