

硕士学位论文

开放式并联机器人控制平台  
关键技术研究

**RESEARCH ON THE KEY TECHNOLOGY FOR  
THE OPEN CONTROL PLATFORM OF  
PARALLEL ROBOTS**

张运强

哈尔滨工业大学

2012 年 12 月

国内图书分类号：TP242.2  
国际图书分类号：621

学校代码：10213  
密级：公开

## 工学硕士学位论文

# 开放式并联机器人控制平台 关键技术研究

硕 士 研 究 生：张运强

导 师：黄瑞宁 助理教授

申 请 学 位：工学硕士

学 科：控制科学与工程

所 在 单 位：深圳研究生院

答 辩 日 期：2012 年 12 月

授予学位单位：哈尔滨工业大学

Classified Index: TP242.2

U.D.C: 621

Dissertation for the Master Degree in Engineering

**RESEARCH ON THE KEY TECHNOLOGY FOR  
THE OPEN CONTROL PLATFORM OF  
PARALLEL ROBOTS**

<b>Candidate:</b>	Yunqiang Zhang
<b>Supervisor:</b>	Assistant Prof. Ruining Huang
<b>Academic Degree Applied for:</b>	Master of Engineering
<b>Speciality:</b>	Control Science and Engineering
<b>Affiliation:</b>	Shenzhen Graduate School
<b>Date of Defence:</b>	December, 2012
<b>Degree-Conferring-Institution:</b>	Harbin Institute of Technology

## 摘 要

工业机器人是自动化制造生产线中必不可少的设备，能代替人完成重复长时间的或是危险、恶劣环境下的作业，极大的提高了生产效率。为了让机器人能够完成更复杂的任务，提高机器人智能，许多传感器，如触觉、力觉或视觉，先进的控制算法，如模糊控制、鲁棒控制，需要整合到工业机器人中，使其具有识别功能或更进一步的自适应、自学习功能。但传统的商用机器人控制平台是不开放的，对平台的改造，需要开发商的协助，使用开发商指定的设备，不仅费时费力，甚至这种改造是不可能的。

本文在分析机器人控制平台的基础上，为了达到平台开放性与实时性要求，为先进的控制算法和多传感器融合提供基础，首先提出工业计算机与嵌入式运动控制器相结合的结构，两者通过网络通信。工业计算机作为主控制器使用，完成非实时任务，嵌入式运动控制器符合 IEC61131 规范，用于实现机器人控制。并在此结构下，完成了平台的硬件搭建与软件设计。

接着为了提高机器人的操作速度和控制性能，对路径规划与轨迹跟踪进行了研究。针对并联机器人典型的拾放操作路径，提出了更合适的椭圆路径与改进正弦速度规划相结合的方案，并推导出基于机器人动力学的时间最优方程。这种运动规划是连续的，在高速应用中可以避免振动。通过对比机器人的控制方式，在分析了运动学与动力学控制优劣后，采用了控制性能更优的计算力矩控制算法。通过仿真，验证了控制算法的有效性。建立了 Delta 机器人非简化的动力学模型，为动力学控制与优化算法提供基础。

最后，为了验证平台的性能及模型的正确性，在建立的平台上进行了相关实验。实验对比了两种控制方式、两种运动规划及时间最优算法的轨迹误差与力矩大小，实验结果表明采用计算力矩控制算法、椭圆路径运动规划及相应的时间最优算法后，Delta 机器人的性能得到较大的提高。

**关键词：**开放式控制平台；并联机器人；路径规划；动力学模型

## Abstract

Industrial robots are essential equipment for the automated manufacturing production line. It can instead of people to complete repeated long-time operations, or execute tasks in dangerous harsh operating environment, which will greatly improved production efficiency. For the purpose of making the robots having the power to complete more complex tasks and enhance robot intelligence, many sensors, such as tactile sensors, force sensors or vision sensors, and many advanced control algorithms, such as fuzzy control, robust control, need to be integrated into the industrial robot, for making it has the ability to identify or further adaptive, self-learning ability. But the normal commercial robot platforms are not openness, the transformation of the platform, need the assistance of developers and equipment specified by the developer. And this transformation is not only time-consuming, but also need great effort. Sometimes this transformation is even impossible.

In this thesis, based on the analysis of the robot control platform, in order to achieve openness and real-time requirements of the platform, providing a basis for advanced control algorithms and multi-sensor fusion, first, the structure of combination of industrial computers and embedded motion controllers are proposed, both communicate through network. Industrial computers are used as the main controller, which perform non real-time tasks. Embedded motion controller is compatible with the IEC61131 standard, which are used to completing for robot control. And according to this structure, hardware construction and software design of the platform are completed.

Then, intending to improve the operation speed and control performance of the robot, the research of the path planning and path tracking has been done. For the typical pick-and-place operation path of parallel robots, the elliptical path with modified sine motion profile is purposed, and the dynamics based time-optimal motion planning is derived. This motion is smooth and suitable for high speed applications to avoid vibration. By comparing robot control algorithms, in the analysis of the pros and cons of kinematics and dynamics control, a control algorithm, computed torque control algorithm which has a better performance, is used. A simulation has been established to verify the effectiveness of the control algorithm. For providing a basis for dynamic control algorithms and optimization algorithms, a full Delta robot dynamics model has been derived.

Finally, in order to verify the performance of the platform and the accuracy of the Delta robot model of this thesis, some related experiments have been done on the established platform. The comparison of the trajectory error and the torque of the two control algorithms, two motion planning and time-optimal algorithm are presented. The results show that the performance of the Delta robot has been greatly improved by using the computed torque control algorithm and elliptical path motion planning with the time-optimal algorithm.

**Keywords:** open control platform, parallel robot, motion planning, dynamic model

# 目 录

摘 要 .....	I
ABSTRACT .....	II
第 1 章 绪 论 .....	1
1.1 课题背景 .....	1
1.1.1 开放式机器人控制平台 .....	1
1.1.2 开放式机器人控制平台的意义 .....	2
1.2 研究现状 .....	3
1.2.1 开放式机器人控制平台的国外研究现状 .....	3
1.2.2 开放式机器人平台系统的国内研究现状 .....	3
1.3 本文的主要研究内容 .....	4
1.3.1 本文的主要贡献 .....	4
1.3.2 本文主要研究内容 .....	4
第 2 章 平台架构与理论基础 .....	5
2.1 机器人控制平台的组成 .....	5
2.1.1 控制平台的硬件架构 .....	5
2.1.2 控制平台的软件架构 .....	6
2.1.3 IEC61131 .....	7
2.2 运动规划基本理论 .....	8
2.3 机器人控制理论 .....	10
2.3.1 经典 PID 控制理论 .....	10
2.3.2 计算力矩控制理论 .....	12
2.4 小结 .....	13
第 3 章 Delta 机器人模型与运动规划 .....	14
3.1 Delta 机器人运动学建模 .....	14
3.1.1 Delta 机器人逆解 .....	14
3.1.2 Delta 机器人正解 .....	16
3.2 Delta 机器人动力学建模 .....	16
3.2.1 Delta 机器人雅可比矩阵 .....	16
3.2.2 Delta 机器人动力学方程 .....	17
3.3 控制系统仿真 .....	20
3.4 运动规划及时间最优算法 .....	21

3.4.1 椭圆路径及速度规划 .....	22
3.4.2 时间最优算法 .....	25
3.5 小结.....	30
<b>第 4 章 平台搭建与软件设计 .....</b>	<b>31</b>
4.1 平台基本架构.....	31
4.1.1 工业计算机与固高运动控制器 .....	31
4.1.2 控制电气图 .....	32
4.2 软件基本架构.....	33
4.2.1 运动控制器流程图 .....	33
4.2.2 工业计算机流程图 .....	34
4.3 小结.....	35
<b>第 5 章 实验与分析 .....</b>	<b>36</b>
5.1 实验平台介绍.....	36
5.2 控制方式对比实验.....	37
5.3 轨迹性能对比实验.....	39
5.4 优化对比实验.....	40
5.5 小结.....	43
<b>结 论 .....</b>	<b>44</b>
主要工作和创新点.....	44
后续研究工作展望.....	44
<b>参考文献 .....</b>	<b>45</b>
攻读学位期间发表的学术论文 .....	49
哈尔滨工业大学学位论文原创性声明及使用授权说明 .....	50
<b>致 谢 .....</b>	<b>51</b>



# 第1章 绪 论

## 1.1 课题背景

### 1.1.1 开放式机器人控制平台

机器人在人们日常生活中已经有不少的应用了，例如工业机器人用于搬运、焊接、喷涂等生产现场，移动机器人用于探测、拍摄等，当然也有特殊应用例如用于手术、康复等。工业机器人（如图 1-1）为人类完成了大量高强度、重复性的工作，成为现代制造系统中不可或缺的一种自动化装备，被广泛应用在汽车、飞机、电子产品等行业的制造过程中。机器人技术一直是人们最关注的科学研究领域之一，作为高自动化、智能化的机电一体化设备，由于采用计算机编程，可以自主完成各种任务，具有较高的可靠性、灵活性和快速的反应能力。可以说，机器人技术已成为衡量一个国家综合技术水平的标志之一<sup>[1,2]</sup>。



图 1-1 工业机器人

机器人控制平台作为机器人的必要组成部分，是不可或缺的。机器人控制平台一般包括一组硬件，例如传感器、计算机、电机，和一组软件，例如规划、控制算法，用于实现对机器人控制，完成所需要的任务。根据不同的需求和机器人的类型，对控制平台功能的要求也各不相同，包括轨迹生成、伺服控制算法、路径规划以及各种智能优化方法。机器人性能受很多方面的影响，但控制平台作为机器人的基础部分，它必定对机器人技术的提升起到重要作用<sup>[3]</sup>。

随着生产水平的提高，人们对机器人的性能提出了越来越高的要求，人们将

越来越多的先进的控制算法应用到机器人中，例如力/位置混合控制、多机器人协调控制等，相应的，对机器人控制平台要求也越来越高。要提高机器人的性能或使其拥有对不确定环境的适用能力即拥有一定的智能，具有开放结构使其能集成各种外部设备并提供实时性的保证是控制平台必需<sup>[4,5]</sup>。而传统的商用机器人平台采用不开放的专用控制平台，通常是由机器人厂家自己开发的，使用自己的架构，采用封闭的协议，并把各种算法都固化到芯片上，虽然可以降低平台成本，提高平台的可靠性，却导致控制平台扩展性差，功能单一，软件移植困难，网络通信功能弱。如果应用环境有所改变，当平台无法适用新的应用时，改造的代价将会非常大。这样的平台不仅无法满足研究的需要，而且也无法满足现代化工业生产尤其是制造业对机器人在不同的应用场合和多品种生产适应能力的要求。

为了适应发展，许多组织都对开放式控制平台进行了研究。“开放式控制平台”，不同项目和组织对其的定义各有侧重。例如 IEEE 主要表述为一个开放式控制平台应用应该能提供不同组件甚至不同平台的互操作能力并展示给用户的接口是统一的，无差别的<sup>[6,7]</sup>。一个开放式控制平台应该具有良好的平台独立性，能够将不同厂家的不同设备整合到一起，能够与不同平台相互通信，具有统一的用户接口，而且必不可少的是平台能够提供实时性保证。在生产厂家只需提供少量的甚至不提供帮助的情况下，用户可以根据不同需求对平台各部分，包括硬件及软件，都能进行替换或修改。更具体来说，控制平台的硬件结构和软件结构应该是标准的，开放的，是基于模块化设计的，硬件上可以集成各种外部设备，软件上可以建立新的架构，当然包括新的功能模块，例如伺服算法。

### 1.1.2 开放式机器人控制平台的意义

虽然开放式平台的研究已经发展了不短的时间，但公认的开放式平台的定义却没有，大多都只是定性的描述了开放式系统特征。通常认为开放式平台通常有以下特征：可扩展性；可移植性；互操作性；可减裁性，当然也有更具体的如：使用常用的开放的计算机硬件平台；采用标准操作系统和标准开发语言；采用结构化设计；采用标准总线结构；采用网络通信等<sup>[8-10]</sup>。

开放式机器人控制平台对用户、厂家、制造产业都是有重要意义的。对用户来说，是有很强吸引力的。开放式机器人控制平台使得机器人组件模块化，接口标准化，使用户可以用最小的代价改造平台，以适用新的需求，包括增减功能，实现新的算法。而且由于标准是开放的，用户可以很容易的从遵守相同标准的不同厂家获得所需的功能模块，减少再次完成重新开发的必要。在遵守现有标准前提下，用户完全可以自由选择模块，以组成自己的、专有的机器人控制平台。

由于开放式的结构，相同的模块可以由不同的厂家提供，用户可以自由选择。厂家之间的竞争，可以使得搭建平台时的可选择性变大，可以选择最适合需求的产品，产品的可靠性也会越高。由于开放式平台的可重用性，可以利用现在的成熟模块和技术，不断提高各模块的性能，相应的平台性能也会提高，从而促进机器人产业的发展。所以对于厂家和制造产业来说，也是有很大意义的。

## 1.2 研究现状

### 1.2.1 开放式机器人控制平台的国外研究现状

国际上，开放式机器人控制平台的研究虽然经过多年发展，但依然不够成熟。虽然很多研究团体及组织已经有不少的研究成果，但大多只是停留在制定标准、理论层次或实验阶段，实际应用并不多。比较大的几个计划，例如美国的 NGC<sup>[11]</sup> 及开放式模块体系结构控制机构<sup>[12]</sup>，欧共体的自动化系统中的开放式系统结构<sup>[13]</sup>，日本的控制器的开放式环境计划<sup>[14]</sup>，都是有比较大的影响力的。当然。这些都不是纯粹的只单纯应用于机器人的计划，其不仅仅包括硬件及软件结构，还包括系统的开发，它们主要是针对控制器，包括电机、机床、机器人等。除此之外，还有许多人致力于开放式系统的实际应用。在 2012 年的 ICRA 会议上，有专门的小组讨论了相关问题。有相关论文讨论了一些机器人制造厂家在开放式控制平台上的努力。例如 KUKA，其将控制需要划分为几个层次，从非实时到实时，在不同应用上都提供了相关的模块和开发支持，较好的满足了平台的开放与实时要求<sup>[15]</sup>。还有 Comau，将不同的机器人模块连接到工业网络，根据不同的需求，可以采用不同的控制方式，控制不同的机器人模块，开发了一个较好的控制平台<sup>[16]</sup>。当然也有对以前非开放系统的改造相关的论文<sup>[17]</sup>。可以说开放式控制平台在国外发展势头是很好的。当标准方法的制定与实际应用可以相结合的时候，开放控制平台的发展一定可以有长足的发展。

### 1.2.2 开放式机器人平台系统的国内研究现状

国内开放式机器人控制平台的研究相比于国外就比较少了，首先并没有统一的计划或组织推动开放式控制平台的发展，大多数都是具体的单机器人的实际研究。虽然没有国内机器人厂商的研究支持，但学校研究方面与国外相似，都借助开 PC 这个标准平台来提供开放性，具体在实性的保证上，就有不同的实现方式。例如上海交通大学机器人研究所描述了一种基于 IPC+实时扩展的结构模型，借助于 RTX 的方式，可以保证实时性，并快速实现控制平台的建立，而且易用性好，

可以实现纯软件控制方式<sup>[18]</sup>。华中科技大学的潘炼东则采用 PC+PMAC (DSP 运动控制卡) 的实现方式, 由 PMAC 提供时性<sup>[3]</sup>。哈尔滨工业大学研制出了 PC+分布式处理器的控制结构, 由分布式处理器上的编程实现可以满足实时性<sup>[19]</sup>。机器人的发展需要控制平台提供坚实的基础, 国内虽然还处于起步阶段, 但对机器人行业一定会起到积极作用。

## 1.3 本文的主要研究内容

### 1.3.1 本文的主要贡献

传统的机器人控制平台由于是封闭的, 无法进行改造, 即使能够进行改造, 也需要开发商的协助, 使用特定的设备, 平台的改造难度大, 费时费力。为了解决上述问题, 本文提出工业计算机及嵌入式运动控制器相结合的结构, 由于工业计算机采用 PC 架构并且嵌入式控制器符合 IEC61131 规范, 可以满足平台的开放性和实时性要求。

本文对 Delta 机器人进行动力学建模, 提出其非简化的动力学模型。为了提高并联机器人典型的拾放操作速度, 提出椭圆路径与改进正弦速度规划相结合的方式, 并提出基于机器人动力学的最优时间算法。采用计算力矩控制方式, 机器人控制性能有了较大改善。

### 1.3.2 本文主要研究内容

本文首先从传统的机器人控制平台入手, 阐述了传统平台的缺点, 并结合当今国内外研究状况, 在此基础上提出开放式控制平台的研究内容。阐述了通常所用的开放式控制平台实现方式, 描述了本文中机器人控制平台所采用的架构, 包括硬件组成和软件架构。讨论了机器人的基础理论, 包括运动规划与控制算法。

接下来, 本文对 Delta 机器人进行运动学和动力学建模, 提出非简化的动力学模型。通过仿真, 验证了控制算法的正确性。提出更适合于并联机器人拾放操作的运动规划算法, 并推导出基于机器人动力学的最优时间算法。

本文接着描述了控制系统具体的硬件组成及软件架构, 将实时与非实时任务分开, 并在此平台上实现了计算力矩控制算法及本文描述的拾放操作路径。

最后, 本文在此平台基础上进行了相关实验。实验表明, 机器人控制平台可以较好的满足开放性和实时性要求, 非简化的 Delta 机器人动力学模型是正确的。并且计算力矩控制算法可以取得较好的控制效果, 新的运动规划及优化算法可以提高机器人的操作速度。

## 第2章 平台架构与理论基础

本章主要介绍机器人控制平台的实现架构及机器人的相关理论。

控制平台架构虽然并不是机器人研究中最热门的方向，但是没有强大的控制平台架构，其它机器人研究方向就都不能实现，其重要性可见一斑。随着机器人研究的深入，机器人平台除了要求开放性外，还需要实时性保证，才能为各种先进的控制算法、信息融合提供实现的基础。

影响机器人性能有两个关键技术，一个是机器人运动规划，另一个是机器人控制算法。

运动规划是根据操作任务，在已知的约束条件下，包括路径约束及执行机构的约束等，规划出理想的机器人运动，例如路径和与时间有关的速度及加速度等，如何使机器人的运动准确、平稳，高效率的完成任务，一直都是机器人研究的热点，对实际应用具有重要意义。

系统在受到不可测且不可预知的扰动时能保持被控量与给定之间没有稳态误差的控制系统称为伺服系统<sup>[20]</sup>。当给定机器人期望位置或运动轨迹时，就需要控制算法不断缩小误差，使机器人按要求运动。本文描述了两比较常用的机器人控制方式，都可以有效的应用于机器人控制。

### 2.1 机器人控制平台的组成

#### 2.1.1 控制平台的硬件架构

合适的硬件实现架构对开放式控制平台的研发会起事半功倍的效果。如何减少控制平台基础部分的研发比例，把主要精力放在实际应用研发，硬件平台的选择是很重要的。硬件平台应满足：

- (1) 硬件平台接口是标准，尽量使用网络，以提供网络通信能力；
- (2) 硬件平台必须提供实时计算能力，最好使用硬件方式提供；
- (3) 硬件平台是模块化设计的，并且尽量提供最大化的统一接口；
- (4) 低成本，具有可剪裁性。

PC 机虽然发展的时间不长，却已形成了统一的开放的标准。硬件层面，不仅拥有标准的内部结构，其组件可以自由选择，而且有众多满足不同需求的总线和外部接口，当需要扩展时，可以将设备连接到总线或外部接口；软件层面，不仅包括操作系统作为基础软件平台，还有各种应用软件，开源的与非开源的，易于

满足各种需求；还有就是提供 PC 相关组件的厂家多，行业已发展得相当完备，有低成本、开放式特征。无论是国内还是国外，已经有相当多的研究组织都以 PC 为基础研发开放式控制平台<sup>[21]</sup>。当然，PC 机发展之初是以个人电脑为目的的，其开放性毋庸置疑，但应用到机器人控制时却达不到要求，特别是实时性方面。虽然可以通过软件系统的改造，例如商业的 Windows 实时性扩展及一些开源的实时扩展 RTLinux<sup>[22,23]</sup>，或所采用实时操作系统，例如 RTEMS，VxWorks<sup>[24]</sup>，使其达到实时性，但是要么价格昂贵，要么就支持性不好，开发难度大，开放性差。故还需要一个开放的，能满足实时性要求的运动控制器。

可编程控制器（简称 PLC）在工控领域应用极为广泛。伴随计算机、通讯、控制技术的不断发展，PLC 控制系统也不断经历着重大变革，国际标准 IEC61131-3 的颁布可以说是 PLC 发展的一个里程碑。该标准的颁布为 PLC 的软件设计提供了标准化的编程概念和编程方法<sup>[25]</sup>。随着标准的推广，越来越多的公司都提供了相应的支持，包括一些运动控制厂家，如 Banchman，GoogolTech。PLC 由于其稳定的控制响应及高度的可靠性，非常适用于机器人控制。国际标准的颁布，其开放性的特性也符合控制平台的要求。

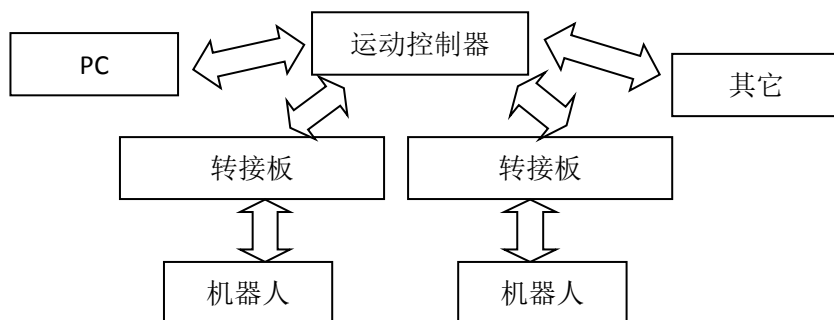


图 2-1 基于 PC+运动控制器的开放式机器人示意图

故本文提供采用 PC+运动控制器（符合 IEC61131 标准）的实现模式，借助于 PC 平台的开放性及其功能强大的各种软件，为系统开发提供便捷路径，可以满足大多数应用，搭配符合 IEC61131 标准的 PLC，由于其可靠性、高实时性，可以为机器人控制算法等应用提供条件，以满足机器人控制平台开放性与实时性要求。图 2-1 为系统的硬件控制框图。

### 2.1.2 控制平台的软件架构

在硬件未进行扩展时，控制平台的硬件资源是固定的、有限的。如何利用有限的资源提供尽可能多功能，完成尽可能多的任务，是软件设计时必须考虑。一个机器人控制平台一般具备如图 2-2 所示的功能，这些任务可以分为三类：

- (1) 突发性任务，如检测到碰撞、急停按钮被按下、平台温度过高等任务；

- (2) 周期性任务，如运动学正逆解、机器人控制算法、传感器采样等；
- (3) 非实时性任务，主要包括人机界面、机器人程序解析系统、手动操作命令的读取、非实时视觉的处理、机器人状态显示等。

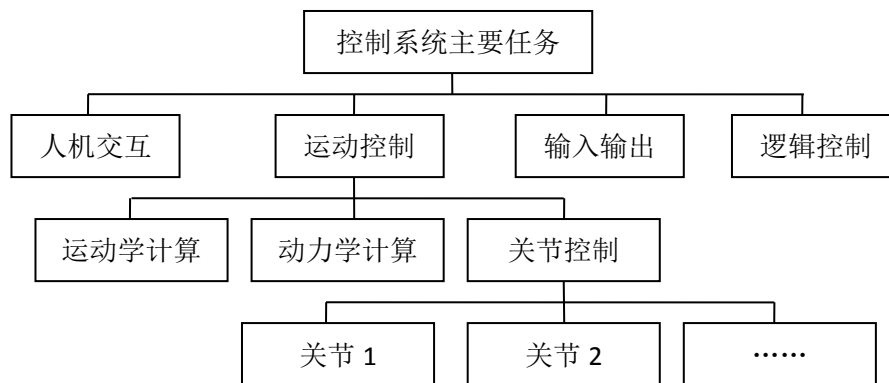


图 2-2 控制系统主要任务

在上述硬件平台的基础上，对任务进行了划分，如图 2-3，工业计算机主要用于完成非实时任务，采用 Windows NT 作为系统平台，可以满足平台的独立性和开放性。运动控制器主要用于实现机器人的控制，如控制算法、紧急状态处理、机器人信息采集，由于符合 IEC61131，开发比较容易。最后是驱动器，控制电机完成操作。

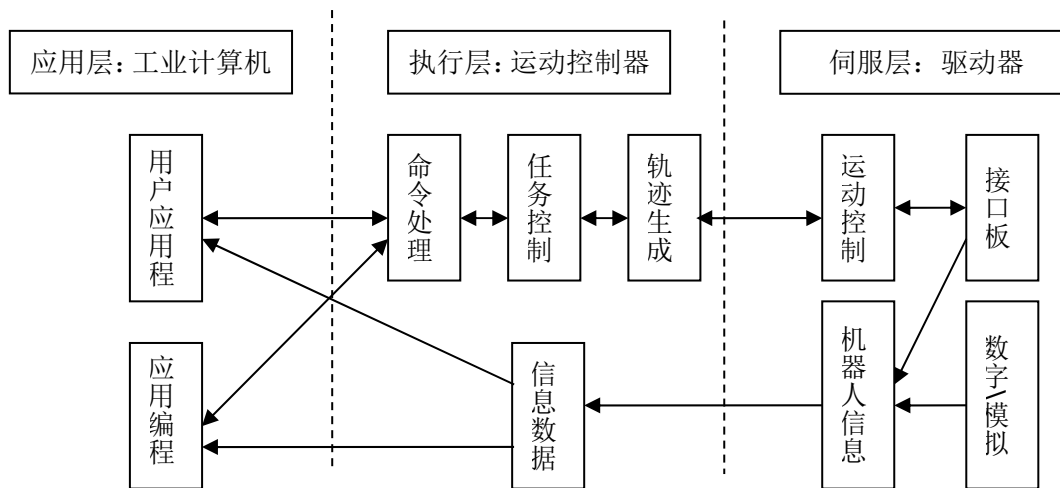


图 2-3 控制系统软件架构

### 2.1.3 IEC61131

IEC61131，在 IEC 编号系统更改之前称为 IEC1131，标准分成以下几个部分：通用信息、设备要求与测试、编程语言规范、用户向导、通讯规范、模糊控制软件的实施，语言部分实现的导则<sup>[26]</sup>。自从 1993 年 IEC61131 颁布以来，由于是在众多 PLC 厂商的参与下发展而来的，不仅对以前的 PLC 程序有很好的兼容性，而

且为了适用未来的发展,克服 PLC 的缺点,标准很好的扩展了 PLC 的功能,从此 PLC 不仅具备了高可靠性,而且也拥有丰富的功能。由于得到了众多设备供应商的支持,使其得到快速的推广。虽然符合 IEC61131 标准的设备已经在控制工业中得到了很大的发展与应用,但将其应用到机器人中却并不多,或者说应该是刚刚开始。虽然将标准应用于机器人还并不是很成熟,但已经有人开始尝试。例如 GoogolTech 就有基于 IEC61131 的机器人控制器。由于 IEC61131 可以在 PC 平台上实现,其实就是在 PC 平台上搭建一个实时的操作平台,但由于平台是兼容,只要是兼容的平台都可以运行相同程序。随着时间的推移,IEC61131 也显示出一些不足,由于其是根据单 PLC 模型建立的,不符合多 PLC 的应用,于是为了适应需求,标准 IEC61131-3 开始与 IEC61499 相结合,借助现场总线和工业以太网,让 IEC61131 还能应用于分布式控制<sup>[27]</sup>。

## 2.2 运动规划基本理论

机器人的运动规划一般分为两个阶段,分别为路径规划和轨迹跟踪<sup>[28,29]</sup>。在路径规划阶段,需要考虑的有路径形状,避障等几何问题,这决定了实际的运动路径。而路径跟踪阶段,主要任务是找到一个优化目标函数,并在保证系统稳定的情况下,沿着指定路径移动机器人。优化目标函数有:(1)最短运动时间;(2)最小能量;(3)最小加加速度。此外,混合目标函数也被提出(如时间能量最优轨迹规划)<sup>[30]</sup>。时间最优算法因为可以有效的提高机器人操作速度,所以在机器人研究中相当有吸引力。

机器人路径规划是轨迹跟踪的基础,轨迹跟踪是路径规划的实现。工业机器人根据不同的需求,对路径规划的要求也不一样。机器人的路径规划可以分为两个部分,一是任务空间,另一是关节空间。任务空间的路径规划,一般是在笛卡尔坐标系下,通常包括直线、圆弧两种基本路径,而且路径参数通常是六维的,包括位置和姿态,有时也会受机器人自由度的限制,路径参数会有所减少。在实际应用中,其他复杂的路径都由这两种基本路径来描述。关节空间的路径规划,由角度坐标表示,其是关于时间的函数,通常需要由机器人逆解求得相应每个关节的角度坐标,运动时各关节独立运动,同时到达目标点,中间所经过的路径是不清楚的,所以在许多对路径没要求的场合得到采用<sup>[31]</sup>。

轨迹跟踪目标是使机器人按照理想的任务空间或关节空间路径运动,如何在满足运动学约束、动力学约束的情况下,使机器人轨迹跟踪的精度足够高,当然这也与机器人控制算法有关。根据不同的目标,不同的约束方程,会得到不同的插补值。在关节空间下的轨迹跟踪,控制软件一般会根据路径参数进行插补,插补数据由一系列位姿点来表示,通常用于需要轨迹控制的场合,但由于每个位姿



点都需要机器人的逆解运算，运算量大。在关节空间中，由于其是关于时间的单变量函数，插补算法简单、快速，且可以避免机器人的奇异点。

轨迹跟踪的算法有很多，但是很少能够应用到工业现场中。工业应用需要可靠、简单、快速算法。在不同路径连接的地方，控制软件会进行特殊处理，将过渡路径变光滑。现在常用于机器人轨迹跟踪的算法有梯型曲线，S 型曲线。现在就常用的曲线进行对比，把距离与时间都规定为单位长度。结果如图 2-4。

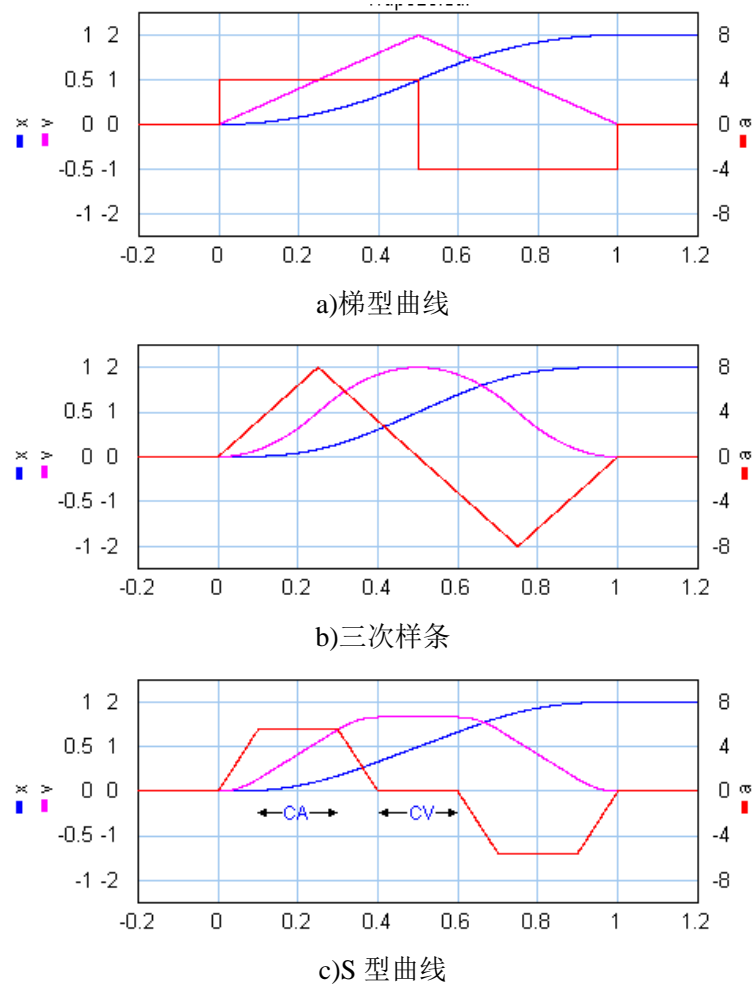


图 2-4 轨迹跟踪规划

图 2-4 中 c)参数为  $CV = 20\%$ ， $CA = 20\%$ 。具体数值如表 2-1，从表中可以看出，梯型曲线由于是二阶规划，有无穷大的加加速度，不利于机器人稳定长久运行。而对于三阶规划，虽然加加速度是有限的，但是却有更大的加速度。

表 2-1 规划对图

规划	阶数	最大速度	最大力速度	初始加加速度	中间加加速度
梯型曲线	2	2	4	正无穷	负无穷
三次样条	3	2	8	32	-32
S 型曲线	3	1.65	5.55	55.6	0

## 2.3 机器人控制理论

### 2.3.1 经典 PID 控制理论

PID 控制理论是现阶段应用最广泛的控制算法,其在机器人控制领域也是应用得最多。由于其控制的鲁棒性及快速性,即使在许多不能建模的对象上也可以取得较好的控制效果。对于工业机器人这种高度非线性的控制对象,采用 PID 算法也可以满足大多数应用的要求。

经典 PID 算法可以用式(2-1)描述:

$$u(t) = K_p[e(t) + \frac{1}{T_i} \int_0^t e(\tau) d\tau + T_d \frac{de(t)}{dt}] \quad (2-1)$$

式中  $u$ ——控制器输出

$e$ ——参考值和反馈值的误差

$T_d, T_i, K_p$ ——PID 控制器比例增益、积分时间常数、微分时间常数

PID 算法由比例环节、积分环节、微分环节组成。大多数情况下,微分环节或积分环节会与比例环节组成相应的控制器,例如 PD、PI、PID 控制器。当然也可以使用纯比例环节(P),其控制方程为:

$$u(t) = K_p e(t) + u_b \quad (2-2)$$

式中  $u_b$ ——控制器输出偏移值。

引入比例环节后,随着  $K_p$  的增大系统响应速度会提高,而且还能有效减少稳态误差,但过大的  $K_p$  会导致系统的相对稳定性下降,甚至造成不稳定<sup>[40]</sup>。

当使用比例加微分(PD)控制算法时,控制方程可写为:

$$u(t) = K_p [e(t) + T_d [de(t)/dt]] \quad (2-3)$$

微分环节可以预测误差的变化趋势,并及时采取措施以控制系统,可以改善系统的动态性能,但其控制作用仅在误差变化时有效。虽然微分环节有超前控制的作用,但若存在高频干扰信号时,也会使系统陷入震荡而不能稳定<sup>[40]</sup>。

当使用比例加积分(PI)控制算法时,控制方程可写为:

$$u(t) = K_p [e(t) + \frac{1}{T_i} \int_0^t e(\tau) d\tau] \quad (2-4)$$

理论上积分环节可以完全消除系统的稳态误差,但系统稳定性会有所下降。通过调节积分时间常数,可以在对系统的稳定性影响不大的前提下,有效地改善

系统的稳态性能<sup>[40]</sup>。

经典 PID 控制理论是模拟系统使用的。由于本平台采用符合 IEC61131 标准的运动控制器，是数字控制系统，其反馈和控制信号都是离散的，只能在指定间隔运行控制算法，所以需要离散化的 PID 算法，以满足要求。

假设系统的采样周期为  $T$ ，采样序号为  $k$ 。将连续时间  $t$  离散为  $kT$ 。则有：

$$\begin{cases} t = kT \\ \int_0^t e(\tau) d\tau = T \sum_{j=0}^k e(jT) \\ \frac{de(t)}{dt} = \frac{e(kT) - e[(k-1)T]}{T} \end{cases} \quad (2-5)$$

代入经典 PID 控制算法(2-1)可得：

$$u_k = K_p \left[ e(k) + \frac{T}{T_i} \sum_{j=0}^k e(j) + \frac{T_d}{T} (e(k) - e(k-1)) \right] \quad (2-6)$$

令  $K_i = K_p T / T_i$ ，称为积分系数， $K_d = K_p T_d / T$ ，称为微分系数，则：

$$u_k = K_p e(k) + K_i \sum_{j=0}^k e(j) + K_d (e(k) - e(k-1)) \quad (2-7)$$

式中  $u_k$ ——第  $k$  次控制器的输出值

$e(k)$ ——第  $k$  次输入的误差值

无论是模拟还是数字 PID 控制器，参数的调节作用都是相同的，其次，采样周期对参数的选择也有较大影响的。另一种常见的数字 PID 算法称为增量式算法。

按照式(2-7)，第  $k-1$  次控制周期的输出值为：

$$u_{k-1} = K_p e(k-1) + K_i \sum_{j=0}^{k-1} e(j) + K_d (e(k-1) - e(k-2)) \quad (2-8)$$

而第  $k$  次控制周期的输出值已经由式(2-7)给出，因此，将式(2-7)与式(2-8)相减得到每次控制输出的增量值：

$$\begin{aligned} \Delta u_k &= u_k - u_{k-1} \\ &= K_p (e(k) - e(k-1)) + K_i \left( \sum_{j=0}^k e(j) - \sum_{j=0}^{k-1} e(j) \right) \\ &\quad + K_d [(e(k) - e(k-1)) - (e(k-1) - e(k-2))] \\ &= K_p (e(k) - e(k-1)) + K_i e(k) + K_d (e(k) - 2e(k-1) + e(k-2)) \end{aligned} \quad (2-9)$$

从式(2-9)可以看出,增量式 PID 算法并不需要进行  $e(k)$  的累加,可以通过三次控制周期误差值与上次控制输出值即解算出当前控制输出值,以后增量式 PID 算法更适用于离散控制器的场合<sup>[41]</sup>。

由于数字控制系统的原因,离散 PID 控制算法能够实现模拟元件难以实现的特殊 PID 控制规律。许多改进的算法被提出来,并在实际中得到广泛应用。例如对饱和效应的改进,可采用积分分离法。用于防止微分冲击可采用不完全微分法等。在一些有特殊要求的应用中,提出了带死区的 PID 控制算法,纯滞后算法等。

### 2.3.2 计算力矩控制理论

虽然 PID 控制算法对于大多数应用,包括机器人控制,都可以取得比较满意的效果,但由于机器人的高度非线性,由于离心力、科氏力及重力项的影响,对于一些特殊应用,特别的高速高加速的情况下,往往不能达到很好的控制效果,为此,许多非线性控制算法被应用到机器人控制器。

计算力矩控制理论,算法简单,有效。通过逆运动学前馈和非线性反馈相结合的方式,不断调节伺服电机输出,根据已经状态,不断调节前馈,对误差进行反馈补偿,极大的减少了误差<sup>[42]</sup>。其控制框图如图 2-5:

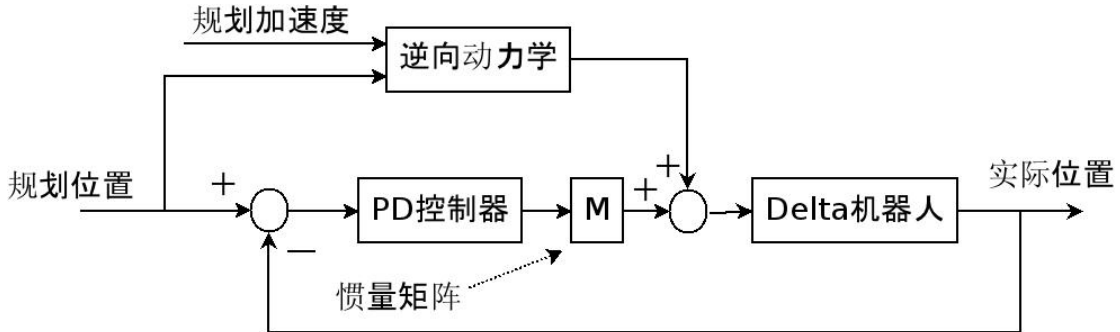


图 2-5 计算力矩控制框图

假设已知机器人当前位置和速度,并忽略所有非线性因素,根据实际机器人准确的模型,为了使机器人达到所需加速度,则当前所需力矩如式(2-10):

$$\tau = M(\theta)\ddot{\theta} + C(\theta, \dot{\theta})\dot{\theta} + N(\theta, \dot{\theta}) \quad (2-10)$$

式中  $\tau$ ——控制信号

$M, C, N$ ——惯量矩阵,科氏矩阵,重力矩阵

当然,这只是有理想状态下,如果模型或当前状态不准确,则采用式(2-10)控制规律并不能使机器人按要求运动,于是可以引入控制算法如式(2-11):

$$\tau = M(\theta)(\ddot{\theta}_d + K_v\dot{e} + K_p e) + C(\theta, \dot{\theta})\dot{\theta} + N(\theta, \dot{\theta}) \quad (2-11)$$

式中  $\tau$ ——控制信号

$e$ ——给定和反馈的误差

$K_v, K_p$ ——PD 控制器的比例增益，微分增益

将式(2-11)与式(2-10)相减得：

$$M(\theta)(\ddot{e} + K_v\dot{e} + K_p e) = 0 \quad (2-12)$$

由于  $M$  正定的，所以由式(2-12)可知，系统可以减少误差。

计算力矩控制算法是一个很好的反馈线性化的例子，通过一个全状态的非线性反馈，把一个非线性系统变为一个线性系统<sup>[43]</sup>。

虽然计算力矩控制在理论上拥有很好的控制效果，但在实际应用中，却要避免由于机器人奇异点而产生的矩阵无解、输出突变等情况的发生。计算力矩控制算法包括两个部分，控制算法可以被重写为式(2-13)：

$$\tau = \underbrace{M(\theta)\ddot{\theta}_d + C(\theta, \dot{\theta})\dot{\theta} + N(\theta, \dot{\theta})}_{\tau_{ff}} + \underbrace{M(\theta)(-K_v\dot{e} - K_p e)}_{\tau_{fb}} \quad (2-13)$$

$\tau_{ff}$  是前馈项，提供所需力矩驱动机器人按预定轨迹运动， $\tau_{fb}$  是反馈项，提供矫正力矩，减少机器人运动时的轨迹误差<sup>[43]</sup>。至于 PD 控制器，可以采用前小结采用的数字 PID 控制算法实现。如何建立  $M, C, N$  是实现计算力矩控制的关键所在，不仅要考虑机器人结构、物理规律，防止机器人到达奇异位型，还要避免数学无解，解无穷大等情况。机器人动力学的建立一直是机器人研究的热门方向。

## 2.4 小结

本章首先介绍了机器人控制平台的实现方式，讨论了现在常用的平台架构，提出采用 PC+运动控制器（符合 IEC61131 标准）模式，并指出了平台硬件架构及软件架构，可以较好的满足系统开放式与实时性的要求。接着讨论了机器人运动规划相关的理论，包括路径规划与轨迹跟踪，总结了现在常用的算法。最后，讨论了在机器人控制方面使用最多的 PID 控制算法，以及为了满足特殊需要，针对机器人特点而使用的计算力矩控制算法。

## 第 3 章 Delta 机器人模型与运动规划

工业机器人根据结构的不同通常分为串联的和并联的。串联机器人一般采用开链式结构，其往往具有多个关节，每一个执行机构连接一个关节，因结构简单，工作范围大，可多自由度运动，其在工业应用上是最多的，例如搬运、焊接等。并联机器人则是多条支链连接在基座和末端上，其具有运动惯量低、负载能力强、刚度大等优点，使得并联机器人成为一个高速度、高精度的运动平台<sup>[44,45]</sup>。Delta 机器人是并联机器人的典型例子，其虽然只有三个平动自由度，却拥有高加速、高速特性，特别适用于快速抓取的拾放操作，在食品包装行业应用广泛。

如何发挥并联机器人最大的性能，极大的提高生产效率，对实际应用有重要的意义。运动规划是影响机器人操作速度关键，需要从路径规划与轨迹跟踪入手，根据实际，采用最适合机器人的运动规划算法。

本章对 Delta 机器人结构进行数学建模，着重分析了动力学模型的建立。由于 Delta 机器人主要用于高速抓取，虽然常用的单轴 PID 控制算法可以满足要求，但控制性能却不尽如意，需要动力学模型以采用计算力矩控制算法提高控制性能。

### 3.1 Delta 机器人运动学建模

#### 3.1.1 Delta 机器人逆解

本文研究对象 Delta 机器人主要由固定的动平台、主动臂、从动臂，动平台组成，有三条支链，分别连接在定平台和动平台上。图 3-1 为 Delta 机器人基本结构。

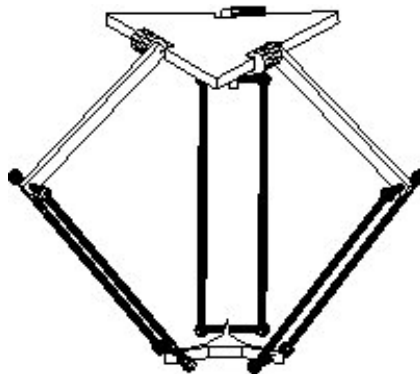


图 3-1 机器人的基本结构

Delta 机器人其实已经有很多相关方面的建模<sup>[46,47]</sup>。Delta 机器人的三个主动臂等角度均匀的分布在定平台上，角度分别为  $0^\circ$ ， $120^\circ$ ， $240^\circ$ 。由于其结构的特殊性，机器人动平台永远平行于定平台，也就是说 Delta 机器人拥有 3 个平动自由

度。Delta 机器人是最快的一款机器人之一。

数学建模时，可以只分析一个支链，因为每条支链是等效的，故建立如图 3-2 所示的数学模型，并在定平台中心建立如图所示的 o-xyz 坐标系。根据其几何结构，可以得到式(3-1)及式(3-2)，于是每支条链下的每个角度都可以计算得出。

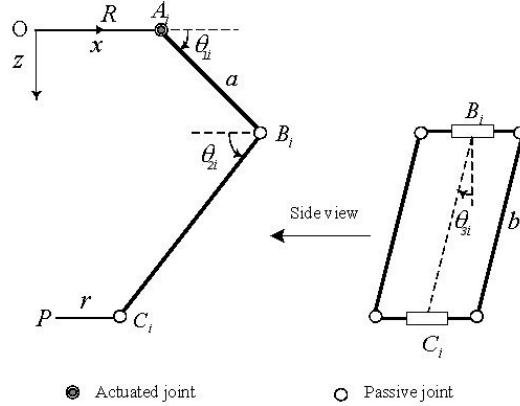


图 3-2 Delta 的第 i 条支链的数学模型

$$a \begin{bmatrix} \cos(\theta_{1i}) \\ 0 \\ \sin(\theta_{1i}) \end{bmatrix} + b \begin{bmatrix} \sin(\theta_{3i}) \cos(\theta_{1i} + \theta_{2i}) \\ \cos(\theta_{3i}) \\ \sin(\theta_{3i}) \sin(\theta_{1i} + \theta_{2i}) \end{bmatrix} = \begin{bmatrix} C_{xi} \\ C_{yi} \\ C_{zi} \end{bmatrix} \quad (3-1)$$

$$\begin{bmatrix} C_{xi} \\ C_{yi} \\ C_{zi} \end{bmatrix} = \begin{bmatrix} \cos(\varphi_i) & \sin(\varphi_i) & 0 \\ -\sin(\varphi_i) & \cos(\varphi_i) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_x \\ X_y \\ X_z \end{bmatrix} - \begin{bmatrix} R-r \\ 0 \\ 0 \end{bmatrix} \quad (3-2)$$

式中  $i$ ——为每个轴的序号，分别为 1, 2, 3

$\theta_{1i}, \theta_{2i}, \theta_{3i}$ ——每条支链的主动角和两个从动臂角

$\psi$ ——每条支链在定平台的分布角

$R, a, b, r$ ——为定平台半径、主动臂长度，从动臂长度，动平台半径

$X, C$ ——分别为动平台直角坐标，动平台球角直角坐标

对于每个已知的动平台坐标，可以解出  $\theta_{2i}, \theta_{3i}$ 。

$$\theta_{3i} = \cos^{-1} \left( \frac{C_{yi}}{b} \right) \quad (3-3)$$

$$\theta_{2i} = \cos^{-1} \left( \frac{C_{xi}^2 + C_{yi}^2 + C_{zi}^2 - a^2 - b^2}{2ab \sin(\theta_{3i})} \right) \quad (3-4)$$

对于  $\theta_{1i}$  可以得到式(3-5):

$$2aC_{xi} \cos(\theta_{1i}) + 2aC_{zi} \sin(\theta_{1i}) = C_{xi}^2 + C_{yi}^2 + C_{zi}^2 + a^2 - b^2 \quad (3-5)$$

对式(3-5)代入三角函数万能公式，可以得解  $\theta_{1i}$ 。

### 3.1.2 Delta 机器人正解

对于 Delta 机器人正解，并没有通用的解析表达式，通常是利用几何约束，联合求解式(3-6)和式(3-7)，最终可以得到机器人的数值解。

$$b^2 = (X_x - B_{1i})^2 + (X_y - B_{2i})^2 + (X_z - B_{3i})^2 \quad (3-6)$$

$$\begin{bmatrix} B_{1i} \\ B_{2i} \\ B_{3i} \end{bmatrix} = \begin{bmatrix} (a \cos(\theta_{1i}) + R - r) \cos(\varphi_i) \\ (a \cos(\theta_{1i}) + R - r) \sin(\varphi_i) \\ a \sin(\theta_{1i}) \end{bmatrix} \quad (3-7)$$

式中  $B$ ——为主动臂与从动臂相接点的直角坐标。

上两式为三元两次方程组，有三个方程，故可以得到数值解。

## 3.2 Delta 机器人动力学建模

### 3.2.1 Delta 机器人雅可比矩阵

雅可比矩阵描述的是关节速度与末端速度、关节力矩与末端力的关系。动力学建模中最重要的一环就是获得机器人的雅可比矩阵，也是分析机器人奇异点、性能的基础。从图 3-2 可得出闭环等式(3-8)：

$$OP + PC_i = OA_i + A_i B_i + B_i C_i \quad (3-8)$$

式(3-8)对时间的微分可以得到式(3-9)：

$$\dot{X} = \omega_{1i} \times a_i + \omega_{2i} \times b_i \quad (3-9)$$

式中  $\omega_{1i}$ ,  $\omega_{2i}$ ——为主动臂和从动臂的角速度向量

$a_i$ ,  $b_i$ ——向量  $A_i B_i$ ,  $B_i C_i$

为了消除  $\omega_{2i}$ ，可以让等式两边点乘  $b_i$ ，可以得到式(3-10)：

$$b_i \bullet \dot{X} = \omega_{1i} \bullet (a_i \times b_i) \quad (3-10)$$

式中的相关表示式如下：

$$a_i = a [\cos(\theta_{1i}) \quad 0 \quad \sin(\theta_{1i})]^T \quad (3-11)$$

$$b_i = b [\sin(\theta_{3i}) \cos(\theta_{1i} + \theta_{2i}) \quad \cos(\theta_{3i}) \quad \sin(\theta_{3i}) \sin(\theta_{1i} + \theta_{2i})]^T \quad (3-12)$$

$$\omega_i = [0 \quad -\dot{\theta}_{1i} \quad 0] \quad (3-13)$$



$$\dot{X}_i = \begin{bmatrix} \cos(\varphi_i) & \sin(\varphi_i) & 0 \\ -\sin(\varphi_i) & \cos(\varphi_i) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{X}_x \\ \dot{X}_y \\ \dot{X}_z \end{bmatrix} \quad (3-14)$$

把以上表达式写成矩阵形式，最终我们可以得到式(3-15)：

$$J_x \dot{X} = J_q \dot{q} \quad (3-15)$$

式(3-15)中  $J_x$  和  $J_q$  都可以写成解析形式，表达式如下：

$$J_q = a \begin{bmatrix} J_{q1} & 0 & 0 \\ 0 & J_{q2} & 0 \\ 0 & 0 & J_{q3} \end{bmatrix} \quad (3-16)$$

$$J_{qi} = C_{zi} \cos(\theta_{li}) - C_{xi} \sin(\theta_{li}) \quad (3-17)$$

$$J_x = [J_{x1} \quad J_{x2} \quad J_{x3}]^T \quad (3-18)$$

$$J_{xi} = \begin{bmatrix} X_x - (a \cos(\theta_{li}) + R - r) \cos(\varphi_i) \\ X_y - (a \cos(\theta_{li}) + R - r) \sin(\varphi_i) \\ X_z - a \sin(\theta_{li}) \end{bmatrix}^T \quad (3-19)$$

最终我们可以得到雅可比矩阵的数值解：

$$J = J_x^{-1} J_q \quad (3-20)$$

### 3.2.2 Delta 机器人动力学方程

由于 Delta 机器人三条支链通过动平台连接在一起，其动学方程较复杂。通过观察得出，可以分析从动臂的动力学方程，从而把 Delta 机器人整体连接起来<sup>[48]</sup>。

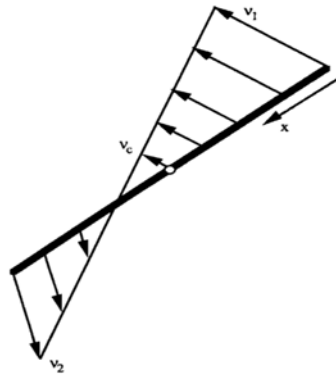


图 3-3 从动臂分析模型

图 3-3 为从动臂模型，假设臂两端有速度  $v_1, v_2$ ，则其中心速度  $v_c$  及角速度  $\omega_c$  为：

$$v_c = \frac{1}{2}(v_1 + v_2) = \frac{1}{2}(J_1 + J_2)\dot{q} \quad (3-21)$$

$$\omega_c = \frac{1}{L}\bar{e}_L(v_1 + v_2) = \frac{1}{L}\bar{e}_L(J_1 + J_2)\dot{q} \quad (3-22)$$

式中  $J_1, J_2$ ——为首端和末端的雅可比矩阵

假设臂两端有加速度  $a_1, a_2$ ，根据虚功原理，从动臂所需力矩为：

$$\tau = \frac{1}{3}m \left[ J_1^T \left( a_1 + \frac{1}{2}a_2 \right) + J_2^T \left( a_2 + \frac{1}{2}a_1 \right) \right] \quad (3-23)$$

通过以上的简单介绍，可以得到整个 Delta 机器人的数学模型，设主动臂、肘关节、从动臂、动平台质量为  $m_a, m_e, m_b, m_p$ ：

$$\tau_p + \sum_{i=1}^3 \tau_{ai} + \sum_{i=1}^3 \tau_{bi} = 0 \quad (3-24)$$

式中  $\tau_p, \tau_{ai}, \tau_{bi}$ ——为动平台、主动臂和从动臂所受力矩

具体表达式如下：

$$\tau_p = J^T (m_p \ddot{X} - G_p) \quad (3-25)$$

$$\tau_a = \sum_{i=1}^3 \tau_{ai} = I_a \ddot{q} - G_a - \tau \quad (3-26)$$

$$\tau_{bi} = \frac{1}{3}m_b \left[ J^T \left( \ddot{X} + \frac{1}{2}a_{ui} \right) + J_{ui}^T \left( a_{ui} + \frac{1}{2}\ddot{X} \right) \right] - \frac{1}{2}(J + J_{ui})^T G_b \quad (3-27)$$

式中  $\tau$ ——为电机输出力矩

$G_p, G_a, G_b$ ——为动平台、主动臂和从动臂所受重力

其中主动臂与从动臂连接点的速度及加速分别为：

$$v_{ui} = \dot{\theta}_{li} = J_{ui}\dot{q} \quad (3-28)$$

$$a_{ui} = J_{u^2i}\dot{q} + J_{ui}\ddot{q} \quad (3-29)$$

将具体算式代入并化简得式(3-30)：

$$\begin{aligned} \tau = & \left( I_{at} + \frac{1}{6}J^T J_u m_b \right) \ddot{q} + \left[ J^T (m_p + m_b) + \frac{1}{6}J_u^T m_b \right] \ddot{X} \\ & + \frac{1}{6}J^T J_{u^2} m_b \dot{q}^2 - G_{pt} - G_{at} \end{aligned} \quad (3-30)$$

式中的具体表达达到如下：

$$I_{at} = \begin{bmatrix} I_{at1} & 0 & 0 \\ 0 & I_{at2} & 0 \\ 0 & 0 & I_{at3} \end{bmatrix} \quad (3-31)$$

$$I_{ati} = I_m + a^2 \left( \frac{1}{3} m_a + m_e + \frac{1}{3} m_b \right) \quad (3-32)$$

$$G_{at} = a \left( \frac{1}{2} m_a + m_e + \frac{1}{2} m_b \right) g [\cos(\theta_{11}) \quad \cos(\theta_{12}) \quad \cos(\theta_{13})]^T \quad (3-33)$$

$$G_{pt} = J^T \left( m_p + \frac{3}{2} m_b \right) [0 \quad 0 \quad g]^T \quad (3-34)$$

如果采用普通方式，主动臂角速度和动平台加速度之间并不能得到解析表达式，于是为了得到运算结果，许多文献都采用简化形式，将从动臂的 1/3 质量计入动平台，而将 2/3 质量计入主动臂中，以得到更简单的模型而又不失正确性的模型<sup>[48]</sup>。这种方式对于分析不利，特别是在高速、高加速情况下，误差会变得很大，而且如果用于控制，由于不能得到两者之间的转换，最终只能使用数值差分的方式求得，容易引起数值噪声。现在计算机速度越来越快，对于非简化模型依然可快速计算出来，只要有合适的表达式。为了得到非简化模型的计算，对式(3-15)求导得：

$$J_q \ddot{q} + \dot{J}_q \dot{q} = J_x \ddot{X} + \dot{J}_x \dot{X} \quad (3-35)$$

通过式(3-35)，可以消去关节加速度或末端加速度，可以分别建立关节空间或任务空间的动力学方程。式(3-36)将模型转换到任务空间：

$$\ddot{q} = J^{-1} \ddot{X} + J_q^{-1} (\dot{J}_x \dot{X} - \dot{J}_q \dot{q}) \quad (3-36)$$

式(3-36)中每个表达式都可以得到数值解，故可以得主动臂角加速度与动平台加速度之间的关系。将式(3-36)代入式(3-30)得：

$$\begin{aligned} \tau = & \left[ \left( I_{at} + \frac{1}{6} J^T J_u m_b \right) J^{-1} + J^T (m_p + m_b) + \frac{1}{6} J_u^T m_b \right] \ddot{X} \\ & + \left( I_{at} + \frac{1}{6} J^T J_u m_b \right) J_q^{-1} (\dot{J}_x \dot{X} - \dot{J}_q \dot{q}) + \frac{1}{6} J^T J_{u^2} m_b \dot{q}^2 - G_{pt} - G_{at} \end{aligned} \quad (3-37)$$

此表达虽比较复杂，但计算时间并不长，在平台为 linux 下的 CPU 为 Q8200，主频为 2GHz 的计算时间为每 1000 次 4506us，其中 Delta 机器人正解为每 1000 次 719us，逆解为每 1000 次 2218us，故足够在控制算法中应用。

### 3.3 控制系统仿真

在已建立的 Delta 机器人动力学模型的基础上，可以对控制系统进行仿真。对于本文提到的两种控制算法，单轴 PID 控制算法在实际中已有众多成功的应用，而控制力矩算法则应用较少。对于新的控制算法，可以通过仿真，来初步验证其有效性，并熟悉控制算法相关注意事项，避免实际应用中的原则性错误。当然，这只是对实际控制对象抽象化简而建立的数学模型，还需要实验的进一步验证。

本次仿真采用 MATLAB Simulink 工具箱，Simulink 是 Matlab 软件下的一个附加组件，功能强大，易于使用，可以为各类应用提供仿真组件与条件。Simulink 为用户提供了一个基于 Windows 的用户界面（GUI），利用鼠标单击和拖拉方式，就可以建立系统模型，具有更直观、更方便、更灵活的优点。用户的精力可以主要集中在到系统模型的构建而非软件的实现上<sup>[49]</sup>。且 MATLAB Simulink 工具箱中的经过多代的修改与优化，还集成了调试功能，比起使用其它仿真软件具有易用、可靠的优点。

此次仿真计算力矩控制算法，考虑实际使用时，模型参数并不能准确知道，故控制器所使用的参数与机器人参数并不使用相同值，但几何参数相同，其参数如表 3-1：

表 3-1 仿真参数

参数	控制器使用参数	机器人使用参数
定平台半径 $R$	0.200m	0.200m
主动臂长度 $a$	0.347m	0.347m
从动臂长度 $b$	0.806m	0.806m
动平台半径 $r$	0.050m	0.050m
电机惯量 $I_m$	$1.8 \times 10^{-4} \text{ kg} \cdot \text{m}^2$	$2.0733 \times 10^{-4} \text{ kg} \cdot \text{m}^2$
主动臂质量 $m_a$	0.7 kg	0.80294 kg
肘关节质量 $m_e$	0.5 kg	0.532165 kg
从动臂质量 $m_b$	0.4 kg	0.42144 kg
动平台质量 $m_p$	1 kg	1.8310481 kg

仿真图主要有控制器，电机模型，Delta 机器人模型。控制器采用计算力矩控制算法，通过调节  $K_v$ ， $K_p$  来调节控制量。电机采用简化模型，等效为一阶惯性环境，时间常数为 1ms。

根据计算力矩控制算法的结构框图，构建如图 3-4 所示控制系统仿真图。输入是椭圆路径规划，包括位置、速度、加速度，单程时间是 0.3s，仿真时间为 4s。起点是[0.150, 0, 0.750]m，终点为[-0.150, 0, 0.750]m，高度为 0.025m，总共 6 个来回。对于输出的角度值，经过正解转换到任务空间。

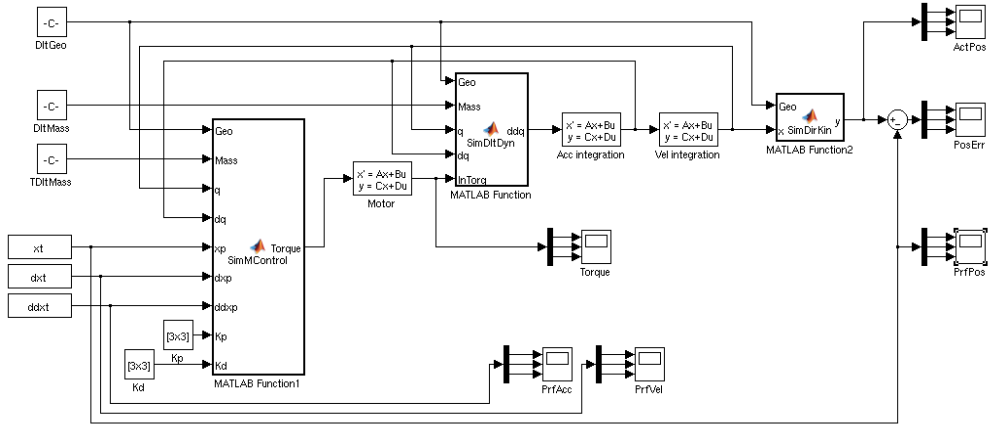


图 3-4 计算力矩控制算法仿真图

图 3-5 为输出结果，分别为实际路径和规划路径，从上到下分别为 x-y-z 轴坐标值。从结果可以看出，计算力矩可以很好的控制机器人的运动轨迹。在仿真中调节控制参数时，可以得出只要  $K_p$  和  $K_v$  足够大，即使模型参数与实际参数存在一定的误差，也可以通过非线性反馈得到纠正。通过仿真，可以得出算法是有效的。

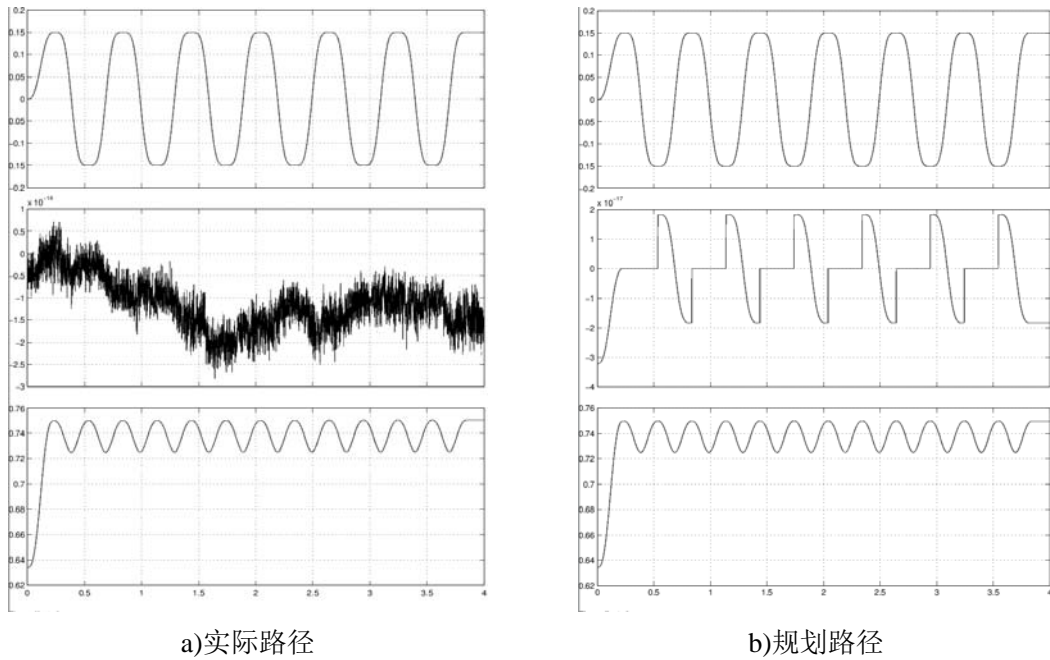


图 3-5 计算力矩控制算法仿真结果

### 3.4 运动规划及时间最优算法

时间最优算法一直是机器人运动规划中热门的研究方向。在早期文献中，时间最优算法大致可分为两种类型：基于运动学的和基于动力学的。基于运动学的，会使用最大加速度和最大加加速度限制，显然，由此产生的运动是平滑的<sup>[32,33]</sup>。

然而，由于该方法并没有考虑到动力学，它并不是“真正的”最优时间。因为机器人在不同的位置下其真正的最大加速度和最大加加速度是变化的，而上述方法在整个工作空间中的都是恒定值。

基于动力学的，通过引入机器人动力学，在整个工作空间内，都可以得到系统可以达到的最大加速度，所以是真正的时间最优算法。有的把时间最优算法一般被写成一个数学优化问题，它可以通过各种优化技术解决。例如，凸优化方法在<sup>[28,29]</sup>和的可变容差法<sup>[34]</sup>。也有基于在相平面上的伪速度曲线的方法。伪速度约束曲线可以由动力学限制多边形生成的，在该约束内寻找使速度尽可能大的最优曲线<sup>[35,36]</sup>。然而，以上基于动力学的时间最优算法都会导致加速度的突然变化。虽然其路径是任意的，但其实对于并联机器人的特殊路径就没有太大作用。并且这样的轨迹跟踪算法，因加速度存在不连续的情况，通常会导致严重的振动、机械磨损或输出饱和度等，特别是在高速应用，这种情况是一定避免的。虽然也有相关文献讨论了相关问题<sup>[37,38]</sup>，例如通过增加有关加速度的约束或电机力矩变化率的约束，但却运算时间长，不适用实际应用。

### 3.4.1 椭圆路径及速度规划

并联机器人由于其结构特性，特别适合快速的拾放操作，Delta 机器人就是一个典型例子。拾放操作路径通常由 4 个点定义，组成一个长方形，由长度和高度描述，通常为矩形路径。机器人拾放速度通常是以来回一次  $25 \times 300 \times 25\text{mm}$  的轨迹时间表达，称为节拍时间。对于高速应用来说，机器人运动的平滑性是非常重要的，要求运动轨迹的位置、速度、加速度都是连续的。常用轨迹为矩形路径，直线之间会采用圆弧连接，轨迹只有一阶连续性，只在位置及速度上是连续的。通常会将其变换到关节空间中或采用更高阶的插补路径算法，使其在转接处有较好的连续性，但这种算法存在计算复杂、转换不方便的缺点。对于这种典型轨迹，椭圆路径可以更好的满足要求<sup>[39]</sup>。两种路径如图 3-6 和图 3-7 所示。

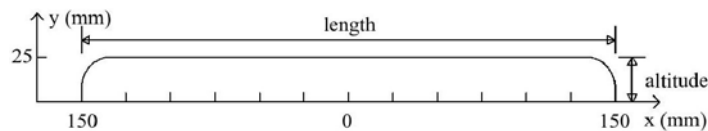


图 3-6 矩形路径

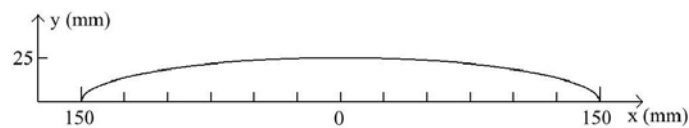


图 3-7 椭圆路径

椭圆的三维路径如图 3-8，对于并联机器人的抓取应用，只需规定起点  $P_1$  和终点  $P_2$  及需要升高的距离，就可以完成路径的运算。此轨迹的特点是在  $P_1$  和  $P_2$  其运动路径是垂直于  $z$  轴的，是垂直的伸高和落下，且会高于  $P_1$  和  $P_2$  连接线。对比矩形路径，这种路径是由三角函数描述的，路径是加速度连续的，很好的满足了拾放操作的需求。

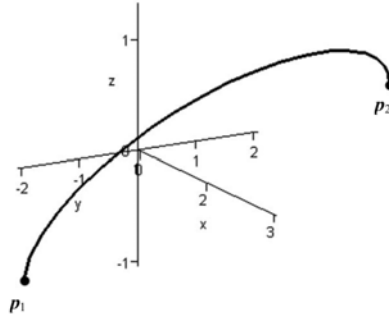


图 3-8 椭圆路径示意图

以下是对这种路径的数学分析，在规定一个起点  $P_1$ 、终点  $P_2$  和升高距离  $\delta$  后，就可以确定具体的路径。数学模型如图。通过分析，得到式(3-38)，从式中可以得知，除  $s$  规划因子外，其他参数都是在插补前确定的，最终只需要与一个常数矩阵相乘，就可以得到所需的路径中间值。

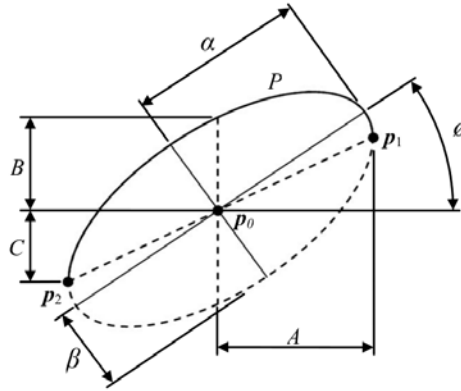


图 3-9 椭圆路径的数学描述

$$P = P_0 + \begin{bmatrix} \cos(\varphi) & -\sin(\varphi) & 0 \\ \sin(\varphi) & \cos(\varphi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} A & 0 & 0 \\ 0 & 1 & 0 \\ C & 0 & B \end{bmatrix} \begin{bmatrix} \cos(\pi s) \\ 0 \\ \sin(\pi s) \end{bmatrix} \quad (3-38)$$

式中  $P$ ——椭圆路径点

$P_0$ ——椭圆路径中心点

$\varphi$ ——椭圆路径在  $xy$  轴平面的投影与  $x$  轴的夹角

$s$ ——椭圆路径的规划因子

式中各参数具体表达式如下：

$$P_o = \frac{1}{2}(P_1 + P_2) \quad (3-39)$$

$$\varphi = \tan^{-1}(P_{1y} - P_{2y}, P_{1x} - P_{2x}) \quad (3-40)$$

$$A = \frac{1}{2} \sqrt{(P_{1x} - P_{2x})^2 + (P_{1y} - P_{2y})^2} \quad (3-41)$$

$$B = \delta \quad (3-42)$$

$$C = \frac{1}{2}(P_{1z} - P_{2z}) \quad (3-43)$$

在规划因子的选择上，也就是速度规划的选择，本文采用改进正弦速度规划。此规划的特点是通过两个参数，即加加速度因子  $a$  及不对称因子  $A_f$ ，可以控制路径上的最大速度、最大加速度，这种规划是单位时间下的规划，当然其他速度规划也是可以应用的，但综合比较这种规划拥有更好的性能。改进正弦速度规划的数学表达式如式(3-44)。此速度规划可以自由调节加加速度时间和加减速时间，对于机器人的非线性特点可以取得较好的规划效果。

$$s = \begin{cases} f_1(u) & 0 \leq u \leq A_f a \\ f_2(u) & A_f a < u \leq A_f \\ f_3(u) & A_f < u \leq 1 - (1 - A_f)a \\ f_4(u) & 1 - (1 - A_f)a < u \leq 1 \end{cases} \quad (3-44)$$

式中具体表达式如下：

$$\begin{aligned} f_1(u) &= \frac{C_a a}{\pi} u - \frac{2C_a A_f a^2}{\pi^2} \sin\left(\frac{\pi}{2A_f a} u\right) \\ f_2(u) &= \frac{2C_a A_f (1-2a)}{\pi^2} u + \frac{C_a a}{\pi} u \\ &\quad - \frac{2C_a A_f (1-a)^2}{\pi^2} \sin\left(\frac{\pi}{2A_f (1-a)} u + \frac{(1-2a)\pi}{2(1-a)}\right) \\ f_3(u) &= 1 - \frac{2C_a (1-A_f)(1-2a)}{\pi^2} u - \frac{C_a a}{\pi} (1-u) \\ &\quad + \frac{2C_a (1-A_f)(1-a)^2}{\pi^2} \sin\left(\frac{\pi}{2(1-A_f)(1-a)} (1-u) + \frac{(1-2a)\pi}{2(1-a)}\right) \\ f_4(u) &= 1 - \frac{C_a a}{\pi} (1-u) - \frac{2C_a (1-A_f)a^2}{\pi^2} \sin\left(\frac{\pi}{2(1-A_f)a} (1-u)\right) \end{aligned} \quad (3-45)$$

其中：



$$\frac{C_a a}{\pi} + \frac{2C_a(1-2a)}{\pi^2} = 1 \quad (3-46)$$

$$u = \frac{t}{T} \quad (3-47)$$

在  $a=0.25$  及  $A_f=0.5$  时, 改进正弦速度规划如图 3-10 所示。由图可知, 此速度规划是一个加速度连续的规划。由于使用正弦函数作为规划函数, 可以得到很光滑的加速度曲线。

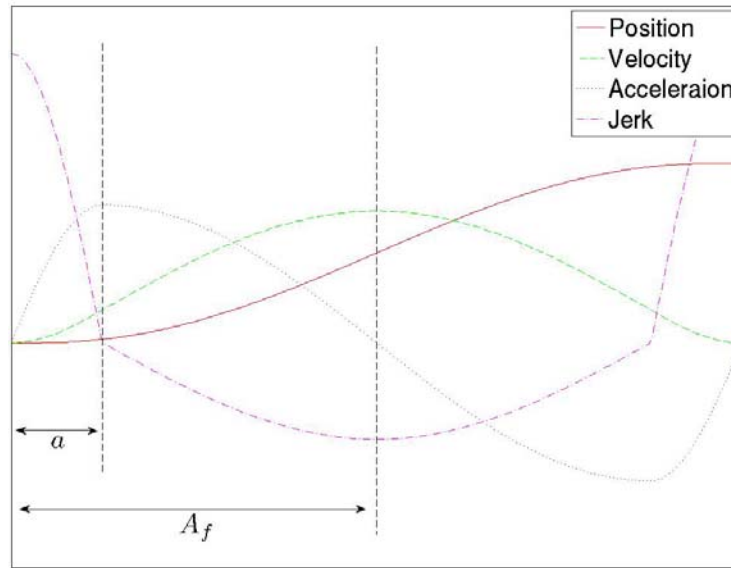


图 3-10 改进正弦速度规划示意图

将路径与速度规划合并, 可以表达成式(3-48),  $t$  是自变量, 其它参数是根据实际需要而改变, 例如路径及运动时间等:

$$s = f(t, P_1, P_2, \delta, a, A_f, T) \quad (3-48)$$

### 3.4.2 时间最优算法

对于以上运动规划, 其实最后可以调节的参数有三个, 分别为  $a$ ,  $A_f$ ,  $T$ 。如何通过调节  $a$ 、 $A_f$ , 得到一个最优的  $T$ , 是本文的关键。

由于运动规划方程中, 路径通常由于实际应用决定, 也就是说路径是不可改变的, 故优化方程可以改写为与路径相关的函数。从此出发, 机器人的动力学模型需要写成与位置相关的方程。由于运动规划通常多维的, 例如本文中的运动规划就是三维的, 可以将其表达成参数方程的形式, 如上节所示。这样多变量的路径方程就可以变成单变量形式。

在建立优化方程时, 可以将路径的参数方程与机器人逆解相结合, 最终机器

人路径会表达成机器人关节坐标下的参数方程，如式(3-49)和式(3-50)：

$$\dot{q}(s) = q'(s) \dot{s} \quad (3-49)$$

$$\ddot{q}(s) = q'(s) \ddot{s} + q''(s) \dot{s}^2 \quad (3-50)$$

机器人的动力学方程通常表达如式(3-51)：

$$\tau = M(q) \ddot{q} + C(q, \dot{q}) \dot{q} + F(q) \operatorname{sgn}(\dot{q}) + N(q) \quad (3-51)$$

式中  $M$ ——机器人惯量矩阵

$C$ ——机器人科氏力矩阵

$F$ ——机器人中与速度方程有关的力矩，如摩擦力

$N$ ——机器人中与位置有关的力矩，如重力

将式(3-49)和式(3-50)代入式，可得到式(3-52)：

$$\tau(t) = M_s(s(t)) \ddot{s}(t) + C_s(s(t)) \dot{s}(t)^2 + N_s(s(t)) \quad (3-52)$$

式中表达式如下：

$$M_s(s) = M(q(s)) q'(s) \quad (3-53)$$

$$C_s(s) = M(q(s)) q''(s) + C(q(s), q'(s)) q'(s) \quad (3-54)$$

$$N_s(s) = F(q(s)) \operatorname{sgn}(q'(s)) + N(q(s)) \quad (3-55)$$

基于机器人动力学的最优时间方程可以表达成式(3-56)和(3-57)：

$$\min T \quad (3-56)$$

$$\text{s.t.} \begin{cases} \tau_{\min}(s) \leq M_s(s) \ddot{s} + C_s(s) \dot{s}^2 + N_s(s) \leq \tau_{\max}(s) \\ s(0) = P_1 \\ s(T) = P_2 \\ \dot{s}(0) = \dot{P}_1 \\ \dot{s}(T) = \dot{P}_2 \end{cases} \quad (3-57)$$

for  $t = [0, T]$

除了电机输出力矩的约束处，还有起点和终点及两点速度的约束。根据实际情况，约束方程可以有所不同。通常情况下，约束条件是已知的，而且始末速度通常为0。对于此方程的求解，由于可以求得机器人指定位姿下的约束方程，故通常是采用路径等分的形式，求出路径中每点的可行解，以所有可行解中最小时间为最优解，其解的精度与最小长度有关。

具体到椭圆路径运动规划，虽然路径是已知或者是已经被限定的了。但速度规划方程是可调的，通过时间缩放因子，可以调节规划中速度和加速度大小，通常的规划如式(3-58)：

$$s = f(t) \quad (3-58)$$

令原时间与缩放后的时间关系如式(3-59)：

$$t = \lambda t' \quad (3-59)$$

于是在相同路径下，两者速度与加速度之间的关系如式(3-60)和式(3-61)：

$$\dot{s}(t') = \lambda \dot{s}(t) \quad (3-60)$$

$$\ddot{s}(t') = \lambda^2 \ddot{s}(t) \quad (3-61)$$

其实约束方程中，除了电机力矩的约束外，还可以有电机速度和加速度的约束。其约束表达如式(3-62)和式(3-63)：

$$\ddot{q}_{\min}(s) \leq q'(s)\ddot{s} + q''(s)\dot{s}^2 \leq \ddot{q}_{\max}(s) \quad (3-62)$$

$$\dot{q}_{\min}(s) \leq q'(s)\dot{s} \leq \dot{q}_{\max}(s) \quad (3-63)$$

于是最优时间方程表达成式(3-64)和(3-65)：

$$\min T \quad (3-64)$$

$$\text{s.t.} \begin{cases} \tau_{\min}(s) \leq M_s(s) \frac{\ddot{s}}{T^2} + C_s(s) \frac{\dot{s}^2}{T^2} + N_s(s) \leq \tau_{\max}(s) \\ \ddot{q}_{\min}(s) \leq q'(s) \frac{\ddot{s}}{T^2} + q''(s) \frac{\dot{s}^2}{T^2} \leq \ddot{q}_{\max}(s) \\ \dot{q}_{\min}(s) \leq q'(s) \frac{\dot{s}}{T} \leq \dot{q}_{\max}(s) \\ s = f(t, P_1, P_2, \delta, a, A_f, 1) \\ s(0) = P_1 \\ s(T) = P_2 \\ \dot{s}(0) = 0 \\ \dot{s}(T) = 0 \end{cases} \quad (3-65)$$

for  $t = [0, T]$

此优化方程将机器人动力学方程与椭圆规划方程结合在一起，将本复杂的优化方程，变为三个变量的优化方程，极大的减少了优化的参数。此方程的求解，

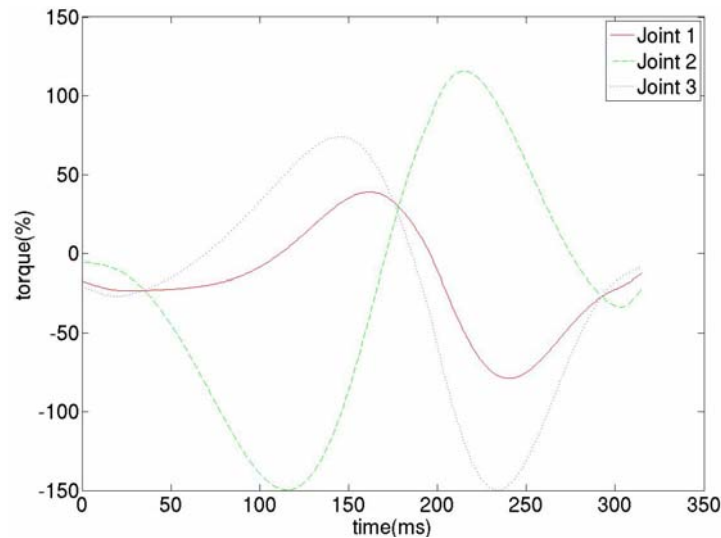
对于参数  $a$ ，只是控制了加加速度的时间，也相当于控制了力矩变化率，故通常是保持其参数不变，而求解最优的  $A_f$ 。

在建立了其动力学方程的基础上，进一步完成了其的时间最优算法。对于此方程的求解方式，采用对半查找的方式。可以很快的求得满足要求的最优解。在平台为 linux 下的 CPU 为 Q8200，主频为 2GHz 的计算时间为每 1000 将 2560ms，每个轨迹的计算时间为 2.5ms 左右，故足够在实际中应用。

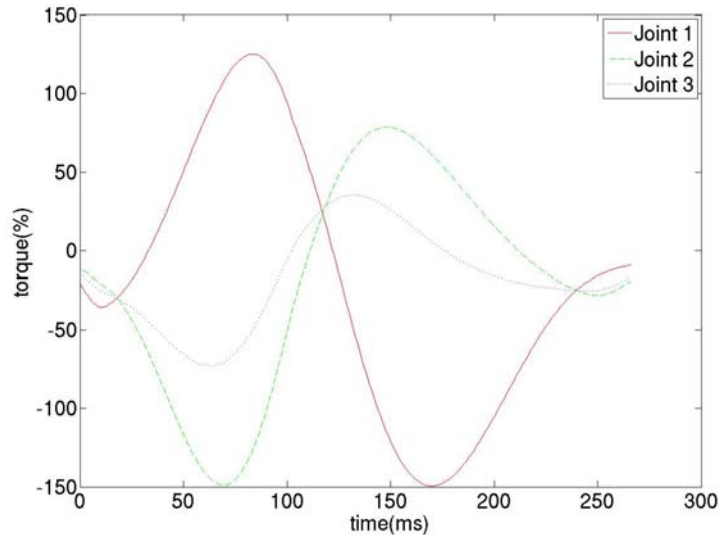
表 3-2 优化算法中机器人参数

参数	机器人仿真参数
定平台半径 $R$	0.200m
主动臂长度 $a$	0.347m
从动臂长度 $b$	0.806m
动平台半径 $r$	0.050m
电机惯量 $I_m$	$2.0733 \times 10^{-4} \text{ kg} \cdot \text{m}^2$
主动臂质量 $m_a$	0.80294 kg
肘关节质量 $m_e$	0.532165 kg
从动臂质量 $m_b$	0.42144 kg
动平台质量 $m_p$	1.1766 kg
关节粘滞摩擦系数 $K_v$	2.3N/(rad/s)
关节库仑摩擦力 $F_v$	1.3N

假设输出端力矩额定值为  $36.3580\text{N} \cdot \text{m}$ ，最大角速度为  $39.27\text{rad/s}$ ，最大角加速度为  $1937.5\text{rad/s}^2$ ，按照 1.5 倍额定力矩为输出极限。以下是两个条路径的规划仿真情况，第一条路径从  $[0.200, 0.200, 0.750]\text{m}$  到  $[-0.200, -0.200, 0.750]\text{m}$ ，高度为  $0.025\text{m}$ ，第二条路径从  $[-0.050, 0.200, 0.750]\text{m}$  到  $[0.050, -0.200, 0.750]\text{m}$ ，高度为  $0.025\text{m}$ 。机器人参数如表 3-2。



a) 路径 1,  $a=0.125$ ,  $A_f=0.6201$ ,  $T=0.315\text{s}$



b) 路径 2,  $a=0.125$ ,  $A_f=0.3881$ ,  $T=0.266s$

图 3-11 路径优化后的力矩输出

由图 3-11 可以看出,通过调节  $A_f$  参数,可以得到最短时间  $T$ ,使最大输出力矩限制在需要的范围内。就 Delta 机器人而言,现在普通电机已经可以满足其最大速度的要求,但力矩却依然表现不足,通常设计时,会采用大减速比的减速器来实现大加速度。表 3-3 与表 3-4 是在不同条件下得到的最短时间,随着  $a$  的增加,最短时间也会随之增加,而在相同  $a$  的情况下,调节  $A_f$  可以使时间更短。当然  $A_f$  的调节也不是无限制,在实际应用,会使其在 0.1-0.9 中取值。

表 3-3 路径 1 优化结果

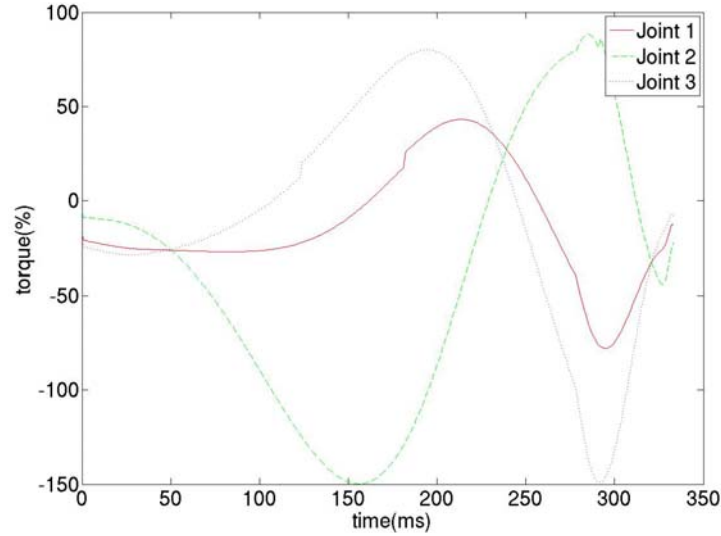
轨迹参数 $a$	轨迹参数 $A_f$	优化时间 $T$
0.125	0.6201	0.315s
0.250	0.6166	0.334s
0.500	0.6072	0.377s
0.125	0.5000	0.337s

表 3-4 路径 2 优化结果

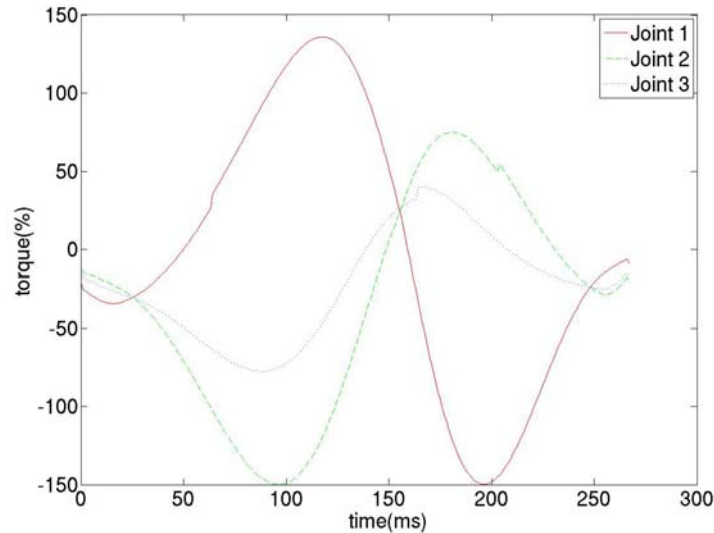
轨迹参数 $a$	轨迹参数 $A_f$	优化时间 $T$
0.125	0.3881	0.266s
0.250	0.3893	0.281s
0.500	0.3986	0.318s
0.125	0.5000	0.283s

以上优化参数是将机器人当作一个理想刚体的情况下完成,忽略了关节连接处的摩擦下完成。为了建立更一般的机器人动力学优化模型,本文还建立了相应的在有关节摩擦情况下的优化方程。路径及机器人参数跟上面优化相同。在增加了摩擦力的模型后,虽然路径相同,但优化出来的结果却有相当大的差别。这主要是由于基于动力学的优化方程,对机器人模型的准确性要求高。如何得到机器

人准确的模型，是优化的关键。



a) 路径 1,  $a=0.125$ ,  $A_f=0.8367$ ,  $T=0.333s$



b) 路径 2,  $a=0.125$ ,  $A_f=0.5852$ ,  $T=0.267s$

图 3-12 带摩擦模型路径优化后的力矩输出

### 3.5 小结

本章对 Delta 机器人结构进行数学建模，分析了 Delta 机器人运动学正逆解，并推导了雅可比矩阵及非简化动力学数学模型，着重分析了动力学模型的建立。接着对机器人的控制算法进行仿真，并且在控制模型与实际模型有一定误差的情况下，控制算法依然可以很好的跟踪轨迹。最后，描述了拾放操作中的典型路径，并提出椭圆路径与改进正弦速度规划相结合的方式，并推导了基于机器人动力学最优时间算法。

## 第 4 章 平台搭建与软件设计

本章主要介绍具体的平台硬件与软件实现。硬件主要包括主要硬件模块，机器人电气系统设计。软件开发是根据整体设计要求和硬件限制开展的。软件实现包括机器人的初始化，控制算法的建立，上位机界面设计。

### 4.1 平台基本架构

#### 4.1.1 工业计算机与固高运动控制器

工业计算机（IPC）是为恶劣的工业环境设计的，有如下特点：抗干扰能力强、采用模块化设计技术、高可靠性、开放性好，吸收了 PC 机的全部功能，可直接运行 PC 机的各种应用软件。适合作为机器人平台的上位机使用，如图 4-1。



图 4-1 工业计算机

运动控制器采用 GUC-X00-TPV/TPG-ZX-L2，是固高科技 GUC 系列运动控制器的成员之一。OtoStudio 是用户应用软件开发环境，用于 GUC 的开发，是基于 IEC61131-3 工业控制语言标准的组态软件。GUC 的运行采用扫描的方式，扫描可以设定为固定周期、自由循环、事件触发、外部事件触发。用户可以根据需要，通过任务配置功能可快速的创建不同优先级的任务，创建多个线程。支持在线调试和仿真调试<sup>[50]</sup>。此符合控制平台的需求，如图 4-2。



图 4-2 固高运动控制器

## 4.1.2 控制电气图

机器人控制平台需要控制柜的支持，也就是电气设备，用于操作强电和伺服电机。电气设计主要包括主电路设计，驱动器电路设计，外部接口设计等。设计时需要注意电流大小，选择合适的电线；电磁兼容性，将主电路与信号线分开走线，防止干扰，做好控制线的电磁屏蔽。控制原理如图 4-4、图 4-3 所示。

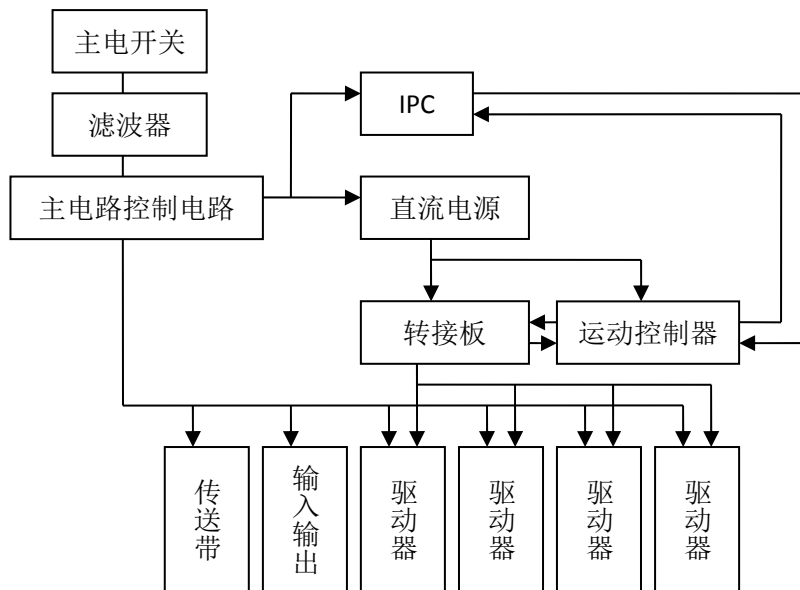


图 4-3 电气原理图

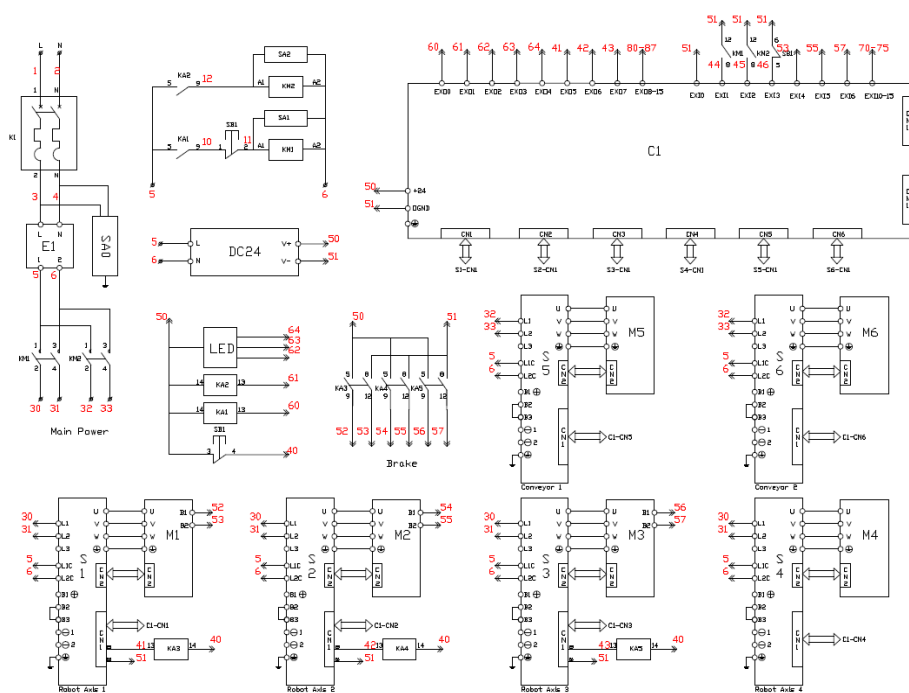


图 4-4 电气图



主电路通过断路器连接进入控制平台，经过电源滤波器，再通过接触器给伺服系统供电，控制开关有急停、总开关。电气柜实物如图 4-5 所示



a)控制柜



b)控制柜正面



c)控制柜背面

图 4-5 控制柜实物图

## 4.2 软件基本架构

在上述硬件平台的基础上，对软件进行设计实现。将非实时任务，例如人机界面，放在工业计算机上，将实时任务，例如控制算法，放在运动控制器中完成。

### 4.2.1 运动控制器流程图

运动控制器主要完成机器人控制算法及基本的轨迹规划。由于将运动控制器模式设置成开环，故不能借助运动控制卡原有的闭环调节功能，需要自己完成控制与规划。由于控制需要循环实现，故将系统配置成固定周期形式，间隔时间为

1ms。每次循环周期都要完成两个基本任务，一个是机器人的闭环控制，一个是规划机器人的运动。由于在上电时，系统处于未知状态，故需要回复到机器人原点位置，系统将原点定为[0, 0, 0.700]m。轴复位采用基本的电机复位方式，先将轴运动到下极限位置，再回到零点，再根据零点位置运行一个偏移值，就可以到达指定的机器人原点。由于控制都是在循环中完成，故需要维持系统的状态，PLC中通常使用功能顺序流程图来完成，但鉴于本文情况，为减少不必要的调用开销，全部过程都在一个程序中实现。根据功能，可以分为控制及规划，控制实现了两种，PID 控制及计算力矩控制，规划实现了初始化及正常路径规划。基本的系统状态转移如图 4-6 所示。

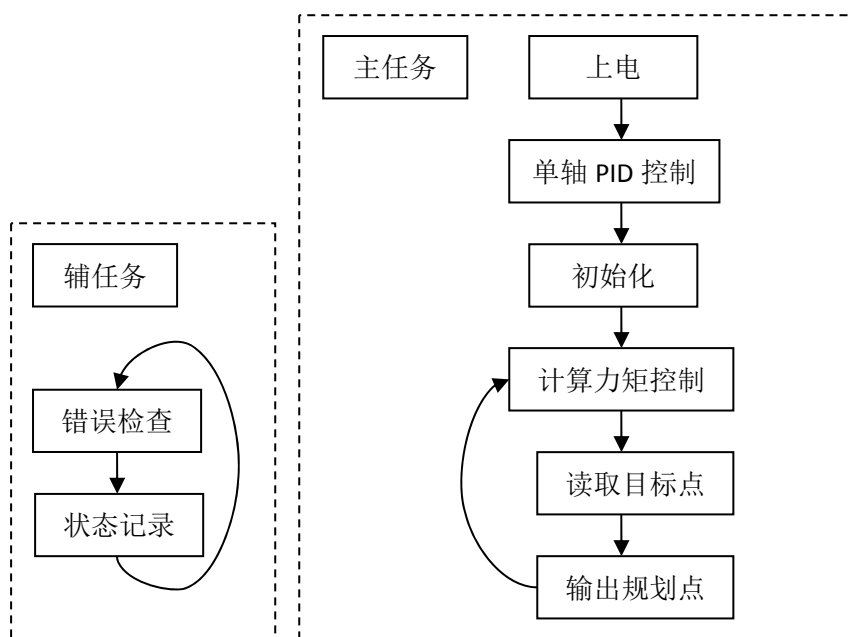


图 4-6 系统状态转移图

当系统上电时，进行未知状态，之后驱动上电，由于运动控制器是开环控制，需要在循环中做闭环。而机器人现在的状态是不清楚的，所以只能使用单轴 PID 控制方式，使各轴稳定下来之后，使用 Jog 模式，驱动电机寻找原点，在找到原点之后，将恢复系统初始状态，接着切换控制方式为计算力矩控制方式。之后进入正常循环方式，每次进入循环，先读取机器人当前状态，包括位置和速度，然后根据规划信息，计算此时的控制输出。之后进入规划，计算下一个运动点。

#### 4.2.2 工业计算机流程图

工业计算机主要完成人机交互，视觉识别系统的任务。其流程如图 4-7。由于系统是基于 Windows NT 的，界面采用 VC++，MFC 编程实现。其操作界面如图

4-8。操作界面上主要包括有系统启动、停止、手动、自动、回原点等方式。另外包括了视觉系统参数及图像处理参数的设置。

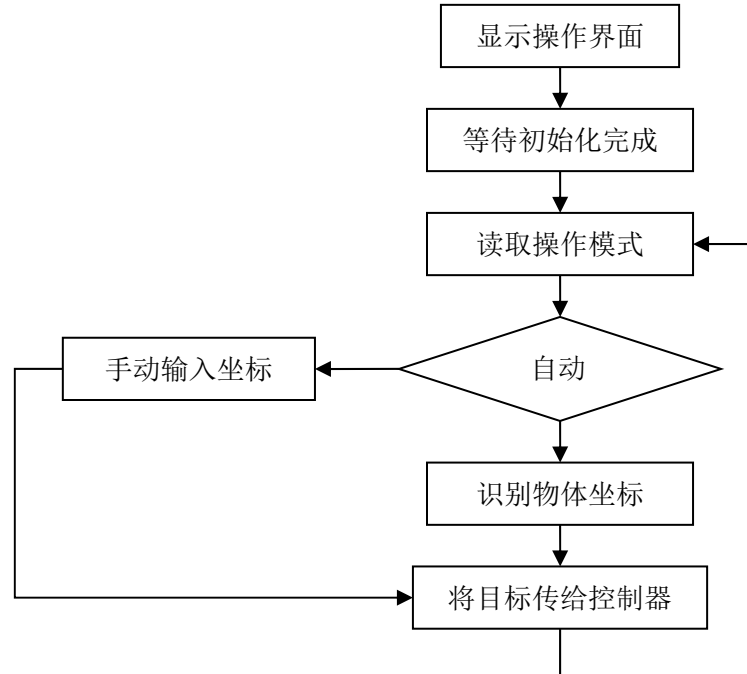


图 4-7 上位机流程图



图 4-8 操作界面

### 4.3 小结

本章主要介绍具体的平台硬件与软件实现。介绍了主要硬件模块、电气设计方案。软件实现上介绍了运动控制器的实现流程，与上位机的实现流程。在系统实现上，此方式可以较好的满足系统开放与实时性的需求。

## 第 5 章 实验与分析

本章就本文提出的非简化 Delta 机器人动力学模型、机器人计算力矩控制算法、路径规划及优化算法进行验证实验。借助于开放式机器人控制平台的特性，在机器人控制方面可以实现许多更适合于机器人控制的算法。

### 5.1 实验平台介绍

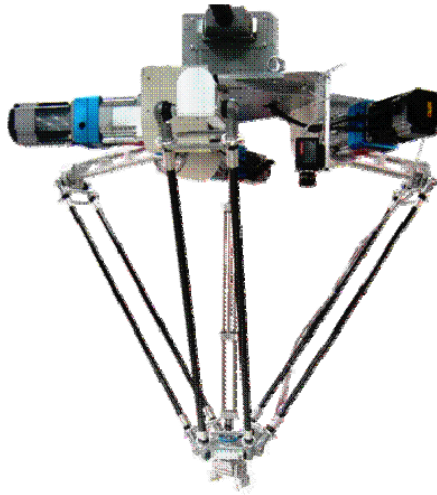


图 5-1 Delta 机器人

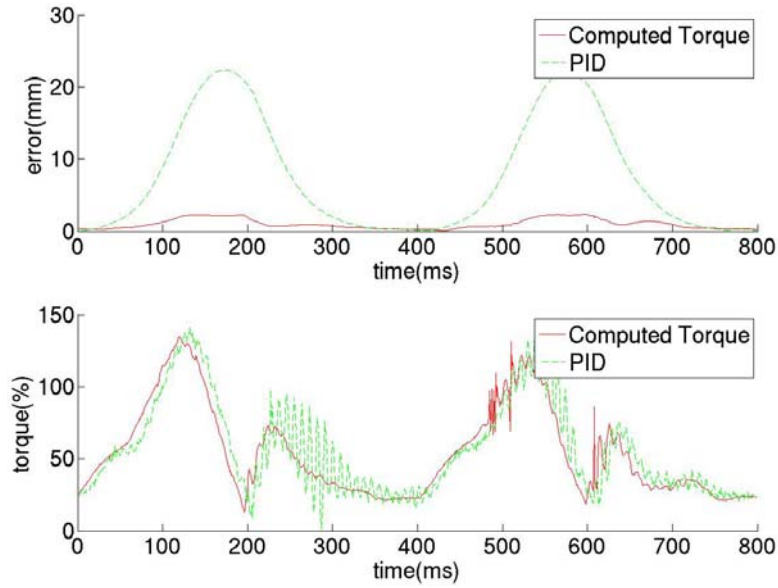
实验平台如图 5-1 所示，就控制系统而言，主要知道电机驱动器及电机参数即可。Delta 机器人使用的是三洋电机，使用 16 倍减速器。电机资料：厂商：SANYO；型号：R2AA08075F；相数：3；最大输入电压：200V；电流：4.6A；最大输出功率：750W；最高转速：6000r/min；编码器：增量式，131072P/R。表 5-1 为 Delta 机器人各关节经测量的值，其中电机惯量以厂家说明书上标的为准。

表 5-1 Delta 机器人测量参数

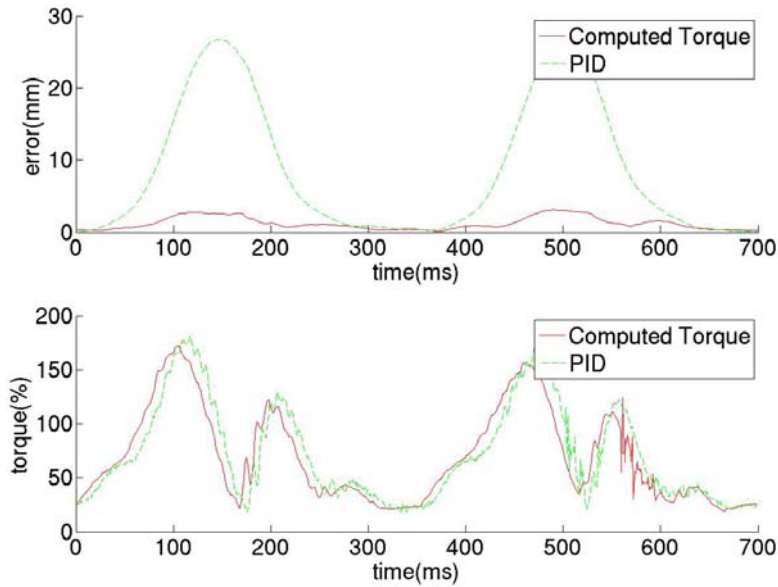
参数	机器人使用参数
定平台半径 $R$	0.200m
主动臂长度 $a$	0.347m
从动臂长度 $b$	0.806m
动平台半径 $r$	0.050m
电机惯量 $I_m$	$2.0733 \times 10^{-4} \text{ kg} \cdot \text{m}^2$
主动臂质量 $m_a$	0.80294 kg
肘关节质量 $m_e$	0.532165 kg
从动臂质量 $m_b$	0.42144 kg
动平台质量 $m_p$	1.1766 kg

## 5.2 控制方式对比实验

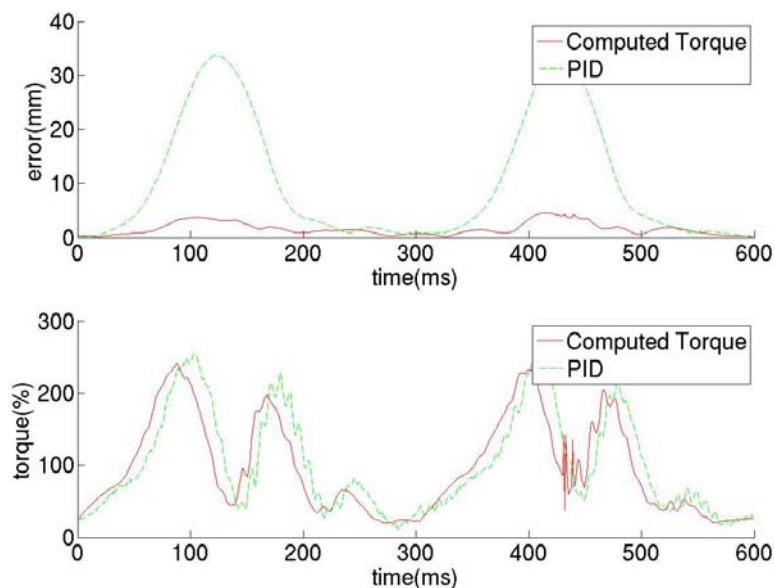
以下是普通的单轴 PID 控制方式与计算力矩控制方式的对比图。运动规划为椭圆路径加改进正弦速度规划，起点是 $[0.150, 0, 0.750]\text{m}$ ，终点为 $[-0.150, 0, 0.750]\text{m}$ ，高度为  $0.025\text{m}$ ，时间分别为  $0.35\text{s}$ ， $0.30\text{s}$ ， $0.25\text{s}$ ，规划参数为  $a=0.125$ ， $A_f=0.5$ ，在规划完成后，停止  $50\text{ms}$  后，进入下一个运动。由于 Delta 机器人有三个自由度及三个电机，所以用如下指标进行对比。误差为规划位置与实际位置的距离，力矩为三个电机出力矩百分比的平方和的开方。



a) 运动周期为 0.35s



b) 运动周期为 0.30s



c) 运动周期为 0.25s 时

图 5-2 相同轨迹下单轴 PID 控制与计算力矩控制跟踪误差与力矩图

采样的数据是从运动控制器中取得的，运动控制器本身的循环周期为 125us，而本文编写的控制算法的控制周期为 1ms，足够控制使用。控制时，计算力矩使用的是任务空间下的控制方程，PID 只能是关节空间的。计算力矩中 Delta 机器人模型参数使用的是测量值，PD 控制器的参数为  $P=8000$ ， $I=0$ ， $D=80000$ 。PID 控制中位置环参数为  $P=0.11$ ， $I=0$ ， $D=0$ ，速度环参数为  $P=35000$ ， $I=120$ ， $D=35000$ 。此参数是在同一条轨迹下进行调节的，并没有进行全局的试验。在调节 PID 时，分两次调节。速度环下以尽快稳定为标准。速度环调节完成后，对位置环进行调节，以机器人不震动为标准。计算力矩调节方式相同。

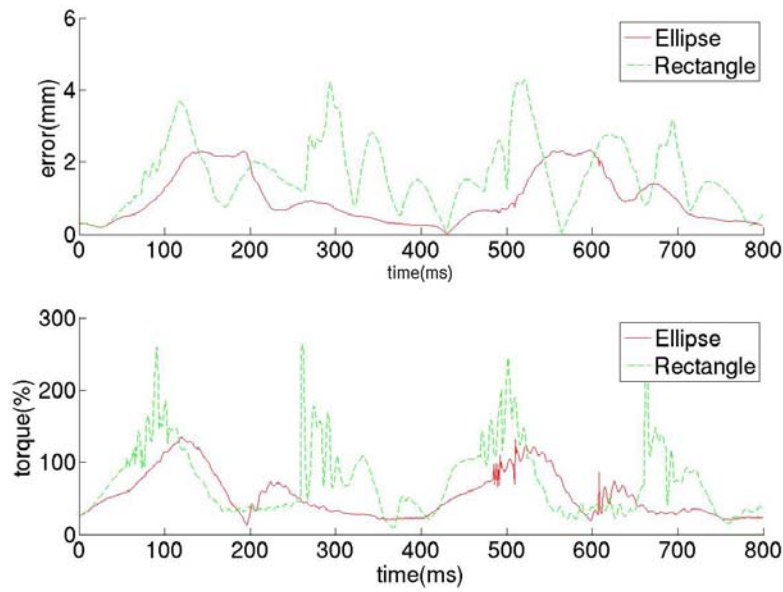
由图 5-2 中可知，计算力矩控制方式的误差比单轴 PID 的误差要小很多，同时也验证了非简化动力学模型的正确性。在相同运动时间下，计算力矩控制算法比单轴 PID 控制算法的跟踪误差要小 7~9 倍。随着运动时间的变短，相同控制方式下，跟踪误差都会有所增加，但相对来说，单轴 PID 控制算法误差增加比较多。表 5-2 为图中的数据值，计算力矩除了跟踪误差小之外，最大力矩也比单轴 PID 控制方式小。

表 5-2 控制性能对比表

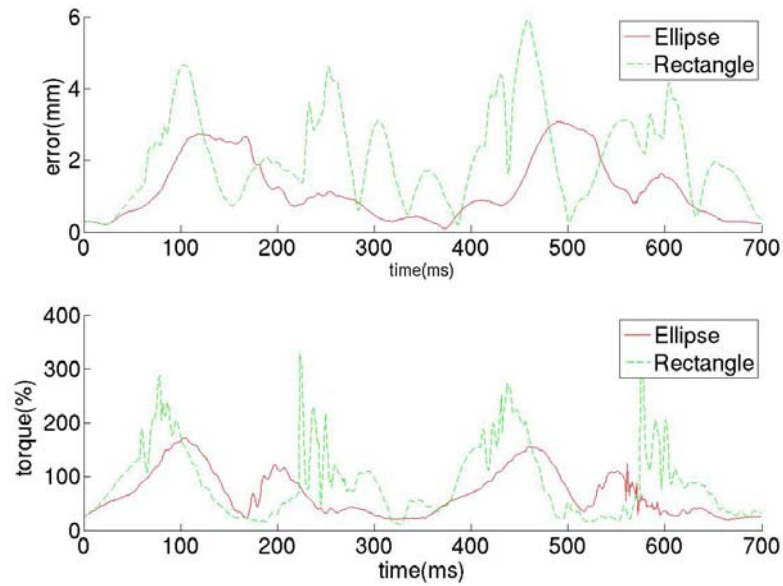
时间	单轴 PID 控制				计算力矩控制			
	跟踪误差		总力矩		跟踪误差		总力矩	
	最大	平均	最大	平均	最大	平均	最大	平均
0.35s	22.39mm	7.63mm	70.33%	28.07%	2.33mm	1.00mm	67.32%	27.17%
0.30s	26.89mm	8.72mm	90.48%	35.79%	3.08mm	1.21mm	86.15%	34.83%
0.25s	34.09mm	10.36mm	133.10%	50.57%	4.50mm	1.60mm	120.86%	48.31%

### 5.3 轨迹性能对比实验

Delta 机器人由于其高速性，特别适合于快速拾放操作。现在就两种轨迹进行对比实验，一种是矩形路径加 S 型速度规划，别一种是椭圆路径加改进正弦速度规划。起点是 $[0.150, 0, 0.750]$ m，终点为 $[-0.150, 0, 0.750]$ m，高度同为 0.025m，时间分别为 0.35s, 0.30s, 0.25s，椭圆路径规划参数为  $a=0.125$ ,  $A_r=0.5$ ，矩形规划中圆的半径为亮度的 1/2，在规划完成后，停止 50ms，进入下一个运动。控制方式使用计算力矩控制方式，参数同控制方式对比实验。

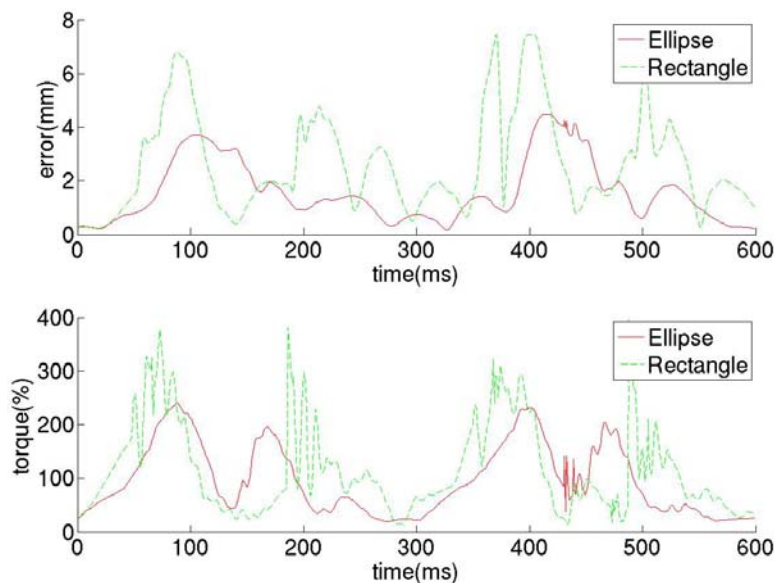


a) 运动周期为 0.35s



b) 运动周期为 0.30s





c) 运动周期为 0.25s

图 5-3 相同控制算法下椭圆轨迹与矩形轨迹跟踪误差与力矩图

由图 5-3 可知，由于椭圆路径及速度规划的连续性，系统力矩输出更平稳，输出力矩都不会突变，有利机器人的高速运行。就跟踪误差而说，矩形路径的误差值会比椭圆路径大 50%。而力矩方面，在运动当中，所需力矩的突变会导致系统震荡，增加系统的损耗及减少使用时间。表 5-3 为轨迹性能对比。从两者对比可以得出，机器人的运动规划对机器人的影响是很大的，特别是在力矩输出及稳定性方面。较好的运动规划会提高系统性能和使用时间。

表 5-3 轨迹性能对比表

时间	矩形运动规划				椭圆运动规划			
	跟踪误差		总力矩		跟踪误差		总力矩	
	最大	平均	最大	平均	最大	平均	最大	平均
0.35s	4.28mm	1.67mm	132.95%	37.12%	2.33mm	1.00mm	67.32%	27.17%
0.30s	5.92mm	2.12mm	164.75%	44.92%	3.08mm	1.21mm	86.15%	34.83%
0.25s	7.50mm	2.66mm	198.16%	58.01%	4.50mm	1.60mm	120.86%	48.31%

## 5.4 优化对比实验

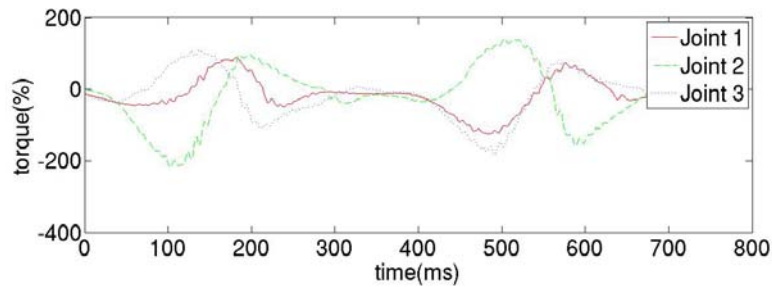
按照 1.5 倍额定力矩为输出极限。在计算路径的最优时间时，采用的是有摩擦的 Delta 机器人动力学模型。由于并没有建立有关摩擦的计算表达式，所以摩擦系数是估计值，通常多次测量，采用模型来估计系统值。以下是二个条路径的实验结果。由于摩擦是关于关节速度与方向的函数，虽是同一条路径，来回的轨迹参数都不一定会一样，需要对来回参数进行优化计算。其优化结果如表 5-4。对比参数  $a=0.125$ ， $A_f=0.5$ ，运动时间与优化时间相同。



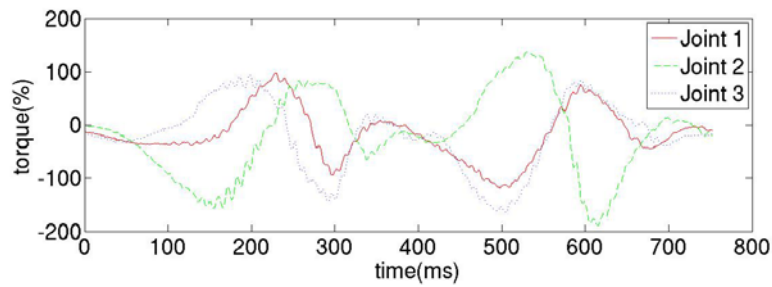
表 5-4 优化结果

路径	轨迹参数 $a$	轨迹参数 $A_f$	优化时间 $T$
1	0.125	0.8367	0.333s
	0.125	0.5992	0.320s
2	0.125	0.6617	0.305s
	0.125	0.7664	0.311s

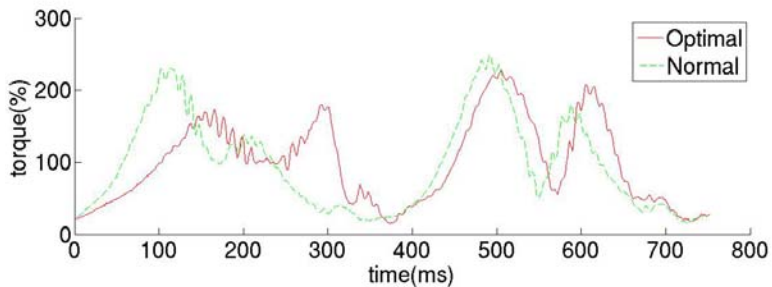
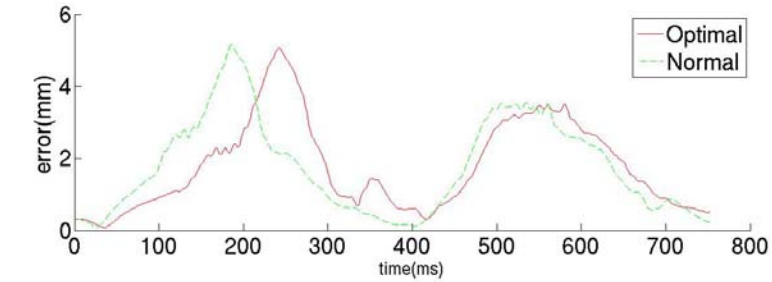
图 5-4 及图 5-5 是两条路径优化前与优化后的对比结果，分别为三个电机的输出力矩、跟踪力矩及总力矩的对比。



a) 优化前电机力矩大小

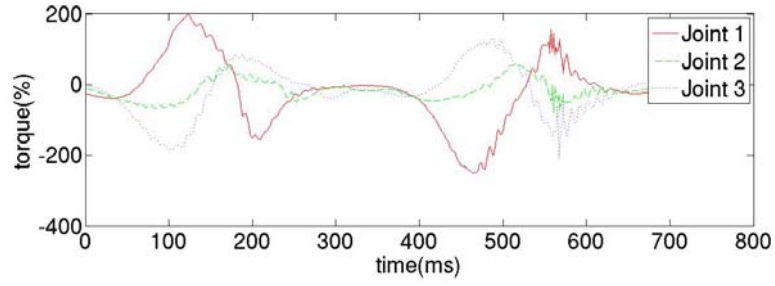


b) 优化后电机力矩大小

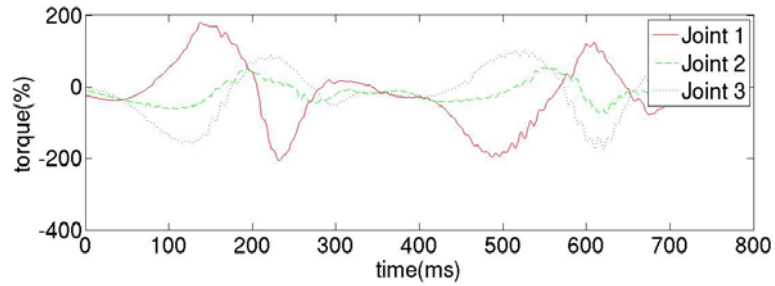


c) 误差及总输出力矩

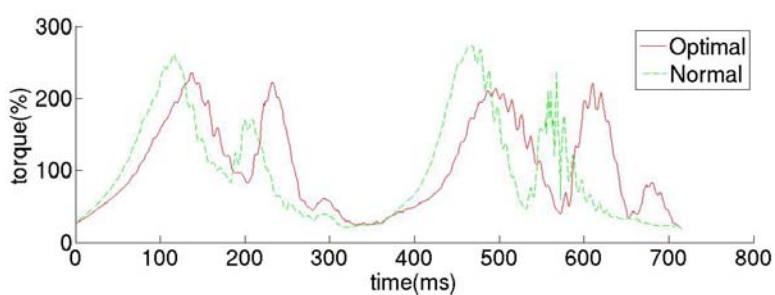
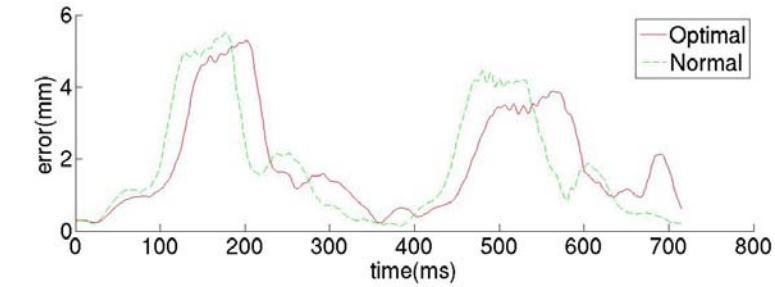
图 5-4 路径 1



a)优化前电机力矩大小



b)优化后电机力矩大小



c)误差及总输出力矩

图 5-5 路径 2

具体数据对比见表 5-5 和表 5-6, 从实验结果可以得出, 优化算法可以有效减少力矩最大输出值, 最大值的减幅达到 20%~30% 额定力矩, 也证明了 Delta 机器人数学模型是可用的。相同运动时间, 机器人所用的力矩更小, 也就是说机器人在相同力矩下可以得到更短的运动时间。

从中也可以知道, 由于基于机器人动力学, 其优化的实际效果取决于机器人建模的准确性上, 而且与建模参数也有很大关系。虽然在仿真时, 可以得到很理想的效果, 但实际上却并不如意, 特别是力矩限制这个方面。如何可以建立更准

确的模型及测量更准确的参数，是优化算法的关键。

表 5-5 优化实验对比表

路径	优化前				优化后			
	跟踪误差		总力矩		跟踪误差		总力矩	
	最大	平均	最大	平均	最大	平均	最大	平均
1	5.14mm	1.77mm	248.12%	96.89%	5.07mm	1.78mm	228.31%	99.88%
2	5.50mm	1.86mm	273.89%	100.13%	5.28mm	1.92mm	234.91%	105.61%

从表 5-5 可以得出，总力矩的大小减少了 30%额定力矩左右，而平均输出力矩和跟踪误差的最大值与平均值均没有较大的变化。表 5-6 表明，三个关节中最大输出力矩的最大值也减少了。

表 5-6 电机最大输出力矩对比

路 径	优化前力矩输出				优化后力矩输出			
	轴 1	轴 2	轴 3	最大值	轴 1	轴 2	轴 3	最大值
1	125.58%	218.15%	182.25%	218.15%	119.03%	188.61%	165.37%	188.61%
2	250.00%	78.83%	208.24%	250.00%	205.72%	74.77%	173.60%	205.72%

## 5.5 小结

本章在现有的 Delta 机器人的基础上，借助开放式控制平台，首先对普通 PID 与计算力矩控制两种算法进行对比实验，得出计算力矩控制比普通 PID 有更好的控制性能。接着，对比了矩形路径与椭圆路径的特点，得出椭圆路径更适合于快速的拾放操作。最后，从运动规划优化前与优化后的对比性能来看，优化后的规划可以有效减少机器人的最大输出力矩。

## 结 论

### 主要工作和创新点

机器人控制平台是机器人的关键技术，然而现有的商用控制平台是封闭的，不满足集成新传感器及控制算法的需求。本文在研究机器人控制平台的基础上，提出开放式控制平台的设计研究。主要的工作及创新点如下：

采用 PC 与符合 IEC61131 标准的运动控制器相结合的结构，提出了机器人控制平台的硬件架构与软件架构，并完成了控制平台的硬件与软件设计。

阐述了机器人控制原理，讨论了常用的 PID 控制算法及计算力矩控制算法的实现，使用 Matlab/Simulink 完成了计算力矩控制算法的仿真。

建立了 Delta 机器人非简化动力学模型，通过实验，验证了机器人模型的正确性。

研究了机器人基本路径规划，改进了并联机器人拾放操作的运动轨迹，并提出基于机器人动力学时间最优规划算法，以提高机器人的操作速度。

完成了机器人单轴 PID 控制算法及计算力矩控制算法对比实验，从实验数据可知，计算力矩拥有更好的控制性能。对比了两种拾放轨迹，改进后的轨迹有更好的性能。实验结果表明优化后的规划，可以有效减少机器人的最大输出力矩。

### 后续研究工作展望

(1) 本文中控制平台的硬件，虽然使用了模块方式设计，但在模块的连接采用硬连接方式，且接口不够标准，不利于改造。因此，在后续工作中首先做好硬件接口的标准化，提高开放性。

(2) 本文完成了运动控制卡及上位机的基本软件框架，但两者通信方面还没有完善。本文将进一步完成软件系统的设计，完成上位机与运动控制器的联调，完成 Delta 拾放演示系统。

(3) 多传感器可以提高机器人智能和增强环境的适应能力。本文由于平台的开放性，可以整合多传感器，例如力传感器。在后续工作中，将研究多传感器融合算法，进一步挖掘机器人能力。

(4) 机器人动力学模型的参数是通过测量得出的，但测量总是存在误差，而且由于数学模型是简化模型，并不精确。研究自适应控制算法，以弥补系统模型的不正确及测量误差。

## 参考文献

- [1] 李明. 机器人[M]. 上海: 上海科学技术出版社, 2012: 1-2
- [2] 熊有伦, 唐立新, 丁汉等. 机器人技术基础[M]. 湖北: 华中科技大学出版社, 1996: 2,4
- [3] 潘炼东. 开放式机器人控制器及相关技术研究[D]. 湖北: 华中科技大学博士研究生学位论文, 2007: 1-4
- [4] 范永, 覃民. 机器人控制器的现状及展望[J]. 机器人, 1999, 2(1): 75-80
- [5] 孙骥, 杨汝清. 开放式机器人控制器综述[J]. 机器人, 2001, 23(4): 374-378
- [6] Prischow G, Danirl C, Junghans G, et al. Open System Controllers-A Challenge for the Future of the Machine Tool Industry[J]. Annals of the CIRP, 1993, 42(1): 30-33
- [7] Cho Y J, You B J, Oh S R, et al. A compact/open network-based controller incorporating modular software architecture for a humanoid robot[J]. Journal of Intelligent and Robotic Systems, 1999, (25): 341-355
- [8] Greenfeld I, Hansen F B, Wright P K. Self-Sustaining Open-System Machine Tools[J]. Transaction of the 17th North American Manufacturing Research Institution, 1989, 17(3): 304-310
- [9] Bradley N D, Chles W Y, Robert J G. Open Architecture Controls for Precision Machine Tools[J]. Proceeding of Mechanism and Controls for Ultraprecision Motion, 1994, 4(3): 6-8.
- [10] Willian E F. What is an Open Architecture Robot Controller? [C]. IEEE International Symposium on Intelligent Control, 1994:27-32
- [11] Bruce M A, Cole J R, Holland R G. An Open Standard for Industrial Controllers[J]. Manufacturing Review, 1993, 6(3):180-191
- [12] Golden E H. Open Module Architecture Controllers(OMAC) [J]. Modern Machine Shop, 1995, 68(11): 160-162
- [13] Lutz P, Sperling W. OSACA-the vendor neutral Control Architecture[C]. Proceedings of the European Conference on Integration in Manufacturing, 1997 : 247-256
- [14] Chihiro S, Okano A. Open Controller Architecture OSEC-II : Architecture Overview and Prototype Systems[C]. Proceedings of IEEE 6th International Conference on Emerging Technologies and Factory Automation, 1997 : 543-550

- [15] Gianluca A, Setfano C, Valerio P, et al. A Modular and Task-oriented Architecture for Open Control System: the Evolution of C5G Open towards High Level Programming [C]. IEEE ICRA workshop on Innovative Robot Control Architectures for Demanding (Research) Applications. 2010 : 30-37
- [16] Gunter S, Andreas S, Rainer B. The Fast Research Interface for the KUKA Lightweight Robot[C]. IEEE ICRA workshop on Innovative Robot Control Architectures for Demanding (Research) Applications. 2010 : 15-21
- [17] Gamesz G J, Gomez O J, Sanchez G A, et al. Open Software Architecture for Advanced Control of Robotic Manipulators in Non-structured Environments[C]. IEEE ICRA workshop on Innovative Robot Control Architectures for Demanding (Research) Applications. 2010 : 7-14
- [18] 谈世哲. 工业机器人控制器开放性、实时性分析方法以及单处理器模式下的实现. 上海: 上海交通大学博士学位论文, 2002: 6-7
- [19] 安彦波. 工业机器人开放式控制系统的研究. 黑龙江: 哈尔滨工业大学硕士学位论文, 2010: 7-8
- [20] Theodorou E, Buchli J, Schaal S. Reinforcement Learning of Motor Skills in High Dimensions: A Path Integral Approach[C]. Anchorage, Alaska, USA: 2010 IEEE International Conference on Robotics and Automation, 2010: 2397-2403
- [21] 武传宇. 基于 PC+DSP 模式的开放式机器人控制系统及其应用研究. 浙江: 浙江大学博士学位论文, 2002: 27-29
- [22] Chung J W, Park III-Woo, Oh J H. On the Design and Development of a Quadraped Robot Platform[J]. Advanced Robotics, 2010, 24(1): 277-298
- [23] Petrosino A, Miralto M, Ferone A. A real-time streaming server in the RTLinux environment using VideoLanClient[J]. Journal of real-time image processing. 2011, 6(4): 247-256
- [24] 孙鲁毅. 四种流行的嵌入式实时操作系统的比较研究——VxWorks, QNX, ucLinux, RTEM[J]. 计算机应用与软件, 2007, 24(8): 196-197
- [25] IEC61131-3 Programmable controllers-Part3: Programming languages[S]. 1993
- [26] 彭瑜, 何衍庆. IEC 61131-3 编程语言及应用基础[M]. 北京: 机械工业出版社, 2009 : 1-10
- [27] 魏军. 基于 IEC61131-3 的控制软件平台的设计和实现[S]. 浙江: 浙江大学硕士学位论文, 2005: 77-79

- [28]Verscheure D, Demeulenaere B, Swevers J, et al. Time-optimal path tracking for robots: a convex optimization approach[J]. IEEE Trans. Automatic Control, 2009, 54(10) : 2318-2327
- [29]Verscheure D, Swevers J, Schutter J D, et al. “On-line time-optimal path tracking for robots”[C]. In Proc. IEEE International Conference on Robotics and Automation, 2009 : 599-605
- [30]Gasparetto A, Zanotto V. A new method for smooth trajectory planning of robot manipulators[J]. Mechanism and Machine Theory, 2007, 42(4) : 455-471
- [31]刘松国. 六自由度串联机器人运动优化与轨迹跟踪控制研究[S]. 浙江: 浙江大学博士学位论文, 2009: 40-41
- [32]ShiehP R, Lu Y S. Jerk-Constrained Time-Optimal Control of a Positioning Servo[C]. Int. Conf. Control, Automation and Systems, 2010 : 1473-1476
- [33]Alessandro G, Zanotto V. A technique for time-jerk optimal planning of robot trajectories[J]. Robotics and Computer-Integrated Manufacturing, 2008, 24(3) : 415-426
- [34]Croft E A, Constantinescu D. Smooth and time-optimal trajectory planning for industrial manipulators along specified path[J]. Journal of Robotic Systems, 2000, 17(5): 233-249
- [35]Bobrow J E, Dubowsky S, Gibson J S. Time-optimal control of robotic manipulators along specified paths[J]. The International Journal of Robotics Research, 1985, 4(3): 3-17
- [36]Slotine J J E, Yang H S. Improving the Efficiency of Time-Optimal Path-Following Algorithms[J]. IEEE Trans Robotics and Automation, 1985, 5(1) : 118-124
- [37]Xiao Y Q, Du Z J, Dong W. Smooth and near time-optimal trajectory planning of industrial robots for online applications[J]. Industrial Robot, 2012, 39 : 169-177
- [38]Verscheure D, Demeulenaere B, Swevers J, et al. Time-optimal path tracking for robots: a convex optimization approach[C]. IEEE Trans. Automatic Control, 2009, 54(10): 2318-2327
- [39]Moreno Masey R J, Gray J O, Dodd T J, et al. Elliptical point to point trajectory planning using electronic cam motion profiles for high speed industrial pick and place robots[J]. In Proc. Emerging Technologies and Factory Automation, 2009, 1-8
- [40]Astrom K J, Hagglund T. Advanced PID Control[M]. USA: Research Triangle Park, 2006: 65, 67, 68

- [41] 吴红鑫, 沈少萍. PID 控制的应用与理论依据[J]. 控制工程, 2003 (1): 37-38
- [42] 马骁, 杨志永, 王攀峰等. 高速并联机械手动力学建模及计算力矩控制. 机械设计, 2006, 23(2): 13-16
- [43] Murray R M, Li Z X, Sastry S S. A Mathematical Introduction to Robotic Manipulation[M]. Boca Raton: CRC Press, 1994: 246-249
- [44] Daniela Craita C C , Stefan S. Dynamics of Delta Parallel Robot with Prismatic Actuators[C]. IEEE International Conference on Mechatronics, 2005: 870-875
- [45] 丛爽, 尚伟伟. 并联机器人——建横、控制优化与应用. 北京: 电子工业出版社. 2010: 1-5
- [46] Hsu K S, Karkoub M, Tsai M C, et al. Modelling and index analysis of a Delta-type mechanism[J]. In Proc. Institution of Mechanical Engineer part k-journal of muti-body dynamics, 2004, 28(3): 121-132
- [47] Lopez M, Castillo E, Garcia G, et al. Delta robot: inverse, direct, and intermediate Jacobians[J]. Journal of Mechanical Engineering Science, 2006, 200(1): 103-109
- [48] Codourey A. Dynamic Modeling of Parallel Robots for Computed-Torque Control Implementation[J]. International Journal of Robotics Research, 1998, 17(12): 1325-1336
- [49] 王正林, 王胜开, 陈国顺, 王琪. MATLAB/Simulink 与控制系统仿真[M]. 北京: 电子工业出版社, 2007: 44-45
- [50] <http://www.googoltech.com.cn/webnew/index.php>



## 攻读学位期间发表的学术论文

- [1] Zhang Y Q, Huang R N, Lou Y J, et al. Dynamics based Time-Optimal Smooth Motion Planning for the Delta Robot[C]. IEEE International Conference on Robotics and Biomimetics, 2012: 1789-1794
- 、

本人郑重声明：此处所提交的学位论文《开放式并联机器人控制平台关键技术研究》，是本人在导师指导下，在哈尔滨工业大学攻读学位期间独立进行研究工作所取得的成果。据本人所知，论文中除已注明部分外不包含他人已发表或撰写过的研究成果。对本文的研究工作做出重要贡献的个人和集体，均已在文中以明确方式注明。本声明的法律结果将完全由本人承担。

## 学位论文使用授权说明

(1) 已获学位的研究生必须按学校规定提交学位论文；(2) 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；(3) 为教学和科研目的，学校可以将学位论文作为资料在图书馆及校园网上提供目录检索与阅览服务；(4) 根据相关要求，向国家图书馆报送学位论文。

## 致 谢

藉此论文脱稿之际，首先向恩师黄瑞宁副教授表示衷心的感谢。没有他耐心的、富有启发性的教导，不可能顺利完成我的硕士论文。在攻读硕士期间，黄老师给予了我很大的关心与帮助，在学习上，工作上，他为我解决了不少难题。。特别是本设计的进行，黄老师给予我指导，帮我把握住大方向，启发我思考研究的切入点和创新点。

我还要感谢楼云江副教授，在很长一段跟他相处的时间里，他不但将自己的知识倾囊相授，更是教会我对待科研工作的严谨态度，使我在完成研究的过程中逐渐变得更为注重每一个细节。

感谢我的师兄赵学超，是他带领我进入机器人的世界。

感谢廖斌，杨瑞鹏，刘丽丽，王立志，王燕伟，张松振，他们帮助我学习各种相关的知识，准备实验平台。其中廖斌，杨瑞鹏是我学习上的同学，生活上的好友，他们跟我一起共度了美好的校园时光。

感谢我的父母多年以来一直对我的关心与支持，感谢所有关心我，帮助我的亲人和朋友。