

智能家庭服务机器人 Tinker

设计与实现 v6.0.1

(申请清华大学工学学士学位论文)

培养单位：计算机科学与技术系
学科：计算机科学与技术
研究生：霍江浩
指导教师：方斌

二〇二〇年三月

An Introduction to a Domestic Robot

Tinker v6.0.1

Thesis Submitted to
Tsinghua University
in partial fulfillment of the requirement
for the degree of
Bachelor of Science
in
Computer Science and Technology

by

Huo Jianghao

Thesis Supervisor: Fang Bin

March, 2020

关于学位论文使用授权的说明

本人完全了解清华大学有关保留、使用学位论文的规定，即：

清华大学拥有在著作权法规定范围内学位论文的使用权，其中包括：(1) 已获学位的研究生必须按学校规定提交学位论文，学校可以采用影印、缩印或其他复制手段保存研究生上交的学位论文；(2) 为教学和科研目的，学校可以将公开的学位论文作为资料在图书馆、资料室等场所供校内师生阅读，或在校园网上供校内师生浏览部分内容。

本人保证遵守上述规定。

(保密的论文在解密后应遵守此规定)

作者签名：_____

导师签名：_____

日 期：_____

日 期：_____

摘要

随着计算机技术的飞速发展以及智能制造水平的不断提高，智能机器人成为新的学科热点。其中家用智能机器人涉及领域众多，结构复杂，软硬件设计难度极大，且融合了机器人及相关行业的多项技术，具有极大的研究价值。

本文介绍了清华大学未来机器人团队设计实现的多功能家用探索型机器人 Tinker。本文以 Tinker 的设计与功能为核心，系统的介绍了智能机器人领域现阶段广泛使用的各种软硬件解决方案，并分析了在过去的几年时间中未来机器人团队在开发过程中积累的经验与教训。

本文还详细的给出了 Tinker 机器人实现的各项功能，详尽的给出了各个功能实现过程中的演进过程、最终方案、性能开销和大致的软硬件成本，并且辅以全面的图片、视频资料。

关键词：Tinker；Robotics Learning；Domestic service robotics

Abstract

Intelligent robot in domestic environment is regard of utmost importance in the field of industrial and every-day life. Therefore many efforts have recently focused on this problem.

In this paper, we describe the joint effort of the Team of Tinker in the past year. RoboCup@HOME consists of a settled set of benchmarking tests that cover multiple skills needed by domestic service robots. We present the hardware chosen, the approaches used and the system established to accomplish the tasks assigned in the paper, and also improvements achieved since RoboCup@HOME 2019. It includes a framework for behavior modeling and communication employed between robot and humans, as well as the policy decisions made within the robot itself. We describe our main contributions in arranging the logical model, integrated with various open source algorithm.

Key Words: Tinker; Robotics Learning; Domestic service robotics

目 录

第 1 章 项目简介	1
1.1 Tinker 机器人的设计目标	1
1.2 RoboCup@Home 简介	1
1.3 RoboCup@Home 比赛规则介绍	3
1.3.1 RoboCup@Home 经典任务分析	4
1.4 Tinker 的历史版本	4
第 2 章 Tinker 硬件设计介绍	7
2.1 Tinker 机械结构设计	7
2.1.1 Tinker 骨架搭建	7
2.1.2 Tinker 头部设计及版本迭代	8
2.1.3 Tinker 机械臂方案选择	8
2.1.4 Tinker 主控与供电	9
2.1.5 Tinker 底盘设计	10
2.2 Tinker 电路设计及通信设计	11
2.2.1 供电系统设计及演化	12
2.2.2 通信系统设计	13
第 3 章 Tinker 软件及算法介绍	15
3.1 软件设计概览	15
3.1.1 基本功能概览	15
3.1.2 控制逻辑的演进	16
3.1.3 编码规范与版本控制	17
3.2 算法方案选择与迭代	17
3.2.1 算法准备——传感器内外参标定	17
3.2.2 导航方案的迭代	20
3.2.3 抓取方案与环境维护	20
3.2.4 基于视觉的感知方案	20
3.2.5 语音交互方案的迭代	20
参考文献	21
声 明	22

第 1 章 项目简介

1.1 Tinker 机器人的设计目标

清华大学未来机器人团队成立于 2012 年，旨在帮助对机器人领域各个方面感兴趣的同学提升技术水平，增进校内交流，提供必要的设施支持，搭建一个校内的，开放有趣的机器人交流平台。随着团队的不断发展壮大，未来机器人团队逐渐积累了相当一批技术人才，并且开始有机会参于国际比赛。其中 Tinker 机器人作为团队的主要开发项目，每年固定参加著名的 RoboCup 比赛 @HOME 分赛场^[?]。

作为清华校内机器人方向最具影响力的学生社团，未来机器人团队在促进校内机器人相关领域的社区发展，提高同学们的技术水平方面起到了重要影响。其中 Tinker 作为未来机器人团队的重要支撑项目，在凝聚团队，提升技术水平，扩大队员眼界方面起到了重要的作用。并且 Tinker 本身的技术迭代和技术积累也对校内科创发展起到了极大的推动作用，这些珍贵的技术经验不仅惠及了未来机器人团队内部成员，也广泛的影响了校内科创领域的许多组织。

1.2 RoboCup@Home 简介

RoboCup@Home 作为机器人领域的国际顶级赛事，旨在帮助提升机器人在复杂的开放环境中的表现，促使家庭服务型机器人行业的发展。RoboCup@Home 赛



图 1.1 Tinker 外观

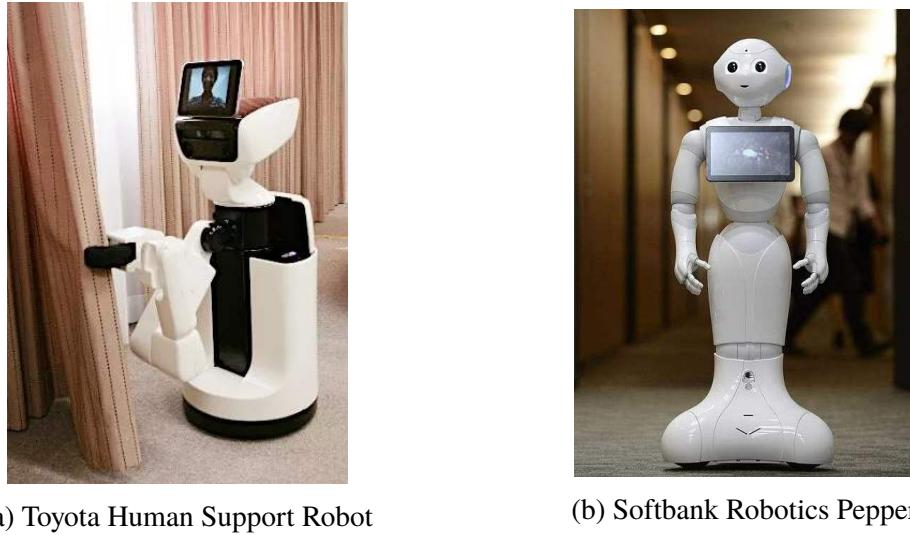


图 1.2 RoboCup@Home 指定机器人，左 HSR 右 Pepper

场分为三个联盟：Open Platform League (OPL)，Domestic Standard Platform (DSPL)，Socal Standard Platform League (SSPL)。

三大赛事联盟中 DSPL 与 SSPL 均指定机器人的比赛，开发者不允许使用除特定机器人之外的任何机器人参赛，且不允许替换、改装、外挂任何除机器人本身带有的部件，这两个联赛旨在促进社区内对相关算法及软件工程的提升。其中 DSPL 规定使用 Toyota Human Support Robot (HSR)^[1] 机器人，SSPL 规定使用 Softbank Robotics Pepper^[2] 机器人。两款机器人如图1.2所示。

OPL 为开放平台比赛，参赛者必须自行设计搭造参赛机器人，并且为机器人编写特定的软件完成比赛中制定的任务。OPL 旨在通过开放的硬件要求培养机器人领域结构、硬件相关的人才，提升各个高校和组织中的硬件设计能力，并探索结构、硬件方面的更多可能性。Robocup@Home 自 2009 年开始举办，多年的比赛中 OPL 赛场中涌现了大量的优秀机器人设计作品，Tinker 即为其中优秀的一员。除 Tinker 外，还有 homer@ UniKoblenz^[3]，Pumas^[4]，CATIE Robotics^[5] 等一批外观独特，功能强大的参赛作品出现，如图 1.3。

Robocup@Home 比赛每年举行一次，举行地点在国际几大城市巡回举办，由高校或者相关组织承办。通常比赛分为 3 个 stage，每个 stage 设有若干任务，参赛队伍一次完成任务，根据完成程度和完成时的表现（速度，流畅性等等）按点得分，分数靠前的队伍可以获得晋级资格，进入下一 stage。stage I 为基础功能比赛，主要考察避障导航、抓取、识别、语音交互等基础功能，辅以合理的机器人软件控制程序即可完成，这一个 stage 主要考察机器人基本的软硬件设计的自洽性及稳定性，相当多新手队伍会在 stage I 直接淘汰。

stage II 的任务内容同 stage I 基本相当，但是难度会更高，逻辑也更复杂，特



图 1.3 RoboCup@Home 指定机器人，左 HSR 右 Pepper

别的，在近几年的比赛中，增加了 General Purpose Service Robot (GPSR) 任务，即裁判对机器人下达任意在 RoboCup@Home 赛场上曾经出现过的任务，考察机器人的完成情况，这一任务旨在帮助参赛者更好的构建机器人控制软件，为家用机器人在真实场景下的应用做铺垫。

Final stage 为两个得分最高的团队关于冠军奖项的争夺，每年的任务并不固定，2020 年组委会将送餐机器人作为 Final Stage 的考察目标，要求参赛队伍在一个嘈杂的餐厅环境中将食物托盘送到某一个特定食客手上，并完成必要的交互。

1.3 RoboCup@Home 比赛规则介绍

经过多年的发展 RoboCup@Home 已经培育起一个完善的社区，每年 RoboCup@Home 的比赛任务即由社区相关人员协助商议决定，RoboCup@Home 组委会共同维护一份 RuleBook^[2]，组委会成员使用 GitHub 完成 RuleBook 的编写任务，该仓库是完全开放且欢迎参赛者提交修改与贡献的。RoboCup@Home 组委

会中，设有专门的任务制定部门 Technical Committee (TC)，这部分成员大都有深厚的相关从业背景和多年的研究经验，另外有一部分成员来自各个主力参赛队伍的核心开发人员（比如笔者作为 Tinker 机器人的主要程序员参与了 2020 年的 TC 工作），以确保比赛规则同时兼顾学科研究热点和可完成性。

随着机器人行业相关技术的不断发展，RuleBook 中的任务也在每年更新，并且向着越来越复杂、越来越贴近真实生活场景的方向演进。在多年的比赛中，组委会在任务内容、任务形式上也作出了很多改进与尝试，2019 年的 RoboCup@Home 比赛就及其大胆的将任务数量进行了爆炸式的扩容，尝试了很多新的复杂的任务，其中有一些被证明并不成功，因此 2020 年的规则制作过程中又将不合理的任务去除，将任务数量固定为 Stage I 5 个，Stage II 4 个。

1.3.1 RoboCup@Home 经典任务分析

1.4 Tinker 的历史版本

在近 5 年的参赛过程中，Tinker 机器人经过机器人队几代同学的不断开发，经历了若干次迭代，其硬件设计、软件架构、传感器使用、方案选择都发生了本质的变化。在 Tinker 不断迭代的过程中，机器人队也积累了大量的相关经验和教训，培养出了一代代机器人相关的人才。

早期机器人团队技术实力差资金紧张，因此我们自制了整个机器人参加比赛。当时的机器人如图 1.4 所示。当时的虽然工艺粗糙，精度很低，但是辅以合适的软件算法和恰当的策略，仍然在 RoboCup 的赛场上取得了不错的成绩。当时的队员们如今有的已经走上工作岗位，有的还留在学校继续攻读更高的学位，都在各自的领域中继续发挥自己的能量。

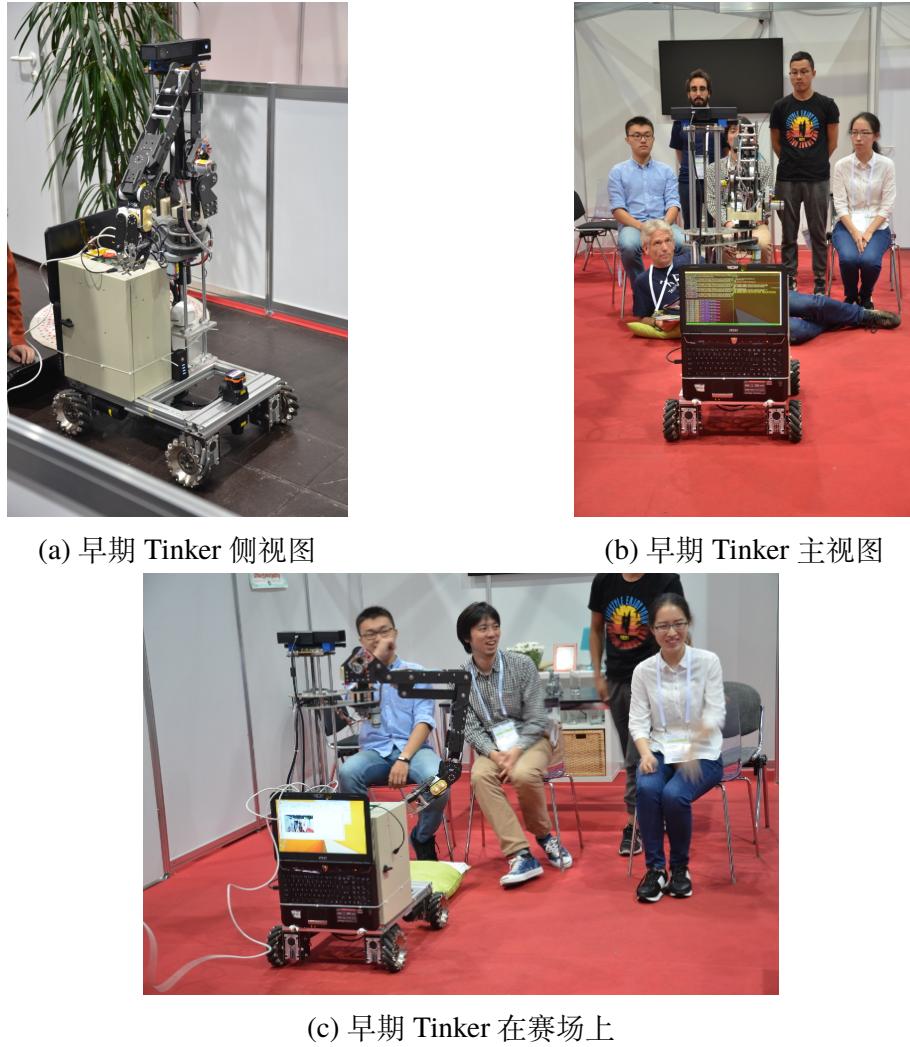
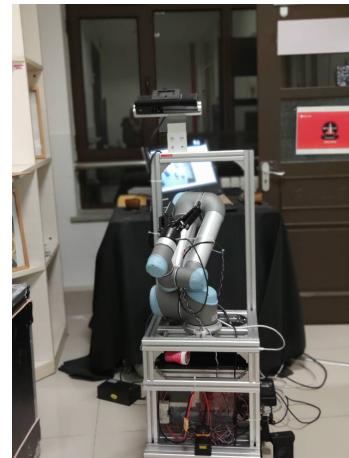


图 1.4 RoboCup@Home 指定机器人，左 HSR 右 Pepper

经过几年的积累，机器人团队于 2019 年对 Tinker 进行了彻底的改装，包括使用市售成品机械臂及夹爪、使用更高性能的笔记本和更高性能的电机等等改造，外观如图 1.5(a)。硬件的大幅改进对机器人开发带来了极大的冲击，新版 Tinker 最初的很多设计并不合适，因此经过 2019 年 RoboCup@Home 比赛后，我们又对 Tinker 进行了一次大规模的改装。第二次改装吸取了前面的教训，大幅裁剪了很多冗余设计，改良了供电和控制电路方案。在软件上也有诸多变化，一方面固化了许多经过前人验证的优良开发范式，一方面积极缩减代码依赖，在轻量化和稳定性方面不断提高，以争取更好的赛场表现。改装后的 Tinker 外观如图 1.5(b)所示。



(a) 改装 Tinker 1.0



(b) 改装 Tinker 2.0

图 1.5 新版 Tinker 二次改装前后的外观，左改装前，右改装后

第2章 Tinker 硬件设计介绍

2.1 Tinker 机械结构设计

按结构划分，可以将 Tinker 的机械结构粗略的划分为头部，机械臂，骨架，身体主控以及底盘。机器人的整体外观如图2.1所示。

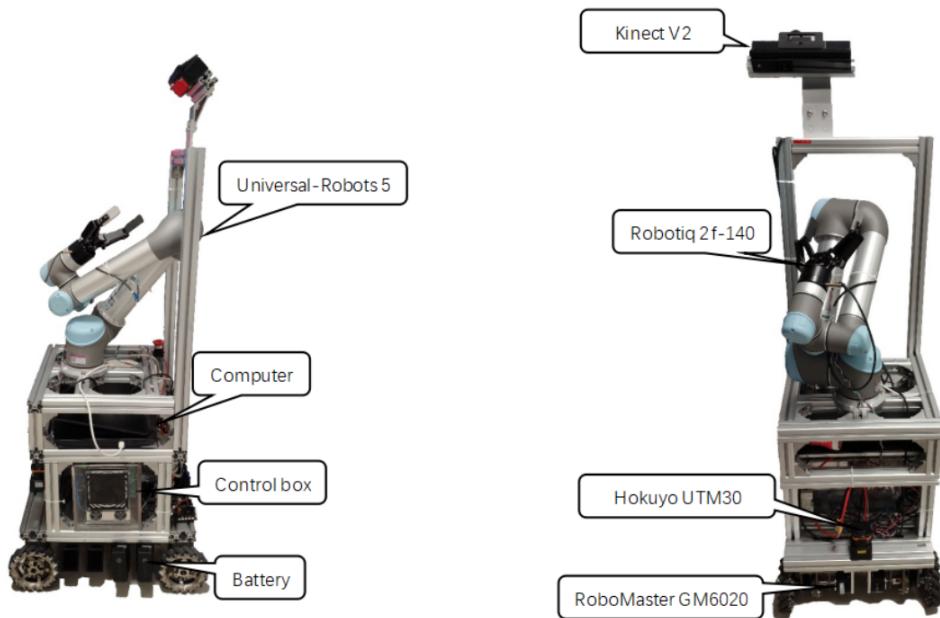


图 2.1 Tinker 前视图与侧视图

2.1.1 Tinker 骨架搭建

为方便交通运输与结构迭代，Tinker 基本骨架使用 40cm*40cm 铝型材搭建。骨架的主要作用为搭建起整个机器人的框架，将各个传感器假设到理想的位置上，并且为机器人的主控、供电、走线留出足够的空间。经过多轮版本迭代后，目前 Tinker 底盘尺寸固定在 35cm×45cm，净高 140cm，总质量 70kg。

为了方便 Tinker 机器人机械结构长期的维护，降低人员交接成本，我们将 Tinker 使用的型材尺寸做了规范化处理，并且对骨架进行了手动建模，以方便机器人的拆装与改进，如图2.2所示。

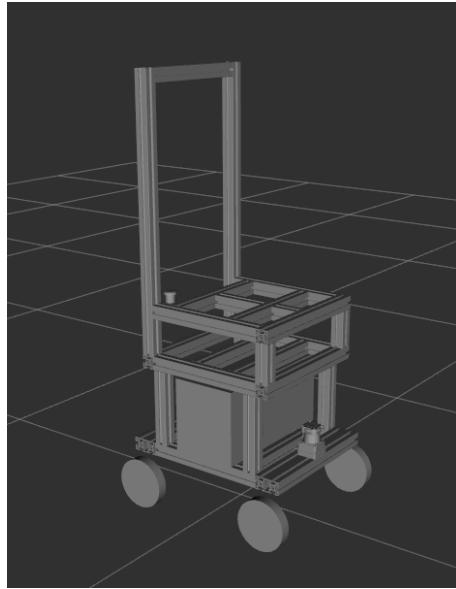


图 2.2 Tinker 骨架的 STL 建模结果

2.1.2 Tinker 头部设计及版本迭代

新版的 Tinker 头部主要用来放置顶部摄像头，在背景环境特别复杂的情况下还会在头部旁边加装专业的采音麦克风用于语音识别和声源定位。在一般情况下使用头部安装的 kinect v2 摄像头自带的麦克风阵列即可满足需求。在 Tinker 中期我们曾经使用过 2 自由度云台支持头部摄像头，使头部摄像头具有上下和左右转动的能力，以扩大摄像头的视野，如图2.3 所示。该转台是机器人团队根据现有的通用二自由度转台自行改装的，主要改动有：更换了原有的舵机，改为精度更高且有位置反馈的 dynamixel（TODO：具体型号需要核实）舵机，并且相应的改装了转台本身的尺寸和各种装配零件。客观上这一设计确实提高了机器人头部传感器的视野，但是经过漫长的实践，这一结构被证明是一个彻头彻尾的失败设计。原因有二：dynamixel 本身需要合理的供电和信号线，摄像头本身也有相应的传输线，这些线在频繁转动的时候很容易被拉断造成整个控制系统崩溃；摄像头本身作为抓取和部分避障的主力传感器，其位置精度是非常重要的，但是由于舵机空程以及装配过程中产生的空程等等问题，转台的位置精度一直无法达到要求，大大减弱了摄像头的数据效果。因此在新版的 Tinker 中，转台这一设计被机器人团队彻底抛弃了。

2.1.3 Tinker 机械臂方案选择

Tinker 使用的机械臂为 Universal Robotics 公司生产的 6 自由度机械臂 UR5。在决定使用该机械臂之前我们对市面上在售的各种成品机械臂进行了广泛的调研，包括 RoboCup@Home 赛场上使用过的 Kinova2.4(a) 机械臂配合三指夹爪，Retinker

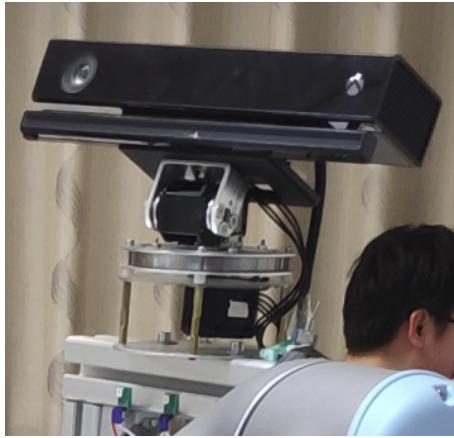


图 2.3 早期 Tinker 头部的 2 自由度转台

Robotics 公司开发的 Sawyer2.4(b) 机械臂，以及 Universal Robotics 公司出产的各个型号的机械臂2.4(c)。最终综合考虑了价格、售后、供电、开发者社区等各个因素后，我们最终确定使用了 UR5 这个型号的机械臂，并且对机械臂的控制与供电进行了适当的改装，以符合比赛的需要。

早期夹爪使用 Robotiq 公司生产的 Robotiq 2f-85 二指夹爪（最大张角 85mm），后经过不断的测试与赛场上的较量，我们发现 2f-85 的张角对于家庭机器人来说过窄，于是换用了 robotiq 2f-140 款式的二指夹爪，该夹爪张角为 140mm，能够更好的满足抓取物品、拾取袋子的需求。因此 Tinker 团队最终使用 UR5 + Robotiq 2f-140 作为最终比赛方案，如图2.7所示。

2.1.4 Tinker 主控与供电

Tinker 的核心控制由一台搭载了 i9-9900K cpu 及 RTX2080 显卡的笔记本电脑完成。在前几代 Tinker 设计中，一般主控由工控机或者经过电源改装的台式机担任，但是随着笔记本性能的不断增强，我们最终选择使用成品机器完成这一任务。成品笔记本性能好，散热完善且主板设计精良，稳定性好，并且笔记本有自己独立的供电系统，可在机器人断电时单独运行大大的提升了系统的稳定性和易用性。

整个机器人的供电系统都放在机器人腹部，包括机械臂、底盘、各种传感器、笔记本电脑所需的所有供电电路以及电池组。2019 年 Tinker 使用单一磷酸铁锂电池供电，标准输出电压 29.4V 到 24V，容量 50AH，放在底盘上部。经过 19 年比赛的反思之后，机器人队认为，使用单一锂电池供电太过危险，且参赛过程中运输太过麻烦，成本过高，于是我们将供电方案改成了多块标准 29V 1500mAH 电池并接供电的形式，并 3D 打印了特制的电池架将电池倒挂在底盘下，这样一方面大大缩小了 Tinker 底盘的面积提高了底盘避障导航的灵活性，另一方面降低了整个机器人的中心，使得机械臂运动时的型变更小，提高了机器人的性能。在供电电



(a) Kinova



(b) Sawyer



(c) UR 系列机械臂

图 2.4 机器人队前期调研时关注过得机械臂型号

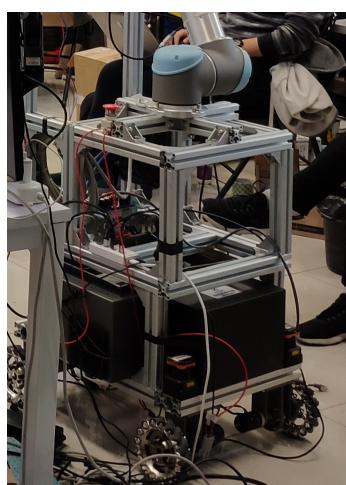
路布线方面，Tinker 也有了显著的改进，前一版本中 Tinker 直接借用了 UR5 自带的配电箱摆放机械臂及其他部件需要的供电线路、DCDC、分流排等等元件。但是配电箱本身尺寸过大，不透明，且强度过大，给机器人装配、调试带来了很大的困难，也对导航的表现造成了影响。之后我们在充分考虑了散热、消防等等制约之后，使用亚克利版重新制作了一版配电箱，且对机器人的供电线路进行了梳理，大大缩小了机器人的底盘尺寸，也使得整个供电设计更加简洁易于调试。前后两个版本的供电外观如图2.6所示。

2.1.5 Tinker 底盘设计

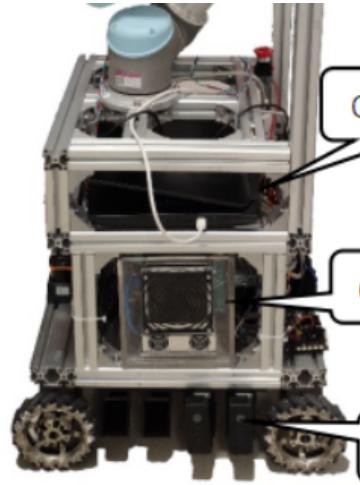
Tinker 的底盘设计受前几版机器人的影响最多，早期的 Tinker 使用自制的 3 自由度机械臂，其精度有限且自由度受限太严重，需要具有左右额外自由度的底盘来弥补这一不足，因此 Tinker 的底盘被设计为使用 4 个麦克纳母轮的万向底盘^[?]，底盘设计如图所示。



图 2.5 Tinker 最终使用的机械臂及夹爪方案



(a) 改版前的电池及供电线路放置



(b) 改版后的电池及供电线路放置

图 2.6 Tinker 前后两个版本的供电布局

TODO：插入麦轮的底盘的 solidworks 图。

新版 Tinker 改为成品机械臂后抓取的自由度和覆盖范围都有了质的飞跃，不再要求底盘具有万向移动的性能了。但是经过多年的改进，机器人团队中对万向底盘的制作技术已经成熟，于是我们继续沿用了这一设计，仅在电机选型上做了一些与时俱进的改良，选用了扭矩较大且噪音较小的大疆 GM6020 直流无刷电机做底盘的动力电机，如图 2 所示。

2.2 Tinker 电路设计及通信设计

Tinker 机器人作为一个有复杂功能和结构的完整机器人系统，经过长时间的开发和改进，其控制电路及各个部件之间的通信是及其复杂的。在漫长的演进过



图 2.7 Tinker 底盘的电机传动结构

程中，机器人队的队员们创造性的将软件设计中分定义接口、分层描述的概念引入到机器人的硬件设计中，对机器人的供电、通信均给出了清晰详尽的分层描述。这一描述大大的方便了机器人队成员之间的分工协作，提高了开发的效率。

2.2.1 供电系统设计及演化

新版 Tinker 在搭建之初由于时间紧迫，使用一台逆变器将电池输出的 29V 直流电源经过一台逆变器直接转到 220V 交流电（图2.9(a)所示），再连接各个执行控制部件与传感器。在一段时间内，这个方案虽然丑陋，但很好的满足了机器人的任务需求，为机器人队的发展作出了贡献。但是这个方案的缺陷也非常明显，逆变器本身效率很低，且体积不小，再加上市电转到各个传感器的开销，整个机器人变得异常臃肿，且续航极短。逆变器工作时还会发出高频啸叫，对机器人的安全和性能都有严重损害。

经过一段时间的调研，机器人队对 Tinker 的供电系统做了 2 次系统的改造，逐步的将供电改为直流直接供电，并且理顺了机器人内部的供电逻辑。其供电分层描述如图2.8 所示

在供电改造的过程中，最消耗精力的是 UR5 机械臂的供电改造。UR5 机械臂本身所需的功率极大且其本身的保护机制过于复杂，在机械臂上电时有一套复杂的自检逻辑，会在直流供电端尝试拉取一个十几 A 的大电流对电源进行冲击检测，并且在电流拉取结束后还会反复切断电路进行电源测试。这一逻辑过于复杂，市面上常用的 DC-DC 普遍无法满足需求。机器人队成员经过长时间的反复测试和大量的文献检索，最终使用明纬 SD-1000L-48V 开关电源（图2.9(b)所示）替代普通电源对机械臂单独供电才满足机械臂的供电改造需求。

概括的说，Tinker 的供电系统中有 4 个主要电压，分别是电池直接输出且底

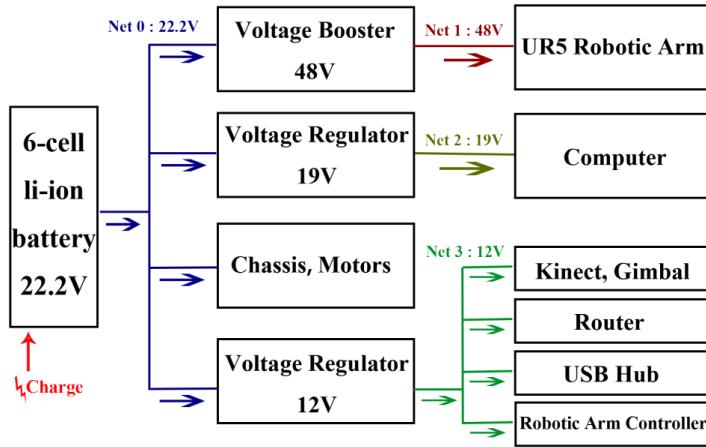


图 2.8 Tinker 的分层供电示意图



(a) 明纬 TS-700-248B 700W 逆变器 (b) 明纬 SD-1000L-48V 开关电源

图 2.9 Tinker 电路改造中使用过的供电元件

盘直接使用的 22.2V，机械臂工作电路需要的 48V，笔记本电脑充电需要的 19V，各个传感器、路由器以及机械臂主控需要的 12V 电压。除机械臂供电使用了明纬 SD1000L-48 开关电源外，其他变压需求均使用了一般市售的大功率直流 DCDC 电源。在经过两轮改版后，Tinker 的供电系统以基本稳定，目前 Tinker 整机上电待机时长能达到 1.5h 左右，高负荷运行时（机械臂高负荷运行且同时执行高强度计算任务，或者底盘不间断运行）可连续工作 20 分钟左右，基本满足比赛需求。

2.2.2 通信系统设计

由于 Tinker 使用了大量的成品部件，包括机械臂、夹爪、电机、各种传感器等等，其通信设计受限比较大。机器人团队使用开源且在机器人科研领域广泛的 ROS 框架完成高层数据及控制通信，辅以必要的底层通信控制，完成整个系统的数据获取与命令执行功能，其分层设计示意图如 2.10 所示。

头部 Kinect、激光雷达及额外的摄像头（一般安装在机械臂腕部，为一台 realsense d435）通过 usb 直接连接到中央控制电脑上，通过 ros 驱动打开，获取的数据流直接发布到 ROS 的通信系统中，夹爪使用串口同样连接到电脑上，使用 ROS 驱动进行控制。UR5 机械臂本身有主控机器，它通过路由器于中央控制电脑连接，

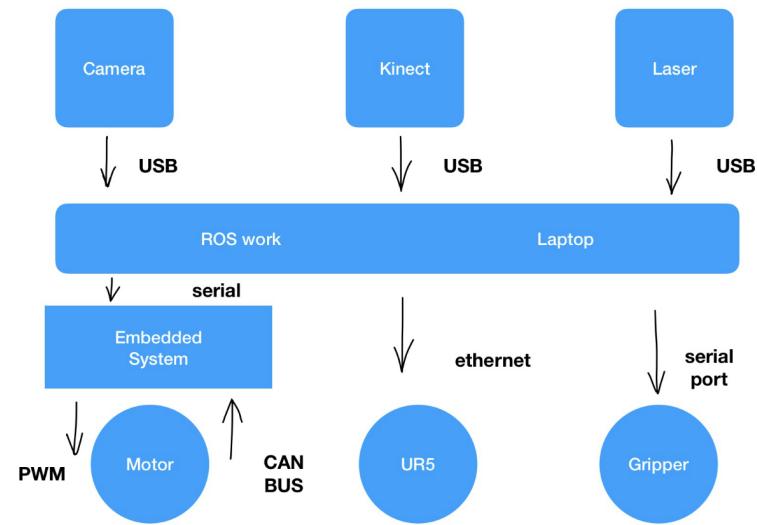


图 2.10 Tinker 的分层通信示意图

两者利用 ROS 的 ethernet 通讯机制通信。麦轮底盘的四个电机统一使用 CAN 总线连接到一块 STM32F4 单片机开发板上，开发板通过 PWM 控制电机转动，电机内部的编码器数据通过 CAN 回传到单片机中，单片机再通过串口与主控电脑通信，为了方便底盘接入 Tinker 的通信系统，我们使用 rosserial 对单片机的串口进行封装，完成控制底盘、获取底盘数据的功能。

第3章 Tinker 软件及算法介绍

Tinker 作为一台服务型机器人，一切设计都是为实现功能服务的。而编写软件则是实现功能的直接手段。近几十年，一代代程序员和学者们为程序的开发的管理提出了一个又一个理论，各式各样的开发方式和管理范式层出不穷，但真正面对实际项目时，人员的分配，时间的管理，甚至硬件的掣肘无处不在。这些制约笔者作为机器人队的主力程序员深有体会。本文作为 Tinker 机器人设计的客观描述论文无意对项目管理这种不太容易客观表述的内容进行长篇的论述，但是笔者深深的希望本文的读者能够在阅读本章关于软件设计及算法选型的内容中意识到：程序的编写从来不以单一性能的最优为目的，而是在平衡整体功能的完成和系统稳定性过程中的产物。代码的美丽与优雅固然可贵，但是无法实现需求的程序就是一摊牛屎。

3.1 软件设计概览

本节主要大致讲述 Tinker 需要实现的基本功能以及将基本功能串联的控制逻辑的实现。并且大致介绍一下 Tinker 搭载的各种主要传感器及它们的主要性能参数。具体的方案及方案演进将在下一节进行详尽的展开。

3.1.1 基本功能概览

基础功能是支撑机器人完成复杂任务的基石，按照机器人领域的研究方向和习惯约定，可以将机器人的功能大致的划分为 4 个主要方面。

- 导航：建图、定位、路径规划与避障
- 抓取：机械臂运动规划与避障
- 感知：主要为基于视觉及 RGBD 的人脸检测、物体识别、及人类与物品的追踪
- 语音交互：主要为语音识别以及语音生成，以及简单的自然语言处理

这几大方向在学术界都有自己完善发展，社区中也各自有相当可用的几种开源方案。但是将这些基本功能妥善的实现并且开放相应的功能接口供主控逻辑调用以完成复杂的功能仍非易事。Tinker 的开发过程中，每一种基本功能都经过了至少 2 版的迭代，方案也几经更替，除了算法表现和稳定性的考量外，cpu 和显卡占用，代码开放性，社区成熟度等因素都有着巨大的影响。

3.1.2 控制逻辑的演进

RoboCup@Home 赛场上完成的任务本质上就是机器人通过调用几大基础功能实现状态的转移，其间发生的结果处理以及异常管理功能其实都可以通过一个类似状态机的机制来处理。基于这一思想，新版 Tinker 早期使用了 ROS 社区一个比较出名的状态机控制包 smach 进行功能调用，完成主控逻辑的编写。

smach 是一个较成熟的使用 python 编写的状态机控制仓库。它不但提供了完善的状态命名、转移、分层分块机制，还显式的提供了良好的异常处理机制（如图3.1 所示）。早期的 Tinker 控制逻辑几乎完全依赖于 smach 实现，甚至整机的软件上电控制都由 smach 编写。但是在经过反复的测试后 smach 的缺陷很快暴露出来，首先，Tinker 使用的很多基础功能是由 c++ 编写的，在 smach 中想要直接调用他们不太现实，需要额外开发 python 的 wrapper 或者使用 ros 的消息通信机制来调用他们，这在客观上大大的增加了编码的工作量，且引入了非常多的 bug，降低了系统的稳定性。其次，使用 python 编码变成了一个门槛，将很多同学拦在开发之外，python 近几年是软件界的新宠，受到广大程序员及学者的追捧，但是不得不承认我校非计算机专业的很多院系对同学们提供的编程教育是滞后的。相当多非计算机专业且无计算机竞赛背景的同学刚加入机器人队时几乎完全不会写 python，学习成本也不太低。综合上述两个原因，除已经经过反复测试异常稳定的部分外，目前 Tinker 大部分新编写的逻辑都是由 c++ 编写的，只在开发简单的辅助工具或者脚本时才使用 python 或者 shell 语言编写。

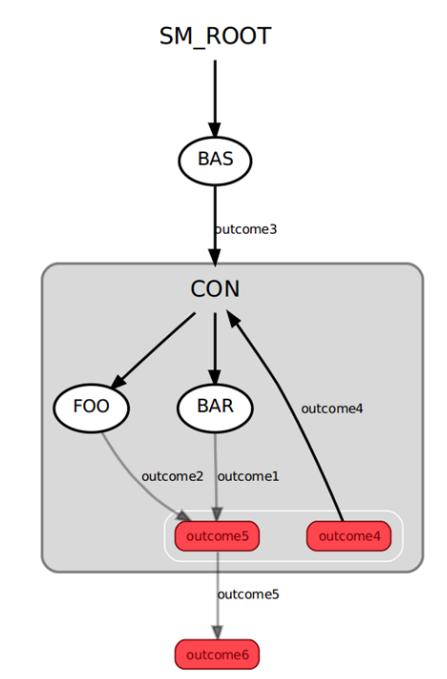


图 3.1 smach 的典型状态图

当然 c++ 编写会引入新的问题，比如 c++ 可使用的功能性的包较少，包管理也几乎没有，进行异常处理时非常麻烦等等。但是同完成功能协调队员合作比起来，这些问题机器人团队也只能暂时忍耐，并不断寻求新的解决方案了。

3.1.3 编码规范与版本控制

在3.1.2节中本文曾简短的探讨过开发语言的问题，以及 python 在机器人开发过程中的劣势。事实上除了上述所说的困难外，使用 python 编写软件还存在一个比较致命的缺陷，那就是 python 缺乏严格的类型检查机制。和 c 及 c++ 不同，python 这种脚本语言缺乏“编译器”这个重要角色。这就导致在开发过程中少了一个重要的约束，当项目合作的人员增多，代码开始膨胀的时候，相当多的函数参数开始返回约定的规范之外的数据（这是程序员的惰性决定的），机器人队没有条件像现代软件企业那样为研发的同学提供强有力的测试和 review 职能。这就导致预期之外的千奇百怪的数据对象在层层函数中肆无忌惮的传递，最终在某处崩溃，让正在调试的程序员陷入一种尴尬的莫名其妙的境地，久久无法找到错误原因，甚至无法稳定复现。因此笔者个人认为，在实现小而轻的功能时，python 固然是一把有力的匕首，但是在开发功能复杂规模庞大的软件时，使用 python 进行开发对开发团队素质的要求比使用 c++，java，c# 这种工程语言的要求高的多。

长期以来，机器人队在开发基于 ROS 的 c++ 程序时遵循 Google Coding Style 规范，并使用 git 作为主要的版本控制工具。尽管在培训队员使用 git 方面我们付出了一些努力，甚至曾经出现过灾难性的损失，但总的来说这些代价是值得的，Tinker 的软件迭代一直以稳定可控的方式进行着，且随着开发时间的曾长，Tinker 的功能和稳定性都有着长足的进展。

3.2 算法方案选择与迭代

3.2.1 算法准备——传感器内外参标定

对于多传感器机器人而言，各传感器之间的时间、位置校准，以及传感器本身的校准对于系统的可用性和方法的有效性有至关重要的影响。在开发过程中，机器人团队充分的认识到准确的传感器内参和校准是任何算法的基石。在不断探索中，我们逐步形成了一套稳定的内外参标定流程。

3.2.1.1 摄像头内参标定

为了将相机图像与真实世界联系起来，我们需要掌握基本的相机投影模型的知识。同时，由于工艺及装配误差的影响，摄像头采集到的图像会产生一定程度

的畸变，下面本文将会分别介绍相机基本光学模型以及1种通用的畸变拟合模型。

无畸变针孔相机模型

理想相机被认为是一个完美的小孔成像装置，如图3.2所示所有光线均穿过相机透镜的光心，投影到像平面上。

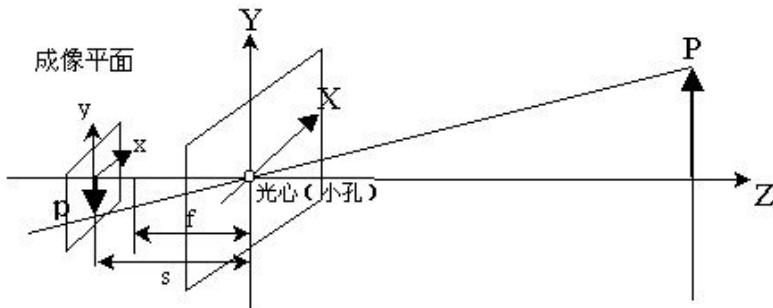


图 3.2 小孔成像原理示意图

考虑到相机将图片从 CMOS 中读取出来制成完整图片时，xy 方向会有不同程度的拉伸，因此，我们分别使用两个参数 f_x, f_y 表示 x 方向和 y 方向上的光心，正常情况下， f_x, f_y 即使相互不同也不应当相差太大。另外，在从相机坐标向图片坐标转换时，需要将像平面上的点统一平移一个便宜量，这样使得所有像点在图片中的坐标均大于 0。因此，假设我们有一 3 维世界坐标点 $[X, Y, Z]$ ，在图片中对应的图像坐标为 $[u, v]$ ，则二者之间的转换可表示为：

$$\begin{cases} u = f_x * \frac{X}{Z} + c_x \\ v = f_y * \frac{Y}{Z} + c_y \end{cases} \quad (3-1)$$

经过整理可得到较为方便书写的版本：

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \frac{1}{Z} \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

一种通用的相机校准模型——Kannala Brant 模型

如??所示，理想状态下，图片中记录下来的物体形状应当与真实世界中的物体满足相似关系，但是由于相机透镜的工艺、相机装配时的误差，真实相机获取的图片中会发生畸变。Kannala Brant 模型^[6] 提供了一种较为普遍且有效的畸变拟合手段，在工程实践中被广泛的应用在窄视角以及鱼眼相机的正畸过程中。

按照畸变产生的效果来分，主要存在两种畸变，分别是径向畸变及切向畸变。

径向畸变产生的主要原因为相机透镜的折射率无法做到完全一致，径向畸变是以图像中心为中心，按点到中心的绝对距离 r 为半径对称分布的，一般来说，径向畸变有桶型和枕型两种，两种畸变如图3.3所示。尽管两种径向畸变在表现上不太一样，但是在使用 Kannala Brant 模型时，可以使用同一种近似手段来拟合两种畸变。

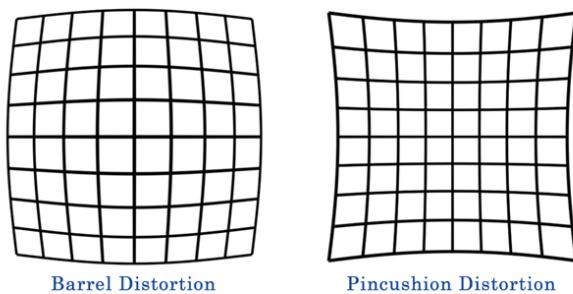


图 3.3 左桶型畸变右枕型畸变

切向畸变则是由于相机装配过程中，相机透镜平面与 CMOS 感光平面不完全平行造成的，具体造成原因如图3.4所示。尽管在各种相机校准方法中，关于切向畸变都有对应的处理手段，但是在实际工程应用中，由于切向畸变一般较小，影响也较小，因此大多数情况下都不予处理。Tinker 的标准校准流程中，对切向畸变也不予处理。

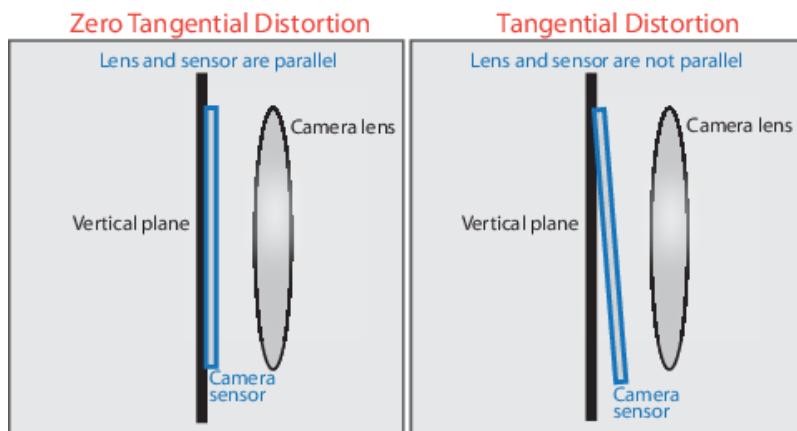


图 3.4 左正常装配结果右有装配误差的装配结果

Kannala Brant 模型使用光线的入射角 θ 对图片上对应点的径向距离 r 进行拟合，拟合形式如下式所示。

$$r(\theta) = k_1\theta + k_2\theta^3 + k_3\theta^5 + k_4\theta^7 + \dots \quad (3-2)$$

根据工程实践经验，使用 KB 模型校准时，将 k_1 取为 1，并对拟合参数 k 取到 5 阶，即有效畸变参数为 $[k_2, k_3, k_4, k_5]$ 。如图 TODO 为 Tinker 顶视摄像头校准前后的图片。

TODO：插入一张校准前后的图片

3.2.2 导航方案的迭代

3.2.2.1 路径规划与机器人控制

3.2.2.2 SLAM 方案的迭代与性能分析

3.2.3 抓取方案与环境维护

3.2.4 基于视觉的感知方案

3.2.5 语音交互方案的迭代

参考文献

- [1] Toyota shifts home helper robot r&d into high gear with new developer community and upgraded prototype [EB/OL]. <https://global.toyota/en/detail/8709541/>.
- [2] Pandey A K, Gelin R. A mass-produced sociable humanoid robot: pepper: the first machine of its kind. *IEEE Robotics & Automation Magazine*, 2018, 25(3):40-48.
- [3] Memmesheimer R, Seib V, Paulus D. homer@ unikoblenz: winning team of the robocup@ home open platform league 2017 // Robot World Cup. Springer, 2017: 509-520.
- [4] Savage J, Negrete M, Matamoros M, et al. Pumas@ home 2013 team description paper. *RoboCup*, 2013.
- [5] FABRE R, ALBAR B, DUSSIEUX C, et al. Catie robotics@ home 2019 team description paper. 2018.
- [6] Kannala J, Brandt S S. A generic camera model and calibration method for conventional, wide-angle, and fish-eye lenses. *IEEE transactions on pattern analysis and machine intelligence*, 2006, 28(8):1335-1340.

声 明

本人郑重声明：所呈交的学位论文，是本人在导师指导下，独立进行研究工作所取得的成果。尽我所知，除文中已经注明引用的内容外，本学位论文的研究成果不包含任何他人享有著作权的内容。对本论文所涉及的研究工作做出贡献的其他个人和集体，均已在文中以明确方式标明。

签 名： _____ 日 期： _____