[](http://news.cau.edu.cn/upload/2005/caunews_20050711181514.jpg)

**本科生毕业论文（设计）**

基于手机屏书写记录的机械臂模仿书写系统设计与实现

The design and implementation of robot arm imitation writing system based on the writing record of mobile phone screen

学生姓名： 陈友亮

指导教师： 胡标副教授

合作指导教师： 无

专业名称： 机械电子工程

所在学院： 工学院

2022年 5 月

**摘要**

随着各种通信技术和自动化技术发展越来越繁荣，作为人类工业生产自动化中重要的一名角色，机械臂的应用范围越来越宽广，目前已被广泛用于各种工业生产、医疗行业、航空航天等行业。并且随着人类对技术创新的追求和无限制的想象，人类对机械臂的应用也越来越千奇百怪，性能也逐渐提高以满足各种需求。本设计以UR5机械臂为研究对象，进行了一种基于手机屏远程指定轨迹的机械臂仿写方法。搭建了用于采集用户书写轨迹的网页平台，配置了与其配套的Nodejs服务器，同时还搭建了基于ROS系统的仿真平台，在ROS对整个控制方法进行了实验和仿真。并基于ROS对实际UR5机械臂进行控制，同时为其添加了用于夹持笔的末端结构，以方便验证轨迹的再现性，最终验证了该控制方法的可行性。

**关键词：**网页画板编写 Nodejs服务器 ROS UR5机械臂

**Abstract**

With the development of various of communication technologies and automation technologies, the robotic arm plays an important role in the automation of human industrial production, and its applications are becoming more and more extensive. The robotic arm has been widely used in a variety of industrial production, medical industry, aerospace and other industries. With the pursuit of technological innovation and unlimited imagination of human beings, the application of robotic arms is more and more diverse, and their performance is gradually improved to meet various needs. This design takes UR5 robotic arm as the tool and implements an imitation writing method of robotic arms based on the remote specified trajectory of mobile phone screen. During the study ,a web page platform for collecting user writing trajectory is built, and its supporting Nodejs server is configured. Meanwhile, a simulation platform based on ROS system is built, and the whole control method is tested and simulated in ROS. The actual UR5 robotic arm is controlled based on ROS. After the simulation, the end structure for holding the pen is added to verify the reproducibility of the trajectory. Finally, the feasibility of the control method is verified.

**Keywords:** Webpage drawing, Nodejs server, ROS, UR5 manipulator.

**目 录**

[第一章 绪论 1](#_Toc105172237)

[1.1 课题研究目的及意义 1](#_Toc105172238)

[1.2 研究现状 2](#_Toc105172239)

[1.3 课题研究的主要内容 5](#_Toc105172240)

[1.4 设计方案简介 6](#_Toc105172241)

[第二章 绘画网页设计 7](#_Toc105172242)

[2.1 HTML、CSS、JS、Apache及其功能介绍 7](#_Toc105172243)

[2.2 可视轨迹的画板 8](#_Toc105172244)

[2.3 撤销、反撤销与清空 10](#_Toc105172245)

[2.4 记录并输出坐标 12](#_Toc105172246)

[2.5 局域网访问网页 13](#_Toc105172247)

[第三章 服务器端设计 14](#_Toc105172248)

[3.1 NodeJs、Socekt通信及功能介绍 14](#_Toc105172249)

[3.2 服务器编写和运行 14](#_Toc105172250)

[第四章 机械臂 16](#_Toc105172251)

[4.1 UR机械臂 16](#_Toc105172252)

[4.2 ROS系统简介 16](#_Toc105172253)

[4.3 基于MoveIt/Rviz仿真和控制机械臂 19](#_Toc105172254)

[4.4 基于External Control的机械臂控制 26](#_Toc105172255)

[4.5 机械臂末端执笔结构设计 28](#_Toc105172256)

[第五章 实验结果 30](#_Toc105172257)

[5.1不同运动规划算法的比较 30](#_Toc105172258)

[5.2不同时间参数化算法的比较 31](#_Toc105172259)

[5.3结果展示 32](#_Toc105172260)

[第六章 研究成果 35](#_Toc105172261)

[总结 36](#_Toc105172262)

[参考文献 38](#_Toc105172263)

[致谢 40](#_Toc105172264)

[附录A 41](#_Toc105172265)

[ROS安装过程 41](#_Toc105172266)

[附录B 42](#_Toc105172267)

[ROS与机械臂相关配置 42](#_Toc105172268)

[作者简介 44](#_Toc105172269)

**插图与附表清单**

图[1-1 基于TCP/IP的七自由度机械臂远程控制系统 3](#_Toc105172238)

图[1-2 基于远程控制的写字机器人结构 4](#_Toc105172238)

图[1-3 智能仿人写字的COREXY机器人 4](#_Toc105172238)

图[1-4 仿人写字机器人 5](#_Toc105172238)

图[1-5 技术路线图 6](#_Toc105172238)

表[2-1 鼠标事件 9](#_Toc105172238)

表[1-2 事件触发函数 9](#_Toc105172238)

图[2-1 JavaScript事件函数程序图 1](#_Toc105172238)0

图[2-2网页输出坐标信息 12](#_Toc105172238)

图[2-3 Phpstudy局域网站点文件夹 13](#_Toc105172238)

图[3-1 服务器正常运行图 15](#_Toc105172238)

图[4-1 UR机械臂 16](#_Toc105172238)

图[4-2 ROS文件系统级 18](#_Toc105172238)

图[4-3 Moveit框架图 20](#_Toc105172238)

图[4-4规划流程图 21](#_Toc105172238)

图[4-5 Rviz可视化工具 23](#_Toc105172238)

图[4-6 坐标处理流程图 24](#_Toc105172238)

图[4-7 末端结构 28](#_Toc105172238)

表[4-1 末端结构各零件介绍 28](#_Toc105172238)

图[4-8 末端结构实物图 29](#_Toc105172238)

图[5-1 原轨迹 30](#_Toc105172238)

图[5-2 OMPL 30](#_Toc105172238)

图[5-3 OMPL+CHOMP 30](#_Toc105172238)

图[5-4 OMPL+STOMP 30](#_Toc105172238)

表[5-1 各算法实验结果 30](#_Toc105172238)

图[5-5 TOTG速度曲线 31](#_Toc105172238)

图[5-6 ISP速度曲线 31](#_Toc105172238)

图[5-7 IPTP速度曲线 31](#_Toc105172238)

图[5-8 网页绘画效果 33](#_Toc105172238)

图[5-9 机械臂仿写效果 33](#_Toc105172238)

图[5-10 二值化处理结果 34](#_Toc105172238)

图[5-11 相似度分析 34](#_Toc105172238)

# 第一章 绪论

## 1.1 课题研究目的及意义

### 1.1.1课题简介：

本课题旨在开发一种可以通过移动设备驱动同一网络内的机械臂远程仿写的系统。机器人的模仿书写是一种重要的人机交互方式，在本课题中，通过人在手机屏上书写，手机上浏览器运行的JavaScript脚本将书写轨迹记录下来，根据这些记录数据，机器人末端模仿相应的书写轨迹。

### 1.1.2研究背景：

传统的控制系统表现出来的可扩展性差，效率低，布线复杂，成本高等问题使得传统的控制越来越难以满足现如今系统对性能的高要求[1]。为了满足各领域的开发需求，网络化控制系统诞生了，通过网络通信协议实现实时闭环控制系统，使得各个部分能够通过网络实时传通信。网络化控制系统，创建了一个以信息流动为主要技术难点和特征的交叉学科研究方向。目前网络化的控制系统已经逐渐被应用在社会的各个领域，因此，对其深入地研究是具有重大意义的[1]。

与网络化控制系统一样，作为人类自动化发展过程中不可缺少的一员，机器人也是被热门研究的对象之一，机器人学作为一门高度融合交叉的跨领域的前沿学科，包含大多数理工科学科，如各种力学、机械设计、计算机、控制理论与控制工程等[1]。可以说，机器人技术是涉及国家未来发展及前沿科学技术的核心。机器人的发明主要是为了代替人执行他们所不能完成的工作或者是不愿意完成的简单重复性劳动。一方面，在一些对人类有毒害的恶劣环境下，机器人可以代替人工执行任务[1]。另一方面，机器人可以完成一些单调重复的工作，减少人工不必要的耗费。伴随着新型传感器、新型材料以及新通信手段等技术的出现和繁荣，机器人的种类也在增加，其应用领域也随之扩展。

带有可操作机械手的机器人是最早出现的工业机器人，它取代了人的简单劳动，实现生产流水线的智能、自动化，能在对人类害的环境下进行工作，保护工作人员安全，因而被宽泛地应用于各个领域。随着科学技术不断的发展和创新，工业机器人也随之变得越来越智能。

而机器人书写作为智能机器人的一项非常新奇地应用领域，在机械臂兴起地早期就已经得到了较多的关注。书写动作主要是靠机械手的各个关节和其他传感控制设备的协同配合来完成。目前所说的机械臂，主要指的是多关节机械手爪及其控制系统，它能模仿人手臂的某些动作，是一种由程序控制可以实现指定目的的机械电子设备。目前机械手也是智能机器人研究领域重要的组成部分和课题之一。

将网络化控制技术与机器人技术相结合具有不同寻常的意义，目前相关技术已经涵盖了现代科技涉及的许多领域。

### 1.1.3研究意义

本课题所实现的可以远程在线实现书写的机械臂系统，可以实现各种即时在现指定平面轨迹的跟踪。在生活中的应用场景是非常广泛的。

机器人书写功能最直接的应用是个性化的艺术喷涂，特别是在一些对人来说没有很强安全保障的外墙上以及有毒工作车间进行涂鸦或是写字，都可以用机器人工作来提高工作效率；另一方面，在一些有毒害、重度污染环境下的工业喷漆以及个性化涂鸦的喷涂工作，也很适合能够在线指定轨迹书写的机器人去代替人进行工作。配合在线指定和执行的功能，可以丰富该类机器人在各种领域的功能。由于喷涂与书写能力是可以直接投入使用的，因此若在某个生产领域有需要时，可以直接投入实际应用，对生产企业来说其商业效益也是非常高的。

除此以外，用机器人实现书写对机械手臂灵活性和精准性是非常高的，根据这一特性，若机械臂的书写性能在非常优秀的情况下，是可以被投入到需要机械臂进行操作的医疗中去发挥作用的。如外科手术需要将手术用具控制在非常狭窄的范围内精确地执行轨迹。与本课题的在线指定机械臂运行轨迹有相似之处，因此本课题也为该种机器人的研究提供了一种有趣的思路。

并且本课题的方案，即通过机械臂收到一系列坐标控制机械臂写字，若加以结合智能学习算法，可以实现机械臂模仿个性化的笔迹功能。

综上论述，我们可以得到该课题的应用范围是非常广泛的。该系统不但可以单独工作实现简单的在线书写功能，若结合现代的智能算法，便可以实现自由书写和字迹模仿功能；同时也可以通过替换其末端执行器来实现其他功能。

## 1.2 研究现状

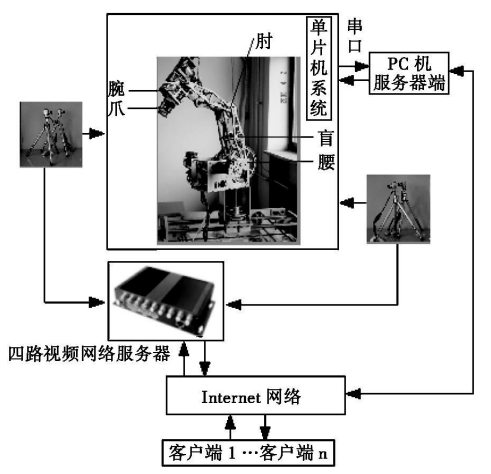
在机械臂的远程控制领域，目前不乏研究。以下内容是对部分现有研究成果的分析。有学者用摄像头进识别目标位置，通过CH340串口通信模块实现开发板与PC机通信。用C#编写函数，建立基于STM32开发的六自由度机械臂的D-H模型，并对机械臂进行正逆运动学求解，在加上实时采集到机械臂的各关节角度，可以最终实现控制机械臂到达指定位置。以此实现机器人的网络化控制[1]。除了通过该类通信模块实现外，也有学者通过两台PC机构建机械臂的远程控制系统，该系统是基于Internet-PC和Internet-DSP两种机器人远程控制系统[2]，通过两台PC机分别作为服务器端和客户端进行通信，实现远程控制。文中选取了简单的二自由度串联机械臂系统进行研究，该机械臂可以进行简单的二维图像绘制，其两个旋转关节由交流伺服电机控制[2]，并装有光电编码器进行位置控制。在基于internet-PC的系统中，以服务器端PC机作为上位机，机械臂控制箱为下位机控制机械臂，其通过计算机对机械臂进行正逆运动学解算以及姿态读取来进行控制；在Internet-DSP的系统中，将DSP控制器嵌在服务器端PC机的插槽中，PC机将轨迹点序列送入DSP，由DSP进行解算和控制[2]。相比于此类冗杂的系统，王大勇构建了基于TCP/IP网络的七自由度机械臂远程控制系统，该系统利用高端ARV单片机与PC机实现了七自由度机械臂本地端控制[3]，构建了一个融串口类对象和WinSockets类对象于一体的基于C/S ( Client/Server)结构的远程控制体系，并可通过视频网络服务器反馈视频信息，较好地实现了七自由度机械臂的远程控制[3]。

图 1-1基于TCP/IP的七自由度机械臂远程控制系统

随着应用的逐渐广泛，写字机器人的研究也开始变得热门起来。西安工程大学的姚丽娜，苏雪平研究了一种远程控制的写字机器人系统，以STM32F103ZET6为主控制器开发控制系统，通过TFT触摸屏实现人机交互和读取字迹。通过433M射频模块HC-12实现机械臂与主控板的无线通信。其机械臂采用的是越疆科技的Dobot Magician轻量级智能机械臂，可以直接向机械臂输入坐标，经过机械臂自解算自控制，减轻了主控板负担。但是上述方案采用TFT屏幕限制了分辨率，其触点坐标共320\*240，且屏幕大小有限；除此以外，其移植性和便携性也较差，只能通过该套单片机设备来实现书写控制机械臂[4]。

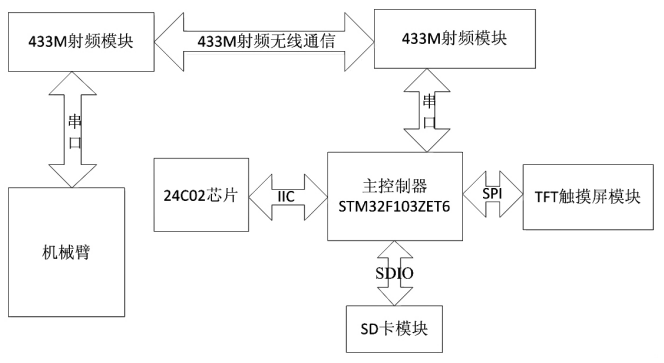
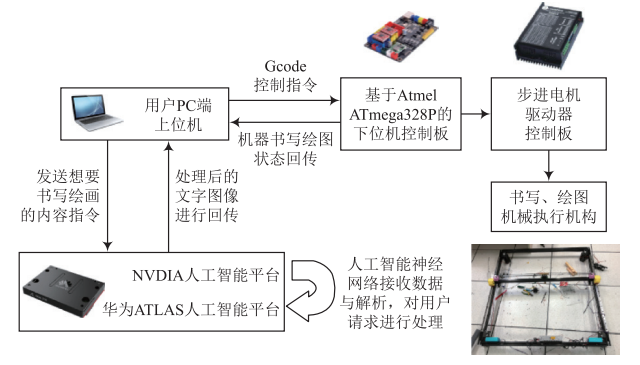
 吉林大学的赵祯祥，吕文博等学者提出一种智能自动绘图写字机器人，以人工智能神经网络结构自主生成可调节的图文，并由PC机将数据图文数据转换为机器人的动作指令。其机械臂为采用COREXY运动结构的平面关节型机械臂，采用同步带和步进电机驱动机械臂。主控板部分，该机器人采用Atmel ATmega328P为主控芯片搭建控制电路，以PC机为上位机，可以实现将图文转换为动作指令，再结合人工智能与深度学习可以实现模仿和自动创作。该设备所采用的平面关节机器人不但面积受到场地限制，而且其工作平面有限，只能在水平面书写，可开发性较低，应用范围也相应受限制；并且采用COREXY结构虽然提高了稳定性，但是提高了结构复杂度[5]。

图1-3智能仿人写字的COREXY机器人

图1-2基于远程控制的写字机器人结构

王亚光开发了一款仿人写字机器人控制系统，在该系统中采用SolidWorks设计仿人写字机器人的手臂机械结构，并构建了仿人写字机器人机械平台。各个关节的控制模块采用以STM32F103C8T6为核心控制器，选择TB6112FNG直流电机驱动模块，每个模块设计CAN总线通信，实现各控制模块和上位机的通信以协调机器人各关节的运动[6]。上位机界面软件采用QT5开发[6]，采用串口转CAN通信实现上下位机数据的通信，直接控制机械臂移动协调各关节的控制模块以实现机器人写字的基本过程。

图1-4仿人写字机器人

在白昊天，李萍开发的多自由度书写机械臂控制系统中，利用基于51单片机的舵机控制器作为下位机。在实验前提前录入每个末端位置对应的机械臂的角度，实验时将文字数据库中存储的对应文字坐标通过串口通信传递给下位机控制器，下位机控制舵机转动相应的角度，实现了汉字的书写的六自由度汉字书写机器人[7]。该种控制方法的精确度与标定点的分度有关。

综上所述，对于写字机器人的研究，从结构上来看，以曲关节型机器人为主流，并且工业机器人也以可以实现灵活动作的曲关节型机器人为主；直角坐标关节的机器人体积受限制，且也难以应用和移植到其他领域。从通信角度来看，基于CAN、串口或是蓝牙的通信方式居多，多采用特定的下位机设备与通过坐标解算姿态并向机械臂发出指令的PC机通信，也有少数研究基于TCP/IP协议建立通信实现数据交换。

## 1.3 课题研究的主要内容

为了实现上述的功能，本课题需要研究的内容可以分为三个部分。第一个部分为网页端的画板，包括在画板中如何显示出用户书写轨迹，将用户轨迹坐标存储，轨迹撤销反撤销、放缩等必要功能；第二个部分为如何利用TCP/IP通信原理实现移动设备对网页的远程访问并将轨迹坐标数据从移动端发送给机械臂的上位机；第三个部分为选取机械臂的上位机，即决定机械臂的控制方式，需要结合网页端和学习成本进行选择，使其能够使用由移动设备端发送来的坐标数据驱动机械臂移动执行轨迹模仿；同时还需考虑到本设计所采用的UR5机械臂能否与平台联动。

## 1.4 设计方案简介

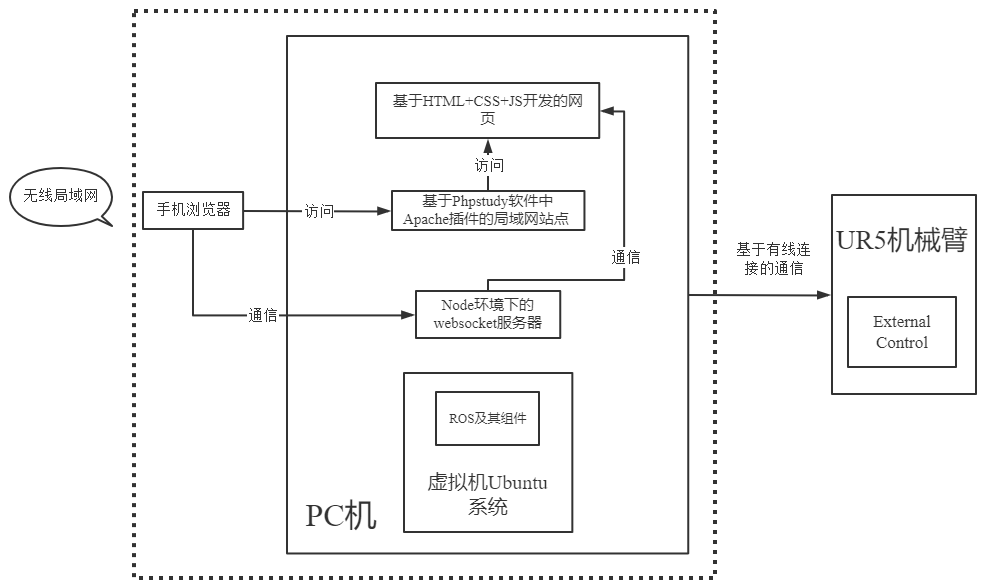
 本课题的基本技术路线如下：网页端由HTML、CSS、JS结合实现所需功能和页面，由HTML结合CSS可以高效率的设计网页布局和样式，核心部分则为JavaScript，通过JS编写函数实现通信、网页通知、按键函数以及鼠标函数触发绘画等必须功能；移动设备的访问通过phpstudy\_pro软件中的Apache插件实现，通过该插件可以建立本地局域网网页，使得移动设备可以通过局域网访问上述网页；坐标数据的传输由同一目录下运行的服务器实现，该服务器由Nodejs编写，由Node运行，可以将坐标存储在指定的系统文件夹中，在系统正常工作期间，该服务器需要处于运行状态；机械臂采用UR5机械臂，上位机由主机中运行的虚拟机担任；机械臂的仿真由ROS中的Gazebo和Moveit实现；机械臂的控制由虚拟机中运行的C++脚本实现，该脚本基于Moveit自带的默认库函数以及move group C++接口编写，可以下实现以下功能，实时读取PC机中的坐标文件夹，并对机械臂的轨迹进行预处理、规划以及后处理，在根据得到的轨迹实时驱动机械臂进行仿写。

图1-5技术路线图

# 第二章 绘画网页设计

## 2.1 HTML、CSS、JS、Apache及其功能介绍

在本设计中，为了满足课题要求，实现目的，让移动设备能够方便、快捷的访问客户端，实现功能，一种方法是采用APP，使用APP无疑能够实现更丰富的功能和更高端的用户体验，但是考虑到APP开发所需学习成本和开发成本偏高，并且对于客户端来说也不是一个很方便的访问方式，需要下载APP并且受到各个版本、系统的兼容性限制；第二种方法就是使用网站——即本课题采用的方法，通过开发网页，我们可以编写出自己的互联网站点，使得用户使用浏览器通过局域网——TCP/IP协议就可以访问客户端页面，无需下载多余APP，使用便捷，快速，同时相对于APP来说各更低的开发成本和学习成本也是一个不可忽略的优点。

对于网页端的开发工作，目前广泛应用的技术就是HTML/HTML5。作为一种超文本的标记语言，值得注意的是，它并不是一种编程语言，在更底层由其他的语言来解释该标记语言，且由HTML所编写的脚本可以用浏览器进行解释。HTML作为一种简单、易上手的标记性语言，实质上是一套标记标签，我们在编写时采用标记标签来描述网页，在整个HTML文档中包含的文本内容有两个部分：一是HTML自身所规定的标签，我们也称之为元素，这些元素由浏览器解释并在运行时表现为网页中各种各种各样的表格、段落、标题、图片、小窗口等；二是在浏览器访问页面时可以直观看见的文本内容，常常我们将文本嵌入在元素中以提高页面的可读性和上手性。

HTML虽作为一种简单易上手的语言，但是其标签系统是不乏复杂性的，每个标签包含多个属性和值，这也是为什么我们可以看见网页中各种样式的表格和窗口，以及段落中文字的样式、颜色、边框的各种变式。当我们需要对网页的样式或是某几个，当网页够复杂时甚至几十个标签的属性值进行修改时，若采用逐个修改的方式，则会大大降低我们的工作效率。

为了解决该类冗杂的工作，CSS（层叠样式表）实现了将同类标签或是隶属于同一调整范围的各种元素进行规整的管理，提高了编写效率和修改效率，使得我们可以同时控制多重网页的样式和布局。在HTML中，各种元素的显示方式决定于样式，样式默认存储于样式表中，CSS的基本原理就是将样式表提出，实现分离样式表和内容，通过外部样式表可以大大提高工作效率，而外部样式表就存储在CSS文件中，在该文件中，多个样式定义我们可以层叠为一个样式，因此我们可以通过一个简单的CSS文档就可以改变所有页面的布局和外观。

当采用HTML和CSS编写完一个网页后，我们需要对网页赋予功能。我们说到，HTML是一种标签语言，采用标签描述网页上的特定元素，CSS在此过程中帮助我们调整样式和布局，因此涉及到函数关系和行为的内容我们需要采用其他语言实现。JavaScript作为一种轻量级解释型的web编程语言，我们可以将其插入HTML页面，在插入HTML页面后，JS的脚本可以由现代的所有浏览器执行，借此来实现网页浏览者和网页的交互。值得一提的是现代所有HTML网页都使用了JavaScript，还被广泛应用于HTML、web、服务器、PC、笔记本电脑以及各种移动设备。

Apache HTTP Server（简称Apache）是Apache软件基金会的一个开放源码的网页服务器，被当前所使用的大多数计算机系统所支持，由于可移植性高，可运行平台广，并且具有很高的安全性和可靠性而被广泛应用于网页的相关开发工作，是目前非常流行的Web服务器端软件之一。其效率高，性能可靠并且能够通过一些简单的API扩展，将Python等解释器通过编译嵌入服务器中。因此借助Apache强大的功能我们可以实现搭建自己的web服务器，但在本设计中，我们需要实现的功能为创建一个局域网内可访问的网页因此为了简化步骤，我们采用PHPstrudypro软件进行配置，而不是直接用Apache，通过该软中的Apache插件我们可以轻松的实现配置局域网网页。

## 2.2 可视轨迹的画板

### 2.2.1技术背景

为了实现在移动设备网页端书写，首先要实现能够反馈用户书写。因此，本设计中采用画板的方式来记录用户的书写轨迹。HTML5标准定义了<canvas>标签在程序中定义一个画布，该标签最终由网页解释并呈现一块画布。一个Canvas标签象征着网页中的一个矩形区域，在这个矩形中，可以调用JavaScript的函数来向画板上添加各种图形。HTML5定义了一系列函数方便用户直接使用绘画，如定位路径，创建渐变对象和图形的变形。但是由于HTML标记语言的有限性，由HTML标记的画板仅仅是一个图形容器，我们需要结合JavaScript与HTML才能够实现绘画。而画板的作画函数为独立函数，使用时方法为指定坐标让其在特定位置绘画圆、矩形等图形，无法与用户进行交互。因此为了实现用鼠标或手指在画板上画画，我们需要实现配合画板的作画函数和鼠标、触摸的各种事件。通过点击、触摸事件的触发调用绘画函数，实现轨迹的可视化。同时由于点击和移动、触摸等事件在网页中的触发频率是很高的，我们需要避免误触而产生的各种问题。

### 2.2.2实现方法

为实现按下鼠标进行绘画，松开鼠标结束绘画。绘画功能通过HTML5中的ctx.draw()函数实现，该函数可以在指定两个点之间绘画直线，并指定笔画类型。绘画思路如下:

1.鼠标在绘画区被按下即为绘画开始信号，设置一个变量为true，并存储坐标。

2.当鼠标在绘画区移动，并变量为true，则在上一个鼠标坐标与此时鼠标坐标之间进行绘画直线，即调用ctx.draw()

3.当鼠标进入按键区，按下时不作为绘画信号，但松开作为绘画结束标志

4.当鼠标在按下时移动到按键区，则绘画坐标为画板边缘

根据上述思路，结合HTML5的事件和函数，可以整理出如下表格：

|  |  |
| --- | --- |
| **事件** | **事件触发条件** |
| Mouseenter | 当鼠标指针穿过（进入）被选元素时，会发生 mouseenter 事件 |
| Mousedown | 当鼠标指针位于元素上方并按下鼠标左键时，会触发Mousedown事件函数 |
| Mouseup | 当在元素上放松鼠标按钮时，会发生Mouseup事件 |
| Mousemove | 当鼠标指针在指定的元素中移动时，就会发生Mousemove事件。并且每移动一格像素就会触发该事件，处理所有Mousemove事件会耗费系统资源 |

|  |  |  |
| --- | --- | --- |
| **鼠标事件** | **触发函数** | **函数功能** |
| Mousemove | Draw() | 根据isdrawing和xx，yy的值判断是否调用ctx.draw()函数，该函数为在画板上直接绘画轨迹的函数 |
| Mousedown | Startdraw() | 若鼠标不在按键操作区，则使得变量Isdrawing=true，否则return |
| Mouseup | Stopdraw() | 根据变量xx，yy的值决定是否改变isdrawing的值，即判断此次鼠标松开是否是绘画结束的标志 |
| Mousedown  Mouseenter | Setposition() | 1.将鼠标相对画板的坐标存入变量xx，yy中（用于确定是否开始绘画）  2.将鼠标相对网页窗口左上角（随滚动条变化）的坐标存入posx、posy变量中（用于绘画函数） |

表2-1.鼠标事件

表2-2.事件触发函数

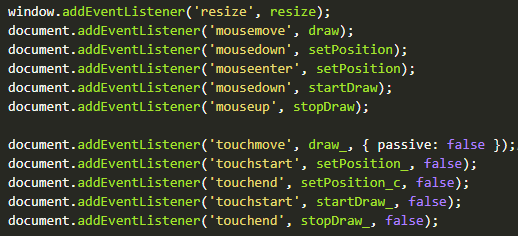
 根据上述表格，结合JavaScript和HTML，可以通过如图程序绑定事件与函数，实现鼠标或触屏触发对应函数实现功能

图2-1.JavaScript事件函数程序图

## 2.3 撤销、反撤销与清空

### 2.3.1方案阐述

为了提高用户体验以及调试方便性，在网页中添加了绘画轨迹的撤销与反撤销。由于在绘画轨迹时经常会出现手抖或是其他一些不可抗力因素导致绘画出现轨迹偏离理想轨迹，因此为了满足用户的基本使用体验，绘画轨迹的撤销与反撤销时必不可少的因素。

在本设计中，我们前面提到，通过采集用户在画板上鼠标\触摸的轨迹用draw函数中的ctx.draw函数进行每两个点之间的直线绘画。由于整个网页的输出不只是画板上的轨迹，在会话结束之后，我们需要将画板轨迹对应的坐标存储到文件中供机械臂使用。因此，在撤销与反撤销函数中，我们不仅需要保存用户绘画的轨迹图像，还需要保存轨迹的坐标。为了方便管理和撤销，我们还需要按照笔画来分类每一笔绘画所对应的图像和坐标，在撤销与反撤销时，便可以以笔画为单位，将用户的轨迹进行撤销与反撤销。

为了实现上述功能，在实现绘画的画板各函数中，就需要添加记录笔画，存储绘画轨迹图像，绘画轨迹坐标的函数及变量。为了实现删除或者再现用户绘画的轨迹图像，可以选择以下两种方式之一：

1.根据笔画数记录每一步的所有点坐标，在撤销与反撤销时，则可以根据笔画数对应的坐标进行再次绘画轨迹。即撤销时则需要执行清空画板，减小笔画数n，并将对应笔画数x小于笔画数n-1的所有坐标调用绘画函数再一次进行绘画，实现将0到（n-1）笔画的轨迹再现，即撤销第n笔对应的绘画轨迹，再通过将第n笔笔画对应的坐标移动到另一变量中便可以实现坐标撤销，反撤销同理。

2.同样根据笔画数记录坐标，同时通过调用canvas.todataURL()函数，实现以图像格式记录每一个笔画数对应的总绘画轨迹。则在撤销与反撤销时，只需要将笔画数对应的总绘画轨迹调用在画板上即可以实现轨迹的撤销反撤销，同样移动被撤销笔画的坐标可以实现坐标的撤销反撤销。

### 2.3.2实现方法

通过实验比较两种方法后发现，方法一需要对轨迹进行二次绘画，降低了撤销与反撤销的效率，对用户来说无疑是极差的用户体验，因此本课题采用了方法二进行绘画轨迹的撤销与反撤销。在实际函数中通过如下变量的变化实现功能：

1.Step：该变量的值代表着当前的笔画数，为整型变量。每一次绘画开始或者反撤销该变量都会加一；每一次撤销该变量都会减一。

2.PositionPensizeTime：该变量按顺序存储了每个单独绘画点的坐标，放缩比例以及绘画的时间点，并以空格分离各个变量，以换行符分隔各点，即每一行代表每一个点的数据。在Mouseup触发了startdraw()并被判断为绘画开始的标志时，会额外增加一行全零数据“0 0 0 0\n”，以此作为每一笔笔画的分隔标志，不作为轨迹绘画函数参考坐标；在每一次Mousemove触发draw()函数并将该次鼠标移动判断为绘画行为时会记录该时间点的各项数据。值得注意的是，即使Mousemove的触发阈值为一个像素，但是由于在执行对应的事件函数draw()时需要耗费时间，因此轨迹绘画速度稍快一些便会出现该变量中存储的坐标之间相隔几个像素，经过实验发现正常的绘画速度并不影响记录和再现绘画轨迹。

3.strHistory[]：该变量为数组对象，以数组形式存储变量PositionPensizeTime。在每一次绘画结束时会将该次绘画对应的PositionPensizeTime用push()函数存入数组中。值得注意的是，由于PositionPensizeTime并没有在每一笔绘画结束时清空，因此该数组的每一行对应的PositionPensizeTime为该笔画之前所有笔画的轨迹数据。在每次撤销和反撤销时，将PositionPensizeTime赋值为当前笔画数对应的行数据即可方便实现功能。

4.canvasHistory[]：该变量为数组对象，以数组形式存储了每一笔绘画结束时当前画板轨迹的图像。对象canvas.toDataURL()可以将当前画板上的轨迹以image/png形式存储，每次绘画结束时通过push该对象则可以在该数组的每一列存储每一次绘画结束对应的画板图像。并在每次进行撤销反撤销时对画板清空后将该数组中某一行数据作为源在画板上加载对应的图像。通过该方法可以不用调用绘画函数进行二次绘画就可以实现绘画轨迹的撤销反撤销，增加了撤销反撤销的效率并且使得程序更加简单可靠。

除此以外，网页还增加了清空功能，在画板上方放置了一个清空按钮，当用户点击清空，该按钮触发的函数就会将画板清空并重置上述四个变量，将整个绘画系统设置为初始状态。

## 2.4 记录并输出坐标

在实现了可视化轨迹、轨迹放缩、轨迹撤销与反撤销的功能后，为了与后续机械臂的工作配合，还需要将记录的绘画轨迹坐标输出，供机械臂进行轨迹跟踪使用。

前文提到，每一次被判断为绘画行为的鼠标移动行为都会导致PositionPensizeTime的改变，通过该变量存储了需要重现轨迹的所有点的数据，包括坐标和放缩比例。在网页端的画板上方设置了一个发送按钮，当用户点击该按钮时，则会将该变量通过第三章的服务器发送到主机端进行存储；除此以外，为了保证下次发送数据的可靠和准确，需要对用到的四个变量PositionPensizeTime、strHistory[]、canvasHistory[]、step进行清空，以保证在后续发送坐标时，不会出现对已发送坐标的二次发送现象。

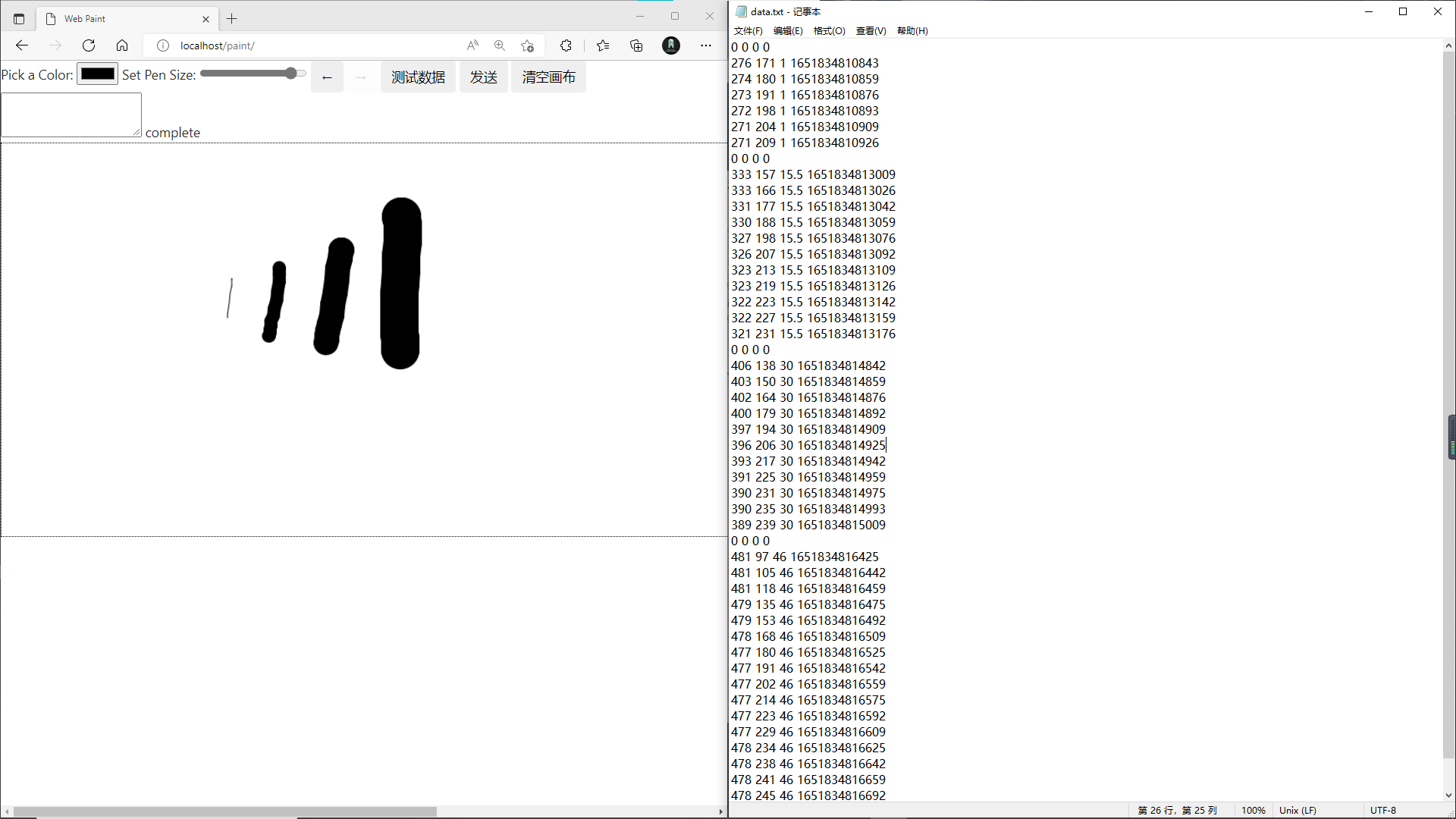
 按照上述思路，可以完成网页编写，下图为运行情况。

图2-2.网页输出坐标信息

图中左边为在画板上绘制不同粗细的直线效果图。可以看到网页上除了画板还安排了按键操控区，可以实现调节粗细，颜色，撤销、反撤销等功能，界面十分简洁友好，对于用户来说是非常方便的。右边的是储存在data.txt文件中的坐标信息，包含了笔的粗细程度和绘画时间；其中，笔的粗细信息会在后面用于轨迹的放缩功能。

## 2.5 局域网访问网页

### 2.5.1：局域站点创建

在网页可以正常运行后，需要保证在局域网内的移动设备能够正常的访问并使用网页。在建立局域网网页时，考虑到学习成本和开发成本，本课题选择了使用Phpstudy软件进行配置，该软件作为国内12年老牌公益软件，集安全，高效，功能与一体。支持100多项服务器管理功能，通过该集成环境的Apache插件可以轻松的实现配置局域网网页，实现移动设备访问上述网页。在使用时，先将编写好的网页放在软件的WWW文件夹内，随后运行软件的apache插件，此时便创建好了局域站点。

### 2.5.2局域站点访问

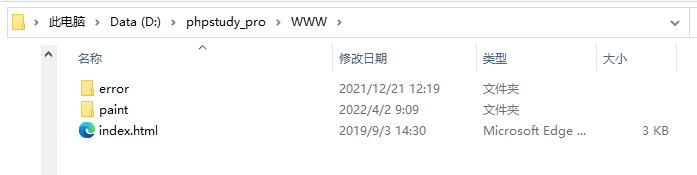
 在访问时，可以选择性访问文件夹WWW内的所有网页，如下图，本设计将绘画网页放在了paint文件夹中，此时根据IP地址访问即可。当访问的设备与主机位于同一局域网络时，打开浏览器，在浏览器端输入对应网址，主机IP地址/paint，即可访问paint中的绘画网页。

图2-3.Phpstudy局域网站点文件夹

# 第三章 服务器端设计

## 3.1 NodeJs、Socekt通信及功能介绍

Node.js 是一个开源和跨平台的 JavaScript 运行环境，它使得JavaScript可以用于开发后端程序，实现其他后端语言得几乎所有功能，因此被广泛用于各种项目。由于Node.js 在浏览器之外运行 V8 JavaScript 引擎（Google Chrome 的内核），其开发服务器的性能是很好的。并且由于V8 JavaScript引擎全新的编译方式，使得用高级语言JavaScript编写的脚本同用底层语言C编写的脚本运行效率相近。因为Node.js，数百万为浏览器编写JavaScript的前端开发者现在无需学习完全不同的语言，就可以编写除客户端代码之外的服务器端代码，大大降低了开发成本。具有学习成本低，开发效率高的优势。就像学习C语言最常用的例子就是控制台输出“hello world”，在Nodejs的教程中，也存在其独特的“hello world”方式，最常见的就是web服务器。

Websocket作为一种常用于实时通信的网络通信协议，建立在TCP/IP协议之上，协议标识符为WS，与HTTP协议类似。在HTTP通信的过程中，只能由客户端发起请求，服务端根据请求url（服务器网址）和传过去的参数返回对应结果，而在Websocket中，通信是双向的，在Websocket连接建立起来之后，客户端和服务器端可以主动向对方发送数据，既可以发送文本，也可以发送二进制数据，因此被用于聊天室这类的场景中。

## 3.2 服务器编写和运行

基于Nodejs开发web服务器的便捷性，采用该方法开发无疑是学习成本最低的方式。

为了实现通信，需要首先创建websocket通信的服务器端。在本课题中，需要在一个Nodejs文件中编写服务器代码，通过创建ws服务器对象，定义其端口及函数实现所需功能。在通过createServer创建服务器后，为“text”、“close”、“error”事件配置相应的回调函数既可以实现功能。

而在网页端，则需要在JavaScript脚本中创建ws对象并指定url和协议。当用户点击发送触发发送函数时，则运行send函数将坐标数据发出，触发服务器端的text回调函数，实现所需功能。

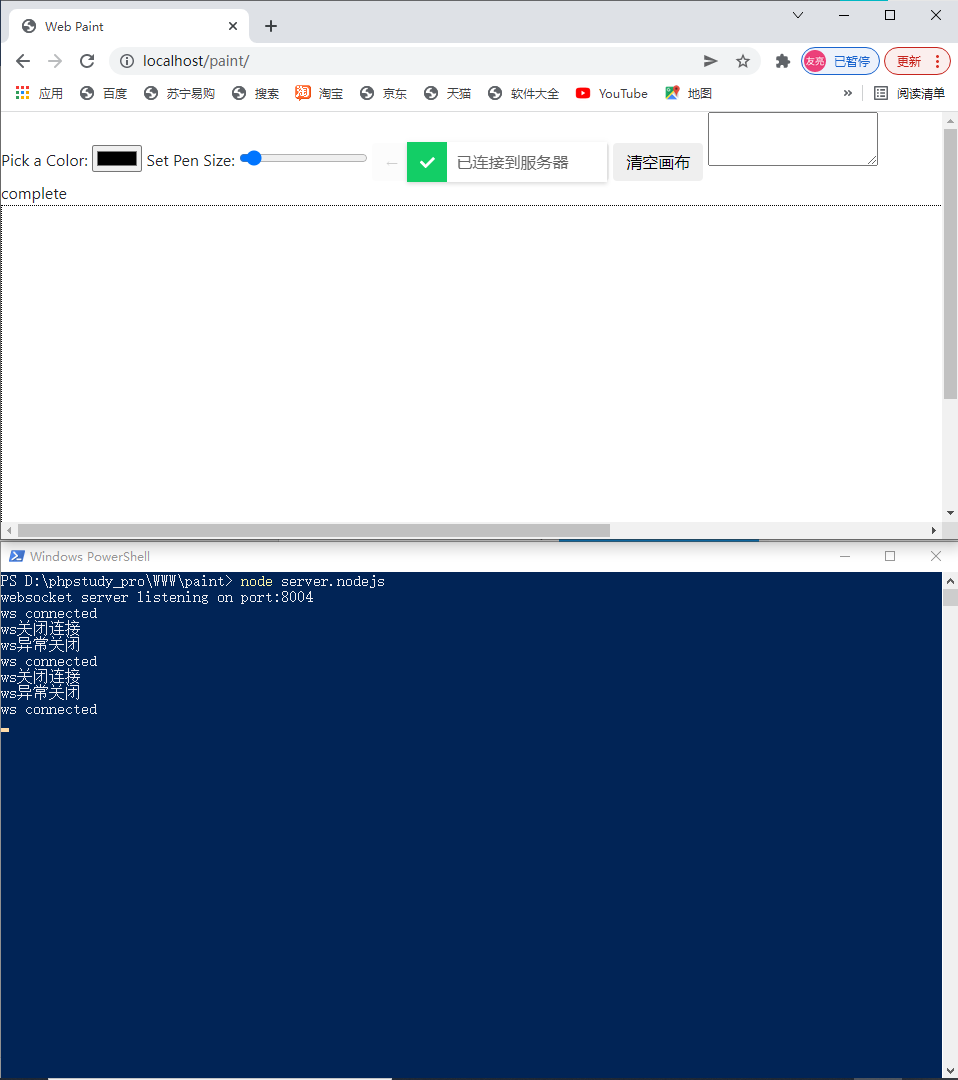
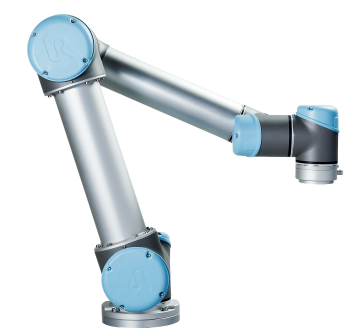
 为了两端的正常通信，需要安装Nodejs并配置websocket相关插件。在运行时，需要在主机端服务器目录下运行powershell，并在该powershell中用Nodejs环境运行服务器。

图3-1.服务器正常运行图

# 第四章 机械臂

图4-1 UR机械臂

## 4.1 UR机械臂

本实验所采用的机械臂为优傲公司开发的UR5机械臂。该公司开发的UR系列机械臂是一种安全、灵活的协作式六轴机械臂，可以通过编程来移动工具并使用电信号与其他的机器设备实现同通信。并且，该公司还配套已获取专利的PolyScope编程界面。PolyScope提供了一种便捷的编程方式，无需涉及运动规划，通过指定特定的轨迹点坐标或是角度实现驱动。通过该编程界面用户可以轻松的实现对机器人进行编程，使机械臂特定关节旋转指定角度，或是使机械臂工具端沿固定的轨迹运动。

## 4.2 ROS系统简介

### 4.2.1ROS系统概述

前面提到，PolyScope编程界面适用于简单的编程，而在本项目中，由于坐标为实时指定的，若想通过PolyScope配合网页端的坐标实现功能，则需要每一次都进行手动将坐标输入PolyScope的编程步骤中，并通过其指令驱动机械臂重复进行每两个点之间的移动。该方法不仅效率低，而且实现效果也差强人意。为了能够实现更加便捷的控制机械臂，需要选择一个合适的易上手的机械臂仿真平台。因此，在本方案中，采取用ROS-Robot Operate System来对机器人进行控制。

ROS简单来说是一款面向机器人的基于Linux的开源次级操作系统，它拥有许多像windows、Linux等操作系统所包含的功能，提供了操作系统应有的服务，包括硬件的抽象，底层设备控制，常用函数的实现，进程间消息传递，以及工作包的管理，因而被广泛应用于各种机器人开发工作。同时，该系统也提供了获取、编译、编写以及跨计算机运行代码所需的工具和库函数。

### 4.2.2ROS的优点

在考虑了多种方法后，本课题选择了ROS作为仿真平台，优点如下：

1.高度集成。ROS平台与多种工业机器人平台之间可以相互通信使用，如在本课题中，可以通过一个插件实现ROS仿真机械臂与UR机械臂的同步驱动；ROS中集成了一系列的工具包和库，集成了多种路径规划、计算机视觉、导航驱动、机器语音算法等，方便初学者研究调查使用；ROS中的开发工具又包括一些原生开发工具，如RVIZ，QT工具箱等，还有编译和测试工具以及一些第三方工具；通过ROS丰富的应用功能，我们可以实现底层硬件驱动，涉及很多传感器和执行器，对于机器人的开发是非常方便的。

2.编程语言的灵活性。ROS之所以被广泛应用于各种机器人的开发工作，其平台所支持多个语言是功不可没的，目前多种常用的编程语言如C++、C#、python等已经在ROS中编译并得到了广泛应用，对于初次开发机器人的初学者来说可以灵活选择编程语言，大大降低了用ROS开发机器人所需要的学习成本。

3.点对点设计模式——ROS的核心。ROS的一大核心特色就是点对点设计模式，很明显的优点就是可以减轻计算机的计算压力，各个节点之间通过TCP/IP协议实现松耦合连接，可以执行若干种类的通信；基于服务管理器的分布式运算特点也使得ROS系统可以高效的解决多进程运算问题，为部分特殊机器人提供了重要的技术。

4.学习成本。学习一个新的仿真软件是需要极多的精力和时间的，因此在注重课题应用的前提下，选择一个易上手的平台是首要的。ROS在该方面有着得天独厚的优势，ROS不但有体验良好的用户交互界面，还有强大的生态系统供学习和分享。

### 4.2.3ROS的框架结构组成

ROS的框架组成可以由三个部分组成，首先是文件系统级，其次是计算图级，最后是生态系统级。

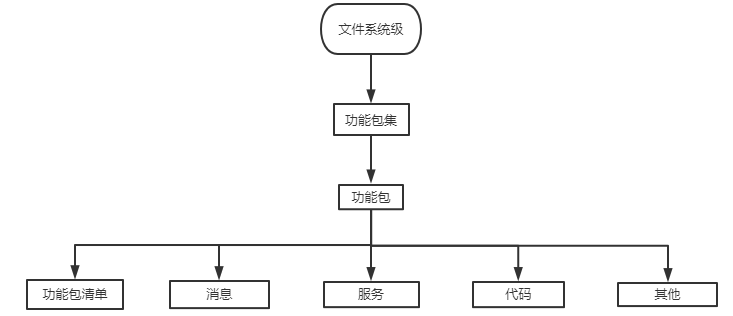
 ROS的基本工作框架为一个分布式进程框架，在系统中，常将进程表述为节点，即各种教程中的Node。正如ROS的第一层级，下图所示的文件系统级，这些Node被封装在便于被分享和发布的程序包和功能包中，并放在开源平台上，大大提高了各个用户开发的效率，降低了重复开发率。

图4-2.ROS文件系统级

上图中包含以下文件

（1）功能包。功能包在ROS中即为了单独实现某一功能的文件，其中包含节点文件、头文件、配置文件以及组织文件等等。

（2）Manifest文件。即功能包清单，为功能包的说明文件，文件中记录了功能包的依赖、编译说明等。并在整个文件系统中说明包与包之间的依赖关系。

（3）功能包集合。功能包集合中集结了多个用于实现某一个高级功能所需的所有功能包，功能包具有相应版本号。ROS的官方函数多以功能包集合的形式发布。

（4）Stack Manifest文件。即功能包集合清单，类似功能包清单，用于描述功能包集合的属性。

（5）msg文件。在ROS的系统中的通信需要实现进行消息说明，用户不但可以使用ROS官方的消息类型，也可以自己定义消息类型。该类说明常位于功能包下msg后缀的文件中。

（6）srv文件。在ROS中除了消息还有服务通信类型，因此每个进程关于服务的请求和响应数据结构由该文件阐述说明。

ROS的第二层框架即为计算图级。该层框架主要用于说明ROS中数据处理的方式。前面提到，ROS的一大特色就是点对点的模式，该模式下，ROS系统会创建一个平台，通过该平台各个节点之间得以联系通信，在平台上发布或接受其他节点的消息以实现相互消息交换。在计算图级中有以下重要概念。

（1）节点。节点是ROS实现点对点模式的重要程序。通过运行不同节点实现各自功能，再各自连接到ROS网络，节点就可以调用不同的功能包实现功能并实现相互通信。节点文件的编写可以通过C++或python语言。

（2）节点管理器。正如其名字，该管理器用于管理ROS运行过程中的各节点。在ROS系统中，管理器主要帮助节点进行注册以及查询服务，同时也为各个节点之间的直接通信提供了服务，用户可以通过指令查询当前运行的节点，还可以调用QT工具查看节点通信情况。

（3）参数服务器。与节点管理器类似，参数服务器主要用于管理在节点运行过程中各个参数的值和变化情况，当需要时，用户或者系统服务器可以通过参数服务器调用相关数据。

（4）消息。在通信时，消息由一个节点发出，另一个或一些节点接收实现通信。在通信过程中，实现目的的关键在于消息的类型和消息对应的话题。

（5）话题。话题是由ROS网络对各节点通信进行路由以及管理的关键。ROS中由节点发布的每一条消息都要对应一个话题，同一个话题可以由多个节点发布，由多个节点订阅。对各个节点的订阅和发布无强制性，节点可以发布多个话题的消息。这种消息流通形式分离了发布者和订阅者，保证了流畅性，减少了系统的运算负担。

（6）服务。不同于消息话题的通信方式。服务针对于点对点之间的直接交流，在服务过程中，由某个节点发出请求，由特定节点提供服务，并且在整个工作空间中，该服务的名字需唯一，各个节点可以通过代码对服务进行请求。

（7）消息记录包。消息记录包用于储存数据。在实际应用中，可以用记录包记录某个操作过程，随后借助其所配置的回放功能，可以实现无器材条件下反复进行实验，可以在不同环境下进行大量实验。

ROS的第三个层级，社区级，即代表ROS所提供的给用户用于交流的社区。在该社区中，可以通过借助网络环境的便利获取来自全球各地用户分享和交流的软件和系统相关知识，这正是软件的开源之处，最大化了用户的参与度，提高了开发的共享度，成功避免了重复造轮子的工作。

## 4.3 基于MoveIt/Rviz仿真和控制机械臂

### 4.3.1Moveit/Rviz简介

**1.框架结构：**

在本设计的开发中，主要应用了ROS系统中自带的Moveit和Rviz工具实现了机械臂的仿真和驱动工作。

Moveit是一个ROS系统中机器人相关的工具集软件，目前被广泛应用于各种机器人的开发工作。Moveit提供了便捷实用的平台供机器人开发，集成了各种各样的库，包括运动规划、操作、3D感知、碰撞检测、导航等等，是非常方便的机器人开发平台。Moveit以move group为核心Node，集成了各个模块为用户提供ROS行为和服务，其系统架构图如下图。

下面对Moveit的结构进行解释。

1.用户接口。在使用时，用户可以使用 move\_group 提供的一下三种方式来处理行为和服务。C++ move\_group\_interface；Python moveit\_commander；GUI（图形界面接口）：Motion Planning plugin to Rviz。

2.配置文件。作为一个ROS中的节点，move\_group需要通过ROS的参数服务器（param server）获取以下三种信息来作为工作的配置文件。URDF：robot\_description 参数，获取机器人 URDF 模型信息。SRDF：robot\_description\_semantic参数，获取机器人模型的配置信息，SRDF 是通过MoveIt Setup Assistant配置机械臂时默认生成的；MoveIt configuration：机器人的其他配置信息，如关节限位、运动学插件、运动规划插件等，该部分的配置文件均由MoveIt Setup Assistant默认配置生成，并存储在对应的配置文件夹里。

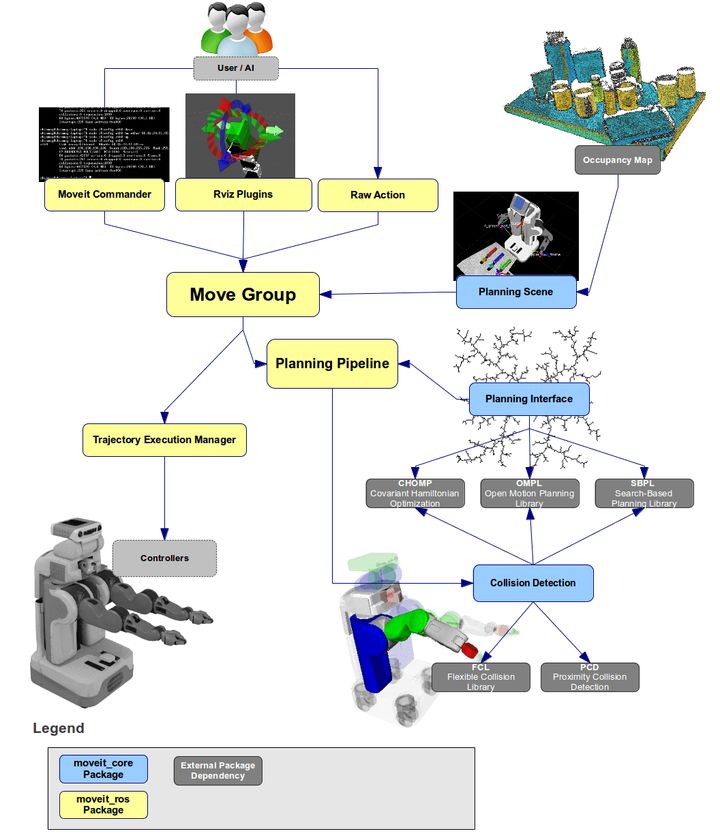
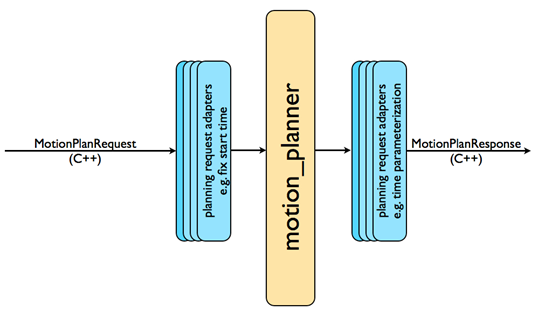
 3.机器人接口。move\_group 通过以下ROS topics和actions与机器人进行通信，可以获取机器人当前的状态（如关节位置）、获取点云或者其他的感知数据、与机器人控制器通信等。通过监听 /joint\_states话题来获取当前的关节状态信息；通过 TF 库监听坐标转换信息，使得节点可以掌握机器人的全局信息。值得注意的是， move\_group 只能监听 TF，若想要从机器人发布 TF 信息，必须在机器人上运行 robot\_state\_publisher节点；通过 FollowJointTrajectoryAction 接口——ROS中的一个行为接口，与机器人控制器进行交互，而机器人端则需要运行一个服务器来服务该行为，move\_group在通信过程中只负责实例化一个客户端与机器人控制器的动作服务器通信；使用 Planning Scene Monitor 来维护一个“规划场景”（呈现机器人状态以及周围场景），并且机器人的状态还可以包含机器人所抓持或固定的其他物件。

图4-3 Moveit框架图

**2.运动规划**

**** 在MoveIt中，运动规划的算法由运动规划器完成和执行，并且由于通过插件与运动规划器进行交互，用户可以使用多个库的不同运动规划器并保证各个规划器间的通信，每个规划器在此过程中都是一个插件，从而使得 MoveIt! 易于扩展。Move\_group默认使用的算法为OMPL算法。

运动规划的过程如上图，首先我们需要发送运动规划的请求，即The Motion Plan Request，以清楚地指明运动规划器的目标，比如将机械臂移动到某个特定姿态，当然，为了保证可行性，用户还需要根据实际情况设置一些约束，如位置、方向、关节约束以及用户定义约束等等，在该过程中会默认执行碰撞检测； 根据设定的最大速度、加速度限制，以及机械臂各个关节设定的限制等约束条件和规划请求，运动规划期会通过指定的算法计算出一条合适的运动轨迹，并回复给机器人控制器，即The Motion Plan Result；在上图中，还存在一个planning request adapters，即规划请求适配器，作为一个适配器接口，其主要功能为预处理运动规划请求和相应的数据，如应对诸如关节起始状态超出关节限制之类的场景。还可以对规划相应做后处理，如将生成的路径转化为进行时间参数化轨迹，使其满足运动规划和用户使用的需求。

图4-4 规划流程图

**3.规划场景**

Planning scene 规划场景用于表示机器人周围的世界，并存储机器人本身的状态，可以通过该功能为机械臂创建一个具体的工作环境，加入需要避开的障碍物等。其功能由move group节点内的规划场景监控器完成，该监控器通过订阅joint\_states主题获取机器人状态，通过读取来自机器人上的传感器和用户输入的信息构建世界几何模型。

**4.运动学算法**

运动学算法是机械臂各种算法的核心，尤其是反运动学算法。MoveIt的插件机制允许用户灵活的选择已有的算法，或是添加并使用自己的逆运动学算法。

**5.碰撞检测**

MoveIt中的碰撞检测是通过Planning Scene的CollisionWorld对象来配置的，主要是通过FCL（Flexible Collision Library）库来实现的。

**6.轨迹处理**

上述的运动规划器通常只生成没有时间信息的路径，而 MoveIt! 通过 trajectory processing routine 来生成 time-parameterized 的轨迹，以此满足关节速度和加速度的限制，这些限制由 joint\_limits.yaml文件规定。

### 4.3.2 机械臂的参数引入和可视化

由于UR机械臂和ROS系统在工业上或是商业上的广泛应用性，通过ROS强大的生态系统，在github上可以轻松找到UR5机械臂对应的配置文件和Moviet功能包。其中包括了机械臂的URDF、SRDF和Moveit Config文件等，还有部分便于仿真和调试的launch文件。通过已有开源的文件，可以轻松通过Moveit仿真机械臂，并调用Moveit的Rviz和QT插件进行机械臂模型、工作环境以及角度参数的可视化。

以下为上述概念做解释。

（1）URDF/SRDF文件。URDF一般使用XML语言编写的文件，用于描述机器人的模型。机械臂在仿真时是有关节、撞击、角度等概念的，URDF就是为此而存在的。在建立完机器人的三维模型之后，为了让仿真系统能够识别他的关节和角度等概念，需要用文件描述机器人的连杆，其中包含了机器人的连杆名字、连杆间的父子关系、关节以及关节对应的连杆。对于每个关节和连杆对象可以通过URDF文件定义他的各种物理特性和外观，除此以外对关节还需要定义其关节特征，如最大角度和最大控制速度、力矩等。而SRDF是Moveit根据URDF所改进的更加详细的机器人描述文件，功能同上。

（2）moveit setup assistan。在Moveit中，URDF文件是不可以直接使用的，为了让Moveit完成对机器人的运动学配置，需要借助Moveit setup assistant辅助配置，完成碰撞免检矩阵生成、虚拟关节添加、机械臂规划群组分组、初始姿态指定等配置。

（3）Moveit config文件。在机器人模型建立完毕后，为了实现仿真，Moveit还需要为机械臂配置相应的规划器、运动学解算器、控制对象组以及关节限制等其他因素。这些文件由Moveit Config文件完成指定和配置。

（4）启动文件。在配置结束后，需要启动相应的launch文件来实现仿真，该文件描述了所需所有节点和文件的位置，并指定启动的参数。其功能是为了将所有需要用到的节点集成在一起启动，而不必要去调用一个个的节点。

在完成上述文件的配置后，需要创建一个用于储存源文件、头文件和配置文件的工作空间。在主目录下创建一个名为UR\_ROBOT的文件夹，并在文件夹内创建source文件夹，将上述配置文件放入source中，并在UR\_ROBOT目录下打开终端，输入指令进行catkin编译完成创建工作空间。

在成功创建工作空间后，运行以下三个指令启动相应的launch文件，即可实现机器人的仿真-：

**Roslaunch ur\_gazebo ur5.launch limited:=true**

**Roslaunch ur5\_moveit\_config ur5\_moveit\_config\_planning\_execution.launch sim:=true**

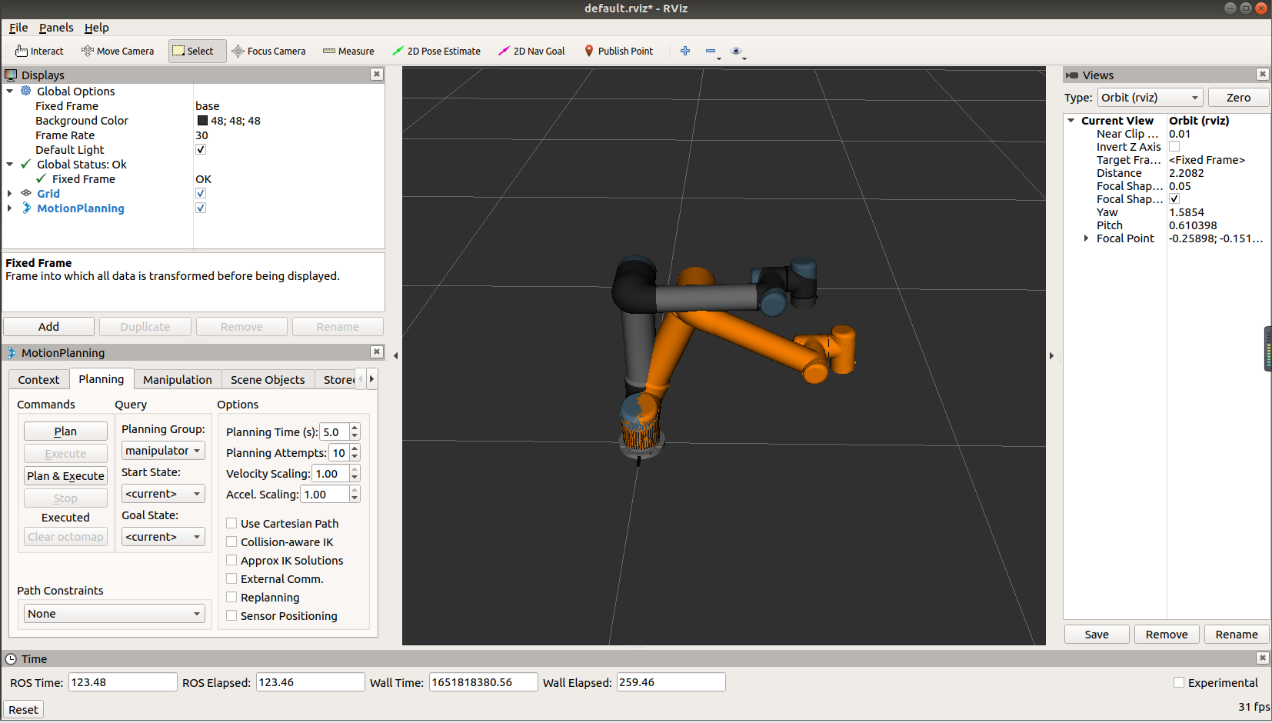
**Roslaunch ur5\_moveit\_config moveit\_rviz.launch config:=true**

图4-5 Rviz可视化工具

运行结果如上。上图为在Rviz中对机械臂进行仿真的界面，图中黄色机械臂为机械臂的目标姿态，在设置好机械臂的目标姿态后，点击左下角的plan&execute即可将机械臂移动到指定位置。除此之外，在Motion planning模块的功能中，还可以实时控制机械臂的六个关节角度，提高了调试机械臂时的效率。

### 4.3.3 坐标转化

为了实现与网页端的协同工作，需要在Moveit中实现虚拟机械臂的轨迹跟踪工作。目前，用户的绘画坐标已经存储在了虚拟机的本地文件中，因此下一步工作为读取坐标并根据坐标驱动机械臂。在本设计中采用了C++用户接口。

读取坐标的流程图如上图，由于网页端将数据存储在TXT文件中，因此在读取时首先需要将字符串型数据根据空格进行分隔，在将每一个字符串转化为double型数据，供后续使用。

在驱动机械臂时，出现以下两种方案。

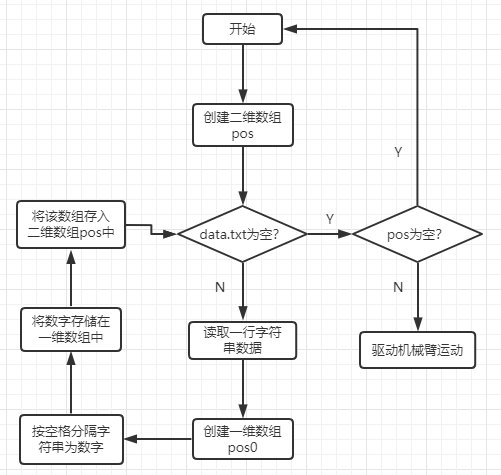
 1.通过指定机械臂末端坐标，在脚本中直接调用Moveit自带函数为机械臂进行路径规划和运动规划。在Moveit中，通过默认函数调用的末端轨迹跟踪方法如下，通过OMPL算法在各个路径点之间增加轨迹点，在生成了一系列不含速度、加速度信息的轨迹点之后，再由Moveit中的插件通过时间参数化插件对轨迹添加速度、加速度等信息，满足机械臂的运动限制。

图4-6 坐标处理流程图

2.在读取文件的坐标点集合之后，通过机器人逆运动学解算出机械臂的姿态驱动机械臂的每个关节到达指定角度。根据逆运动学解算结果，UR5机械臂对应的每个末端点会解出八个位姿。在驱动时，通过选取距离上一个位姿/当前位姿最近的位姿作为下一个点的位姿。在解算出所有点对应的位姿之后，由于各个位姿之间的角度差距过大，因此在实际驱动机械臂时，会在各个位姿之间进行插值，减小位姿之间各个关节的角度变化值。

在经历过两个方法的实验之后，发现方法一的流畅度和执行速度远大于方法二。原因如下，在方法二中，由于是通过控制机械臂的各个关节到达指定角度实现轨迹跟踪的，在此过程中机械臂则会在每两个点之间关节进行完整的加速、匀速、减速过程，整个过程表现的流畅度远远不够，因此本设计采用了方案一。

在实现方案一时，如上图所示，通过将坐标存储在二维数组中实现了按顺序储存一系列路径点。除此以外，由于网页端与ROS端对空间坐标的单位和坐标系不同，在实际调试时还需要将坐标进行转换，以保证机械臂能够在指定方位的纸上复现出同样的轨迹。

### 4.3.4 运动规划和后处理

前面提到，在整个Moveit的运动规划过程中，在发出了规划需求之后。Moveit中的palnning request adapters插件会对规划请求进行预处理，在motion planner解算出路径以后，再由适配器插件对路径进行后处理。

1.规划器产生轨迹点

Moveit通过插件形式来调用多个运动规划器插件，如OMPL，CHOMP，STOMP，其默认的运动规划器是OMPL。规划器的基本功能根据一系列目标点是提供一个连续运动轨迹，给定机械臂的关节数N后，给定一个目标位姿，设置一个工作环境，如障碍物，OMPL就会以长度为N的数组的形式提供机械臂的一系列关节位置，最终实现规划目标，不同的规划器所提供的轨迹各有不同特点。OMPL是基于采样/随机运动规划算法的开源库，此类算法的特点在于概率性，只要存在解，就一定能够找到解决方案，因而可以快速找到解决方案。值得注意的是，OMPL不包含任何碰撞检测和可视化的功能，该类规划器中规划者是抽象的，是没有机器人这样的概念的；CHOMP，用于运动规划的协变哈密顿量优化，是一种基于梯度的轨迹优化程序，其设计的是完全基于轨迹优化的运动规划算法。该程序可以使机械臂在避障时平滑自然的运动，效率非常高；STOMP，运动规划的随机轨迹优化，是一种概率优化框架。其能够在合理的时间内产生平滑良好的无碰撞路径，通过产生噪声轨迹探索初始（可能不可行）轨迹周围的空间，然后将这些轨迹组合，以此来产生具有较低成本的新轨迹，在每次迭代过程中还会进行成本函数的优化。

在使用过程中需要注意，STOMP只能用于机械臂关节空间的路径规划,不能用于空间末端的运动运动规划。通常在进行运动规划时，会通过两两规划器配合的方法来优化轨迹提高性能和平滑性，如在本设计中便采用了OMPL规划器为CHOMP规划器的规划的轨迹做预处理，通过OMPL在短时内产生一个可行的轨迹，然后CHOMP再根据该可行轨迹生成一个优化的轨迹。

2.适配器完成轨迹后处理

完成上述工作后，目前所得到的是一系列连续空间位姿的组合。若想要驱动机械臂，则还需要运动规划适配器根据规划请求时以及机械臂自身所设置的限制为各个位姿添加各自的速度和加速度。Moveit会默认遵循由文件joint\_limits.yaml文件中所规定的速度和加速度限制，并默认调用IPTP作为适配器。除了IPTP以外，Moveit还支持ISP和TOTG两种适配器。

IPTP适配器采用等间距插补，可以实现速度和加速度平滑，但是无法避免加速度抖动，适合用在低俗精确的轨迹控制下；ISP是在IPTP的基础上，为其拟合了一个合适的三次曲线进行了优化，虽然相比IPTP提高了一些性能，但是还是没能解决加速度抖动问题；TOTG输出等时间间距的轨迹，小间距轨迹下相比其余两种可以实现更加平滑的合理的运动，在0.005s下插补可以有较为稳定和理想的规划效果。因此本课题采用了TOTG作为适配器为轨迹添加时间参数化轨迹，使用时可以修改其各项参数，插补时间在ROS系统文件中的trajectory\_processin中找到对应的include文件即可以修改，调用时需要首先在文件库函数中添加trajectory\_processing库，并调用对应模块即可以实现轨迹的时间参数化。

## 4.4 基于External Control的机械臂控制

### 4.4.1网络配置

为了实现ROS中仿真机械臂与实际机械臂的联动，即通过ROS控制实际机械臂。首先需要在UR机械臂的控制器PolyScope端安装URCaps External Control，该URCaps是一个用户接口，适用于ROS，ROS2以及Isaac SDK系统，URCB3以及e系列机械臂，并且需要保证PolyScope的版本需要高于3.7。

在成功安装External Control之后，需要在机器人设置中查看机械臂的IP地址，随后用网线将机械臂与主机端进行连接。

在PC端，需要手动配置有线连接的IP地址，并保证与机械臂在同一网段内，以保证通信，如本课题中机械臂的IP地址为192.168.56.101，由于本课题的ROS运行在虚拟机中，因此需要首先将虚拟机的网络适配器设置为桥接模式，随后在虚拟机中将有线连接的IP地址设置为192.168.56.100。

### 4.4.2同步控制

同仿真工作，在实际控制机械臂时，除了需要设定网络IP地址以外，还需要启动相应的launch文件，将所需节点启动。

第一条指令，如下，需要官方指定机械臂驱动文件夹ur\_robot\_driver中的launch文件启动驱动所需的所有节点。并在启动时指定机器人的IP地址；除此以外，还需要根据calibration文件对机械臂的URDF文件进行参数矫正，以消除由于URDF文件带来的末端坐标毫米级误差。

**roslaunch ur\_robot\_driver ur5\_bringup.launch limited:=true robot\_ip:=192.168.56.101 [reverse\_port:=REVERSE\_PORT]kinematics\_config:="${HOME}/catkin\_ws/src/fmauch\_universal\_robot/ur\_description/config/ur5\_calibration.yaml"**

第二、三条指令用于启动相应的路径规划执行的节点以及Rviz仿真节点，并指定启动参数的值。指令如下

**roslaunch ur5\_moveit\_config ur5\_moveit\_planning\_execution.launch limited:=true**

**roslaunch ur5\_moveit\_config moveit\_rviz.launch config:=true**

此时在通过C++脚本或是通过拖动机械臂末端、控制各个关节角度控制仿真机械臂移动时，真实的UR5机械臂会与仿真机械臂同步运动，从而实现了机械臂的控制由仿真到实际机械臂的转化。

## 4.5 机械臂末端执笔结构设计

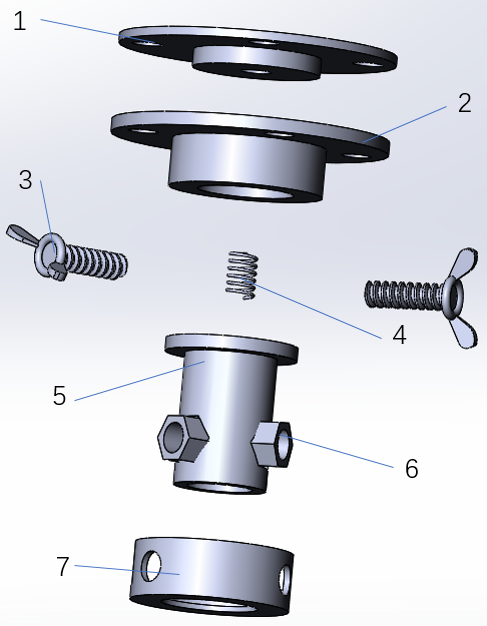
 由于设计时间有限，本设计将重点放在了网页端开发与机械臂的控制工作上，未对机械臂末端做过多设计，无法实现机器人自动执笔，放笔等功能。但是为了让机器人在纸上再现轨迹，设计了一款机械臂的手动松紧执笔结构。其机械机构如下：

图4-7 末端结构

|  |  |
| --- | --- |
| 零件序号 | 零件说明 |
| 1 | 零件1通过上有四个通孔，通过螺栓与机器人工具端法兰连接；在零件1底部有一柱形小孔，用于固定弹簧的上端 |
| 2 | 零件2上有四个通孔，与零件1对齐，同样以螺栓跟零件1一起被固定在机器人工具端法兰；零件2还为零件5限制上下移动的自由度 |
| 3 | 羊角螺栓，用于从两端将笔固定在零件5的柱形内壁上 |
| 4 | 弹簧，为了保证在书写时能够对桌面施加一定的压力的同时保护笔和机械臂不受硬冲击而损坏或降低寿命而设置 |
| 5 | 零件5的外圆柱面有两个放螺母的槽，与零件7配合，保证螺栓螺母配合稳定；零件5的内圆柱面与羊角螺栓一起配合夹持住用于书写的笔 |
| 6 | 螺母，固定在零件5与零件7之间，与羊角螺母配合，使得羊角螺母内外移动，实现夹紧和松开 |
| 7 | 零件7与零件5配合用于固定羊角螺栓和螺母 |

表格4-1 末端结构各零件介绍

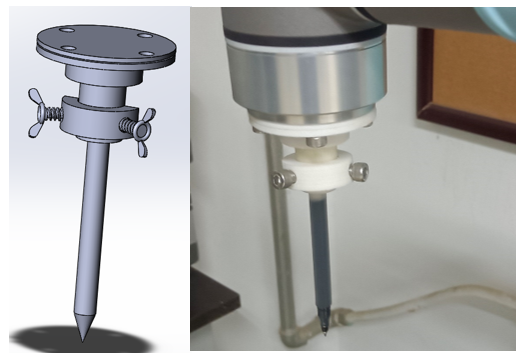
**** 在设计结束后，本课题对末端执行器采用了3D打印加工执行器，最终实验效果如下图，可以完成对正常粗细的签字笔进行夹持。

图4-8 末端结构实物图

# 第五章 实验结果

## 5.1不同运动规划算法的比较

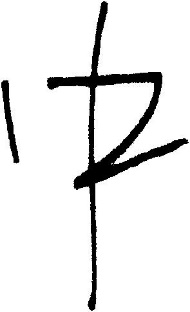
 前面提到，在ROS中存在三种运动规划算法，OMPL，CHOMP和STOMP。其中CHOMP的优势在于避障规划时能够提供平滑的轨迹，因此在本设计中——不存在避障的要求下，CHOMP会使得轨迹解算速度变慢；而STOMP算法只能用于关节空间的运动规划不能用于末端轨迹的运动规划；OMPL作为基于采样的规划库，其概率性保证了其规划的速度。在ROS中存在不同运动规划算法的结合使用，因此本设计以OMPL算法作为预处理算法，比对了在单独作用以及在分别STOMP和CHOMP算法的后处理后机械臂的性能。

图5-4 OMPL+STOMP

图5-3 OMPL+CHOMP

图5-2 OMPL

图5-1原轨迹

|  |  |  |
| --- | --- | --- |
| 算法 | 相似度 | 完成时间 |
| OMPL | 0.9994 | 10.44s |
| OMPL+CHOMP | 0.99993 | 9.88s |
| OMPL+STOMP | 0.99995 | 10.23s |

表格 5-1 各算法实验结果

上述实验结果是采用TOTG时间参数化算法条件下进行的，经过数据对比可以发现OMPL+CHOMP的速度是优于其余两个算法的。除了执行时间外，并且彼此之间相似度并无很大差异，相似度的评估方法是采用基于直方图余弦相似度的方法来评估的。因此本设计将采用OMPL+CHOMP作为运动规划算法。

## 5.2不同时间参数化算法的比较

针对前述提到的三种时间参数化算法，IPTP，ISP，TOTG，在使用OMPL+CHOMP作为运动规划算法的前提下，对三种时间参数化算法进行了实验比较，主要从轨迹执行时间、机械臂外在表现以及各关节速度的变化曲线来评估。实验结果如下：

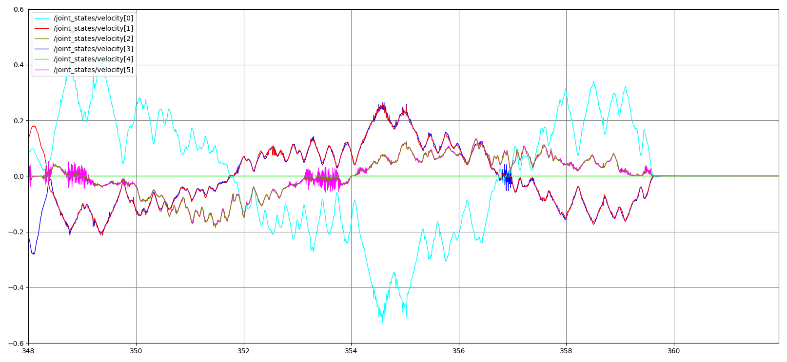
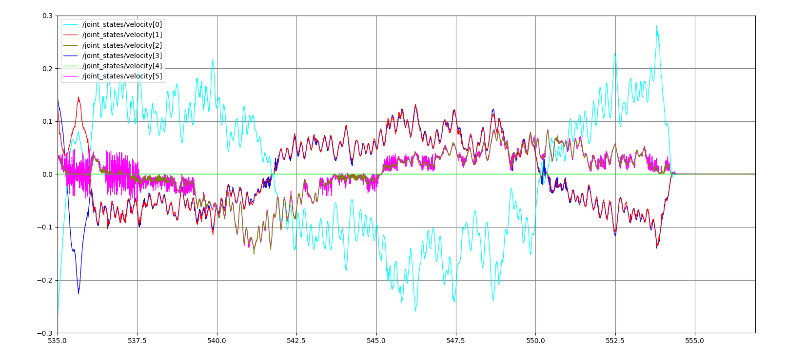
**** 以下为三种算法执行同一个轨迹时机械臂的各关节加速度曲线图，选图截取了机械臂从开始绘画到绘画结束的一段作为参考，其余时间段内机械臂的移动不作为参考。

图5-5 TOTG速度曲线

图5-6 ISP速度曲线

图5-7 IPTP速度曲线

上述结果表明，从速度变化曲线来看，TOTG算法的曲线图具有相对平滑度高、尖峰少等优点；各算法的轨迹完成时间如下，TOTG：9.88s，ISP：14.23，IPTP：11.62s；结合现场对机械臂的观察，采用TOTG算法时，机械臂的运动更加流畅，卡顿少，相对的，采用ISP和IPTP算法时，机械臂的抖动现象和卡顿现象更加频繁和剧烈。三种算法下，机械臂都存在不同程度的抖动和卡顿，但结合执行时间和速度曲线图，TOTG毫无疑问是最优选择。

## 5.3结果展示

**1.实验条件**

1.PC机内存16g，显卡GTX1050，系统win10。

2.虚拟机由软件VMWare运行，分配4GB内存，硬盘20GB，双处理器，系统Ubuntu18.04。

3.ROS系统版本Melodic。

4.机械臂型号UR5。

5.实验材料为普通记号笔和普通草稿纸。

6.手机运行浏览器为夸克浏览器。

**2.实验步骤：**

1.准备好一切设备供电，正常运行。

2.配置所需的网络环境。

3.将机械臂和用于书写的纸移动到指定位置并将机械臂的轮子锁死。

4.将笔固定在机械臂上。

5.在win10系统中运行Phpstudy软件的Apache插件，开启局域网站点访问通道。

6.在Node环境下，在浏览器文件目录下运行位于同一目录的Wbesocket服务器。

7.运行ROS系统Moveit仿真和控制实际机械臂所需的各组件。

8.运行用于实现坐标读取、运动规划、驱动机械臂的C++脚本。

9.在手机端运行浏览器并输入“IP/文件名”访问局域网网页，如本课题中访问时输入“192.168.124.1/paint”。

10在浏览器网页的画板上绘画，结束后点击发送按钮。

**3.实验结果：**

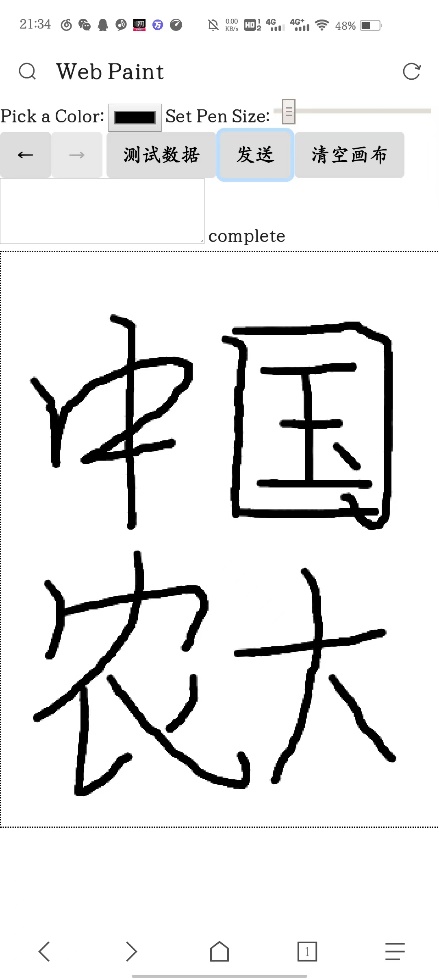


图5-8 网页绘画效果

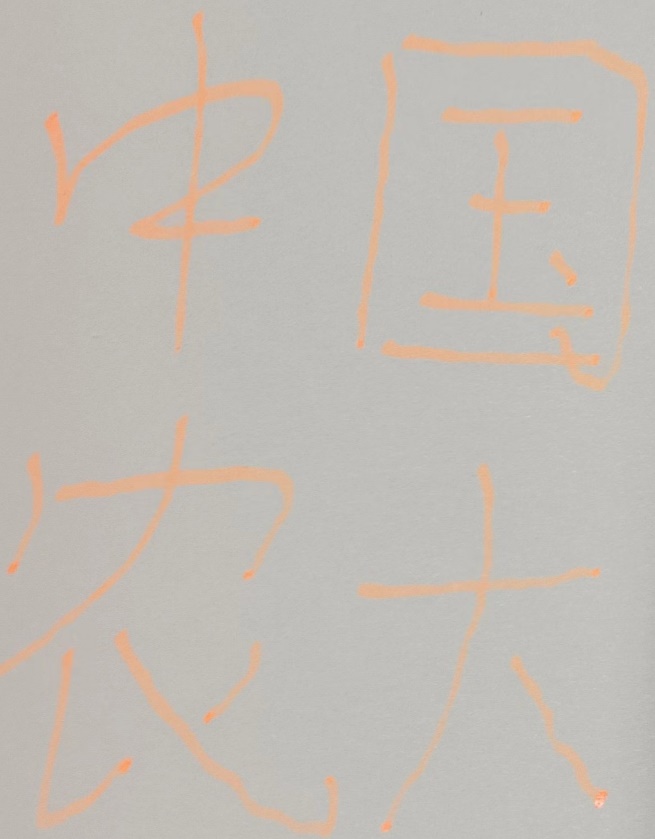


图5-9 机械臂仿写效果

**4.实验结果分析：**

 为了对实验结果进行相似度分析，对纸上的轨迹进行二值化处理，处理后的图如下：

图5-10 二值化处理结果

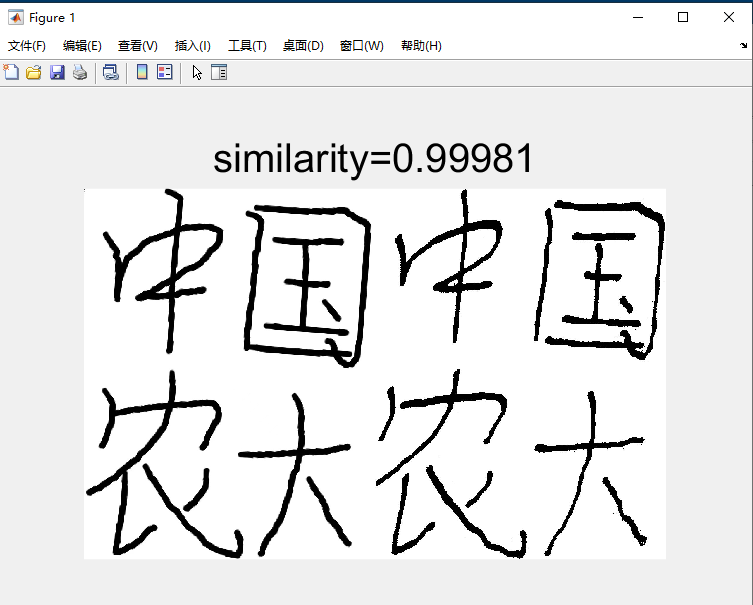
 在完成二值化后，将轨迹同原图进行相似度分析，结果如下。可见机械臂的仿写是比较成功的。

图5-11 相似度分析

# 第六章 研究成果

本课题在研究过程中获得以下研究成果

1.一套基于手机屏书记录的机械臂模仿书写系统

2.可以用于读取文件文本坐标，将文本转化为一系列数组坐标并调用Moveit组件进行运动规划和驱动机械臂的C++文件。

3.用该系统申请的发明专利。

# 总结

在工业自动化高度发展的近年来，机械臂很早就被用在各个领域中代替人工进行操作，在工业生产、医疗、航天研究以及各种研发领域扮演着强有力的角色。早期机械臂只是被用来做重复简单的工作，通过提前输入指定的程序，或用示教器在机械臂的微处理器内存储特定动作序列，让机械臂代替工人做重复简单的工作，降低工作人员的劳累程度。随着各种科学技术发展迅速，出现了许多科学技术交叉的研究成果。而机械臂因其巨大的潜能，其应用和控制方法也作为一个非常热门、激烈的领域拥有着广泛的研究成果。在网络、蓝牙等各种通信协议迅速发展的信息时代，机械臂的远程控制自然而然地作为一个热门课题而被广泛研究。本课题基于这样的背景，对机械臂进行了一种基于手机屏书写记录的远程控制方法的研究和实现，相比于示教器以及现有的一些网路控制方法，本课题的对机械臂的控制方法更加直接和便捷。

本课题中的机械臂控制主要是基于TCP/IP协议和ROS系统的，控制端的开发则是基于JavaScript和HTML。在课题开发工作中对机械臂的远程控制以及移动设备与PC设备的无线网络通信进行了一些探索和实验。实验中机械臂工作稳定，对简单轨迹和文字可以实现复现，并且从实验结果来看，机械臂的重复度是可观的。但是本课题还是有很多可以改进和深入研究的地方，主要包括：

1.目前网页端只能实现向机械臂指定二维的图画轨迹和放缩系数，导致机械臂的工作平面只能够在C++脚本中指定，在工作过程中也无法更换机械臂的工作平面。由于机械臂与手机端的网页通信是基于文本文件的，因此若对网页端发送的文本加以各种标志位，如工作平面以及工作平面坐标等，在加以对虚拟机端的C++脚本做相应的添加，则可以实现通过网页在线指定机械臂的工作平面，使机械臂可以得到更加灵活的应用和控制。

2.机械臂目前只能够对网页端的轨迹进行复现，因此很难实现逐字书写的功能。而在手机端的书写轨迹本身就会占满手机屏幕，若想要实现书写一整段文字则十分困难。若对整个系统加以改进，增加段落模式，实现机械臂每写一个字移动固定距离以及换行等功能，则可以使得系统的功能更加强大。

4.机械臂的执笔装置仍然为手动装卡，对于实际应用来说也是不便利的。若可以为机械臂增加一个可以远程控制的专用末端执行器，可以实现自动执笔、存笔、换笔的功能，则可以实现更加方便的控制。

4.机械臂的运动规划是由Moveit默认的OMPL规划器实现的，后处理是由默认的TOTG算法实现的，在调用过程中其各项参数也是默认值。导致了机械臂在仿写微小复杂的轨迹时会产生轻微的抖动和卡顿。因此若对机械臂的运动规划算法和后处理算法加以修改和优化，则可以克服机械臂在复杂轨迹仿写过程中出现的卡顿问题。

由于课题的开发时间和成本的原因，本文章对该系统的开发深度有限，希望本课题能够为未来机械臂的发展和应用贡献一份薄弱的力量

# 参考文献

[1] 叶达文，网络化机械臂跟踪系统及其平台构建，硕士，南京理工大学，2019。

[2] 周海燕，基于Internet的机械臂网络远程控制系统拓展研究，硕士，中国科学技术大学，2011。

[3] 王大勇，基于Tcp/Ip的七自由度机械臂远程控制，冶金设备，2007(04)：46-49

[4] 姚丽娜，苏雪平，基于远程控制的写字机器人系统，自动化与仪器仪表，2021(09)：240-243

[5] 赵祯祥，吕文博，安海宁，龚怡然，智能自动绘图写字机器人，物联网技术，2020，10(03)：7-8

[6] 王亚光，仿人写字机器人建模及控制系统研究与设计，硕士，大连海事大学，2019。

[7] 白昊天，李萍，多自由度书写机械臂控制系统设计，绿色科技，2019(02)：155-156

[8] 田西勇，机器人轨迹规划方法研究，硕士，北京邮电大学，2008。

[9] 李伙钦，Html5网页电子画板的设计与实现，科技视界，2012(02)：101-102

[10] 刘乃军，鲁涛，席宝，蔡莹皓，王硕，基于Ros的Ur机器人遥操作系统设计，兵工自动化，2018，37(03)：88-90

[11] 赵晓贺，基于Ros的飞行机械臂自主导航与抓取关键性技术研究，硕士，山东科技大学，2019。

[12] 朱建军，张博文，王明园，基于Ros的机械臂建模与路径规划仿真，科学技术创新，2021(19)：180-181

[13] 吉志敏，基于Ros的同位素分装机器人运动规划及控制仿真研究，硕士，兰州理工大学，2018。

[14] 刘宇程，姜振华，胡俊，基于Ros平台的六自由度机械臂轨迹规划及仿真研究，制造业自动化，2022，44(1)：38-41, 50

[15] 余明江，李辉，谭彦祺，王佳兵，池成焕，基于Stm32的智能物流分拣系统的应用与研究，仪表技术，2021(05)：52-54

[16] 陈娟，基于Tcp/Ip的局域网通信系统的设计与实现，信息与电脑(理论版)，2018(22)：179-182

[17] 梁敦毫，杨力，Html5自定义画板的设计与实现，信息与电脑(理论版)，2020，32(13)：98-100

[18] 高程希，吴成东，张云洲，商博，孟婷婷，基于Tcp/Ip协议的远程图像/数据传输研究，机电工程，2011，28(09)：1102-1105

[19] 单涵琪，基于Ur5机械臂的运动学分析及轨迹规划研究，硕士，沈阳工业大学，2020。

[20] 包文祥，基于WebSocket协议的实时网页通信的研究与实现，硕士，江苏科技大学，2018。

[21] 刘承磊，基于Web的智能家居控制器的设计与实现，硕士，山东农业大学，2009。

[22] 李双双，工业机器人建模、运动仿真与轨迹优化，硕士，内蒙古大学，2012。

[23] 陈钢，高贤渊，赵治恺，黄泽远，符颖卓，费军廷，空间机械臂智能规划与控制技术，南京航空航天大学学报，2022，54(01)：1-16

[24] 马强，六自由度机械臂轨迹规划研究，硕士，哈尔滨工程大学，2007。

[25] 臧庆凯，六自由度机械手运动学与运动规划研究，硕士，广西工学院，2012。

[26] 孙浩，六自由度苹果采摘机械臂路径规划与仿真分析，南方农机，2021，52(23)：47-50

[27] 侯晓玲，六自由度协作机器人轨迹规划研究，硕士，桂林电子科技大学，2021。

[28] 王凯，面向手术机器人的软体机械臂关键技术研究，硕士，中国科学院大学(中国科学院长春光学精密机械与物理研究所)，2021。

[29] 刘敏，嵌入式Tcp/Ip协议单片机技术在网络通信中的运用，信息记录材料，2021，22(09)：161-162

[30] 曾华琳，黄雨轩，晁飞，周昌乐，书写机器人研究综述，智能系统学报，2016，11(01)：15-26

[31] 钟彩意，王欣，吴迪，王殿龙，四臂医疗转运机器人设计及仿真平台搭建，机电工程技术，2021，50(07)：20-24

[32] 陈昱琦，网络安全之Tcp/Ip协议，科技风，2021(16)：77-78

[33] 陆晨，冯向阳，苏厚勤，HTML5 WebSocket握手协议的研究与实现，计算机应用与软件，2015，32(01)：128-131

[34] 李芳芳，孙乾，我国工业机器人发展现状的调查分析，机械传动，2019，43(06)：172-176

[35] 徐威震，孙其龙，崔金鹏，一种基于Ros系统的机械臂设计与仿真，信息与电脑(理论版)，2021，33(21)：112-114

[36] 钟蕴玉韩基业周华伟，用JavaScript实现Web页交互性，哈尔滨师范大学自然科学学报，2002(03)：42-46

[37] 刘世平，胡竹，程力，付艳，仿人机器人手臂动作模仿系统的研究与实现，机械设计与制造，2022(02)：300-304

[38] 沈洪锐，李俊杰，自主路径规划的同步机械臂的研究与实现，控制工程，2020，27(01)：188-193

[39] 曾华琳，黄雨轩，晁飞，周昌乐，书写机器人研究综述，智能系统学报，2016，11(01)：15-26

# 致谢

四年的时间，转瞬即逝。在这四年里，经历了许多的挫折与成功，无论在学习上还是生活上，或是个人能力上，都有了不小的收获。这一切都得益于我身边的同学与老师。

首先感谢我的指导老师胡标副教授，本课题是在胡老师的悉心指导和大力支持下完成的。这短短的半年里，胡老师对我课题上的指导是十分尽心尽力的，从开题到实验到申请专利再到结题，几乎每一次PPT每一次材料胡老师都认真的指导我改进，在这一次又一次的改正中，我的个人能力和经验也得到了提升。可以说本课题是倾注了胡老师大量心血的。在此我向我的导师胡标副教授表示真诚的感谢和祝福！

其次非常感谢本专业的各位老师以及各位辅导员，在这四年的学习里，若没有各位老师的悉心指导，学生不会有今天这一步。老师的每一次精心授课都让学生受益匪浅，每一次答疑老师的细心讲解也让学生保持着高昂的学习热情；除了学习外，辅导员在生活中、以及各种资助上也给予了我很多帮助，让我能够专心学习。

感谢我的家人和朋友，在我的个人成长经历中，陪伴我最多的就是我的家人和朋友。我非常感谢我的父母对我学业以及个人选择的支持和鼓励，即使相隔千里，我父母也时时刻刻关心着我的学习与我的个人生活，每一次嘘寒问暖都让我意识到背后有人支持着我；每当我进入人生低谷，变得几近颓废时，总会有我的好朋友们在一旁安慰我，鼓励我，在学习和生活上给我带来了很大的正能量；每每感到孤独时也是他们陪在我的身边，让我感受到了生活的乐趣，让我有了前进的动力。

最后，对那些所有帮助过我、跟我共同努力过的所有人表示衷心感谢！

# 附录A

## ROS安装过程

在安装Ubuntu18.04后，打开终端，依次输入以下指令：

1.添加软件源

sudo sh -c '. /etc/lsb-release && echo "deb http://mirrors.ustc.edu.cn/ros/ubuntu/ $DISTRIB\_CODENAME main" > /etc/apt/sources.list.d/ros-latest.list'

2.添加密钥

sudo apt-key adv --keyserver 'hkp://keyserver.ubuntu.com:80' --recv-key C1CF6E31E6BADE8868B172B4F42ED6FBAB17C654

3.更新软件包

sudo apt update

4.安装ROS桌面完整版

sudo apt install ros-noetic-desktop-full

5.初始化rosdep。该步骤需要访问境外网站，若频繁失败则需要对网络进行特殊配置。

sudo rosdep init

6.更新rosdep

Rosdep update

7.配置环境变量

echo "source /opt/ros/noetic/setup.bash" >> ~/.bashrc

8.安装rosinstall

sudo apt install python-rosinstall python-rosinstall-generator python-wstool build-essential

至此ROS安装完毕。

# 附录B

## ROS与机械臂相关配置

### 1.UR5相关配置文件

一、在虚拟机桌面创建文件夹ur\_robot/source

二、在文件夹source内打开终端，依次输入以下指令向工作空间源文件夹内添加机器人配置和驱动文件

1.下载驱动文件

git clone https://github.com/UniversalRobots/Universal\_Robots\_ROS\_Driver.git src/Universal\_Robots\_ROS\_Driver

2.下载驱动文件

git clone -b calibration\_devel https://github.com/fmauch/universal\_robot.git src/fmauch\_universal\_robot #-b表示下载分支

3.安装依赖文件

sudo apt update -qq # -qq表示不输出信息，错误除外

rosdep update

rosdep install --from-path src --ignore-src -y #-y 表示当出现[yes/no]的问题时默认选择yes

4.编译工作空间

catkin\_make

### 2.UR5与机械臂连接

一、用网线连接机械臂与PC机。

二、将虚拟机网络适配器设置为桥接模式。

三、设置虚拟机有线连接，手动为虚拟机指定任意可用IP地址。

四、在PolyScope端手动配置机械臂网络，将IP地址配置为与虚拟机有线连接的IP地址位于同一网段。

五、在虚拟机终端中输入运行连接机械臂的launch文件的ROS指令，并在指令后指定机械臂的IP地址，该指令同时指定了用于矫正模型与实物参数差的文件。roslaunch ur\_robot\_driver ur5\_bringup.launch limited:=true robot\_ip:=192.168.56.101 [reverse\_port:=REVERSE\_PORT]kinematics\_config:="${HOME}/catkin\_ws/src/fmauch\_universal\_robot/ur\_description/config/ur5\_calibration.yaml"

六、在虚拟机终端中输入以下指令运行用于轨迹规划、执行和Rviz可视化界面。

roslaunch ur5\_moveit\_config ur5\_moveit\_planning\_execution.launch limited:=true

roslaunch ur5\_moveit\_config moveit\_rviz.launch config:=true

# 作者简介

基本介绍（姓名、性别、出生日期、籍贯等）：

陈友亮，性别男，2000年1月8日出生于贵州省六盘水市。

教育经历：

2006-2012于贵州省六盘水市钟山区第十小学读小学。

2012-2015于贵州省六盘水市第四中学读初中。

2015-2018于贵州苏六盘水市第四中学读高中。

本科期间发表的学术论文：无

本科期间主持/参与的科研项目：

国家级大学生创新创业项目：“基于视觉处理的智能行道树石灰水喷涂车”

中国农业大学校级URP项目：基于电磁导航智能车控制系统设计

本科期间获得的奖励和荣誉：

2018-2019学年、2020-2021学年获中国农业大学学习优秀二等奖学金，国家励志奖学金，曾宪梓奖学金

2019-2020学年获中国农业大学学习优秀一等奖学金、儒省奖学金、曾宪梓奖学金

2018-2019获中国农业大学优秀团员称号，2020-2021获中国农业大学三好学生称号

其他成果：

2021年1月获北京市大学生工程设计表达竞赛二维个人一等奖