



Y1436007

内部编号:

分类号:

授学位单位编号:10011

密级:

北京工商大学

硕士研究生学位论文

论文题目: 基于虚拟现实技术的遥操作机器人协调控制
系统研究

TITLE: Research on Teleoperator Robot Coordinated
Control System Based on Virtual Reality

学 科: 工 学

专 业: 机械制造及自动化

学生姓名: 战晓磊

指导教师: 辛洪兵 (职称) 教授

(职称)

日 期: 2008 年 5 月

摘 要

遥操作机器人是一类非常实用的机器人,基于网络的机器人系统结合了传统的机器人控制技术与先进的网络通讯技术,广泛应用于危险环境下的远程作业、远程医疗、远程教学、远程监护以及传统生产模式的改造等领域。基于虚拟现实技术的遥操作机器人系统,将虚拟现实技术应用到遥操作机器人上,有效的解决了遥操作机器人系统中存在的某些问题。

本文首先阐述了国内外机器人的研究状况,并介绍了遥操作机器人系统的各阶段类型,以及最新的研究状况。对国内外现有的基于 Internet 的远程控制机器人系统软件框架、控制方法和网络通信协议等方面进行了详细的分析和研究。

在此基础上,本文研究了基于 Solidworks-VRML 的虚拟建模方法。采用 Transform 节点分层的方法引入各实体模型的 VRML 文件,使 VRML 程序设计模块化,从而有利于对组成虚拟机器人的各实体模型再进行脚本编程,实现运动控制的虚拟仿真。

本系统以首钢莫托曼机器人 MOTOMAN-HP3 为实验对象,设计了基于客户端/服务器端(C/S)的分布式网络控制系统,结合机器人运动学和虚拟工具箱,构建了基于 MATLAB/Simulink 的虚拟现实仿真平台,通过基于 TCP 的机器人数据控制协议,来实现客户端和服务端的数据通信,并最终实现了基于虚拟现实技术的遥操作机器人协调控制系统。

关键词: 机器人, 遥操作机器人, 虚拟现实, 控制系统, Matlab/Simulink

abstract

A teleoperator is a kind of robot, the robot system based on Internet integrates traditional robot controlling technology and advanced network communication technology systematically. It can be applied in long range projects, medicine, teaching, monitoring in dangerous environment, and in reconstruction of traditional produce mode and so on. Remote operation based on virtual reality(VR) technology, as a good resolution, combine the VR with the teleoperator in order to overcome the difficulties in remote controlling.

This paper first elaborates the research condition of remote control robot system based on the internet domestic and abroad, introduces the typical model robot system for each stage and the latest research condition of this aspect. Then it analyzes the framework of the remote control robot system based on internet, controlling method, research categorization and network communication protocol internationally.

On this foundation, this paper studies the methods of virtual model building based on Solidworks-VRML. Every model's VRML files are assembled by layered transform nodes. The VRML program is modularized, which is beneficial for programming the script of virtual robot model, and virtual Simulation of motion control is carried out.

This system adopts Shougang MOTOMAN's robot MOTOMAN-HP3, and designs a multilayer distributed network control system on account of Client/Server, sets up the system of VR emulator base on Matlab/Simulink, robot kinematics and VR toolbox. The data is communicated between client and server by control protocol on the base of TCP. The teleoperator robot coordinated control system based on VR is carried out finally.

Keywords: Robot, Teleoperator robot, VR, Control system, Matlab/Simulink

北京工商大学学位论文原创性声明

本人郑重声明：所呈交的学位论文是本人在导师指导下进行的研究工作所取得的研究成果。除了文中已经注明引用的内容外，论文中不包含其他个人或集体已经发表或撰写过的研究成果。对本文的研究做出重要贡献的个人和集体，均已在文中以明确方式标明。本声明的法律后果完全由本人承担。

学位论文作者签名：钱晓磊 日期：2008 年 5 月 12 日

北京工商大学学位论文授权使用声明

本人完全了解北京工商大学有关保留和使用学位论文的规定，即：研究生在校攻读学位期间论文工作的知识产权单位属北京工商大学。学校有权保留并向国家有关部门或机构送交论文的复印件和电子版，允许学位论文被查阅和借阅；学校可以公布学位论文的全部或部分内容，可以采用影印、缩印或其它复制手段保存、汇编学位论文。（保密的学位论文在解密后遵守此规定）

学位论文电子版同意提交后，可于 ☐ 当年 ☐ 一年 ☐ 二年后在学校图书馆网站上发布，供校内师生浏览。

学位论文作者签名：钱晓磊 导师签名：李洪义 日期：2008 年 5 月 12 日

第一章 绪 论

1.1 机器人概述

机器人是一种自动化的机器，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^[1]。

机器人集中了机械工程、材料科学、电子技术、计算机技术、自动控制理论及人工智能等多学科的最新研究成果，代表了机电一体化的最高成就，是当代科学技术发展最活跃的领域之一。

1.1.1 机器人的发展历史

1920 年，捷克作家雷尔·卡佩克发表了科幻剧本《罗萨姆的万能机器人》。卡佩克在剧本中把捷克语“Robota”写成了“Robot”，引起了大家的广泛关注，被当成了机器人一词的起源。

1950 年，美国作家艾萨克·阿西莫夫在他的科幻小说《I, Robot》中首次使用了“Robotics”，即“机器人学”。阿西莫夫提出了“机器人三原则”：(1)机器人不应伤害人类，且在人类受到伤害时不可袖手旁观；(2)机器人应遵守人类的命令，与第一条违背的命令除外；(3)机器人应能保护自己，与第一条相抵触者除外。机器人学术界一直将这三原则作为机器人开发的准则，阿西莫夫因此被称为“机器人学之父”。

1954 年，美国人 George C. Devol 提出了第一个工业机器人方案并在 1956 年获得美国专利。

1960 年，Conder 公司购买专利并制造了样机。

1961 年，Unimation 公司成立，生产和销售了第一台工业机器“Unimate”，即万能自动之意。

1962 年，A.M.F. (机械与铸造)公司，研制出一台数控自动通用机，取名“Versatran”，即多用途搬运之意，并以“Industrial Robot”为商品广告投入市场。

1967 年，Unimation 公司第一台喷涂用机器人出口到日本川崎重工业公司。

1968 年, 第一台智能机器人 Shakey 在斯坦福研究所诞生。

1972 年, IBM 公司开发出直角坐标机器人。

1973 年, Cincinnati Milacron 公司推出 T3 型机器人。

1978 年, 第一台 PUMA 机器人在 Unimation 公司诞生。

1982 年, Westinghouse 公司兼并 Unimation 公司, 随后又卖给了瑞士的 Staubli 公司。

1990 年, Cincinnati Milacron 公司被瑞士 ABB 公司兼并。

日本、西欧各国、前苏联也相继引进或自行研制工业机器人。60~70 年代是机器人技术获得巨大发展的阶段。

80 年代, 机器人在发达国家的工业中大量普及应用, 如焊接、喷漆、搬运、装配。并向各个领域拓展, 如航天、水下、排险、核工业等, 机器人的感知技术得到相应的发展, 产生第二代机器人。

90 年代, 机器人技术在发达国家应用更为广泛, 如军用、医疗、服务、娱乐等领域, 并开始向智能型(第三代)机器人发展。

随着机器人技术的发展形成了新学科——机器人学。建立了相应学术组织, 定期举办学术活动。

我国机器人技术起步较晚, 70 年代末, 一些院校和企业开始研制专用机械手, 80 年代初, 开发小型的教育机器人。1985 年, 哈工大研制出国内第一台弧焊机器人(华字 I 号)。

国家“863”计划把机器人技术作为重点发展技术来支持。建立了“机器人示范工程中心”和机器人国家开放实验室(沈阳自动化所、哈工大、上海交大、南开大学)。我国也建立了机器人学的学术组织, 定期举办学术活动^[2]。

1.1.2 机器人的特点及分类

工业机器人最显著的特点有以下几个:

(1)可编程。工业机器人可随其工作环境变化的需要再编程, 因此它在小批量多品种具有均衡高效率的柔性制造过程中能发挥很好的功用。

(2)拟人化。工业机器人在机械结构上有类似人的行走、腰转、手臂等部分, 在控制上有电脑。此外, 智能化工业机器人的传感器, 如皮肤型接触传感器、力传感器、视觉传感器等, 提高了工业机器人对周围环境的自适应能力,。

(3)通用性。除了专门设计的专用的工业机器人外, 一般工业机器人在执行不同的作业

任务时具有较好的通用性。

(4)机电一体化。工业机器人技术涉及的学科相当广泛，但是归纳起来是机械学和微电子学的结合——机电一体化技术。智能机器人不仅具有获取外部环境信息的各种传感器，而且还具有人工智能，这些都和计算机技术的应用密切相关。因此，机器人技术的发展必将带动其它技术的发展，机器人技术的发展和水平也可以从一个方面验证一个国家科学技术和工业技术的发展和水平。

机器人的种类很多，按智能水平划分：

(1)第一代机器人包括：可变顺序机器人、示教再现型机器人及数控型机器人。

可变顺序机器人，按预定的顺序及条件，依次控制机器人的机械动作，但顺序和条件可作适当改变；示教再现型机器人，通过手动或其它方式，先引导机器人动作，记录下工作程序，机器人则自动重复进行作业；数控型机器人，不必使机器人动作，通过数值、语言等为机器人提供运动程序，能进行可编程伺服控制。

(2)第二代机器人为感知型机器人，利用传感器获取的信息控制机器人的动作，机器人对环境有一定的适应性。

(3)第三代机器人为智能机器人，机器人具有感知和理解外部环境的能力，即使环境发生变化，也能够成功的完成任务。

1.2 遥操作机器人

遥操作机器人(Telerobot)是指能代替人在难以接近或对人有坏的环境中，完成比较复杂操作任务的一种远距离作业系统。遥操作机器人的实现不仅可以满足高技术领域发展的急需，如空间探索和海洋开发以及原子能应用，而且可以广泛地应用于军事领域和民用领域。在这些场合中，遥操作机器人的应用使人摆脱了传统操作者的角色，由直接操作(Operation)变成了遥操作(Teleoperation)，提高了人类在一些危险有害或远距离作业环境的可达到性和安全性。如图 1.1。

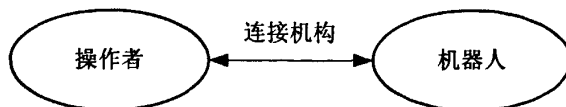


图 1.1 遥操作机器人系统基本结构

1.2.1 遥操作机器人的发展

遥操作机器人是最早的实用型机器人, 1947 年在美国 Argonne 出现的世界上第一台主从遥控机器人是机器人遥操作理念的雏形。在这个系统中, 主、从机械手是通过机械装置连接的, 传输距离有限, 跟踪性能也不是很好。此后, 1954 年出现的电子可编程机器人是一个带伺服反馈的机电遥操作系统, 由操作者对车辆进行远程控制, 操作性能得到很大改善。

二十世纪五十年代到七十年代期间形成了双向力反馈的控制系统, 出现临场感等概念, 遥操作机器人系统基本机构已形成, 这期间是遥操作技术迅速发展的黄金时代, 具有代表性的有 R.Mosher 等在通用电器公司研制的具有两只手臂共 20 个自由度的 Handyman 与 Aaron Kobrinskii 等研制的助残假肢等^[3]。

二十世纪八十年代末, 空间遥操作技术不断完善。加拿大 SPAR 公司建造了 20m 长具有 6 个自由度的遥控操作臂 RMS(Remote Manipulator System), 以及 1993 年德国宇航中心的空间机器人试验 ROTEX(Space Robot Technology Experiment)计划在哥伦比亚号飞机上的空间实验室内顺利完成^[4]。1996 年末, NASA 向太空发射了被称为“Ranger”的带有机器人手臂的飞行器。它在地面的遥控下, 完成各种空间操作任务。它代表了最早的用于空间研究、服务的机器人^[5]。

进入二十世纪九十年代, 网络技术得到了飞速发展, 与主从遥操作技术相结合, 人们提出了基于网络的主从遥操作技术这一概念, 并使它得到了快速发展。国外通过 Internet 技术进行远程操作机器人有很多比较成功的实例。如由澳大利亚 Western 大学研制的六个自由度的工业机器人 Australia's Telerobot、Bradford 大学研制的远程机器人望远镜 Bradford Robotic Telescopic、加拿大艺术家 Garnet Hertz 设计的可画图的 web 机器人 Interface 以及美国 Carnegie Mellon 大学研制开发的通过 web 进行控制的移动机器人 Xavier, 都实现了通过 Internet 远程控制机器人。此外, 通过计算机网络遥控操作的机器人, 对于灾害救助、远程医疗、远程监护以及传统生产模式的改造, 也具有潜在的应用价值^[6]。

近年来, 我国逐渐开展了对临场感遥操作机器人系统的理论与试验研究, 在国家“863”计划中, 智能机器人主题已将临场感技术列为一项关键技术进行研究。清华大学已完成一套用于驾驶 THMR-II 型智能车的临场感系统^[7]。该系统具有立体视觉临场感、听觉临场感、采用无线通讯, 作用距离 5km, 车速 3.6km/h。哈尔滨工业大学机器人研究所针对空间大时延、Internet 传输时延下的空间遥操作及网络遥操作技术, 开展时延传输特性、虚拟环境建模、遥编程、人机交互接口设备、双向力反馈控制、力觉临场感、系统体系结构、网络

协调控制、系统性能评价、多传感器信息融合等相关理论及关键技术的研究，建立了基于 Web 的遥操作机器人系统，得到了 3 项国家 863 计划的支持，取得了阶段性的成果，其中空间仿人灵巧手遥操作系统如图 1.2 所示。此外，中科院沈阳自动化所、航天部 520 所等单位都在积极开展临场感技术的研究^[8]。

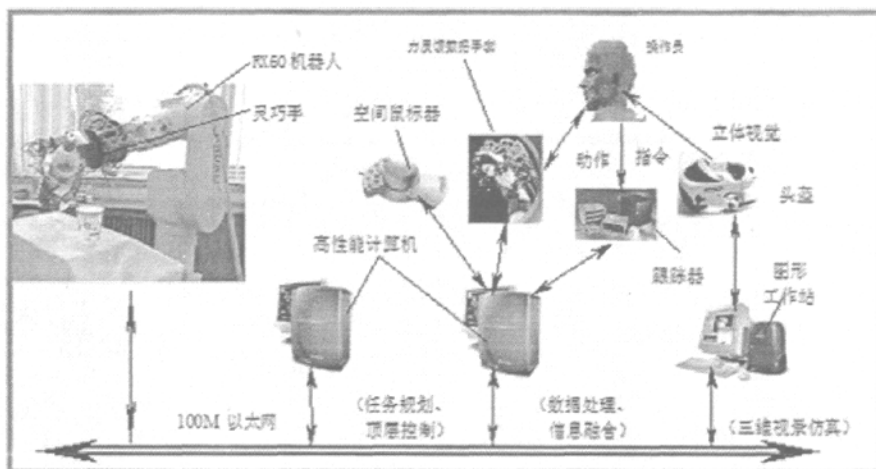


图 1.2 空间仿人灵巧手遥操作系统

1.2.2 遥操作机器人的分类

与机器人发展经历了三代相似，随着各种新技术不断应用到遥操作机器人上，到目前为止，遥操作机器人的发展也经历了四代^[9]：第一代是基于距离开关控制的遥操作机器人；第二代是具有临场感的遥操作机器人；第三代是监控式遥操作机器人；第四代是自主式遥操作机器人。另外，按照主从手之间的连接方式，可以将遥操作机器人分为：通过机械结构连接的遥操作机器人、通过伺服系统连接的遥操作机器人和通过计算机网络连接的遥操作机器人。若按照主、从手是否具有相同的结构，还可以将遥操作机器人分为：同构的遥操作机器人和异构的遥操作机器人。监控式和自主式遥控机器人是机器人智能化发展的必然结果，而通过计算机网络连接的遥操作机器人是计算机网络与机器人技术结合的产物。这些遥操作机器人广泛应用于空间探测、海洋探测、危险场所作业等领域，具有极大的研究价值和社会、经济价值，因此，本文将主要以计算机网络连接的遥操作机器人为研究对象。

1.2.3 遥操作机器人的主要研究方向及重点问题

遥操作机器人不是一个单独的机器人，而是一个系统，由操作机、远程机器人、通信系统组成，因此，它所涉及的研究领域几乎涵盖了机器人的所有相关领域。但是，遥操作机器人还是有自己的研究方向和重点问题，主要有^[9,10,11]：

- 1、遥操作控制平台研究：遥操作机器人系统的最佳结构、功能分配等；遥操作方式，直接控制、监控控制、预测控制、遥编程控制等；
- 2、操作机研究：专用操作机、通用操作机等；
- 3、通信、网络和传输延迟：延迟的产生原因、规律、对系统的影响、解决方法等；
- 4、临场感、虚拟现实和人机接口技术：Tele-presence、Tele-existence、操作机仿真、机器人仿真等；
- 5、控制与伺服系统研究：延迟环境下控制器设计、基于PC的网络控制器、通用控制器等；
- 6、遥操作机器人多机协同；
- 7、其他与特殊作业环境相关的研究：如，空间操作时的震动和变形控制等^[12]。

1.3 课题研究的主要内容

1.3.1 研究的意义

机器人技术一直沿着智能化的方向发展，智能机器人通常被认为是全自主系统，多年来人们按照这种思路去开发实际的智能系统。二十世纪八十年代以后，随着对智能机器人的进一步研究，研究人员认为工作在负载或非结构化环境中的全自主式机器人系统在短期内难以实现，因此对于许多机器人系统应用来说，都需要有人的操作介入。这让人们认识到智能系统应该是一个人—机集成系统，这种认识的基本思想就是要充分利用人和机器人各自的优点，互补地完成更加复杂的工作。这种思想成功的关键不是开发出一种全自动系统而是允许操作者与机器人系统相互合作，以使总体的效率和生产力得到提高。由此可见，在智能机器人技术成熟之前，遥操作机器人将是最合适的。

由于机器人技术在工业、国防、空间探测和海洋开发中的重要地位，各国对机器人的研究均十分重视，纷纷将对机器人的研究作为重点纳入本国的科研计划之中。近年来，美国、欧盟、日本等发达国家的政府先后推出并实施了多个与机器人技术、自动化工艺装备

相关的国家研究计划，以支持相关的核心技术和关键重大装备的研发与应用。我国政府也非常重视机器人技术与开发，863 计划和 973 计划中包含了大量的机器人相关的项目。经过多年的努力，我国在机器人技术与自动化工艺装备等方面已取得了突破性进展，缩短了同发达国家之间的差距。但是在机器人与自动化装备中核心及关键技术的原创性研究、高性能关键工艺装备的自主设计和制造能力、重大成套装备的系统集成与开发能力、高可靠性基础功能部件的批量生产应用等方面，同发达国家相比我国仍存在较大的差距^[10]。

随着神舟 5 号、6 号飞船相继发射成功，探月计划(嫦娥工程)的成功实施，表明我国在空间技术上有了实质性突破。今后一段时间内，我国仍将加大空间技术方面的研究力度，建立空间站、登月计划。在这些宏伟的工程中，遥操作机器人都将扮演重要角色^[13]。遥操作机器人方面的研究，不仅意义重大，而且是一项十分紧迫的任务。

1.3.2 研究的主要内容

如前所述，本文将以计算机连接的遥操作机器人为研究对象，同时，并将从计算机科学的角度来进行研究。研究的主要内容如下：

- 1、基于 SolidWorks-VRML 建立机器人虚拟模型，采用结构化、模块化、层次化，提高软件的设计质量；
- 2、探讨虚拟现实技术在机器人遥操作中的作用和应用方式，并将虚拟现实技术应用到遥操作机器人系统中来解决遥操作中的延迟、人机接口问题；
- 3、对遥操作机器人系统结构、控制网络和分布式计算机平台进行研究，在此基础上建立一个基于计算机网络的遥操作机器人系统；
- 4、其他相关研究工作。包括对机器人通讯控制、机器人控制建模和机器人运动学、路径规划、轨迹生成等的研究和实现。

1.3.3 论文的结构

本论文由前后两大部分组成。前半部分包括第一、二、三章，主要是关于遥操作机器人和虚拟现实技术相关问题的研究及结论。后半部分包括第四、五、六、七章，主要是遥操作机器人系统的实现和改进思想。

第一章：绪论。关于机器人和遥操作机器人的相关概念，以及对本论文的相关说明。

第二章：虚拟现实技术研究。关于虚拟现实中的关键技术及在遥操作机器人中的应用研究。

第三章：基于网络的遥操作机器人系统研究。关于遥操作机器人控制系统结构和基本控制方式的研究。为实现基于网络的遥操作机器人系统奠定基础。

第四章：基于 Solidworks-VRML 构建虚拟环境中机器人模型。关于虚拟机器人模型建立的方法研究。

第五章：基于虚拟现实技术的遥操作机器人系统的设计方案。给出了一个具体的遥操作机器人系统的实现方案。

第六章：基于虚拟现实技术的遥操作机器人协调控制系统的实现。给出了遥操作机器人系统平台框架，并演示相关的实验。整个系统使用 Matlab/Simulink 和 Visual C++6.0，在 Windows2000/XP 下实现。

第七章：总结和展望。关于研究工作的总结和系统改进的设想。

第二章 虚拟现实技术研究

2.1 虚拟现实技术概述

虚拟现实(Virtual Reality)一词最早是 19 世纪 70 年代在麻省理工学院(MIT, Massachusetts Institute of Technology)出现,表示人类存在于计算机空间中的概念。

虚拟现实技术,简称 VR 技术,也成灵境技术或人工环境。它综合了计算机技术、多媒体技术、传感器与测量技术、人工智能技术、网络技术以及仿真技术等多个信息技术,是近年来十分活跃的技术研究领域。

2.1.1 虚拟现实的发展

1962 年 Morton Heilig 研制成功首台全传感仿真系统 Sensorma。

1965 年 Sutherland 博士提出 Ultimate Display(终极显示)概念。

1968 年 Sutherland 首次研制陈宫头盔式显示器(Head Mounted Display, HMD)。

20 世纪 70 年代中期美军战斗机飞行模拟器研制成功,仅头盔显示器就价值数百万美元。

20 世纪 80 年代中期,美国 NASA 研制成功应用于载人航天使用的(Virtual Interactive Environment Workstation, VIEW)系统,VIEW 系统具有一下特征:头部安装有跟踪器、单色宽视场立体头盔显示器、语音识别器、三维声音输出装置以及带有跟踪器的数据手套等。VIEW 系统配置已成为当今虚拟现实系统的典型配置。

20 世纪 90 年代,VR 的研究热潮开始向民用高科技企业转移。研究出第一套数据手套的 Jaron Lanier 称为销售 VR 产品的第一家商业公司 VPL 的总裁,该公司销售的第一套数据手套被命名为 DataGloves,第一套 HMD 被命名为 EyePhones。

20 世纪 90 年代中期,标准化的虚拟现实建模语言 VRML 为在 Internet 上构建可共享、可交换的 WWW(World Wide Web,万维网)虚拟环境奠定了良好基础。随着科技的进步,VRML 本身也由早期的 VRML1.0 发展到 VRML2.0,再发展为 VRML97 国际标准,直至目前的 X3D 规范。

目前,虚拟现实系统已由单机系统发展到分布式虚拟现实系统,进一步还要发展到支持协同工作的分布式虚拟现实系统,即由过去只支持多用户共享,发展到需要支持以多人相互感知、协同操作等为目的的协同感知领域。若要让异地的多个用户协同工作,共同生存于一个虚拟协同空间,则所有用户当前所获得的信息必须保持一致;同时一个用户应能感知其他用户的工作状态,并与之进行交流^[14]。

2.1.2 虚拟现实技术的特征

虚拟现实是发展到一定水平上的计算机技术与思维科学相结合的产物,是由计算机产生一种人为虚拟的环境,使人产生一种沉浸于这个环境的感觉,可以直接观察、操作、触摸、检测周围环境及事物的内在变化,并能与之发生“交互”,给人一种“身临其境”的感觉。它的基本特征有:沉浸性,交互性和想象性(Immersion-Interaction-Imagination),人们形象的将其称为 I3,它们共同地支撑起虚拟现实技术,如图 2.1 所示。另外,多感知性、自主性也是虚拟现实的重要特征。

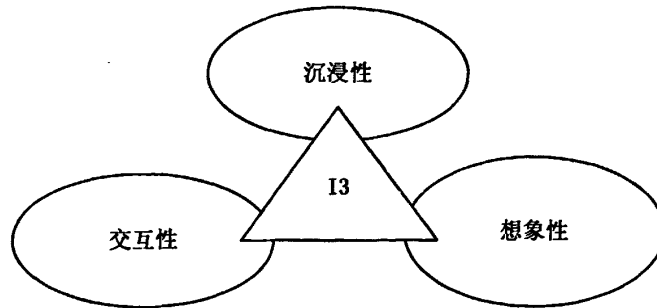


图 2.1 虚拟现实的三大特征

2.1.3 虚拟现实系统的组成及分类

虚拟现实系统由硬件、软件以及各种传感设备所构成的。软件主要由检测模块(检测用户的操作命令)、反馈模块、传感器模块(服务于检测模块和反馈模块)、控制模块和建模模块组成。传感设备有立体显示头盔、数据手套、数据衣服、全方位监视器、风镜型显示屏和全景大屏幕显示屏等位置跟踪设备和交互设备,由它们产生信号,与计算机实现交互作用。

虚拟现实系统按其功能不同,可分成三种类型:

简易型虚拟现实系统,也称为桌面虚拟现实系统或窗口中的 VR。这是最简单的一种虚拟现实系统,由一台普通的计算机系统组成,使用者通过键盘、鼠标便可与虚拟环境进行交互。这种系统不能够实现真正意义上的“现实”,但它结构简单、价格低廉,易于普及推广,是一套经济实用的系统目前应用十分广泛。

沉浸型虚拟现实系统。沉浸型虚拟系统是一套比较复杂的系统。使用者必须头戴头盔、手带数据手套等传感跟踪装置,才能与虚拟世界进行交互。由于这种系统可以将使用者的视觉、听觉与外界隔离,因此,用户可排除外界干扰,全身心地投入到虚拟现实中去。这种系统和优点是用户可完全沉浸到虚拟世界中去,缺点是系统设备价格昂贵,难以普及推广。

共享型虚拟现实系统。共享型虚拟现实系统是利用远程网络,将异地的不同用户联结起来,共享一个虚拟空间,多个用户通过网络对同一虚拟世界进行观察和操作,达到协同工作的目的^[15]。

2.2 虚拟现实中的关键技术

VR 系统的建立涉及到一系列的关键技术^[17]:

一、人机接口与多通道人机交互技术

虚拟现实技术的目的是提供一种自然的人机交互环境。这种人机交互环境与以往的三维交互环境不同,它提供给使用者的不再是简单的三维物体,而要求使用者真正参与到系统中,具有沉浸感。从这一点来看,虚拟现实系统不仅仅要利用计算机这个工具,更多的需要将人的行为,感觉和思维方式融合于系统中,突出人在虚拟环境中的主导地位。在虚拟现实系统中不仅可以采用传统的输入输出设备作为人机交互设备,如鼠标、键盘、显示器等,而且还应利用一些新型的人机交互设备和技术,如头盔显示器、数据手套、跟踪器、力反馈设备、语言识别技术、手势识别技术、姿势识别技术等。

二、建模技术与虚拟环境表达

虚拟环境的建模是整个 VR 系统建立的基础,主要包括三维视觉建模和三维听觉建模。这种视觉建模与传统 CAD 建模和动画建模有着本质的区别,虚拟环境中的建模不仅要求在图形上达到高度逼真的三维视觉效果,而且要考虑系统中各部件模型的功能,实时响应能力及完成任务的满意度和可靠度等。三维视觉虚拟环境由不同的对象(模型)组成。从视觉的角度来看,三维虚拟环境可以看作是视觉的高层处理的场景表达,这种场景表达可归

结为三维重建问题。目前获得这样的三维数据模型的方法主要有两种^[16]：一种是基于图像的虚拟环境产生技术，该技术避开了繁琐的建模过程和复杂的绘制计算，它利用全景图集合来构成虚拟环境，在虚拟环境中漫游相当于选择不同的全景图，全景图可用计算机生成，也可用全景相机拍摄，或用普通相机拍摄后再加以拼接。

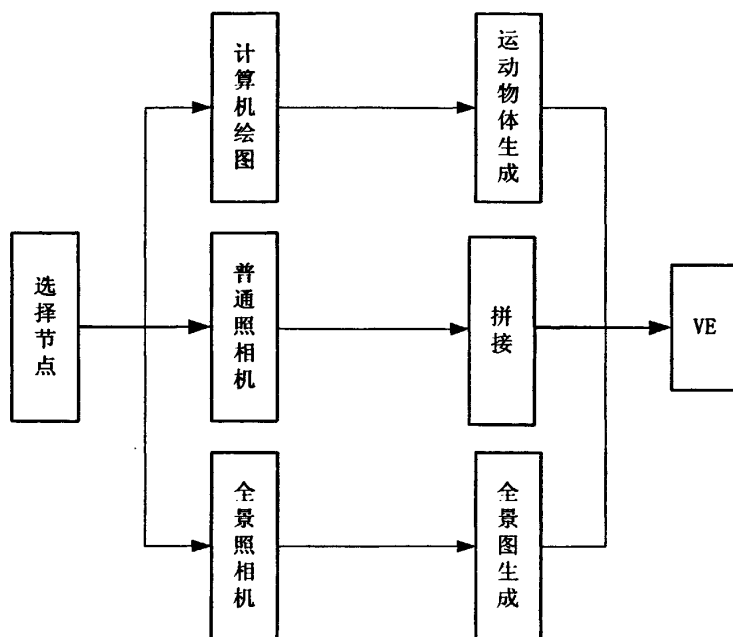


图 2.2 基于图像的虚拟环境模型

此方法具有无需繁琐的场景建模工作，能实时地显示生成的环境的优越性，但其存在的限制在于，(1)用户只能在事先创建的视点间漫游；(2)场景必须是静态的。即它的自由度较小，交互性较差，且面临多视点，有限图象数据的融合问题。

另一种方法是基于几何模型的虚拟环境产生技术，该技术运用计算机图形学的方法来构造虚拟环境，其虚拟环境是由各类三维几何体合成的，在虚拟环境中漫游是根据观察点，及其观察方向通过实时计算，实时绘制三维几何实现的。

该方法具有可任意选择观察点和观察方向的优点，具有很大的自由度和很强的交互性，对于非常繁琐的几何体的建模，必须应用限时图形绘制技术、预处理技术、实例技术及纹理技术等处理技巧来解决“真实感”与“实时性”的矛盾。

三、系统设计方法和体系生成

虚拟现实系统的复杂性，要求在系统分析和设计中，以科学的方法论作为指导，从系统的角度标出该系统的人，作业体及结构本身、作业对象、现场环境等的约束和协调关系。

另外，还应充分考虑到系统中人的因素以及系统的鲁棒性、稳定性、可靠性等因素。

2.3 虚拟现实技术在遥操作机器人中的应用

2.3.1 虚拟现实技术的应用

遥操作机器人系统中，虚拟现实技术的应用方式既取决于遥操作机器人系统的操作方式，也取决于实现虚拟现实的程度。虚拟现实技术在遥操作机器人系统中的重要性可以分为 3 级，替代的、重要的、必需的。虚拟现实技术在遥操作系统中应用方式也可分为三类 [11]：

A、三维场景的表达和自动建模：

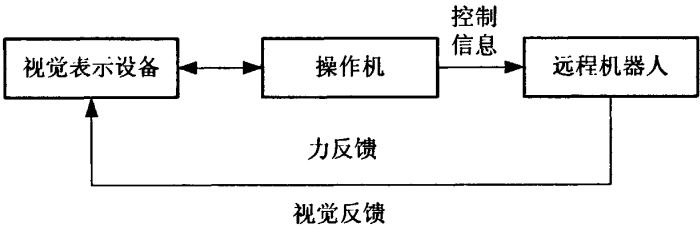


图 2.3 双向遥操作中的视觉表示

典型的应用环境是遥操作机器人以双向遥操作方式工作，如图 2.3 所示。这时操作人员需要通过视觉信息来获得机器人的运行状态和环境信息，虚拟机器人及其虚拟的工作环境代替了传统的由摄像机获得视觉信息，再由电视设备显示信息的方式。这种应用方式下，远程机器人的虚拟模型容易建立，对结构化环境的建模也比较容易，但是对非结构化环境建模则难度很大，困难的根本原因在于从摄像机获得的图像中识别物体并自动建模的可靠性和实时性还不能达到指导操作人员进行操作的要求。对于遥操作机器人来说，虚拟现实的这种应用是替代性的。对于低通信带宽的遥操作机器人系统，采用虚拟现实技术作为三维场景的表达可以节省大量的通信带宽，所以，对这样的遥操作系统来说虚拟现实的这种应用则是重要的。

B、作为遥操作界面：遥操作机器人系统中，虚拟现实技术作为遥操作的界面也就是作为较完整的人机接口。如图 2.4 所示。与虚拟现实作为三维场景的表达和自动建模不同的是，操作人员与遥操作机器人之间，已经被虚拟现实技术所创建的虚拟操作环境完全隔离。

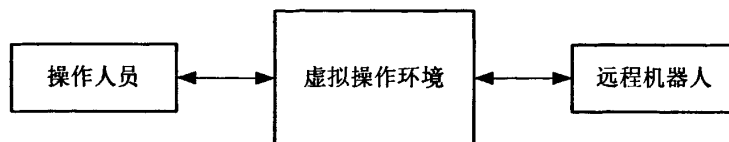


图 2.4 虚拟现实作为人机接口

典型的应用环境是遥操作机器人以遥编程方式进行操作或者以监控方式运行。在这种操作方式下，操作人员所操作的不是远程机器人，而是由虚拟现实技术建立的虚拟环境中的虚拟机器人模型，由虚拟环境来将操作人员的具体操作转化为适当的符号指令发送给远程机器人执行，虚拟环境仿真远程机器人及其环境，直接产生输出并表现给操作人员。从远程机器人传回的反馈信息很少，一般是出错信息，由虚拟现实环境进行处理。这种应用中虚拟现实技术是必需的，也是解决通信延迟问题的辅助方法。

C、建立具有真实感的多传感器融合系统仿真平台：对于遥操作机器人系统，虚拟现实技术的这种应用方式仍然是作为人机接口，不同的是这种遥操作机器人系统可以提供多种遥操作方式，操作人员可以不同层次，不同方式介入对远程机器人的操作中。远程机器人侧的各种传感器信息将大量传回操作端，虚拟环境将融合这些信息并进行处理，然后真实再现给操作人员。这种具有真实感的多传感器融合系统仿真平台系统是建立在虚拟现实技术基础上的，虚拟现实技术是必需的。

2.3.2 虚拟现实技术的实现方法

遥操作机器人系统中，最主要的是视觉虚拟现实技术，这里仅给出视觉虚拟现实技术的实现方法，如图 2.5 所示：

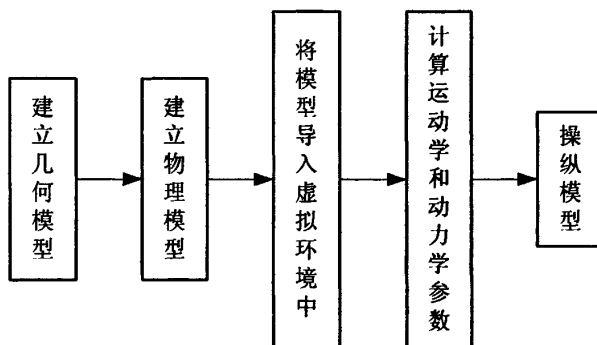


图 2.5 视觉虚拟现实的实现方法

其中建立几何模型的主要目的在于实现机器人外形和运动状态的显示，以及对操作机输入进行变换处理，对于不采用操作机而直接使用鼠标、键盘等设备进行输入的遥操作系统，

提供一种可视化的图形用户接口。建立物理模型的主要目的在于进行机器人的运动学和动力学计算,使机器人的运动符合物理规律,从而给操作者以真实的感觉。

2.3.3 机器人模型的建立

实现 2-3-1 节中虚拟现实技术的应用方式中的第一种应用,即将虚拟机器人模型作为输入手段和输出的图形表示手段,就需要对机器人建模,得到远程机器人的几何模型。结合上述基于模型的三维视觉再现技术,虚拟机器人几何建模的基本方法有三种:

A、利用 OpenGL, DirectX 等图像库提供的基本建模功能进行机器人的几何建模。由于 OpenGL, DirectX 提供对低级图元的显示和编辑支持,对建模和模型的编辑提供的支持很少,因此,这种方法对于简单型体的建模是可行的,而对于复杂型体的几何建模则难以实现。图 2.6 给出了利用 OpenGL 图元建模的简单示例:

```
//多边形图案
glColor3f(0.9,0.86,0.4);
glPolygonStipple (pattern);
glBegin(GL_TRIANGLES);
glVertex2i(310,310);
glVertex2i(220,80);
glVertex2i(405,80);
```

图 2.6 OpenGL 几何建模示例代码

B、使用 VRML 语言。VRML 语言由 SGI 公司开发,主要用来与浏览器集成,从而在 WWW 上提供虚拟现实功能。VRML2.0 已经为 ISO 接受为 VRML97 标准,提供立体视觉、听觉效果,支持动态显示和交互。另外,提供与 Java 的编程接口,以方便与虚拟环境中物体的交互,以及实现虚拟物体的人工智能。

C、用 Maya、3D Max 等 3D 开发软件,或者 Auto CAD 进行机器人的几何建模,然后将几何模型加载到虚拟环境中。这可以通过直接读取 3D 模型文件或者利用 View3DS 等工具将 3D 模型文件转换成可在 OpenGL 中使用的模型的显示列表。借助这些 3D 开发软件或者 Auto CAD 可以对复杂型体进行精细建模,并且可以对材质、纹理、光照等进行编辑,达到逼真的视觉效果。然后将这些模型加载到虚拟环境中,利用 OpenGL, DirectX 图形库中的函数对创建的机器人模型进行操纵,达到输入交互和输出显示的目的。

建立模型的各种方法,各有利弊。当实际显示运动的模型和操纵模型动作时,方法 A、B 由于仅需根据各个关节角度对几何模型进行重画,实现起来相对简单,而方法 C 由于要

进行多次坐标系的重新计算,则实现的难度要大的多。本文采用 VRML 语言对机器人模型进行建模和编辑^[12]。

2.4 小结

现代的遥操作机器人系统中,普遍采用了计算机技术和虚拟现实技术。虚拟现实技术的本质上是一种人机接口技术,对机器人遥操作是不可或缺的,也是解决通信延迟的辅助手段之一。虚拟现实技术在遥操作机器人系统中有三种用途。当虚拟现实作为简单的人机交互手段时,需要建立机器人的几何模型,也即显示模型,用于实际控制时,则还需要建立机器人的物理模型,也即控制模型。

第三章 基于网络的遥操作机器人系统研究

3.1 机器人网络控制系统

基于 Internet 的远程控制机器人系统是一个覆盖面很广的复杂系统。它不仅拓展了传统的机器人遥操作的应用,而且随着互联网的迅猛发展,它在许多新兴领域中,如自主机器人系统、远程制造业、远程培训、分布式制造系统以及家用机器人和娱乐机器人等,也有良好的应用前景。以下按照网络技术在机器人学中不同的应用方向及不同的技术侧重点,对国外研究机构基于 Internet 的遥操作技术、基于 Internet 的自主移动机器人控制技术、基于中间件技术的分布式机器人系统和分布式机器人软件库几个方面进行控制系统的分析和研究。

3.1.1 基于网络的遥操作技术

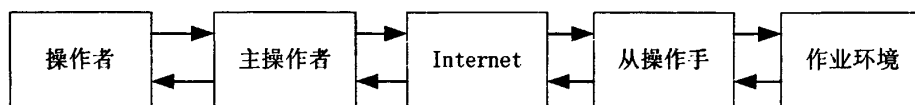
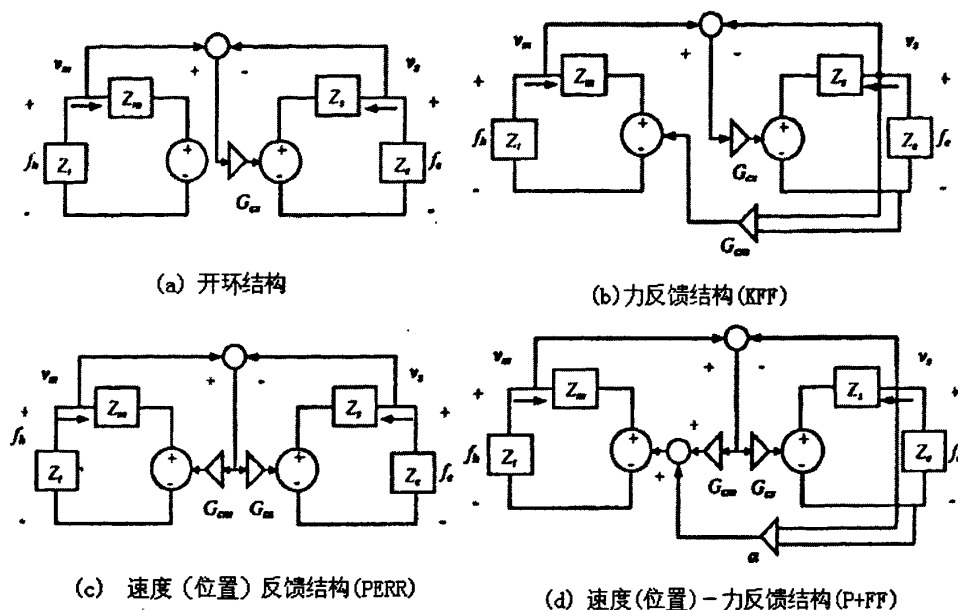


图 3.1 基于 Internet 的远程控制遥操作系统结构

基于网络的远程控制机器人技术继承和发展了遥操作技术,是给用户通过 Internet 对远程设备实施遥操作控制。传统的遥操作系统允许操作者通过主从机器人来实现对远程设备的控制。基于网络的远程控制遥操作系统的一般结构如图 3.1 所示。按系统结构的不同,还可以将其分为两大类:基于 Internet 的双向力反馈遥操作机器人系统和基于 Web 浏览器的遥操作机器人系统。前者只是在通讯手段上与传统遥操作不同,从而在结构上和技术上继承了传统遥操作系统采用的方法和技术。后者完全以网络为主体,系统中只存在从操作手,而主操作手被 Web 浏览器代替。

现有的遥操作系统多采用以下四种控制结构,如图 3.2 所示:


 图 3.2 遥操作系统的控制结构^[18]

Z_t 、 Z_e 、 Z_m 、 Z_s 分别为操作者阻抗、环境阻抗、主机械手阻抗和从机械手阻抗。这四种结构都在从端构成速度的本地闭环控制, G_{cs} 为控制器。开环结构不含有任何信息的反馈, 速度(位置)反馈结构返回两机械手速度或位置的差值, 这两种结构都不具有力觉临场感, 多用在自由运动任务中, 依靠视频信息来获得临场感。在受限运动即从机械手和环境有相互作用的任務中, 多采用力反馈或速度(位置)—力反馈结构。力反馈结构仅返回从机械手与环境的相互作用力, G_{cm} 为控制器, 速度(位置)—力反馈结构反馈的是从端作用力和两机械手速度(位置)差的线性组合, 它是速度(位置)反馈结构和力反馈结构的合体。后三种控制方式都属于双边控制, Sherman 对这三种控制方式的性能进行了分析^[19], 并指出速度(位置)—力反馈结构的遥操作系统具有最好的控制性能。但是这种方式比较复杂, 对速度或位置的差比较敏感。

遥操作系统的临场感可以分为视觉临场感和力觉临场感, 遥操作系统的透明性是针对这两种临场感的一个遥控机器人学专有名词。不言而喻, 透明就是操作者借助主从机械手和信息传输通道与远端环境相互作用, 感觉就像在直接对环境进行操作一样。首先操作者看到的要是能反映真实从端环境的图像信息, 其次操作者手上感觉到的力是真实的环境作用力。因此透明是遥操作系统实现操作者临场感的根本。力觉临场感即“力觉”上的透明性, 以下的分析中简称为透明性。

一般认为, 操作者感觉到的阻抗定义为:

$$Z_t = \frac{f_h(s)}{v_m(s)} \quad (3-1)$$

$$\text{环境阻抗: } Z_e = G_e = \frac{f_e(s)}{v_s(s)} \quad (3-2)$$

对于力觉临场感系统，如果操作者感觉到的阻抗与从机械手和环境作用的阻抗相等，即 $Z_t = Z_e$ ，操作者感觉像在直接对环境进行操作，则说这个系统是透明的。系统要达到透明，除了阻抗匹配以外，还要实现主从手之间行动和力的跟踪，即： $v_s(s) = v_m(s)$ ， $f_e(s) = f_h(s)$ 。显然，由于时延的存在，速度与力的信息滞后，系统完全透明是无法实现的。透明性比静态跟踪性能更强的操作性能，它更强调了系统的动态跟踪，而且要求跟踪比率相等。临场感遥操作系统各性能之间的关系如图 3.3 所示：

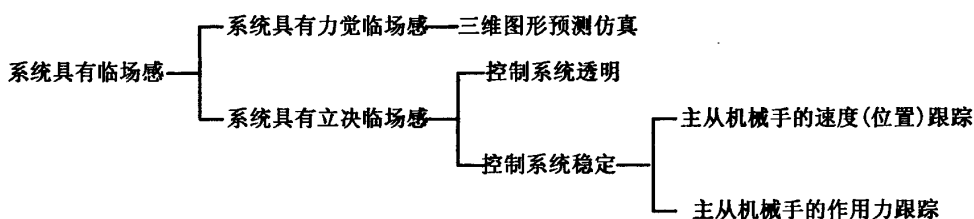


图 3.3 临场感遥操作系统各性能之间的关系^[20]

由于遥操作机器人上的双工控制器负责主从设备间的双向信息流传输，所以双工控制器是否被设计成稳定且透明的系统是非常重要的。但是，由于通讯中的延时和滞后和系统中自身存在的不稳定因素，要实现稳定且透明的双工系统操作仍有很大的困难。由于网络系统存在显著的通讯延迟，对于基于 Internet 的遥操作机器人系统而言，如何设计具有自适应性的双工控制器，提高它对网络时滞的鲁棒性并对环境阻抗能进行在线辨识，是远程控制机器人在遥操作方面的研究重点。基于 Internet 的机器人遥操作技术的提出开创了一个崭新的研究领域，使遥操作技术的应用走向网络化和全球化。

3.1.2 基于 Internet 的远程控制自主式移动机器人技术

国外在基于 Internet 的远程控制自主式移动机器人系统方面已经开展了深入研究，最初基于网络的移动机器人的自主性能有限，而且同一时间内只能向单一用户提供网络服务。瑞士联邦洛桑科技研究所的 Keep On The web 就是其中典型代表。用户可通过网络控

制小型移动机器人在人造迷宫中进行运动并能同时经摄像头的图像反馈进行观察。

移动机器人根据控制目标的不同，可以分为三类典型的控制问题，分别是：路径跟踪问题、轨迹跟踪问题和点镇定问题。

(1) 路径跟踪问题

所谓路径跟踪问题，就是在惯性坐标系里，机器人必须从一个给定的初始状态出发，到达和跟踪一条理想的几何路径，而机器人的初始点可以在这条路径上，也可以不在路径上。

在移动机器人的路径跟踪问题上，一般地，我们会选取两个输入量中的前进量为任意的常量或时变量，而另一个输入量用作控制量。所以路径跟踪问题可以转换成关于路径跟踪误差的一个标量函数的零点稳定问题。控制的目标是使到机器人和给定路径之间的距离为零。

(2) 轨迹跟踪问题

所谓轨迹跟踪问题，就是在惯性坐标系里，机器人必须从一个给定的初始状态出发，到达和跟随一条理想轨迹，同样的，机器人的初始点可以在这条轨迹上，也可以不在轨迹上。而与路径跟随问题不同的是，这条理想轨迹是一条和时间成一定关系的几何路径。

在移动机器人的轨迹跟踪问题里，机器人必须跟踪满足特定时间规律的笛卡尔路径，也即机器人必须跟踪一个移动的参考机器人。通常的，我们将移动机器人的轨迹跟踪问题转化为这样的描述：要求机器人参考点的位置跟踪一个随时间的理想变化曲线。

其数学描述如下，给定一参考机器人^[21]

$$\begin{cases} \dot{x}_r = v_r \cos \theta_r \\ \dot{y}_r = v_r \sin \theta_r \\ \dot{\theta}_r = \omega_r \end{cases} \quad (3-3)$$

其中，参考机器人的状态向量 $q_r = [x_r \quad y_r \quad \theta_r]^T$ 参考速度向量 $V_r = [v_r, \omega_r]^T$ 。

(3) 点镇定问题

所谓点镇定问题，指的是机器人从一个给定的初始状态到达一个理想的目标状态，并稳定在给定目标点，该问题也即是在机器人状态空间的(平衡)点的稳定问题。

其数学描述为：给定一任意的参考机器人的状态 q_r ，先设计一个平滑的速度控制 $v_c = f_c(e_p, v_r, K, t)$ ，使 $\lim_{t \rightarrow \infty} (q_r - q) = 0$ 。然后再计算加到机器人上的力矩 τ ，使到 $t \rightarrow \infty$ 时， $v \rightarrow v_c$ 。

我们注意到, 在移动机器人的这三类控制问题中, 路径跟踪问题由于只需要通过改变机器人的角速度(线速度可以设定为一常值)来减小机器人和参考路径之间的距离, 所以它是相对简单的。而点镇定问题相对复杂一些, 它的速度控制 v 是时变的。相对于路径跟踪和点镇定问题, 轨迹跟踪问题更具代表性。

3.1.3 分布式机器人系统

Internet 技术使人能够与网络上任何地点的设备相连接、交互。随着接入控制系统的机器人数量不断增多, 以及需要同时访问系统的用户数量的增加, 原有的远程控制机器人系统服务器已不能满足网络负荷。因此, 需要将服务器上的控制功能模块分解到其他计算处理设备上。比如, 将机器人本地控制器或图像处理控制器等管理由其他的计算机完成。这些设备在物理空间上可以处于不同的地点。这样就形成了基于 Internet 的分布式机器人控制系统。分布式机器人系统最大的特点就是低价、灵活、可扩展。对于一对多结构、多对一结构还有多对多结构的远程控制机器人系统, 分布式结构是其理想的选择。美国航空航天管理局的寻路者计划(Pathfinder mission)就是通过 Internet 实施分布式控制的成功实例。

对于分布式机器人控制系统来说, 需要解决网络上分布式对象之间的数据交换问题, 即构建高性能的通讯协议体系以实现多个用户和代理间协作控制。目前不同的分布式机器人系统使用了不同的通讯协议和技术, 这些机制包括^[22,23,24]:

(1)RMI(Remote Method Invocation, 远程方法调用)是 Java 虚拟机之间对象互相调用对方函数、启用对方进程的一种机制。RMI 采用序列化机制通过网络发送对象, 实现了 Serializable 接口的类的信息可以被复制, 而文件格式、TCP/IP, Socket 和 I/O 等细节问题被隐藏。

(2)CORBA(Common Object Request Broker Architecture)规范是由 OMG(国际对象管理组织)发布并制定的标准。它定义了一种互相交换数据和发现服务的机制。其基本思路是以 ORB(对象请求代理)作为中间件来建立两个对象间的客户/服务器关系。

(3)MOM(Message-Oriented Middleware, 面向报文的中间件)与 CORBA 和 RMI 不同, 它不是工业标准, 而是在分布式应用环境中支持特定种类通用目标报文交换的中间件的集成术语。MOM 数据的交换是通过报文传输或报文队列, 并支持分布式计算进程间的同步或异步交互。MOM 系统使用可靠队列来确保报文传送, 并对所支持的报文提供目录检索、

安全及管理等服务。队列特别适用于那些逐步递进的进程。MOM 模型中的报文是基于事件驱动的系统，而不仅仅是简单程序调用。客户可应用具有优先级机制的队列，这使得高优先级的报文可超越不重要的报文，这对分布式机器人的多任务控制十分重要。

通过比较可发现上述三种机制的通讯机理和应用范围不尽相同。RMI 和 CORBA 把客户和服务端间的通信建立在更高的对象机制上。对象间互相使用对方所提供的方法，而不必知道它们是远程方法，数据编发的协议被隐藏。RMI 的最大优点是实现了整个对象在客户和服务端之间的传输，因此对于本系统采用纯 Java 编写的客户和服务端间的分布式对象间的通信，无疑是最佳方案。CORBA 与 RMI 的最大不同之处在于 CORBA 实现了不同语言写成的对象之间的调用。尽管 CORBA 避免了 RMI+JNI 的间接实现，但 CORBA 机制的实现，依赖于第三方提供的软件，如 Inprise 公司的 VisiBroke，IONA 公司的 Orbix 等。相比之上，CORBA 适合于更加庞大、复杂的分布式机器人系统。由于 MOM 支持的是报文格式，因此它主要适用于有延迟的异步通讯；RMI 和 CORBA 均基于 RPC(远程过程调用)语义，是设计用以支持同步通讯的。此外基于报文的通讯允许向多个同位体广播信息。虽然 RMI 受到所有基于其上的应用必须是用 Java 编写的限制，但 RMI 也和 CORBA 一样是通用标准，所以具有巨大的互联潜力。与之比较 MOM 可提供嵌入于应用程序中的轻型客户端应用编程接口，而且由于 MOM 不存在同步通讯也不需要预装任何先决软件，所以它在简单环境中更具有应用潜力。

3.1.4 分布式机器人系统软件库

网络技术的发展已经能使各地的、各种各样的机器人终端连接到互联网上，并通过网络对机器人进行实时有效控制和及时的技术更新。这项技术包括网络接口装置、众多信息组的压缩与解压方法及先进的通讯传输技术。科研人员对能使大家共享各种最新机器人技术的机器人软件库(分布式机器人软件库 Distributed robotic software library)的需求非常迫切。当 CORBA 技术被采用时，在 Internet 上构建分布式机器人软件库就变得可行。研发分布式机器人软件库主要基于以下原因：

(1)机器人系统本身是多种应用技术的综合，其基本技术就包括有机器人视觉、力的感知、机器人运动规划及各种相关控制等；

(2)当机器人系统较复杂时，对各技术细节无需做出具体要求，只要能达到目标性能即可；

(3)随着机器人的复杂度不断提高,单个科研机构对各种技术都进行独立研发比较困难;

(4)机器人控制软件的编写、安装和测试的工作量巨大;

(5)科研人员和研发机构需要分布式机器人软件库来不断进行技术更新;

要实现基于网络的分布式机器人软件库首先要制定安全且高性能的通讯协议。通过对系统的资源管理可提高系统的容错性,并能在服务器间执行任务的调度和实现负载的均衡。由于 Internet 的不可预测时滞及有限的通讯带宽,导致基于其上的分布式软件库传输的数据类型不能太大、计算精度不能太高及进行实时处理的能力有限^[25]。

3.2 遥操作系统中的基本控制方式

最初的操作机器人,操作人员通过操作机直接操作远程机器人,操作的方式只有所谓的直接操作。但是,随着遥操作机器人的发展,当遥操作机器人用于空间探测、海洋探测等领域时,或者通过计算机网络来进行遥操作时,通信延迟和通信的不确定性引入系统,为了适应和解决延迟问题,遥操作的方式也发生了改变。

目前常用的机器人远程控制方式主要有以下 3 种:

3.2.1 直接控制

直接控制(Direct Control)。这种控制方式也被称为木偶式(Puppet Approach),即由操作人员完全控制遥机器人,对机器人某些设备、装置直接发送动作指令函数以完成某项任务。其优点在于,充分利用人的感知、判断和决策能力;增强系统的适应能力,具有较强的恢复能力;其缺点是在直接控制方式下,需要向操作人员提供关于操作环境的包括视觉、运动觉在内的大量即时信息和所需带宽高,而在大时延条件下,这种基于传感器信息的遥操作是很难实现的。另外,为了保证操作的稳定性,操作员不得不降低操作速度,这将大大降低系统的动态响应性能^[26]。

3.2.2 监督控制

在监督控制方式中,操作人员以人一机对话的方式进行机器人作业信息的收集并对机

机器人下达各种指令，使与计算机有密切联系的传感器和驱动器有效的进行工作^[9]。操作人员和计算机 A、计算机 B 构成一个信息循环—监控回路，由计算机 B、机器人和摄像机构成另外一个信息循环—闭环控制回路，这两个回路之间以反馈信息为主，只有很少量的子任务指令信息。整个系统的按操作人员预期运行需要满足以下三个条件^[27]：监控回路中的通信延迟小于每个子任务执行所需的时间，所以每个子任务应相当大；远程机器人工作环境中的无法预料的因素不能变化过快；远程机器人可靠性很高。

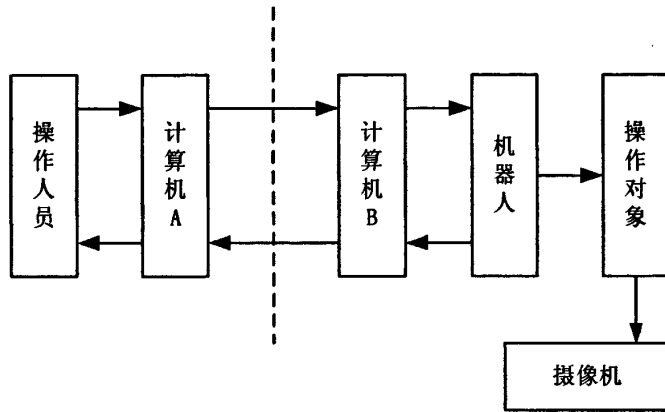


图 3.4 监督控制方式

以监督方式运行的遥操作机器人系统中，远程机器人应该是具有感知功能的机器人或者智能机器人，可以半自主方式运行。借助机器人的局部智能，及时处理远程工作环境中的突发紧急事件，提高整个系统的工作效率和适应能力。这也是目前遥操作机器人最常见的操作方式。

3.2.3 自动控制

典型的自动控制方法，如图 3.5 所示。自动控制方法中，操作人员即不需要对机器人进行监视，也不需要向机器人下达指令，远程机器人处于完全自主工作状态中。操作人员不参与控制，仅从远程机器人处获得信息，这要求远程机器人必须具有高级智能，能够完成任务的规划和执行，以及评价等工作。

这种控制方式中，操作人员也不是反馈控制回路的一部分，控制回路中也不包含通信延迟，同时，不存在操作人员和远程机器人的实时交互，因此，该控制方式是从根本上解决通信延迟最理想的方法。限于目前智能机器人的研究进展，完全自主运行的智能机器人还难以实现，因此，现实的遥操作机器人系统中很少采用这种控制方式。

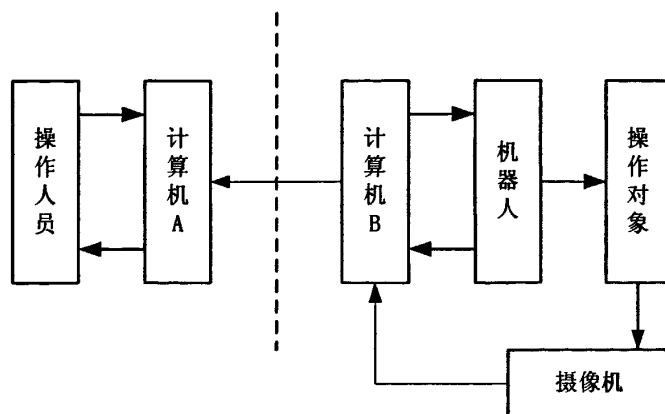


图 3.5 自动控制方式

3.2.4 其他控制

其他常见的控制方法有遥编程控制和预测显示控制。

预测显示控制的基本原理是，利用虚拟现实技术建立机器人的虚拟模型，操作人员对操作机进行操作，通过机器人模型快速执行以决定操作是否可行，如果可行则将控制命令发送远程机器人执行。通过机器人模型来实时显示虚拟运行结果，而实际的运行结果则是经过通信延迟后传回操作端的，并以图像的方式显示出来，形成虚拟模型与实际机器人图像的迭加，从而可以判断预测执行的结果与实际运行结果是否一致。预测显示控制不能消除通信延迟对遥操作机器人系统的影响，采用的是移动—等待控制策略^[28,29,30]。

遥编程控制是对预测显示控制的改进和发展。遥编程控制完全依赖于虚拟现实技术。遥编程控制的基本原理是，利用虚拟现实技术实现一个虚拟环境，将操作人员与真实的遥操作机器人系统完全隔离开，操作人员只与虚拟环境交互，由虚拟环境负责与遥操作机器人系统进行交互，如图 3.6 所示。具体过程是，在操作端的计算机 A 上建立远程机器人及其工作环境的虚拟环境，操作人员所有操作均针对虚拟环境进行，虚拟现实环境对操作人员的行为进行解释和处理，将其变换为相应的任务指令或者操作指令，机器人根据这些指令完成具体操作，远程机器人及其工作环境不向操作端返回信息，除非异常情况发生。操作人员所获得的反馈信息直接来源于虚拟现实环境，而不是远程机器人和远程环境。这样，操作人员不在实际的控制循环之中，比监督方式更远离实际的控制循环，从而避免了通信延迟对操作人员造成的影响，借助虚拟现实环境，操作人员可以获得逼真、流畅的操作。

实现遥编程控制有三个基本条件：一、利用虚拟现实技术在操作端实现虚拟环境，否

则不能有效地将操作人员和遥操作机器人系统隔离开；二、远程机器人端是结构化环境。对于在非结构化环境中工作的遥操作机器人，该方法因无法对远程环境进行实时虚拟建模，从而不适用；三、遥操作机器人系统需要高的可靠性。否则，整个系统只能进行演示，难以完成实际工作^[12]。

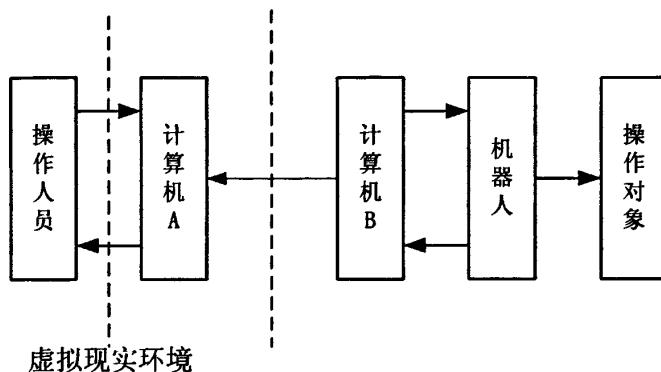


图 3.6 遥编程控制方式

3.3 C/S 与 B/S 客户端系统模式

现有的基于互联网的机器人远程控制系统一般采用两种客户端实现，一种是客户端/服务器(C/S)端应用模式，一种是浏览器/服务器(B/S)应用模式,下面详细的讨论它们的各自的优点和不足，对本系统采用的方式有鉴见意义^[25]。

3.3.1 客户端/服务器(C/S)网络应用模式

传统远程控制系统一般采用客户机/服务器模式。其最初的结构为二层应用结构，系统由客户端(Client)应用程序和服务器(Server)组成。客户机是指运行用户服务请求程序，并将这些请求传送到服务器的计算机。服务器是指管理数据资源，响应并受理由客户机发出的请求，并将计算结果传送给客户机的计算机。服务器可接受多个客户机的多个请求，将请求排队或同时处理。服务器用于运行服务器程序，响应并执行来自前端客户的服务请求，最后向前端返回计算结果；而客户机，前端运行客户端程序，向服务器发出请求。二层应用结构的优点在于^[32]：

(1)客户机和服务器基于网络通信机制实现了任务的分工和资源的共享。

(2)通过而向连接的网络协议，在保证通信带宽的前提下，客户机/服务器结构提高了系统的实时性；

(3)能够满足用户端和服务器端之间的双向功能触发；

(4)软件模块的分发授权容易控制。

随着网络技术的发展，在二层应用结构的基础上，将企业逻辑从客户端划分出来形成了客户机/服务器结构的三层应用结构，主要分为：用户界面、企业逻辑、数据库。其具体结构如图 2.9 所示。

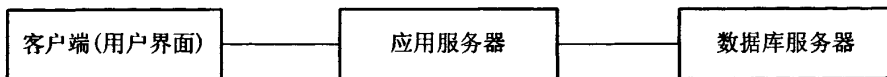


图 2.9 客户机/服务器系统三层结构模型

客户端主要是用户界，而应用服务器中存放的是企业逻辑，即对数据库的一切操作。客户端通过代理对象向应用服务器提出对数据库操作的请求，由应用服务器执行相应的数据库操作并将结果返回到客户端。三层客户机/服务器结构与传统的一层客户机/服务器结构相比优势主要在：

(1)安全性增强：由于应用服务器把客户端与数据库服务器分开了，客户不能直接访问数据库服务器。增强了系统的安全性。

(2)效率提高：应用服务器减轻了客户端的负担，也降低了数据库服务器的连接代价。

(3)可伸缩性：三层结构强调逻辑意义而不是物理意义，因此它们的硬件系统构成不受限制，具有很大的灵活性。

(4)易于维护：由于应用逻辑被封装到了应用服务器中。当应用逻辑发生变化，仅需修改应用服务器中的程序，客户端的程序无需更新，维护代价大大降低。

虽然客户机/服务器结构将数据存放在服务器上，应用程序存放在客户端计算机上解决了数据共享等问题，但却具有不易维护、操作风格不一致、处理的数据类型不易扩展等缺点。尽管后来将企业逻辑从客户端划分出来，但随着网络应用系统的发展，传统客户机/服务器模式应用系统的缺点还是不断地暴露出来：

(1)客户机/服务器结构的软件系统中各功能模块都是专用型的，系统功能的表现依赖于各功能模块的组合装配关系，这种装配关系会随着业务空间位置的延伸而显得烦琐和困难，系统的发布方式有很大的局限性；

(2)客户机/服务器系统大多固定于一种运行平台，平台间的可移植性差；

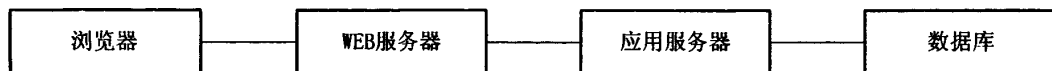
(3)开发周期长，升级维护困难。

3.3.2 浏览器/服务器(B/S)多层分布式系统模式

由于客户机/服务器应用结构的局限性,随着网络技术的发展,促使了浏览器/服务器模式应用系统的产生和发展。浏览器/服务器模式应用系统由浏览器(Browser)和服务器(WebServer, OtherServer, Middle Ware)组成。数据和应用程序分别存放在应用服务器和数据库服务器上,浏览器可以通过 Web 服务器调用应用服务器程序,从而得到动态的结果。

典型的浏览器/服务器结构的应用主要由四大部分组成:浏览器、Web 服务器、应用服务器、数据库服务器及其它企业系统。其具体结构如图 2.10 所示。

随着网络技术和互联网的迅速普及,网络业务信息交互平台越来越多地选择基于浏览器/服务器结构。浏览器/服务器结构具有开放性、灵活性和易用性的特点。以浏览器/服务器模式开发的系统维护上作集中在服务器上,客户端不用维护,操作风格比较一致,只要有浏览器的合法用户都可以十分方便地使用。对于用户来说,面对的是界面统一、易于操作并且与平台无关的浏览器,不需要接受过多的操作培训,也不需要经历烦琐复杂的配置过程,在任何联网的地方都可以处理业务;对于开发者来说,毋需开发专用的客户端软件,客户端功能模块经过 Web 服务器就可以发布,系统的升级和维护都变得简单化。如果解决了浏览器/服务器结构中的实时通信问题,就可以构造一个以浏览器作为通用客户端的浏览器/服务器结构的远程监测系统,监测人员可以在任何与互联网相连的地方进行实时监测活动,这无疑会以一种低成本的方式拓展监控系统的空间分布范围。



2.10 B/S 应用结构

3.4 小结

本章归纳了目前几种网络远程控制系统体系结构的区别和特点,分析了不同控制系统在不同条件下的优点和不足。也对机器人控制模式进行了详细的分析,最后介绍基于 Internet 的网控机器人采用的客户端/服务器端 C/S、浏览器/服务器 B/S 模式。

第四章 基于 SolidWorks-VRML 构建虚拟环境中机器人模型

4.1 SolidWorks 系统简介

SolidWorks 软件是由 SolidWorks 公司推出的世界上第一个专门在 Windows 环境下进行机械设计的三维 CAD 软件，其底层核心技术是在 Windows 环境下开发完成，具有强大的实体建模功能和用户熟悉的 Windows 操作界面。

4.1.1 装备设计建模

在 SolidWorks 的装配设计中可以直接参照已有的零件生成新的零件。不论设计用“自顶而下”方法还是“自底而上”的方法进行装配设计，SolidWorks 都将以其易用的操作大幅度地提高设计的效率。快速的装配设计有一个鼠标引导的自动装配对准功能，可以捕捉要定义装配关系的位置。能观察完全动态地装配设计中可运动的零部件的运动形式。在调用大装配时，“轻量化”零部件的功能极大地提高了运行速度。富有创意的装配管理器和产品配置管理器可对整个装配过程进行管理。这使得系列产品和同类零部件基于单一的设计，文档的建立和编制从而更加合理和高效。设计直接面向产品配置，通过产品配置管理器，设计者可以建立和修改指定产品配置，几何形状、装配关系、零部件颜色和其它属性都能在产品配置中控制。装配模型支持设计表，可用设计表创建和驱动产品配置，也可通过设计表的参数来控制产品配置的选择。支持多重的子装配；在装配模型中，可进行前后关联的设计。在进行自动装配对准或关联的设计时，直接或间接地导出装配约束关系。

4.1.2 零件设计建模

SolidWorks 有全面的零件实体建模功能，其丰富程度有时会出乎设计者的期望。变量化的草图轮廓绘制，并能够自动进行动态过约束检查。用 SolidWorks 的拉伸、旋转、倒角、

抽壳和倒圆等功能可以更简便地得到要设计的实体模型。高级的抽壳可以在同一实体上定义不同的抽壳壁厚。在用户可定义坐标系,能自动计算零部件的物性和进行可控制的几何测量。用高级放样、扫描和曲面拱顶等功能可以生成形状复杂的构造曲面。通过直接对曲面的操作,能控制参数曲面的形状。通过简单地取并延伸分型线,能生成非平面的分型面。典型应用是模具的设计。在三维模型上标注,标注的内容支持超级连接。把有公共边界的曲面缝接成单一曲面。所有特征都可以用拖动手柄改变尺寸,并有动态的形状变化预览。从独特的、支持 Internet 的特征模板中用拖动放置的操作引用特征。变半径倒圆、指定区域倒圆、填角和圆角过渡。特征管理器对模型捕捉设计意图,双重支持几何选择,拖动放置特征换序。支持产品配置的控制和设计表的系列定义。

4.1.3 数据转换

通过数据转换接口, SolidWorks 可以很容易地将目前市场几乎所有的机械 CAD 软件集成到现在的设计环境中来。支持的数据标准有: IGES、DXF、DWG、SAT (ACSI)、STEP、STL、ASCII 或二进制格式、VDAFS(VDA)、VRML、Parasolid 等。

为比较评价不同的设计方案,减少设计错误,提高产量, SolidWorks 强劲的实体建模能力和易用友好的 Windows 界面形成了三维产品设计的标准。机械工程师不论有无 CAD 的使用经验,都能用 SolidWorks 提高工作效率,使企业以较低的成本、更好的质量更快将产品投放市场^[33]。

4.2 VRML 语言

4.2.1 VRML 语言概述

VRML 是 Virtual Reality Modeling Language 的缩写,即虚拟现实建模语言。这是一种用于在 Internet 上构筑 3D 多媒体世界和共享虚拟世界的开放式语言标准,是早期桌面虚拟现实技术的典型代表。VRML 的基本目标是建立因特网上的交互式三维多媒体,基本特征包括分布式、三维、交互性、多媒体集成、境界逼真性等。

1994 年 3 月在日内瓦召开的第一届 WWW 大会上,首次正式提出了 VRML 这个名字。

1994 年 10 月在芝加哥召开的第二届 WWW 大会上公布了规范的 VRML 1.0 标准。1996 年 8 月在新奥尔良召开的 SIGGRAPH96 会议上公布了规范的 VRML2.0 标准。1997 年 12 月 VRML 作为国际标准正式发布。1998 年 1 月正式获得国际标准化组织 ISO 批准, 简称 VRML97。1999 年底, VRML 的又一种编码方案 X3D 草案发布。X3D 整合正在发展的 XML、JAVA、流技术等先进技术, 包括了更强大、更高效的 3D 计算能力、渲染质量和传输速度。2000 年 6 月 Web3D 防会发布了 VRML2000 国际标准。2002 年正式发表了 X3D 标准和相关的 3D 浏览器^[34]。

4.2.2 VRML 语言的特性

VRML 的基本特性有^[35]:

(1)基于客户端/服务器模式的访问方式。服务器作为中心工作站负责协调绝大多数仿真活动, 提供 VRML 文件及其资源, 并维护环境中所有虚拟对象的状态。客户端则通过 Internet 下载期望访问的文件, 运行客户进程, 管理用户化身, 实用输入/输出工具进行交互, 并通过本地平台上的 VRML 浏览器实时完成场景绘制和系统交互。

(2)分布式环境设计及可扩充性。VRML 文件的一个重要特性是其内建了支持多个分布式文件的内联机制, 通过内联机制嵌入其他.wrl 文件, 从而允许将一个巨大的虚拟环境进行分割设计而后合并展示。

(3)ASCII 文本格式的描述性语言。VRML 与 HTML 相同, 使用 ASCII 文本格式的三维场景式描述语言进行编程。这在保证各种平台通用的同时, 也降低了数据通信量, 从而提高网络传输速率。而且当 VRML 在本地机器人运行时, 由于不再受网络带宽及传输速率的限制, 其效果更佳。

(4)增强了的静态场景。VRML2.0 新增的一些功能节点可以使三维静态场景的真实感更强。

(5)可交互性与动画。VRML2.0 设置的接触传感器、环境传感器、感知传感器以及碰撞传感器可用于对三维造型进行实时交互仿真, 以避免物体穿墙而过等现象的发生; 而系统动画插补器则可以在场景中实时渲染关键帧动画。

(6)Java 或 JavaScript 支持。VRML 通过 Script 节点引入 Java 或 JavaScript 语言编写的脚本程序来扩展其功能。Java 与 VRML 具有很强的互补性; Java 具有 Internet 环境下程序设计的优势, 而 VRML 具有虚拟现实场景建模的优势, 两者结合可开发出较为复杂的、交

互性强的虚拟三维系统^[36]。

4.2.3 VRML 的文件构成及分类

VRML 文件由文件头语句、节点语句、事件路由语句和注释语句构成。

场景图由各种节点组成。这些节点是对现实场景的物体在视觉与某些相互作用关系的描述。例如物体的形状、颜色可由 Shape 等节点的域及其属性的节点来描述，而它与其他物体间的相互作用可以通过另一些可产生事件的节点和事件路由传输来描述。

组成场景图的第一类节点是按一定的层次和编组关系组合在一起，从视觉和听觉角度来描述对象的。这些节点可以是 VRML 的内部节点，也可以是自定义的原型节点即 PROTO 节点，还可以是在该 VRML 文件外实现了的原型节点。第二类节点用于产生动态场景，或可与用户信息交互的场景，是由事件的产生和传输来驱动的。节点定义包括节点名、节点类型名和节点域以及事件的定义^[37]。

VRML97 规范定义了 54 种基本节点类型，创作者也可以通过原型机制定义自己的节点类型。如造型节点(Shape、Appearance、Material)、空间变换节点(Transform)、传感器节点(TouchSensor、CylinderSensor、PlaneSensor、SphereSensor)、背景节点(Background)、脚本节点(Script)以及用于创建新节点类型的节点(PROTO、EXTERNPROTO、IS)。

4.2.4 VRML 支持的脚本语言编程

脚本语言的编程方式结构简单，易于编写，不需要编译，因而运行方便。缺点是难以实现复杂的场景控制。

Script 节点允许用户创建自己的域和事件，然后编写脚本或通过用户的指令序列进行操作，实现 VRML 编程。VRML 支持的脚本编程语言有：

(1) VRMLScript 语言：该语言是一种 CosmoPlayer(浏览器插件)所支持的语言。它实际上是 JavaScript 语言的子集。它非常灵活，是在任何 VRML 浏览器中都可用的最简单的语言。

(2) JavaScript 语言：该语言是解释型的，以对象为基础的，易于使用 and 学习的脚本语言。它的语言和基本结构与 Java 是相似的，但两者的功能范围以及编程风格大不相同。

(3) ECMAScript 语言：它是一种通用的，跨平台的编程语言，可进行脚本编程以及对

外事件，对象，行为进行定义^[38]。

4.3 基于 SolidWorks-VRML 实现机器人的虚拟建模

虚拟模型的设计，一方面要求能呈现复杂形体的内、外结构形状；另一方面也要求能展示模型的组成和装配关系。基于 SolidWorks-VRML 建立机器人的虚拟模型包括以下 3 个步骤：

(1)虚拟模型的组成。对于简单形体，可直接创建一整体三维模型；对于可剖分实体，应以剖切面为界作两部分处理；对于装配形体、复杂组合形体，应分别创建各组成部分。

(2)制作虚拟模型。借助 SolidWorks 三维处理软件准确地构造虚拟模型的各部分，若分别储存为几个图形文件，它们的坐标系统要相同，以便准确定位。但这样生成的图形文件一般都不能直接用网络浏览器打开，需要进行格式转换，生成以 WRL 为后缀的 VRML 文件，才能方便地在 Internet 网上传输、浏览。

(3)赋予虚拟模型动态组合、动态分离以及其它虚拟现实特性。按照 VRML 的国际标准，充分利用 VRML 的多种功能，在文本编辑工具中编辑上面生成的*.wrl 文件，通过反复调试，完成一虚拟模型的制作^[39]。

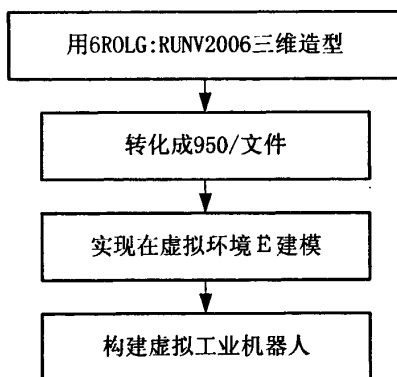


图 4.1 基于 SolidWorks-VRML 实现机器人在虚拟环境中建模流程

4.4 MOTOMAN-HP3 机器人模型的建立

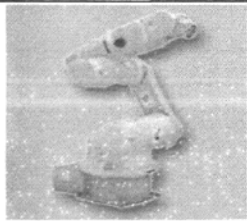
本控制系统使用日本安川工业机器人 MOTOMAN-HP3 作为实验研究对象，它的动作速度、精度及可靠性体现了机器人的先进水平。

4.4.1 MOTOMAN-HP3 机器人简介

MOTOMAN-HP 系列机器人具有很快的轴动作速度。轻型机体和具备高轨迹精度控制及振动抑制控制的 NX100 控制柜的有机结合,减弱了机器人启动和停止瞬间的颤动,从而缩短了机器人的运行周期。HP 系列的机器人操作机具备最大的工作半径和最小的干涉半径,工作范围变大,在系统设计上提供了较大的灵活性。机器人后方的工作范围的扩大,HP3 可以提供更广泛的应用方案,使夹具、剪丝机等设备都是更效的安装方式。

它的控制柜(NX100)运用高精度轨迹控制算法缩短了对命令响应的滞后时间。误差补偿功能使机器人绝对位置精度提高 2—5 倍。机器人的轨迹重复精度可提高 50%。NX100 控制柜可协调控制多达 36 个轴。它也为多机器人协调以及多个机器人接入网络控制平台创造了有利条件^[40]。

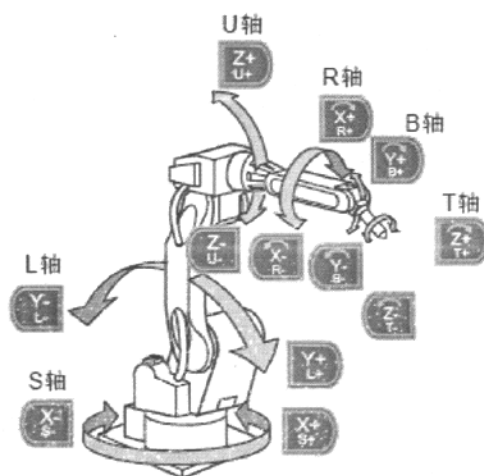
表 4.1 MOTOMAN-HP3 基本参数

	机械结构	垂直多关节型(6 自由度)
	载荷重量	3kg
	定位精度	$\pm 0.03\text{mm}$
	安装方式	地面安装
	本体重量	45kg
	电源容量*	1kVA
最大动作范围	S 轴(回旋)	$\pm 170^\circ$
	L 轴(下臂倾动)	$+150^\circ \sim -45^\circ$
	U 轴(上臂倾动)	$+210^\circ \sim -152^\circ$
	R 轴(手臂横摆)	$\pm 190^\circ$
	B 轴(手腕俯仰)	$\pm 125^\circ$
	T 轴(手腕回旋)	$\pm 360^\circ$
最大速度	S 轴	3.66rad/s, 210°/s
	L 轴	3.14rad/s, 180°/s
	U 轴	3.93rad/s, 225°/s
	R 轴	6.54rad/s, 375°/s
	B 轴	6.54rad/s, 375°/s
	T 轴	8.73rad/s, 500°/s
允许力矩	R 轴	7.25N·m
	B 轴	7.25N·m
	T 轴	5.21N·m
允许惯性力矩 ($\text{GD}^2/4$)	R 轴	0.30kg·m ²
	B 轴	0.30kg·m ²
	T 轴	0.1kg·m ²

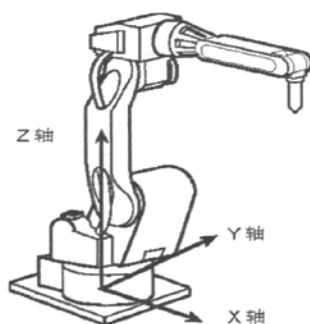
MOTOMAN-HP3 工业机器人的运动方式在不同的坐标系下会有不同的运动趋势,这

些坐标系包括：直角坐标系、关节坐标系、圆柱坐标系、工具坐标系和用户坐标系，如图 4.2 所示。

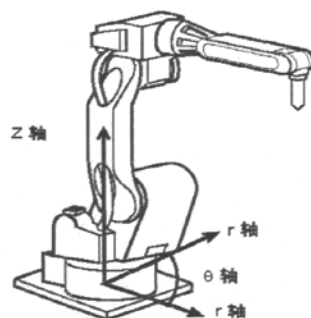
- 关节坐标系机器人各轴进行单独动作，称关节坐标系。
- 直角坐标系不管机器人处于什么位置，均可沿设定的 X 轴、Y 轴、Z 轴平行移动。
- 圆柱坐标系中， θ 轴绕 S 轴运动，R 轴沿 L 轴臂、U 轴臂轴线的投影方向运动，Z 轴运动方向与直角坐标完全相同。
- 工具坐标系工具坐标系把机器人腕部法兰盘所持工具的有效方向作为 Z 轴，并把坐标定义在工具的尖端点。
- 用户坐标系机器人沿所指定的用户坐标系各轴平行移动。在关节坐标系以外的其他坐标系中，均可只改变工具姿态而不改变工具尖端点(控制点)位置，



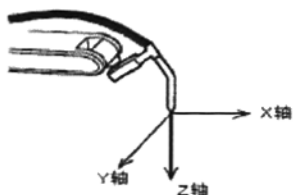
(a) 关节坐标系



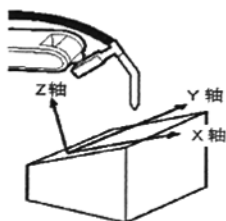
(b) 直角坐标系



(c) 圆柱坐标系



(d)工具坐标系

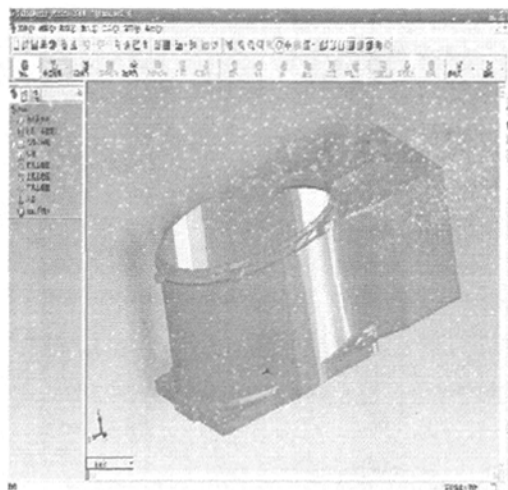


(e)用户坐标系

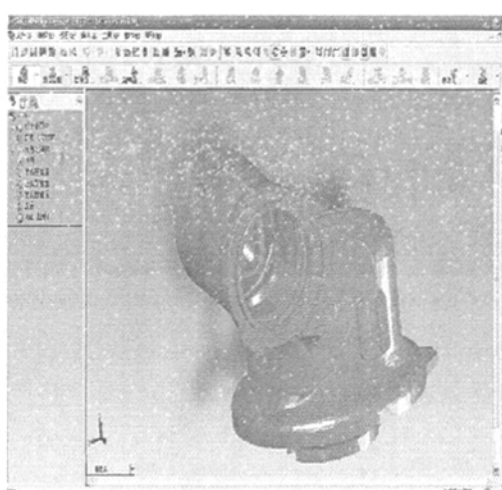
图 4.2 机器人的坐标系

4.4.2 用 SolidWorks2006 建立机器人模型

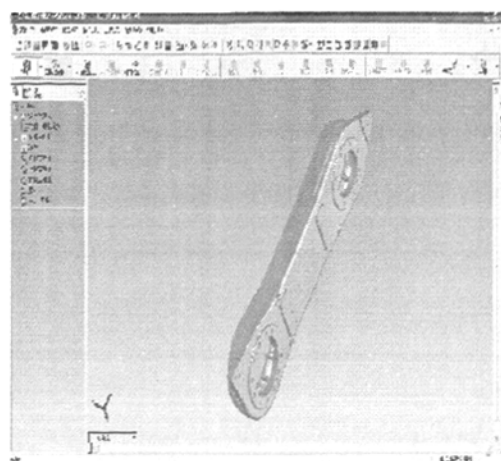
SolidWorks2006 三维 CAD 软件用特征造型技术取代了传统布尔运算,使机器人在几何造型上过的计算更加简洁、高效。根据 MOTOMAN-HP3 的结构组成可以将其分解成底座、S 轴、L 轴、U 轴、R 轴、B 轴、T 轴等实体模型,先将他们单独进行三维建模。



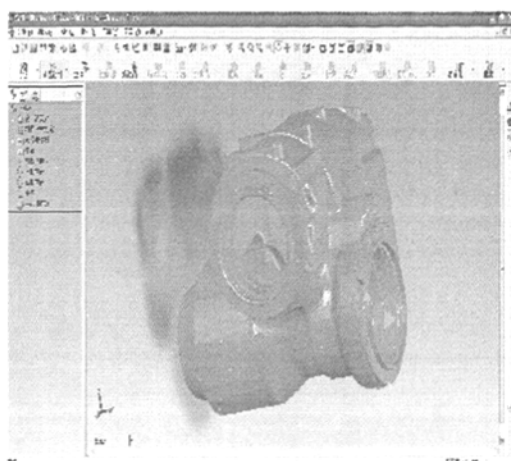
(a) 底座



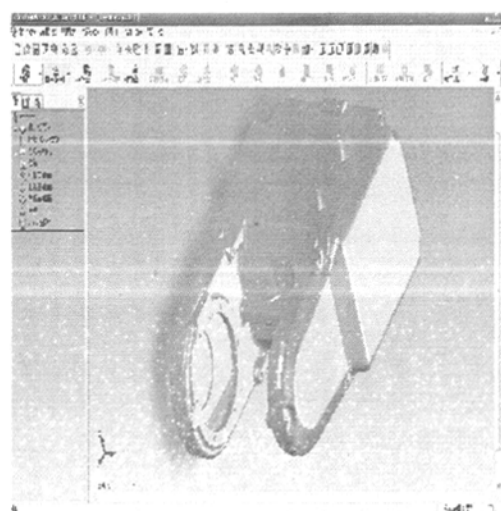
(b) S 轴



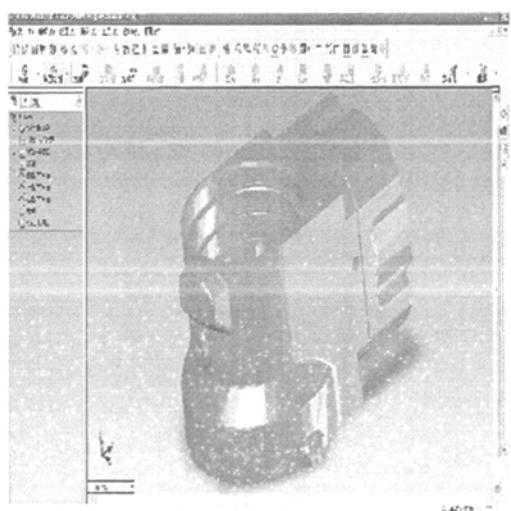
(c) L 轴



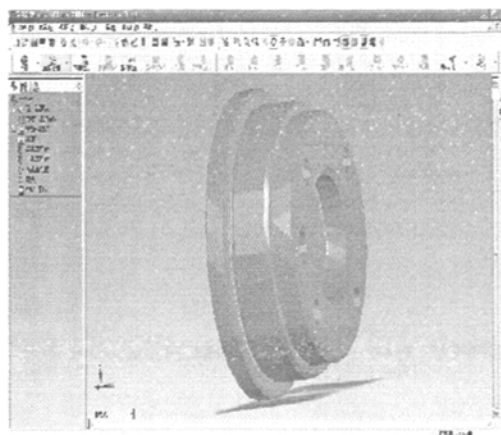
(d) U 轴



(e) R 轴



(f) B 轴



(g) T 轴

图 4.3 MOTOMAN-HP3 机器人的各组成部分实体模型

4.4.3 机器人在虚拟环境中建模

由于 VRML 造型具有较强的移植功能,具有便于网上传输、运动图像编程简单等特点,将其应用实现 MOTOMAN-HP3 机器人在虚拟环境中建模。

在用 SolidWorks2006 建立机器人三维实体模型后,利用 SolidWorks 的 VRML 转换程序输出为 VRML 文件:“文件—另存为:保存类型—VRML(*.wrl)—选项...文件格式:VRML—版本:VRML97”,即可输出为 VRML 文件。

利用 SolidWorks-VRML 实现 MOTOMAN-HP3 机器人虚拟现实的精确建模,大大减少了建模的工作量,提高了工作效率,更重要的是能够满足虚拟现实场景中对实体的精确度的要求,虚拟机器人模型的精确程度直接关系到过程控制的效果。

在 Cortona 浏览器的基础上,在虚拟环境中构建可视的实体模型,如图 4.4 所示。

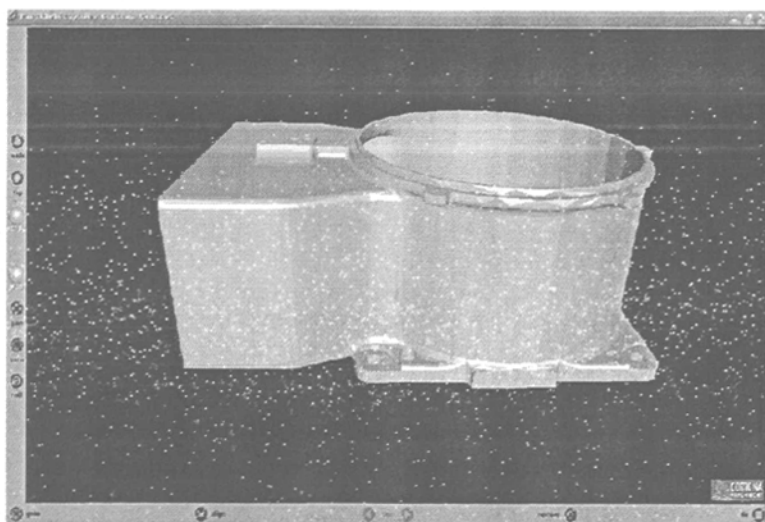


图 4.4 虚拟环境中的机器人底座

Cortona 浏览器功能十分强大,使用方便,能够直接插入 Windows 操作系统的 IE 浏览器使用,而且 3D 效果、交互性能近乎完美,是目前使用最多的 VRML 浏览器。Cortona 浏览器界面包括两部分:(1)控制栏,有两排按钮,分别位于浏览器窗口的左边和底边,能够进行浏览模式、控制方式、观察视点等的切换,实现用户与虚拟环境的完美交互;(2)三维窗口,显示 3D 场景画面。

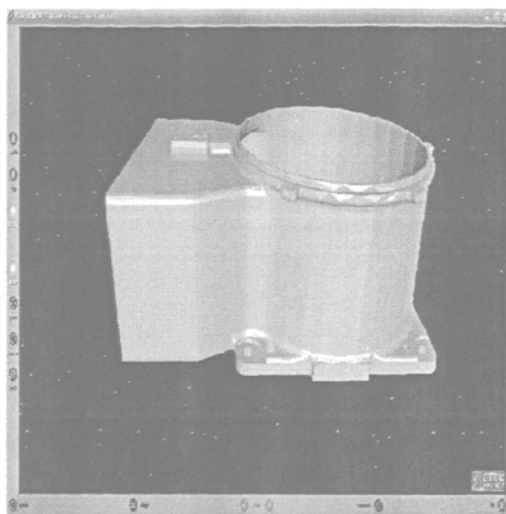
4.4.4 虚拟机器人的构建

在虚拟环境中对 MOTOMAN-HP3 机器人的各组成实体进行建模后, 利用 VRML 作为建构的工具, 再进行虚拟工业机器人的构建, 这时就需要在源程序中对各实体模型作以修改。

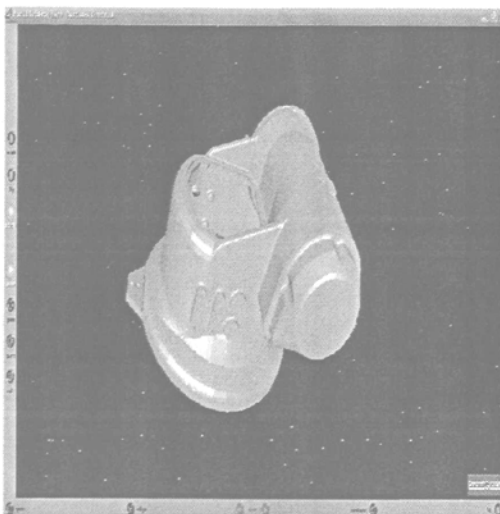
VrmlPad2.1 支持 VRML97 标准。通过 VrmlPad2.1 可以对 VRML 文件进行浏览编辑, 对资源文件进行有效的管理, 并且提供了 VRML 文件发布向导, 可以帮助开发人员编写和发布自己的 VRML 虚拟现实作品。另外, 它提供的插件功能也能方便地完成 VRML 模型创建。

以 MOTOMAN-HP3 机器人 S 轴为例, 利用 VrmlPad2.1 虚拟建模软件修改后的程序如下:

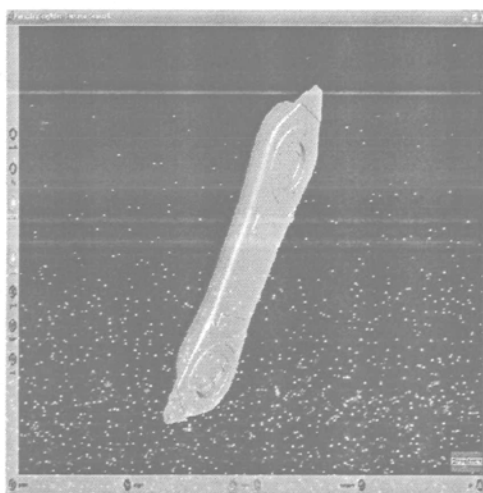
```
#VRML V2.0 utf8
Shape {
  appearance Appearance {
    material Material {
      diffuseColor .....
      emissiveColor .....
      specularColor .....
      shininess .....
      transparency .....
    }
  }
  geometry IndexedFaceSet {
    coord Coordinate {
      point [
        .....
      ]
    }
    coordIndex [
      .....
    ]
  }
}
Shape {
  .....
}
```



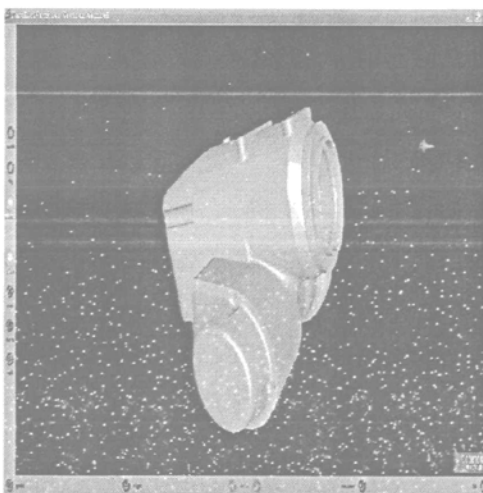
(a) 虚拟环境中的底座



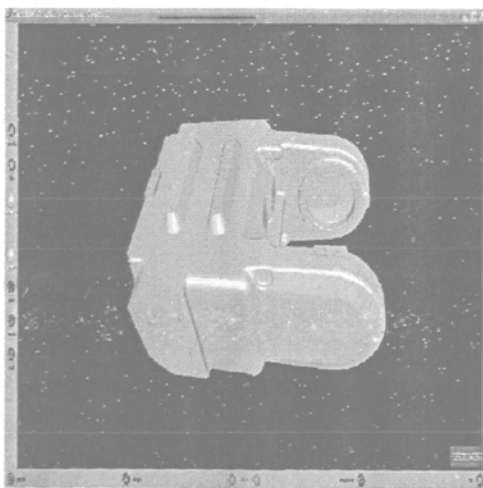
(b) 虚拟环境中的 S 轴



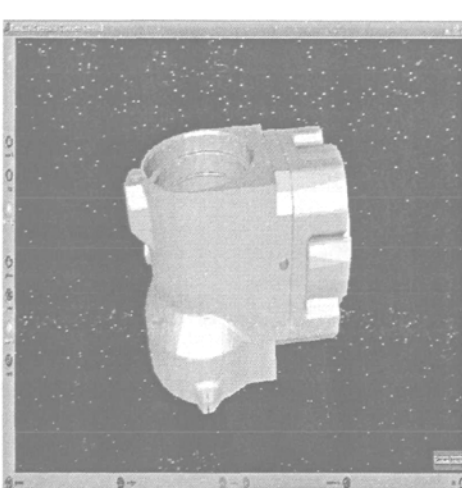
(c) 虚拟环境中的 L 轴



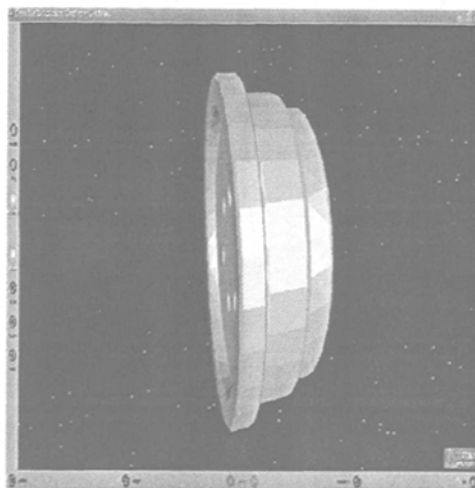
(d) 虚拟环境中的 U 轴



(e) 虚拟环境中的 R 轴



(f) 虚拟环境中的 B 轴



(g) 虚拟环境中的 T 轴

图 4.5 MOTOAMN-HP3 机器人各组成部分在虚拟环境中的模型

在虚拟机器人中的构建中，使用 VrmIpad2.1 软件进行编程，程序设计思想如下：采用嵌套式结构，建立 Transform 节点，每个 Transform 节点对应一个实体模型；实体模型的导入通过增加节点实现，同时这些节点逐层嵌套包含；虚拟机器人的虚拟装配由 Transform 节点的 translation 域、rotation 域和 scale 域控制，进行平移、旋转和缩放变换，以达到外形尺寸的一致性，这样实现了 MOTOMAN-HP3 机器人在虚拟环境中的构建。

VRML 文件代码编写如下：

```
#VRML V2.0 utf8
DEF Robot Transform {
    translation 0.0 0.0 0.0
    rotation 0.0 0.0 0.0 0.0
    scale 1 1 1
    children [
    DEF base Transform {
        center .....
        rotation .....
        children [
            Group {}
        ]
    }
    ]
}
DEF s-zhou Transform {
    center .....
    rotation .....
    children [
        Group {}
    ]
}
```

```

}
DEF l-zhou Transform {
    center .....
    rotation .....
    children [
        Group {}
    ]
}
DEF u-zhou Transform {
    center .....
    rotation .....
    children [
        Group {}
    ]
}
DEF r-zhou Transform {
    center .....
    rotation .....
    children [
        Group {}
    ]
}
DEF b-zhou Transform {
    center .....
    rotation .....
    children [
        Group {}
    ]
}
DEF t-zhou Transform {
    center .....
    translation .....
    rotation .....
    children [
        Group {}
    ]
}
}

```

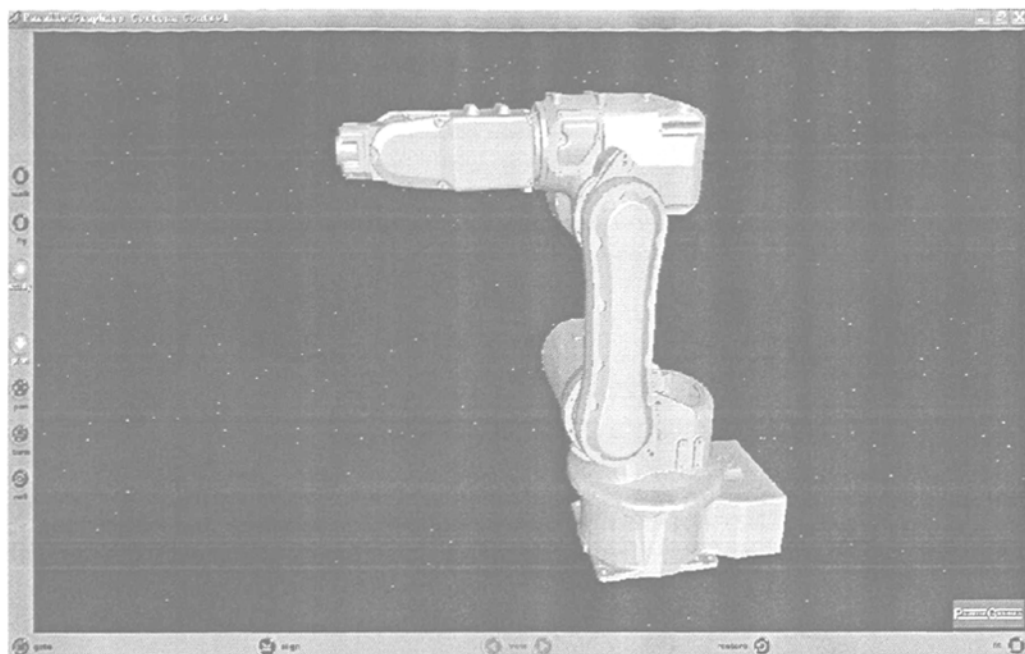


图 4.6 虚拟环境中的机器人模型

4.5 小结

基于 SolidWorks-VRML 的虚拟建模方法采用 Transform 节点分层的方法引入各实体模型的 VRML 文件，是 VRML 程序设计模块化，这也是软件工程的设计思想，采用结构化、模块化、层次化，以提高软件的设计质量；另外可以大大节省空间，减小文件的长度，给程序的调试和定义动画带来很大的方便；也可进一步用于各种支持 VRML 文件格式的开发程序中，有利于对组成虚拟机器人的各实体模型再进行脚本编程，实现运动控制的虚拟仿真。

第五章 基于虚拟现实技术的遥操作机器人系统的设计方案

5.1 系统的结构和功能设计

根据第三章中对各个不同机器人网络控制平台的研究,总结了优缺点,提出了本系统平台。此遥操作机器人系统的设计采用的是分布式机器人控制系统,把单一服务器上的功能模块分解到其它的服务器上,这样方便了机器人功能的扩张,并且提高了整体的运行效率。本遥操作机器人系统控制平台按照功能设计为如下三个服务器:应用程序服务器、数据库服务器和机器人控制服务器。各个服务器主要功能如下:

(1)应用程序服务器:负责虚拟现实环境运动仿真,并与机器人控制服务器进行通讯发送和接收数据;

(2)数据库服务器:负责提供存储、管理系统运行中所要用到的数据,主要包括系统信息数据和机器人运动状态数据。

(3)机器人控制服务器:负责接收和发送应用程序服务器端控制命令以及应用层协议命令的解析,并直接调用机器人服务器相应的控制模块。机器人控制服务器通过串口或网线与 Motoman-HP3 进行通讯。

图 5.1 为遥操作机器人系统控制平台结构图:

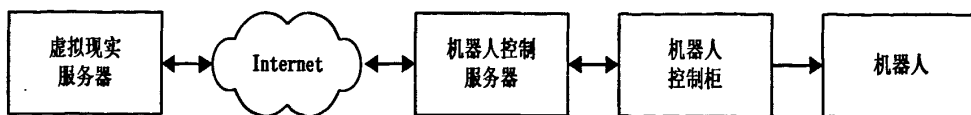


图 5.1 基于虚拟现实技术的遥操作机器人系统结构图

我们从上文中已经分析了 B/S 和 C/S 两种客户端的实现,这两种控制模式都有自己的优点和缺点,根据不同的应用特点可以进行针对性的选择。由于本系统采用的是分布式架构的机器人控制平台系统,每一个功能都有相对的应用服务器来进行控制,因此该系统采用遥编程控制方式和客户端/服务器(C/S)控制模式,其中 Client 客户机端为虚拟现实环境仿真平台。

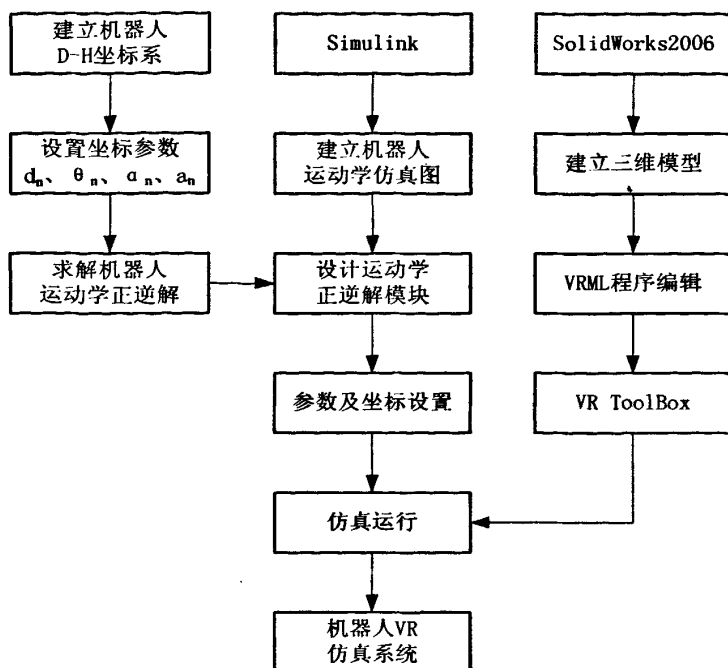


图 5.2 C/S 模式中 Client 流程图

Client 端的流程图如图 5.2 所示，用户通过 VRML 建立机器人虚拟模型，结合 Matlab/Simulink 工具进行机器人虚拟仿真，再将机器人运行结果发送给机器人。

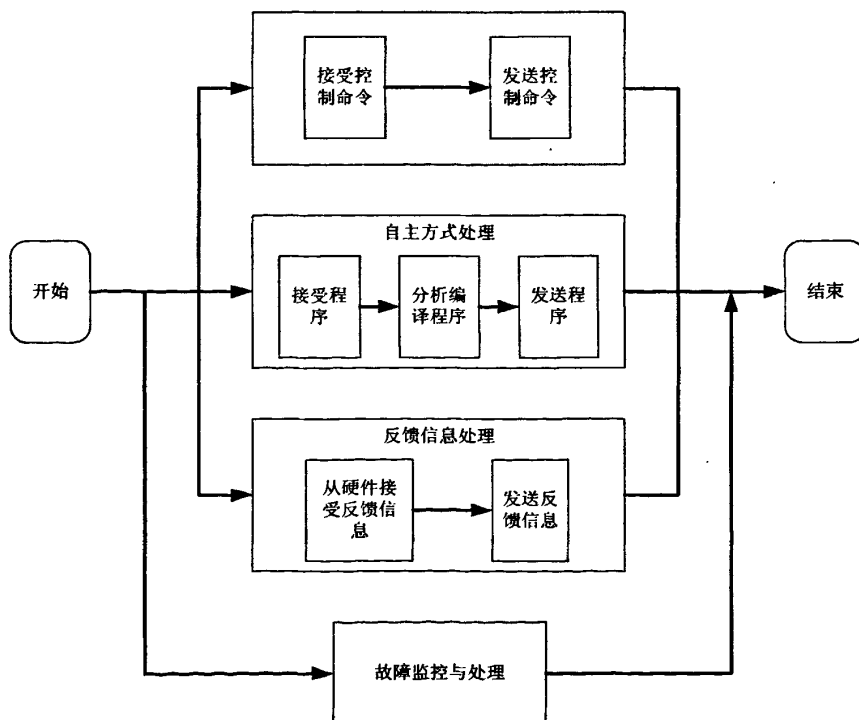


图 5.3 C/S 模式中 Server 流程图

Server 端的流程图如 5.3 所示，客户端发送命令，进入相应的控制模块，进行控制，

发生异常就调用故障处理模块。通过前文对机器人的几种控制方式研究中，了解到各种控制方式的优缺点，结合本遥操作机器人平台的特点，将采用遥编程控制方式。利用虚拟现实技术实现一个虚拟环境，将操作人员与真实的遥操作机器人系统完全隔离开，操作人员只与虚拟环境交互，由虚拟环境负责与遥操作机器人系统进行交互。机器人接到命令后，自主得执行相应得命令。

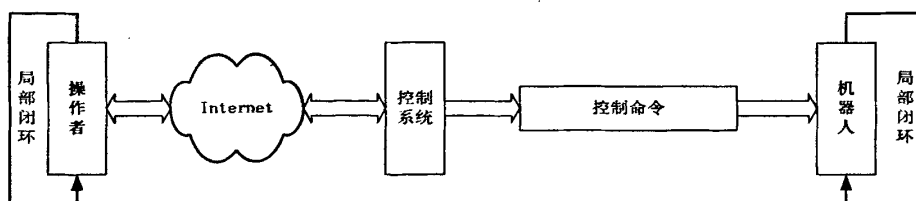


图 5.4 控制系统的流程图

5.2 机器人运动学分析

5.2.1 机器人正向运动学求解

Motoman-HP3 型机器人具有 6 个自由度，各关节均为旋转关节，如图 5.5 所示，为分析其运动学问题，结合机器人的具体结构，建立机器人 D-H 坐标系如图 5.6 所示，表 5.1 中列出的是根据建立好的坐标系得到的 D-H 参数 d_n ， α_n ， a_n ， θ_n 。

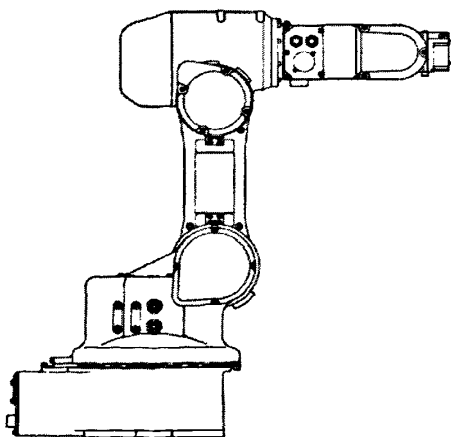


图 5.5 Motoman-HP3 型机器人

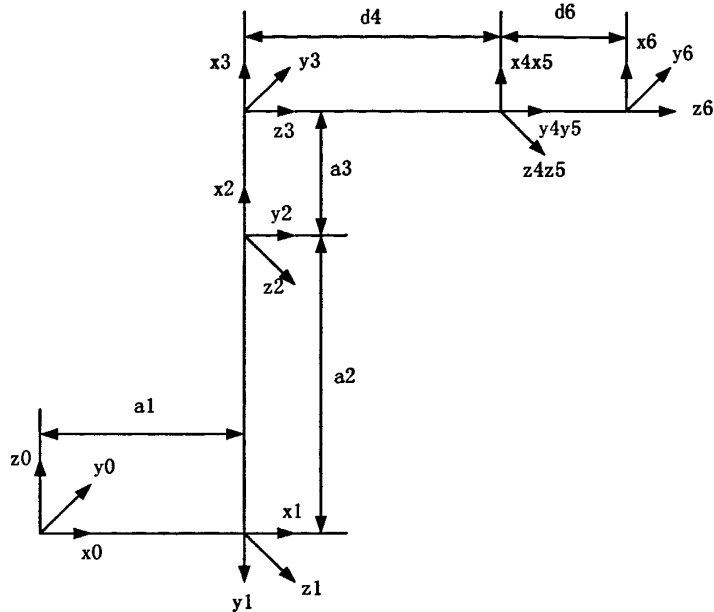


图 5.6 Motoman-HP3 型机器人坐标与参数

按 D-H 法确定杆件坐标系，可取坐标系 n 的 z 轴与关节 $n+1$ 的轴线重合， x 轴取为相邻 z 轴的公垂线， y 轴则按右手法则确定。任何一个连杆 n ，两端有关节 n 和 $n+1$ ，该连杆可用两个量来描述：一是两个关节轴线沿公垂线的距离 a_n ，即连杆长度；另一是在垂直 a_n 的平面内两个轴线的夹角 α_n ，即连杆扭角。这两个参数为连杆本身的参数。再考虑杆 n 和杆 $n+1$ 的关系，若它们通过关节 n 相连，其相对位置可用两个参数 d_n 和 θ_n 来确定，其中 d_n 是沿关节 n 的轴线两个公垂线的距离， θ_n 是垂直于关节 n 轴线的平面内两公垂线的夹角。这是表达相邻杆件相互关系的两个参数^[41]。

表 5.1 Motoman-HP3 型机器人坐标参数

关节	d_n (mm)	α_n (°)	a_n (mm)	θ_n (°)	θ_n 范围 (°)
1	0	90	100	0	-170~+170
2	0	0	290	90	-135~+60
3	0	90	85	0	-242~+130
4	300	-90	0	0	-190~+190
5	0	90	0	0	-125~+125
6	90	0	0	0	-360~+360

根据本坐标系, θ_2 的值应该加上 90° 为机器人该轴实际旋转角度。

根据表 5.1 所给的连杆参数, 以及坐标系建立结果, 可得连杆变换矩阵 A_n , ($n=0,1,2,\dots,6$)。

$$A_n = R(z, \theta_n) T(0, 0, d_n) T(a_n, 0, 0) R(x, \alpha_n)$$

$$A_n = \begin{bmatrix} \cos \theta_n & -\sin \theta_n \cos \alpha_n & \sin \theta_n \sin \alpha_n & a_n \cos \theta_n \\ \sin \theta_n & \cos \theta_n \cos \alpha_n & -\cos \theta_n \sin \alpha_n & a_n \sin \theta_n \\ 0 & \sin \alpha_n & \cos \alpha_n & d_n \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5-1)$$

方程中采用以下缩写符号:

$$s_i = \sin \theta_i, c_i = \cos \theta_i, s_{ij} = \sin(\theta_i + \theta_j), c_{ij} = \cos(\theta_i + \theta_j)$$

将表(5.1)中的参数代入式(5-1)中可得相邻坐标系之间的位姿变换矩阵如下:

$$T_0^1 = A_1 = \begin{bmatrix} c_1 & 0 & s_1 & a_1 c_1 \\ s_1 & 0 & -c_1 & a_1 s_1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad T_1^2 = A_2 = \begin{bmatrix} c_2 & -s_2 & 0 & a_2 c_2 \\ s_2 & c_2 & 0 & a_2 s_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$T_2^3 = A_3 = \begin{bmatrix} c_3 & 0 & s_3 & a_3 c_3 \\ s_3 & 0 & -c_3 & a_3 s_3 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad T_3^4 = A_4 = \begin{bmatrix} c_4 & 0 & -s_4 & 0 \\ s_4 & 0 & c_4 & 0 \\ 0 & -1 & 0 & d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$T_4^5 = A_5 = \begin{bmatrix} c_5 & 0 & s_5 & 0 \\ s_5 & 0 & -c_5 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad T_5^6 = A_6 = \begin{bmatrix} c_6 & -s_6 & 0 & 0 \\ s_6 & c_6 & 0 & 0 \\ 0 & 0 & 1 & d_6 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

根据机器人的坐标变换矩阵, 可得机器人末端的位姿:

$$T_6 = A_1 A_2 A_3 A_4 A_5 A_6 = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5-2)$$

其中:

$$n_x = \{[(c_1 c_2 c_3 - c_1 s_2 s_3) c_4 + s_1 s_4] c_5 + (-c_1 c_2 s_3 - c_1 s_2 c_3) s_5\} c_6 + [-(c_1 c_2 c_3 - c_1 s_2 s_3) s_4 + s_1 c_4] s_6$$

$$o_x = -\{[(c_1 c_2 c_3 - c_1 s_2 s_3) c_4 + s_1 s_4] c_5 + (-c_1 c_2 s_3 - c_1 s_2 c_3) s_5\} s_6 + [-(c_1 c_2 c_3 - c_1 s_2 s_3) s_4 + s_1 c_4] c_6$$

$$a_x = [(c_1c_2c_3 - c_1s_2s_3)c_4 + s_1s_4]s_5 - (-c_1c_2s_3 - c_1s_2c_3)c_5$$

$$p_x = c_1(c_2c_3 - s_2s_3)(c_4s_5d_6 + a_3) + c_1(s_2c_3 + c_2s_3)(c_5d_6 + d_4) + s_1s_4s_5d_6 + c_1c_2a_2 + a_1c_1$$

$$n_y = \{[(s_1c_2c_3 - s_1s_2s_3)c_4 - c_1s_4]c_5 + (-s_1c_2s_3 - s_1s_2c_3)s_5\}c_6 + [-(s_1c_2c_3 - s_1s_2s_3)s_4 - c_1c_4]s_6$$

$$o_y = -\{[(s_1c_2c_3 - s_1s_2s_3)c_4 - c_1s_4]c_5 + (-s_1c_2s_3 - s_1s_2c_3)s_5\}s_6 + [-(s_1c_2c_3 - s_1s_2s_3)s_4 - c_1c_4]c_6$$

$$a_y = [(s_1c_2c_3 - s_1s_2s_3)c_4 - c_1s_4]s_5 - (-s_1c_2s_3 - s_1s_2c_3)c_5$$

$$p_y = s_1(c_2c_3 - s_2s_3)(c_4s_5d_6 + a_3) + s_1(c_2s_3 + s_2c_3)(d_6c_5 + d_4) - c_1s_4s_5d_6 + s_1c_2a_2 + a_1s_1$$

$$n_z = [(s_2c_3 + c_2s_3)c_4c_5 + (-s_2s_3 + c_2c_3)s_5]c_6 - (s_2c_3 + c_2s_3)s_4s_6$$

$$o_z = -[(s_2c_3 + c_2s_3)c_4c_5 + (-s_2s_3 + c_2c_3)s_5]s_6 - (s_2c_3 + c_2s_3)s_4c_6$$

$$a_z = (s_2c_3 + c_2s_3)c_4s_5 - (-s_2s_3 + c_2c_3)c_5$$

$$p_z = (s_2c_3 + c_2s_3)(c_4s_5d_6 + a_3) - (c_2c_3 - s_2s_3)(c_5d_6 + d_4) + a_2s_2$$

由上式，结合机器人 D-H 参数，把已知的机器人的各关节的角度代入上式就可以很方便的求出机器人的运动学正解。在机器人运动学正向求解时，若能在进行求解式(5-2)的代数运算以前检验中间结果，将是十分有利的。并且，通过移动一个关节或将关节已知增量组合起来，对于这些关节矢量中的每一个都可以获得 A 矩阵对应的数值矩阵和对应的数值 T 矩阵，将其与方向矩阵对比。可证实关节变量组合的解式导致旋转关节旋转 360°。这就保证了正弦和余弦函数的符号，且对轴进行分配时确定的符号和旋转方向一致。一旦对处理结果感到满意，便计算出符号 T 矩阵。通过在已知关节矢量的情况下进行计算并将结果与已知的对应手矩阵进行对比，就可以很容易地检查乘法的正误。

5.2.2 机器人逆运动学求解

由于机器人的逆向求解并没有通用算法可利用，根据机器人的位置结构和姿态结构的特点，可引用一种避免大量逆矩阵相乘的方法，即用 $A_2^{-1}A_1^{-1}$ 左乘式(5-2)，用 A_6^{-1} 右乘式(5-2)得

$$A_3A_4A_5 = A_2^{-1}A_1^{-1}T_6A_6^{-1} \quad (5-3)$$

然后，观察式(5-4)左右各项对应元素，发现等式左边的第三行第四列的元素为常数 0，由

此可求得其中一个关节角 θ_1 。同理，利用式(5-4)左右各项对应元素相等来建立方程，从而求出各个关节角变量的值^[42]。

根据 A_n 矩阵的逆矩阵公式：

$$A_n^{-1} = \begin{bmatrix} \cos \theta_n & \sin \theta_n & 0 & -a_n \\ -\sin \theta_n \cos \alpha_n & \cos \theta_n \cos \alpha_n & \sin \alpha_n & -d_n \sin \alpha_n \\ \sin \theta_n \sin \alpha_n & -\cos \theta_n \sin \alpha_n & \cos \alpha_n & -d_n \cos \alpha_n \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5-4)$$

可得各个变换矩阵的逆矩阵如下：

$$\begin{aligned} (A_1)^{-1} &= \begin{bmatrix} c_1 & s_1 & 0 & -a_1 \\ 0 & 0 & 1 & 0 \\ s_1 & -c_1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, & (A_2)^{-1} &= \begin{bmatrix} c_2 & s_2 & 0 & -a_2 \\ -s_2 & c_2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \\ (A_3)^{-1} &= \begin{bmatrix} c_3 & s_3 & 0 & -a_3 \\ 0 & 0 & 1 & 0 \\ s_3 & -c_3 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, & (A_4)^{-1} &= \begin{bmatrix} c_4 & s_4 & 0 & 0 \\ 0 & 0 & -1 & d_4 \\ -s_4 & c_4 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \\ (A_5)^{-1} &= \begin{bmatrix} c_5 & s_5 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ s_5 & -c_5 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, & (A_6)^{-1} &= \begin{bmatrix} c_6 & s_6 & 0 & 0 \\ -s_6 & c_6 & 0 & 0 \\ 0 & 0 & 1 & -d_6 \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{aligned}$$

由 $T_6 = A_1 A_2 A_3 A_4 A_5 A_6$ 两边左乘 $A_2^{-1} A_1^{-1}$ ，右乘 A_6^{-1} 得式(5-3)。

式(5-3)左边：

$$A_3 A_4 A_5 = \begin{bmatrix} c_3 c_4 c_5 - s_3 s_5 & -c_3 s_4 & c_3 c_4 s_5 + s_3 c_5 & s_3 d_4 + a_3 c_3 \\ s_3 c_4 c_5 + c_3 s_5 & -s_3 s_4 & s_3 c_4 s_5 - c_3 c_5 & -c_3 d_4 + a_3 s_3 \\ s_4 c_5 & c_4 & s_4 s_5 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

右边：

$$A_2^{-1} A_1^{-1} T_6 A_6^{-1} = \begin{bmatrix} \dot{n}_x & \dot{o}_x & \dot{a}_x & \dot{p}_x \\ \dot{n}_y & \dot{o}_y & \dot{a}_y & \dot{p}_y \\ \dot{n}_z & \dot{o}_z & \dot{a}_z & \dot{p}_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

其中：

$$\dot{n}_x = (c_1 c_2 n_x + s_1 c_2 n_y + s_2 n_z) c_6 - (c_1 c_2 o_x + s_1 c_2 o_y + s_2 o_z) s_6$$

$$\dot{o}_x = (c_1 c_2 n_x + s_1 c_2 n_y + s_2 n_z) s_6 + (c_1 c_2 o_x + s_1 c_2 o_y + s_2 o_z) c_6$$

$$\dot{a}_x = c_1 c_2 a_x + s_1 c_2 a_y + s_2 a_z$$

$$\dot{p}_x = -(c_1 c_2 a_x + s_1 c_2 a_y + s_2 a_z) d_6 + c_1 c_2 p_x + s_1 c_2 p_y + s_2 p_z - c_2 a_1 - a_2$$

$$\dot{n}_y = (-c_1 s_2 n_x - s_1 s_2 n_y + c_2 n_z) c_6 - (-c_1 s_2 o_x - s_1 s_2 o_y + c_2 o_z) s_6$$

$$\dot{o}_y = (-c_1 s_2 n_x - s_1 s_2 n_y + c_2 n_z) s_6 + (-c_1 s_2 o_x - s_1 s_2 o_y + c_2 o_z) c_6$$

$$\dot{a}_y = -c_1 s_2 a_x - s_1 s_2 a_y + c_2 a_z$$

$$\dot{p}_y = -(c_1 s_2 a_x - s_1 s_2 a_y + c_2 a_z) d_6 - c_1 s_2 p_x - s_1 s_2 p_y + c_2 p_z + s_2 a_1$$

$$\dot{n}_z = (s_1 n_x - c_1 n_y) c_6 - (s_1 o_x - c_1 o_y) s_6$$

$$\dot{o}_z = (s_1 n_x - c_1 n_y) s_6 + (s_1 o_x - c_1 o_y) c_6$$

$$\dot{a}_z = s_1 a_x - c_1 a_y$$

$$\dot{p}_z = -(s_1 a_x - c_1 a_y) d_6 + s_1 p_x - c_1 p_y$$

现在求解运动学逆解：

(1)求 θ_1

令式(5-3)左右两边元素(3,4)相等(其中(3,4)表示该矩阵的第三行第四列的元素，下同)得：

$$-(s_1 a_x - c_1 a_y) d_6 + s_1 p_x - c_1 p_y = 0$$

$$\theta_1 = \arctg \frac{p_y - a_y d_6}{p_x - a_x d_6} + k\pi, (k=0, \pm 1) \quad (5-5)$$

(2)求 θ_3

令式(5-3)左右两边元素(1,4),(2,4)分别相等并化简得

$$-(c_1 c_2 a_x + s_1 c_2 a_y + s_2 a_z) d_6 + c_1 c_2 p_x + s_1 c_2 p_y + s_2 p_z - c_2 a_1 - a_2 = s_3 d_4 + a_3 c_3$$

$$-(c_1 s_2 a_x - s_1 s_2 a_y + c_2 a_z) d_6 - c_1 s_2 p_x - s_1 s_2 p_y + c_2 p_z + s_2 a_1 = -c_3 d_4 + a_3 s_3$$

上面两式可以变形为：

$$c_2 [-(c_1 a_x + s_1 a_y) + (c_1 p_x + s_1 p_y)] - s_2 (a_z d_6 - p_z) - c_2 a_1 = s_3 d_4 + a_3 c_3 + a_2$$

$$-s_2[-(c_1a_x + s_1a_y) + (c_1p_x + s_1p_y)] - c_2(a_zd_6 - p_z) + s_2a_1 = -c_3d_4 + a_3s_3$$

两式平方相加得

$$[(c_1a_x + s_1a_y) - (c_1p_x + s_1p_y) + a_1]^2 + (a_zd_6 - p_z)^2 = a_2^2 + d_4^2 + a_3^2 + 2a_2(a_3c_3 + d_4s_3)$$

$$\text{令 } t_1 = [(c_1a_x + s_1a_y) - (c_1p_x + s_1p_y) + a_1]^2 + (a_zd_6 - p_z)^2,$$

$$t_2 = \frac{(t_1 - d_4^2 - a_3^2 - a_2^2)}{2a_2},$$

上式整理得 $t_2 = a_3c_3 + s_3d_4$

$$\text{可求得: } \theta_3 = -\arctg \frac{a_3}{d_4} \pm \arctg \frac{\frac{(t_1 - d_4^2 - a_3^2 - a_2^2)}{2a_2}}{\sqrt{a_3^2 + d_4^2 - [\frac{(t_1 - d_4^2 - a_3^2 - a_2^2)}{2a_2}]^2}} + k\pi, (k=0, -1)$$

(5-6)

(3)求 θ_2

$$\text{令 } u = (c_1a_x + s_1a_y) - (c_1p_x + s_1p_y) + a_1, \quad v = a_zd_6 - p_z,$$

$$\text{式 } c_2[-(c_1a_x + s_1a_y) + (c_1p_x + s_1p_y)] - s_2(a_zd_6 - p_z) - c_2a_1 = s_3d_4 + a_3c_3 + a_2$$

可化简为 $c_2u - s_2v = t_2 + a_2,$

$$\theta_2 = \arctg \frac{u}{v} \pm \arctg \frac{t_2 + a_2}{\sqrt{u^2 + v^2 - (t_2 + a_2)^2}} + k\pi, (k=0, -1)$$

(5-7)

(4)求 θ_6

令式(5-3)两边元素(1,2),(2,2)分别相等, 可得

$$(c_1c_2n_x + s_1c_2n_y + s_2n_z)s_6 + (c_1c_2o_x + s_1c_2o_y + s_2o_z)c_6 = -c_3s_4$$

$$(-c_1s_2n_x - s_1s_2n_y + c_2n_z)s_6 + (-c_1s_2o_x - s_1s_2o_y + c_2o_z)c_6 = -s_3s_4$$

化简整理得:

$$\frac{s_6}{c_6} = \frac{(-c_1s_2c_3 - c_1c_2s_3)o_x + (-s_1s_2c_3 - s_1c_2s_3)o_y + (c_2c_3 - s_2s_3)o_z}{(c_1s_2c_3 + c_1c_2s_3)n_x + (s_1s_2c_3 + s_1c_2s_3)n_y + (-c_2c_3 + s_2s_3)n_z}$$

可求得:

$$\theta_6 = \arctg \frac{(-c_1 s_2 c_3 - c_1 c_2 s_3) o_x + (-s_1 s_2 c_3 - s_1 c_2 s_3) o_y + (c_2 c_3 - s_2 s_3) o_z}{(c_1 s_2 c_3 + c_1 c_2 s_3) n_x + (s_1 s_2 c_3 + s_1 c_2 s_3) n_y + (-c_2 c_3 + s_2 s_3) n_z} + k\pi, (k=0, \pm 1) \quad (5-8)$$

(5)求 θ_4

令式(5-3)左右两边元素(3,2)相等, 可得

$$(s_1 n_x - c_1 n_y) s_6 + (s_1 o_x - c_1 o_y) c_6 = c_4$$

可求得:

$$\theta_4 = \arctg \left(\frac{\pm \sqrt{1 - [(s_1 n_x - c_1 n_y) s_6 + (s_1 o_x - c_1 o_y) c_6]^2}}{(s_1 n_x - c_1 n_y) s_6 + (s_1 o_x - c_1 o_y) c_6} \right) + k\pi, (k=0, \pm 1) \quad (5-9)$$

(6)求 θ_5

令式(5-3)左右两边元素(3,1),(3,3)分别相等得:

$$(s_1 n_x - c_1 n_y) c_6 - (s_1 o_x - c_1 o_y) s_6 = s_4 c_5$$

$$s_1 a_x - c_1 a_y = s_4 s_5$$

将上面两式相除, 可以得到:

$$\frac{s_5}{c_5} = \frac{s_1 a_x - c_1 a_y}{(s_1 n_x - c_1 n_y) c_6 - (s_1 o_x - c_1 o_y) s_6}$$

$$\text{则: } \theta_5 = \arctg \frac{s_1 a_x - c_1 a_y}{(s_1 n_x - c_1 n_y) c_6 - (s_1 o_x - c_1 o_y) s_6} + k\pi, k = (0, \pm 1) \quad (5-10)$$

这样就求出了 $\theta_1 \sim \theta_6$, 机器人的运动学逆解得以解决。

5.2.3 基于 Matlab 的机器人运动学仿真

在开发或分析一个机器人系统时, 运动学仿真提供十分有效的途径, 可以分析机器人的运动学或检查逆问题求解是否正确。它不同于动力学或控制仿真, 由于动力学或控制仿真是忽略了许多次要因素并通过线性化而得到的近似模型, 因此仿真情况与现实情况可能有较大差距;但运动学仿真结果完全是真实可靠的, 这由运动学正、逆问题公式的准确与可靠保证的。

运动学仿真的目的在于:

(1)求出运动空间,以评估是否满足工作需要。运动学正问题,是已知关节角,求解手端的位置和姿态,因此在已知各关节角运动范围条件下,使用运动学正问题就能很好地解决运动空间分析问题。

(2)完善和验证运动学逆解公式。逆向运动学求解过程,会遇到多解问题。甚至某个关键的关节角正负会对应不同求解公式,这一切给逆问题求解带来许多困难和不定因素。但通过仿真手段,会发现问题,改正问题,最后使逆解公式完全可靠。

(3)为机器人运动控制提供必要条件。绝大多数工业机器人运动控制是建立在一套正确的运动学正逆解问题求解公式上,也就基于运动学控制,当然没有一套可靠的运动学公式,运动控制也无法进行。

通过运动学分析,获得给定机器人的一组运动学正解公式和逆解公式。然后,编写一个程序,把正解和逆解联系起来,并且在整个运动空间搜索,如果正逆解问题的结果每一步都是一致的,说明运动学分析时获得的正逆解是正确的,一旦出现不一致,就说明这一组正逆解公式有问题,而且大多数出现在逆解公式上。

MATLAB 是由美国 Mathworks 公司于 1984 年推出的一套高性能的数值计算和可视化软件,具有良好的用户界面、通用性好、功能强大,是工程人员进行科研工作的有效工具。利用 MATLAB 语言对机器人操作机进行运动学分析,得到不同关节角度下的机器人末端执行器的位姿。由于 MATLAB 编程简洁,可充分有效地完成简单的仿真过程,加深对机器人运动学理论的研究。计算机仿真结果表明,只要机器人各关节角在给定范围内选择,得出的结果完全符合实际情况。

5.3 MATLAB 中 Simulink 虚拟现实仿真

5.3.1 MATLAB 介绍

MATLAB 具有非常强大的计算功能,凭借其杰出的性能, MATLAB 现在已成为世界上应用最广泛的工程计算应用软件。

MATLAB 的功能强大,主要有:数值计算功能、符号计算功能、数据分析和可视化功能、文字处理功能、Simulink 动态仿真功能。使用 MATLAB 语言设计的程序,其编译和执行速度都远远超过传统的 C 和 Fortran 语言设计的程序,可以说, MATLAB 在工程计算方面的编程效率远远高于其他编程语言。

MATLAB 系统包括五个部分: MATLAB 语言、MATLAB 工作环境、MATLAB 图形处理系统、MATLAB 数学函数库和 MATLAB 应用程序接口。

5.3.2 Simulink 仿真

Simulink 是 MathWorks 公司开发的一个有重要影响力的软件产品, Simulink 不能独立运行, 只能在 MATLAB 环境中运行。Simulink 是一个用来对动态系统进行建模、仿真和分析的软件包, 它支持连续、离散或两者混合的线性 and 非线性系统, 也支持具有多种采样速率的多速率系统。Simulink 为用户提供了用方框图进行建模的图形接口, 采用这种结构化模型就像用笔和纸来画一样容易。与传统的仿真软件包用微分方程和差分方程建模相比, 它具有更直观、方便、灵活的优点。

在模型创建以后, 用户可以通过 Simulink 的菜单命令或 MATLAB 的命令窗口输入命令来对它进行仿真。在进行仿真的同时还可以观察仿真的结果。除此之外, 用户还可以通过改变参数来迅速观察系统中发生的变化。仿真的结构可以同时存放在 MATLAB 的工作空间内, 供以后计算、分析用。MATLAB 自身所带的所有应用工具箱, 同样适用于 Simulink 环境中, 这就大大扩展了 Simulink 的适用范围, 也使 Simulink 进入到许多领域, 成为科学计算和动态系统仿真有力工具。

Simulink 是由模块库、模型构造及指令分析、演示程序等几部分组成的。它已经发展成为一个系列化的产品, 它可以对具有时间触发和离散事件系统进行分析 and 仿真, 可以将 Simulink 中的仿真框图转换成 C 语言等其它高级语言, 并可直接运行于相应的硬件上, 还可以完成对诸如通信系统、非线性控制系统、电力系统等建模、分析和仿真。因此在许多辅助工具的帮助下, Simulink 不再是单一的动态系统仿真软件, 它的作用和应用已经得到了进一步的扩展。

Simulink 的工作环境主要是由库浏览器与模型窗口组成的, 前者为用户提供了展示 Simulink 标准模块库和专业工具箱的界面, 而后者是用户创建模型方框图的主要地方。Simulink 库是 Simulink 仿真环境的基本组成部分, 对于一般的系统建模仿真, Simulink 库就完全可以胜任了, 其中, Simulink 库又可以分为标准库与扩展库。Simulink 标准库是学习 Simulink 建模与仿真的基础, 可以从标准库开始由浅入深地学习和了解 Simulink 中丰富的模块资源^[43]。

Simulink 模型是以 ASCII 码形式存储的.mdl 文件(称为 MDL 模型文件), 利用保存图标

或是菜单的 File/Save 都可以实现。

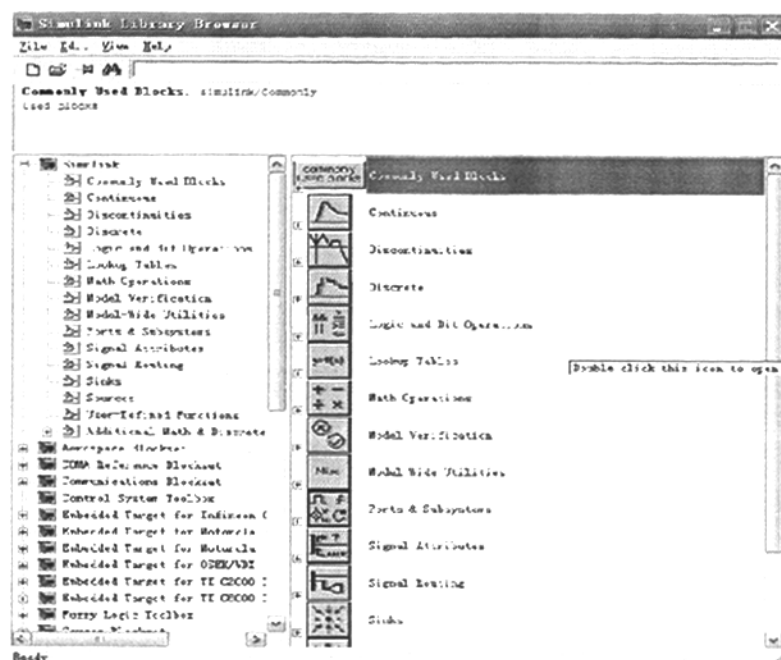


图 5.7 Simulink 模块库

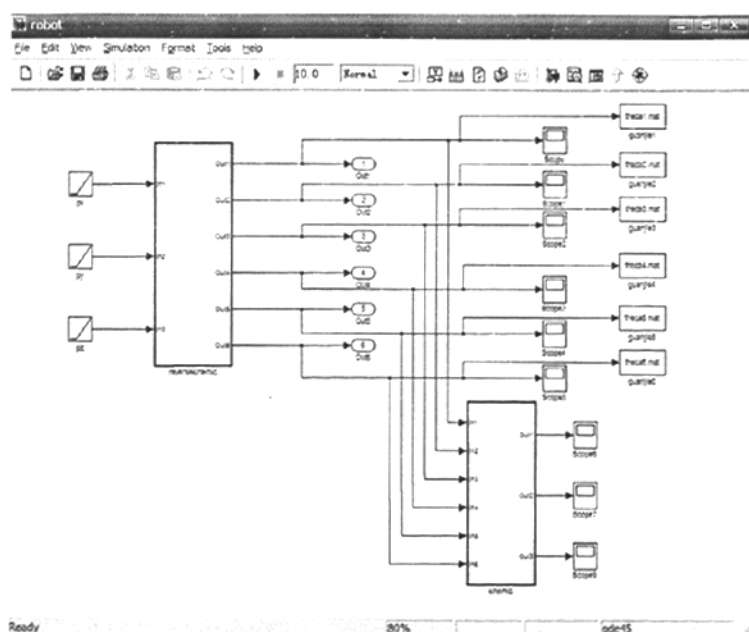


图 5.8 动态控制模型

5.3.3 虚拟现实工具箱

虚拟现实工具箱为三维虚拟现实环境中进行可视化操作和与动态系统进行交互提供了一种有效的解决方案, 这些动态系统用 MATLAB 和 Simulink 来描述, 虚拟现实工具箱拓展了 MATLAB 和 Simulink 处理虚拟现实图像的能力。使用标准的 VRML 技术, 通过 MATLAB 和 Simulink 环境生成三维场景。通过 Simulink 的接口在一个虚拟真实的三维模型中观察动态系统的模拟, 虚拟现实工具箱的大多数特征可以使用 Simulink 模块来实现, 一旦在 Simulink 对话框中包含了这些模块, 我们就可以选择与 Simulink 信号链接的虚拟世界。VRML 工具箱中包括以下几个模块^[44]:

(1)VR Sink 模块: 将信号写入虚拟世界。VR Sink 模块从它的端口写数值到模块参数指定的虚拟世界域中, 它支持的信号数据格式是 double 类型;

(2)VR Source 模块: 从虚拟世界中读取数据。VR Source 模块监视在模块参数中指定的虚拟世界域的数值, 并且监视它们输出的值。它所支持的数据类型是 double 类型的 VR Source 模块的输出类型。

(3)Joystick 输入设备: 异步输入操作杆。此工具为 Simulink 模型与虚拟世界之间提供了一种非常方便的交互手段。操作杆的输入端使用标准的 Microsoft Windows 操作杆接口, 可以被其它的 Simulink source 使用。它的输出端口反映了操作杆控制的状态。

(4)Magellan 空间鼠标输入设备: 从 Magellan 空间鼠标处理输入。Magellan 空间鼠标在功能上是一种类似于操作杆的设备, 但它提供六自由度的运动控制。它允许读取空间鼠标的状态和提供某些普通的输入转化。它支持双精度类型的信号。

(5)VR Placeholder 模块: 发送一个特殊的不确定的数值, 产生能覆盖 VRML 域中信号矢量不用的或不受影响的部分。VR Placeholder 送出一个特定的值, 此值被 VR Sinkblock 解释成“未指明的”。当此值出现在 VR Sink 模块时, 不论作为一个简单值还是作为矢量的元素, 虚拟世界的其它值仍然保持不变。

(6)VR Signal Expander: 在输出端口使用 VR 占位符信号填充空位置, 将输入矢量放大成一个完全的 VRML 域信号。VR 放大器生成一个预先定义好长度的矢量, 从输入端口使用某些数值填充剩余的 VR 未指明的值。它接收的是 double 数据信号。

所有 VRML 节点分别列在等级树样式的观察窗口中, 我们可以选择控制的自由度, 当关闭接口对话框后, 虚拟现实工具箱模块自动更新在虚拟世界中与选择节点有关的输入和输出。当连接这些输入到一定的 Simulink 信号上时, 就可以在浏览器中观察到可视化的模拟了, 另外, 虚拟现实工具箱还提供了一个灵活的与虚拟现实世界相连的 MATLAB 接口。

虚拟现实工具箱中的对象分为 vrworld 对象、vrnode 对象和 vrfigure 对象三种。vrworld

对象是虚拟场景的句柄。表 5.2 为 vrworld 对象的方法。通过这些方法，可以与场景交互并控制场景。vnode 对象是 VRML 节点的句柄。表 5.3 为 vnode 对象的方法。通过这些方法，可以获取和设置节点属性值。vnode 对象是 vrworld 对象的子对象。vrfigure 对象是虚拟现实工具箱查看器的句柄，通过它，可以获取和设置查看器的属性。Vrfigure 对象是 vrworld 对象的子对象。

表 5.2 vrworld 对象的方法

方 法	描 述
Vrworld	创建一个虚拟世界相关的 vrworld 新对象
Vrworld/close	关闭虚拟世界
Vrworld/delete	从内存中删除虚拟世界
Vrworld/open	打开虚拟世界
Vrworld/save	从虚拟世界写到 VRML 文件
Vrworld/view	查看虚拟世界
Vrworld/get	读取 vrworld 对象的属性值
Vrworld/set	改变 vrworld 对象的属性值
Vrworld/reload	从相关 VRML 文件中重新载入虚拟世界

表 5.3 vnode 对象的方法

方 法	描 述
vnode	给已存在的节点创建一个新的节点或句柄
vnode/delete	删除一个 vnode 对象
vnode/fields	返回节点对象 VRML 字段的综述
vnode/get	读取 vnode 对象或 VRML 字段的属性值
vnode/set	改变虚拟世界节点的属性或 VRML 字段
vnode/getfield	获取一个 vnode 对象的字段值
vnode/setfield	改变 vnode 对象的字段值

5.3.4 虚拟现实模型与 Simulink 的接口

虚拟现实模型与 Simulink 的接口，即通过虚拟现实工具箱导入虚拟场景，再用 Simulink 模块驱动场景。主要步骤如下^[45,46,47]：

- (1) 打开 Simulink 模块，建立一个 Simulink 程序；
- (2) 打开虚拟现实工具箱库，从库的窗口中拖拽 VR Sink 模块和 VR Signal Expander 模块到 Simulink 程序窗口中，虚拟现实仿真模型中有几个欲控制的节点，就要拖出几个 VR Signal Expander 模块；
- (3) 双击 VR Sink 模块，出现一个对话框。单击 browse 按钮浏览选定已经编辑好的 VRML 文件。选定以后，一个 VRML 文件的结构树出现在右边“VRML tree”的窗口内，逐

次展开结构树后可以看到机器人的总体层次结构。结构树中在每一个 Transform 节点下都有域 translation、rotation 和 center 等选择框；

(4)选择要控制的自由度。本课题中要选择 S 轴、L 轴、U 轴、R 轴、B 轴和 T 轴这六个自由度进行转动控制，因此勾选这六个 Transform 节点下的 rotation 选项。当关闭接口对话框后，VR Sink 模块中就会出现与所选标记点相关输入端口。在这些端口中输入一定的 Simulink 信号就可以实现虚拟模型的可视化操作了。输入信号的形式任意，连续、离散的 Simulink 信号都可，这一点上充分体现出了虚拟现实工具箱库与 Simulink 模块的良好兼容性。Simulink 程序运行后，双击 VR Sink 模块，就可以在一个浏览器中观察到可视化的虚拟仿真了。

5.4 服务器与机器人通讯的实现

5.4.1 TCP 协议和 UDP 协议

TCP(传输控制协议)和 UDP(用户数据报协议)是 TCP/IP 协议族中运输层上两个互不相同的传输协议。TCP 为两台主机提供高可靠性的数据通信。它所做的工作包括把应用程序交给它的数据分成合适的小块交给下面的网络层，确认接收到的分组，设置发送最后确认分组的超时时钟等。由于运输层提供了高可靠性的端到端的通信，因此应用层可以忽略所有这些细节。而另一方面，UDP 则为应用层提供一种非常简单的服务。它只是把称作数据报的分组从一台主机发送到另一台主机，但并不保证数据报能到达另一端。任何必需的可靠性必须由应用层来提供。这两种根据它们的特点分别在不同的应用程序中有不同的用途。

在 TCP/IP 协议族中，网络层 IP 提供的是一种不可靠的服务，它只是尽可能快地把分组从源结点送到目的结点，但是并不提供任何可靠性保证。TCP 在不可靠的 IP 层上提供了一个可靠的运输层。为了提供这种可靠的服务，TCP 采用了超时重传、发送和接收端到端的确认分组等机制。TCP 和 UDP 是两种最为著名的运输层协议，二者都使用 IP 作为网络层协议。虽然 TCP 使用不可靠的 IP 服务，但它却提供种可靠的运输层服务。UDP 为应用程序发送和接收数据报。一个数据报是指从发送方传输到接收方的一个信息单元(例如，发送方指定的一定字节数的信息)。但是与 TCP 不同的是，UDP 是不可靠的，它不能保证数据报能安全无误地到达最终目的。

由此可见, TCP 协议是面向连接的协议, 它比较安全、稳定, 而此网络控制平台系统在监督模式对数据的准确性要求较高, 因此采用了 TCP 为数据传送的协议。UDP 协议系统开销小、速度快、效率高、占用资源少, 本系统中的机器人状态信息和即时消息的发送采用的是 UDP 协议。

在 Windows 下的各种网络编程接口中, Windows Sockets 脱颖而出, 越来越得到大家的重视, 这是因为 Windows Sockets 规范是一套开放的、支持多种协议的, 已成为 Windows 网络编程的事实上的标准。

通讯的基石是套接字, 一个套接字是通讯的一端。在这一端上你可以找到与其对应的一个名字。一个正在被使用的套接字都有它的类型和与其相关的进程。套接字存在于通讯域中。通讯域是为了处理一般的线程通过套接字通讯而引进的一种抽象概念。Windows Sockets 规范支持单一的通讯域, 即 Internet 域。各种进程使用这个域互相之间用 Internet 协议族来进行通讯。

用户目前可以使用两种套接字, 即流套接字和数据报套接字。流套接字提供了双向的, 有序的, 无重复并且无记录边界的数据流服务(基于 TCP 协议)。数据报套接字(基于 UDP 协议)支持双向的数据流, 但并不保证是可靠, 有序, 无重复的。也就是说, 一个从数据报套接口接收信息的进程有可能发现信息重复了, 或者和发出时的顺序不同。数据报套接字的一个重要特点是它保留了记录边界。对于这一特点, 数据报套接字采用了与现在许多包交换网络(例如以太网)非常类似的模型。

套接字有三种类型: 流式套接字, 数据报套接字及原始套接字。

(1)流式套接字

流式套接字定义了一种可靠的面向连接的服务, 实现了无差错无重复的顺序数据传输。面向连接服务器处理的请求往往比较复杂, 不是一来一去的请求应答所能解决的, 而且往往是并发服务器。使用面向连接的套接字编程, 其时序可以通过下图 5.9 表示。

套接字工作过程如下: 服务器首先启动, 通过调用 `Socket()` 建立一个套接字, 然后调用 `bind()` 将该套接字和本地网络地址联系在一起, 再调用 `listen()` 使套接字做好侦听的准备, 并规定它的请求队列的长度, 之后就调用 `accept()` 来接收连接。客户在建立套接字后就可调用 `connect()` 和服务器建立连接。连接一旦建立, 客户机和服务器之间就可以通过调用 `read()` 和 `write()` 来发送和接收数据。最后, 待数据传送结束后, 双方调用 `close()` 关闭套接字。

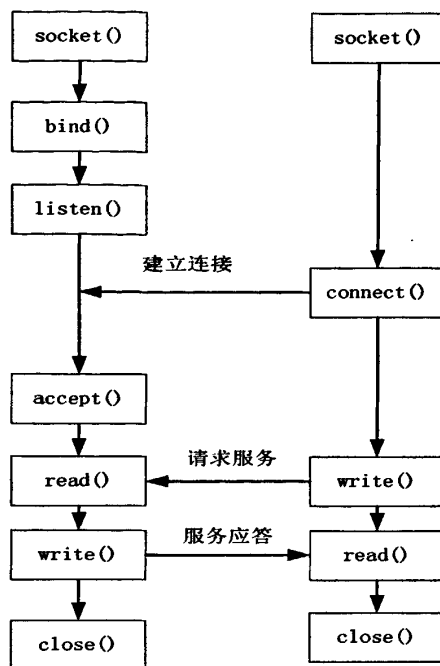


图 5.9 流式套接口应用程序

(2) 数据报套接字

数据报套接字定义了一种无连接的服务，数据通过相互独立的报文进行传输，是无序的，并且不保证可靠，无差错。无连接服务器一般都是面向事务处理的，一个请求一个应答就完成了客户程序与服务程序之间的相互作用。若使用无连接的套接字编程，程序的流程可以用图 5.10 表示。

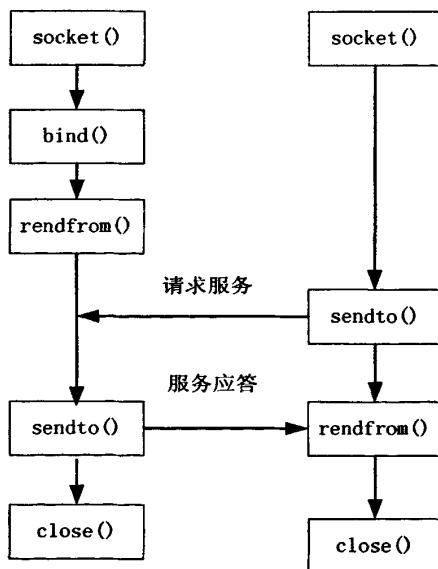


图 5.10 数据报套接口应用程序

(3) 原始套接字

原始套接字允许对低层协议如 IP 或 ICMP 良接访问，主要用于新的网络协议实现的测试等。Windows Sockets 规范并没有规定 Windows Socket 必须支持原始套接字。然而 Windows Sockets 规范鼓励 Windows Sockets DLL 提供原始套接字支持。

5.4.2 机器人基类的实现

机器人行为抽象可以简单的分为：初始化机器人、得到机器人状态信息、定点行走、执行任务、停止任务、暂停任务、继续执行和删除任务等等。如下所示：

```
class BaseRobot
{
... .. //一些数据定义
virtual short BuildConnect( ); //建立和机器人的连接
virtual short disConnect( ); //断开和机器人的连接
virtual void Init (HWND hwd) ; //机器人初始化
//获得机器人的状态信息
virtual void GetRobotStatus (STATUSINFORMATION &status) ;
virtual bool StartStep (POSTION &pos) ; //定点行走
virtual bool StartJob (CString tJobName) ; //开始一个任务。
virtual void ContinueRun( ); //任务暂时停止运行后，重新运行任务
virtual void StopRun( ); //暂时运行任务
virtual void JobDelete (CString jobname) ; //任务删除。
virtual bool JobDownLoad (CString jobName) ; //把任务下载到机器人上
Robot( );
virtual ~Robot( );
... .. //一些数据定义
};
```

下面是一个 Motoman 机器人的类模型，它包括了机器人共有的行为方法，和自己特有的行为。

```
class MotomanRobot : public BaseRobot
{
```

```

... .. //一些数据定义

public:

    virtual void GetRobotStatus ( STATUSINFORMATION &status );

    virtual bool StartStep ( POSTION &pos );

    virtual void JobDelete ( CString jobname );

    virtual void Init ( HWND hwd );

    virtual void ContinueRun( );

    virtual void StopRun( );

    virtual bool StartJob (CString jobName );

    Robot( );

    virtual ~Robot( );

private:

    bool ServoOn( );

    virtual bool JobDownLoad( CString jobName );

    short disconnect( ); //中断和机器人连接

    HWND inputHwnd; //inputHwnd 是使用此 robot 类的窗口句柄

    short nCid; //和控制柜连接的通道值

    virtual short BuildConnect( ); //建立和机器人的连接

    ... .. //一些数据定义

};

```

5.4.3 网络通讯模块

本系统传输的信息主要是字符串、文本类信息。机器人的控制指令和位置反馈信息都属此类。这类信息要求数据传输正确、可靠。根据上文中介绍，TCP(Transmission Control Protocol, 传输控制协议)是一个具有重传机制和拥塞控制机制的面向连接的协议，具有高度可靠性，因此系统采用它作为字符类信息的传输协议。

字符流的通信选用 Socket 机制来实现。由于 Socket 是将用户代码和协议底层实现隔离开的编程对象，并且提供了连接任何流到通信接口的方法，这样就可以通过输入输出流来抽象网络层的具体操作，把应用程序与网络技术细节分开，只需考虑流的读写操作而不用

关心网络的具体细节。

本系统采用基于 TCP 协议之上的网络机器人数据控制协议,协议的数据结构包括机器人状态数据(Robot Status Data)和机器人点位数据(Robot Position Data)。

```
struct STATUSINFORMATION
{
    CString    curJobName;
    short      curJobLine;
    short      curJobStep;
    WORD       d1;
    WORD       d2;
    short      error;
    double     xp[6];
    double     jp[6];
    WORD       rconf;
};
```

表示机器人系统当前的一些位置信息和任务信息,用来反馈给客户端。

```
struct POSTION
{
    bool       controlMode;
    CString    movtype;
    CString    vtype;
    double     spd;
    CString    framename;
    short      rconf;
    short      toolno;
    double     p[12];
};
```

表示的是机器人接受的位置控制信息和机器人运动状态信息。

5.4.4 与机器人接口类库

MotomanRobot 类是本系统与具体的机器人的接口类，在上文中已经提到，它是个独立的类，负责发送具体的控制信息给机器人，并且返回机器人的状态。它包括机器人初始化、定点移动、下载任务、执行任务等 Motoman 机器人所具有的一些行为。

MotomanRobot 类的具体实现是建立在 MOTOCOM32 通讯软件提供了一套动态连接的库函数，在类的设计中调用这些库函数，对机器人进行控制。这些库函数包括了机器人的主要操作功能，例如：状态读取函数用于读取机器人的位置值、当前程序名、运行方式、操作坐标、控制轴组等状态信息以及文件的上传、下载等；系统控制函数可对机器人进行各种操作，包括伺服电源的通断、程序运行的起停、机器人以各种坐标方式按不同的速度进行目标点或者增量运动。通过这些库函数的组合使用，实现对机器人系统的各种控制与操作，扩展了机器人的系统集成功能。

(1) 动态连接库的调用

为了使 MotomanRobot 类能够调用动态连接库中的库函数，首先，机器人类的实现文件中必须包含：“direct.h”、“motocom.h”头文件；其次，复制 Motocom32.DLL、Motolk.DLL、Motolkr.DLL 数据传输动态库到工程文件目录下；最后，把“motocom32.lib”文件加入的类库管理目录中。

(2) 通信的建立

首先利用语句：`nCid = BscOpen(jobname, 1)` 取得通讯句柄 `nCid`，在后续的语句中调用 `BscSetCom(nCid, 1, 9600, 2, 8, 0)` 设置通讯口参数，最后进行通讯线路的连接：`BscConnect(nCid)`。这样就可以进行文件的上传(函数 `BscUpLoad(file-name, nCid)`)、下载(函数 `BscDownLoad(nCid, file-name)`)等操作了。

(3) 通讯线路的关闭

调用关闭通讯口的函数 `BscClose(nCid)`，实现和机器人通讯的安全结束。

5.5 小结

本章详细的介绍了本系统实现的方法和原理。介绍了系统采用的 Motoman-Hp3 机器人，机器人运动学及 Matlab 中 Simulink 虚拟仿真。讨论了数据传输机制，详细的介绍了系统各个功能模块和编程模块的实现。提出了一个基于 TCP 协议的机器人数据控制协议。

第六章 基于虚拟现实技术的遥操作机器人协调控制系统的实现

6.1 实验系统组成

在前面的章节中分别对机器人控制模式、网络数据传输协议和基于虚拟现实技术的遥操作机器人系统平台的具体实现作了较为详细的研究和分析,提出了基于客户端/服务器端的分布式的机器人控制系统,主要包括了机器人监督控制方式, MATLAB 中的 Simulink 仿真,并定义了一个基于 TCP 协议为底层协议的机器人数据传输协议。本章中将对相应的研究给出实验验证。



图 6.1 基于虚拟现实技术的遥操作机器人系统平台硬件图

本系统所用硬件为 MOTOMAN-HP3 机器人、机器人控制柜 NX100、应用程序服务器、客户端 PC。

6.2 MATLAB 中虚拟现实模型的实现

虚拟现实仿真中,最重要也是最复杂的部分是三维仿真模型的建立,这一过程直接关系到虚拟现实仿真的“逼真性”和“交互性”的效果。在第四章中,介绍了如何利用 Solidworks-VRML 实现机器人的虚拟模型的建立,虚拟模型建立完成后存为.wrl 文件。为了实现在 MATLAB 中的虚拟现实仿真,需将.wrl 文件导入 MATLAB 中的 V-Realm Builder2.0 形成可视窗口。如图 6.2。

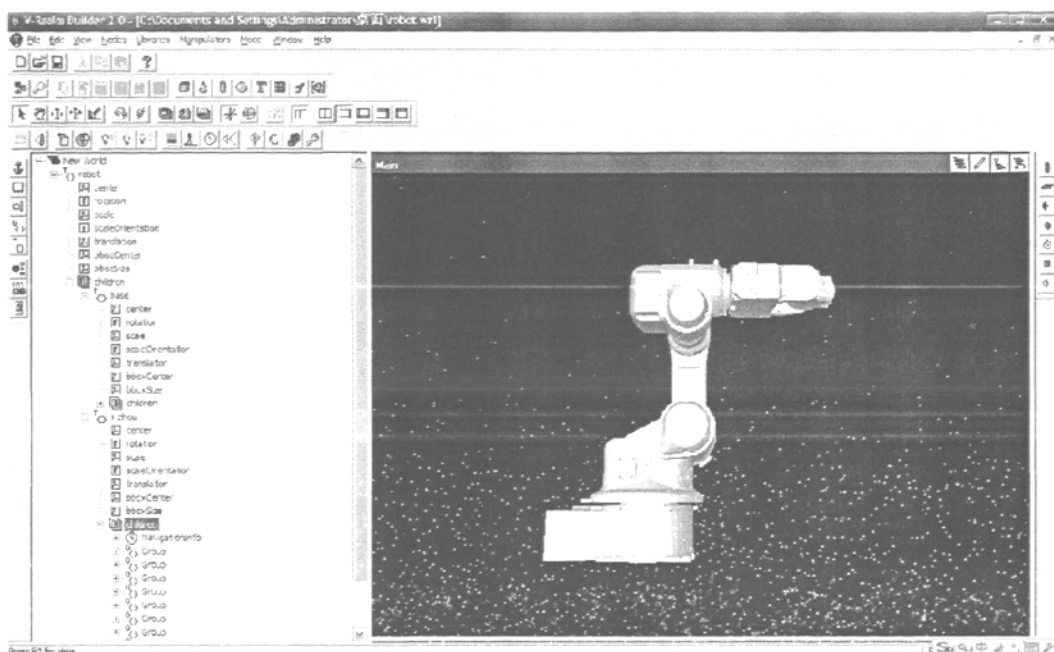


图 6.2 V-RealmBuilder2.0 中的虚拟模型

在第四章中,我们对每一个零件进行转化操作,生成独立的 VRML 文件并组装了机器人的虚拟模型,但是无法满足虚拟现实仿真的要求,仍然需要修改。在 V-Realm Builder2.0 中,我们将对机器人的虚拟模型进一步改进。

首先介绍一下变换节点 Transform 的性质。Transform 节点是一个组节点,其下包含一个子节点的列表。这些子节点可以是 Shape 节点、其它 Group 节点或 Transform 节点。与组节点 Group 不同,在 Transform 节点中内置了一个坐标系,Transform 的所有子节点都将在它的坐标系原点处创建。通过域 Translation 和 rotation 可以控制 Transform 坐标系的平移和旋转,而如果这个坐标系的方向、位置发生变化,则 Transform 节点内的所有节点将同一变化。Transform 节点可以层层嵌套,因此通过 Transform 节点可以把运动传递下去。

本系统中,机器人的虚拟模型结构如图 6.3 所示。

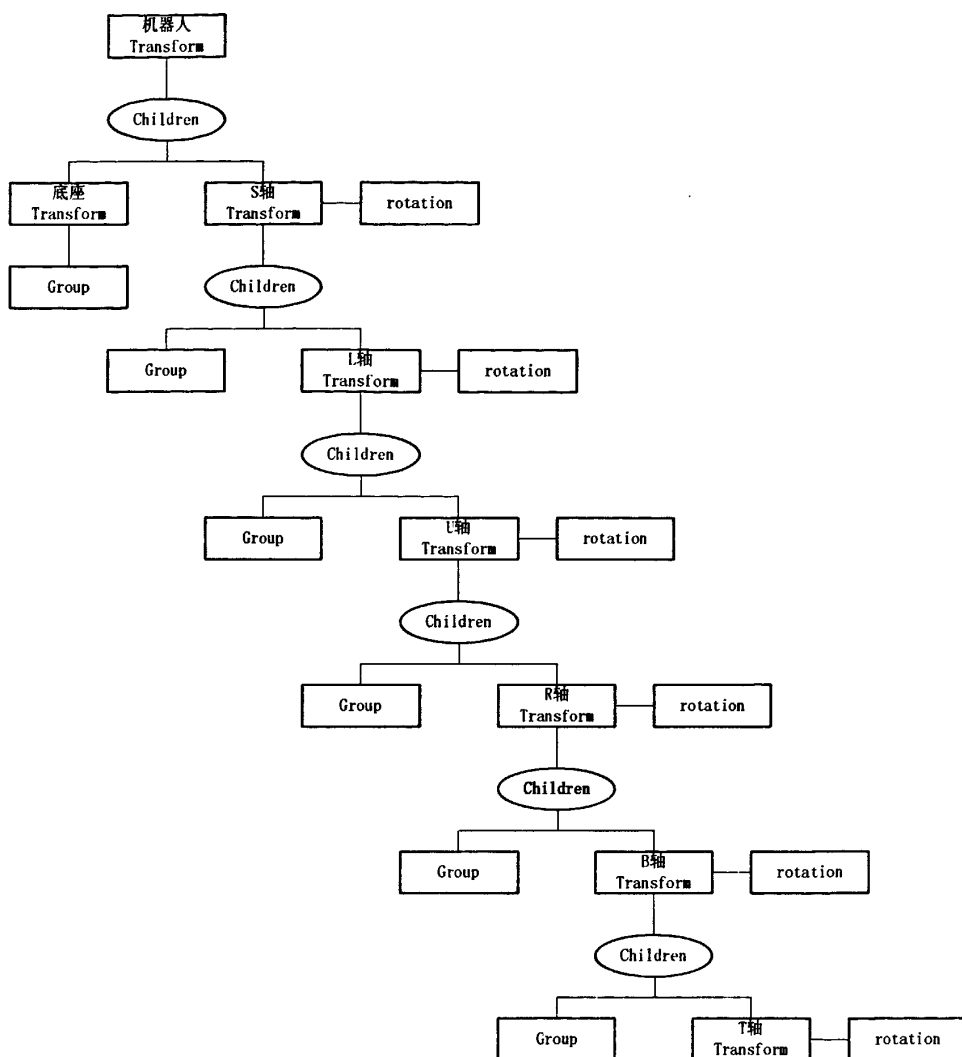


图 6.3 机器人的结构

V-Realm Builder2.0 的优点是能够非常直观的看到虚拟模型的状态，左侧的“VRML tree”完整的显示了虚拟模型的各个参数，方便用户修改并及时看到修改后的模型，为之后的虚拟仿真打下了良好的基础。

6.3 基于 MATLAB/Simulink 运动学仿真的实现

6.3.1 运动学仿真系统的实现

虚拟现实工具箱的大多数特征可以实用 Simulink 模块来实现，本节将主要介绍利用

Simulink 模块来实现机器人仿真模型的运动特征。

基于 MATLAB/Simulink 的机器人运动学仿真模型如下图 6.4 所示, 根据 5-2 章介绍的机器人运动学编制了运动学正解 kinemic.mdl 和逆解 reversekinemic.mdl 仿真模块, 图 6.5 中机器人末端位置指令坐标(px, py, pz), 由轨迹规划给出, 经过逆运动学模块如图 6.7 得到 S、L、U、R、B、T 轴的转角信号 θ_1 、 θ_2 、 θ_3 、 θ_4 、 θ_5 、 θ_6 , 这些转角信号输出为.mat 文件, 在 MATLAB WORKSPACE 中打开并输出为记事本格式的点坐标, 为服务器端读取数据驱动机器人运转做准备, 再由 kinemic.mdl 如图 6.6 调用运动学正解进行运动学运算, 得到机器人末端轨迹, 跟输入的末端轨迹作比较来校验运动学和逆运动学的正确性。

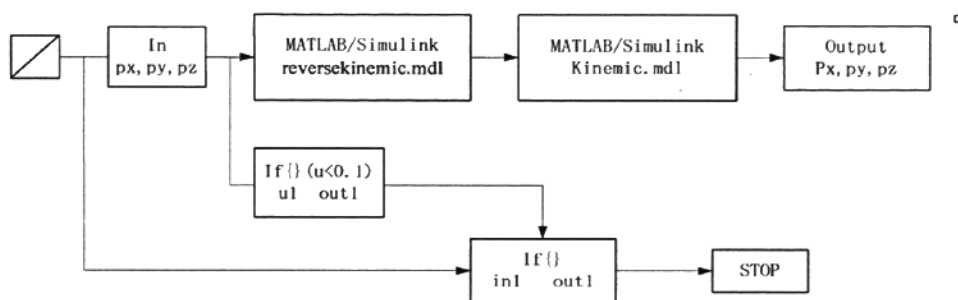


图 6.4 机器人运动学仿真结构图

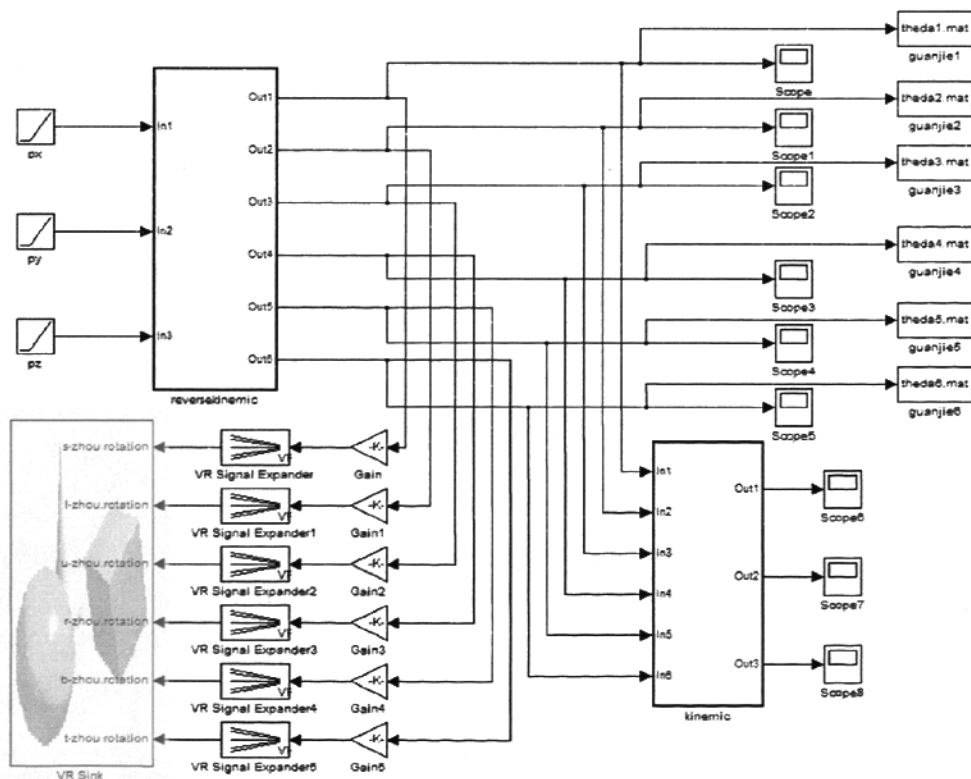


图 6.5 机器人运动学仿真结构图

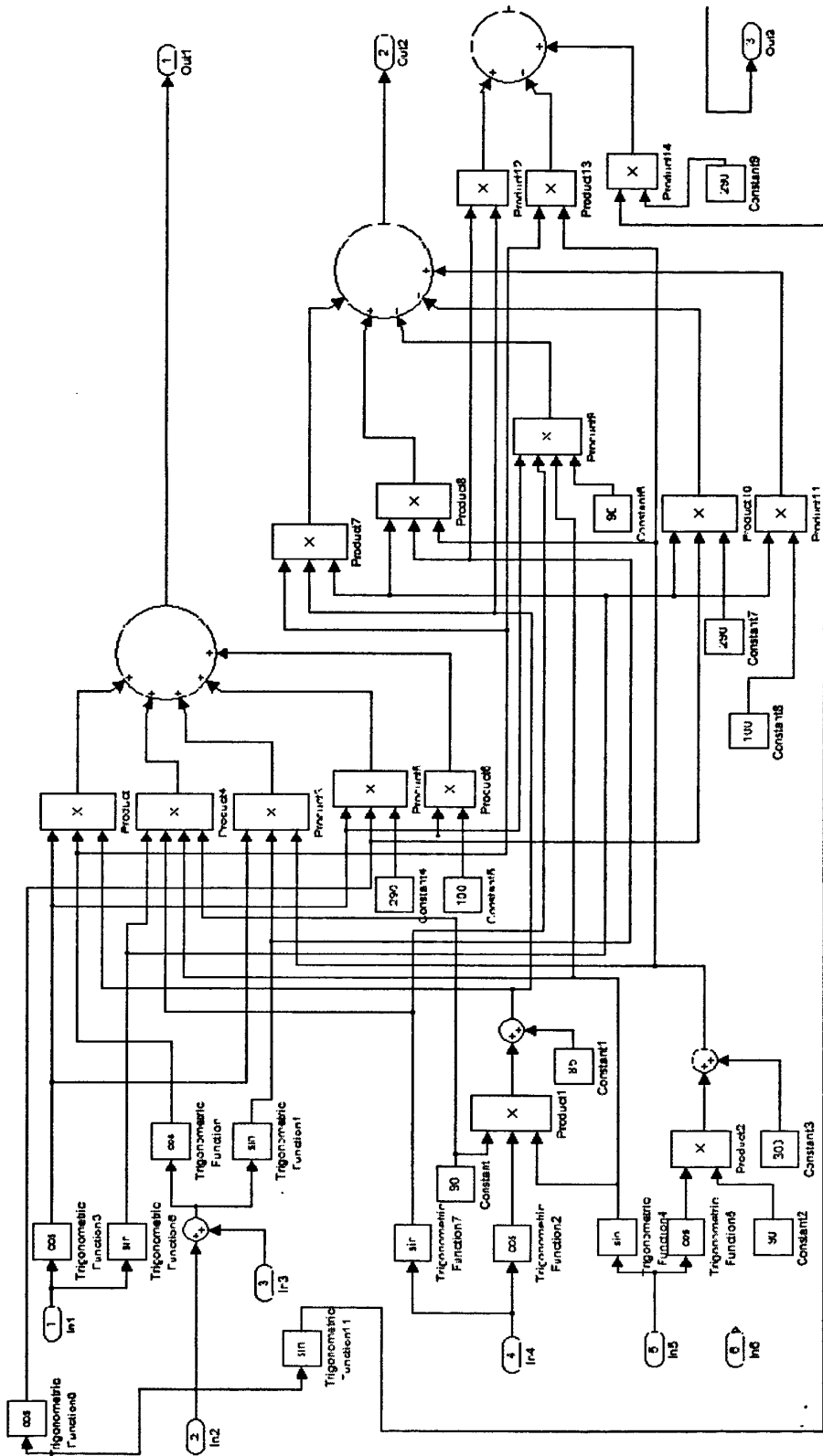


图 6.6 运动学正解 kinemic.mdl 模块

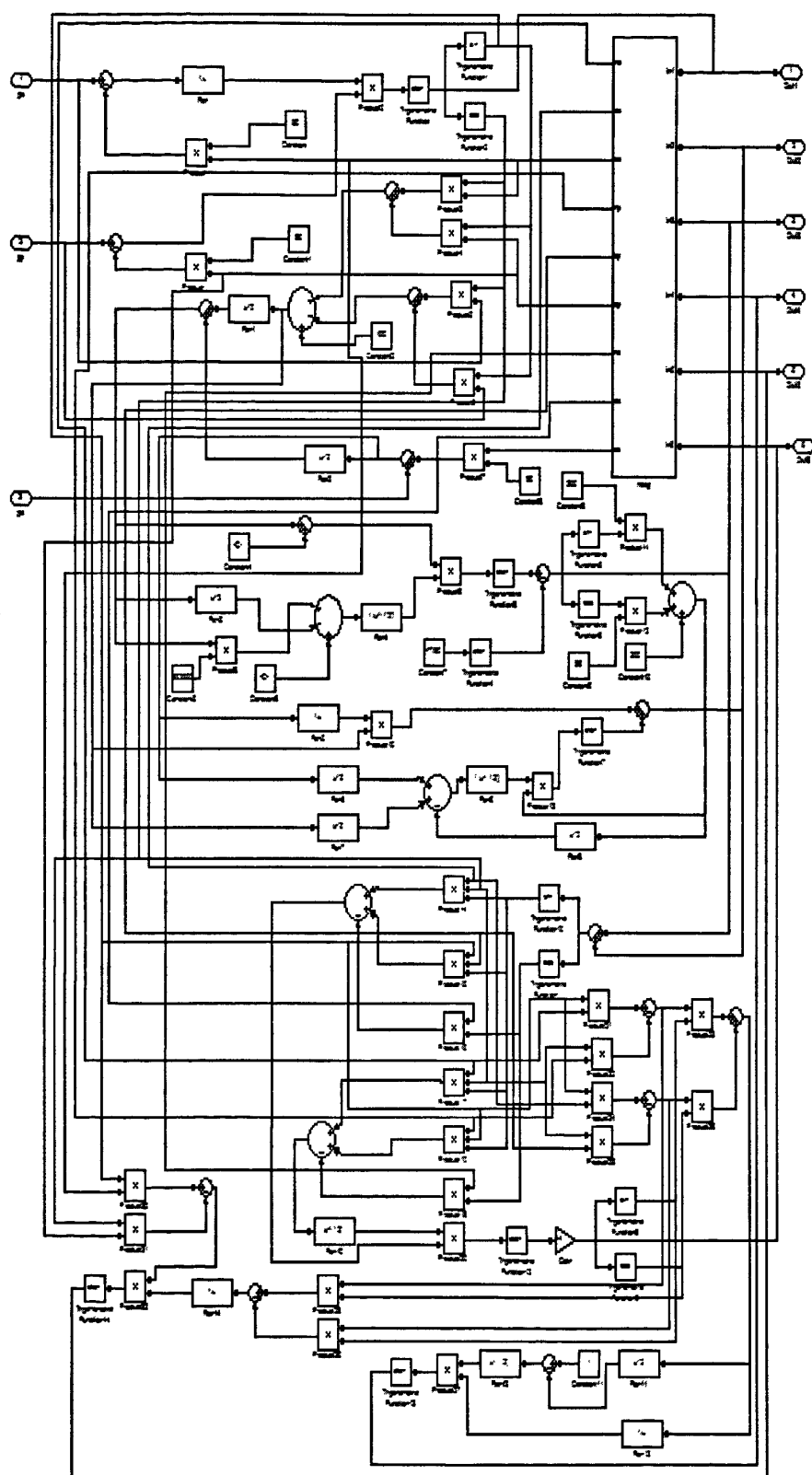


图 6.7 运动学逆解 reversekinemic.mdl 模块

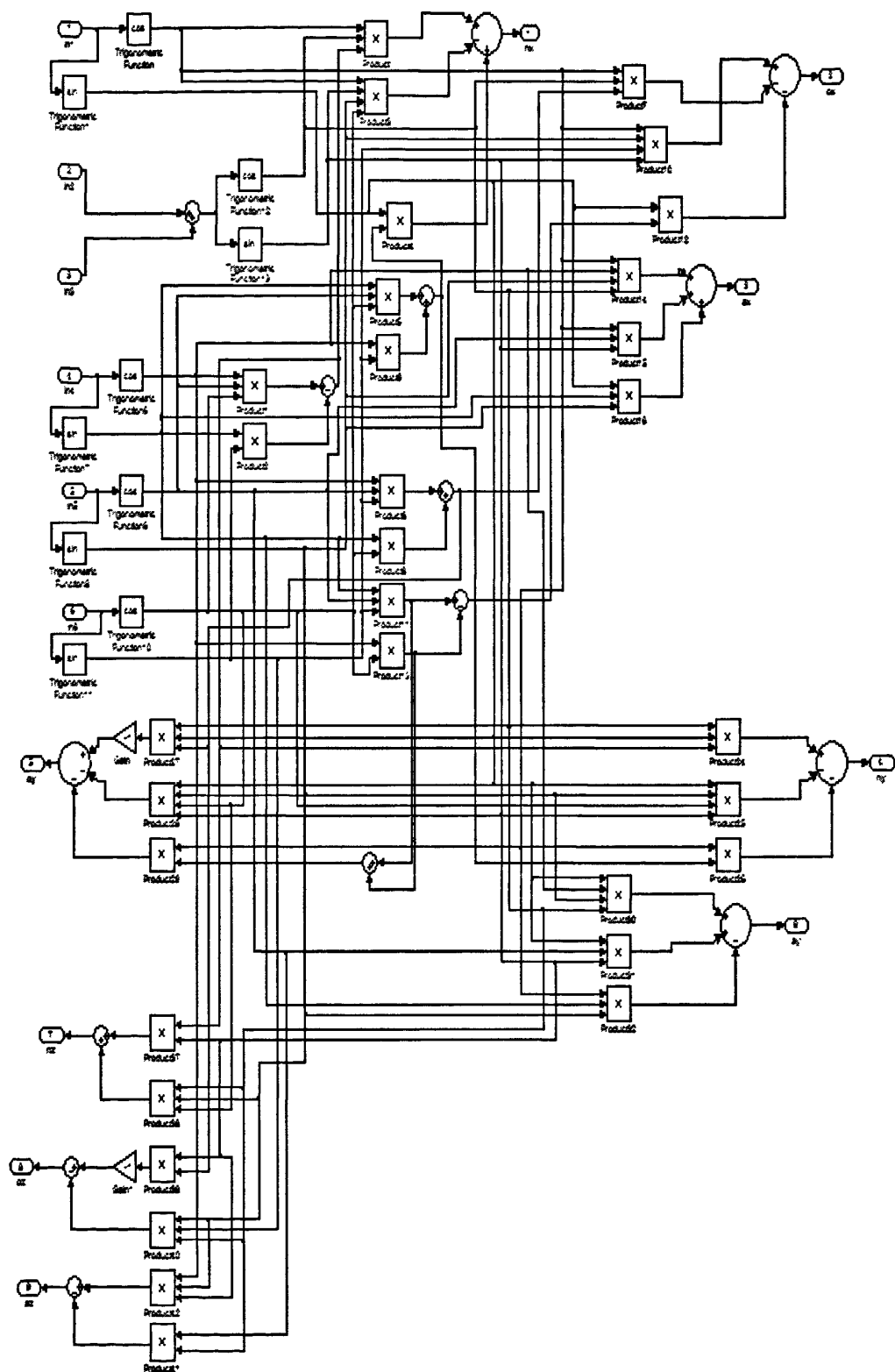


图 6.8 运动学逆解模块中的计算模块

在图 6.5 中, VR Sink 模块为 Simulink 与虚拟工具箱的接口, 即为与虚拟机器人的接口。双击 VR Sink 模块, 打开“Parameters: VR Sink”对话框, 如图 6.9 所示。

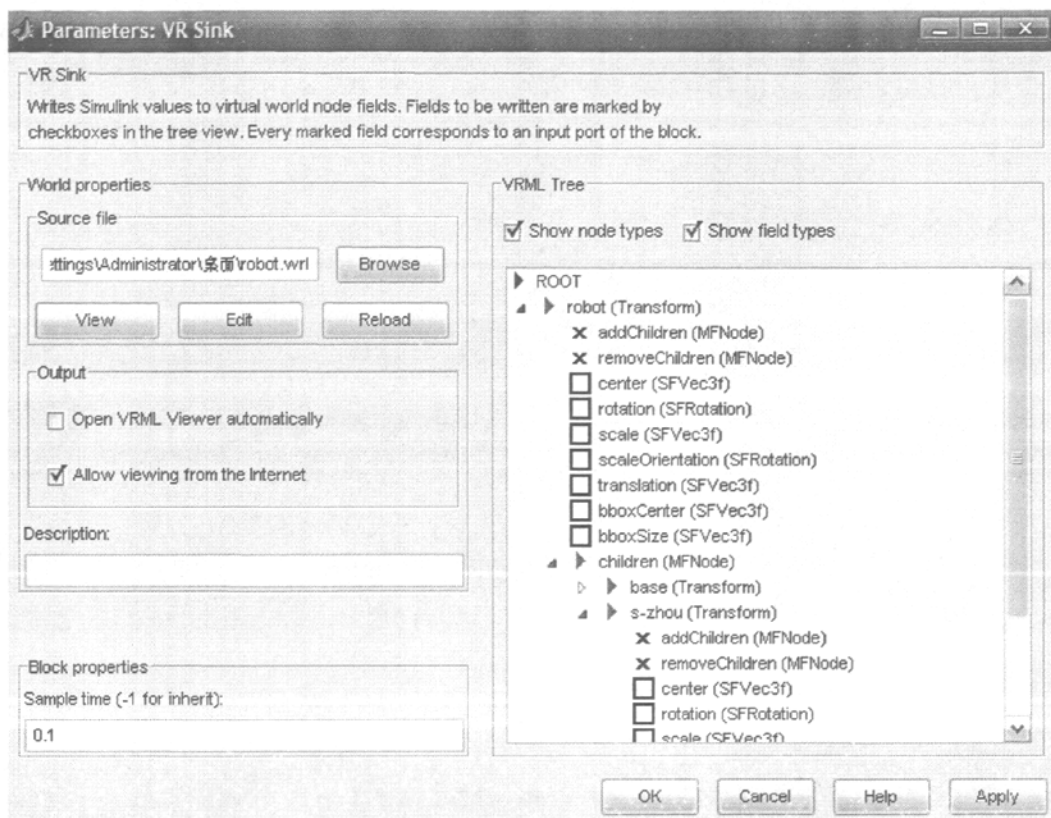


图 6.9 “Parameters: VR Sink”对话框

勾选上 S、L、U、R、B、T 六个轴 Transform 节点下的 rotation 后, 点击“OK”, 当关闭接口对话框后, VR Sink 模块中就会出现与所选标记点相关输入端口, 如图 6.10 所示。

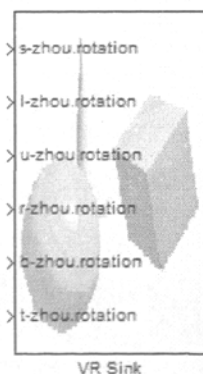


图 6.10 VR Sink 模块的输入节点

程序编译无错误后双击 VR Sink 模块, 打开虚拟世界的浏览器界面, 如图 6.11 所示。单击“运行”命令, 就可以看到机器人的虚拟现实仿真模型按着先前设计的方式进行仿真运

动了。

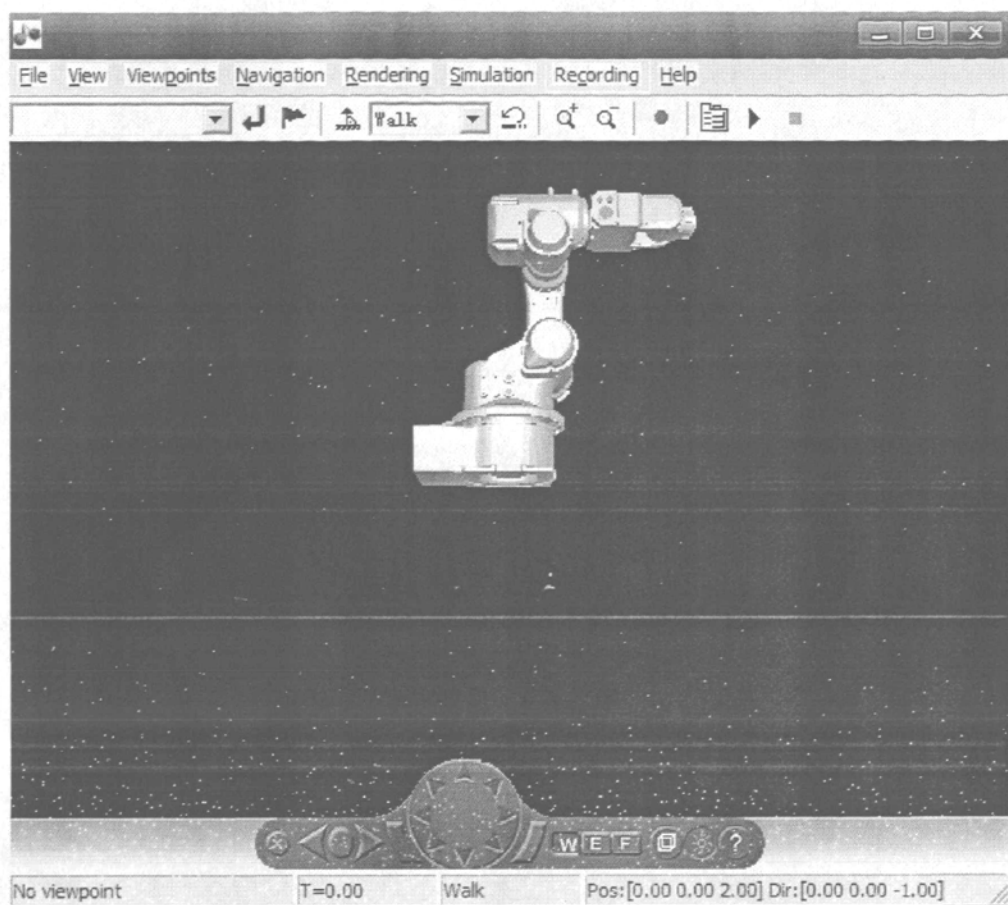


图 6.11 虚拟世界的浏览器界面

6.3.2 机器人运动学逆解的验证

根据 Motoman-HP3 机器人的结构特点, 采用 D-H 方法建立了其运动学模型, 推导出机器人的运动学正逆解公式, 但是正解的方法比较成熟, 结论的正确性已经得到证实。在本系统中, 逆解的正确与否直接关系到对现实机器人的遥控准确性。因此, 本节将对机器人运动学逆解进行验证。

机器人逆运动学是已知机器人末端执行器的位姿, 求机器人机构各个关节的运动情况。在 HP3 型机器人中, 先输入末端执行器的期望的位置值, 代入 5-2-2 小节中机器人逆解求解公式中, 依次求出关节角 θ_1 、 θ_3 、 θ_2 、 θ_6 、 θ_4 、 θ_5 , 然后根据公式求出 $\dot{p}_x$ 、 $\dot{p}_y$ 、 $\dot{p}_z$, 并判断是否等于 p_x 、 p_y 、 p_z , 如相等则说明逆运动学正确。

通过对运动学逆解所建立的数学模型编程后,如图 6.7 所示,由计算机进行模拟仿真。最后仿真出各组数据所对应的空间位置,如表 6-1 所示。

表 6-1 Motoman-HP3 机器人正逆解角度对比

θ_1	θ_2	θ_3	θ_4	θ_5	θ_6	p_x	p_y	p_z
20	-40	50	10	35	0	599	207	421
19.9997	-40.0035	49.9949	10.0003	35.0000	0.0000	598.9995	207.0009	420.9993
-30	-50	40	-40	20	-70	633	-343	226
-30.0002	-50.0004	40.0003	-39.9997	19.9968	-70.0005	633.0008	-342.9960	226.0057
30	-60	30	0	45	0	641	368	92
29.9997	-60.0035	29.9949	0.0000	45.0000	00000	641.0053	368.0003	92.9986

从表 6.1 中可以看到,通过第 5-2 节中所提出的机器人逆解求解公式求得的各关节角度与应得值之间的误差很小,一般可以保持在 10^{-4} 数量级。所得结果完全可以满足实际工业生产重的精度要求。故可知本文中的机器人逆解的解析算法是合理可行的。

图 6.12、6.13 和 6.14 分别是三组数组通过 MATLAB 仿真得出的对应空间位置。

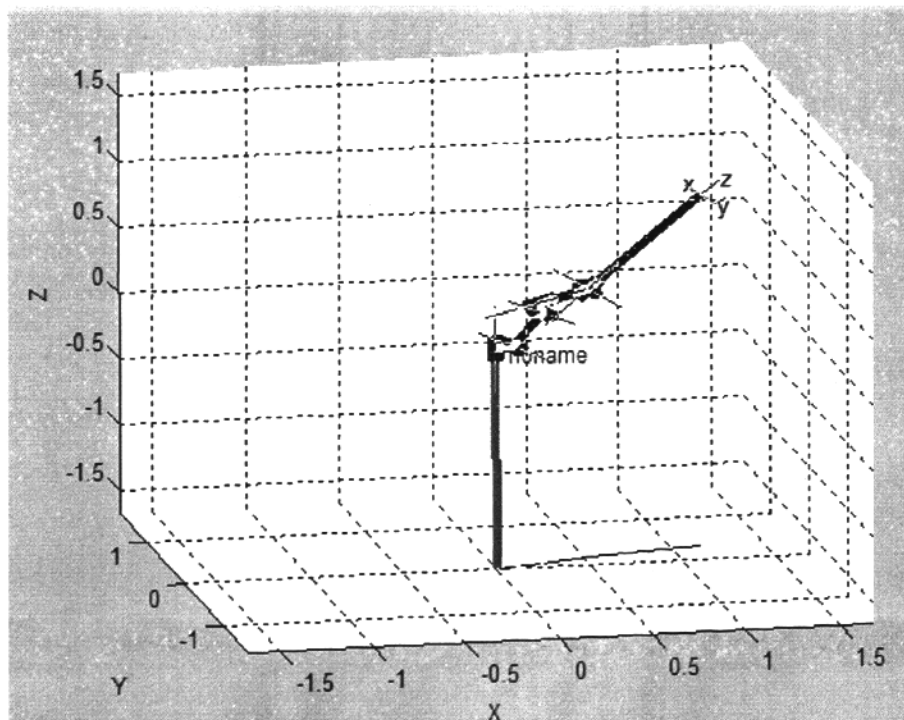


图 6.12 第一组数据对应的空间位置

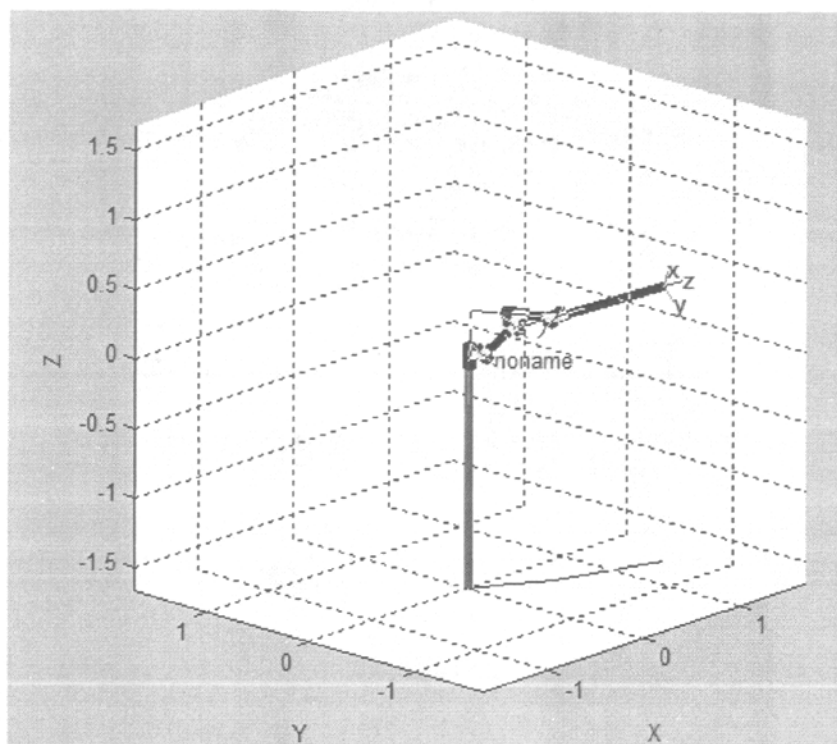


图 6.13 第二组数据对应的空间位置

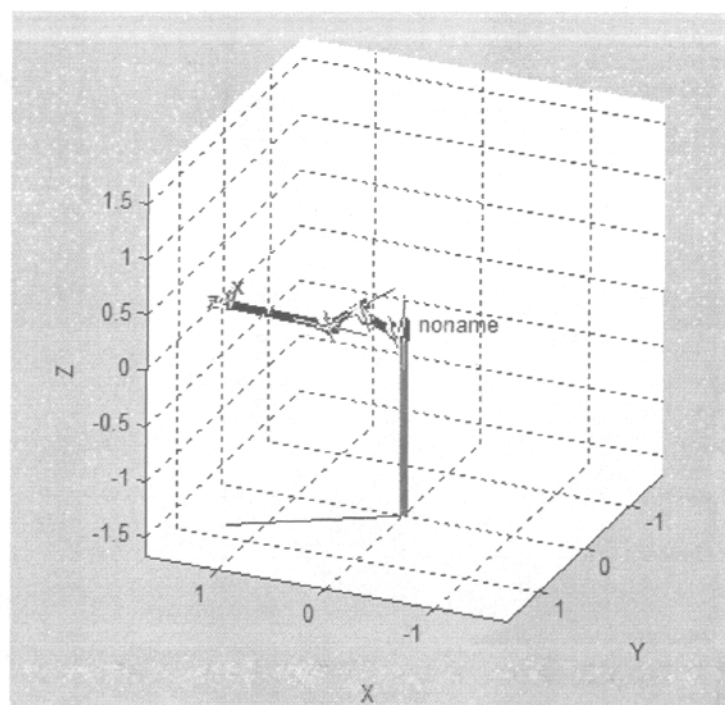


图 6.14 第三组数据对应的空间位置

6.4 Sever 服务器端的测试

本系统服务器端部分程序如下:

```

struct STATUSINFORMATION                                     //机器人返回的状态信息结构
{
    CString      curJobName;
    short        curJobLine;
    short        curJobStep;
    WORD         d1;
    WORD         d2;
    short        error;
    double       xp[12];
    double       jp[12];
    WORD         rconf;
};
class Robot
{
public:
void JobWait();                                           //此函数当任务等待任务完成后才返回
bool ServoOff();                                         //关闭伺服电源
void GetRobotStatus(STATUSINFORMATION &status);         //获得机器人的状态信息
bool StartStep(POSTION &pos);                           //单指令执行
.....
bool ServoOn();                                         //打开伺服电源
bool JobDownLoad(CString jobName);                     //把任务下载到控制器上
short disConnect();                                    //中断和机器人连接
HWND inputHwnd;                                         //inputHwnd 是使用此 robot 类的窗口句柄
short nCid;                                             //和控制柜连接的通道值
short BuildConnect();                                  //建立和机器人的连接
};
#endif
// !defined(AFX_ROBOT_H__F59A748B_2768_466F_A388_46277C8A8B06__INCLUD
ED_)
Robot::Robot()
{
}
Robot::~~Robot()
{
    short rc;
    rc=disConnect();
    if(rc!=0)
        AfxMessageBox("disClose()失败");
}
short Robot::BuildConnect()
{
    short ncid;

```

```

short rc;
char cur_dir[ MAX_DIR];           //当前的目录
char *IPAddress= TEST_IP_ADDRESS; // 机器人的 ip 地址
_getcwd( cur_dir, sizeof(cur_dir) );
if( TEST_TRANS_MODE==0 )
{
    // 使用 com 口进行通讯
    ncid = BscOpen( cur_dir, PACKETCOM );
    if( ncid < 0 )
    {
        return( ncid );
    }
    // 设置串口的通讯参数
    rc = BscSetCom( ncid, 1, 9600, 2, 8, 0 );
    if( rc != 1 )
    {
        rc = BscClose( ncid );
        return( -1 );
    }
} else
{
    rc = BscConnect( ncid );           // 与机器人建立连接
    if( rc != 1 )
    {
        rc = BscClose( ncid );
        return( -1 );
    }
}
return( ncid );
}
short Robot::disConnect()
{
    short rc;                         //取消和控制柜的连接
    rc = BscDisconnect( nCid );       // 取消连接句柄
    rc = BscClose( nCid );
    return( rc ); // 0:Normal completionOthers:Failed to release
}
.....

```

图 6.15 为本系统的 Sever 服务器端界面，首先要在机器人控制服务器端打开此程序。当系统开始运行后，点击“建立服务器”，界面上会显示“服务器建立成功”，此时表示正常工作。

服务器建立完成后，将会接收由 MATLAB/Simulink 虚拟现实仿真的数据结果，并发送给现实机器人执行，完成基于虚拟现实的遥操作机器人的功能。

MATLAB/Simulink 虚拟现实仿真中的数据结果将会由 Simulink 中 Sinks(系统输出模块

库)中的 To File 模块保存并生成.mat(数据格式为 $2 \times n$ 的矩阵形式)文件, 存放于 X:\MATLAB701\work 文件夹中。

通过实验, 实现了客户端与机器人之间的通讯功能。

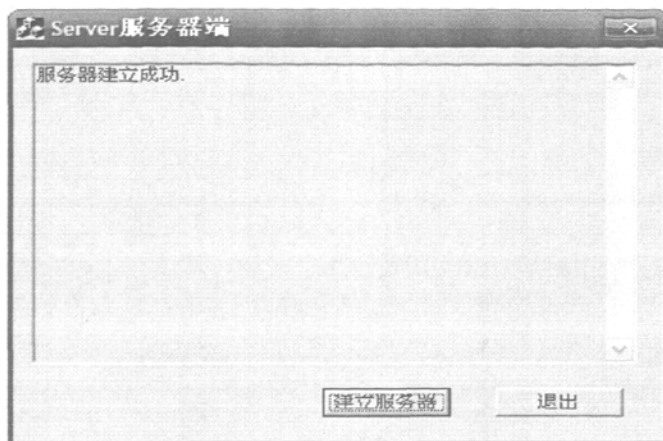


图 6.15 Sever 服务器端界面

6.5 机器人协调控制的实现

在之前的系统实验中, 实现了对一台机器人的单独控制, 为了实现两个机器人之间的协调控制, 需要扩充之前的实验程序。首先, 在机器人虚拟模型建立时, 将两台机器人的模型导入 V-Realm Builder2.0 中, 如图 6.16 所示。

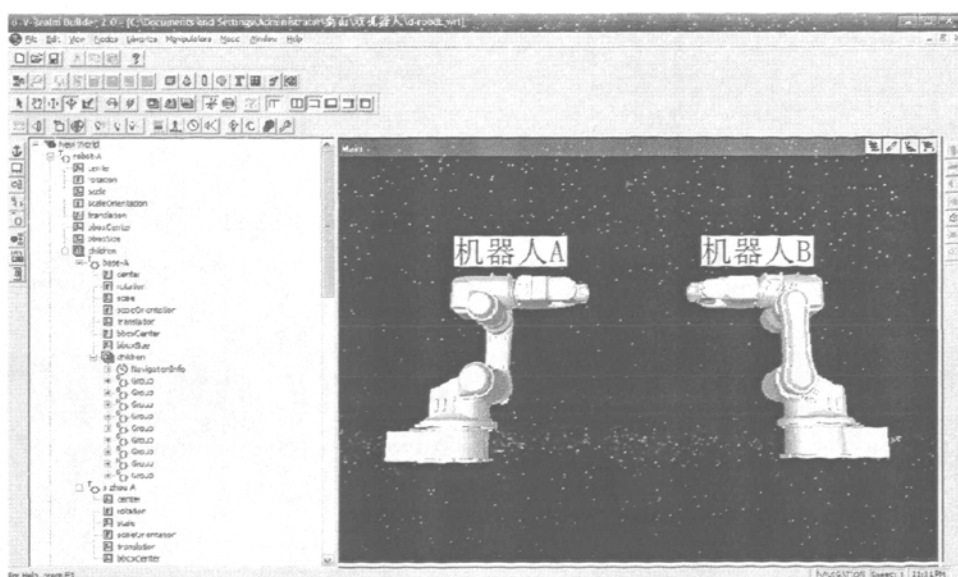


图 6.16 两个机器人的虚拟模型

在 MATLAB/Simulink 仿真中，机器人运动学仿真结构如图 6.17 所示。

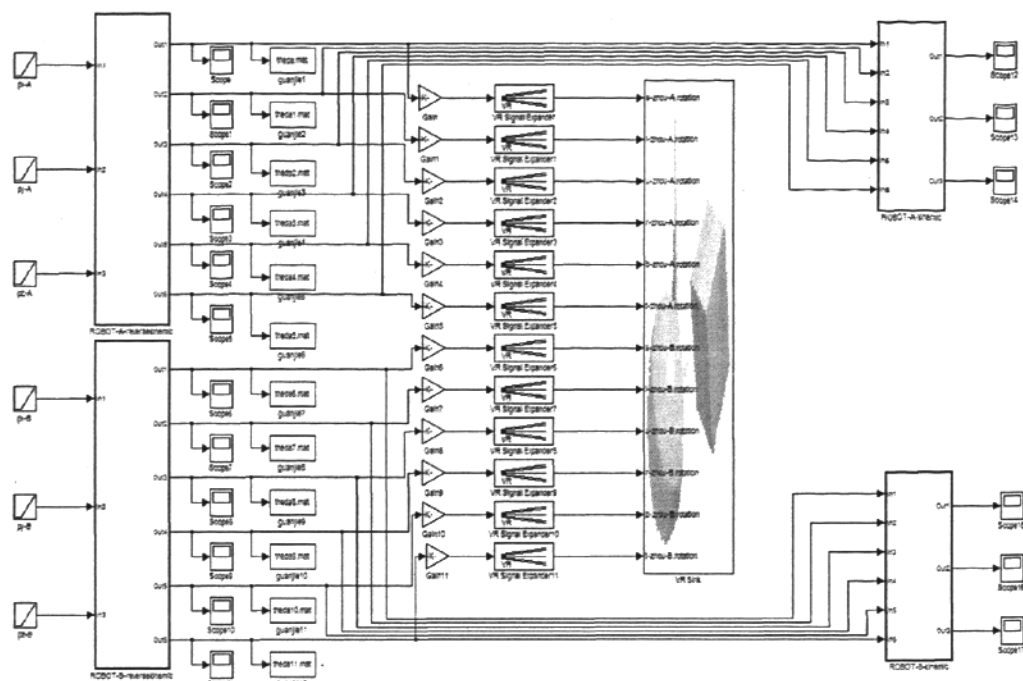


图 6.17 两个机器人的运动学仿真结构图

双击“VR Sink”模块，显示虚拟世界的浏览器界面，如图 6.18 所示。



图 6.18 两个机器人的虚拟世界浏览器界面

接下来,设计机器人 A 和机器人 B 要完成的动作和运动的时间,设定一套固定轨迹运动,虚拟世界中的机器人将按照仿真步骤运行,并将数据传输给现实机器人,带动现实机器人运动。

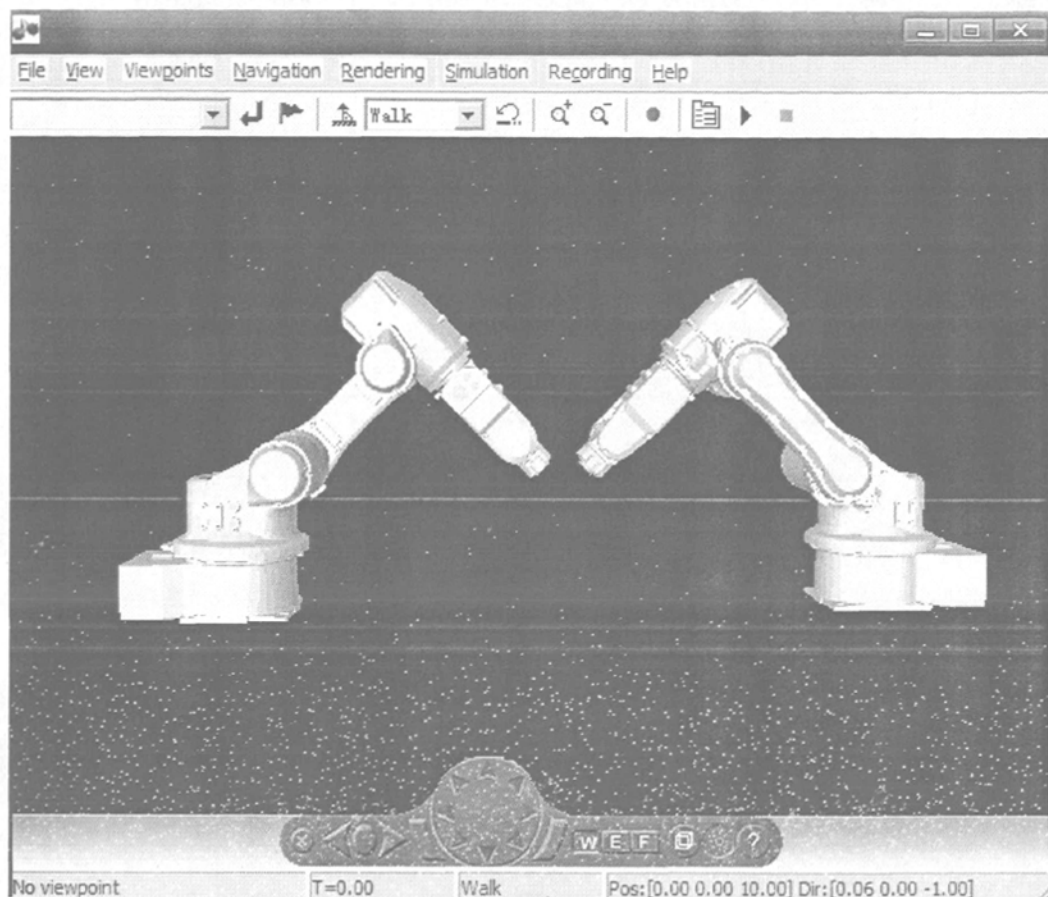


图 6.19 虚拟世界中的机器人协调仿真

6.6 小结

本章实现了基于虚拟现实技术的遥操作机器人协调控制系统。实验证明机器人模型的建立, MATLAB/Simulink 中机器人运动学的求解和虚拟现实的仿真都达到了设计的功能目标, 服务器端也能够正常的运行。作为机器人学专业方向研究生的实验系统, 达到了很好的效果。

第七章 总结和展望

基于虚拟现实的机器人遥操作本质上是一种远程控制的方式，是在遥操作机器人系统上实现的。而遥操作机器人是除工业机器人外的另一类具有极大实用价值的机器人，因此对遥操作机器人以及基于虚拟现实的遥操作的研究是很有意义的。

本文通过对国内外基于虚拟现实技术的遥操作机器人系统的分析和研究，从系统硬件组成、三维建模、通信协议、控制方式等方面进行了全面的归纳和总结，为系统的研究与开发提供重要的参考依据。本系统基于客户端/服务器端模式，采用 MATLAB/Simulink 中的虚拟现实仿真，并通过程序接口的通讯，成功实现了系统的各个功能。

在系统的设计和开发中，体现了以下一些特点：

(1)通过虚拟现实技术，有效地解决了通信延迟带给遥操作系统的影响，并实现了机器人遥操作的可视化；

(2)实现机器人遥编程控制方式，设计并实现了机器人数据控制协议；

(3)实现遥操作机器人的协调仿真控制，为分布式多机器人协作打下基础。

但是本系统依然存在不足及需要扩展和完善的地方：

(1)遥操作机器人软件中的部分设计功能还没有真正实现。受实验条件所限，整个系统只完成了对一台机器人的遥操作控制，两台机器人的协调控制在虚拟环境下进行了仿真；

(2)完善本系统中 MATLAB/Simulink 的虚拟现实仿真平台，加入 Joystick 输入设备和 Magellan 空间鼠标输入设备，增强操作者与虚拟现实环境的交互性，更好的通过虚拟现实技术控制遥操作机器人；

(3)增加双机器人协调控制的轨迹规划，更好地完成机器人的运行路径。

上述不足之处说明对遥操作机器人的研究还有待进一步深入，现有的基于虚拟现实技术的遥操作机器人软件依然不很完善，需要在今后的工作中继续学习相关理论知识，不断改进和完善整个系统。

参考文献

- [1] <http://www.kepu.net.cn/gb/technology/robot/secret/sec102.html>
- [2] 谢存禧, 张铁. 机器人技术及其应用[M]. 北京: 机械工业出版社, 2005
- [3] Sheridan T. B. Telerobotics, Automation and Human Supervisory Control. The MIT Press. 1992: 1~12, 98~108
- [4] Hirzinger G, Brunner B, Dietrich J, Heindl J. Sensor-Based Space Robotics-ROTEX and Its Telerobotics Features. IEEE Transactions on Robotics & Automation. 1993, 9(5): 649~663
- [5] Taylor K, Dalton B, Trevelyan J. Web-base(17): 49~57
- [6] 于涛. 具有力觉反馈的遥操作机器人研究[D]. 长春: 吉林大学, 2005
- [7] 蒋新松. 机器人学导论[M]. 沈阳: 辽宁科学技术出版社, 1994
- [8] 樊滨温. 遥操作空间机器人系统研究[J]. 哈尔滨工业大学学报, 1999, 1:111~113
- [9] 日本机器人工业学会. 机器人技术手册[M]. 北京: 科学出版社, 1996:1~13
- [10] 王田苗. 国家 863 计划先进制造与自动化技术领域机器人技术主题发展战略的若干思考[J]. 机器人技术与应用, 2002, No.3:2~7
- [11] 杨磊, 何克忠, 郭木河等. 虚拟现实技术在机器人技术中的应用和展望[J]. 机器人, 1998, 20(1)
- [12] 王跃虎. 基于虚拟现实的遥操作机器人研究[D]. 天津: 河北工业大学, 2006
- [13] 雷拥军, 吴宏鑫. 空间机械臂控制及其在空间探测中应用综述[C]. 中国空间科学学会空间探测专业委员会第十六次学术会议议论文集(下), 2003:502~507
- [14] 段新昱. 虚拟现实基础与 VRML 编程[M]. 北京: 高等教育出版社, 2004.03
- [15] 姬莉霞, 魏斌, 张雷. 虚拟现实技术及其应用[J]. 黑龙江科技信息, 38~39
- [16] 杨宝民, 朱一宁. 分布式虚拟现实及应用[M]. 北京: 科学出版社, 2000
- [17] 李刚俊. 基于虚拟现实的冗余度机器人运动规划及仿真研究[D]. 西安: 西安交通大学, 1998
- [18] 鉴萍. 基于 Internet 的遥操作机器人系统时延控制研究[D]. 济南: 山东大学, 2006.05
- [19] Sherman A, Murat C C, Tendick F. Comparison of Teleoperation Control Architectures for Palpation Task[A]. Proc. of Symposium on Haptic interfaces for Virtual Environment and

- Teleoperator Systems[C]. Orlando, Florida, 2000: 1996-1923
- [20] Hannaford B. Stability and Performance Trade-offs in Bilateral Telemanipulation[A]. Proc. Of IEEE Int. Conf, on Robotics and Automation[C]. 1989: 1764-1767
- [21] 朱徐华. 网络环境下移动机器人的建模与控制方法研究[D]. 南京: 南京理工大学. 2006.06
- [22] 冯华山. 基于 Internet 的远程控制机器人系统[D]. 西安: 西北工业大学, 2004.03
- [23] 郭志刚, 李治柱, 赵春云. 分布式对象计算技术及对象互连技术[J]. 上海交通大学学报. 1998, 32(4):135~139
- [24] 黄海峰. CORBA 的通信机制及性能研究[J]. 计算机工程与应用. 2003, 23:157~159
- [25] 刘振宝. 机器人网络控制平台系统设计和实验研究[D]. 北京: 北京工商大学. 2007
- [26] 郭亚宁, 陈一民. 基于 Web 的机器人遥操作[J]. 计算机工程, 2003, 29(18):154~156
- [27] Sheridan, Thomas B. Space Teleoperation Through Time Delay: Review and Prognosis, IEEE Transactions On Robotics and Automation, October 1993, 9(5): 592~605
- [28] Thomas B.Sheridan. Space Teleoperation Through Time Delay: Review and Prognosis. IEEE Transaction on robotics and automation, Vol.9, No.5,1993.592-606
- [29] Brady K., Tarn T.J. Internet-Based Remote Teleoperation. IEEE International Conference on Robotics&Automation, 1998, Leuven:65-70
- [30] Hirzinger G, Brunner B, Dietrich J, Heindl J. Sensor-Based Space Robotics-ROTEX and Its Telerobotics Features. IEEE Transaction on Robotics&Automation, 1993, 9(5): 649-663
- [31] 孙启媛, 曹作良, 彭商贤. 在 Internet 环境下移动机器人的远程控制策略[C]. 中国人工智能学会第 10 届全国学术年会论文集: 下卷. 北京邮电大学出版社, 2003:864~870
- [32] 孙启媛. 基于互联网的移动机器人网络控制系统研究[D]. 天津: 天津大学机械工程学院. 2004.06
- [33] 李臻. 基于 SolidWorks 的齿轮参数设计与三维造型软件的开发[D]. 兰州: 兰州理工大学, 2006.04
- [34] 熊剑. 基于 VRML 多用户环境的虚拟协作学习平台设计与实现[D]. 成都: 四川师范大学, 2007.03
- [35] 陆昌辉. VRML 入门与提高[M]. 北京: 北京大学出版社, 2003
- [36] 黄铁军, 柳健. VRML 国际标准与应用指南[M]. 北京: 电子工业出版社, 1999
- [37] 黄文丽, 卢碧红, 杨志刚. VRML 语言入门与应用[M]. 北京: 中国铁道出版社, 2003

- [38] 李红霞. 基于网络的虚拟三维环境中人机交互实现途径[J]. 工程图学学报, 2002(1):131~136
- [39] 汤晓燕, 云忠. VRML 网络虚拟建模技术在工程图学中的应用[J]. 理工高教研究, 2003, 22(1):102~103
- [40] <http://www.sg-motoman.com.cn/>
- [41] 石炜. MOTOMAN-UP6 机器人动力学分析[D]. 北京: 北京科技大学, 2003.03
- [42] 陈平. MOTOMAN-UPJ 型机器人运动学研究[D]. 镇江: 江苏大学, 2006.04
- [43] 张金峰. 带电小球在复合场中运动规律的三维演示方法研究[D]. 长春: 东北师范大学, 2007.05
- [44] 苏金明, 王永利. MATLAB7.0 实用指南(下册)[M]. 北京: 电子工业出版社, 2004.11
- [45] 陈伟斌, 张鑫. 一种新的基于 MATLAB 的虚拟现实交互性实现模式[J]. 计算机与数字工程, 2005(6):145~147
- [46] 王建中, 胡光艳, 田会方. 基于 Simulink 和虚拟现实工具箱的机器人行驶控制系统计算机仿真[J]. 国外建材科技, 2003(5):46~49
- [47] 张宏立. MATLAB 的虚拟倒立摆仿真[J]. 电器技术与自动化, 2005(9):101~105

研究生期间发表的论文目录

- [1] 战晓磊, 辛洪兵, 徐正兴.基于虚拟现实技术的网络控制机器人系统研究[J].北京工商大学学报(自然科学版)2007, 25(5): 25-28
- [2] 徐正兴, 辛洪兵, 战晓磊.行星齿轮传动均载问题的研究[J].北京工商大学学报(自然科学版)2007, 25(6): 31-34

致 谢

本论文完成之际，谨向所有给予我指导、帮助和关心的老师、同学以及我的家人们致以衷心的感谢！

首先感谢我的导师辛洪兵教授！

本论文是在导师辛老师的悉心指导下完成的。在我三年的研究生学习、生活过程中，辛老师时刻给予我悉心的指导和无私的关怀。他的高深的学术造诣，严谨的治学态度，踏实的研究精神，给我留下了深刻的印象，将使我受益终生。在日常生活和人生道路上，导师也以他丰富的人生阅历给我提了许多宝贵的建议和忠告。在此，对导师的培养谨致以最衷心的感谢。

其次，我还要感谢实验室中徐正兴、朱冉等同学，他们给予我的帮助和大力支持使我受益匪浅。

最后，还要感谢我的父母，从小到大无私培养我、关怀我，在我求学的道路中不断鼓励我、支持我，正是由于他们多年来对我的信任和理解，给了我克服多种困难的勇气与信心，使我在学习中、在人生旅程中不断前进。