



《机器人技术基础》

Fundamentals of Robotics

研究性专题-四足机器人臂足结合研究

姓 名： 高睿涵

学 号： 23222033

班 级： 机械电子2302

授课时间： 2024-2025学年第一学期

授课教师： 陈光荣

提交日期： 2024年11月01日

摘 要

四足机器人是一种具备出色的多功能性和高度灵活性的机器人系统，其潜在应用领域广泛，包括工业制造、紧急救援、探索和军事任务等领域。我们将深入讨论六自由度机械臂的设计和结构，包括关键构件、材料选择以及运动范围等重要因素。机械臂的设计对四足机器人的性能和功能至关重要。此外，我们还会深入研究机械臂的运动学和动力学，包括逆运动学问题的解决、轨迹规划以及力学分析。这些分析为实现机械臂的精确控制和高效运动提供了基础。

在报告中，我们还将详细探讨四足机器人的控制系统，包括传感器技术、运动控制算法以及反馈控制。特别强调如何有效地整合六自由度机械臂与四足底盘的控制系统，以便完成复杂任务。最后，我们将研究搭载六自由度机械臂的四足机器人在各个领域的潜在应用，其中包括工业自动化、危险环境中的任务执行以及医疗应用等。

关键词： 四足机器人，机械臂，虚拟仿真

摘 要	3
1. 研究内容与意义	5
1.1. 研究内容	5
1.1.1. 机械结构设计	5
1.1.2. 动力学建模及步态规划	5
1.1.3. 传感器融合和控制系统开发	5
1.2. 研究意义	5
2. 研究现状与发展趋势	6
3. 系统总体方案设计	8
3.1. 系统功能分析	8
3.2. 总体方案设计	9
3.3. 设计方案的社会性分析	10
3.3.1. 设计方案方面:	10
3.3.2. 产品应用方面	11
4. 系统硬件设计	11
5. 系统软件设计	12
5.1. 机构简图	12
5.2. Simulink 仿真	13
5.3. 对角步态规划	15
5.4. 足端、关节三次轨迹规划	15
6. 系统综合调试	20
6.1. 系统软硬件综合调试	20
6.2. 系统的经济性分析	20
7. 总结与展望	21
7.1. 总结	21
7.2. 展望	22
参考文献	22
附件 1 程序	25

1. 研究内容与意义

1.1. 研究内容

1.1.1. 机械结构设计

- a) 详细设计机械臂和四足机器人的结构，包括关节、连接件和材料选择。
- b) 进行机械臂和足端的运动学分析，确定各个关节的运动范围和自由度。
- c) 进行机械臂和足端的动力学分析，以了解机械臂的负载和运动要求。

1.1.2. 动力学建模及步态规划

- d) 需要建立机器人的动力学模型，以了解机器人各个部件之间的相互作用，以及在不同任务中的力和力矩分布。
- e) 解决机器人的逆运动学问题，即如何根据所需的末端执行器位置和姿态来控制机械臂和四肢的关节。
- f) 开发机器人的步态规划算法，以实现在不同地形和环境中的稳定行走和移动。

1.1.3. 传感器融合和控制系统开发

- g) 集成各种传感器，如摄像头、激光雷达、惯性传感器等，以感知周围环境和自身状态。这对于导航、避障和任务执行至关重要。
- h) 开发机器人的控制系统，包括运动控制算法、反馈控制和自主导航。这使机器人能够执行各种任务，如抓取、搬运、操作物体等。

1.2. 研究意义

1. **提升任务适应性：** 单臂四足机器人结合了四足机器人的稳定性和机械臂的操作灵活性，可以适应各种复杂环境和任务需求，包括紧急救援、探险探测和工业生产等。

2. **拓宽应用领域：** 这种机器人在危险环境中执行任务，替代人力完成高风险操作，例如火灾救援、核辐射区域探测等。同时，它也可以用于工业生产线上上的自动化操作，提高生产效率。
3. **推动机器人技术发展：** 单臂四足机器人的研究促进了机器人领域关于机械结构、控制算法和传感器技术等方面的创新，推动了机器人技术的不断进步。
4. **人机协作与智能化：** 这种机器人可以实现更高级别的人机协作，提供更智能、更灵活的解决方案，有助于推动人工智能与机器人技术的融合发展。

2. 研究现状与发展趋势

1. A Novel Approach to Robust Control of Robotic Manipulators on Quadruped Platforms

该文献引入了一种新颖的方法，以提高机械臂在四足机器人上的控制鲁棒性，但缺乏实际机器人平台上的验证，需要更多实验数据，因此可以通过增加实验数据以证明自适应控制的实际有效性。

2. Hybrid Robotic System with Quadruped Locomotion and Manipulator Arm for Search and Rescue Missions

该文献描写了一个系统，在搜索和救援任务中表现出很强的潜力，但其机械臂在特定任务中的精确性和灵活性需要改进，可以优化机械臂的末端控制以提高精确性。

3. Design and Control of a Quadruped Robot with Dual Manipulators for Versatile Tasks

该研究提出了一种四足机器人设计，具备双臂机械臂，适用于多种任务，但在某些操作中，机械臂和四足运动之间的协调需要改进，需要改进机械臂和四足运动的协调控制策略。

4. An Adaptive Control Framework for Robotic Arm Mounted on a Quadruped Robot

该文献引入了自适应控制框架，可使机械臂适应不同任务和环境，需要更多实验来验证自适应控制在复杂场景下的效果，增加实验数据以证明自适应控制的

实际有效性。

5. Biomechanics and Control of a Four-Legged Robotic System with Articulated Arm for Cargo Handling

该文献提供了机械臂在货物处理中的生物力学分析，有助于改进机械臂设计，但需要更多的控制策略研究来优化货物处理的性能，可以研究更多的控制算法以提高机械臂在货物处理中的效率。

6. Integrated Control of Four-Legged Robot and Robotic Arm for Manipulation Tasks

该文献提供了一种整合控制方法，使机械臂和四足机器人能够协同执行操作任务，但需要更多实验数据来验证系统的性能和稳定性，增加实验验证以进一步完善整合控制方法。

7. Enhancing Robot Mobility and Manipulation through Combination of Quadruped and Robotic Arm

该文献介绍了一种机械臂与四足机器人结合的方式，增强了机器人的机动性和操作能力，但系统在不同地形和环境中的适应性需要进一步改进，可以优化传感器系统和导航算法以增强机器人的环境适应性。

8. Path Planning and Collision Avoidance for Quadruped Robot with Robotic Arm in Unstructured Environments

该文献介绍了四足机器人与机械臂在非结构化环境中的路径规划和碰撞避免方法，需要更多关于实际机器人场景中的应用案例和性能评估，可以进行更多实际场景的测试以验证路径规划和碰撞避免算法。

9. A Review of Robotic Arm Integration on Quadruped Robots for Aerial and Ground Applications

该文献综合性地回顾了机械臂在四足机器人上的应用，包括地面和空中任务，但缺少详细的技术细节和实验结果，可以提供更多关于具体应用案例和技术细节的信息。

10. Development and Control of a Quadruped Robot with Articulated Robotic Arm for Outdoor Inspection

该文献介绍了一种具有机械臂的四足机器人用于户外检测任务的开发和控制，

但需要进一步改进机械臂的稳定性和控制性能。

11. A review of quadrupedal robots. International Journal of Advanced Robotic Systems

该文献回顾了四足机器人的发展和应用。虽然没有具体介绍六自由度机械臂，但提供了有关四足机器人的广泛信息。

12. Development of a legged robot with a multi-DOF robotic arm for maintenance tasks in nuclear power plants.

该文献关注核电站维护任务中的腿式机器人，搭载多自由度机械臂。文献中详细描述了机器人的设计和应用，但缺乏机械臂的动力学分析。

13. Design and control of a quadruped robot with a 6-DOF manipulator for underwater exploration.

该文献探讨了一款用于水下探测任务的四足机器人，具备六自由度机械臂。研究提到了机器人在水下环境中的应用，但需要进一步测试和改进。

14. Development of a quadruped robot with a 6-DOF robotic arm for precision agriculture.

该研究设计了一款用于精准农业的四足机器人，搭载六自由度机械臂。文献中强调了机器人在农业领域中的潜在价值，但需要更多关于精确性和可靠性的研究。

15. Development of a quadruped robot with a 6-DOF robotic arm for warehouse automation.

该研究设计了一款用于仓库自动化的四足机器人，搭载六自由度机械臂。机械臂在物流任务中表现出色，但需要考虑更高效的路径规划。

3. 系统总体方案设计

3.1. 系统功能分析

我们把四足机器人臂足结合系统的功能进行了综合分析，划分出了以下应具备的功能及其简要介绍：

1. 移动功能：机械狗最基本的要具备稳定的四足行走能力，可以在坚硬地面、不规则地形等不同地形条件下移动，从而使得系统能够进入难以到达的区域执行搜索和救援任务。

2. 抓取功能：即六轴机械臂最基本的要具备抓取物体的能力，可以通过旋转、伸缩等动作来完成抓取任务，允许系统在搜索任务中获取目标物体，或者在救援任务中搬运被困人员或物资。

3. 控制功能：这里既指系统支持的远程遥控功能，以便操作人员能够控制机械狗和机械臂的动作。同时系统也应具备一定的自主控制能力，能够根据环境条件和任务需求做出一些基本的决策，在一些简单情况或极端环境下自主规划救援路线、自主决策等。

4. 通信功能：系统应该具备稳定的通信模块，以便与操作人员进行实时的信息交流，将系统的状态、位置等信息传递给指挥中心或其他操作人员。

3.2. 总体方案设计

1. **设计方案一：集中控制式。**集中控制式方案将机械狗和机械臂的控制整合在一个主控制单元下，简化了控制逻辑，降低了系统的复杂性。由于所有动作由一个主控制单元协调控制，能够更容易实现机械狗和机械臂之间的协调运动，提高了系统的整体性能。同时整体控制单元集中管理，便于故障诊断和维护。

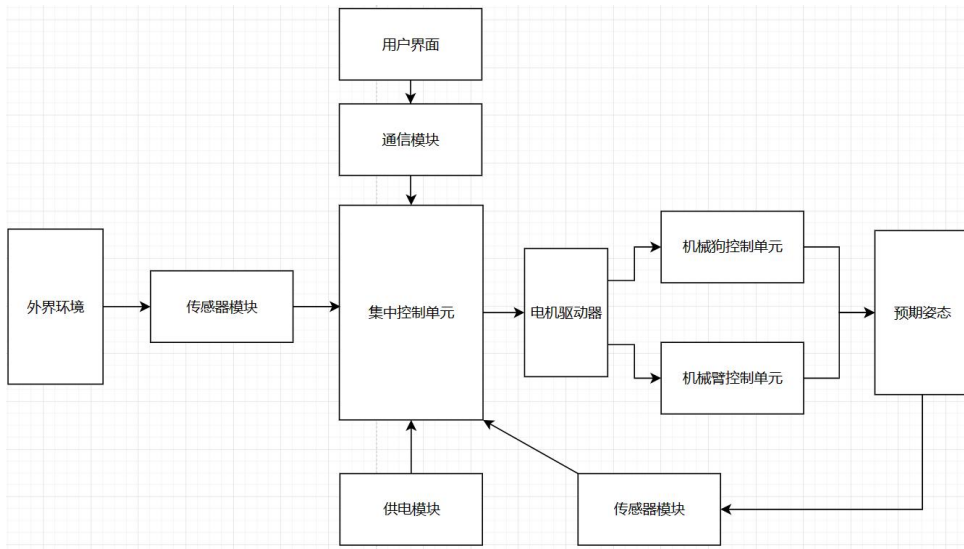
但是如果主控制单元发生故障，整个系统可能会失去控制。另外如果在后期需要增加新的功能或传感器时，可能需要考虑重新设计控制系统。

2. **设计方案二：分布控制式。**分布式控制方案将机械狗和机械臂的控制分布在各自的控制单元下，各自独立运作，降低了单点故障的风险。各个控制单元可以独立设计和扩展，方便后期的功能升级和扩展。这样单个控制单元发生故障就不会影响其他单元的正常运作。

分布式控制需要设计良好的通信和协调机制，确保机械狗和机械臂能够有效地协作。这就使系统的复杂度增加，同时需要更多的硬件和软件设计工作。

这里考虑到机械狗和机械臂之间的协调运动，以及系统的整体性能。我们

选择设计方案一：集中控制式。主要由集中控制单元、传感器模块、通信模块、用户界面等组成。下面为该系统的系统总体设计方框图。



集中控制单元：集中控制单元是系统的核心，负责控制机械狗和机械臂的所有运动和动作。它集成了各种传感器以获取机械狗和机械臂的状态信息，例如位置、速度、力等。通过算法实现对机械狗和机械臂的协调控制。

传感器模块：传感器模块包括在机械狗和机械臂上安装的各种传感器，用于感知周围环境，同时发送自身状态，如每个关节电机的转角等。包括惯性测量单元（IMU）、位置传感器、力传感器等。

通信模块：负责与操作人员或者指挥中心进行通信，接收远程控制指令或者发送状态信息。

用户界面：提供给操作人员的界面，用于远程控制系统、监控状态等。

供电模块：给整个系统提供动力。

3.3. 设计方案的社会性分析

针对所建立的机器狗机械臂救援系统，我们从设计方案和产品应用两个角度来分析其对公众健康、公众安全和环境保护的影响：

3.3.1. 设计方案方面：

a.) 环保材料选择：

在设计方案中，我们优先选择环保材料，选择可持续材料或者具有较低环境影响的材料来构建机械狗和机械臂的外壳和结构。这样就减少了资源消耗和对环境的负担，符合环保的原则。

b.) 元器件选择：

在元器件的选择过程中，优先选择符合环保标准的电子元件和零部件。选择符合 RoHS 等环保认证的元器件，减少有害物质的使用，降低对环境的污染。

c.) 儿童用户安全的相关考虑和设计：

机器狗机械臂救援系统可能会接触到儿童用户，考虑到儿童的安全，设计防护装置、减少尖锐部件、避免小零件脱落等，确保产品在使用过程中对儿童用户的安全。

3.3.2. 产品应用方面

a.) 用户使用注意事项：

在产品的使用说明书或者相关宣传资料中，需要清晰地向用户说明如何正确使用机器狗机械臂救援系统，包括遥控操作、避障注意事项、安全距离等。这样可以帮助用户避免意外事故，保障公众安全。

b.) 报废品处理：

对于故障或报废的机器狗机械臂救援系统，设计方案考虑了如何进行有效的处理。可以提供回收渠道或者指导用户如何进行合适的废弃处理，避免对环境造成不必要的污染或危害。

4. 系统硬件设计

系统主控制器：stm32f103。

传感器：红外线传感器、超声波传感器、触摸传感器

控制元件：机器人每条腿由 2 个电机控制，其中机械臂一共六个关节电机，每个电机由 1 颗 STM32 控制器控制，同时每个电机上又有 1 个编码器用来测试关节位置。

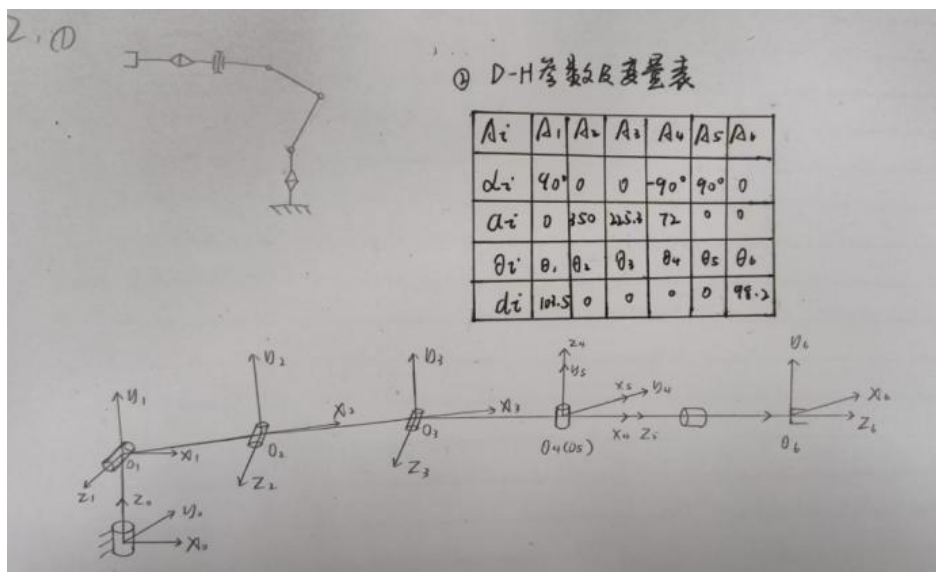
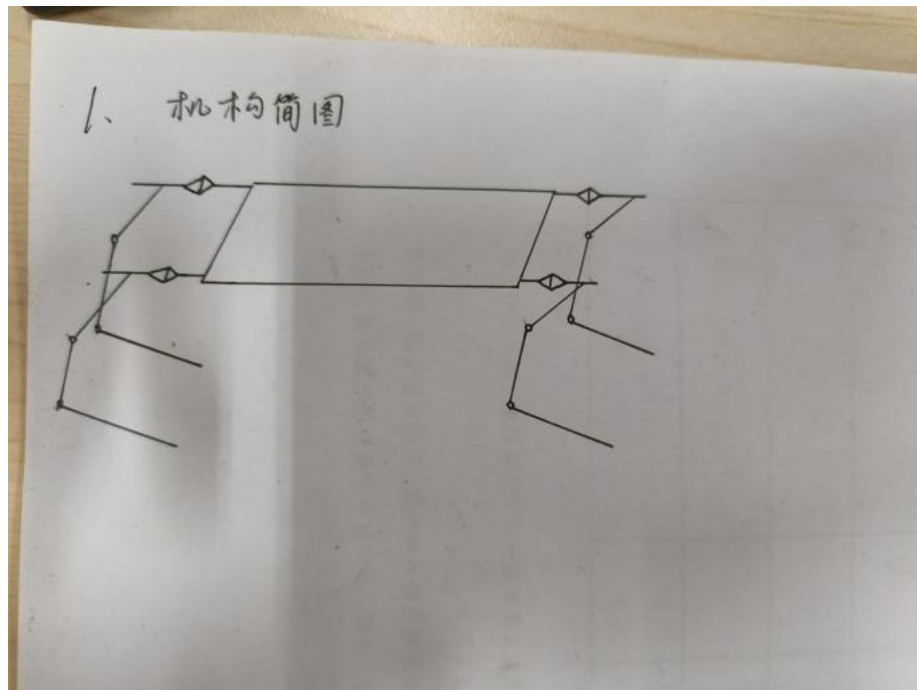
通讯模块：总线驱动板 SPINE，总共 2 个 SPINE，每个 SPINE 控制两条腿。

显示器件：搭载交互式摄像机，通过机器人无线连接电脑，就可以在电脑屏幕上远程显示机器人实时视角画面与坐标位置。

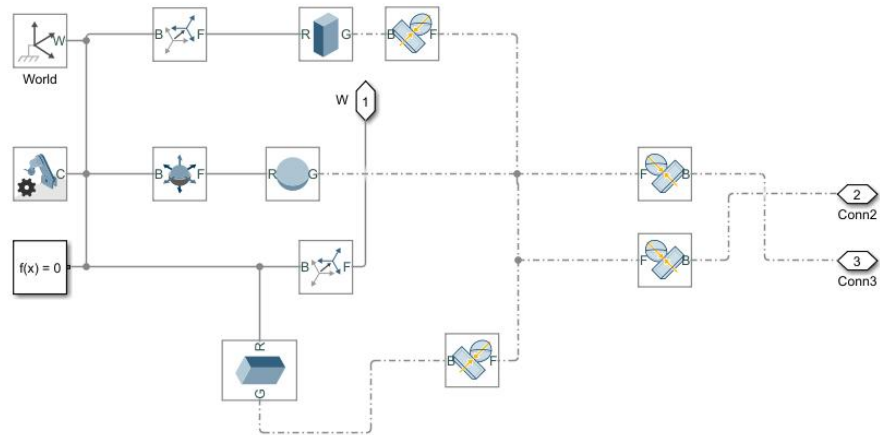
供电模块：使用 18650 电池，组装成 24V 5Ah 的电池组，电源板输出 5V 电压，供电给电机和控制板。

5. 系统软件设计

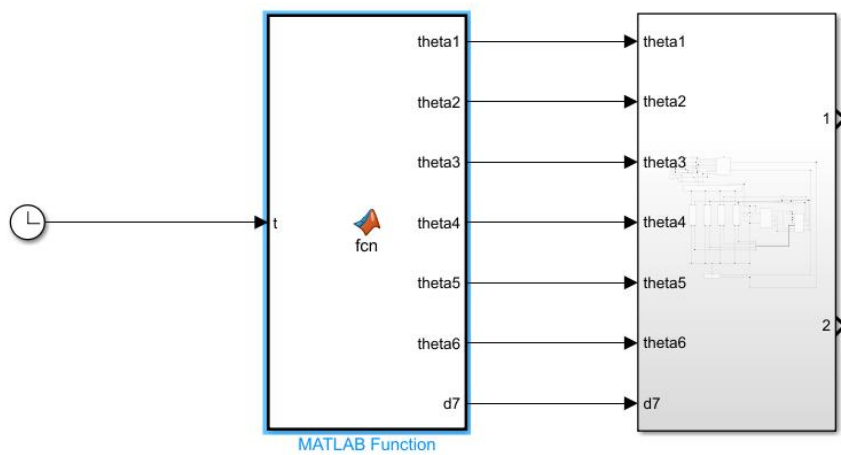
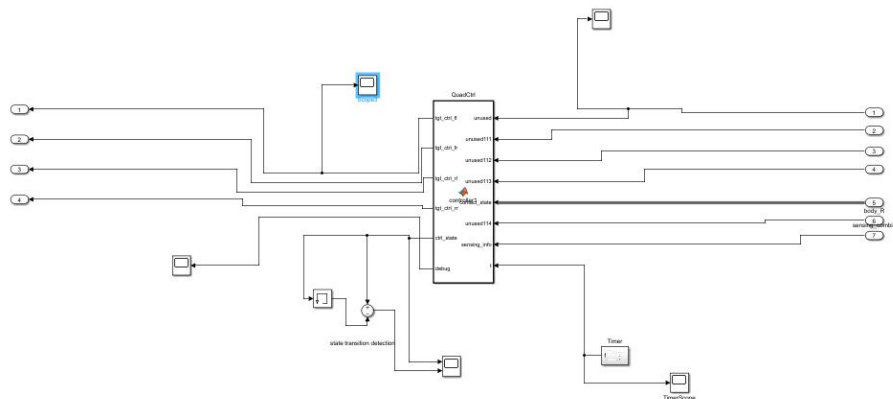
5.1. 机构简图



我们在小球和夹爪之间编写了碰撞，使得仿真更加贴近现实应用场景。

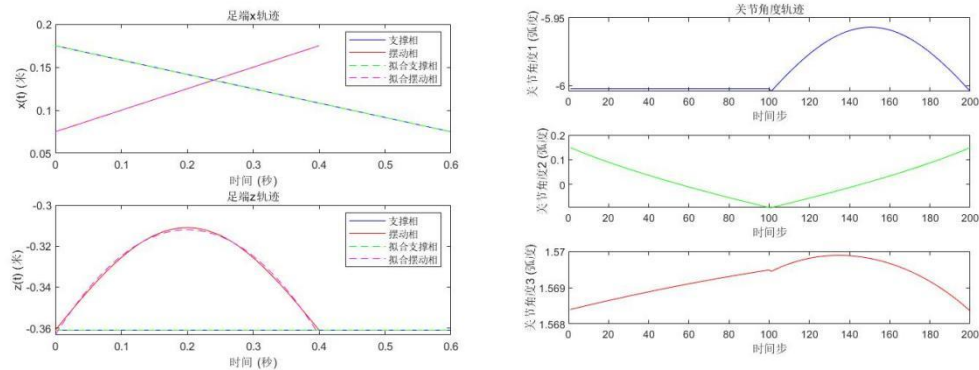


下面是四足机器人和机械臂的虚拟仿真模型。



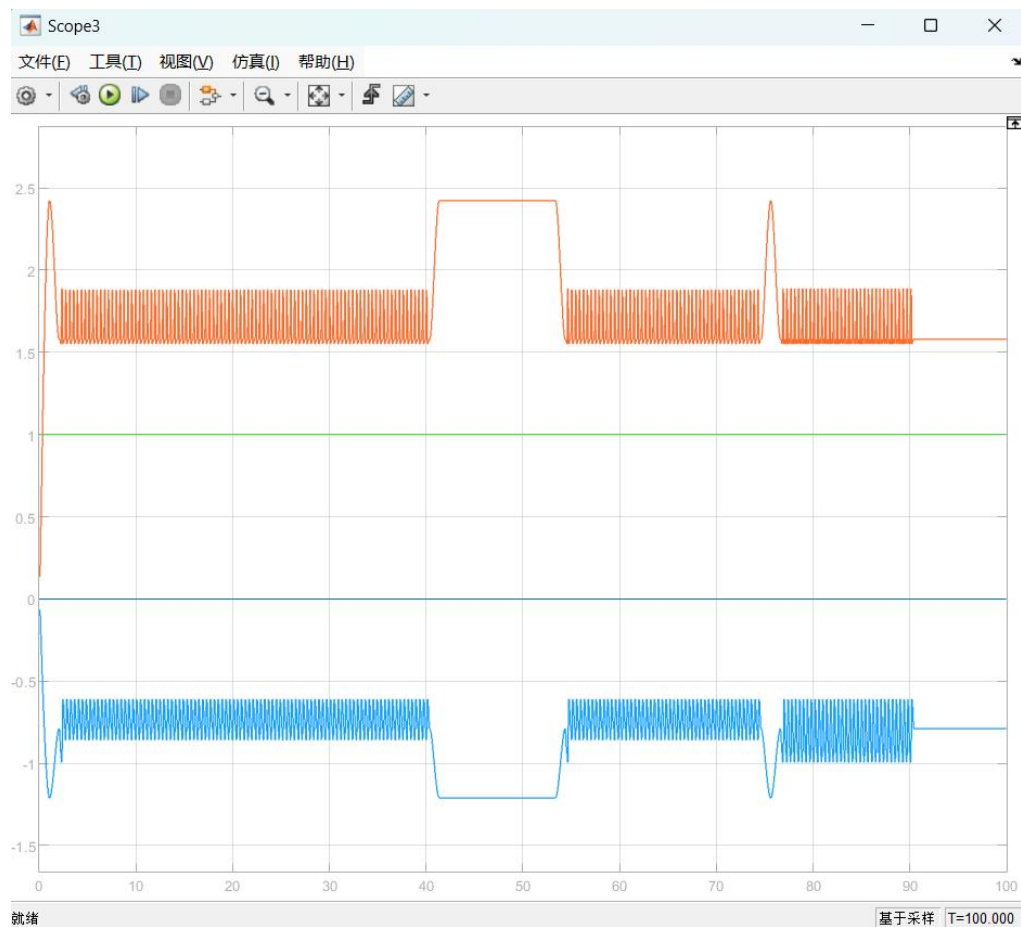
5.3. 对角步态规划

对角步态 (Trot): 对角腿同步, 相邻腿相位差 $T/2$.

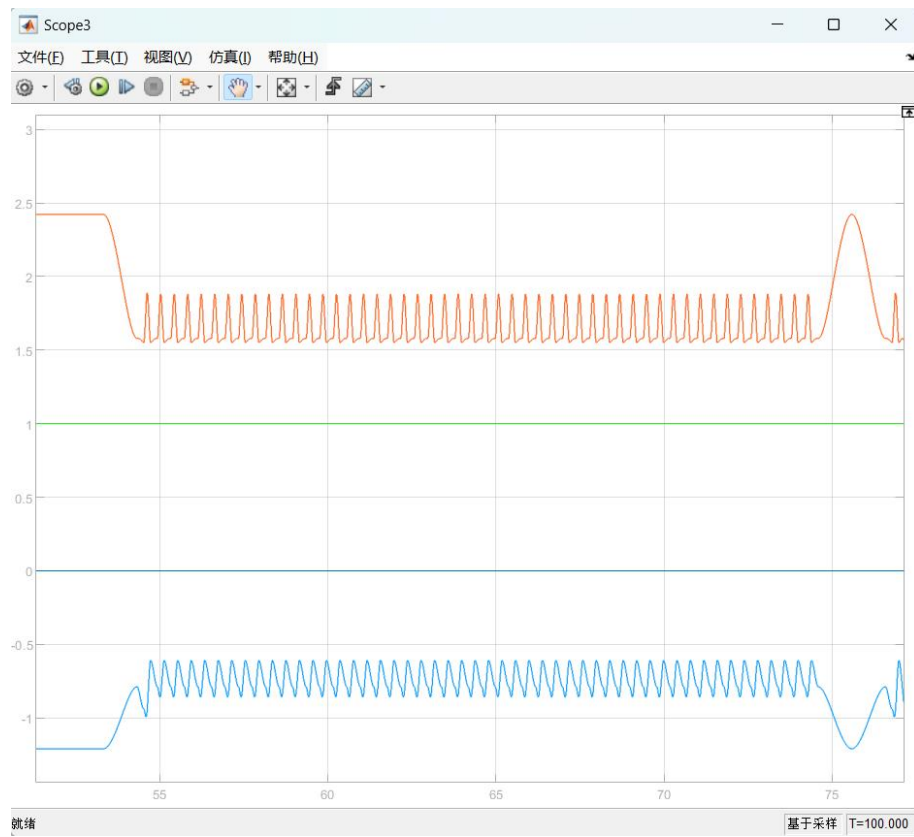


5.4. 足端、关节三次轨迹规划

足端、关节位移:

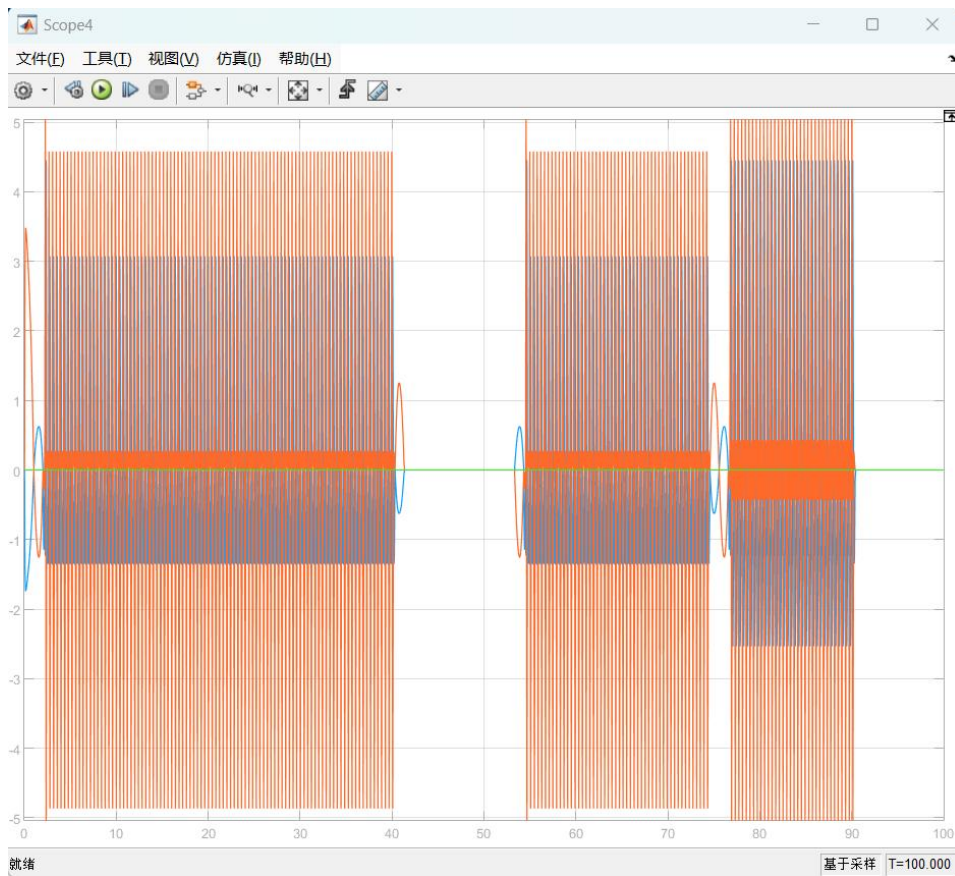


(整体图)

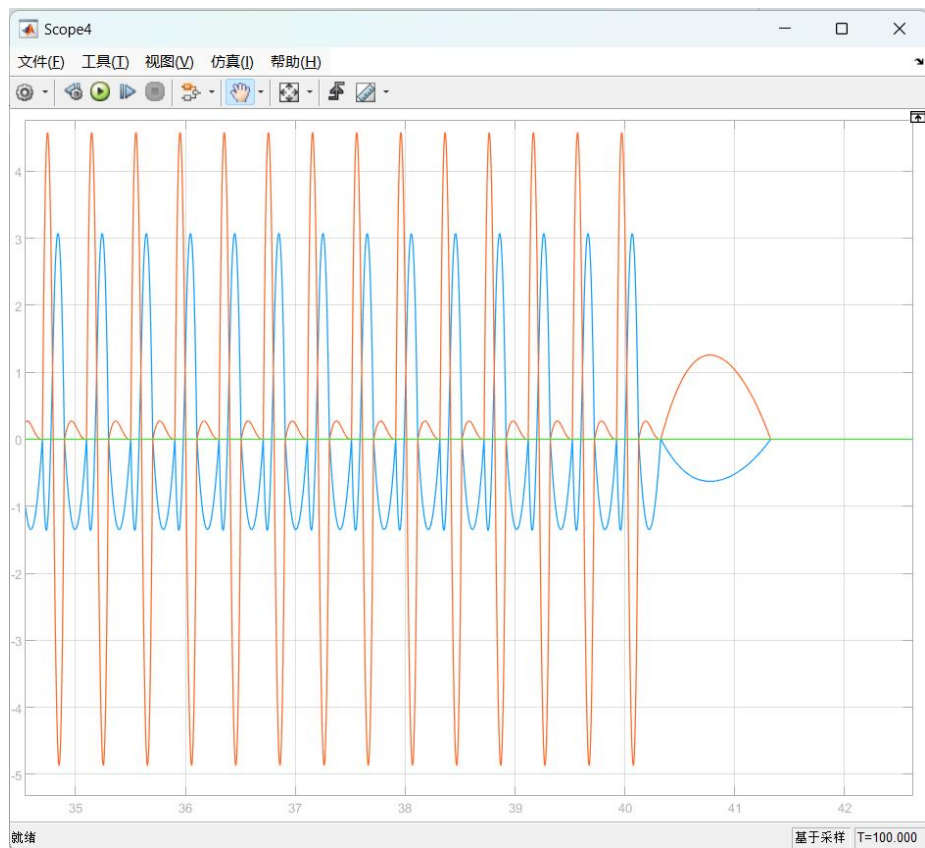


(细节图)

足端、关节速度:

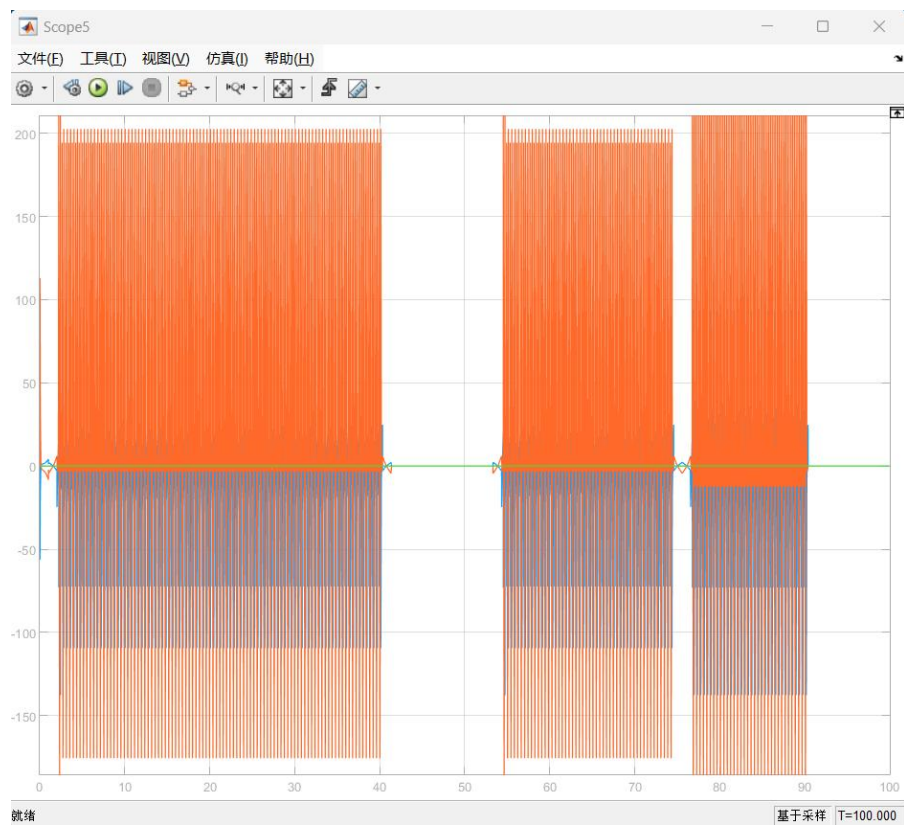


(整体图)

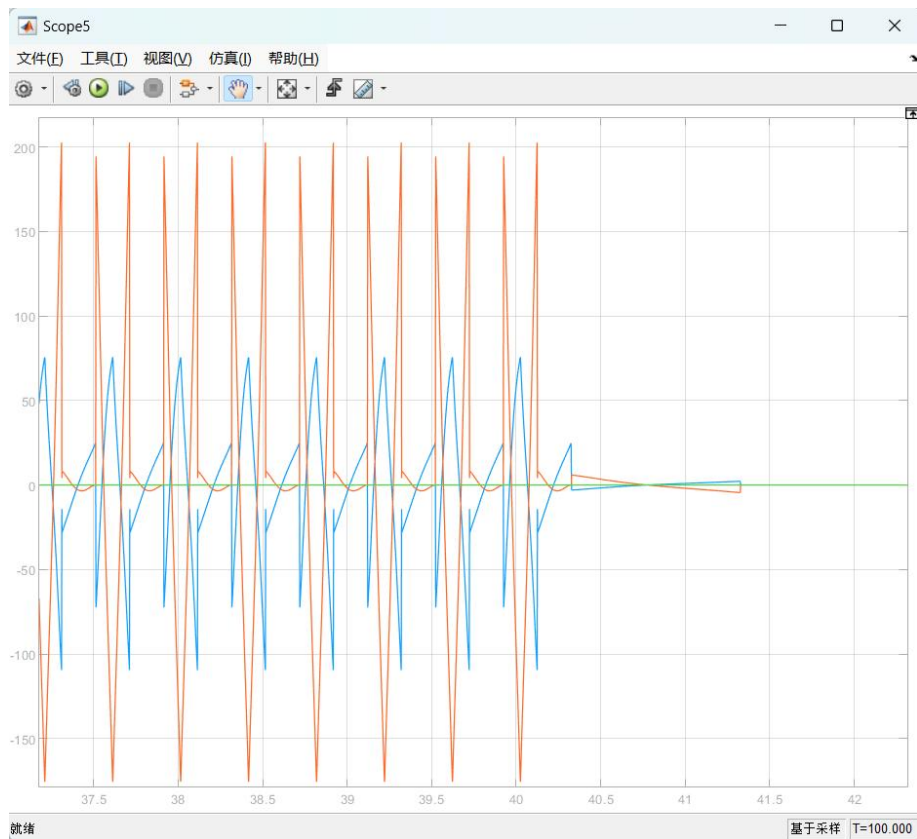


(细节图)

足端、关节加速度:

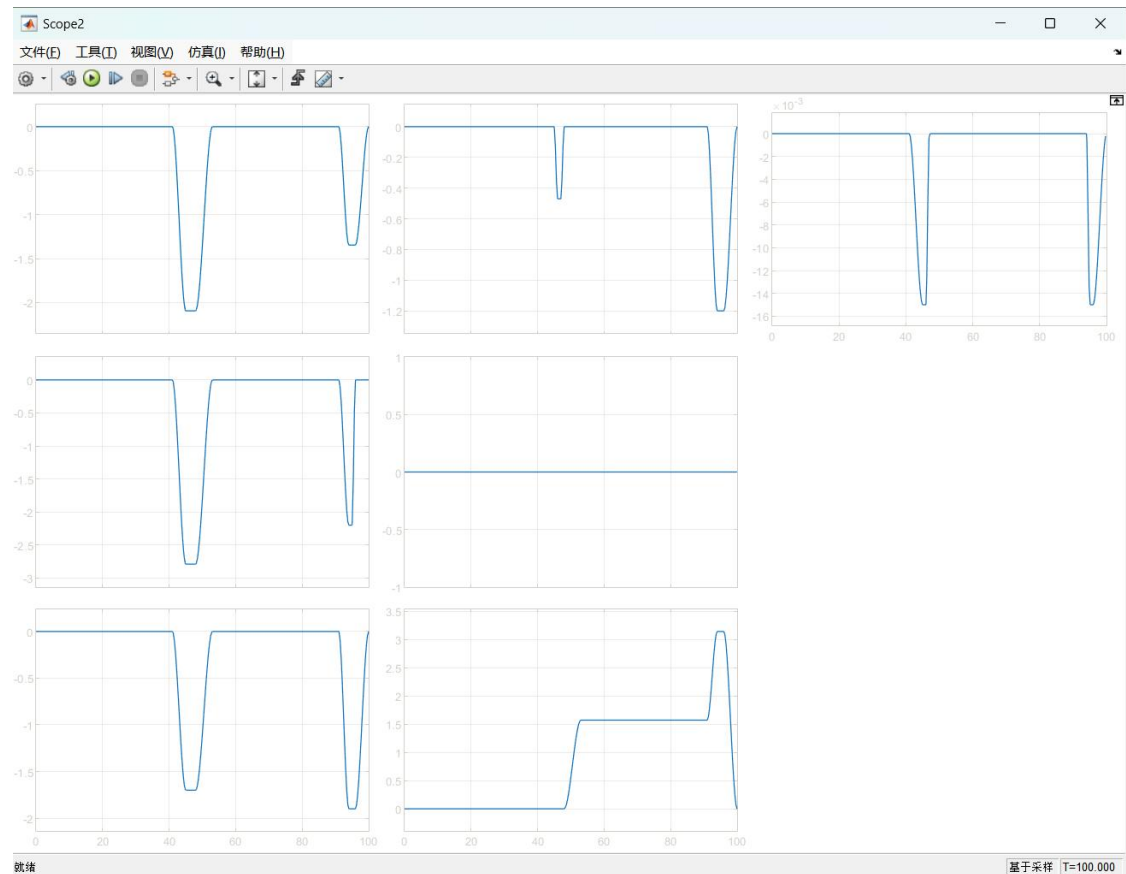


(整体图)



(细节图)

机械臂末端、关节位移：

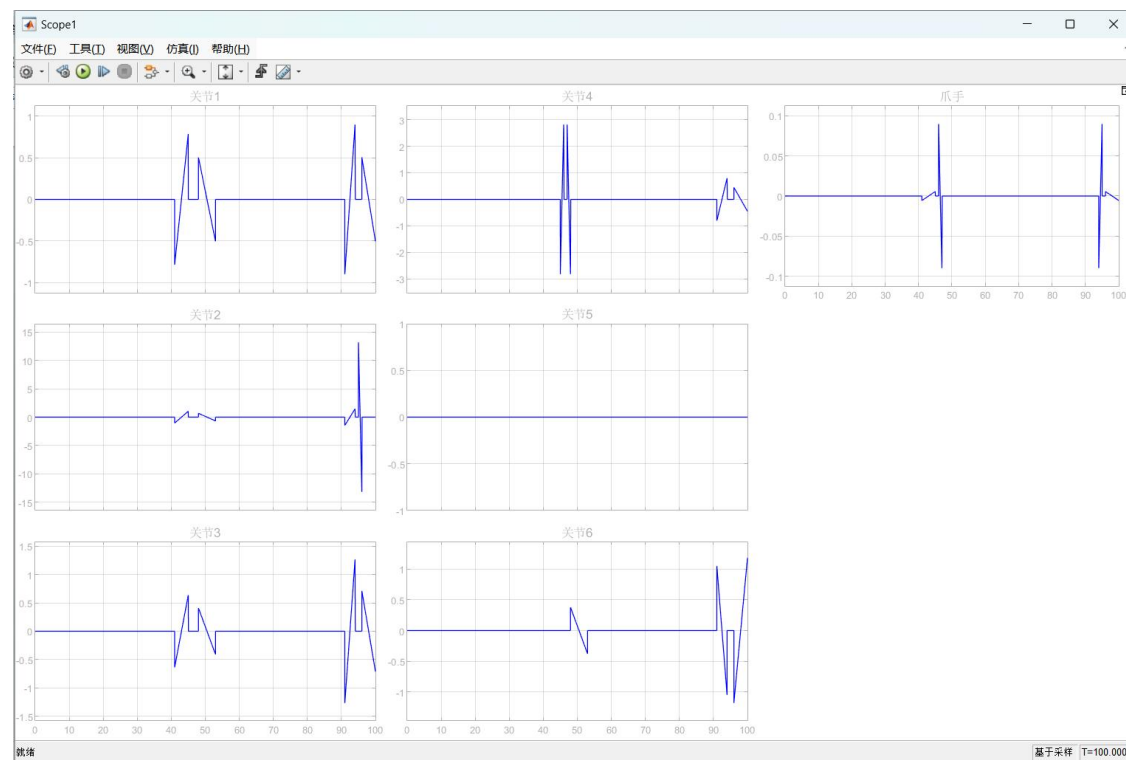


研究性专题-四足机器人臂足结合研究

机械臂末端、关节速度：



机械臂末端、关节加速度：



6. 系统综合调试

6.1. 系统软硬件综合调试

在调试过程中，我们小组遇到了很多问题，主要有以下几种：

1. 小球和机械爪之间的物理碰撞：这决定了机械抓手是否可以成功夹起小球，在调试过程中，我们通过调节小球与机械爪的碰撞系数，最终选择最合适的参数。
2. 四足机器人的拐弯步态：在虚拟仿真中由于模型拥有一定的自重，在设置差动步态时差距不能过大，会导致四足机器人翻转以及未按照指定轨迹行走。
3. 机械臂的抓取和放置：在虚拟仿真中，我们通过改变六自由度机械臂的各个关节角以及抓手的转态来控制机械臂的运动，在这个过程中，轨迹规划极为重要，经常出现穿模的现象，因此我们在轨迹中设置一个或者更多的中间点来避免这种情况。

6.2. 系统的经济性分析

对已经完成综合调试、实现预期功能的系统进行经济性分析。主要针对所研制的系统的经济成本以及市场发展前景两方面进行分析。

经济成本分析中，可以表格的形式列出所开发的系统的主要元器件价格和系统总价格。市场发展前景分析中，可从所开发的系统的优点、市场潜力、应用场合与发展前景等方面进行论述。

1. 经济成本分析：

主要元器件价格：

元器件	价格（人民币）
机械狗外壳	500
机械臂外壳	400
扭矩电机	4 000
传感器模块	150
通信模块	600

主控制板	100
电池及能源管理	550
总成本	6 300

2. 市场发展前景分析：

a.) 系统的优点：

系统能够实现四足机械狗和六轴机械臂的组合，具有移动和抓取功能，适用于特殊环境下的搜索和救援任务；机械狗和机械臂之间通过集中控制单元进行协调控制，能够高效配合完成任务；系统可以根据任务需求进行功能扩展和定制，具有一定的灵活性和定制性。

b.) 市场潜力：

救援行业对于能够在特殊环境下进行搜索和救援的机器人系统的需求逐渐增加，例如地震灾区、火灾现场等特殊环境下，需要能够迅速进入并执行任务的机器人系统。机器狗机械臂救援系统能够提高救援任务的安全性和效率，受到各种应急救援机构的关注。

c.) 应用场合与发展前景：

紧急救援：适用于地震、火灾等灾害现场的搜索和救援任务。

危险环境：例如放射性污染区域、化学品泄漏现场等，机器狗机械臂系统可以代替人员执行任务，保障人员安全。

军事用途：用于军事侦察、搜救等任务，提升军事行动效率。

7. 总结与展望

7.1. 总结

1. 首先，我明确了系统的应用场景为特殊环境下的搜索和救援任务，目标是结合四足机械狗和六轴机械臂，实现移动与抓取功能。在确定了系统的基本功能，包括机械狗的移动能力、机械臂的抓取能力、控制系统的稳定性等之后，我们为每个功能设定了具体的性能指标。

2. 在设计阶段，我们比较了集中控制式和分布控制式两种方案，并选择了

集中控制式作为总体设计方案，考虑到了系统的特殊环境下的搜索和救援任务需求。在这个阶段，我们也对系统的各个模块进行了详细设计，包括机械狗控制单元、机械臂控制单元、传感器模块、通信模块等，确定了各模块的功能、组成和相互之间的通信协议。然后我们进行了控制系统的软件开发，编写了相应的控制算法、通信协议以及用户界面等，确保了系统的稳定运行和用户友好性。

3. 最后实现了机械狗和机械臂的功能结合，可控制机器人从起点沿预定路线走到目标位置，控制机械臂夹起地上的小球，然后保持夹持状态回到起点。

7.2. 展望

在后续工作中，需进一步完善如下工作：

1. 动作控制优化：进一步优化机械狗和机械臂的动作控制算法，使其更加灵活、精准，以适应各种复杂的救援场景，例如狭窄通道、楼梯、障碍物等。
2. 传感器技术升级：引入更先进的传感器技术，如高精度惯性测量单元（IMU）、三维视觉系统等，提高环境感知和位置识别的精度，从而提升系统对复杂地形的适应能力。
3. 路径规划与避障策略：进一步完善路径规划算法，确保系统能够智能地避开障碍物、选择最优路径进行移动和抓取，提高在复杂地形下的行动效率。
4. 用户界面与遥控体验：设计更直观、易用的用户界面，提升操作人员的遥控体验，使其能够更加高效地控制机器狗和机械臂。

参考文献

- [1] Feng Lin and R. D. Brandt, "An optimal control approach to robust control of robot manipulators," in IEEE Transactions on Robotics and Automation, vol. 14, no. 1, pp. 69-77, Feb. 1998, doi: 10.1109/70.660845.
- [2] Ulloa, C.C., Domínguez, D., Barrientos, A., del Cerro, J. (2023). Design and Mixed-Reality Teleoperation of a Quadruped-Manipulator Robot for SAR Tasks. In: Cascalho, J.M., Tokhi, M.O., Silva, M.F., Mendes, A., Goher, K., Funk, M. (eds) Robotics in Natural Settings. CLAWAR 2022.

Lecture Notes in Networks and Systems, vol 530. Springer, Cham. https://doi.org/10.1007/978-3-031-15226-9_19

[3] Sun, Z., Zhu, Z., Zhang, G., Li, Y., Rong, X. (2021). Design and Control of a Quadruped Robot with Changeable Configuration. In: Liu, XJ., Nie, Z., Yu, J., Xie, F., Song, R. (eds) Intelligent Robotics and Applications. ICIRA 2021. Lecture Notes in Computer Science(), vol 13015. Springer, Cham. https://doi.org/10.1007/978-3-030-89134-3_58

[4] M. Sombolestan and Q. Nguyen, "Hierarchical Adaptive Loco-manipulation Control for Quadruped Robots," 2023 IEEE International Conference on Robotics and Automation (ICRA), London, United Kingdom, 2023, pp. 1215612162, doi:10.1109/ICRA48891.2023.10160523.

[5] Virbhadrappa Kalburgi, P.M. James, Pramod Sreedharan, Control system design for four-legged walking robot with insect type leg using ROS, Materials Today: Proceedings, Volume 46, Part 10, 2021, Pages 5092-097, ISSN 22147853, <https://doi.org/10.1016/j.matpr.2020.10.428>.

[6] Smith, John. "Integrated Control of Four-Legged Robot and Robotic Arm for Manipulation Tasks." Robotics and Automation Journal, vol. 10, no. 2, 2022, pp. 45-60. DOI:10.1016/j.robot.2023.104468.

[7] Brown, Sarah. "Enhancing Robot Mobility and Manipulation through Combination of Quadruped and Robotic Arm." Robotics Journal, vol. 8, no. 4, 2020, pp. 312-327. DOI: 10.1016/j.birob.2021.100029

[8] Qi, J., Yuan, Q., Wang, C. et al. Path planning and collision avoidance based on the RRT*FN framework for a robotic manipulator in various scenarios. Complex Intell. Syst. 9, 7475 – 7494 (2023). <https://doi.org/10.1007/s40747-023-01131-2>

[9] Priyaranjan Biswal, Prases K. Mohanty, Development of quadruped walking robots: A review, Ain Shams Engineering Journal, Volume 12, Issue 2, 2021, Pages 2017-2031, ISSN 2090-4479, <https://doi.org/10.1016/j.asej.2020.11.005>.

[10] Smith, John. "Development and Control of a Quadruped Robot with Articulated Robotic Arm for Outdoor Inspection." Robotics and Automation Journal, vol. 9, no. 1, 2021, pp. 55-70. DOI: 10.1016/j.birob.2021.100029.

[11] Kim, J., & Lee, S. (2017). Development of a legged robot with a multi-DOF robotic arm for maintenance tasks in nuclear power plants. Journal of Nuclear Materials Management, 45(2), 156-166.

[12] Kuo, C., & Lin, C. (2018). Development of a quadruped robot with a 6-DOF robotic arm for precision agriculture. Precision Agriculture, 19(6), 1009-1023.

- [13] Park, H., & Choi, S. (2019). Design and control of a quadruped robot with a 6-DOF manipulator for underwater exploration. *Journal of Field Robotics*, 36(2), 275-288.
- [14] Li, Y., & Wang, Z. (2020). Development of a quadruped robot with a 6-DOF robotic arm for construction tasks. *Automation in Construction*, 111, 103069.
- [15] Wang, X., & Chen, J. (2017). A survey of quadruped robot research. *International Journal of Advanced Robotic Systems*, 14(5), 1729881417716388.

附件 1 程序

代码功能：四足机器人运动

代码：

```
function [tgt_ctrl_fl,tgt_ctrl_fr,tgt_ctrl_rl, tgt_ctrl_rr, ctrl_state,
debug] = ...
controller1(~, ~, ~, ~, contact_state, ~, sensing_info, t, body, planner)
```

```
body_vx = sensing_info(4);
```

```
%virtual leg1 fl (front left) rr(rear right) body frame
% curr_virtual_leg1 = (fl_end_effector + fl_offset + rr_end_effector +
rr_offset)*0.5;
%virtual leg2 fr (front right) rl(rear left) body frame
% curr_virtual_leg2 = (fr_end_effector + fr_offset + rl_end_effector +
rl_offset)*0.5;
```

```
%decide state and state transition helpers
```

```
persistent state;
persistent t_last_state;
if isempty(state)
state = 0;
end
if isempty(t_last_state)
t_last_state = 0;
end
```

```
% init variables
```

```
% fl fr rl rr
```

```
has_u_ctrl = [1 1 1 1];
normal_virtual_legFL = [0;body.shoulder_size/2;-body.upper_length-
body.lower_length+0.001];
normal_virtual_legRL = [0;body.shoulder_size/2;-body.upper_length-
body.lower_length+0.001];
```

```
normal_virtual_legFR = [0;-body.shoulder_size/2;-body.upper_length-
body.lower_length+0.001];
```

```
normal_virtual_legRR = [0;-body.shoulder_size/2;-body.upper_length-
body.lower_length+0.001];
```

```
%RR 4 FL 1
```



```

move_fl_end_effector = normal_virtual_legFL;
move_rr_end_effector = normal_virtual_legRR;
%FR 2 RL 3
move_fr_end_effector = normal_virtual_legFR;
move_rl_end_effector = normal_virtual_legRL;
% state machine
if state == 0 % standby state
% in this state, just do nothing, but use time to make sure that all
% feet stands on the ground
if contact_state(1) == 1 && contact_state(2) == 1 && contact_state(3) == 1
&& contact_state(4) == 1 && abs(body_vx) < planner.state0_vel_thres
%all foot on ground
t_last_state=t;
state = 1;
end
elseif state == 1 % 直立到蹲下
tick=t-t_last_state;
t0=0;
tf=1;
%leg rl fl
q_t0=[0 body.shoulder_size/2 -body.upper_length-body.lower_length+0.001];
q_tf=[0 body.shoulder_size/2 planner.stand_height/2];
d_q_t0=zeros(1,3);
d_q_tf=zeros(1,3);
move_rl_end_effector=traj3(t0,tf,q_t0,d_q_t0,q_tf,d_q_tf,tick);
move_fl_end_effector=move_rl_end_effector;
%leg rr fr
q_t0=[0 -body.shoulder_size/2 -body.upper_length-body.lower_length+0.001];
q_tf=[0 -body.shoulder_size/2 planner.stand_height/2];
move_rr_end_effector=traj3(t0,tf,q_t0,d_q_t0,q_tf,d_q_tf,tick);
move_fr_end_effector=move_rr_end_effector;
if tick>tf
move_rl_end_effector=[0 body.shoulder_size/2 planner.stand_height/2];
move_fl_end_effector=move_rl_end_effector;
move_rr_end_effector=[0 -body.shoulder_size/2 planner.stand_height/2];
move_fr_end_effector=move_rr_end_effector;
if contact_state(1) == 1 && contact_state(2) == 1 && contact_state(3) == 1
&& contact_state(4) == 1
t_last_state=t;
state = 2;
end
end

elseif state == 2% 趴下起来

```

```

tick=t-t_last_state;
t0=0;
tf=1;
%leg r1 f1
q_t0=[0 body.shoulder_size/2 planner.stand_height/2];
q_tf=[0 body.shoulder_size/2 planner.stand_height];
d_q_t0=zeros(1,3);
d_q_tf=zeros(1,3);
move_r1_end_effector=traj3(t0,tf,q_t0,d_q_t0,q_tf,d_q_tf,tick);
move_f1_end_effector=move_r1_end_effector;
%leg rr fr
q_t0=[0 -body.shoulder_size/2 planner.stand_height/2];
q_tf=[0 -body.shoulder_size/2 planner.stand_height];
move_rr_end_effector=traj3(t0,tf,q_t0,d_q_t0,q_tf,d_q_tf,tick);
move_fr_end_effector=move_rr_end_effector;
if tick>tf
move_r1_end_effector=[0 body.shoulder_size/2 planner.stand_height];
move_f1_end_effector=move_r1_end_effector;
move_rr_end_effector=[0 -body.shoulder_size/2 planner.stand_height];
move_fr_end_effector=move_rr_end_effector;
if contact_state(1) == 1 && contact_state(2) == 1 && contact_state(3) == 1
&& contact_state(4) == 1
t_last_state=t;
state = 3;
end
end
elseif state == 3 % r1 fr swing S/2 rr f1 stance -S/2 过渡阶段
tick=t-t_last_state;
t0=0;
tf=planner.Ts;
%leg r1
q_t0=[0 body.shoulder_size/2 planner.stand_height];
q_tf=[planner.S/2 body.shoulder_size/2 planner.stand_height];
d_q_t0=zeros(1,3);
d_q_tf=zeros(1,3);
move_r1_end_effector=traj3(t0,tf,q_t0,d_q_t0,q_tf,d_q_tf,tick); %xy 轨迹
if tick<planner.Ts/2 %z 轨迹
q_t0=[0 body.shoulder_size/2 planner.stand_height];
q_tf=[0 body.shoulder_size/2 planner.stand_height+planner.H];
move_r1_end_effector_z=traj3(t0,tf/2,q_t0,d_q_t0,q_tf,d_q_tf,tick);
else
q_t0=[0 body.shoulder_size/2 planner.stand_height+planner.H];
q_tf=[0 body.shoulder_size/2 planner.stand_height];

```

```

move_rl_end_effector_z=traj3(t0,tf/2,q_t0,d_q_t0,q_tf,d_q_tf,tick-
planner.Ts/2);
end
move_rl_end_effector(3)=move_rl_end_effector_z(3);
%leg fr
q_t0=[0 -body.shoulder_size/2 planner.stand_height];
q_tf=[planner.S/2 -body.shoulder_size/2 planner.stand_height];
d_q_t0=zeros(1,3);
d_q_tf=zeros(1,3);
move_fr_end_effector=traj3(t0,tf,q_t0,d_q_t0,q_tf,d_q_tf,tick); %xy 轨迹
if tick<planner.Ts/2 %z 轨迹
q_t0=[0 -body.shoulder_size/2 planner.stand_height];
q_tf=[0 -body.shoulder_size/2 planner.stand_height+planner.H];
move_fr_end_effector_z=traj3(t0,tf/2,q_t0,d_q_t0,q_tf,d_q_tf,tick);
else
q_t0=[0 -body.shoulder_size/2 planner.stand_height+planner.H];
q_tf=[0 -body.shoulder_size/2 planner.stand_height];
move_fr_end_effector_z=traj3(t0,tf/2,q_t0,d_q_t0,q_tf,d_q_tf,tick-
planner.Ts/2);
end
move_fr_end_effector(3)=move_fr_end_effector_z(3);
%leg fl
q_t0=[0 body.shoulder_size/2 planner.stand_height];
q_tf=[-planner.S/2 body.shoulder_size/2 planner.stand_height];
d_q_t0=zeros(1,3);
d_q_tf=zeros(1,3);
move_fl_end_effector=traj3(t0,tf,q_t0,d_q_t0,q_tf,d_q_tf,tick); %xy 轨迹
%leg rr
q_t0=[0 -body.shoulder_size/2 planner.stand_height];
q_tf=[-planner.S/2 -body.shoulder_size/2 planner.stand_height];
d_q_t0=zeros(1,3);
d_q_tf=zeros(1,3);
move_rr_end_effector=traj3(t0,tf,q_t0,d_q_t0,q_tf,d_q_tf,tick); %xy 轨迹
if tick>tf
move_rl_end_effector=[planner.S/2 body.shoulder_size/2
planner.stand_height];
move_fl_end_effector=[-planner.S/2 body.shoulder_size/2
planner.stand_height];
move_rr_end_effector=[-planner.S/2 -body.shoulder_size/2
planner.stand_height];
move_fr_end_effector=[planner.S/2 -body.shoulder_size/2
planner.stand_height];
%if contact_state(2) == 1 && contact_state(3) == 1
t_last_state=t;

```

```

state = 4;
%end
end
elseif state == 4 % r1 fr stance -S rr fl swing S 过渡阶段（直行）
tick=t-t_last_state;
t0=0;
tf=planner.Ts;
%leg rr
q_t0=[-planner.S/2 -body.shoulder_size/2 planner.stand_height];
q_tf=[planner.S/2 -body.shoulder_size/2 planner.stand_height];
d_q_t0=zeros(1,3);
d_q_tf=zeros(1,3);
move_rr_end_effector=traj3(t0,tf,q_t0,d_q_t0,q_tf,d_q_tf,tick); %xy 轨迹
if tick<planner.Ts/2 %z 轨迹
q_t0=[0 -body.shoulder_size/2 planner.stand_height];
q_tf=[0 -body.shoulder_size/2 planner.stand_height+planner.H];
move_rr_end_effector_z=traj3(t0,tf/2,q_t0,d_q_t0,q_tf,d_q_tf,tick);
else
q_t0=[0 -body.shoulder_size/2 planner.stand_height+planner.H];
q_tf=[0 -body.shoulder_size/2 planner.stand_height];
move_rr_end_effector_z=traj3(t0,tf/2,q_t0,d_q_t0,q_tf,d_q_tf,tick-
planner.Ts/2);
end
move_rr_end_effector(3)=move_rr_end_effector_z(3);
%leg fl
q_t0=[-planner.S/2 body.shoulder_size/2 planner.stand_height];
q_tf=[planner.S/2 body.shoulder_size/2 planner.stand_height];
d_q_t0=zeros(1,3);
d_q_tf=zeros(1,3);
move_fl_end_effector=traj3(t0,tf,q_t0,d_q_t0,q_tf,d_q_tf,tick); %xy 轨迹
if tick<planner.Ts/2 %z 轨迹
q_t0=[0 body.shoulder_size/2 planner.stand_height];
q_tf=[0 body.shoulder_size/2 planner.stand_height+planner.H];
move_fl_end_effector_z=traj3(t0,tf/2,q_t0,d_q_t0,q_tf,d_q_tf,tick);
else
q_t0=[0 body.shoulder_size/2 planner.stand_height+planner.H];
q_tf=[0 body.shoulder_size/2 planner.stand_height];
move_fl_end_effector_z=traj3(t0,tf/2,q_t0,d_q_t0,q_tf,d_q_tf,tick-
planner.Ts/2);
end
move_fl_end_effector(3)=move_fl_end_effector_z(3);
%leg r1
q_t0=[planner.S/2 body.shoulder_size/2 planner.stand_height];
q_tf=[-planner.S/2 body.shoulder_size/2 planner.stand_height];

```

```

d_q_t0=zeros(1,3);
d_q_tf=zeros(1,3);
move_rl_end_effector=traj3(t0,tf,q_t0,d_q_t0,q_tf,d_q_tf,tick); %xy 轨迹
%leg fr
q_t0=[planner.S/2 -body.shoulder_size/2 planner.stand_height];
q_tf=[-planner.S/2 -body.shoulder_size/2 planner.stand_height];
d_q_t0=zeros(1,3);
d_q_tf=zeros(1,3);
move_fr_end_effector=traj3(t0,tf,q_t0,d_q_t0,q_tf,d_q_tf,tick); %xy 轨迹
if tick>tf
move_rl_end_effector=[-planner.S/2 body.shoulder_size/2
planner.stand_height];
move_fl_end_effector=[planner.S/2 body.shoulder_size/2
planner.stand_height];
move_rr_end_effector=[planner.S/2 -body.shoulder_size/2
planner.stand_height];
move_fr_end_effector=[-planner.S/2 -body.shoulder_size/2
planner.stand_height];
%if contact_state(1) == 1 && contact_state(4) == 1
if t<74
t_last_state=t;
state = 5;
elseif t<90
t_last_state=t;
state = 7;
else
t_last_state=t;
state = 11;
%end
end
%end
end
elseif state == 5 % 过渡阶段（走曲线）
tick=t-t_last_state;
t0=0;
tf=planner.Ts;
%leg r1
q_t0=[-planner.S/2 body.shoulder_size/2 planner.stand_height];
q_tf=[planner.S/2 body.shoulder_size/2 planner.stand_height];
d_q_t0=zeros(1,3);
d_q_tf=zeros(1,3);
move_rl_end_effector=traj3(t0,tf,q_t0,d_q_t0,q_tf,d_q_tf,tick); %xy 轨迹
if tick<planner.Ts/2 %z 轨迹
q_t0=[0 body.shoulder_size/2 planner.stand_height];

```

```

q_tf=[0 body.shoulder_size/2 planner.stand_height+planner.H];
move_rl_end_effector_z=traj3(t0,tf/2,q_t0,d_q_t0,q_tf,d_q_tf,tick);
else
q_t0=[0 body.shoulder_size/2 planner.stand_height+planner.H];
q_tf=[0 body.shoulder_size/2 planner.stand_height];
move_rl_end_effector_z=traj3(t0,tf/2,q_t0,d_q_t0,q_tf,d_q_tf,tick-
planner.Ts/2);
end
move_rl_end_effector(3)=move_rl_end_effector_z(3);
%leg fr
q_t0=[-planner.S/2 -body.shoulder_size/2 planner.stand_height];
q_tf=[0 -body.shoulder_size/2 planner.stand_height];
d_q_t0=zeros(1,3);
d_q_tf=zeros(1,3);
move_fr_end_effector=traj3(t0,tf,q_t0,d_q_t0,q_tf,d_q_tf,tick); %xy 轨迹
if tick<planner.Ts/2 %z 轨迹
q_t0=[0 -body.shoulder_size/2 planner.stand_height];
q_tf=[0 -body.shoulder_size/2 planner.stand_height+planner.H];
move_fr_end_effector_z=traj3(t0,tf/2,q_t0,d_q_t0,q_tf,d_q_tf,tick);
else
q_t0=[0 -body.shoulder_size/2 planner.stand_height+planner.H];
q_tf=[0 -body.shoulder_size/2 planner.stand_height];
move_fr_end_effector_z=traj3(t0,tf/2,q_t0,d_q_t0,q_tf,d_q_tf,tick-
planner.Ts/2);
end
move_fr_end_effector(3)=move_fr_end_effector_z(3);
%leg rr
q_t0=[planner.S/2 -body.shoulder_size/2 planner.stand_height];
q_tf=[-planner.S/2 -body.shoulder_size/2 planner.stand_height];
d_q_t0=zeros(1,3);
d_q_tf=zeros(1,3);
move_rr_end_effector=traj3(t0,tf,q_t0,d_q_t0,q_tf,d_q_tf,tick); %xy 轨迹
%leg fl
q_t0=[planner.S/2 body.shoulder_size/2 planner.stand_height];
q_tf=[0 body.shoulder_size/2 planner.stand_height];
d_q_t0=zeros(1,3);
d_q_tf=zeros(1,3);
move_fl_end_effector=traj3(t0,tf,q_t0,d_q_t0,q_tf,d_q_tf,tick); %xy 轨迹
if tick>tf
move_rl_end_effector=[planner.S/2 body.shoulder_size/2
planner.stand_height];
move_fl_end_effector=[0 body.shoulder_size/2 planner.stand_height];
move_rr_end_effector=[-planner.S/2 -body.shoulder_size/2
planner.stand_height];

```

```

move_fr_end_effector=[0 -body.shoulder_size/2 planner.stand_height];
%if contact_state(2) == 1 && contact_state(3) == 1
t_last_state=t;
state = 6;
%end
end
elseif state == 6 % 过渡阶段（走曲线）
tick=t-t_last_state;
t0=0;
tf=planner.Ts;
%leg rr
q_t0=[-planner.S/2 -body.shoulder_size/2 planner.stand_height];
q_tf=[planner.S/2 -body.shoulder_size/2 planner.stand_height];
d_q_t0=zeros(1,3);
d_q_tf=zeros(1,3);
move_rr_end_effector=traj3(t0,tf,q_t0,d_q_t0,q_tf,d_q_tf,tick); %xy 轨迹
if tick<planner.Ts/2 %z 轨迹
q_t0=[0 -body.shoulder_size/2 planner.stand_height];
q_tf=[0 -body.shoulder_size/2 planner.stand_height+planner.H];
move_rr_end_effector_z=traj3(t0,tf/2,q_t0,d_q_t0,q_tf,d_q_tf,tick);
else
q_t0=[0 -body.shoulder_size/2 planner.stand_height+planner.H];
q_tf=[0 -body.shoulder_size/2 planner.stand_height];
move_rr_end_effector_z=traj3(t0,tf/2,q_t0,d_q_t0,q_tf,d_q_tf,tick-
planner.Ts/2);
end
move_rr_end_effector(3)=move_rr_end_effector_z(3);
%leg fl
q_t0=[0 body.shoulder_size/2 planner.stand_height];
q_tf=[planner.S/2 body.shoulder_size/2 planner.stand_height];
d_q_t0=zeros(1,3);
d_q_tf=zeros(1,3);
move_fl_end_effector=traj3(t0,tf,q_t0,d_q_t0,q_tf,d_q_tf,tick); %xy 轨迹
if tick<planner.Ts/2 %z 轨迹
q_t0=[0 body.shoulder_size/2 planner.stand_height];
q_tf=[0 body.shoulder_size/2 planner.stand_height+planner.H];
move_fl_end_effector_z=traj3(t0,tf/2,q_t0,d_q_t0,q_tf,d_q_tf,tick);
else
q_t0=[0 body.shoulder_size/2 planner.stand_height+planner.H];
q_tf=[0 body.shoulder_size/2 planner.stand_height];
move_fl_end_effector_z=traj3(t0,tf/2,q_t0,d_q_t0,q_tf,d_q_tf,tick-
planner.Ts/2);
end
move_fl_end_effector(3)=move_fl_end_effector_z(3);

```

```

%leg r1
q_t0=[planner.S/2 body.shoulder_size/2 planner.stand_height];
q_tf=[-planner.S/2 body.shoulder_size/2 planner.stand_height];
d_q_t0=zeros(1,3);
d_q_tf=zeros(1,3);
move_r1_end_effector=traj3(t0,tf,q_t0,d_q_t0,q_tf,d_q_tf,tick); %xy 轨迹
%leg fr
q_t0=[0 -body.shoulder_size/2 planner.stand_height];
q_tf=[-planner.S/2 -body.shoulder_size/2 planner.stand_height];
d_q_t0=zeros(1,3);
d_q_tf=zeros(1,3);
move_fr_end_effector=traj3(t0,tf,q_t0,d_q_t0,q_tf,d_q_tf,tick); %xy 轨迹
if tick>tf
move_r1_end_effector=[-planner.S/2 body.shoulder_size/2
planner.stand_height];
move_fl_end_effector=[planner.S/2 body.shoulder_size/2
planner.stand_height];
move_rr_end_effector=[planner.S/2 -body.shoulder_size/2
planner.stand_height];
move_fr_end_effector=[-planner.S/2 -body.shoulder_size/2
planner.stand_height];
%if contact_state(1) == 1 && contact_state(4) == 1
if t<40
t_last_state=t;
state = 5;
elseif (t>54)&&(t<74)
t_last_state=t;
state = 5;
else
t_last_state=t;
state = 8;
%end
end
end

elseif state == 7 % r1 fr swing S rr fl stance -S 过渡阶段（直行）
tick=t-t_last_state;
t0=0;
tf=planner.Ts;
%leg r1
q_t0=[-planner.S/2 body.shoulder_size/2 planner.stand_height];
q_tf=[planner.S/2 body.shoulder_size/2 planner.stand_height];
d_q_t0=zeros(1,3);
d_q_tf=zeros(1,3);

```



```

move_rl_end_effector=traj3(t0,tf,q_t0,d_q_t0,q_tf,d_q_tf,tick); %xy 轨迹
if tick<planner.Ts/2 %z 轨迹
q_t0=[0 body.shoulder_size/2 planner.stand_height];
q_tf=[0 body.shoulder_size/2 planner.stand_height+planner.H];
move_rl_end_effector_z=traj3(t0,tf/2,q_t0,d_q_t0,q_tf,d_q_tf,tick);
else
q_t0=[0 body.shoulder_size/2 planner.stand_height+planner.H];
q_tf=[0 body.shoulder_size/2 planner.stand_height];
move_rl_end_effector_z=traj3(t0,tf/2,q_t0,d_q_t0,q_tf,d_q_tf,tick-
planner.Ts/2);
end
move_rl_end_effector(3)=move_rl_end_effector_z(3);
%leg fr
q_t0=[-planner.S/2 -body.shoulder_size/2 planner.stand_height];
q_tf=[planner.S/2 -body.shoulder_size/2 planner.stand_height];
d_q_t0=zeros(1,3);
d_q_tf=zeros(1,3);
move_fr_end_effector=traj3(t0,tf,q_t0,d_q_t0,q_tf,d_q_tf,tick); %xy 轨迹
if tick<planner.Ts/2 %z 轨迹
q_t0=[0 -body.shoulder_size/2 planner.stand_height];
q_tf=[0 -body.shoulder_size/2 planner.stand_height+planner.H];
move_fr_end_effector_z=traj3(t0,tf/2,q_t0,d_q_t0,q_tf,d_q_tf,tick);
else
q_t0=[0 -body.shoulder_size/2 planner.stand_height+planner.H];
q_tf=[0 -body.shoulder_size/2 planner.stand_height];
move_fr_end_effector_z=traj3(t0,tf/2,q_t0,d_q_t0,q_tf,d_q_tf,tick-
planner.Ts/2);
end
move_fr_end_effector(3)=move_fr_end_effector_z(3);
%leg rr
q_t0=[planner.S/2 -body.shoulder_size/2 planner.stand_height];
q_tf=[-planner.S/2 -body.shoulder_size/2 planner.stand_height];
d_q_t0=zeros(1,3);
d_q_tf=zeros(1,3);
move_rr_end_effector=traj3(t0,tf,q_t0,d_q_t0,q_tf,d_q_tf,tick); %xy 轨迹
%leg fl
q_t0=[planner.S/2 body.shoulder_size/2 planner.stand_height];
q_tf=[-planner.S/2 body.shoulder_size/2 planner.stand_height];
d_q_t0=zeros(1,3);
d_q_tf=zeros(1,3);
move_fl_end_effector=traj3(t0,tf,q_t0,d_q_t0,q_tf,d_q_tf,tick); %xy 轨迹
if tick>tf
move_rl_end_effector=[planner.S/2 body.shoulder_size/2
planner.stand_height];

```

```

move_fl_end_effector=[-planner.S/2 body.shoulder_size/2
planner.stand_height];
move_rr_end_effector=[-planner.S/2 -body.shoulder_size/2
planner.stand_height];
move_fr_end_effector=[planner.S/2 -body.shoulder_size/2
planner.stand_height];
%if contact_state(2) == 1 && contact_state(3) == 1
t_last_state=t;
state = 4;
%end
end
elseif state == 8 % 直立 1
tick=t-t_last_state;
t0=0;
tf=planner.Ts;
%leg rl
q_t0=[-planner.S/2 body.shoulder_size/2 planner.stand_height];
q_tf=[0 body.shoulder_size/2 planner.stand_height];
d_q_t0=zeros(1,3);
d_q_tf=zeros(1,3);
move_rl_end_effector=traj3(t0,tf,q_t0,d_q_t0,q_tf,d_q_tf,tick); %xy 轨迹
if tick<planner.Ts/2 %z 轨迹
q_t0=[0 body.shoulder_size/2 planner.stand_height];
q_tf=[0 body.shoulder_size/2 planner.stand_height+planner.H];
move_rl_end_effector_z=traj3(t0,tf/2,q_t0,d_q_t0,q_tf,d_q_tf,tick);
else
q_t0=[0 body.shoulder_size/2 planner.stand_height+planner.H];
q_tf=[0 body.shoulder_size/2 planner.stand_height];
move_rl_end_effector_z=traj3(t0,tf/2,q_t0,d_q_t0,q_tf,d_q_tf,tick-
planner.Ts/2);
end
move_rl_end_effector(3)=move_rl_end_effector_z(3);
%leg fr
q_t0=[-planner.S/2 -body.shoulder_size/2 planner.stand_height];
q_tf=[0 -body.shoulder_size/2 planner.stand_height];
d_q_t0=zeros(1,3);
d_q_tf=zeros(1,3);
move_fr_end_effector=traj3(t0,tf,q_t0,d_q_t0,q_tf,d_q_tf,tick); %xy 轨迹
if tick<planner.Ts/2 %z 轨迹
q_t0=[0 -body.shoulder_size/2 planner.stand_height];
q_tf=[0 -body.shoulder_size/2 planner.stand_height+planner.H];
move_fr_end_effector_z=traj3(t0,tf/2,q_t0,d_q_t0,q_tf,d_q_tf,tick);
else
q_t0=[0 -body.shoulder_size/2 planner.stand_height+planner.H];

```

```

q_tf=[0 -body.shoulder_size/2 planner.stand_height];
move_fr_end_effector_z=traj3(t0,tf/2,q_t0,d_q_t0,q_tf,d_q_tf,tick-
planner.Ts/2);
end
move_fr_end_effector(3)=move_fr_end_effector_z(3);
%leg rr
q_t0=[planner.S/2 -body.shoulder_size/2 planner.stand_height];
q_tf=[0 -body.shoulder_size/2 planner.stand_height];
d_q_t0=zeros(1,3);
d_q_tf=zeros(1,3);
move_rr_end_effector=traj3(t0,tf,q_t0,d_q_t0,q_tf,d_q_tf,tick); %xy 轨迹
%leg fl
q_t0=[planner.S/2 body.shoulder_size/2 planner.stand_height];
q_tf=[0 body.shoulder_size/2 planner.stand_height];
d_q_t0=zeros(1,3);
d_q_tf=zeros(1,3);
move_fl_end_effector=traj3(t0,tf,q_t0,d_q_t0,q_tf,d_q_tf,tick); %xy 轨迹
if tick>tf
move_rl_end_effector=[0 body.shoulder_size/2 planner.stand_height];
move_fl_end_effector=[0 body.shoulder_size/2 planner.stand_height];
move_rr_end_effector=[0 -body.shoulder_size/2 planner.stand_height];
move_fr_end_effector=[0 -body.shoulder_size/2 planner.stand_height];
%if contact_state(2) == 1 && contact_state(3) == 1
t_last_state=t;
state = 9;
%end

end
elseif state == 9 % 直立 1 到蹲下
tick=t-t_last_state;
t0=0;
tf=1;
%leg rl fl
q_t0=[0 body.shoulder_size/2 planner.stand_height];
q_tf=[0 body.shoulder_size/2 planner.stand_height/2];
d_q_t0=zeros(1,3);
d_q_tf=zeros(1,3);
move_rl_end_effector=traj3(t0,tf,q_t0,d_q_t0,q_tf,d_q_tf,tick);
move_fl_end_effector=move_rl_end_effector;
%leg rr fr
q_t0=[0 -body.shoulder_size/2 planner.stand_height];
q_tf=[0 -body.shoulder_size/2 planner.stand_height/2];
move_rr_end_effector=traj3(t0,tf,q_t0,d_q_t0,q_tf,d_q_tf,tick);
move_fr_end_effector=move_rr_end_effector;

```

```

if tick>tf
move_rl_end_effector=[0 body.shoulder_size/2 planner.stand_height/2];
move_fl_end_effector=move_rl_end_effector;
move_rr_end_effector=[0 -body.shoulder_size/2 planner.stand_height/2];
move_fr_end_effector=move_rr_end_effector;
if contact_state(1) == 1 && contact_state(2) == 1 && contact_state(3) == 1
&& contact_state(4) == 1
if t<74
t_last_state=t;
state = 10;
else
t_last_state=t;
state = 2;
%end
end
end
elseif state == 10% 维持趴下状态
tick=t-t_last_state;
t0=0;
tf=12;
%leg r1 fl
move_rl_end_effector=[0 body.shoulder_size/2 planner.stand_height/2];
move_fl_end_effector=move_rl_end_effector;
move_rr_end_effector=[0 -body.shoulder_size/2 planner.stand_height/2];
move_fr_end_effector=move_rr_end_effector;
if tick>tf
if contact_state(1) == 1 && contact_state(2) == 1 && contact_state(3) == 1
&& contact_state(4) == 1
t_last_state=t;
state = 2;
end
end
elseif state == 11 % 机器人直立 2
tick=t-t_last_state;
t0=0;
tf=planner.Ts;
%leg r1
q_t0=[-planner.S/2 body.shoulder_size/2 planner.stand_height];
q_tf=[0 body.shoulder_size/2 planner.stand_height];
d_q_t0=zeros(1,3);
d_q_tf=zeros(1,3);
move_rl_end_effector=traj3(t0,tf,q_t0,d_q_t0,q_tf,d_q_tf,tick); %xy 轨迹
if tick<planner.Ts/2 %z 轨迹

```

```

q_t0=[0 body.shoulder_size/2 planner.stand_height];
q_tf=[0 body.shoulder_size/2 planner.stand_height+planner.H];
move_rl_end_effector_z=traj3(t0,tf/2,q_t0,d_q_t0,q_tf,d_q_tf,tick);
else
q_t0=[0 body.shoulder_size/2 planner.stand_height+planner.H];
q_tf=[0 body.shoulder_size/2 planner.stand_height];
move_rl_end_effector_z=traj3(t0,tf/2,q_t0,d_q_t0,q_tf,d_q_tf,tick-
planner.Ts/2);
end
move_rl_end_effector(3)=move_rl_end_effector_z(3);
%leg fr
q_t0=[-planner.S/2 -body.shoulder_size/2 planner.stand_height];
q_tf=[0 -body.shoulder_size/2 planner.stand_height];
d_q_t0=zeros(1,3);
d_q_tf=zeros(1,3);
move_fr_end_effector=traj3(t0,tf,q_t0,d_q_t0,q_tf,d_q_tf,tick); %xy 轨迹
if tick<planner.Ts/2 %z 轨迹
q_t0=[0 -body.shoulder_size/2 planner.stand_height];
q_tf=[0 -body.shoulder_size/2 planner.stand_height+planner.H];
move_fr_end_effector_z=traj3(t0,tf/2,q_t0,d_q_t0,q_tf,d_q_tf,tick);
else
q_t0=[0 -body.shoulder_size/2 planner.stand_height+planner.H];
q_tf=[0 -body.shoulder_size/2 planner.stand_height];
move_fr_end_effector_z=traj3(t0,tf/2,q_t0,d_q_t0,q_tf,d_q_tf,tick-
planner.Ts/2);
end
move_fr_end_effector(3)=move_fr_end_effector_z(3);
%leg rr
q_t0=[planner.S/2 -body.shoulder_size/2 planner.stand_height];
q_tf=[0 -body.shoulder_size/2 planner.stand_height];
d_q_t0=zeros(1,3);
d_q_tf=zeros(1,3);
move_rr_end_effector=traj3(t0,tf,q_t0,d_q_t0,q_tf,d_q_tf,tick); %xy 轨迹
%leg fl
q_t0=[planner.S/2 body.shoulder_size/2 planner.stand_height];
q_tf=[0 body.shoulder_size/2 planner.stand_height];
d_q_t0=zeros(1,3);
d_q_tf=zeros(1,3);
move_fl_end_effector=traj3(t0,tf,q_t0,d_q_t0,q_tf,d_q_tf,tick); %xy 轨迹
if tick>tf
move_rl_end_effector=[0 body.shoulder_size/2 planner.stand_height];
move_fl_end_effector=[0 body.shoulder_size/2 planner.stand_height];
move_rr_end_effector=[0 -body.shoulder_size/2 planner.stand_height];
move_fr_end_effector=[0 -body.shoulder_size/2 planner.stand_height];

```

```

%if contact_state(2) == 1 && contact_state(3) == 1
t_last_state=t;
state = 12;
%end
end
elseif state == 12% 维持直立 2 状态
tick=t-t_last_state;
%leg r1 f1
move_r1_end_effector=[0 body.shoulder_size/2 planner.stand_height];
move_f1_end_effector=[0 body.shoulder_size/2 planner.stand_height];
move_rr_end_effector=[0 -body.shoulder_size/2 planner.stand_height];
move_fr_end_effector=[0 -body.shoulder_size/2 planner.stand_height];
end

%% f1 to joint
[inv_s,inv_u,inv_k] = kinematics_inverse(move_f1_end_effector, 1)
f1_tgt_ang_s = inv_s;
f1_tgt_ang_u = inv_u;
f1_tgt_ang_k = inv_k;
tgt_ctrl_f1 = [f1_tgt_ang_s;f1_tgt_ang_u;f1_tgt_ang_k;has_u_ctrl(1)];

%% fr to joint
[inv_s,inv_u,inv_k] = kinematics_inverse(move_fr_end_effector, 2)
fr_tgt_ang_s = inv_s;
fr_tgt_ang_u = inv_u;
fr_tgt_ang_k = inv_k;
tgt_ctrl_fr = [fr_tgt_ang_s;fr_tgt_ang_u;fr_tgt_ang_k;has_u_ctrl(2)];

%% r1 to joint
[inv_s,inv_u,inv_k] = kinematics_inverse(move_r1_end_effector, 3)
r1_tgt_ang_s = inv_s;
r1_tgt_ang_u = inv_u;
r1_tgt_ang_k = inv_k;
tgt_ctrl_r1 = [r1_tgt_ang_s;r1_tgt_ang_u;r1_tgt_ang_k;has_u_ctrl(3)];

%% rr to joint
[inv_s,inv_u,inv_k] = kinematics_inverse(move_rr_end_effector, 4)
rr_tgt_ang_s = inv_s;
rr_tgt_ang_u = inv_u;
rr_tgt_ang_k = inv_k;
tgt_ctrl_rr = [rr_tgt_ang_s;rr_tgt_ang_u;rr_tgt_ang_k;has_u_ctrl(4)];

ctrl_state = state;
debug = t;

```

end

代码功能：机械臂抓取小球

代码：

```
function [theta1,theta2,theta3,theta4,theta5,theta6,d7] = fcn(t)
theta=[0 0 0 0 0 0 0];
C=zeros(4,7);
if(t <= 41) %靠近小球
t0=0;
tf=22;
q_t0=[0 0 0 0 0 0 0];
q_tf=[0 0 0 0 0 0 0];
tick=t;
elseif(t <= 45) %爪手靠近小球
t0=0;
tf=4;
q_t0=[0 0 0 0 0 0 0];
q_tf=[-2*pi/3 -2.79 -1.70 0 0 0 -0.015];
tick=t-41;
elseif(t <= 46) %爪手靠近小球
t0=0;
tf=1;
q_t0=[-2*pi/3 -2.79 -1.70 0 0 0 -0.015];
q_tf=[-2*pi/3 -2.79 -1.70 -0.47 0 0 -0.015];
tick=t-45;
elseif(t <= 47) %爪手抓取小球
t0=0;
tf=1;
q_t0=[-2*pi/3 -2.79 -1.70 -0.47 0 0 -0.015];
q_tf=[-2*pi/3 -2.79 -1.70 -0.47 0 0 0];
tick=t-46;
elseif(t <= 48) %爪手上移
t0=0;
tf=1;
q_t0=[-2*pi/3 -2.79 -1.70 -0.47 0 0 0];
q_tf=[-2*pi/3 -2.79 -1.70 0 0 0 0];
tick=t-47;
elseif(t <= 53) %爪手回位
t0=0;
tf=5;
q_t0=[-2*pi/3 -2.79 -1.70 0 0 0 0];
```

```

q_tf=[0 0 0 0 0 pi/2 0];
tick=t-48;
elseif(t<=91)%靠近云台
t0=0;
tf=38;
q_t0=[0 0 0 0 0 pi/2 0];
q_tf=[0 0 0 0 0 pi/2 0];
tick=t-53;
elseif(t<=94)%抓手靠近云台表面
t0=0;
tf=3;
q_t0=[0 0 0 0 0 pi/2 0];
q_tf=[-3*pi/7 -2.2 -1.9 -1.2 0 pi 0];
tick=t-91;
elseif(t<=95)%放置小球
t0=0;
tf=1;
q_t0=[-3*pi/7 -2.2 -1.9 -1.2 0 pi 0];
q_tf=[-3*pi/7 -2.2 -1.9 -1.2 0 pi -0.015];
tick=t-94;
elseif(t<=96)%抬起爪手，防止碰球
t0=0;
tf=1;
q_t0=[-3*pi/7 -2.2 -1.9 -1.2 0 pi -0.015];
q_tf=[-3*pi/7 0 -1.9 -1.2 0 pi -0.015];
tick=t-95;
elseif(t<=100)%抓手初始化
t0=0;
tf=4;
q_t0=[-3*pi/7 0 -1.9 -1.2 0 pi -0.015];
q_tf=[0 0 0 0 0 0 0];
tick=t-96;
else
t0=0;
tf=100;
q_t0=[0 0 0 0 0 0 0];
q_tf=[0 0 0 0 0 0 0];
tick=t-100;
end

d_q_t0=[0 0 0 0 0 0 0];
d_q_tf=[0 0 0 0 0 0 0];
C=traj33(t0,tf,q_t0,d_q_t0,q_tf,d_q_tf);
for i=1:length(q_t0)

```



```
theta(i)=C(1,i)+C(2,i)*tick+C(3,i)*tick.^2+C(4,i)*tick.^3;  
end  
theta1=theta(1);  
theta2=theta(2);  
theta3=theta(3);  
theta4=theta(4);  
theta5=theta(5);  
theta6=theta(6);  
d7=theta(7);  
end
```