

浙江大学
硕士学位论文
机器人控制系统设计
姓名：徐益
申请学位级别：硕士
专业：控制理论与控制工程
指导教师：颜文俊
2003. 3. 1

## 摘要

对机器人控制系统结构的研究，是机器人学中的一个重要内容。本文以 Puma560 机器人的控制系统为例，提出了两个机器人控制系统设计方案：基于微机的设计和基于 Internet 的设计，在系统结构设计、技术方案选择、功能模块设定等各方面进行了详细的展开，并对相关领域的技术应用做了简要介绍。

基于微机的机器人控制系统给研究者、操作者提供了一个开放的、可扩展的平台，实现了机器人控制中的离线编程、图形仿真、轨迹规划等功能，提高了系统的软、硬件扩展能力和计算能力。其设计涵盖了机器人学的很多方面，并涉及到计算机、图形学等诸多领域。本设计方案提出的控制平台，开发周期短，模块独立，功能全面，是一个很好的研究平台，也是一个简便的直观的操作平台，移植性强。

基于 Internet 的机器人控制系统是基于微机系统的进一步发展，其实现采用了较新的网络技术。本设计方案提出的控制平台，在系统功能上侧重于机器人教学的普及和实验室设备的校园化操作，采用了分布式系统结构。方案设计注重模块化，引入了协同控制概念，并充分考虑到了网络控制形式所带来的系统特点，实现了较为完善的安全机制和实时的监控功能。

关键词：机器人控制系统结构；离线编程；仿真系统；机器人遥操作系统；Java

## **Abstract**

The study on the architecture of robot control system is an important part of robotics. A PC-based design and an Internet-based design for robot control system are reported in this paper. The key technique in these system designs is also introduced.

The PC-based robot control system in this paper, a system with functions of off-line programming, 3D graphic simulation and path planning, is an open, extendable system for researchers.

The Internet-based robot control system using advanced Internet techniques can be applied to the education of robotics and visitor can use the robot in the lab from distance. Compared with the PC-based control system, this distributed architecture system has the additional function of collaborative control. And as a net control system, there are some other differences in the form of robot control.

Both of these robot control system can be easily replanted and their calculate ability and software/hardware extendability are powerful. The example about Puma560 is given in this paper.

**Keywords:** Architecture of Robot Control System, Off-line Programming System, Simulation System, Tele-robotic System, Java

# 第一章 绪论

机器人是一种自动化的机器，所不同的是这种机器具备一些与人或生物相似的智能能力，如感知能力、规划能力、动作能力和协同能力，是一种具有高度灵活性的自动化机器。作为一个复杂的综合系统，机器人控制系统涉及到很多学科（如计算机、机械、传感器技术、人工智能等）。其中，机器人控制系统的结构设计具有重要的意义。随着机器人功能的日益复杂，如何综合考虑模块功能、控制性能要求，设计一个合乎要求的机器人控制系统，将是一项十分有意义的工作。

## 1.1 体系结构的开放性要求

### 1.1.1 机器人控制体系

研究机器人控制系统，需要我们对机器人控制体系有一个明确的认识。

机器人控制体系<sup>[1]</sup>是指控制机器人的软件和硬件结构，其研究主要集中在机器人控制器的设计和开发上，通常分为功能设计和结构设计。功能设计部分要完成控制功能和算法的定义，结构设计是完成功能在软件和硬件上的分布。

从控制器的结构来划分，可以分为开放式与封闭式<sup>[2]</sup>两种。

封闭式是指：基于自己的独立结构进行开发的，采用专用计算机、专用机器人语言、专用操作系统、专用微处理器。

自机器人诞生以来，特别是工业机器人，由于采用的控制器都是开发者独立开发的，结构为封闭式，在机器人的应用上受到了很多限制<sup>[3]</sup>，无论是硬件还是软件，都难以扩充和更改。目前大多数工业机器人都是封闭结构的。其典型代表 Puma560 工业机器人，采用 PDP-II 作为上层主控计算机，使用专用的机器人语言——VAL 语言，将控制算法固化在 EPROM 中。

在开放式控制器体系结构方面，国外的研究投入了大量的人力、物力，主要是集中在功能的划分、功能之间的信息交换规范以及实现功能的方法上。目前，仍处在“百家争鸣、百花齐放”阶段，还没有形成一个国际公认的技术标准规范和文件。

开放式机器人控制器<sup>[4]</sup>的概念主要包括：开发系统利用标准的开放性的计算机平台（为控制器软、硬件扩展创造条件），采用标准的操作系统和语言（改变

各专用机器人语言并存且互不兼容的情况),采用标准的总线结构(便于传感器、运动控制板的集成),有网络通讯功能(提高系统的应用范围并实现资源共享)。

虽然目前开放式机器人控制器还没有形成一个统一的标准规范,设计一个完全开放的标准机器人控制器存在着很多困难,但是其设计思想在机器人控制系统设计研究中,具有可贵指导意义。目前,工业上大量运用的机器人(包括机械手臂)是封闭式结构的,随着任务的复杂化和应用领域的不断扩大,其局限性日益显现。这就需要我们改进现有的封闭的机器人控制器,将其他领域(如图像处理等)的成果结合进来,从而使控制器具有一定的开放性。

### 1.1.2 开放型控制器模型

目前控制器体系结构方面的研究已经成为热点,出现了很多模型结构。

如:

- 概念模型 NASREM——遥控机器人控制系统结构的参考模型(美国国家航空和宇航局和国家标准局)<sup>[5]</sup>。它是第一个试图完全说明智能系统全部组件的模型。采用层次结构,每一层又分为三模块:任务分解、环境建模和传感信号处理,这种结构为处理大系统提供了一种标准的方法。

- 下一代通用控制器 NGC(Next Generation Controller—NGC)(Martin Marietta 公司)<sup>[6]</sup>。包括三个概念:一是集成化的结构;二是应用程序集;三是集成/配置环境,并且采用由术语——责任(Responsibilities)、要求(Requirement)(输入)和产品(Products)(输出)表示的模块结构来代替应用程序。

- 统一遥控机器人结构计划(UTAP)<sup>[4]</sup>。主要目标是开发维护遥操作机器人。在系统操作方面包括三个单元:对象建模模块、任务描述模块、任务执行模块。硬件结构分为两部分:本地部分和远地部分。同时,也对模型通信方面的实际命令、数据类型和结构进行了比较分析,其目的是将所需的内部数据交换量最小化。

- 基于三层控制结构<sup>[7]</sup>(George Sardis 教授提出)。组织层(Organization Level)用于抽象推理、任务规划和决策。协调层(Coordination Level)由调度员调度,向组织层提供接口。执行层由传感器和运动控制硬件组成,它与上面的协调器形成一对一的相互作用。整个结构是根据“智能降低、精度增加”的原理设计的。

●基于端口的对象的动态重新配置的软件结构框架（Carnegie Mellon 大学 (CMU)发展中心）<sup>[8]、[9]</sup>。将面向对象的设计同数字控制系统的端口设计相结合。一个基于端口的对象被定义为具有通信功能端口的对象。每一个端口对象的输入/输出表示一个状态变量串。对其他对象而言，隐藏了内部的状态和方法。该结构框架设置了一个全局状态信息库，所有对象端口变量的状态以表格的形式存放在该库中。

借鉴开放式机器人控制器的设计思想，机器人控制系统设计中，可以从用户角度划分控制系统，分为用户级、系统级和伺服级。

以机械臂的控制系统为例。

任务级编程位于用户级的最高层，在离线编程系统之上实现，可以通过离线编程工具直接访问。

系统级负责机器人专用库、I/O 设备和传感器驱动的实现。分成两层：较高的一层用于实现专用机器人库、机器人编程工具和机器人编程语言，叫做执行层，较低的一层用于特定应用运动控制，叫做应用层。

伺服级从控制的角度划分为运动控制层、手臂控制层和电机控制层。运动控制层协调并指挥手臂控制单元和外部扩展轴的电机控制，手臂控制层封装了机器人运动的高级反馈控制功能。

根据这分层控制的思路，可以设计出具有扩展功能和一定通用性的机器人控制系统，主要的工作内容在于层次的划分，功能的归类和模块间的信息规范。

在机器人控制系统设计中实现开放性，还可以解决目前存在的计算算法处理问题。

传统系统中的串行计算的计算方式，有着计算负担重，实时性差的弱点。在机器人控制系统设计中，必须充分考虑系统的数据处理能力和处理速度。这方面除了在主处理机上采用并行处理芯片外，还可以从系统组成结构着手，采用分布式网络控制，也可以通过扩展系统，增加开放性，分散计算任务来实现系统性能的改善。

## 1.2 关于离线编程系统

离线编程系统<sup>[10]、[11]</sup>，是机器人控制系统在功能上的开放性拓展，可以改进



机器人控制系统的运行效能。

机器人离线编程是指运用计算机图形学，建立机器人及其环境的几何模型，采用机器人编程语言进行编程，通过语言处理模块生成代码，然后对结果进行三维图形动画仿真以检验编程的正确性。

### 1.2.1 离线编程系统的组成

机器人离线编程系统是一个综合性的系统，应具有交互式的机器人系统 CAD 建模环境、任务规划和调试环境，运动学方程计算，碰撞检测等。

它的主要优点有：减少机器人不工作时间，通过仿真调试可以直观地看到编程结果；可以进行复杂运动轨迹规划，能够进行碰撞和干涉检验；改善编程环境等。

图形仿真是其中必不可少的组成部分，也是离线编程系统实现的难点问题。一般的解决方法是利用现有的先进几何造型系统，在此基础上进行离线系统的开发。这样的系统因为几何造型系统相对独立，显得非常庞大。采用几何造型模型库是一个比较实际的解决方法，随着计算机图形学的发展，建立一个三维几何实体模型库并完成作业任务的动画仿真已经可以在微机上实现了<sup>[12]</sup>。

### 1.2.2 基本问题

离线编程系统的研制和开发涉及的问题很多，包括多个领域的多个学科，主要有：

多媒体技术。实现友好的人机交互、直观的图形显示、生动的信息表达，都是机器人离线编程系统所必须的。

多传感器的融合技术建模与仿真。通过多传感器融合，可以补充系统信息，建立控制回环，提高离线编程系统的实用性。

各种规划算法的研究。如路径规划，抓取规划等，其中环境的复杂性，不确定性和运动性是这类问题的难点所在。

安全机制和修复技术。在系统运行中不可避免发生异常，如何建立健全的安全机制，防范错误的出现或及时正确地停止系统运行，恢复系统正常工作，在离线编程系统的开发中也是必须考虑的问题。

### 1.3 机器人控制系统的进一步发展

在机器人控制系统的开放性方面,无论是资源共享还是多机器人协同工作,都需要系统实现网络功能。机器人控制系统的网络功能,包括本系统内部的网络通讯和系统的网络扩展能力,也包括系统控制方式的网络化。

正如人类活动范围和探索空间的不断扩大是人类进步的标志一样,机器人的智能同样体现在运动空间的大小上。通过网络控制遥操作,可以扩大机器人运用范围,实现远程监控操作和危险污染环境作业;也可以普及机器人应用教学,在有限的硬件资源条件下,提供更多的操作实践,为一般机器人爱好者提供操作可能;还能构成一个实验环境进行协同操作,使不同地域的研究者可以共同讨论和检验控制方法。

传统的远程遥操作机器人技术硬件系统非常复杂,并且需要专用的数据通道,限制了技术的发展和应用<sup>[13]</sup>。Internet 为实现这一思想提供了一种低成本策略。近年来,随着 Internet 技术的发展,基于 WWW 的机器人遥操作系统开始出现。

1993 年 8 月的 Mercury 项目是第一个基于网络的遥操作机器人系统<sup>[14]</sup>,它允许使用者控制一台 IBM 机器人和 CCD 摄像机在充满沙子的工作空间中进行物品挖掘。1994 年 9 月,西澳大利亚大学的 Ken Taylor 等把一台 ABB 机器人联接到 Internet 上,该系统允许用户通过 web 浏览器控制机器人进行抓取和搬运<sup>[15]</sup>。1998 年 6 月, PumaPaint 被联接到 Internet 上,任何使用者都可以通过网络控制一台 PUMA760 机器人在画架上作画<sup>[16]</sup>。Paul G. Backes 等建造了 WITS 系统<sup>[17]</sup>,分布在美国各地的 NASA 的专家们使用该系统进行火星探路者的任务规划与控制,同时公众可以使用该系统进行虚拟操作。

基于 Internet 网络的机器人控制系统覆盖了机器人的主要技术领域,涉及到计算机、网络、通信、图像处理、传感器、自动控制等学科,是综合技术能力的体现。但是目前国内还没有这样的实际系统对外开放。

设计机器人网络控制系统,主要应该解决如下问题:

●控制系统的组成结构和功能模块的划分。

系统的结构构成直接关系到控制方式的选择,组建系统时,需要综合考虑到控制方式的可实现性和控制效果,同时,合理划分功能模块,建立标准的功能模



块接口，实现模块的可置换和辅助功能模块的独立性，以适应更多的应用领域。

●网络实现技术的选择。

目前技术领域飞速发展，零零总总各种技术层出不穷，单就 web 控制而言，无论是仿真图形技术还是网络编程技术都有很多技术方案可供选择。在系统设计中，需要跟踪新技术的发展，寻找更先进的解决方案，也要具体结合机器人网络控制系统的应用领域，根据网络教学，远程监控，远程操作等不同的应用要求和应用者的不同，各有侧重，选择合适的技术方案。

●遥操作技术的安全机制。

网络控制机器人系统由于应用者的普遍化和远程作业的特点，必须建立严格的安全机制，以防范意外发生。

●网络控制带来的自动控制问题。

网络控制带来了一些新的技术问题，如不确定时延、数据包丢失、状态预测与环境预测的仿真、基于任务的动态数据交换等。通过网络控制机器人这些智能设备不同于网络控制电灯，需要有稳定的信号传输，确保控制性能，因此必须解决上述技术问题才能使实验系统走向实际应用。

#### 1.4 本文概要

本文主要着眼于机器人控制系统的设计。以 PUMA560 机器人的控制系统为例，提出了两个机器人控制系统设计方案：基于微机的设计和基于 Internet 的设计，并对他们进行了详细的阐述。

基于微机的机器人控制系统给研究者、操作者提供了一个开放的、可扩展的平台，实现了机器人控制中的离线编程、图形仿真、轨迹规划等功能，提高了系统的软、硬件扩展能力和计算能力。其设计涵盖了机器人学的很多方面，并涉及到计算机、图形学等诸多领域。本设计方案提出的控制平台，开发周期短，模块独立，功能全面，是一个很好的研究平台，也是一个简便的直观的操作平台，移植性强。

基于 Internet 的机器人控制系统是基于微机系统的进一步发展，其实现采用了较新的网络技术。本设计方案提出的控制平台，在系统功能上侧重于机器人教学的普及和实验室设备的校园化操作，采用了分布式系统结构。方案设计注重模

块化，引入了协同控制概念，并充分考虑到了网络控制形式所带来的系统特点，实现了较为完善的安全机制和实时的监控功能。

本文的二~四章对机器人控制系统设计中涉及的机器人学理论进行了简要的介绍，为后继各章提供了基础。

第二章介绍了机器人建模中的运动学建模、几何建模，以 PUMA560 为例，介绍了控制系统中机器人坐标系的建立及变换，并对其中的逆运动变换问题进行了详细介绍。

第三章介绍了机器人运动轨迹规划的基本概念、内容和一些基本算法，为帮助理解，提供了一个详细的算法解释。

第四章则介绍了三维图形仿真在机器人控制系统中的应用，介绍了 PC 机和网络环境下图形仿真的实现，并从机器人控制系统的功能实现角度出发，分析比较了具体的实现方法并作了实验对比。

本文的第五章、第六章分别介绍了基于微机的机器人控制系统设计和基于 Internet 的机器人控制系统设计，在系统结构设计、技术方案选择、功能模块设定等各方面进行了详细的展开。

## 第二章 机器人建模和运动学运算

### 2.1 运动学建模

在进行机器人的运动学分析时，可以对机器人进行某种抽象，将连杆抽象为直线，将关节抽象为点，对分析结果不会产生影响。这是机器人的运动学建模。

运动学建模中，坐标系的选择有很多方法，最常见的 Denavit-Hartenberg 方法（D-H 方法）<sup>[18]</sup>设定，Z 轴为关节的旋转轴，X 轴为本连杆与前一连杆的公垂线，Y 轴根据右手法则来确定。两连杆间的关系由

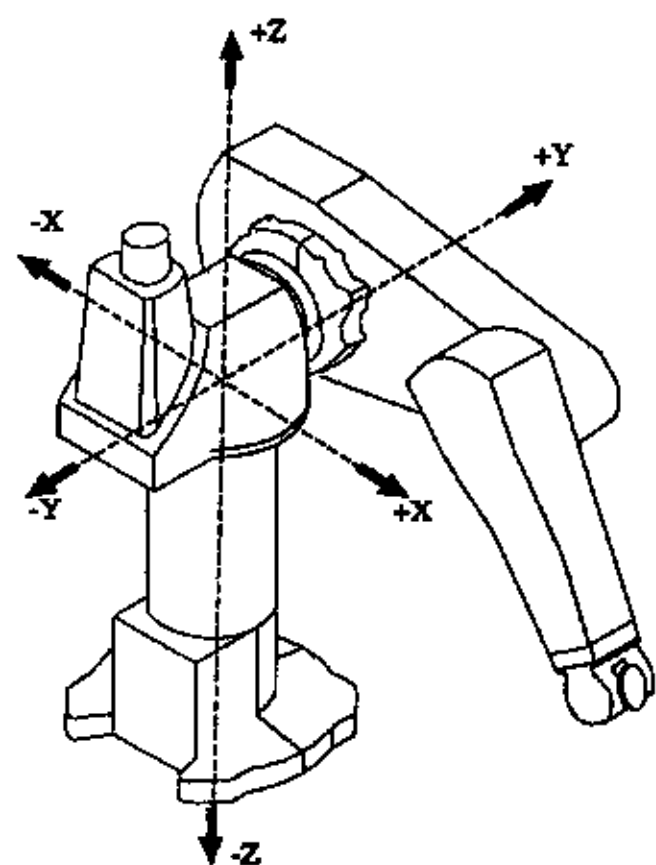


图 1 Puma560 结构简图

四个参量来描述： $a_i$ 、 $\alpha_i$ 、 $d_i$ 和 $\theta_i$ 。其中 $a_i$ 和 $\alpha_i$ 是描述连杆本身结构，而 $d_i$ 和 $\theta_i$ 是描述两相邻连杆的连接关系。对于转动关节， $\theta_i$ 为关节变量，对于移动关节， $d_i$ 是关节变量。

PUMA560 机械手臂的结构简图如图 1 所示。

结合 PUMA560 机械手臂，以各关节杠杆运动（旋转）中心为原点，旋转矩阵  $Rot(n, \theta)$  和平移矩阵  $Trans(n, d)$ ，其中  $n$  为相对旋转（平移）轴， $\theta$  为旋转角度， $d$  为平移距离。对于六个关节，根据坐标变换原理可得：

$${}^0_1T = Rot(z, \theta_1)Trans(z, d_1)$$

$${}^1_2T = Rot(y, \theta_2)Trans(y, d_2)$$

$${}^2_3T = Rot(y, \theta_3)Trans(x, d_3)Trans(y, d_4)$$

.....

同理得其余变换矩阵  ${}^3_4T$ 、 ${}^4_5T$ 、 ${}^5_6T$ ，则可得相对位形矩阵：

$${}^0_2T = {}^0_1T {}^1_2T$$

.....

$${}^0T_6 = {}^0T_1 {}^1T_2 {}^2T_3 {}^3T_4 {}^4T_5 {}^5T_6 = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

## 2.2 几何建模

在进行三维仿真显示的时候，要使机器人部件尽可能和真实情况逼近，达到仿真的目的，这是机器人的几何建模。运动学建模和几何建模是密切联系的，共同保存了机器人的空间位置信息和连杆组合信息。

表 1		
	部件名称	参数
机器人模型	底座(Base)	结构参数、坐标参数
	肩部(Shoulder)	.....
	大臂(Upper Arm)	.....
	小臂(Forearm)	.....
	腕部(Wrist)	.....
	法兰盘(Frange)	.....

机器人的几何模型的建立是根据机器人实际部件的装配关系建立的，是一种粗略的三维实体再现，为了实现机器人运动模拟，可采用表 1 的机器人模型结构。以每个运动单元为独立对象，主要参数有：结构参数、坐标变换参数。结构参数包括：部件尺寸、材质参数。坐标参数包括：旋转坐标，平移坐标。

几何模型坐标系和运动学模型坐标系的关系：在几何模型的绘制中，因为其部件往往具有外形对称性，绘制原点一般在部件的几何中心点上，而运动坐标系原点位于部件旋转中心，对于球体或环形关节，两个中心一致的，但对大部分杠杆部件而言不然。为了便于几何模型的运动控制，应该在绘制结束后，将几何坐标原点移动到旋转中心。

## 2.3 坐标系的转换

坐标系转换运算是机器人的运动控制的基础，坐标系的建立参考前述运动学建模。机器人坐标分为世界坐标系和关节坐标，PUMA560 机器人的坐标表达式为：世界坐标—— $(x, y, z, \alpha, \beta, \gamma)$ ，关节坐标—— $(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6)$ 。

其参数 $\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6$ 为各关节相对转动角度。 $x, y, z$ 为世界坐标,  $\alpha, \beta, \gamma$ 为位姿参数, 其几何定义如图2所示。其中 $\alpha$ 为终端坐标Z轴在世界坐标XY平面上的投影与世界坐标X轴的夹角,  $\beta$ 为终端坐标Z轴与世界坐标XY平面的

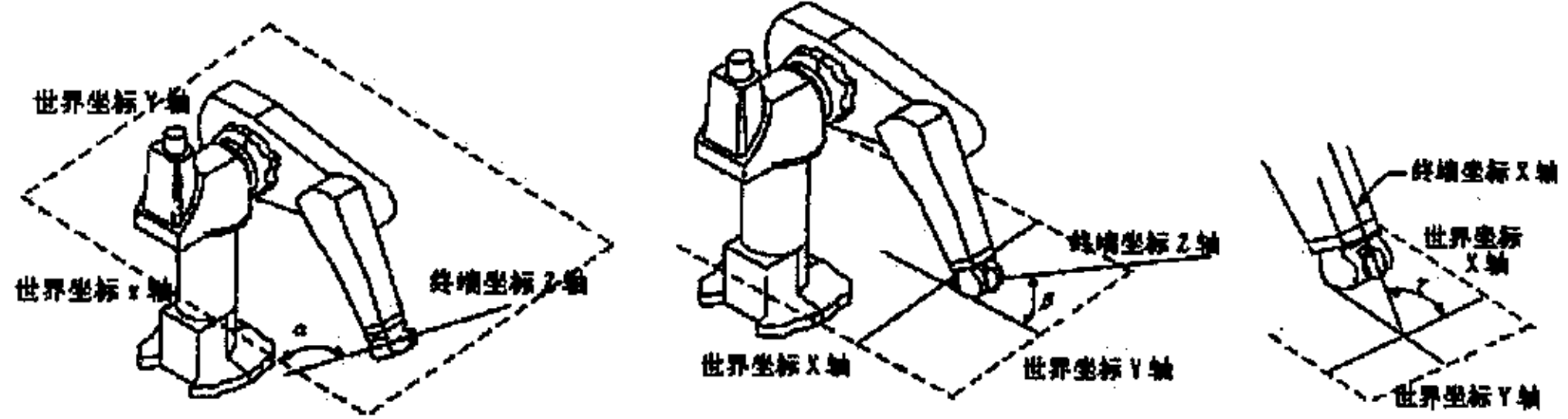


图2 位姿参数几何定义

夹角,  $\gamma$ 为终端坐标X轴和世界坐标XY平面的夹角, 这三个参数共同确定终端姿态。

机器人运动目标点和外部传感器提供的坐标信息往往是世界坐标 $x, y, z$ , 机器人运动控制往往需要关节坐标, 因此, 必须完成世界坐标和关节坐标的相互转换。我们称关节坐标到世界坐标的变换为正变换, 反之, 为逆变换。

将 ${}^{i-1}T_i$ 记为 $A_i$  ( $i=1,2,\dots,6$ ), 由前面推导可知:

$$A_i = \begin{bmatrix} \cos\theta_i & -\sin\theta_i \cos\alpha_i & \sin\theta_i \sin\alpha_i & a_i \cos\theta_i \\ \sin\theta_i & \cos\theta_i \cos\alpha_i & -\cos\theta_i \sin\alpha_i & a_i \sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\text{记 } {}^0_6T = A_1 A_2 A_3 A_4 A_5 A_6 = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} Q & R \\ O^T & 1 \end{bmatrix}。$$

PUMA560 机器人的连杆参数如表2所示。

表2 Puma560 连杆参数表						
关节 $i$	$\alpha_i(^{\circ})$	$a_i(mm)$	$d_i(mm)$	$\cos\alpha_i$	$\sin\alpha_i$	关节变化范围( $^{\circ}$ )

1	-90	0	0	0	-1	-160~160
2	0	431.8	149.09	1	0	-225~45
3	90	-20.32	0	0	-1	-45~225
4	-90	0	433.07	0	1	110~170
5	90	0	0	0	-1	-100~100
6	0	0	56.25	1	0	-266~266

得到各连杆间的变换矩阵为：（参考前述  $A_i$ ）

$$\begin{aligned}
 A_1 &= \begin{bmatrix} c_1 & 0 & -s_1 & 0 \\ s_1 & 0 & c_1 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, A_2 = \begin{bmatrix} c_2 & -s_2 & 0 & a_2 c_2 \\ s_2 & c_2 & 0 & a_2 s_2 \\ 0 & 0 & 1 & d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}, A_3 = \begin{bmatrix} c_3 & 0 & -s_3 & a_3 c_3 \\ s_3 & 0 & c_3 & a_3 s_3 \\ 0 & -1 & 0 & d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
 A_4 &= \begin{bmatrix} c_4 & 0 & s_4 & 0 \\ s_4 & 0 & -c_4 & 0 \\ 0 & 1 & 0 & d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix}, A_5 = \begin{bmatrix} c_5 & 0 & -s_5 & 0 \\ s_5 & 0 & c_5 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, A_6 = \begin{bmatrix} c_6 & -s_6 & 0 & 0 \\ s_6 & c_6 & 0 & 0 \\ 0 & 0 & 1 & d_6 \\ 0 & 0 & 0 & 1 \end{bmatrix}
 \end{aligned}$$

（其中  $c_i = \cos \theta_i, s_i = \sin \theta_i$ ）

## 2.4 正变换

由前述可知，终端位形矩阵为：

$${}^0_6T = A_1 A_2 A_3 A_4 A_5 A_6 = {}^0_2T A_3 A_4 {}^4_6T = {}^0_3T {}^3_6T = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$n_x = c_1 c_{2+3} (c_4 c_5 c_6 - s_4 s_6) + s_1 (s_4 c_5 c_6 + c_4 s_6) - c_1 s_{2+3} s_5 c_6$$

$$n_y = s_1 c_{2+3} (c_4 c_5 c_6 - s_4 s_6) - c_1 (s_4 c_5 c_6 + c_4 s_6) - s_1 s_{2+3} s_5 c_6$$

$$n_z = -s_{2+3} (c_4 c_5 c_6 - s_4 s_6) - c_{2+3} s_5 c_6$$

$$o_x = -c_1 c_{2+3} (c_4 c_5 c_6 + s_4 s_6) - s_1 (s_4 c_5 c_6 - c_4 s_6) + c_1 s_{2+3} s_5 c_6$$

$$o_y = -s_1 c_{2+3} (c_4 c_5 c_6 - s_4 s_6) + c_1 (s_4 c_5 c_6 - c_4 s_6) + s_1 s_{2+3} s_5 c_6$$

$$o_z = s_{2+3} (c_4 c_5 c_6 + s_4 s_6) + c_{2+3} s_5 c_6$$

$$\alpha_x = -c_1 c_{2+3} c_4 s_5 - s_1 s_4 s_5 - c_1 s_{2+3} c_5$$

$$\alpha_y = -s_1 c_{2+3} c_4 s_5 + c_1 s_4 s_5 - s_1 s_{2+3} c_5$$

$$\alpha_z = s_{2+3} c_4 s_5 - c_{2+3} c_5$$



$$p_x = c_1 c_{2+3} a_3 - c_1 s_{3+2} (d_4 + c_5 d_6) - s_1 (d_2 + d_3) + c_1 c_2 a_2 - (s_1 s_4 s_5 + c_1 c_{2+3} c_4 s_5) d_6$$

$$p_y = s_1 c_{2+3} a_3 - s_1 s_{2+3} (d_4 + c_5 d_6) + c_1 (d_2 + d_3) + s_1 c_2 a_2 + (c_1 s_4 s_5 - s_1 c_{2+3} c_4 s_5) d_6$$

$$p_z = -c_{2+3} d_4 - a_3 s_{2+3} + (s_{2+3} c_4 s_5 - c_{2+3} c_5) d_6$$

其中,  $c_{2+3} = c_2 c_3 - s_2 s_3, s_{2+3} = s_2 c_3 + c_2 s_3$

则

$$\alpha = \arctg \frac{-\alpha_y}{\alpha_z}, \text{ or } \arctg \frac{-\alpha_y}{\alpha_z} + 180^\circ \text{ (根据范围合理性取舍)}$$

$$\beta = \arctg \frac{\alpha_x}{-\alpha_y \sin \alpha + \alpha_z \cos \alpha}$$

$$\gamma = \arctg \frac{n_y \cos \alpha + n_z \sin \alpha}{o_y \cos \alpha + o_z \sin \alpha}$$

$$x = p_x$$

$$y = p_y$$

$$z = p_z$$

根据上述方法, 由  $(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6)$  求  $(x, y, z, \alpha, \beta, \gamma)$  得解

在实验程序中采用上述方法, 可得如下结果:

$(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6)$	$(x, y, z, \alpha, \beta, \gamma)$
(35.661, -67.691, 140.940, 29.269, -3.695, 31.292)	(422.231, 484.274, 562.962, 123.74, -19.965, 61.167)
(0, -126.095, 135.571, 32.498, 39.09, -64.994)	(-166.671, 168.038, 817.453, 117.444, -42.688, -58.082)
(-38.287, -50.186, 116.983, 44.822, 23.688, 0)	(667.291, -316.630, 526.609, 68.248, -5.675, 40.621)

## 2.5 逆变换

### 2.5.1 逆变换的解的复杂性

由  $(x, y, z, \alpha, \beta, \gamma)$  求  $(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6)$  是一个很复杂的非线性问题。

$${}^0T(x, y, z, \alpha, \beta, \gamma) = A_1(\theta_1)A_2(\theta_2)A_3(\theta_3)A_4(\theta_4)A_5(\theta_5)A_6(\theta_6)$$

根据左右元素对应相等, 有 12 个方程(见前面等式  $n_{xyz}, o_{xyz}, \alpha_{xyz}, p_{xyz}$  的求解), 未知量六个, 其中独立方程 6 个, 属于非线性超越方程, 其解非常复杂, 求解过程中会遇到解是否存在以及解的多重性问题。

解的是否存在, 通俗一点的讲, 就是给定一个空间三维坐标  $(x, y, z)$ , 机械臂终端是否可以到达该点并保证一定的姿态  $(\alpha, \beta, \gamma)$ 。显然, 由于机械手臂受到的客观约束, 并不是空间的每一个都可以到达, 对于不可达的点, 其解不存在。对于可达的点, 如果姿态不符合给定的  $(\alpha, \beta, \gamma)$  要求, 其解也是无效的。

对于多自由度的机器人来说,  $(x, y, z, \alpha, \beta, \gamma)$  只给定了终端位姿要求, 对其他关节没有硬性约束, 因此存在着多解问题, 即对于位形空间的一个点, 可以有多组关节坐标空间的解相对应。必须综合考虑运动量、避障等因素选择合适的解。

### 2.5.2 研究现状

由于逆运动解的复杂性, 历来是研究者研究的重点, 提出了很多解决方法。主要有几何解法<sup>[19]、[20]</sup>, Paul 的解析法<sup>[21]、[22]</sup>, Dinesh Mancha 和 John F. Canny 的符号及数值方法<sup>[23]</sup>等, 在逆运动求解问题上, 各解决方法都采取了不同程度的近似处理以减少求逆运算。国内王奇志等人提出了一种新的推导方法, 以 Paul 的解析法为基础, 给出了当  $d_6 \neq 0$  时的解法并避免了大量矩阵逆乘。以这种方法为基础, 加入臂形标志, 董明晓等人的求逆方法<sup>[24]</sup>利用机器人终端的位置矢量和姿态转换矩阵分别求解前三个关节变量和后三个关节变量, 使计算得以简化。

### 2.5.3 方法详解

由前述求解正变换可知：

$${}^0_6T = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} \cos \beta \cos \gamma & -\cos \beta \sin \gamma & \sin \beta & x \\ \sin \alpha \sin \beta \cos \gamma + \cos \alpha \sin \gamma & -\sin \alpha \sin \beta \sin \gamma + \cos \alpha \cos \gamma & -\sin \alpha \cos \beta & y \\ \cos \alpha \sin \beta \cos \gamma + \sin \alpha \sin \gamma & \cos \alpha \sin \beta \sin \gamma + \sin \alpha \cos \gamma & \cos \alpha \cos \beta & z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

根据此式，已知  $(x, y, z, \alpha, \beta, \gamma)$ ，可求得  ${}^0_6T$  的参数值。

引入臂形参数：

$$arm \begin{cases} +1(\text{右臂}) \\ -1(\text{左臂}) \end{cases}, \quad elbow \begin{cases} +1(\text{高臂}) \\ -1(\text{低臂}) \end{cases}, \quad wrist \begin{cases} +1(\text{上垂腕}) \\ -1(\text{下垂腕}) \end{cases}$$

并设：末端位置主要受关节 1、2、3 影响，末端姿态主要受关节 4、5、6 影响，

记基础坐标系到关节 4、5、6 轴交点的矢量为  $p$ 。

$$p = p_0 - d_6 \cdot a = [p_x \quad p_y \quad p_z \quad 1]^T$$

$$\text{则由 } {}^0_6T = A_1 A_2 A_3 A_4 A_5 A_6 = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} Q & P \\ 0 & I \end{bmatrix}$$

得

$$A_4 A_5 A_6 = \begin{bmatrix} c_4 c_5 c_6 - s_4 s_6 & -c_4 c_5 s_6 - s_4 c_6 & c_4 s_5 & d_6 c_4 s_5 \\ s_4 c_5 c_6 + c_4 s_6 & -s_4 c_5 s_6 + c_4 c_6 & s_4 s_5 & d_6 s_4 s_5 \\ -s_5 c_6 & s_5 s_6 & c_5 & d_6 c_5 + d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$A_1 A_2 A_3 \cdot \begin{bmatrix} {}^3_6 p_x & {}^3_6 p_y & {}^3_6 p_z & 1 \end{bmatrix}^T = \begin{bmatrix} p_x & p_y & p_z & 1 \end{bmatrix}^T$$

$$\text{所以：} A_2 A_3 \cdot \begin{bmatrix} {}^3_6 p_x & {}^3_6 p_y & {}^3_6 p_z & 1 \end{bmatrix}^T = A_1^{-1} \begin{bmatrix} p_x & p_y & p_z & 1 \end{bmatrix}^T$$

当忽略关节 4、5、6 对位置的影响， $d_6 \approx 0$  时，此等式的左边

$$A_2 A_3 \cdot \begin{bmatrix} {}^3_6 p_x & {}^3_6 p_y & {}^3_6 p_z & 1 \end{bmatrix}^T \approx A_2 A_3 \cdot \begin{bmatrix} 0 & 0 & d_4 & 1 \end{bmatrix}^T$$

$$= \begin{bmatrix} d_4(c_2 s_3 + s_2 c_3) + a_3 c_2 c_3 - a_3 s_2 s_3 + a_2 c_2 \\ d_4(s_2 s_3 - c_2 c_3) + a_3 c_2 c_3 + a_3 c_2 s_3 + a_2 s_2 \\ d_2 \\ 1 \end{bmatrix}$$

此等式右边

$$A_1^{-1} \begin{bmatrix} p_x & p_y & p_z & 1 \end{bmatrix}^T = \begin{bmatrix} p_x c_1 + p_y s_1 & -p_z & -p_x s_1 + p_y c_1 & 1 \end{bmatrix}^T$$

①因为  $-p_x s_1 + p_y c_1 = d_2$

$$\text{所以: } \theta_1 = \tan^{-1} 2 \left( \frac{p_y}{p_x} \right) + \tan^{-1} 2 \left( \frac{d_2}{\text{arm} \cdot \sqrt{p_x^2 + p_y^2 - d_2^2}} \right) \quad (-\pi \leq \theta_1 \leq \pi)$$

②因为  $d_4(c_2 s_3 + s_2 c_3) + a_3 c_2 c_3 - a_3 s_2 s_3 + a_2 c_2 = p_x c_1 + p_y s_1$

$$\text{且 } d_4(s_2 s_3 - c_2 c_3) + a_3 c_2 c_3 + a_3 c_2 s_3 + a_2 s_2 = -p_z$$

所以消去  $s_3$ 、 $c_3$  (移项平方相加) 得

$$\theta_2 = \tan^{-1} 2 \frac{\sin \alpha \cos \beta + \text{arm} \cdot \text{elbow} \cdot \cos \alpha \sin \beta}{\cos \alpha \cos \beta - \text{arm} \cdot \text{elbow} \cdot \sin \alpha \sin \beta} \quad (-\pi \leq \theta_2 \leq \pi)$$

$$\text{其中, } \sin \alpha = \frac{p_z}{\sqrt{p_x^2 + p_y^2 + p_z^2 - d_2^2}}, \cos \alpha = \text{arm} \cdot \frac{\sqrt{p_x^2 + p_y^2 - d_2^2}}{\sqrt{p_x^2 + p_y^2 + p_z^2 - d_2^2}}$$

$$\cos \beta = \frac{p_x^2 + p_y^2 + p_z^2 - d_2^2 + a_2^2 - a_3^2 - d_4^2}{2a_2 \sqrt{p_x^2 + p_y^2 + p_z^2 - d_2^2}}, \sin \beta = \sqrt{1 - \cos^2 \beta}$$

③因为  $d_4(c_2 s_3 + s_2 c_3) + a_3 c_2 c_3 - a_3 s_2 s_3 + a_2 c_2 = p_x c_1 + p_y s_1$

且  $\theta_1$ 、 $\theta_2$  已经求得

$$\text{所以: } \theta_3 = \tan^{-1} 2 \frac{\sin(\delta - \gamma)}{\cos(\delta - \gamma)} \quad (-\pi \leq \theta_3 \leq \pi)$$

其中:

$$\cos \delta = \frac{a_2^2 + d_4^2 + a_3^2 - p_x^2 - p_y^2 - p_z^2 + d_2^2}{2a_2 \sqrt{d_4^2 + a_3^2}}, \sin \delta = \text{arm} \cdot \text{elbow} \cdot \sqrt{1 - \cos^2 \delta}$$

$$\gamma = \tan^{-1} \frac{d_4}{a_3}$$

④因为  $A_4 A_5 A_6 = (A_1 A_2 A_3)^{-1} {}^0 T = (A_1 A_2 A_3)^{-1} \begin{bmatrix} Q & P \\ 0 & I \end{bmatrix}$

所以得:  ${}^3 Q {}^4 Q {}^5 Q {}^6 Q = ({}^0 Q)^{-1} \cdot {}^0 Q = \begin{bmatrix} q_{11} & q_{12} & q_{13} \\ q_{21} & q_{22} & q_{23} \\ q_{31} & q_{32} & q_{33} \end{bmatrix}$

( $\theta_1$ 、 $\theta_2$ 、 $\theta_3$  已解,  ${}^0 T$  已知, 故  $q_{ij}$  可解)

注意, 此处用到一次求逆  $({}^0 Q)^{-1}$

又因为  ${}^4 Q {}^5 Q = \begin{bmatrix} c_5 c_6 & -c_5 s_6 & s_5 \\ s_5 c_6 & -s_5 s_6 & -c_5 \\ s_6 & c_6 & 0 \end{bmatrix}$  且  ${}^3 Q = \begin{bmatrix} c_4 & 0 & s_4 \\ s_4 & 0 & -c_4 \\ 0 & 1 & 1 \end{bmatrix}$

所以:  ${}^4 Q {}^5 Q = \begin{bmatrix} q_{11} c_4 + q_{21} s_4 & q_{12} c_4 + q_{22} s_4 & q_{13} c_4 + q_{23} s_4 \\ -q_{31} & -q_{32} & -q_{33} \\ -q_{11} s_4 + q_{21} c_4 & -q_{12} s_4 + q_{22} c_4 & -q_{13} s_4 + q_{23} c_4 \end{bmatrix}$

注意, 此处亦用到一次求逆  $({}^3 Q)^{-1}$

可得:  $\theta_6 = \tan^{-1} 2 \frac{q_{32}}{-q_{31}}$ ,  $\theta_5 = \tan^{-1} 2 \frac{-q_{31}}{c_6 q_{33}}$ ,  $\theta_4 = \tan^{-1} 2 \frac{q_{23}}{q_{13}}$  ( $-\pi \leq \theta_i \leq \pi$ )

至此, 由  $(x, y, z, \alpha, \beta, \gamma)$  求

$(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6)$  完毕。

在求解过程中, 关键的几点是:

- 1) 引入臂形参数
- 2) 假设前三关节影响位置, 后三关节影响姿态,  $d_6 \approx 0$

- 3) 计算过程中用到两次矩阵求逆

由前面的解可以看出,  $\theta_1, \theta_2, \theta_3$  解并不是唯一的。由于 arm、elbow、wrist 的引入, 有八组可能的解, 需要结合实际工作情况选择合适的解。

其 C 程序实现流程见图 3。

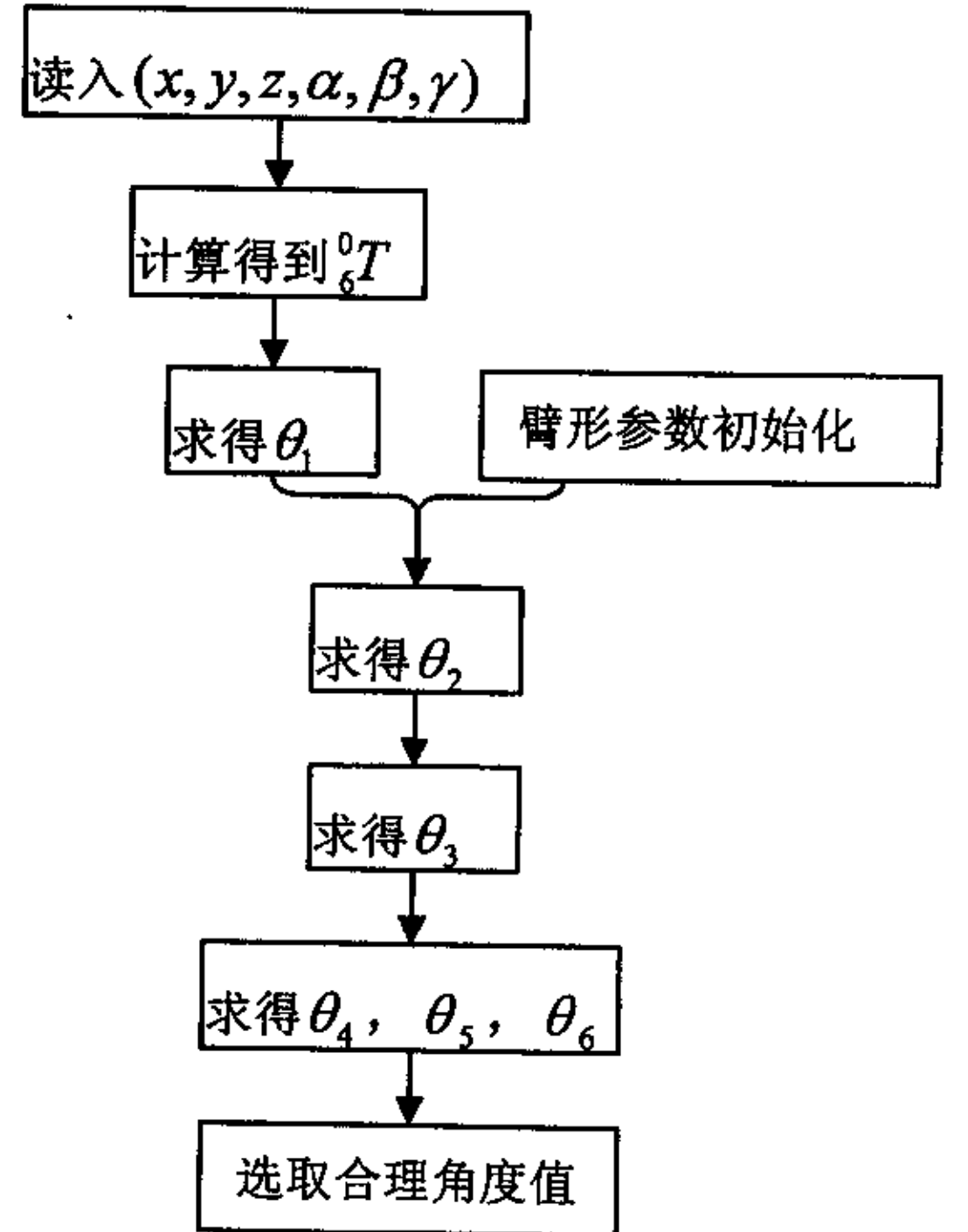


图 3 求逆解程序流程

机器人运动学运算也可以利用 Matlab<sup>[25]</sup>来完成。MATLAB 是 MathWorks 提供的一个科学运算工具软件,有强大的矩阵运算功能,且程序设计流程灵活,新版的 Matlab 还提供了高质量的图形可视化与界面设计,而且它有强大的语言接口功能,其他应用程序(如采用 C++语言编写的应用程序)可以很方便的调用其计算函数,引用其计算结果。

在机器人学的研究上,各个研究者编写了很多 Matlab 运算函数,其中特别值得注意的是 The Robotics Toolbox(release 7.0)<sup>[26]</sup>,这个工具箱提供了比较全面的机器人运动学计算函数,研究者可以结合自己的应用情况调用,也可以将自己用 Matlab 语言编写的函数加入工具箱,此外,研究者还可以利用 Matlab 的 Simulink 进行仿真数据模拟,并以图形化方式显示运动。

## 2.6 和传感器坐标系统的配合

在控制系统设计中,有传感器(视觉系统)作为外部信息提供者参与控制,视觉系统提供的物体空间信息是以其自身视点坐标为原点的,和机器人的世界坐标系不一致,因此需要进行两个坐标系的配合标定。

设视觉系统相对机器人静止固定,坐标原点  $O_v$ , 三个直角坐标轴为  $X_v$ 、 $Y_v$ 、 $Z_v$ , 机器人世界坐标原点

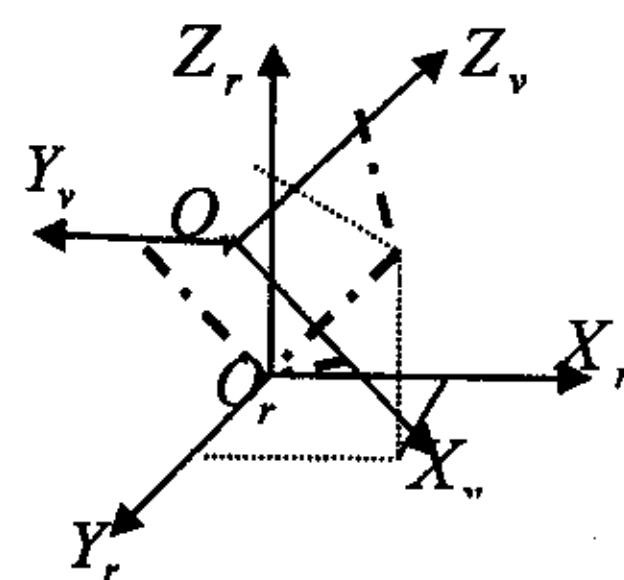


图 4 坐标示意图

$O_r$ , 三个直角坐标轴为  $X_r$ 、 $Y_r$ 、 $Z_r$ , 示意图见图 4。则可知,视觉坐标值  $x_v$ 、 $y_v$ 、 $z_v$  在机器人坐标系的数值  $x_r$ 、 $y_r$ 、 $z_r$  可由两个坐标系之间的旋转平移关系矩阵求得,且两者可相互转换。在视觉系统相对机器人静止固定时,坐标系配合标定的工作主要是矩阵运算。当视觉系统存在运动,则坐标配合标定变得较为复杂,不仅视觉系统本身的坐标辨认需要更多的运算处理,而且在两个系统配合时需要考虑运动变化的存在。

视觉系统和机器人的坐标系转换中,必然存在着数据误差,而视觉系统提供的外部空间信息本身,由于处理方法的局限和辨别能力等客观原因,肯定也存在着误差。在处理这些数据的时候,不仅需要进行坐标系的转换,而且需要引入误



差补偿，以改善控制效果。

## 第三章 机器人运动轨迹规划

### 3.1 轨迹规划介绍

机器人运动轨迹规划是指根据机器人操作任务需要(如目标抓取、避开障碍、目标跟踪等),规划出在机器人动力学约束、关节几何约束下的关节运动时间序列。它是机器人运动控制中的一个重要内容,高层规划包括感知机器人工作环境,任务描述以及监控等,底层规划包括路径规划,即寻求出机器人运动路径中的目标点(取道点)以及其对应的位形、速度、力矩参数的时间序列。

实现机器人运动轨迹规划,主要的工作有:

#### 1) 描述操作任务以及工作环境

操作任务不同,对运动轨迹有直接影响,比如焊接机器人往往得沿着某线段连续运动,而抓取物体只需要考虑到路径中的最终目标点和中间避障点,对于运动中间路径并无严格的任务约束。同样,任务的不同还影响到运动速度、运动平稳性等指标。而工作环境有静态环境与动态时变环境之分,在环境描述时必须加以考虑。

#### 2) 选择合适的算法,指定路径目标点和取道点

根据不同的任务和环境,研究者提出了很多规划算法用于机器人运动轨迹规划。在算法的选择上,还应该考虑到的是算法的复杂度和在控制系统中的可行性。

比如,在轨迹规划中,目标点和中间取道点是动态时变的还是静态的,需要选择不同的规划算法,前者需要实时修正能力和更多的处理外部信息,后者在算法上简单很多。

又比如,在轨迹规划中有关节变量空间轨迹规划和直角坐标空间轨迹规划之分。前者以关节坐标形式给定约束的,直接运用运动时受控变量来规划运动轨迹,方法简便、计算量小、可实时进行轨迹规划。后者以直角坐标形式给定约束的,而关节驱动器是在关节变量空间中受控的,所以必须把直角坐标系中的约束变换为关节坐标中的约束,然后再进行轨迹规划,有时这种变换是病态的(点不可达)。

进行轨迹规划时，应该考虑到在本系统中关节控制方便还是坐标控制可行，从中选取合适的形式，提高效率，减轻计算负担。

### 3) 计算并检验目标点对应的运动控制参数

主要涉及到前述正、逆运动学变换和机器人运动性能参数，应该剔除不实际的计算结果，选择可行或较优的目标点。

### 4) 对机器人运动进行监控（有些任务或系统还要求实时修正路径）

机器人运动轨迹规划得到路径轨迹和控制参数后，在运行过程中加入监控可以及时了解机器人实际运动轨迹，必要时加以修正，对于动态时变环境尤其重要。监控可以检验算法的正确性可行性，更重要的是，可以保障机器人的安全工作。

## 3.2 一些基本的算法

在机器人运动轨迹规划方面，研究者提出了很多规划算法。针对移动机器人，算法集中在避障、自主选择路径方面的问题，比如人工势场法<sup>[27]</sup>、虚力场法、局部路径规划、路径跟踪等。对机器手臂而言，主要是着眼于目标抓取、多关节手臂运动避障<sup>[28]</sup>，例如：

变比例轨迹规划法<sup>[29]</sup>：机械手在抓取目标时，机械手接近矢量与目标运动方向的夹角增大，以至于成钝角，甚至 $180^\circ$ 角。为了准确地抓取目标，一般抓取是在跟踪稳定之后，但如果目标相对于机械手的位置估计精度不够，就不能保证抓取能顺利完成，解决的办法一般有两个：一是研究在跟踪稳定后如何调整机械手姿态，使抓取操作顺利完成，但这会使跟踪过程延长，二是研究更精确地估计目标位姿，但视觉传感和机械手驱动的内在误差使这方面的研究收效不大。变比例轨迹规划法是解决这个问题的一個尝试。它使视角在机械手接近过程以匀速不断变化，这样可以通过改变视角来改变机器人的接近抓取过程，同时由于接近轨迹变成曲线，机械手接近矢量与目标运动方向夹角不断增加，改善了抓取操作性能。

逐段离散轨迹规划法<sup>[30]</sup>：由于目前机器人均采用计算机控制，要求在采样时间间隔上控制机器人，所以机器人运动控制在本质上是离散的，而不是连续的过程。离散轨迹规划法是近几年提出来，它将路径分成许多段，使目标函数达到

最优（如最小时间）的各间断点处的轨迹即为最优轨迹。考虑关节速度约束、力矩约束、关节冲力约束和工具端速度的约束，沿给定路径逐段规划出各点的关节、位置等时间序列。

在机器人通用控制器中，往往提供了一些基本的算法控制，以 PUMA560 机械手臂为例，在其运动控制中，采用了末端点~点直线运动的方法，可以在上层控制软件中进行一些高层算法控制。

### 3.3 基于行为的模糊运动规划

在机器人的运动控制规划上，基于行为的机器人控制已经得到了应用<sup>[31]、[32]</sup>。前期的工作多集中于二维空间的三自由度机械臂避障实验，目前研究者开始考虑三维空间中高自由度机器人的避障规划。其中，Palitha Dassanayake 等人提出的基于行为的模糊运动运动控制<sup>[33]</sup>是一个成功的方案，能很好的解决三维空间六自由度机器人在障碍物预知且静止情况下的避障运动规划问题。

机器人坐标系选择如前所述 D-H 法则，设目标点是一个球体，半径  $r_{\text{球}}$ ，位于  $x_t$ ；障碍物也是一个球体，半径  $r_{\text{障}}$ ，位于  $x_c$ ，考虑到安全及机器人手臂的厚度，将其半径扩展到  $R_{\text{障}}$ ；初始终端坐标  $x_e$ ，初始关节角度  $\theta_0$ ；上述数据均为已知；路径选择原则是：在可选路径中，终端运动路径最短者优先。

假设障碍物坐标为已知，且可得实时的关节角度坐标，将机器臂运动分为三个子集：

集 A，趋向目标点的运动，输入为终端到目标点的向量  $D = [D_x \ D_y \ D_z]^T$ ，终端到目标点的角度向量  $\varphi = [\varphi_x \ \varphi_y \ \varphi_z]^T$ ，输出是终端的路径设置点位姿

$$\rho_{ead} = [x_{ead} \ y_{ead} \ z_{ead} \ \alpha_{ead} \ \beta_{ead} \ \gamma_{ead}]^T。$$

集 B，在工作空间的无约束运动，输入为终端速度向量  $v = [v_x \ v_y \ v_z]^T$  和角速度向量  $\omega = [\omega_x \ \omega_y \ \omega_z]^T$ ，输出是终端的路径设置点位姿

$$\rho_{ebd} = [x_{ebd} \ y_{ebd} \ z_{ebd} \ \alpha_{ebd} \ \beta_{ebd} \ \gamma_{ebd}]^T。$$

集 C，避开障碍的运动，输入为所有关节到障碍物距离的最小值

$d = [d_x \ d_y \ d_z]^T$ , 障碍物到这一最近点的角度向量  $\psi = [\psi_x \ \psi_y \ \psi_z]^T$ , 输出是终端的路径设置点位姿  $\rho_{ecd} = [x_{ecd} \ y_{ecd} \ z_{ecd} \ \alpha_{ecd} \ \beta_{ecd} \ \gamma_{ecd}]^T$ 。

对  $\rho_{ead}$ 、 $\rho_{ebd}$ 、 $\rho_{ecd}$  运用模糊规则运算, 得终端路径设置点位姿  $\rho_{ed}$ 。示意图见图 5。

设 A、B、C 三个子集各制订 M 条规则, 且子集内分为 M 个量化等级, 对应的量化值为  $\omega_{bi}$ , 输入值  $y$ , 规则  $i$  的隶属度函数为  $\mu_i$ , 则可知它们各自输出的模糊控制量化值为

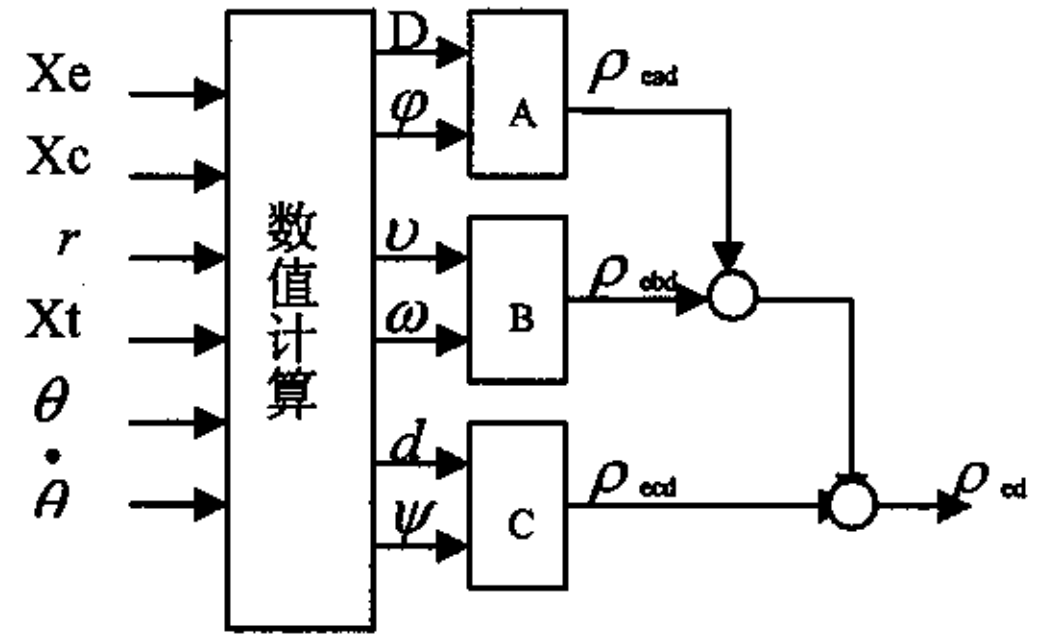


图 5

$$u = \frac{\sum_{i=1}^M \mu_i \omega_{bi}}{\sum_{i=1}^M \mu_i}, \text{ 其中 } \mu_i = e^{-\frac{(y-x_0)^2}{2\sigma^2}} \quad (x_0 \text{ 为数学期望, } \sigma \text{ 是均方差})$$

以 A 集为例, 分 9 个量化等级, 3 个模糊等级, 则有规则 9 条

D	将 D 值划分为 9 个值域								
$\varphi$	.....								
$\rho$	$\omega_{b1}$	$\omega_{b2}$	$\omega_{b3}$	...	...	...	...	...	$\omega_{b9}$
模糊\量化	L1	L2	L3	L4	L5	L6	L7	L8	L9
小	1	$e^{-\frac{(L2-L1)^2}{2\sigma^2}}$	...	...	0	...	...	...	0
中	0	...	...	$e^{-\frac{(L4-L5)^2}{2\sigma^2}}$	1	...	...	...	0
大	0	...	...	...	0	...	...	$e^{-\frac{(L8-L9)^2}{2\sigma^2}}$	1

规则:

1. 如果 D 小,  $\varphi$  小, 则  $\rho_x$  小,  $\rho_a$  小; 2. 如果 D 小,  $\varphi$  中, 则  $\rho_x$  小,  $\rho_a$  中;
3. 如果 D 小,  $\varphi$  大, 则  $\rho_x$  小,  $\rho_a$  大; 4. 如果 D 中,  $\varphi$  小, 则  $\rho_x$  中,  $\rho_a$  小;
5. 如果 D 中,  $\varphi$  中, 则  $\rho_x$  中,  $\rho_a$  中; 6. 如果 D 中,  $\varphi$  大, 则  $\rho_x$  中,  $\rho_a$  大;
7. 如果 D 大,  $\varphi$  小, 则  $\rho_x$  大,  $\rho_a$  小; 8. 如果 D 大,  $\varphi$  中, 则  $\rho_x$  大,  $\rho_a$  中;

9. 如果  $D$  大,  $\varphi$  大, 则  $\rho_x$  大,  $\rho_\alpha$  大;

例如对于  $D$  数值为 L4,  $\varphi$  数值为 L5 的输入, 规则 2、5、8 适用, 则得到:

对于规则 2,  $\mu_2 = \mu_{x小} = \min(\mu_{D小}, \mu_{\varphi中})$ ;

对于规则 5,  $\mu_5 = \mu_{x中} = \min(\mu_{D中}, \mu_{\varphi中})$ ;

对于规则 8,  $\mu_8 = \mu_{x大} = \min(\mu_{D中}, \mu_{\varphi大})$ 。

$$\text{则可得量化输出量 } u = \frac{\sum_{i=1}^M \mu_i \omega_{bi}}{\sum_{i=1}^M \mu_i}。$$

A、B 子集的模糊控制量化值  $u_{AB} = (1-s)u_A + su_B$ , 三个子集的模糊控制量化值为  $u_{ABC} = (1-s)u_{AB} + su_C$ 。其中  $s = |sat(u_i) + sat(u_j)|/2$ ,

$$sat(x) = \begin{cases} \text{sgn}(x) & |x| > \varepsilon \\ x/\varepsilon & |x| \leq \varepsilon \end{cases}。$$

设采样间隔  $t_s$ ,  $\alpha t_s$  后到达目标点, 则前述求得的路径目标点控制量

$$\rho_{ed}(k+\alpha) = [x_{ed} \quad y_{ed} \quad z_{ed} \quad \alpha_{ed} \quad \beta_{ed} \quad \gamma_{ed}]^T, \text{ 可求得各个关节的角度 } \theta(k+\alpha)$$

(参见第二章), 继续求下一个目标点  $\rho_{ed}(k+2\alpha)$  的控制量。



## 第四章 机器人三维图形仿真技术

### 4.1 图形学介绍

三维图形仿真技术是计算机图形技术的一个分支<sup>[34]</sup>, 应用数字手段来处理场景造型、取景变换、视域裁剪等计算, 生成三维真实感图形。

在机器人控制系统中, 三维图形仿真技术的应用并不侧重于图形学底层算法, 研究对象不是底层算法的实现和优化, 而是在底层三维图形支撑环境中创建真实的三维场景, 实现机器人本体及其环境的几何建模, 为机器人控制系统提供一个研究、观察平台。同时, 这个图形仿真系统还应该能够方便地获取以及存储机器人模型位置和状态信息。

计算机技术发展出现了多种底层三维图形生成软件, 主要有: OpenGL<sup>[35]</sup>, Direct3D<sup>[36]</sup>, Glide<sup>[37]</sup>等, 其中 Direct3D, Glide 主要运用于游戏开发领域。

OpenGL 是一种与硬件、窗口系统和操作系统相独立的一系列 API, Microsoft 和 SGI 将其在 Windows98/NT 上实现, 能直接面向硬件调用 3D 处理功能, 支持网络运行, 可将 3D 处理任务分配到其他计算机上处理。

这里需要注意辨别一个概念, OpenGL 是一种图形处理核心, 它的硬件无关性是针对 API 程序员而言的, 对于 OpenGL 实现者来说, 是与硬件相关的。如果硬件有图形加速器, 那么生成指令由硬件执行, 否则完全由 CPU 完成描绘。其底层运行机制对于程序员而言没有必要了解, 是透明的。

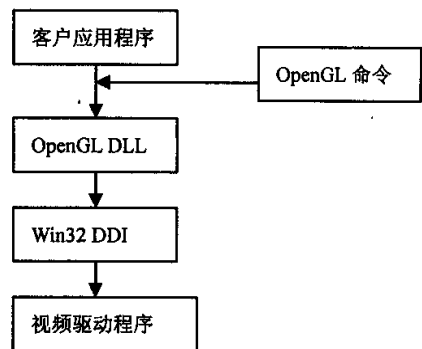


图 6 OpenGL 在 WinNT 下的运行机制

作为图形处理核心, OpenGL 在 WinNT 下的运行机制如图 6 所示。

在 OpenGL 核心层之上, 有多种三维图形绘制方法, 最基本的是 OpenGL API 函数的运用。OpenGL API 函数包括标准库、实用库和辅助库, 提供了最基本的几何图元, 具有的双缓存(Double Buffering)动画功能(后台缓存计算场景, 生成画面, 前台缓存显示后台缓存已经画好的画面)<sup>[38]</sup>。

此外还有名目繁多的各种图形绘制工具,较典型的有 Open Inventor<sup>[39]</sup>、Open Optimizer 等,均是以 OpenGL 为底层基础开发的。

以 Open Inventor 为例,其 Win32 下的结构层如图 7 所示。

由此可见,以 OpenGL 为图形核心的三维图形仿真方法是多样的,技术方案的选择应该根据系统需要进行详细分析。

为达到理想的显示效果和方便的控制功能,虚拟现实技术也开始进入机器人领域<sup>[40]、[41]、[42]</sup>。作为可视化工具,主要应用于:机器人遥控操作界面,可以实现远程控制中的模拟;几何模型与场景的绘制,实现三维图形仿真功能;进行视觉目标三维重建,这是机器人视觉的一个研究内容。在机器人控制系统中,其主要应用是三维图形仿真,场景模型构造语言为虚拟现实构造语言 VRML<sup>[43]</sup>。

VRML 语言是一种面向对象的描绘虚拟世界三维物体的语言,最基本的一个语法单位为节点,其构造方法就是围绕着节点的设置和扩展进行的。另一个基本概念就是传感器,是 VRML 语言场景中事件控制的关键。VRML 的场景描绘具有很好的面向对象的层次结构,并且是嵌套式和可扩展的,将其应用于机器人三维图形仿真,不仅可以利用许多商业三维建模工具强大的图形功能,而且坐标平移旋转设置方便,图形对象操作方法简单直接。

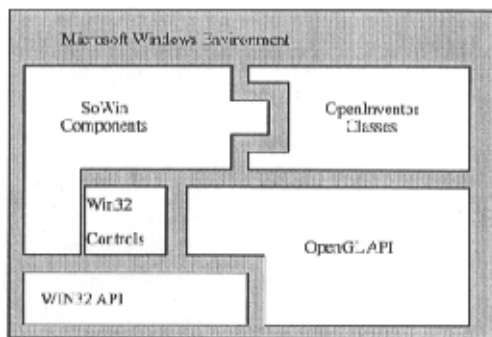


图 7 Open Inventor 结构层次

## 4.2 机器人三维图形仿真的实现

在机器人三维图形仿真的具体实现上,如前所述有各种方法。选择合适的实现方法,在缩短开发周期,满足系统控制要求的基础上达到理想的显示效果,是机器人离线编程控制系统开发工作的一部分。

现取其中几种典型的实现方法进行分析比较: OpenGL 程序设计, Open Inventor 程序设计<sup>[44]</sup>, Cortonas SDK 的运用<sup>[45]</sup>。它们都是以 OpenGL 作为图形核

心的，但是各有其应用特色和技术侧重点。

#### 4.2.1 VRML 文件

Open Inventor 程序设计和 Cortonas SDK 是采用 VRML 进行机器人建模的，在机器人三维图形仿真的应用中，VRML 程序结构设置如下：

采用嵌套式结构，建立 6 个 transform 节点，每个 transform 节点对应一个运动单元，同时这些节点逐层嵌套包含，运动单元的旋转平移等动作控制由 transform 节点的 rotation 域和 translation 域决定。

运动单元的具体形体材质绘制由 3DSMAX 绘制完成，同时将它们的坐标系定位为装配坐标系（对应于上一个装配部件的坐标），3DSMAX 将图形文件输出为 wrl 文件格式，通过 Inline 内联节点，将运动单元的形体包含到 transform 节点中。

工作环境的导入通过增加节点实现，将环境描绘文件作为 Inline 内联节点导入。

这样生成的三维图形文件，图形逼真，绘制过程直观，避免了大量的点、面函数绘制，减少了计算量，可用于各种支持 VRML 文件格式的开发程序中。Open Inventor 和 Cortona ActiveX Control 就可以采用这种文件格式，将它应用于机器人三维图形仿真程序的开发。

VRML 文件代码如下：

```
DEF myscene Transform{children [DEF MyViewPoint Inline {url "..\vp.wrl"}
DEF base Transform{children[Inline{url ".. \base.wrl"}
DEF shoulder Transform{children[Inline {url "..\ shoulder.wrl"}
DEF arm1 Transform{children[Inline {url "..\arm1.wrl"}
DEF arm2 Transform{children[Inline {url ".. \arm2.wrl"}
DEF wrist Transform{children[Inline {url "..\wrist.wrl"}
DEF frange Transform{children Inline {url "..\ frange.wrl"}}}}}}}}}}}
```

#### 4.2.2 OpenGL 程序设计和 Open Inventor 程序设计

OpenGL 程序设计是指用 OpenGL 函数进行三维图形模型的绘制，是一种常

用的三维图形仿真方法,利用其显示列表和双缓存功能能够较好地实现机器人运动模拟。但是它存在着编程工作量大,控制不够方便灵活的缺点。而且没有提供事件处理和窗口管理,对于目前面向对象的程序开发者来说,是一个缺憾。

Open Inventor 是一个基于 OpenGL 的面向对象的三维交互式应用工具箱,能够生成具有 3D 图形功能的 MFC 应用框架,自动添加交互功能和通用的 3D 图形服务。

对于机器人三维图形建模来说,Open Inventor 的主要优点在于:

- 可以利用自动缓存、消隐和属性生成提高开发效率。
- 有方便的 Win32 特定扩展,支持 VRML,能够生成逼真的三维模型。
- 支持窗口管理和事件处理,自动插入 Inventor 操纵体,渲染域,MFC 视图类,管理三维场景,减少代码开发量。

Open Inventor 是面向对象的编程工具,简化和加速了开发周期,但 Inventor 许多方法不是线程安全的,对于要求使用多线程的机器人控制系统程序来说比较困难。同时,虽然有很多子分类和扩展工具箱,但是它提供的抽象级别较高,不能直接控制特定的 OpenGL 参数。

#### 4.2.3 Cortona SDK 的运用

Cortona SDK 包括两个 ActiveX 控件:

a) Cortona Control:

Cortona VRML Client 提供的 ActiveX 控件,能够很方便的应用于 MFC 应用程序框架,实现三维图形的显示和操纵。生成的 CCortona 类主要包含了界面操纵、显示模式设置等函数,控件还支持各种鼠标响应、事件响应。其中的 GetEngine 函数和 SetEngine 函数为 VRML Automation Library 提供了一个接口,使得 VRML Automation 的函数可以通过它操纵三维场景。

b) VRMLAutomation:

提供了修改、控制 VRML 文件节点和参数的各种函数,并通过 Control 的 Engine 机制和程序相连,能够增减节点、域,修改域值,获取事件响应等。

对于机器人三维图形建模来说，它的主要优点在于：

- 支持 VRML 文件格式，能够生成逼真的三维模型。
- 能应用于 MFC 应用程序，提供了事件处理和窗口管理函数，自动生成可操纵界面，管理三维场景，减少代码开发量。
- 能够获取 VRML 文件内的各种节点参数、域值，对三维模型的操纵具有很大的灵活性。

在 Win32 VC++ 环境下，利用 Cortona ActiveX Control 和 VRMLAutomation 进行机器人三维图形仿真建模，主要过程如下：

导入 Cortona ActiveX 控件到工程中，设置各项显示参数，通过 Engine 获得 Automation 接口，生成或者导入绘制机器人模型的 VRML 文件，调用相应的节点、域值管理函数获得模型中各个运动单元的位置状态信息并通过接口将这些数据返回到控制程序中，在控制程序中设置和改变运动位置参数，再通过结构修改三维场景文件的各个节点域值，从而实现机器人模型的三维图形仿真运动。

主要类和函数介绍如下：

CortonaActiveX 控件类	CCortona
获得 Engine	LPDISPATCH CCortona::GetEngine()
设置 Engine	Void CCortona::SetEngine(LPDISPATCH newValue)
创建节点	CreateNodeFromString(/* [in] */ BSTR Vrm1Syntax, /* [retval][out] */ LPNODEOBJECT __RPC_FAR *Node)
获得节点	get_Nodes(/* [retval][out] */ LPNODESCollection __RPC_FAR *Collection)
获得域	get_Fields(/* [retval][out] */ LPFIELDSCollection __RPC_FAR *Collection)
接口传递函数	QueryInterface ( const IID& iid, _InterfaceType*& p )
坐标设置/获取	put_X(/* [in] */ float X) 、 get_X(/* [retval][out] */ float __RPC_FAR *X)

### 4.3 网络环境下图形仿真的实现

#### 4.3.1 技术发展现状

网络环境下图形仿真的实现是一个热点问题。在计算机技术飞速发展和网络日益普及的背景下，在网络上建立一个虚拟的形象世界是非常有吸引力的工作，这里就涉及到了图形仿真技术在网络环境的应用。具体到基于网络的机器人控制系统，对图形仿真的基本要求有：

- 必需的下载量尽可能小
- 对操作系统依赖性小，平台移植性强
- 仿真生动，操作灵活

在 Web 上实现图形仿真，VRML 有其独特优势，除前述模型控制灵活，图形格式简单等优势外，平台移植性强，能够建立基于 Web 的三维可视化服务器，网络浏览速度快，在 Internet 上得到了普遍应用。

如 VRML 2.0 Robot<sup>[46]</sup>，这个机器人仿真系统用 VRML 代码实现，用户可以通过网页远程控制机器人。TEF(Technology Exposure Facility)<sup>[47]</sup>系统也用 VRML 实现了对远程机器人的模拟操作。

但是作为机器人控制系统的组成部分，若图形仿真纯粹使用 VRML 编程实现，不但程序显得臃肿冗长，仿真控制和实体控制之间的通讯较为复杂，而且数据计算、坐标变换难以实现，因此必须借助其他辅助实现手段。

Java 具有强大的移植能力、多线程处理和联网能力<sup>[48]</sup>，非常适合用于图形仿真模块的控制界面编程。目前很多基于网络的机器人控制系统都不约而同在控制部分采用了 Java 编程实现。如 PRT RichODL Group<sup>[49]</sup>提供的“Robot Simulation”，Hironori Hiraishi 提出的基于 Web 的多机器人操作系统<sup>[50]</sup>，Mohamad Fauzi Zakaria 等人提出的基于 Internet 的远程机器人控制系统<sup>[51]</sup>，在系统的控制实现上均采用 Java 技术实现。

#### 4.3.2 关于 Java 技术

目前 Java 技术本身已经具有丰富的图形处理能力，不再局限于简单线条界面，利用其 Java3D 技术，可以很好的完成三维图形的建构和图形仿真控制功能，



那么，为什么还要采用 VRML 技术和 Java 的联合编程而不是直接使用 Java3D 技术？

技术手段的采用是和系统需求相对应的。应用于网络机器人控制系统，要求图形建模文件必须简单易懂易写，对于机器人研究者而言，需要方便地更换机器人三维模型，对于学习者、机器人兴趣爱好者而言，机器人图形文件的易读性是很重要的。作为一个研究、学习平台，基于网络的机器人控制系统应该给使用人员提供一个易于进入的切入点，而不能提出过多的其他领域如计算机编程技术方面的要求。

JAVA3D 是建立在 JAVA2(JAVA1.2)基础之上的，为编写三维应用程序提供了一个非常完善的 API，可以具有行为(Behavior)的处理判断能力（键盘、鼠标、定时等），可以编写非常复杂的应用程序。其优势在于功能强大，但和 VRML 相比较，复杂很多，要求更多的 3D 编程技术。

VRML2.0(VRML97)自 1997 年 12 月正式成为国际标准之后，在网络上得到了广泛的应用，编写 VRML 程序非常方便（VRML 语言可以说比 BASIC 还要简单），同时可以编写简单的三维图形文件，且代码易读。

而从开发工作上来说，完全采用 Java3D 技术需要完成“浏览窗口—模型建立—操作控制”的整套编程过程，利用 VRML 和 JAVA 的联合编程，浏览窗口可以利用可移植性很强的 VRML 浏览器，后继工作也简单很多，可以说是一种比较省时的开发方案。

综上，“VRML 图形文件+Java 控制面板+可移植 VRML 浏览器”可以实现网络机器人控制系统的图形仿真要求。

#### 4.3.3 Java 和 VRML 的联合编程

在网络环境中，结合机器人控制系统的应用要求，采用 VRML 和 JAVA 联合编程实现机器人模型的三维图形仿真，是一个可行的技术解决方案。

VRML 图形文件的建立参见本章前述。

联合编程有两种实现接口：

●外部编程接口 EAI(External Authoring Interface)

使用同一网页上的 Java Applet 程序和 VRML 浏览器相互通讯。目前 VRML

浏览器（如 Cortona）基本都支持这种编程接口。

Java Applet 程序即 Java 小应用程序，是在 Web 页上运行的 Java 程序，能运行于任何与 Java 兼容的浏览器（如 IE、Netscape）。Web 页需要设置 Applet 的一些定位、类参数，主要有：CODE（小应用程序的类文件名）、CODEBASE（类文件的目录）、ARCHIVE（Java 档案文件，包括针对小应用程序的专用类和其他资源（如界面类））。Web 页和 Applet 间通过设置参数来传递消息。

#### ●脚本编程接口 SAI(Script Authoring Interface)

使用传统的 JavaScript 编程。JavaScript 是脚本编写语言，由 Netscape 发明，语法描述性强，较为繁琐。

相比之下，外部编程接口编程方便，开发周期短，控制更加灵活。

采用外部编程接口 EAI 进行 VRML 和 JAVA 联合编程，需要加载 VRML 浏览器的 EAI 支持包（以 cortona 为例，cortecai.zip 提供了支持 Java Applet 的 Browser 类及各种节点控制类）。正确设置 Applet 的参数：codebase, code, archive, mayscript 等。另外注意设置“BrowserIndex”，此参数表明对应的 VRML 浏览器在本 Web 的次序，从 0 开始算。往往很多运行错误都是由于参数的设置错误引起的。

特别提醒，如果在 IE 里选中“使用 Java 2 v1.x.x 用于<Applet>”，那么 EAI 初始化时候会失败。

利用外部编程接口可以很方便的实现 VRML 图形的控制。比如：

选择视点、背景等可以通过 set bind 来完成，在 Applet 里面就是这样的代码：

```
Node viewpoint = m_browser.getNode(nodename);
```

//node 是 wrl 里面 def 的节点名称

```
EventInSFBool bind = (EventInSFBool)viewpoint.getEventIn("bind");
```

```
bind.setValue(true);
```

旋转可以通过增加传感器和 route 来完成，在 Applet 里面就是这样的代码：

```
Button button = (Button)event.target;
```

```
if (button.getLabel() == addRoute.getLabel())
```

```
    browser.addRoute(nodeFrom, eventFrom, nodeTo, eventTo);
```

```
else if (button.getLabel() == delRoute.getLabel())
```

```
    browser.deleteRoute(nodeFrom, eventFrom, nodeTo, eventTo);
```

## 4.4 实验效果分析

### 4.4.1 PC 环境

基于 PC 的机器人控制系统设在 Win32 VC++开发环境下, 分别用 OpenGL 程序设计、Open Inventor 设计、Cortona 应用实现了机器人三维图形仿真, 显示的效果图如图 8、图 9、图 10 所示。三个程序均能在 Windows2K 操作系统中很好地运行, 较好地模拟、再现机器人运动状态, 有较好的动画效果。三者比较之下, Cortona SDK 效果理想, 编程方便, 是一种理想的机器人三维图形仿真工具。

#### 实验 1、OpenGL 程序设计:

OpenGL 程序设计绘制的模型只是简单几何体的组合, 在对机器人模型细节的绘制上不能达到满意的效果, 动画显示运行过程中对事件的响应速度较慢, 由于绘制中需要用到大量的点、面、柱函数, 计算量明显比其他两种程序设计大, 为了增加事件响应和窗口管理, 程序开发的工作量也是最大的。

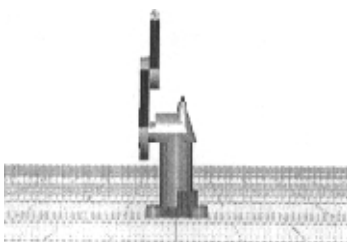


图 8 OpenGL 程序

为减少 OpenGL 程序设计中的绘制计算量并且改善模型绘制效果, 可以考虑用 OpenGL 函数读取 3ds 文件的方式来得到模型的结构参数。用 3DSMAX 绘制模型并输出 3ds 文件, 用 OpenGL 函数读取, 将其中的数据转存到相应数据结构中, 然后放入显示列表, 再调用显示列表进行绘制, 这样能够得到逼真的机器人模型, 同时简化了显示列表中的绘制工作。但是由于用 OpenGL 函数读取 3ds 文件, 需要编写大量代码, 此外在装配、运动模拟过程中, 还需要对各个运动单元的坐标系进行计算设定, 加大了程序设计的工作量, 因此并不是很理想的解决方法。

OpenGL 程序设计, 在导入机器人工作环境的实现上, 没有后两种方法方便。Open Inventor 和 Cortona SDK 的模型绘制都采用了 VRML 文件格式, 因此, 在导入工作环境的时候, 只需要增加一个节点, 将环境描绘文件作为 Inline 内联节点导入即可, 十分方便。而且 VRML 文件自身可以包含各种事件传感器, 为进行机器人运动规划、碰撞测试提供了方便。

#### 实验 2、Open Inventor:

Open Inventor 直接提供了界面类, 简化了程序开发, 但是如前所述, 它不适用于多线程程序, 对于机器人三维图形仿真系统的工作要求, 不能很好的满足, 而且对于 VRML 文件中节点、域值的管理不够方便。

### 实验 3、Cortona SDK:

应用 Cortona ActiveX 控件和 VRMLAutomation 能很好地实现设计要求, 绘制计算量小, 响应快, 动画效果好, 在 VC++ 中进行程序开发十分方便快捷, 而且能够灵活管理 VRML 的节点、域值, 从而实现对机器人模型运动单元的各种控制。需要注意的是, 运用这种方法要求系统必须安装 Cortona 3.1 或更高版本<sup>[10]</sup>。

由上述比较分析可得:

采用 Cortonas SDK 进行机器人建模的, 可以达到近乎理想的仿真效果, 优于其他两种实现方法。

### 4.4.2 网络环境

在基于 Web 的机器人控制系统中, 利用 VRML 和 JAVA 联合编程实现了图形仿真模块, 其效果图见图 11。在三维图形仿真模块的实现上, 模型控制灵活, 下载方便, 平台移植性强, 能很好的满足机器人控制系统的控制和仿真要求。

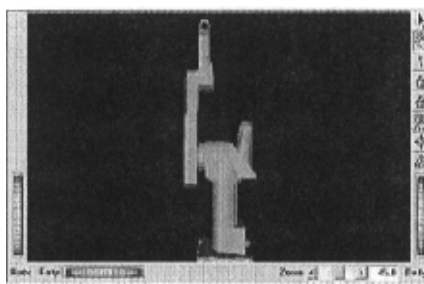


图 9 应用 OpenInventor

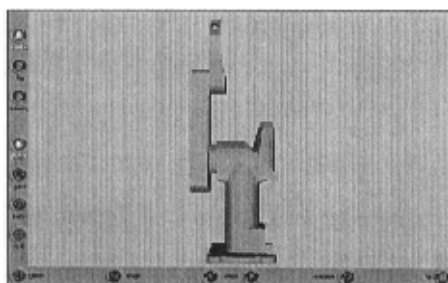


图 10 应用 Cortona SDK

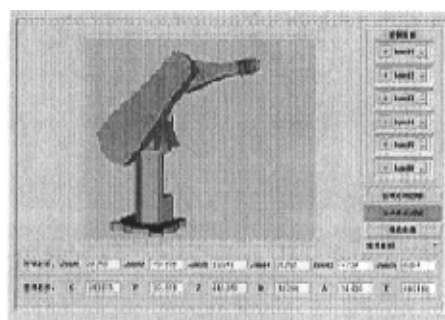


图 11 仿真界面图

## 第五章 基于 PC 的机器人控制系统设计

### 5.1 系统设计目标

打破机器人控制器的封闭式结构，设计具有开放性的控制系统，目标是：采用标准的操作系统和标准的操作语言；利用计算机平台的软、硬件资源对控制器进行扩展；实现友好的人机界面；采用离线规划以减轻计算负担。

为满足上述功能要求，机器人离线编程系统必须具备先进的几何造型系统。目前较为流行和成熟的离线编程系统大多基于工作站以上级别的计算机，系统成本昂贵，限制了离线编程系统的推广使用。近年来，计算机技术迅猛发展，PC 机在多媒体技术、人机界面交流以及用户功能扩展方面表现都十分出色，开发基于微机的机器人控制离线编程系统已经成为可能<sup>[52]</sup>。

针对已有的基于微机的离线编程及仿真系统（如 EASY-ROB<sup>[53]</sup>、Workspace<sup>[54]</sup>、RoboWorks 系统<sup>[55]</sup>等）中通用性软件和机器人控制器的联系不是很紧密、功能也比较单一的不足，本文提出了一种基于微机的机器人控制系统设计方案，以 PUMA560 机器手臂通用控制器为对象，打破其封闭式结构，开发了图形仿真和离线编程系统，其主要特点为：

- 1、系统独立性强，软件设计模块化，易移植、扩展；
- 2、图形仿真采用 OpenGL 核心，动画质量好，效果逼真；
- 3、人机界面友好，改善了编程操作环境；
- 4、数据计算处理能力增强，能够进行实时多任务处理；
- 5、能够进行进一步软件开发，进行运动规划。

### 5.2 系统结构介绍

#### 5.2.1 硬件结构和方案确定

PUMA560 机器手臂通用控制器分为装置本体和外围设备两部分。装置本体包括 16 位微机、伺服电路、CPU 接口（串行接口用于外围设备，并行接口用于伺服电路）、时钟电路。外围设备包括终端、示教盒、软盘（辅助记忆装置），参

见图 12。

其中 SCM 负责 VAL 处理及基本运算，TPB 协助控制机器人伺服马达，AIB 是消息处理中心，完成各板之间的信号传递处理，并和外界交互。

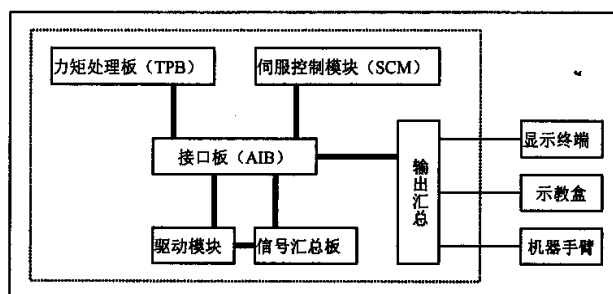


图 12 原系统硬件组成

为打破其封闭的控制结构，扩展控制系统，有多种方案，主要有：

一， 装置本体功能解剖后的重新设计。

目前很多研究者采用这样的方案，详细解剖装置本体，设计新的硬件系统来取代现有控制器，在新的控制器中增加硬件扩展能力，扩展计算功能。这样的方案优势在于，可以在机器人关节控制等底层控制方面得到非常大的自由，力矩、速度等传感器的信号能够切入底层控制策略。但是，这样的设计方案，不但需要详细的解剖，而且在硬件设计上需要大量工作以确保控制精度和提供完善的安全保障机制。这样得到的控制系统，很多功能模块是完全复制性的，却需要很长的开发调试周期。

二， 利用原有装置本体，进行功能扩展。

这样的设计方案更侧重于上层控制的实现和扩展能力，利用原有装置本体的底层控制功能，通过输入输出口的扩展，实现系统控制的开放性。这种方法缺点在于在关节控制方面受到原有系统的局限，不能深入底层，控制自由度不大。其优点在于，开发周期短，新系统可移植性强，在原有装置本体的控制精度和关节控制性能满足要求的情况下，是一个很好的解决方案。

在系统的改进中，采用了第二种方案。考虑使用控制微机取代终端，仍然使用串行接口和原系统进行通讯。新的系统结构为：机器手臂、通用控制器、示教盒、控制微机。其中机器手臂、通用控制器、示教盒三个部分为原有系统部分，控制微机通过串行通讯接口和原有系统相连，取代原有终端系统。结构框图如图

13 所示。

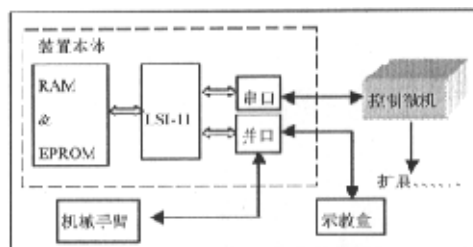


图 13 控制系统组成结构示意图

在控制器改进设计中采用占用终端串口的方式，而不涉及对控制器本体的硬件改进，这样新的控制系统的独立性和通用性更强，可以很方便的移植到另外微机，成为其他控制器的上层控制系统，而且在系统设计过程中，也可以很方便地将通用控制器看成一个黑匣子，只需要考虑数据的传送输出问题，而不用顾及其他硬件设计，简化了软件设计任务。

控制微机为奔腾 III 微机，Windows98 操作系统。软件开发使用 Microsoft Visual C++，采用模块化设计（主要类列于表 3），各个功能块之间独立性强，增强了系统的灵活性和可扩展性。硬件扩展则利用了微机的硬件扩展端口。

表 3	
控制软件中主要的类	解释
通讯类	负责控制微机与通用控制器的信息传递。
CMathCom 类	各类算法（运动规划等）的函数集合
CRenderView 类	调用 OpenGL 函数进行模型绘制和动画显示
CReadMod 类	读取图形文件并且转换为显示列表
CJointRec 类	读/写 mdb 文件，管理关节位置信息和进行坐标矩阵转换运算
CMyCom 类	控制面板的参数处理，生成简单的 VAL 指令

和原系统相比，它的优势在于：硬件扩展能力增强；能够存储大容量的程序文件；具有图形处理能力；改善了机器人操作的开发环境和应用环境，能实现友好的人机界面；数据计算处理能力增强，能够进行实时多任务处理；能够进行进一步软件开发，进行运动规划。

### 5.2.2 功能层次结构

从系统的功能层次结构来看,基于 PC 的机器人控制系统由用户级(任务级)、系统级(执行层、应用层)、伺服级(运动控制层、手臂控制层和电机控制层)这三层组成,其示意图见图 14。

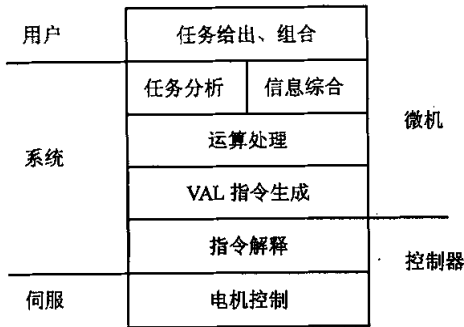


图 14 系统功能层结构示意图

用户级: 用户给出控制指令, 设定控制任务。这一功能层也可称为任务级, 即对于用户而言, 所见为以任务为划分的操作, 底层操作被屏蔽。

系统级: 实现了用户控制意图的解释, 综合处理各种信息, 如控制算法的择优选择, 传感器信息的处理, 控制指令(任务)的 VAL 化及其解

译。这一功能层主要在控制微机实现, 其中有部分(V AL 语言的解译)位于通用控制器。

伺服级: 负责机器手臂的直接运动控制, 包括电机控制, 电信号反馈等。

### 5.2.3 控制回环结构

基于 PC 的机器人控制系统是一个反馈控制系统, 其控制回环见图 15。图表明了整个系统的信号流程, 图 16、图 17 则分别描述了图中的 PC 机控制程序回环和通用控制器控制回环。

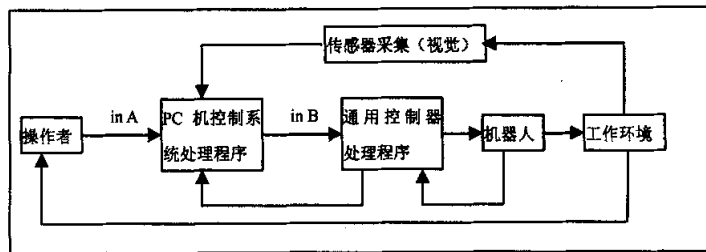


图 15 系统控制回环



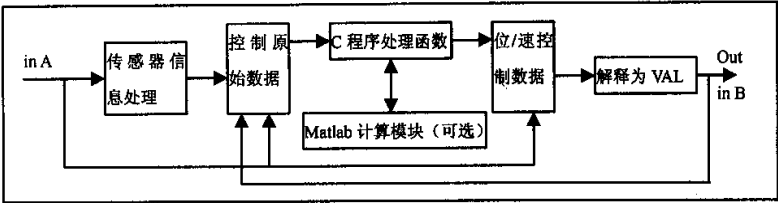


图 16 PC 处理程序框图

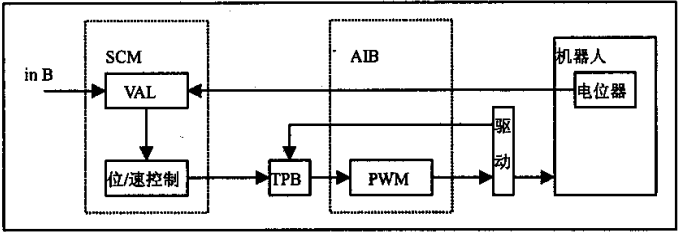


图 17 机器人控制框图

5.3 运行模式和控制方式

系统分为两种运行模式：仿真模式、工作模式。

在仿真模式下，可以看到机器人三维立体模型和机器人工作模型的图形界面，在控制面板中输入不同数据或者设置、选择不同参数，可以进行机器人手臂的运行仿真。在工作模式下，可以通过三种控制方式对机械手臂进行实际控制操作。

三种控制方式如下：

第一种是示教方式，遵循原有系统的示教操作方式。

示教编程是一种传统的编程方式，可以解决机械系统误差这个难题，提高机器人相对空间某指定点重复定位精度，使机器人可以成功地进行固定路径的重复操作。随着机器人任务的复杂化和多变化，采用示教编程占用时间太长，效率不高，尤其工业机器人，当产品变化频繁的时候，这种编程方式需要频繁操作更换，影响生产效率。

第二种是基本的机器手臂操作方式。

在这种控制方式下，通过控制系统视窗，可以直接看到通用控制器所反馈的机械手系统状态，可以直接用 VALII 语言，在此视窗中输入执行命令，对机械手臂进行操作。在这里，命令的解释工作并不是由本系统完成，而是在通用控制器

中完成，本系统所起到的作用只是字符传送。

这种控制方式要求使用者掌握 VAL 编程技术，是原有封闭系统的终端功能模拟。较之原终端，在微机控制系统中操作者还可以通过剪贴、拷贝等编辑方式来保留所编辑程序和反馈信息以供分析，操作得到简化。

第三种是程序执行方式，这种方式下又有两种不同的具体方法，

一是文件读写方法，系统读入运用 VALII 语言编写的机器人执行程序文件并传递给通用控制器，使机械手臂执行此程序；

二是通过控制面板，设定机械手臂的各种运动、位置参数，由系统调用相应的函数，将参数填入 VAL 程序中，然后执行。

相较而言，方法二由于界面的直观性较好，比较便于操作，各项数据的可读性强。但是由于目前系统中已经实现的函数解释还不多，此方法只能用于比较简单的运动操作，对于较为复杂的运动，还是直接编写 VAL 语言程序效果较好。

以上两种操作控制方式都在底层运用了 VAL 语言。

VAL 是一种专用的机器人程序设计语言<sup>[56]</sup>。现有的机器人语言可以分为两大类：机器人级语言（面向机器人的）、任务级语言（面向对象的）<sup>[57]</sup>。

在机器人级语言中，机器人执行的任务被描绘为机器人运动序列，程序的每个语句相当于一个机器人动作，整个程序引导和控制机器人完成任务。在任务级语言中，机器人执行的任务被描述为机器人操作对象的位置目标序列，而不是机器人为完成这些目标所需要的运动。目前大多数机器人语言都属于机器人级语言，VAL 亦在此列。对于普通用户来说，运用专用机器人语言并不是一件愉快的事情。

为了提高系统的可操作性，更加便于操作，降低工作难度，应该在系统设计上考虑界面上的设计，使数据输入和运动控制直观化，更为重要的是，应该从设计语言着手，比如用 C 语言进行算法、运动、位置信息的初步处理，再解释成相对简单的 VAL 指令序列，从而使得操作人员可以用普通的编程语言进行机器人运动程序设计。这样的解释模块有待于进一步研究。

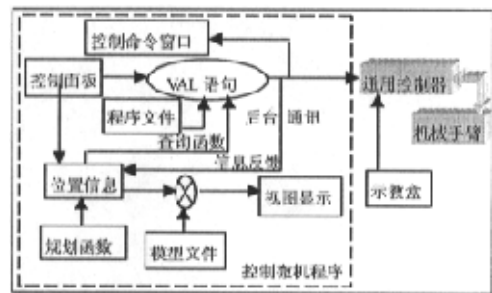


图 18 控制示意图

不同机器人需要不同的机器人操作语言，这造成了系统移植方面的困难。为了增加系统的通用性，应该从系统方法出发，开发一个模块，设计输入编译接口和输出编译接口，输出口负责控制码下装，输入口负责机器人运行情况反馈读入，这样系统可以在统一的语言下进行编程，经过仿真验证后，下载到机器人，并获得实时的运行数据并保存。这样的设计模块需要绕过原有机机器人语言控制系统，涉及到较底层的控制开发。在本系统设计方案中，由于采用的硬件结构特点（未抛弃原有底层控制器），未进行这样的通用模块开发，而是采用了“人机界面接收数据——控制程序理解——VAL 程序填空发送”的处理流程，可以满足使用 VAL 语言的系统。

#### 5.4 控制系统中的图形仿真模块

机器人模型的建立和仿真的实现是其中的关键一步，其中主要涉及到：

##### 5.4.1 模型的基本设计

PUMA 的结构简图见图 1。机器人运动学和几何学建模如第二章所述，得到了坐标变换矩阵，就可以进行机器人手臂的仿真模型绘制了。机器人模型层次结构如图 19 所示。

为了简化计算量，在绘制软件的设计过程中，应该对坐标系统采取嵌套式结构设计，以世界坐标系为第一坐标系，在此基础上建立底座(base)的坐标，而肩部的坐标系统以底座坐标为基准，以此类推。采用这样的嵌套结构绘制模型，能够大大减少运动矩阵计算量，提供软件工作效率。

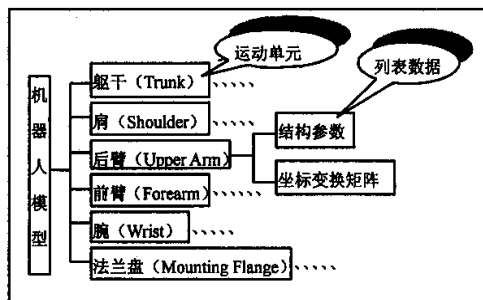


图 19 机器人模型层次结构图

##### 5.4.2 运动模拟显示

模型的图形文件是采用 VRML 文件绘制的，VRML 的一个基本特点就是节点结构（参见第四章）。因此，每个运动部件都和图形文件中的一个 transform 节

点相对应,运动单元的旋转平移等动作控制由 transform 节点的 rotation 域和 translation 域决定。具体绘制方法见第四章。

运动中,机械手臂的位置信息由系统从通用控制器读入(工作模式)或者由操作人员在控制面板中指定(仿真模式),以 mdb 文件格式存储。系统通过读取 mdb 文件获得各个关节的位置状态信息,修改相应的节点域值,从而显示相应的机器人手臂模型(参见图 18)。运动位置信息中世界坐标值与关节角度值的相互转换请参考第二章。在仿真模式的机械手臂模型模拟运动过程中,还可以通过快捷键截取机械手臂模型的各个关节坐标信息并存储,便于数据的分析和研究。

其中,模型的位置信息和实际机械手臂位置信息必定存在误差,如何对此误差进行校正,是保证系统控制精确性的关键,必须增加相应校正计算模块。

而运动速度信息则通过旋转角度时间周期进行控制,速度值可以由操作人员进行设定,也可以通过终端位置变化或关节角度变化计算所得,但在这里,仿真模式下的速度模拟比较容易做到,可以事先指定运动速度,而实体运动过程中的速度模拟跟踪存在着难点。因为这受到信息反馈渠道的限制。无论是从机器人控制器反馈位置信息还是从外部传感器(如视觉系统)获得位置信息,都有一定的滞后期,而且不可预测,再经过运算的耗时,速度的跟踪很难达到理想效果。

机器人仿真模型的运动模拟还提供了多视角观察功能。提供了三个基本视角(正面,左侧面,俯视面),也可以旋转和移动整个场景,实现任意角度的观察,同时提供场景放缩功能。

运动模拟中数据的分析和显示包括实时数据显示和静态处理,实时数据来源于反馈环节,机器人控制器反馈因为需要执行查询命令,只能在机器人实体运行的间隙点进行,数据量不大,但它的数据比较精确。从外部传感器(如视觉系统)获得的信息,信息流速只受数据处理速度的限制,但其数据精度受传感器精度及数据处理方法影响较大,需要引入误差校正运算。数据可以存储于指定文件(mdb)文件中,供程序进行分析形成报告,从中可以得到机器人运动轨迹报告和运动性能分析,为控制方法的检验讨论提供依据。

### 5.4.3 约束空间分析

机器人有其工作空间限制，PUMA560 型机器人的关节运动限制如图 20 所示，其角度数值见表 4，其工作空间在三维图形仿真界面中的模拟如图 21 所示。机器人运动轨迹的目标点必须位于这个工作空间内，否则不可达，系统报告出错。

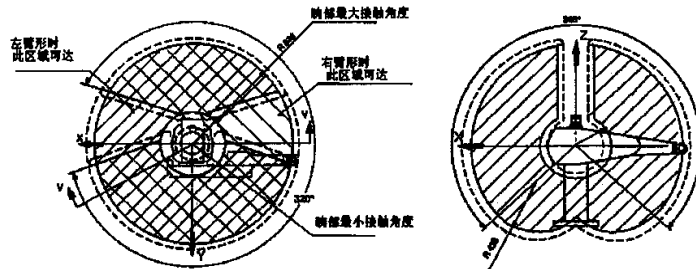


图 20 PUMA560 型机器人的关节运动限制

表 4						
关节	1	2	3	4	5	6
变化范围	-160~160	-225~45	-45~225	110~170	-100~100	-266~26

工作空间中，有灵活的工作空间和可达工作空间。灵活工作空间指机器人终端在所有方位上都能达到的空间领域，即灵活空间的每一点上，机器人终端可以取任意姿态；可达工作空间是机器人终端至少在一个方向上能够达到的空间领域。在运动模拟中，若目标点位于灵活空间，则姿态任意，否则，需要注意运动的可行性。

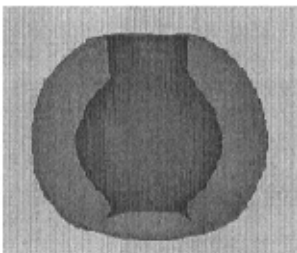


图 21 工作空间的三维模拟

障碍空间也是机器人运动中需要考虑的因素。障碍空间的描绘，以数学函数描绘是比较复杂的，但是通过外部传感器获取障碍点信息然后在三维图形仿真空间描绘障碍面，却是比较直观而可行的方法。例如利用机器人视觉处理系统，可以获取障碍物体的位置信息以及其边界点，数字图像重构处理以后获得了几何信息，在三维图形仿真空间增加这一几何体，就可以得到障碍空间描绘，而障碍空间和机器人工作空间的交集则提供了可能发生碰撞的点面信息，为避障规划运动提供了研究平台。

机器人运动还收到很多其他约束，如速度、加速度限制、技术限制等，在三维图形仿真控制系统中，主要体现和分析的是工作空间约束和障碍空间约束。通

过图形系统来分析约束空间,其优势在于非常形象直观,难度在于空间的正确描绘,其中障碍空间的描绘取决于传感器信息的完整性和数据处理精度。在视觉系统功能不是很强大的情况下,可以采用套用模型库的方法,将障碍物拟合为简单几何体组合进行重构,这可以加快数据处理过程,进行较为粗略的运动模拟分析。

#### 5.4.4 环境置换

机器人工作环境存在着不确定性,固定不变的环境模型是不够的,通过变换 VRML 文件的内联节点,读入不同的场景描绘文件,就可以进行环境置换,模拟不同的工作环境,进行不同的实验操作。

这里涉及到一个环境场景描绘文件的建立问题。

在教学用的机器人控制系统图形仿真中,可以设置场景库,预设实验场景描绘文件。在工业机器人作业中,可以预设零件库。但是,用场景库的形式,不能包含大多数机器人工作场景,实用性受到限制。不仅如此,机器人作业环境还存在着不确定性,有不可预测的变更以及不可避免的误差存在,如何对环境的不确定性进行抽取,并以此修改环境模型,是系统实用化的关键。

因此,在机器人环境场景描绘中,不仅需要实现环境物体的三维重构(图形学上的逆向工程问题),而且要模拟环境物体的运动和实现误差补偿校正。解决的方案是通过外部传感器(如视觉系统)信息综合,实时分析机器人工作环境并且建立模型。这无疑将极大提高系统性能,使之更为实用化。

#### 5.4.5 路径记录

三维图形仿真在机器人控制系统中,还可以提供形象的路径记录。仿真模型在运动模拟过程中,其手臂终端、关节的路径轨迹可以以彩色亮点的形式在三维图形空间中保留下来,这样的表达形式较之传统的数据记录——分析方法,是直观易见的,无需调用其他计算处理函数就可以得到大致的运动轨迹立体图,减少了数据处理量,能够进行简单轨迹的定性判断。

## 5.5 控制系统中的运算模块

### 5.5.1 坐标系运算

控制程序对坐标系运算的处理可参考第二章。

在 PC 机的 Windows 操作环境下，操作系统的多线程处理能力极大的增加了程序的计算能力。控制系统可以同时进行多个运算，将结果存入统一格式的数据文件（mdb 文件），最后一起下传给后继处理程序。

坐标系运算运用到了大量的矩阵计算，有很多数据处理，是原有系统中很难想象的工作，而在 PC 控制系统平台中则非常方便。不仅如此，Windows 操作平台下，还可以引用现有的科学计算软件作为计算辅助，极大地提高工作效率。在基于微机的机器人控制系统中，我们可以利用 MATLAB 的引擎机制，实现 C 程序对 MATLAB 的接口，调用其运算函数。

### 5.5.2 任务级指令的实现

原 PUMA 机器人控制器的封闭控制系统中，操作者只能在终端类 DOS 环境下，进行指令级的命令执行，可以利用 VAL 编写机器人控制程序，但程序代码无法存储，因此程序段无法复现。这样的操作特点限制了操作者的发挥，机器人控制的复杂任务执行较为困难。

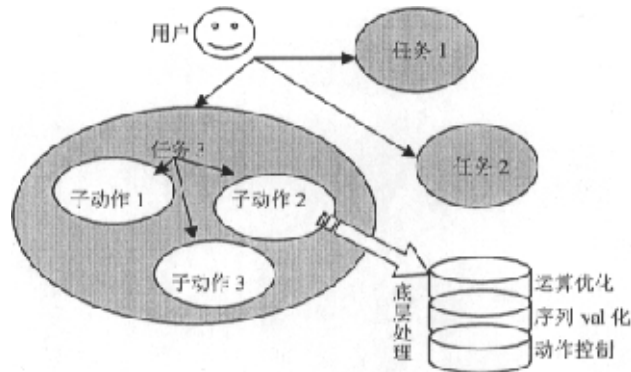


图 22 用户任务级控制示意图

基于微机的机器人控制系统对系统的存储能力进行了扩展，使得程序代码的存储成为可能，不但如此，还可以通过控制功能固化与组合，实现机器人控制的

任务级指令。

控制功能的固化可以隐蔽底层控制函数,使操作者脱离机器人编程语言的限制。机器人操作序列对于操作者而言,是具体实现的功能(如找目标、抓取目标、放下目标、进行设定运算等)而不是抽象的旋转角度、运动速度等指令。操作者可以根据任务需求(如点蜡烛、回到准备位置、拿火柴、放蜡烛),自由组合功能完成操作,控制示意图如图 22 所示。

### 5.5.3 插值点方法

前文中已经提到图形仿真功能对机器人轨迹规划的辅助功能,这里不再赘述。除此之外,控制系统还提供了很好的轨迹规划算法研究平台。由于控制系统的开放性,研究者可以自己编写 C 程序或者 C 程序可调用的计算模块实现具体算法,将运动目标点信息以世界坐标或者关节坐标的形式存储于统一格式的文件(mdb 文件)中,系统接到执行命令后,将按照文件中的轨迹定点进行三维图形仿真运动或者实体运动(运行模式由操作者指定)。

系统本身提供了一些基本的插值算法,同样可以处理比较复杂的规划,具体实现可参考第三章。

### 5.5.4 运动规划流程

运动规划的实现是以 mdb 文件为中介的。用户可以根据自己的控制策略采用不同的路径规划算法,将计算得到的运动路径数据和机械手各关节位置存于 mdb 文件中,由底层程序执行实体操作。

系统程序流程如图 23 所示。

在规划算法的选择上,有开发兴趣的研究者还可以加入自己的优化原则,如最小运动量优先,最快速度优先等衡量标准。

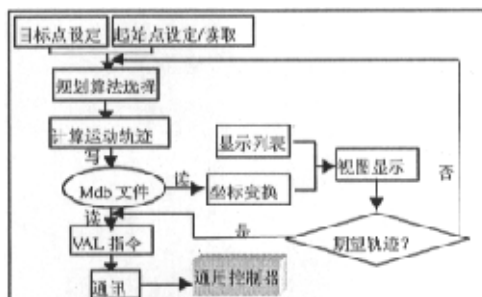


图 23 运动规划流程图

现已完成的机器人控制系统在控制数据传输上是以 mdb 文件为中介的,需要算法代码实现数据库的读写,编制不够方便,在机器人控制系统的进一步开发中,可以考虑增加一个数据读入接口,



比如，算法可以由 Matlab 实现，而系统程序接口可以自己从 matlab 工作文件中读取结论数据并存储到数据库。

## 5.6 安全机制

机器人控制系统设计中，必须建立完善的安全机制。在机器人工作时候，应该确保控制命令的正确执行、参数的合理性、操作人员的安全。除了制定安全操作规程，避免人为错误之外，更重要的是控制系统内部设置安全检查、纠错功能。

### ●控制参数的检验

控制参数，如角度值、速度值，是通过设置阈值来检验的，根据各个关节的运动范围、运动速度限制来选择合适的参数，剔除不合理值。对于以直角坐标形式给出的控制参数，则主要通过几何检验、图形仿真来检查是否出于合理范围，在运算方法上还可以选择转换为关节角度参数形式来检验。

控制参数也存在着误差，例如传感器信息的误差、运算过程中舍入带来的误差等，这些误差会造成理论控制效果与实际控制效果的差别。这个问题必须从多方面着手解决，例如，对传感器信息设置可信度参数，或增加传感器采集辨识精度，或多传感器信息融合；选择合理的处理算法，在障碍物周围设置缓冲范围；实时调整目标位置，实现微调功能等等。

### ●机器人启动的检验

机器人启动时，进行初始状态的检验，其中，装置本体提供了电源启动的硬件检验，而控制微机进行通讯检查、状态检查、关节校正。其过程为：

首先接通通用控制器电源，若启动不成功，就发出警报。通用控制器启动成功后，控制微机接收控制器发出的信号，若正常，可启动机器人手臂电源，否则，控制微机发出警报。手臂启动成功后，控制微机发出查询关节状态信息，若状态正常，则可开始操作，否则，控制微机发出校正命令，校正失败，则发出警报要求重启，否则，开始操作。

机器人的启动过程需要操作人员的全程参与。

### ●运动过程的监控

在机器手臂运动过程中，若出现超范围、超负荷运动，机器手臂采取立即停止不动，切断电源的方法进行自我保护，必须关闭一切电源才能重新启动恢复。

同时,控制微机接到控制器发出的出错信息,发出警报。

对于发生意外碰撞的情况,主要依靠运行前的运动模拟仿真和操作者的现场监视。事先进行运动模拟可以很好的避免意外碰撞,检验程序的合理性。操作者的监视保护,包括在确保自身安全的前提下移走障碍物,若不能进入机器人工作范围,则切断电源。在碰撞较为严重的情况下,机器手臂受到较大的反向力矩,则手臂自动停止运动,切断自身电源并发出警报给控制微机。

由于机器手臂的控制特点(控制指令发出后,系统无法中途切入控制),保证运动过程安全的最有效措施是事先进行详尽的运动模拟,掌握可能发生的情况。

### ●命令传送的安全

PC 控制程序接收控制参数,产生控制命令并将其传送到通用控制器,主要经过两个环节:PC 到控制器的传输、控制器到机器人伺服驱动控制的传输。

在 Windows 操作环境下,PC 控制程序实现了串口通讯的监控线程,采用 RS232 协议通过串口向控制器发送 VAL 语言格式的命令程序并接受控制器反馈信号。若通讯出错,命令不可理解,从控制程序的通讯平台窗口,可以看到控制器反馈的警报,该命令将被抛弃。

控制器到机器人伺服驱动之间的传输是通过 VME(Versa Module Europe)总线传输的,控制器装置内实现解译代码,各板块分别检验信号的正确性并将出错信号集中反馈到接口板。

### ●上层监控

在基于微机的机器人控制系统中采用视觉系统,不仅可以提供工作环境信息、目标坐标信息,还可以向操作者提供监控图像。视觉监控在非本地操作或工作空间不可见时是非常必要的,在现场操作环境中,则可作为一种信息补充和资料存储。监测系统一般采用固定观察点全局观测的方式,也可以附着于机器手臂上进行运动观测。前者视角收到局限,而后者对视觉处理系统的动态标定要求较高。在系统结构上,要求高质量或长期连续的图像监控的任务,最好单独采用一台微机进行图像工作处理,若是粗略观察或间隔性观察,则可采取小型视窗形式,和 PC 控制程序共同运行于一个操作系统内。

机器人控制系统的安全问题,不仅涉及到任务是否能完成,更关系到操作人

员、机械硬件的安全，必须在设计中加以格外重视，针对自身控制系统的控制特点、硬件特性、软件流程，制定严格的操作规范，并从设计结构上完善检测，减少意外。

## 5.7 小结

机器人控制器是机器人控制中的核心装置，针对封闭式机器人控制器的改进，本文提出的基于微机的机器人控制系统增强了控制器的扩展能力和计算能力，实现了以 OpenGL 为图形核心的仿真试验，提供了离线编程功能，在此基础上可以进行进一步的开发研究，如在系统中进行传感器的扩展和传感器信号处理，集成监控系统，对机械手臂的运动进行及时的干预，进行灵巧手和机械手臂的集成研究，解决协调控制问题等。

## 第六章 基于 Internet 的机器人控制系统设计

### 6.1 系统设计目标

机器人控制系统的控制方式网络化是近来机器人研究中的一个重点,网络化控制能够使操作者不受地域限制,对机器人进行遥操作和监控观察。针对不同的机器人系统应用方向和控制特点,设计方案的侧重也有所不同。

基于 Internet 的机器人控制系统在实现上有这样几种形式:

客户机下载专用的独立的应用控制程序,通过 Internet 网络连接机器人服务器,早期的机器人网络控制系统多采用这样的形式,优点是一次下载,无限次使用,但是因为每当控制程序改动时,用户需要重新下载,对于系统维护更新来说,不是很方便。

建立虚拟的网络图形环境,通过数据手套等硬件配合,实现对机器人的遥控操作。如哈尔滨工业大学研制的机器人手臂遥操作系统,操作者佩戴具有多个传感器的数据手套,利用计算机网络通讯,对机器手进行远距离控制操作。这种控制形式操作十分方便且易于理解,但是,硬件成本较大,难以普及。

Web 形式的机器人控制系统,这样的系统有很多,如“Australia's telerobot”<sup>[58]</sup>,用户通过 web 访问实验室里面的机械手臂,遥控抓取和放置各种颜色的积木,通过页面上的图像信息观察抓取效果。“CSC Telerobot”则让用户通过网页操作位于匹兹堡的卡内基科学中心的六自由度机械手<sup>[59]</sup>。Web 形式的系统访问浏览非常方便,控制程序与客户机隔离,维护方便,对于一般的机器人网络控制系统应用而言,能够很好的满足各项要求。

本文提出的机器人控制系统设计方案主要用于实现机器人教学的普及和实验室设备的校园化操作,采用 Web 访问控制形式。用户通过 Web 浏览器即可进入系统的控制界面,通过此界面可以取得与远程机器人的交互,对其进行控制,并通过远程监控模块得到实体运动的实时反馈信息,达到对机器人运动进行实时监控的目的。

结合机器人控制系统的教学用途,其设计目标如下:

- 实现友好的人机界面,在操作上提供详尽的指导服务,易于理解、操作。

- 实现机器人网络环境下的图形仿真，提供完善的运动仿真功能、路径分析功能。

- 实现机器人实体控制，提供固定环境下的基本实验操作、VAL 机器人控制命令的通讯平台、基本操作的模块化搭建。实现一个具有开发性的开发编程平台，操作者可以进行规划算法的检测和比较。

- 实现机器人环境的图像监控和分析，向远程用户提供图像观察服务，并为机器人控制提供环境信息数据。

- 实现用户之间的协同操作，提供在线讨论服务，同时建立严格的安全检查、监督机制，保证机器人的安全工作。

在技术手段的选择上，主要采用基于 Java 的网络技术，配合 VRML、JMF 等来实现各个功能模块。

Java 具有强大的平台移植能力，支持高级编程任务如网络编程、数据库连接以及多线程编程等，完全面向对象，因此，在机器人网络控制系统中采用 Java 编程技术，不仅在网络化实现上较为方便，而且便于进行各模块的配合和通讯。

在具体应用中，出现了两种方法：采用 Java 应用程序实现，或者采用 Java Applet（小应用程序）实现。

Java 应用程序与 Java 小应用程序的区别在于：前者是一个独立运行的程序，用户从网页（或者别处）上下载后可以独立运行，自身提供完善的功能和通讯，其控制实现形式与前述专用的独立应用程序类似，如陈一民等人提出的基于网络的机器人控制系统<sup>[60]</sup>，采用的就是这样的控制形式；而后者是一个程序片段，依附于 Web，在 Web 页上运行，其控制实现形式为真正意义上的基于 Web 的控制。

本文提出的基于 Internet 的机器人控制系统设计方案采用 Java Applet 的实现方法，用户访问 Web 页，通过页面上的 Applet 进行仿真操作和实体控制。其主要特点为：

- 模块独立，易移植、扩展；
- 仿真图形效果逼真，操作方便；
- 必要下载量少，安全机制健全；
- 提供用户协商服务和图像监控服务

## 6.2 系统结构介绍

### 6.2.1 组成框架

综合考虑机器人控制系统的需求,将系统划分为这样几个组成部分:图像服务器、数据库服务器、机器人服务器、web 服务器和客户端应用,其组成结构见图 24。

基于 Internet 的机器人控制系统在服务器端,采用 SQLServer2000 作为数据库服务器,Weblogic 作为 Web 服务器。客户端必须具备支持 Java 运行的 Web 浏览器、支持 VRML 的浏览器插件,此外,如果选择使用图像观察服务,则还必须具备 JMF 运行环境和确保一定的带宽(在校园网环境下带宽约束可以忽略)。

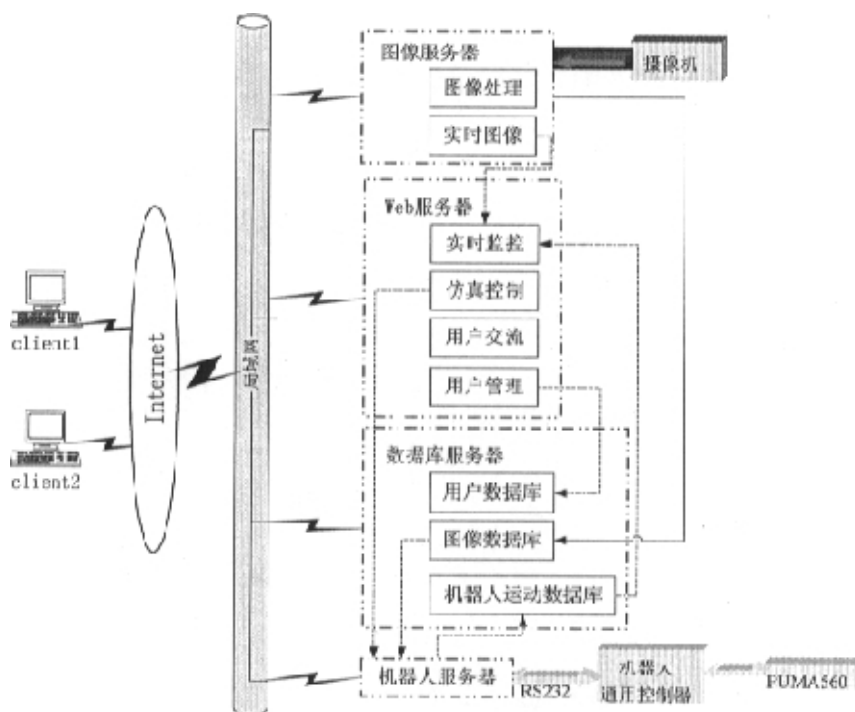


图 24 系统组成结构



## 6.2.2 功能模块

系统的各功能模块相互独立，接口规范，开发者可以很方便的进行更换、移植开发。各模块的主要功能为：（参见图 24）

### ● 图像服务器

包括图像采集处理和图像传输。

图像采集处理是通过摄像头获取机器人工作环境信息，通过计算机图形处理技术（辨识环境、重构环境、形成数据信息）向机器人控制系统提供外部环境信息，如机器人工作空间内的障碍物位置、姿态、范围等，将这些数据信息存储于数据库服务器。

作为机器人控制系统中的一个外部信息提供者和控制情况反馈者，图像的采集处理是机器人遥操作必不可少的环节（系统没有其他外部传感器存在）。其实现瓶颈在于数字图像处理的速度和数据能达到的精度。应用于机器人教学研究的控制系统，可以假设工作环境静态和一定程度上的已知，物体识别可以用“边缘检测一套用模型”的方法解决，主要工作在于坐标的确定。

图像传输是向远程用户提供实时的图像观察，以检查机器人控制命令的执行效果，了解机器人的实际运行情况，为远程用户打开一个现场窗口。考虑到网络的传输负担，此项服务是可选性服务，用户根据自身当时网络性能来决定是否选择图像的实时观察。同时，作为辅助手段，实时图像观察仅提供一定清晰度的图像流，并不追求高分辨率。

### ● 数据库服务器

包括用户数据库，图像数据库，机器人运动数据库。

用户数据库记载登陆用户的信息、权限和操作记录，配合用户登陆权限审核。

图像数据库保存数字图像处理的结果，供机器人服务器调用。

机器人运动数据库又分为运动记录库和运动规划库，以关节角度坐标或终端世界坐标两种形式存储，根据用户要求和操作需要，机器人服务器将机器人运动的数据存储于运动记录库，从运动规划库调用规划的运动数据，或将规划的运动数据写入运动规划库。远程用户还可以根据需要，通过 web 服务器调用、查看机器人运动数据。

### ● 机器人服务器

机器人服务器是机器人网络控制系统的核心部分,其控制的设计和组成可部分参考基于 PC 的机器人控制系统设计(见本文第五章)。两者的区别在于控制上层的接受方式不同和编程实现平台的不同。基于 Internet 的机器人控制系统其控制指令、参数、任务来源于远程客户端,主要利用 Java 技术实现,而在基于 PC 的机器人控制系统本机,控制任务本机接受本机执行,主要利用 C++编程实现。

机器人服务器的控制通道为:上层接受控制命令和控制参数,自身调用运算模块进行运算(坐标转换、路径规划、运动可行性检验),下层形成 VAL 命令,通过串口和机器人控制器实现通讯,操作机器人实体运动。

机器人服务器的反馈通道为:查询并接受机器人控制器的反馈信息,辨别信息类型,提取数据信息,写入运动记录数据库;将运算结果写入运动规划数据库并发运算完毕信号给上层任务提供者。

由上可见,其主要组成是机器人运动学计算模块、机器人控制器通讯模块。

### ● Web 服务器

包括图像监控窗口、三维图形仿真控制、用户管理、用户交流。

图像监控窗口如前所述为可选服务,接受图像服务器提供的图像流文件,根据网络状况和用户需求提供。

三维图形仿真模块中负责建立机器人的三维图形仿真模型(参见第四章)并对其进行控制。在图形环境下,远程用户可以观察检验控制参数变化对机器人运动的影响,是机器人控制的模拟运行,既有示教式按键运动模拟,又有程序执行运动模拟和任务级运动模拟,提供了路径设置的图形仿真功能。是机器人教学的一个生动平台,也为机器人运动研究提供了方便。

用户管理负责用户登陆的权限管理。为了确保机器人工作的安全,机器人网络控制系统采用“多观众单操作”的模式,一定时间内仅允许一个登陆用户拥有实体操作权限,其他用户只能作为观众,了解机器人运动的情况,查看机器人运动记录和状态。对于仿真模型的用户权限控制,分为单人控制和多人控制两种模式。单人控制模式下对仿真模型的控制是独立的,而多人控制模式时,控制权限为“请求—等待—获取—放弃”,仿真模型的控制效果为多人控制命令的累加。

用户交流负责多用户的文字对话,远程用户可以讨论控制方法、参数设定、



规划效果等机器人控制话题,提供控制建议,为不同地域的研究者实现了一个实时的研讨平台,促进学术交流。

各功能模块的信息调度传递关系参见图 24。

### 6.3 系统控制方式

在网络环境中对机器人进行控制,由于网络传输方式中不确定性时延、数据包丢失等固有问题的存在,远程用户不可能实现实时控制,因此在控制方式上,不能采用传统的直接控制和反馈控制,一般采用监督控制<sup>[61]</sup>(Supervisory Control)。在监督控制方式下,机器人系统的局部控制和人的操作控制共同完成控制任务。

远程的 Web 用户发出控制目标(任务)给本地服务器的操作系统,该指令由位于本地的系统控制模块完成运算、解释、执行过程,并将执行情况返回给 Web 用户。在监督控制方式下,选择算法、规划路径和执行运动命令都是本地服务器(机器人服务器)完成的,称其为系统的局部控制。远程用户的控制作用主要体现在任务的给出或最终目标点的设定上。

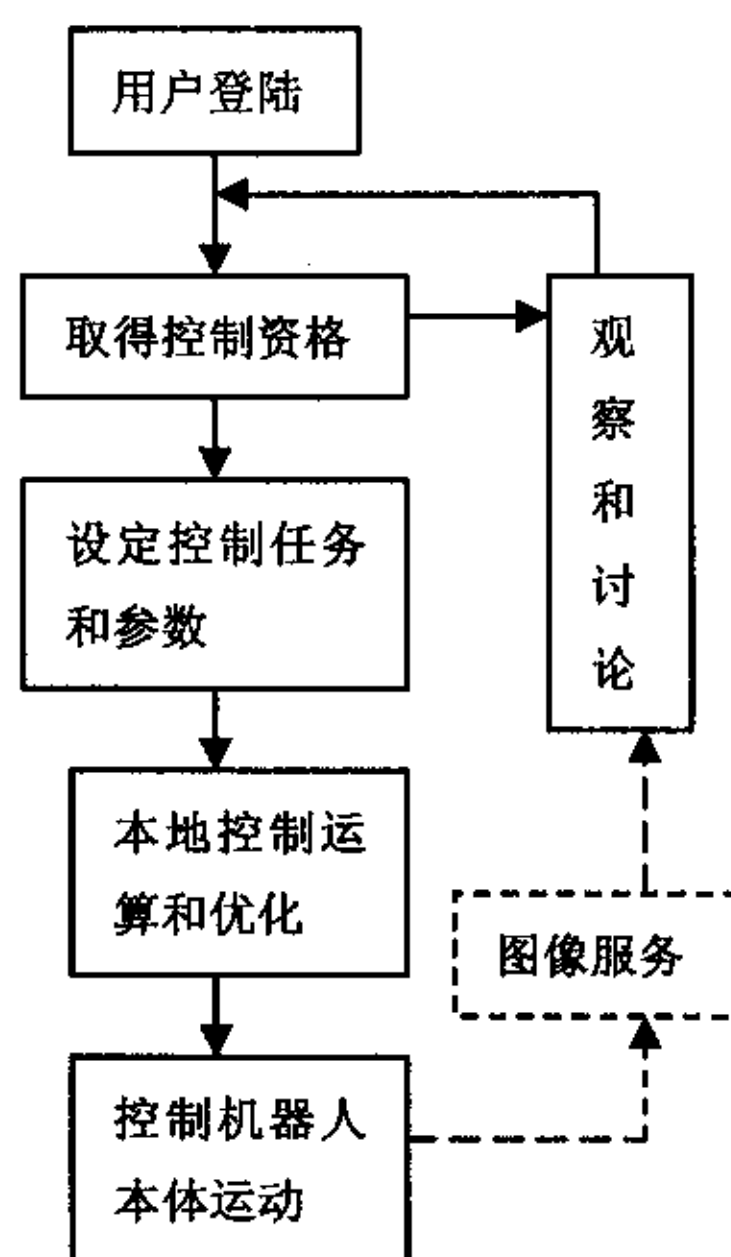


图 25 控制流程图

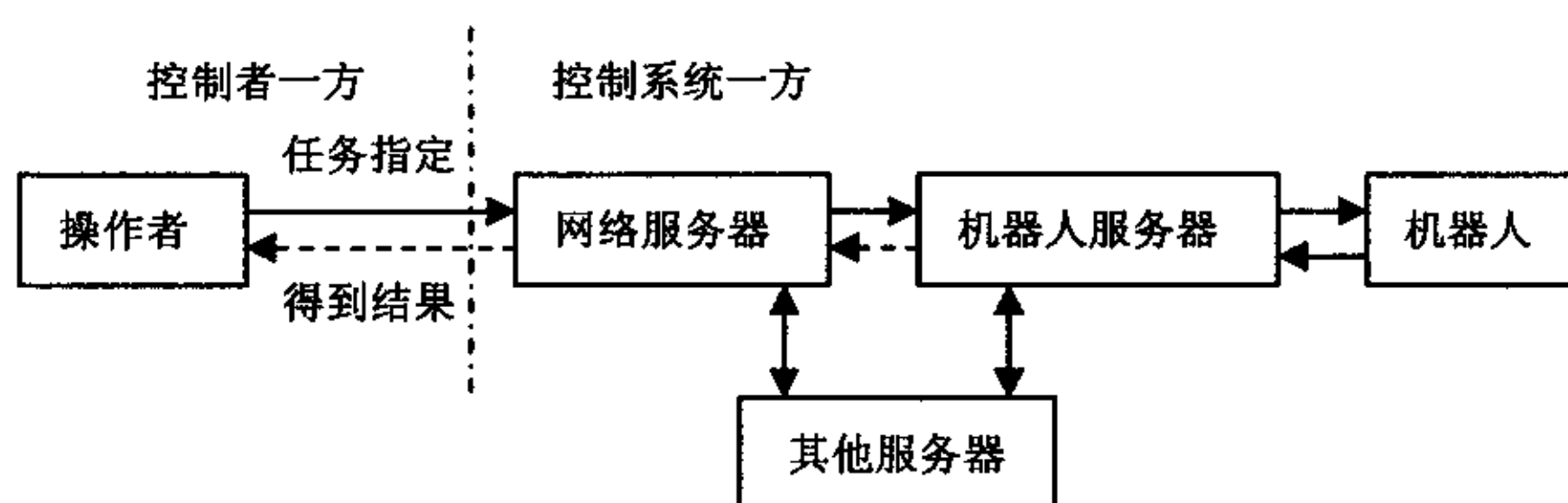


图 26 网络控制回环图

由控制流程图(图 25)和控制回环图(图 26)可见,控制方式的不同,使得基于 Internet 的机器人控制系统在设计中,比基于微机的控制系统,有更高的模块独立要求和通讯安全要求。

## 6.4 系统功能

### 6.4.1 控制界面

Web 世界非常丰富多彩,网络界面的开发可以利用很多功能强大的工具,如 Dreamweaver 等,作为一个机器人控制系统的 Web 形式,网络交互界面的要求是规范化,指引性。规范化可以帮助用户正确理解控制按钮、控制命令的含义,减少误会引起的错误操作。指引性是指在用户界面上,提供辅导帮助,用类似于 wizard 的操作引导,来帮助用户进行控制,选择合适的控制方式和网络服务。例如,正式登陆后提供对网络速度的测试,提醒用户选择或放弃实时图像监控服务,在控制过程中,设置了编程(命令)、仿真、结果查看、实体执行这样的单线执行步骤,引导普通用户进行正确的网络机器人操作控制。

#### ●用户协商的交互板块

用户协商的交互板块采用了通用的文字聊天交流形式。目前网络技术中,用户远程交流有多种实现手段,不但有图像、声音等的实时交流技术,也有 WebEx 等完善的网络会议解决方案。在机器人网络控制系统方案设计中,用户协商的服务是一项辅助功能,并不要求多媒体手段的丰富多样,主要用于学术交流讨论,简单的文字交流就可以满足用户需求。这样简单的设计能省略开发,也使得控制系统的主要工作集中于核心的控制功能上。

当然,如果用户不满足于简单的交流,需要传送文件,表达图形等,完全可以利用网络上其他成熟的交流工具如 MSN 等,作为控制讨论中的补充实现。这也是机器人控制系统在网络上实现的一个目的——网络资源向用户开放。

#### ●控制板块

控制板块由通讯台和程序文件选择组成,可参考基于 PC 的机器人控制系统的控制方式。

通讯台是使用文字框界面进行类 Dos 的操作,控制语言为 VALII,传送控制命令并观察机器人服务器送来的通用控制器反馈。

程序文件选择是选择已经编辑好的 VALII 语言程序进行实体控制,在机器人教学中,可以提供固定的几个实验程序文件,学生进行操作时,可以查看 VAL 程序代码,学习 VAL 程序的编写。

此外, 机器人网络系统控制还提供了简单命令集成板, 这一交互界面既提供实体控制, 也可运行于仿真模式, 用户通过此板块选择合适的命令如移动、复位、放开爪子等, 由机器人服务器负责用户命令的解释、数据运算并控制运动(仿真模型运动演示或实体手臂运动)。通过不同命令的组合, 用户可以完成不同的操作任务。

### ●图形板块

图形板块是机器人网络控制系统的三维图形仿真模块交互界面, 三维图形的实现参见第四章, 控制界面模仿机器人示教器, 提供关节角的逐个变化调节按键, 和机器人控制服务器相联系, 实现机器人图形仿真模块的各个功能(可参考第五章第4节)。

机器人网络控制系统的控制界面, 是用户使用控制系统的直接交互对象, 其操作的简便与否、界面的直观性直接影响到机器人网络控制系统的推广。在系统设计中, 不仅应该考虑到网络技术的实现可能, 也应该考虑到普通用户对于规定动作的操作要求和研究用户对于规划检验信息反馈的要求, 实现控制系统的教学目的和研究目的。

#### 6.4.2 网络控制下指令的传输与执行

在基于微机的控制系统中, 单机或两台微机(控制微机+图像监控微机)组成控制系统, 其主要的调度数据都能在本地操作系统中完成, 而网络环境下, 远程用户通过 Internet 发送命令并接受反馈, 其控制必然是滞后的, 数据调度在多台甚至不同操作系统间进行, 这要求其通讯机制不但能确保传输, 而且要提高调度效率, 采取合理的通讯结构和方法。

本系统的通讯要求和软件限制是:

1. 多机通讯, 有分散的多个合作对象;
2. 各功能模块代码独立, 用 Java 语言实现;
3. 计算模块和控制模块可以采用 C 语言实现;

因此, 可采用的基本通讯方法为远程方法调用<sup>[62]</sup>。

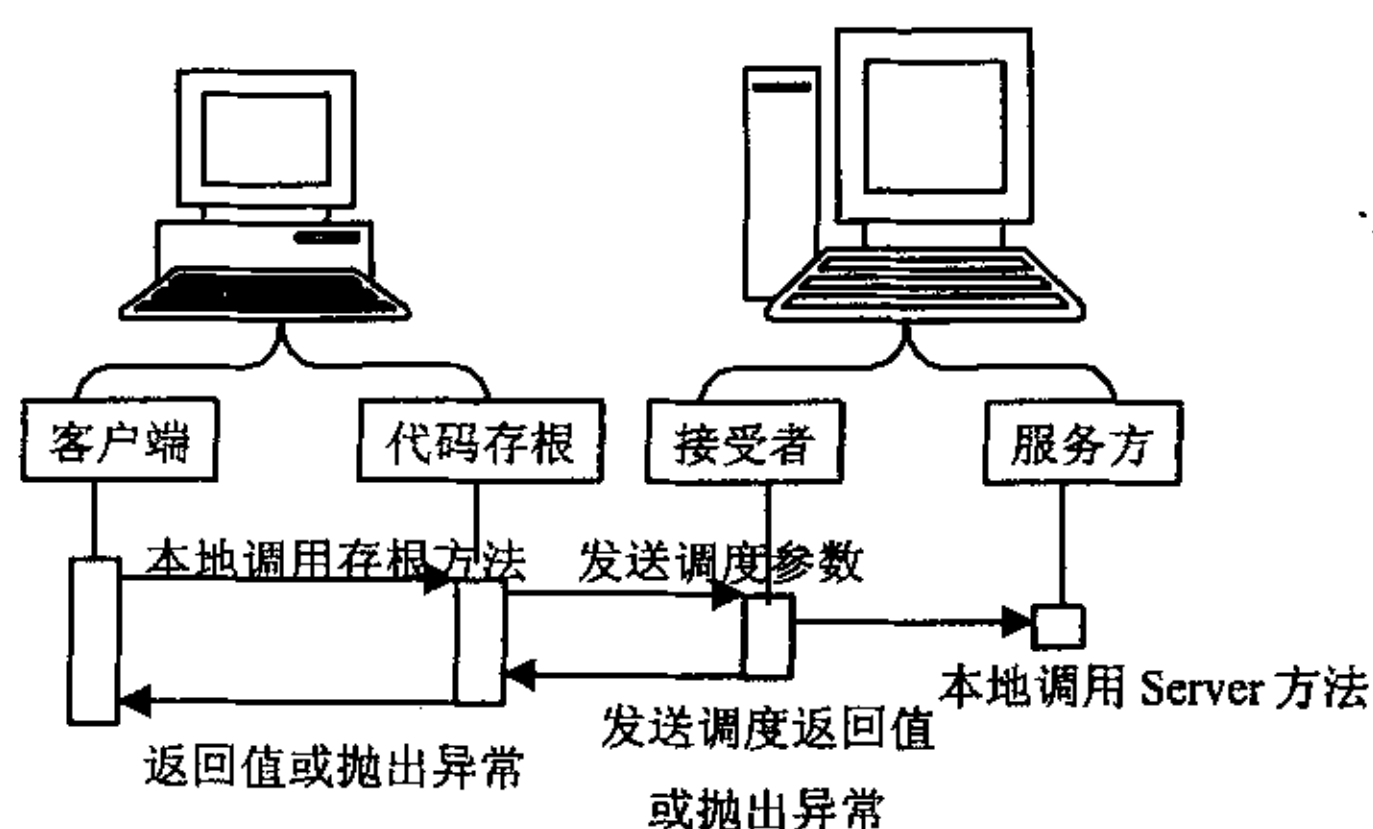


图 27 RMI 参数调度示意图

在采用 Java 实现的各个模块间，利用 RMI(Remote Method Invocation)<sup>[63]</sup> 机制实现通讯。RMI 是 Sun 专门为 Java 应用程序间通信开发的。它将远程方法封装在一个信息块——代码存根中。代码存根驻留在客户机上，接到用户

调用通知时，就将方法标示符和参数发往服务器。服务器上的接收者调用相应方法并发送返回数据包到代码存根。整个发送和返回过程是完全自动的，程序员无需了解其细节。如图 27 所示。

RMI 实现需要以下工作：客户的接口设定—>服务方法的实现—>RMI 注册服务对象（服务程序宣告各服务方法对象的存在，客户程序可以检索这些对象的代码存根）—>部署 RMI 程序。

当小应用程序(Applet)运行 RMI 时，还涉及到一个安全限制问题。由于 Applet 只能同它自己的主机建立网络联系，即只允许一个小应用程序在给它提供服务的主机读取和写入数据。因此，系统的 Web 服务器对象与网页必须驻留在同一主机上。同一服务器必须存储网页、小应用程序代码、代码存根类、服务对象、RMI 注册程序。

当同时存在多台服务器的时候，小应用程序要获得其他服务器上的数据、服务，则必须设法绕开它的安全设置，基本的解决方法是通过 Java 的 Servlet 技术实现代理服务器，这样可以解决 Applet 与服务方法不同机的问题。

如果希望利用外部的辅助程序（例如 Matlab）进行数据处理，或者用其他编程语言实现功能模块（例如，用 C 语言编写串口程序或控制程序），则需要采用公用对象请求代理机构 CORBA(Common Object Request Broker Architecture)，它可以让 Java 对象和其他语言编辑的对象进行调用。

CORBA 实现需要以下工作：用 IDL(Interface Definition Language)开发工作接口并调用相应语言的编译器生成代码存根，编译服务实现代码和客户代码，注册服务对象和部署程序。其中，IDL 是一种特殊的接口语言，用于制定语言中性

形式的接口。

采用远程方法调用的通讯机制可以有效的解决各个服务器之间、用户和服务器的信息传递，这是一种面向方法的通讯手段。

在传统的客户机/服务器模式中，客户请求被翻译成一种中间格式（如 XML 格式）或格式化为 HTML 格式，服务器分析这种格式并计算响应，将响应同样格式传回客户机，客户机分析并显示结果给用户。用这种传统的通讯模式，则对于结构化数据，存在着编码问题：必须找到一种适当的手段对数据和传输进行互译。而采用面向方法的远程方法调用可以很好的解决上述问题，适用于分布式系统内部的通讯。

此外，利用 JMS(Java Message Service)通讯也是一个很值得关注的解决方法。JMS 是一个 Java API，允许使用 Java 语言编写的程序与其他消息实现方式进行通讯，是一种面向消息的通讯机制。利用 JMS 实现机器人网络控制系统的信息传递，可用于有延迟的异步通讯控制，系统通讯自由（不要求同步，不要求知道服务方法）。

#### 6.4.3 功能模块的实现

系统各模块在实现技术上以 Java 编程为主，其中，机器人服务器的运算功能实现和数据库服务实现这两部分的软件结构，可参考本文第五章基于 PC 的机器人的控制系统模块，代码结构基本类似。在本方案的设计中，本地服务器程序采用了部分方法 C 语言实现，主体程序 Java 实现的综合编程，这样可以充分利用已有代码，避免重复性工作。

图像与 Web 服务的监控部分，采用了 JMF 技术<sup>[64]</sup>。JMF 是 Java 语言在多媒体领域的扩展，支持实时传输协议 RTP(Real-Time Transport Protocol)，使 Java 程序能够方便的实现捕捉音频、视频信号，存储、处理以及播放多媒体数据的功能。

在技术实现手段的选择上，本设计方案以功能满足、安全可靠为选取标准，采用较为成熟的网络编程技术，注意各个功能模块的相对独立性，基本能够满足机器人控制系统的控制和监控要求。



## 6.5 安全机制

由于利用网络作为信息交流载体, 基于 Internet 的机器人控制系统在设计中有一些特殊的情况, 比如, 大的不确定时延。控制命令的传输在网络上经过很多的节点, 路由选择不固定, 使得时延比较严重, 而且不确定, 预测和控制都比较困难; 数据包的丢失在网络忙碌和连接不稳定时非常多见, 普通的网络应用可以采用数据包重发解决, 但是重发并不能解决机器人控制的问题。

解决上述问题, 除了采用监督控制的控制方式, 提高本地服务器自我解决问题的能力, 减少网络控制带来的影响外, 还应该建立健全的安全机制, 当命令数据包发生问题时, 及时启动安全防护系统功能, 以避免机器人系统的硬件损伤。

基本的安全保障机制除了机器人本体系统中的安全控制(参考第五章第六节)外, 还采用了如下措施:

### ●用户准入检查

用户采用注册准入制度, 不允许匿名用户登陆, 保证用户的可信度, 并设置管理员, 进行用户授权和剔除工作。在控制进行中采用前述多用户单操作方式, 保证实体控制的一对一执行。

### ●控制流程中的监控

确保机器人控制系统的正常工作, 更多的是从控制流程上进行多重检验来进行的。这方面包括机器人服务器对于控制命令的格式检验、对控制参数的可行性检验等, 涉及到机器人控制语言解释、机器人运动学计算(见第二章)等多个模块, 此外, 引入图像实时监控也是一种辅助安全手段, 可参考第五章第六节的相关内容。在控制流程中的监控, 有软件上的检验, 如控制参数的运算结果合理性, 也有用户(控制人员)的参与, 如仿真动画的运动检验, 图像监视。

### ●其他一些解决方案

在本设计方案中, 系统应用背景为学校网络教学和实验室设备校园化应用, 因此, 对于大时延和数据包丢失问题没有特别加以重视, 没有特殊的针对性手段。如果机器人网络系统控制对于控制的响应执行有较高的要求, 则应增加相应的安全控制措施。

比如, 建立系统延迟的概率分布随机模型, 采用随机最优控制方法, 实现概率意义上的稳定控制<sup>[65]</sup>; R. Luck 提出建立现场和远程过程的数学模型, 进行 N

步预测的技术<sup>[66]</sup>；也有研究者提出，利用虚拟现实技术进行状态预测及环境预测仿真等<sup>[67]</sup>。针对不同的系统需要，设计者应该采用相应的措施来确保控制命令执行的安全性和正确性。

## 6.6 小结

作为机器人控制系统的进一步扩展，本文提出的基于 Internet 的机器人控制系统设计方案，综合利用 Java 网络技术，实现了机器人的远程控制和网络仿真实验，引入协同控制机制，使不同地域的研究者、学习者可以相互交流，结合网络控制特点，采用监督控制方式，并实现了较为完善的安全机制。

## 第七章 结论与展望

机器人控制系统的结构设计,是机器人系统实现的基础性工作,涉及到各个学科领域成果的综合创造性应用。在控制系统设计中,不仅要实现安全有效的机器控制,还需要满足机器人控制中特有的智能化要求和开放性要求。

本文提出的基于 PC 的机器人控制系统和基于 Internet 的机器人控制系统,是对传统的封闭式机器人控制系统的一种改进型设计,对于改造和利用现存的传统机器人控制系统具有很大的参考价值,基本实现了设计目标。

同时,在方案的设计中,体现了以下一些特点:

控制系统的可扩展能力增强,软件可换,硬件扩展方便,为机器人控制系统的改造和系统功能更新提供了方便。

控制系统的图形仿真能力突出,实现了机器人图形的三维仿真,实现成本低,技术普适性强,提供了一个形象直观的机器人控制检验平台。

在两个机器人系统中,系统的控制均可实现对于用户的模糊化,用户只负责任务级控制任务组合,而不必接触底层的机器人控制系统指令,可以极大地推广机器人控制系统的应用。

对于机器人学研究者,这两个设计方案均提供了开放的技术开发平台,可以进行基本操作实验,进行 VAL 程序控制和 C 语言控制算法研究,也可对系统做进一步扩展,加入自己的算法实现或实现其他辅助系统(传感器系统)。

基于微机的机器人控制系统和基于 Internet 的机器人控制系统两个方案都是根据开放性机器人系统结构思想设计的,后者是对前者应用上的延伸和扩展,通过网络手段,打破了机器人应用中地域上的限制。

本文提出的系统设计方案,充分考虑了系统的应用需求,综合运用各学科知识,实现了合理的结构设计和功能划分,并指出了可以实现的技术手段,对其中的重要部分,如运动学运算的实现、仿真技术的应用、网络环境下的结构设计等进行了详细的分析展开。

随着科学技术的不断发展,机器人学的内容、领域、应用也在不断的发生变化。机器人运动实现的关键——机器人控制系统设计,也需要不断更新。



机器人控制系统在机器人工作环境的未知及不确定性处理，传感器信息融合，人与机器人的交互，机器人自我决策控制，机器人运动控制等各个方面，都有大量的问题需要研究，而在结构设计和功能设计上，也需要更具开放性，更具智能性，不断扩大机器人运用领域。

## 参考文献

- [1] 李开生, 张慧慧, 费仁元, 宗光华. 机器人控制器体系结构研究的现状和发展. 机器人, 2000, No. 3
- [2] 周学才等. 开放式机器人通用控制系统. 机器人, 1998, 20 (1)
- [3] 范永、谭民. 机器人控制器的现状及展望. 机器人, 1999, No. 1
- [4] William E. Ford. .What is an Open Architecture Robot Controller. IEEE International Symposium on Intelligent Control, 1994
- [5] Albus, J. S. , McCain, H. G, Lumia, R. . NASA/NBS Standard Reference Model for Telerobot Control System Architecture (NASREM) . NIST Technical Note 1235, April 1989
- [6] Martin Marietta. Draft Volume I of Next Generation Workstation/Machine Controller (NGC) Specification for an Open System Architecture Standard (SOSAS) . Document No.NGC-0001-13-000-SYS, March 1992
- [7] Saridis N. N. . Architecture of Intelligent Control. IEEE Symposium on Implicit and Nonlinear Systems. Ft. Worth. TX, December, 1992,
- [8] Stewort D. B. , Volpe R. A. , Khosla P. K. . Integration of Real-time Software Modules for Reconfigurable Sensor-based Control System. Proceedings 1992 IEEE/RSJ International Conference on Intelligent Robots and Systems, Raleigh, NC, 1992, July 7,10
- [9] Stewort D. B. , Volpe R. A. , Khosla P. K. . Design of dynamically Reconfigurable Real-time Software using Port based Objects. IEEE Transaction on Software Engineering, 1993, July 1
- [10] Miller D. J. , Lennox R. C. . An Object-oriented Environment for Robot System Architectures. Proc of IEEE Int Conf Robotics and Automation,1990.1: 352-361
- [11] 赵东波等. 机器人离线编程系统的研究. 机器人, 1997, No. 4
- [12] Honey W. E. , Jamshidi M. ROBO-SIM. A Robotics Simulation Environment on Personal Computers. Robotics and Autonomous Systems, 1992,9:305-317
- [13] 陈启军, 陈辉堂, 王月娟. 基于 Internet 的机器人控制技术及其应用. 电气自动化, 2000. 1
- [14] Goldberg K. , Mascha M. , Gentner S. . Desktop teleoperation via the world wide web. International conference on robotics and automation, Nagoya: Omnipress, 1995: 654-659
- [15] Taylor K. , Dalton B. , Trevelyan J. . Web-based telerobotics. Robotics, 1999, 17: 49-57
- [16] Stein M. R. . One year of PUMA painting: Site experience. SPIE Telemanipulator and Telepresence Technologies VI , Boston, Omnipress, 1999: 200-209
- [17] Backes P. G. , TSO K. S. . Internet-based operations for mars polar lander mission. International Conference on Robotics and Automation , San Francisco, Omnipress, 2000:2025-2032
- [18] 诸静. 机器人与控制技术. 浙江大学出版社, 1991. 10
- [19] Elgazzar S. . Efficient Kinematic transformations for the PUMA560 robot. IEEE Journal of Robotics and Automation, 1985,1 (3) :142-151
- [20] Lee G. S. G. , Ziegler M. . Geometric approach in solving inverse kinematics of PUMA robots. IEEE Trans AES, 1984,20 (6) :695-706
- [21] Paul R. P. , Shimano B. E. , Mayer G. . Kinematic Control equations for simple

- manipulators. IEEE Trans SMC, 1981,11 (6) :449-455
- [22] Paul R. P. . Robot Manipulator:Mathematics, Programmings and Control. Cambridge, MIT Press, 1981
- [23] Dinesh Mancha, John F. Canny. Efficient Inverse Kinematics for General 6R Manipulators. IEEE Transactions on Robotics and Automation, 1994,10 (5) :648-657
- [24] 董明晓, 陈美华, 赵志超等. PUMA 机器人逆运动学分析. 山东建筑工程学院学报, 2000. 12, Vol. 15, No. 4
- [25] Matlab, <http://www.mathworks.com>
- [26] The Robotics Toolbox (release 7.0), <http://www.cat.csiro.au/cmst/staff/pic/robot>
- [27] Khatib, O. . Real-time Obstacle Avoidance for Manipulators and Mobile Robots. IEEE Int. Conf. On Robotics and Automation, March 1985: 500-505
- [28] 马晨宇, 李伟. 机器人实时运动规划问题. 电子技术运用, 1996. 12
- [29] 苏剑波, 席裕庚. 机器人动态抓取的变比例导引路径规划方法. 控制理论与应用, 2000. 6
- [30] 杜树新, 唐建中. 机器人运动的离散轨迹规划法. 西安交通大学学报, 1994, 28 (1) : 61-67
- [31] Dassanayake P. , Watanabe K., and Izumi K. . Fuzzy Behavior-Based Control of a Task of Three-Link Manipulator with Obstacle Avoidance. J. of Robotics and Mechatronics, 1999, Vol. 11, No. 6 :502-509
- [32] Nearchou A. C. , Aspragathos N. A. . Collision-Free Cartesian Trajectory Generation using Raster Scanning and Genetic Algorithms. J. of Intelligent and Robotic Systems, 1998, Vol. 23, Nos. 2-4: 351-377
- [33] Palitha Dassanayake, Keigo Watanabe, Kazuo Kiguchi, Kiyotaka Izumi. Fuzzy Behavior-Based Motion Planning for the PUMA Robot. Procs. of the 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems
- [34] 陈国军, 叶飞跃, 杨黎明. 三维图形生成支撑环境综述. 计算机应用研究, 2001. 5
- [35] Que Corporation . 3D Graphics Programming with OpenGL
- [36] Microsoft DirectX SDK 7.0. DirectX SDK Help
- [37] Glide 3.0 Programming Guide. 3Dfx Interactive, Inc., 1998.6
- [38] 乔林, 费广正等. OpenGL 程序设计. 清华大学出版社, 2000
- [39] 江早, 王洪成. OpenGL VC/VB 图形编程. 科学出版社, 2001
- [40] 孙怀安, 杨广平. 基于虚拟现实语言的机器人三维仿真系统软件. 西安电子科技大学学报 (自然科学版), 2001. 6
- [41] Martin Rohrmeier. Web based robot simulation using VRML. Proceedings of the 2000 Winter Simulation Conference
- [42] Melinda F. Robinette, Rachid Manseur. ROBOT-DRAW, an Internet-Based Visualization Tool for Robotics Education. IEEE Transactions on education, February, 2001, VOL. 44, NO.1
- [43] 阳化冰, 刘忠丽等. 虚拟现实构造语言 VRML. 北京航空航天大学出版社, 2000
- [44] The Inventor Mentor. <http://www.sgi.com>
- [45] ParallelGraphics Cortona Software Developers Kit. <http://www.parallelgraphics.com>
- [46] VRML2.0Robot. <http://www.robotic.dlr.de/STUDENTS/Martin.Rohrmeier/robot/robot.html>
- [47] TEF (Technology Exposure Facility) . [www.estec.esa.nl/tef/wat/tef/intro.html](http://www.estec.esa.nl/tef/wat/tef/intro.html)
- [48] Cay S. Horstmann, Gary Cornal. Core Java2 Volume I : Fundamentals. Prentice Hall, 2000

- [49] robot simulation. <http://prt.fernuni-hagen.de/pro/richodl/simulation.html>
- [50] Hironori Hiraishi, Hayato Ohwada, Fumio Mizoguchi. Web-based Communication and Control for Multiagent robots. Proceeding of the 1998 IEEE/RSJ Intl. Conference on Intelligent Robots and Systems, Victoria.B.C.,Canada,October 1998: 120-125
- [51] Mohamad Fauzi Zakaria, Shamsudin H.M Amin, Rosbi Mamat. Design and Development of Control System for Internet-based Telerobotics. Intelligent Systems and Technologies for the next millenium, Kuala Lumpur Malaysia, 2000.9
- [52] 汪地, 陈一民, 方明伦, 何永义. 一种新型的基于微机的图形仿真与监控系统的研究. Proceedings of the 3rd World Congress on Intelligent Control and Automation, June28—July2, Hefei, 2000
- [53] EASY-ROB Robot Simulation. <http://www.easy-rob.de>
- [54] Workspace. <http://www.workspace5.com>
- [55] RoboWorks. <http://www.newtonium.com>
- [56] Shimano B. E. . VAL-II: A New Robot Control System for Automatic Manufacturing. Proc Intl Conf on Robotic, Atlanta, USA,1984: 278-293
- [57] 戴齐, 姚先启. 机器人程序设计语言. 机器人, 1997. 9, Vol. 19, No. 5
- [58] Australia's telerobot. <http://telerobot.mech.uwa.edu.au/>
- [59] CSC Telerobot. <http://csc.clpgh.org/telerobt/>
- [60] 陈一民, 张涛等. 基于网络和 Java3D 的机器人仿真与监控系统. Proceedings of the 4<sup>th</sup> World Congress on Intelligent Control and Automation, Shanghai, 2002
- [61] Sheridan T B. . Space Teleoperation Through Time Delay Review and prognosis. IEEE Transactions on Robotics and Automation, 1993, 9 (5) : 592-606
- [62] Songmin Jia, Kunikatsu Takase. An Internet Robotic System Based Common Object Reques Broker Architecture. Proceedings of the 2001 IEEE International Conference on Robotics & Automation
- [63] Cay S. Horstmann, Gary Cornal. Core Java2 Volume II : Advanced Features. Prentice Hall, 2000
- [64] JavaTM Media Framework API Guide. Sun Microsystems, Inc,1999
- [65] 于之训, 陈辉堂, 王月娟. 时延网络控制系统均方指数稳定的研究. 控制与决策, 2000, Vol. 15:278-282
- [66] Luck R. , Ray A. . Experimental verification of a delay compensation algorithm for integrated communication and control system. Int. J. Of Control, Vol. 59:1357-1372
- [67] Schulz D., Burgard W. , Cremers A. B. . Predictive simulation of autonomous robot for tele-operation systems using the World Wide Web. In Proc. Of International conference on IROS: Workshop on the Web Robots, 1998

## 致 谢

首先衷心感谢颜文俊导师多年来在学习、工作上给予的指导和关怀，从课题的确定到研究方法的选择都给了我极大的帮助。他对科研工作认真严肃、精益求精的态度以及踏实勤恳的工作精神都给我留下了深刻的印象，这些无疑将成为我受益终生的宝贵财富。

在硕士学习期间，诸静教授给了我很多帮助，启发并指导我完成各项科研任务。还有姚维老师、王景美老师等许多老师，在工作生活中给予我无微不至的关怀，在此也向他们表示由衷的感谢。

感谢师兄、师弟和师妹们在学习工作上的默契配合和大力支持。特别要感谢邹细勇、夏伟、陈利红、章捷、毛剑飞和李会猛等几位师兄弟，感谢他们在整个科研和论文写作阶段对我的帮助和支持。

衷心感谢所有曾关心我、帮助我的师长、朋友和同学们，你们使我在浙大近七年的求学经历更加美好，终生难忘。

最后，我要深深感谢我的家人，感谢他们多年来所给予我的无私关爱和支持。

徐 益

2003年2月于求是园

## 附录 作者攻读硕士研究生期间完成的论文

- [1] 徐益 颜文俊 诸静, 基于微机的机器人控制系统设计, Proceedings of the 4<sup>th</sup> World Congress on Intelligent Control and Automation, Shanghai, 2002
- [2] 徐益 颜文俊 诸静, 机器人三维图形仿真的实现, 计算机仿真, 2002
- [3] 徐益 颜文俊, NDIS 小端口驱动程序的开发, 计算机应用研究, 2003
- [4] 夏伟 徐益 诸静, 基于 Internet 的 Java 使能的机器人遥操作系统的研究, 计算机工程与应用, 2003



相似文献(10条)

1. 学位论文 罗玮韬 气门电蚀成型工序机器人控制系统的研究 2009

气门电蚀成型工序机器人属于点位控制上下料机器人,是具有环境感知能力、离线编程控制的第二代工业机器人。它在无人工干预的前提下,实现气门电蚀、压力成型环节的自动生产。<br>本文以“离线编程控制的工业上下料机器人控制系统”为研究目标,采用上、下位机一级分布式结构,结合气门电蚀成型工序中的设备状况与功能要求,设计了控制系统的硬件平台。<br>按照“分布式控制”的编程思想,规划了机器人上位机程序的功能模块。对状态监测模块、模式选择模块的实现方式进行说明,描述了基于VC++6.0多线程编程技术下的程序流程及构造。给出了上位机程序的总体结构图。<br>针对本机器人的实用性及通用性,提出了一种适用于工业机器人的离线编程代码——P代码,介绍了P代码的关键词及其功能,阐述了使用P代码编程的语法规则与程序架构。<br>重点研究了基于Boost、Regex正则表达式库的P代码编译器,详细介绍了编译器的构造、流程以及各目标代码的处理方法,分析了各个功能模块之间的关系,给出了各子模块程序的结构与流程图。<br>机器人控制系统通过查询PLC软元件状态来采集工况环境信息,介绍了PC与三菱Fx系列PLC串口通信的协议,说明了实现通信的步骤、参数计算等内容,给出了基于VC++6.0的程序代码。

2. 期刊论文 李成伟, 负超, LI Cheng-Wei, YUN Chao 码垛机器人控制系统的设计与实现 -机电产品开发与创新2008, 21 (5)

介绍了混线机构的搬运码垛机器人控制系统的设计与实现方法,根据工作现场的实际需求,对离线码垛过程进行了研究,通过码垛关键参数的输入和码垛空间判断,建立离线编程的基础,给出了机器人码垛的路径规划,硬件控制系统采用工业PC(IPC)作为主计算机,利用分布式二级控制结构实现系统的监控和作业管理,协调各关节的运动,准确地跟踪轨迹规划,软件系统运用了PLC来控制机械人手爪且自主开发码垛机器人控制软件,通过码垛关键参数的输入及示教盒控制方式,经实际使用结果证明控制系统的有效性合理性。

3. 学位论文 孙雷 开放式高压带电机器人控制器的研究 2002

该文从开放性和可靠性的角度出发,设计了使用以DSP为核心的PMAC(可编程多轴运动控制卡)和工业控制计算机组成的高压带电机器人控制系统,首先,该文设计的高压带电机器人具有主从式和编程式两种运动方式的空间六自由度机器人,设计了开放式机器人控制系统的硬件结构,阐述了控制系统的工程原理,并利用运动学自动建模技术建立了机器人运动学模型,并推导了机器人运动学正、反解算法。其次,该文全面系统的设计了控制系统软件结构,采用了一种模块式的程序结构,将程序设计成为几个互斥的程序模块,并着重介绍了软件的主控制程序模块和通讯程度模块,利用面向对象方法设计了运动控制函数库和利用样条函数规划机器人运动的方法,最后,该文详细地介绍了一个基于Windows NT的人机交互的机器人离线编程系统的原理,该系统采用视窗界面能对机器人系统进行编程。

4. 学位论文 吴绍凤 焊接机器人离线编程系统接口设计与开发 2007

本文论述了焊接机器人离线编程系统接口设计与开发。  
工业机器人的发展经历了示教再现和离线编程两个阶段。近年来,以离线编程为基础的机器人柔性自动化系统的开发,代表了机器人系统二次开发的方向。在前人研究成果的基础上,结合汽车工业中与机器人相关的实际应用研究项目,针对本实验室早期开发的基于PC的机器人三维可视化离线编程与仿真系统WinROBWeid进行了二次开发的。  
WinROBWeid系统在开发之初是针对ABB公司的I182000机器人的,它使用的是S3控制系统和ARL4编程语言,而目前实验室使用的是ABB公司的I182140机器人,它采用的是S4Cplus控制系统和RAPID编程语言,因此系统存在重新建模、程序转换以及与机器人通讯等方面的匹配问题,使该系统一直没有在实验室目前的I182140机器人上得到应用。由于实际生产中的项目研究需要,必须使该系统能够实现I182140机器人的离线编程,针对上述情况,在研究S4Cplus机器人控制系统和RAPID编程语言的基础上,对WinROBWeid离线编程系统进行了重新开发,并重新建立了以下三个功能模块:工件3D建模、通信模块及程序转换模块。  
工件的3D建模分两步实现,首先在其他CAD软件中将工件以VRML格式导出,其次再把工件以VRML格式转换到系统所支持工作格式。这是因为VRML格式是目前多数CAD软件均支持的输出格式之一,且其代码及相关协议均公开。  
在重点分析和研究VRML2.0结构及语法的基础上,通过重新设计转换算法,设计开发了新的工件3D模型文件转换工具,并加入了转换工具对定位参数输入功能的支持。  
通讯模块用来实现系统与机器人的通信,采用两种通讯技术,一种是传统计算机与机器人之间的通信技术,另一种是基于C/S模型的计算机与机器人之间的网络通信技术。前者主要用来实现系统的文件管理及机器人简单控制,比如程序的上传与下载,启动机器人程序的执行等。后者可省去程序转换过程,提高编程效率,方便对程序运行的实时监控。采用基于C/S模型与ABB WebWare SDK,编制了网络通信接口函数,解决了基于串行线路接口协议的通信功能薄弱与不完整、实时性差等问题,并可以通过机器人控制器控制激光器、TIG及MIG等加工设备。  
程序转换模块用来完成系统程序对特定类型机器人语言的程序转换,本文在分析与研究RAPID语言结构及语法规则的基础上,重新开发了系统的程序转换模块,扩充与完善了该模块的功能,使其能同时支持ARL4和RAPID语言。  
最后,结合汽车工业的某研究项目,针对某汽车后车盖的空间3D模型进行了系统离线编程与仿真试验,实现了从工件3D建模、离线编程、程序仿真、程序转换、程序下载运行到在线监视等整个过程。

5. 学位论文 秦志强 精密装配机器人新型控制系统的研制和虚拟装配 2000

该文的主要研究工作是围绕开发新的精密装配机器人控制系统而展开,对“精密1号”机器人控制系统原理进行了深入的研究和剖析,弄清了直接驱动电机的控制原理和电机功率驱动器的工作原理,在此基础上,用一台工控机代替原有控制系统的下位机,在DOS环境下采用定时器中断编制了机器人实时控制软件,在微机Windows环境下为“精密1号”机器人开发了离线编程系统,采用结构化机器人语言对机器人装配任务进行编程,经过词法分析、语法分析后,解释、规划生成机器人关节运动数据文件,然后将关节数据下传给实时控制软件完成装配操作任务,利用PC机打印口实现了上下位微机的并行通讯模块,提出了一种全新的单总方式接收数据的方法,深入分析了数据传输过程中发送、接收方的应答过程和时序,并通过详细的时序分析证明了该方法相对传统的双总方式接收数据的优越性,研究了虚拟装配系统的体系结构和相关基础知识,以“精密1号”装配机器人作为交互设备,提出了一个建立虚拟装配环境的实施方案,详细研究虚拟装配系统的部分基础关键技术,如虚拟环境的几何建模和物理建模方法,虚拟环境的不确定性建模方法,以及虚拟接触状态的建模方法等,以点接触型为基础,提出了虚拟装配操作过程的抓取分析和优化算法,用MATLAB软件包证了算法,针对装配机器人技术的发展方向和“精密1号”机器人新控制系统的现有水平,指出了未来的开发方向。

6. 期刊论文 唐新华, Paul Drews, Paul Drews 机器人三维可视化离线编程和仿真系统 -焊接学报2005, 26 (2)

提出了一种基于PC的交互式三维可视化离线编程和动态仿真系统,利用目前流行的PC机和Windows操作系统所支持的OpenGL三维图形功能,在系统中实现了焊接机器人及其工件的三维几何建模和机器人运动学建模,并研究了机器人运动轨迹的自动规划和编程以及它的图形化三维动态仿真,采用交互式三维“虚拟示教”方式实现了机器人单道焊接路径的规划和编程,对多层多道焊,则通过采用“宏”技术实现机器人焊接运动轨迹的自动规划和编程,通过图形化三维动态仿真实现了向机器人程序及运动轨迹的可靠性和安全性验证,离线编制的程序在向机器人控制系统传送的过程中实现了向机器人语言的自动转换,从而使机器人离线编程的实用性得到大大提高。

7. 学位论文 冯国平 机械式挖掘机的动力学分析与智能化设计 2006

挖掘机是重要的工程机械之一,可连续高效的工作,对于减轻劳动强度,提高劳动效率起了很大的作用,被广泛应用于建筑、化工、水利、矿山等部门,是工程建设项目中不可缺少的施工工具。随着人类活动领域的拓展,实际工程中对工作环境的要求越来越高,同时人对的作业环境的要求也更加严格,于是在应用领域中存在的一些问题也愈来愈不容忽略。然而,国内现有的挖掘机基本上采用人工操作方式,因此进行挖掘机的自动化、智能化研究,利用挖掘机机器人替人进行自主作业显得非常必要而紧迫。近年来,机器人学、电子技术、控制技术、计算机技术、人工智能等相关学科和技术的发展为挖掘机的自动化、智能化研究展现了广阔的前景。通过挖掘机器人模型的建立,可以得出挖掘机工作装置的运动学和动力学方程以及工作轨迹,然后利用计算机控制技术、传感技术、智能控制等技术设计出一套针对挖掘机的智能化控制系统,利用计算机对其进行离线编程编程控制,这将大大改善挖掘机在轨迹规划以及动力学等方面的性能,较好地实现挖掘机自动化、智能化作业。

本文在综合分析了挖掘机机器人研究的历史和现状后,针对本实验室要建立挖掘机机器人智能化实验平台需要解决的问题,进行了系统深入地研究分析,主要涉及以下几方面内容:  
(1)挖掘机器人模型的建立。根据机器人模型的建立方法,结合实验室WS-005机械式挖掘机的结构及运动过程分析,建立挖掘机器人模型。  
(2)挖掘机器人的运动学分析及轨迹规划。根据建立的机器人模型,求出了运动公式,结合工程机械作业特点,探讨了轨迹规划的方法,并给出了一次挖掘过程中的规划结果。  
(3)挖掘机器人的动力学分析。结合模型,运用拉格朗日法求出挖掘机器人的动力学方程,并分析了一次挖掘过程中所受的外力,进而给出了该过程中动力学结果。  
(4)挖掘机器人控制系统的设计。根据得出的机器人动力学方程,可以很容易的求出机器人的控制算法,从监控系统得到的数据,经工控机编写的控制算法求出各种工况下的控制数据,然后由PLC控制系统以及模拟量输出系统对机器人进行具体控制。

8. 学位论文 张长春 具有感知的SCARA型机器人控制及编程系统研究 1994

该论文结合国家“863”高科技计划项目进行,论文首先对国内外装配机器人和机器人柔性装配技术进行了综述,在广泛的工厂调研基础上对中小型机电产品的装配工艺进行了分析,对比国外情况总结了中小型机电产品装配工艺特性的一般规律,该文主要工作是适于中小型机电产品装配工艺特性的一般规律,该文主要工作是适于中小型机电产品装配的SCARA机器人控制系统设计,该论文论述了机器人控制系统的体系结构,介绍了机器人控制系统硬件软件设计方法,分析了控制系统的抗干扰问题,所研制的机器人控制系统已经应用于机器人柔性装配单元中,并经过一年多的实际考验,文章介绍了SCARA装配机器人工作区域的重复定位精度测量和分析方法,给出了测量结果,论文重点论述了机器人离线编程系统和机器人语言的设计方法,开发了具有力感知的机器人编程语言,并介绍了用此机器人编程语言实现的力觉引导主动柔顺的轴孔插入装配作业实验。

9. 学位论文 沈清 教学机器人三维可视化离线编程与远程控制 2006

随着工业机器人的广泛应用,机器人仿真技术也越来越被人们所重视。在缺少机器人实体的情况下,通过仿真技术能够完成机器人学习和研究工作。由于大多数工业机器人控制系统是不开放的,只能利用其自带的示教盒完成一些简单的示教仿真和离线编程。同时,大多数仿真软件通过串口与控制器相连的通讯模式也影响了通讯的稳定性,限制了通讯距离。

本文结合上海交通大学工程训练中心自主研发的5自由度教学机器人EduRobot-III,研究其三维仿真控制软件设计以及基于C/S结构的远程控制实现方法,编写的上层仿真控制软件实现了EduRobot-III的三维仿真、离线编程与远程控制等功能。  
论文具体研究以下几方面内容:  
1. 在简要介绍EduRobot-III的机械结构基础上,给出了针对该机器人的简便运动学逆解算法。根据离线编程语言格式,探讨了计算机语言实现教学机器人关节插补、空间直线插补和多指令衔接等问题。  
2. 介绍了在MFC环境下,调用OpenGL软件包实现EduRobot-III的三维建模、动画方法。结合具体的图形实例,探讨了OpenGL函数库调用、利用Windows定时器实现动画仿真方法。  
3. 给出了本文编制的仿真控制软件的结构及其实现方法,详述了离线编程语言的格式和使用,并通过几个示教动作举例说明了离线编程语言的使用方法。介绍了WinSocket控件的特点和使用,以及基于C/S结构的远程任务文件传输实现。  
4. 通过比较仿真效果和软件演示生成的插补数值文件在EduRobot-III的示教实验平台上的实验结果,表明本仿真控制软件能够满足EduRobot-III的仿真示教要求,不仅可以完全代替传统的示教盒完成离线编程控制,而且编程更方便,功能更为强大。并且该系统的研究为机器人仿真技术的研究提供了一个较好的范例。

10. 期刊论文 丁度坤, 谢存禧, 张铁, 邹焱鹰, DING Dukun, XIE Cunxi, ZHANG Tie, ZOU Yanbiao 喷涂机器人控制系统的开发与研究 -机床与液压2009, 37 (3)

对喷涂机器人技术进行了研究,设计并研制了喷涂机器人控制系统,介绍了控制系统的硬件与软件结构,对喷涂机器人系统的示教和离线编程两种模式的工作流程进行了叙述,同时设计了机器人各个关节的PID控制器,并对所设计的PID控制器进行了仿真试验,试验结果表明:所设计的控制器响应速度快,无超调,基本可满足工业需要。

引证文献(1条)

1. 高爽 基于DSP的仿生步行机器人平台开发与研究[学位论文]硕士 2006