

摘要

大时延遥操作技术是空间机器人领域的一项关键技术。大时延遥操作的一个核心问题是如何在大时延和有限带宽的条件下,保证遥操作的平稳性,同时提高遥操作的透明性。为了解决这一问题,本文的基本思想是利用机器人的局部自主能力,把时间延迟排除在底层控制回路之外,保证遥操作的平稳性;同时,利用基于虚拟现实的人机接口技术提高遥操作的透明性。碰撞检测技术是虚拟现实应用中的一项关键技术,对于复杂虚拟环境,碰撞检测往往成为系统的计算瓶颈。本文针对遥科学实验应用,对多体快速碰撞检测技术进行了研究。本文的主要研究内容分为三个部分:大时延遥操作系统框架研究、遥编程技术研究,以及快速碰撞检测技术研究。

(为了建立灵活的人机协调机制,本文基于监控思想,提出了一个基于 Petri 网的大时延遥操作系统框架。该框架涵盖了任务建模、任务协调,以及任务监控等几个方面;本文研究了基于 Petri 网的的任务建模方法。针对自由运动,提出了一个基于路径关键点提取的动作级任务建模方法;针对复杂任务,提出了一个根据子任务的 Petri 网系统模型,通过变迁细化方法建立任务的分层 Petri 网模型的方法;本文提出了一个基于 Petri 网的任务协调方法,该方法根据任务空间的谓词/变迁网描述,通过状态识别和动作识别,实现了从操作员的连续动作到任务的动作序列的转化。

本文对遥编程所涉及的一些关键技术,包括:虚拟预测仿真、程序自动生成,以及误差校正,进行了研究。本文实现了一个遥编程系统,并基于遥编程建立了一个大时延遥操作试验系统。进行了典型的大时延遥操作仿真试验,成功完成了高精度遥操作任务。

本文研究了基于空间分割的快速碰撞检测方法,其中包括:基于均匀空间分割的快速碰撞检测和基于非均匀空间分割的快速碰撞检测。提出了一种基于均匀空间分割的快速碰撞检测算法 USSCD,与经典的 ICOLLIDE 算法进行了对比试验,试验结果表明,在物体均匀分布的情况下,随物体数量的增多,USSCD 表现出明显的优势;提出了一类非均匀空间分割方法,在该方法中,空间分割由基于密度的非均匀空间分割和对每个子空间进行均匀分割两个步骤实现;提出了一种基于投影的非均匀空间分割方法。在该方法中,通过物体的轴向投影直方图表征空间分布密度,通过区域生长实现空间分割;提出了一种基于聚类的自适应空间分割方法。该方法由两个部分构成,首先,通过一个基于局部密度的聚类算法对物体进行聚类;然后,根据聚类结果,通过简化分割方法实现空间分割。)

关键词: 遥操作、遥机器人、遥编程、虚拟现实、碰撞检测、空间分割 Petri 网

Abstract

Teleoperation through large time delay (T^3D) is one of the key techniques in the field of space robot. One of the core problems in T^3D is how to maintain the stability and to promote the transparency of the teleoperation system, on the condition of large time delay and limited bandwidth in the communication link between the operator control station on the earth and the robot workstation in space. This paper presents two basic points to cope with this problem. First, the impact of large time delay in the servo-control loop is avoided through the local automation of the telerobot, therefore, the stability is maintained. Second, the transparency is promoted by developing the advanced human-computer interface based on Virtual Reality (VR). Collision detection is one of the key techniques in the application of VR, which usually presents as the bottleneck of the computing performance in complex virtual environment. With the background of telescience experiment application, this paper concentrates on the research of accelerated algorithm for N-Body collision detection. The main contents in this paper include three aspects: framework for T^3D system, teleprogramming, and accelerated collision detection methods.

To develop a flexible coordination mechanism between the human operator and the machine, a Petri Nets based unified system framework for T^3D is proposed, which, follows the concept of supervisory control, and covers the main aspects of the system, such as task modeling, task coordination, and task supervision. Under this framework, the task modeling methods based on Petri Nets are studied. An action-level task modeling method for free move is proposed, which is based on a method of key point extraction from the trajectory. A hierarchical task modeling method is proposed, which constructs the hierarchical model through the transition-refinement according to the Petri Nets system model of the subtasks. A task coordination method is proposed, which, according to the Predicates/Transition Nets representation of the task space, through state recognition and action recognition, transacts the continuous actions of the operator to a series of instructions for the telerobot.

In this paper, several key techniques relative to teleprogramming are studied, including: predictive display through virtual environment, automatic program generation, and error correction of the virtual environment. A teleprogramming based experimental T^3D system is developed. The typical peg-in-hole experiment is conducted, and T^3D is simulated by artificial delay setup. The system is succeeded in the part-mating task, which proves the validity of the teleprogramming system for high-precision task.

In this paper, a set of accelerated collision detection algorithms based on space subdivision are developed. One accelerated N-Body collision detection algorithm based on uniform spacial subdivision, called USSCD, is proposed. The performance of USSCD is compared with the well known algorithm, ICOLLIDE, through a series of experiments. The results show that, when the objects are evenly distributed, and the number of objects is

large, the performance of USSCD is better than that of ICOLLIDE. In this paper, a class of non-uniform spacial subdivision method is proposed, which is composed of two steps: first, the space is coarsely subdivided non-uniformly, based on the distribution density of the objects, then, every subspace is uniformly subdivided. A projection based non-uniform spacial subdivision method is proposed, in which the distribution density is measured by the histogram of the objects projection on three coordination axis, and the space is subdivided through a 'region-grown' algorithm. An adaptive space subdivision method based on clustering is proposed, which completes space subdivision through two steps: first the objects are classified through a new local-density based clustering algorithm, then the space is subdivided according to the clustering.

Keywords: Teleoperation, Telerobotics, Teleprogramming, Virtual Reality, Collision Detection, Spacial Subdivision, Petri Nets.

第一章 绪论

1.1 研究背景

在人类走向太空的过程中，空间机器人扮演着越来越重要的角色。自 20 世纪 60 年代以来，从单纯的星球表面探测任务开始，发展到今天，空间机器人的应用已经扩展到包括在轨装配与维修、舱内科学负载服务、星球表面探测等众多领域。随着应用领域的不断扩大，空间机器人所承担的任务的复杂性也不断增加，从结构化环境中的重复性操作到非结构化环境下的非重复性操作，从小范围的短时的星球表面漫游，到大范围的长时间的星球表面勘测。为了适应环境和操作任务的复杂性，需要空间机器人具有较强的自主能力。从长远看，空间机器人必然朝具有高度的感知、判断、决策和规划能力的方向发展。高度自主的空间机器人，能代替人从事各种空间作业，而无需人的介入。然而，限于目前智能技术水平，全自主空间机器人在短期内难以走向实用。在这种情况下，一个现实选择就是走人机结合的道路，发展遥操作空间机器人（Telerobotics，以下简称遥机器人）。对于遥机器人，机器人系统具有局部自主能力，一定程度上能够进行感知、反应和规划，但整个系统的智能是通过操作员和机器人的相互配合实现的。一般，根据自主能力，遥机器人介于完全靠预编程和手动控制的自动化装置，和全自主机器人之间，被称为第二代空间机器人[1]。

本文对于大时延遥操作技术的研究主要以遥机器人为背景。从狭义上讲，大时延遥操作是指操作员对操作器进行连续的、直接的控制，在操作员和操作器之间形成闭合的控制回路，在此控制回路中存在较大时间延迟（一般为秒级）；广义上讲，大时延遥操作泛指在操作员和操作器间存在较大通信延迟的人机交互系统。操作员既可以对操作器进行连续控制，也可以向操作器发送符号命令并对执行情况进行监视。在遥机器人系统中，空间机器人由地面操作员控制，在操作员控制站和机器人工作站的通信线路中往往存在数秒的数据传输延迟（空间站和地球间长达 5 秒，月球和地球间在 6 秒以上，是为大时延）。在空间应用中，由于操作环境的危险性、宇航员的高昂代价等因素，大时延遥操作成为遥机器人的一种主要形式。

大时延遥操作面临的主要困难在于：在操作员和操作器间存在大时延，而且通信带宽有限（很难满足实时视频信息的传输）。这使得传统的主从遥控系统在保持控制系统的稳定性和具有必要的动态性能之间，很难权衡；同时，带宽限制使得操作员难以获得充分的操作现场的反馈信息。因此，研究补偿大时延和有限带宽的影响、提高遥机器人系统性能的大时延遥操作技术，对于空间机器人的发展具有十分重要的意

义。

我国在航天技术领域的发展,从长远的角度必然要发展载人航天技术,建立大型空间站。但在目前的技术和经济条件下,可能的途径是建立小型的空间实验室,在短期有人照料,长期无人职守的状态下,由地面操作人员控制空间的机器人,进行空间科学实验,完成各项维护、操作任务。所以,大时延遥操作技术,是我国空间机器人领域中的一项关键技术[2]。

虚拟现实(Virtual Reality,简称VR)技术是近年来国际上兴起的一种先进人机接口技术,随着计算机科学的快速发展,VR技术已显示出广泛的应用前景。在遥机器人领域,利用VR技术实现系统人机界面已引起了广泛重视。美国航空航天局(NASA)从80年代初就开始支持VR技术的开发,并于1993年10月出版了“NASA虚拟环境研究、应用和技术”的白皮书,介绍了NASA有关VR技术的发展进程、主要研究成果和1994~1998年的发展规划,其中的主要研究项目包括:航天应用中的虚拟环境空间操作的可视化和用于遥操作/遥科学机器人的虚拟环境控制和训练系统。

在大时延遥操作中,一个关键的问题是如何在大时延和有限带宽条件下,使操作员获得及时的、连续的视觉和运动觉反馈信息,实现遥现(Telepresence)。这对于增加遥操作的透明性(Transparency),充分发挥操作员的感知、判断能力,从而提高系统的整体性能具有重要的作用。应用虚拟现实技术,为补偿大时延和有限带宽的影响提供了一种有效手段,主要作用体现在以下两个方面:

1) 实现预测显示(Predictive Display)

根据物体和机器人的几何模型和物理模型,构造虚拟环境和虚拟机器人。一方面,虚拟机器人能够即时响应操作员的动作,进行连续运动,从而提供即时的视觉反馈;另一方面,通过仿真运行,虚拟机器人能够按照规划的程序预测显示机器人的运动轨迹,使操作员可以对指令序列进行验证和优化,增加操作指令的合理性,从而提高系统的安全性和可靠性。

2) 实现遥现

在结构化操作环境中,物体的几何模型是已知的,根据机器人和物体的位姿信息,通过图形重构的办法,可以合成虚拟的操作环境,将操作环境的真实状态显示出来。在虚拟环境中,不仅实现了立体显示,而且操作员可以改变视点进行漫游,从而通过位姿信息而非视频图像实现了操作现场的遥现。这将大大降低对通信带宽的要求,从而有望在有限带宽条件下实现对操作环境的连续观察。

我国在空间机器人技术和虚拟现实技术方面的研究起步较晚。目前,大时延遥操作及虚拟现实技术有许多关键问题亟待解决。在1999年全国第一届空间机器人及遥科学技术研讨会中,明确提出把应用虚拟现实的大时延遥操作技术作为我国空间机器人研究中的一项关键技术[2][3]。本文研究的目的是对大时延遥操作及其相关虚拟现实技术进行探索,为我国空间机器人技术的发展提供基础。

1.2 大时延遥操作国内外现状

国际上,包括美国、加拿大、欧共体各国、日本和俄罗斯等国在内的许多国家和机构都对空间遥机器人展开了研究。其中比较有代表性的是美国航空航天局(NASA)、日本空间发展局(NASDA),以及欧空局(ESA),他们都制定了空间遥机器人的研制和试验计划。

NASA自1993年来就制定了空间遥机器人的大规模研制计划。在NASA的计划中,将空间遥机器人分为三个主要的应用领域:在轨装配与维修、科学负载服务,以及星球表面探测。目前主要集中于在轨装配维修机器人和星球表面探测机器人两个方面,而用于科学负载服务的舱内机器人研制计划基本于1994年中止。在NASA的计划中明确提出应用虚拟现实的人机接口技术和大时延遥操作技术作为遥机器人的关键技术。1996年,美国发射了火星探路者(Mars Pathfinder),其中载有火星漫游机器人索杰纳(Sojourner)。索杰纳主要用于技术实验,验证导航和运动性能,其漫游范围只有几十米。卡内基梅隆大学研制了一个星球探测机器人试验系统Nomad。该系统已在Atacama沙漠地带进行了野外试验,漫游距离达到了200km,表明其驱动、定位、导航、视觉、遥操作等技术已取得关键性突破。从1992年开始,美国马里兰大学与NASA的喷气推进动力实验室(JPL)合作研究了漫游者空间机器人Ranger。Ranger是一个双手自由飞行遥机器人飞行试验系统,主要用于NASA计划中各项技术的在轨验证、核实。

欧空局(ESA)于1993年4月试验了舱内机器人ROTEX,成功地进行了多项操作实验,包括:借助预测显示系统的地——空遥操作,基于传感器的离线编程监控式操作。ROTEX是第一个舱内遥机器人试验系统。

NASDA于1997年11月发射了ETS-VII卫星,到1999年为止,成功地进行了各项科学试验任务,其主要目的之一就是试验地——空遥操作机器人技术,解决大时延问题。ETS-VII是第一个遥控式自由飞行空间机器人。在ETS-VII机器人实验中采用多项技术来克服大时延问题,其中主要有两种操作模式:基于3D预测仿真显示的遥操纵模式(Telemanipulation)和遥编程模式(Teleprogramming)。此外,借助ETS-VII卫星,德国实验了其应用虚拟现实的大时延遥操作系统GETEX。

1.3 主要研究内容

大时延遥操作的一个核心问题是如何保证系统的稳定性,同时提高操作的透明性。这里所谓透明性是对操作员能否有效控制操作器进行灵巧操作以及操作员能否获得即时的充分的反馈信息的一个综合度量。为了解决这一问题,本文的基本思想是利

用机器人的局部自主能力,把时间延迟排除在底层控制回路之外,保证操作的稳定性;同时,利用基于虚拟现实的人机接口技术提高操作的透明性。

本文的主要研究内容分为三个部分:大时延遥操作系统框架研究、遥编程技术研究,以及碰撞检测技术研究。

1.3.1 系统框架研究

大时延遥操作系统,实质上是一个人机合作的智能系统。决定系统性能的核心问题是如何协调人和机器间的功能。人机协调的基本原则是优势互补。一般地说,人的长处在于具有较强的感知、决策和规划能力,而机器的长处在于可以实现平稳的、高精度的操作。对于一个智能系统,其感知、决策、规划,以及执行等功能分布在不同的智能层次上(如决策层、执行层)。因此,人机协调问题表现为如何将不同层次的不同功能在操作员和机器之间进行分配。具体的分配方案取决于任务的复杂性、环境的复杂性、操作员的技能以及机器人的自主能力等方面的因素。比如,对于熟练的专业操作员、简单的任务,可以采用直接的主从控制方式,此时,操作员承担了大部分功能,而机器人只负责底层的执行;对于复杂的操作任务,采取监控方式(Supervisory Control),依靠机器人的局部自主能力,则是自然的选择。在实际操作过程中,由于不同阶段的操作复杂性和精度要求的变化,这种功能分配也应随之发生变化。不同的分配方案,对应为操作员在系统中的不同介入方式,灵活的人机协调机制应该满足操作员在不同层次的介入的需求。综上所述,大时延遥操作的一个核心问题是确立合理的、灵活的人机协调机制。本文对大时延遥操作的研究以监控方式为主,系统框架按照监控的原则建立。为此,需要系统框架满足如下几方面的要求:

1. 人机交互的方式遵从监控模式。在监控方式下,操作员的主要功能可以概括为:任务规划和任务监控两个方面。系统框架应该体现这两个方面的功能要求;
2. 操作员能在多个层次介入系统的控制。操作员能分别在任务级和动作级进行规划和监控;
3. 支持对系统性能的定量分析。基于系统模型,可以对系统的控制性能(包括能控性、稳定性等)进行分析,同时在大时延条件下,对系统的效率进行分析。

本文在大时延遥操作系统框架方面的研究,是为实现上述目标所进行的积极探索。本文从基于 Petri 网的系统框架概念出发,研究了基于 Petri 网的任务建模方法,其中包括基本任务建模方法和面向分层任务描述的复杂建模方法;面向任务规划,本文研究了基于 Petri 网的任务协调方法。

1.3.2 遥编程研究

在监控方式中,机器人的控制是通过抽象的符号命令实现的,操作员根据机器人对命令的执行情况,规划后续的操作。操作员规划的合理性取决于其对操作状态的判断。正确的判断一方面来自于对操作现场的观察,一方面来自于基于先验知识的预测。在大时延遥操作中,对每一个命令,操作员至少要等待一个通讯周期才能获得相应的执行结果,因此,仅仅根据来自操作现场的反馈信息,操作员难以进行连续的规划,只能采取“一步一停”(step-by-step)的方式。然而,在具体的动作过程中,人们习惯于连续地操作。并且,人的操作意图往往是通过直接地连续动作的方式来表达的,而不是抽象的符号命令。这种对于连续操作的需求与间断控制方式间的矛盾,成为大时延遥操作人机接口系统设计中的一个关键问题。解决这一问题的一个思路就是充分利用对于机器人和操作环境的先验知识,通过仿真对操作任务进行预测,同时,将操作员的动作转化为符号命令。类似于人机交互界面中经典的“所见即所得”的思想,遥编程概念利用虚拟现实预测仿真和自动程序生成,为解决上述矛盾提供了一种途径,对克服大时延的影响具有重要的意义。

本文对遥编程所涉及的一些关键问题,包括:虚拟预测仿真、程序自动生成,以及误差校正,进行了研究。在此基础上,基于遥编程概念实现了一个大时延遥操作试验系统,并通过典型的遥操作试验,对系统的性能进行了验证。

1.3.3 碰撞检测研究

在应用虚拟现实进行仿真的过程中,一个不可避免的问题就是碰撞检测。对于遥操作而言,要仿真机器人同操作环境发生接触的动态过程,首要的问题就是进行准确的碰撞检测。对虚拟现实技术本身而言,碰撞检测是影响计算效率的一个瓶颈,要实现实时的仿真和显示,必然要快速的碰撞检测。尽管在碰撞检测方面,已经有一些研究成果,但是快速而准确的碰撞检测仍是一个亟待解决的问题。

除了在大时延遥操作中的应用外,虚拟现实的另一个应用是空间遥科学实验仿真。在给定模型的基础上,应用虚拟现实技术对遥科学实验过程进行直观表达,将在很大程度上弥补由于在地面难以模拟空间微重力环境的不足。然而,在基于虚拟现实的遥科学实验仿真中,面临的一个难题是,表达的对象数量众多(如微粒沉淀仿真),而且在动态仿真过程中,对象间往往发生碰撞,在这种情况下,常规碰撞检测方法无法满足计算实时性的要求。

上述大时延遥操作中接触过程动态仿真的需要和遥科学实验仿真中的问题,促使本文对碰撞检测技术进行了系统的研究。其中,着重于对多体快速碰撞检测技术的研究。本文主要研究了基于空间分割的快速碰撞检测方法,其中包括:基于均匀空间分

割的快速碰撞检测和基于非均匀空间分割的快速碰撞检测。

1.4 主要贡献

本文的主要贡献包括如下几个方面：

1. 提出了基于 Petri 网的大时延遥操作系统框架。该系统框架适用于监控式系统，在逻辑层次上，应用 Petri 网模型，涵盖了任务描述、任务协调，以及任务监控等几个方面；
2. 提出了一套基于 Petri 网的任务建模方法。针对自由运动，提出了一个基于路径关键点提取的动作级任务建模方法；针对复杂任务，提出了一个根据子任务的 Petri 网系统模型，通过变迁细化方法建立任务的分层 Petri 网描述的方法；
3. 提出了一个基于 Petri 网的任务协调方法。该方法根据任务空间的谓词/变迁网描述，通过状态识别和动作识别，实现了从操作员的连续动作到任务的动作序列的转化；
4. 在国内首次研制并实现了一个遥编程系统，并基于遥编程建立了大时延遥操作试验系统，进行了仿真试验，成功完成了高精度遥操作仿真任务；
5. 提出了一种均匀空间分割计算模型，并根据该模型提出了一种基于均匀空间分割的快速多体碰撞检测算法，USSCD；
6. 提出了一类非均匀空间分割方法，在该方法中，空间分割由基于密度的非均匀空间分割和对每个子空间进行均匀分割两个步骤实现；
7. 提出了一种基于投影的非均匀空间分割方法。在该方法中，通过物体的轴向投影直方图表征空间分布密度，通过区域生长实现空间分割；
8. 提出了一种基于聚类的自适应空间分割方法。该方法由两个部分构成，首先，通过一个基于局部密度的聚类算法对物体进行聚类；然后，根据聚类结果，通过简化分割方法实现空间分割。

1.5 论文组织结构

本文的第二章，是对大时延遥操作技术和碰撞检测技术的综述；在第三章，介绍了基于 Petri 网的大时延遥操作系统框架。在这一部分，首先介绍了 Petri 网的基本概念，然后给出了系统框架的描述，在接下来的三个小节中，分别讨论了基于 Petri 网的基本任务建模方法、复杂任务建模方法，以及任务协调方法；在本文的第四章，讨论了遥编程技术，描述了基于遥编程的大时延遥操作系统实现，介绍了试验及其结果；在第五章，讨论了快速碰撞检测技术，先后介绍了基于均匀空间分割的快速碰撞检测和非均匀空间分割方法。

第二章 大时延遥操作技术与碰撞检测技术综述

大时延遥操作有两个基本的思路：双向遥操作和监控。双向遥操作是在传统的主从遥控方式的基础上，通过阻抗控制、预测显示、时延力反馈等手段对大时延进行补偿，操作员以连续的方式直接控制远地的操作器；在监控方式中，操作员主要以间断的符号命令的方式控制遥机器人，遥机器人具有局部自主能力。在这两种基本控制方案的基础上，共享控制通过将直接控制方式和监控方式结合起来，实现人和机器人的有机结合。本章首先对遥操作的体系结构进行了介绍，其中包括比较典型的 MOTES、ROTEX、NASREM、4-D/RCS、 SM^2 、SMART、OBTLC 等系统和参考模型，同时也介绍了其它一些体系结构的研究；然后，分别从双向遥操作和监控两个方面对大时延遥操作技术进行了介绍，在双向遥操作方面，介绍了预测显示技术以及基于虚拟环境的人机接口技术。在监控方面，本文依据 Sheridan 提出的监控理论[7]简要介绍了监控的基本概念，监控系统框架，以及监控中的关键问题——注意力分配问题。遥编程技术在监控框架下，通过先进的人机接口技术，实现了操作员的连续控制。本文对遥编程的基本思想及其涉及的关键技术进行了介绍。最后，本文介绍了在虚拟现实和机器人控制中非常重要的碰撞检测技术。

2.1 遥操作体系结构

NASA 的 JPL (Jet Propulsion Laboratory) 实验室对大时延遥操作进行了大量的研究。研制了基于命令序列结构的模块化遥机器人任务执行系统 MOTES (Modular Telerobot Task Execution System) [8]。该系统支持监控、遥操作，以及共享控制多种控制模式。在 MOTES 中，任务执行是通过对多个先进先出的任务队列的调度实现的，其中包括任务命令队列和反射命令队列。MOTES 面临的问题是由于任务队列是线性的，因而具有一定的局限性，同时由于反射命令的调度是独立于任务命令的，二者没有融合在一起，不便于操作员对其进行干预。

在 ROTEX 遥机器人实验中，系统提供了包括预测显示、任务规划和共享控制在内的多种功能。其控制结构采用分层递阶结构，分为：决策层、协调层和执行层三个层次[9]。一个比较典型的系统模型是为 NASA 的空间机器人项目 FTS (Flight Telerobotics Servicer) 设计的 NASREM (NASA/NBS Standard Reference Model for Telerobot Control System Architecture) 参考模型[10]。在 NASREM 中，整个智能系统的功能分布在不同的层次上，在每个层次，不同的模块形成局部的闭合控制回路。参

考模型对每个模块的功能进行了规范，一般包括三个功能：传感器处理（Sensor Processing, SP）、世界建模（World Modeling, WM），以及行为生成（Behavior Generation, BG）或称为任务分解（Task Decomposition, TD）。在 NASREM 概念基础上，在美国国防部 Demo III 工程中，开发了智能系统体系结构参考模型 4-D/RCS[11]，该模型集成了美国 NIST (National Institute of Standards and Technology) 的实时控制系统 RCS 和德国的动态视觉系统 VaMoRs 4-D。该模型采用分布式结构，系统由不同的智能节点构成，每个节点包括四个方面的功能，在 NASREM 中的 SP、WM、BG 的基础上，增加了价值判断（Value Judgement, VJ）功能，此外，节点一般拥有各自的知识库（Knowledge Database, KD）。在实现中，不同节点分布于不同的计算机系统中，节点之间通过通讯系统进行信息交换。不同节点是通过分层结构来组织的。在 4-D/RCS 中，强调了操作员的介入功能，每个节点的输入输出都与操作员界面相连，从而使得操作员可以介入系统不同的功能。

从 1989 年开始，卡耐基-梅隆大学参与研制了面向自由号空间站（SSF）的自行运动空间操作器 SM^2 （Self Mobile Space Manipulator）[12]。 SM^2 采用了一种模块化实时共享控制体系结构。机器人系统可以协调自主定位功能和来自遥操作的输入，从而实现在 SSF 上可靠地行走。任务是由许多可重用的子任务模块构成的，不同的高层任务如：遥操作任务、半自主任务，以及全自主任务，往往共享相同的子任务模块。通过状态机，可以方便地对高层任务进行编程，实现对子任务模块的共享。

美国 Sandia 国家实验室的智能系统与机器人研究中心开发了一个模块化的遥机器人软件结构 SMART（Sandia's Modular Architecture for Robotics and Teleoperation.）[13, 14]。SMART 实际上是一套遥机器人系统开发工具，它可以提供包括：双向遥操作、自主控制，以及共享控制在内的多种控制功能。SMART 由三部分组成：编辑器、任务监控器，以及实时控制引擎。在 SMART 中，一个面向特定任务的遥机器人系统由一系列的行为模块构成，每个行为模块是由若干基本组件（诸如：外部设备、机器人机构、传感器等）构成，通过无源网络模型描述。SMART 支持多种形式的人机共享。一种方式是把遥操作的输入叠加到轨迹规划器生成的基本轨迹上；另一方式是在虚拟环境中通过示教的方式，把人的规划能力传给机器人；第三种方式是人和机器人分时共享一个任务，将一个任务分解为若干子任务，人和机器人分别负责不同的子任务。SMART 已经被应用于突发事件响应运动操作器系统（Accident Reponse Mobile Manipulator System）[14]。

Stanford 大学的航空机器人实验室开发了一个基于对象的任务级分层控制系统 OBTLIC（Object-Based Task Level Control）[15]，在 OBTLIC 中，将整个系统分为三个层次：用户界面层、策略控制层，以及动态控制层。用户界面层负责世界模型的显示，接受操作员的动作；在策略控制层，通过有限状态机实现机器人自主运行所必需的逻

辑、决策功能；动态控制层通过高带宽的、基于传感器的反馈控制，实现机器人系统的精确、高速控制。可见，OBTLC 中的层次划分基本对应于 Saridis 的分层递阶结构 [16]。

在[17]中介绍了 UTAP 框架结构，UTAP 试图包容多种不同的传感、执行硬件，支持传感器反馈控制和遥操作控制模式。在[18]中提出了遥自主的概念，在遥自主概念中，试图涵盖由人和远地智能系统所构成的多种不同的交互模式。在[19]中，提出了一种基于事件的规划和控制理论，作为融合人的介入和机器智能的一个统一的框架。该框架通过动作参考参数（action reference parameter）的概念，将遥操作中人的介入和机器人的自主功能有机结合起来，从而同时支持多种控制模式，避免由于控制模式间切换造成的系统不稳定。在 Graves 的论文中[20, 21]提出了一种基于动作选择机制（Action Selection Mechanism）的遥自主体系结构，其特点在于，通过将来自不同渠道的控制输入进行融合，从而在一个统一的框架下支持多种不同的控制模式。在[22]中，针对大时延遥操作，提出了一种任务级监控方法。在[23]中，提出了一种基于 Internet 的监控式遥操作方法。在[24]中提出了一种通用的支持多个操作器的实时共享系统框架。以 McCarragher 领导的小组，研究了一种基于离散事件动态系统模型的共享控制系统框架[25-28]。在[29]中，研究了基于 Petri 网的共享控制操作模式的建模方法，涉及手控模式和自主模式的平稳转换、手控命令和自主控制命令的有效混合及系统任务建模和控制等关键技术。

从现有关于遥操作系统体系结构的研究中，可以得出如下两点结论：

- 单纯的双向遥操作和监控模式都具有一定的局限性，实际的遥操作系统需要集成包括双向遥操作、监控、共享控制在内的多种控制模式；
- 遥操作系统研究中的一个核心问题是如何实现人和机器的有机结合，本文称之为人机协调。

不同的控制模式代表了不同的人机协调形式，在同一系统中多种控制模式并存的事实说明大时延遥操作需要灵活多样的人机协调。然而，目前，不同的控制模式往往采用不同的模型，理论体系各不相同，难以在一个统一的框架下实现无缝集成，这就造成不同模式间的切换的平稳性的问题。另外，在体系结构参考模型中一般都对构成系统各部分的功能及其相互关系进行了说明。但是，缺乏定量的模型，因而，难以对整个系统的性能进行定量分析。上述问题，是目前没有解决的，需要大量的深入研究。

2.2 大时延遥操作技术

作为欧洲在空间机器人领域发展的第一步，ROTEX[9, 30]集成了多种当时最先进的遥操作技术，包括：立体图形预测显示、基于局部传感器反馈的共享控制，以及遥传感器编程（tele-sensor-programming）。在 ROTEX 中，克服大时延的一个思想是

针对不同的情况采取不同的控制模式，其中包括：在线遥操作（on-line teleoperation）、遥传感器编程，以及基于虚拟环境的任务级编程。在[31]中，T. B. Sheridan 对大时延遥操作技术进行了综述，详细介绍了预测显示技术、时延力反馈技术，以及基于波变换（Wave Transformation）的时延补偿技术，同时阐述了监控的基本思想。基于大量的实验结果，该文给出了如下结论：

- 1) 预测显示对于自由运动具有良好的效果；
- 2) 局部阻抗控制技术适合于机器人接触作业的情况；
- 3) 高层监控方式对于可以充分预测的任务是一种较好的选择，而且随着传感器和任务建模能力的提高，监控方式将发挥更大的优势。

该文还指出空间遥操作技术的发展趋向于尽量避免大时延的影响而不是从地面直接进行遥操作。

2.2.1 双向遥操作

从双向遥操作的角度，人们对大时延对系统稳定性和性能的影响以及相应的补偿方法进行了一系列的研究。在[42]中，通过简化的线性模型对控制回路中时延对稳定性和性能的影响进行了分析，并通过一个混合模型应用于对传统主从遥控系统的分析。在[43]中，分别从开环和闭环两个角度分析了时延对系统稳定性、性能，以及带宽的影响，并给出了相应的补偿方法：对于开环中的时延影响，通过预测显示进行补偿；对于闭环中的时延，通过对遥测数据和反馈信号的低通滤波进行补偿。在[44]中，对大时延的影响进行了定量的分析，提出了一个通用的多变量系统结构，其中包含了位置、速度信号在主从机器人之间四种不同的传输方式。该文论述了在克服大时延过程中，系统的稳定性和透明性（Transparency）是互为矛盾的。在[45]中，提出了一个基于共享顺应性控制（Shared compliance control）的大时延双向遥操作系统，该系统引入遥监视力反馈（Telemonitoring force feedback）作为一种新的人机运动耦合形式；基于广义阻抗控制，根据局部设计准则对主从手的动力学特性进行了修改；该文将人的动力学模型纳入系统模型中；在给定最大时延条件下，给出了折衷系统的鲁棒性和性能的优化准则。在[46]中，介绍了一个力反馈遥机器人共享控制系统——EC TELEMAN，该系统实现了两种形式的共享控制：基于双向遥操作的共享控制（Teleoperator-based shared control），在这种方式中，操作员可以将某些自由度的控制交由机器人自主完成，同时，遥控其余自由度的运动；第二种方式是基于任务级的共享控制，在遥机器人执行预编程的任务过程中，操作员可以随时介入，对局部路径进行修改。

2.2.2 预测显示

预测显示 (Predictive Display) 是大时延遥操作中的一项关键技术。在双向遥操作方式中, 预测显示是对大时延进行补偿的一种主要手段。在[9]中, 预测显示被认为是能够补偿大时延影响的唯一手段。预测显示的基本思想是基于系统模型, 根据当前状态和控制输入, 对系统状态进行预测, 并以图形的方式显示给操作员。预测显示分为两种类型: 一种是根据系统当前状态和时间导数, 通过泰勒级数进行外推; 另一种是, 建立系统运行的仿真模型, 在模型中融合系统的当前状态、导数, 以及控制输入, 然后让仿真系统以较实际过程快得多的速度运行[31]。在大时延遥操作中, 预测显示最初主要用于双向遥操作系统, 在发展过程中, 逐步被用于监控系统和共享控制系统中, 由于其重要作用, 得到了不断深入的研究。

NASA 的 JPL 实验室, 在大时延遥操作研究中, 开发了两种时延补偿方案, 其中之一就是预测显示[48, 49], 在其预测显示系统 Phantom Robot 中, 三维机器人模型可以对操作员的控制输入进行实时响应, 预测显示通过动态的三维图形和静态的任务背景图像的叠加实现。图形和图像叠加涉及的一个关键问题是匹配校准问题, 为此 JPL 实验室研究了基于交互方式的摄像机校准技术和目标定位技术。预测显示的有效性通过一项在 JPL 实验室和 Goddard 空间中心间进行的 ORU 替换地面仿真试验得到了成功验证。Conway 等人在其预测显示系统[18]中加入时间开关和位置开关, 实现了本地操作员的动作和遥机器人的控制的解耦。其优点是在双向遥操作框架下, 向操作员提供了不同形式的介入方式, 增加了遥操作系统的灵活性。

对于复杂的遥操作任务, 单纯的视觉反馈信息是不充分的, 往往需要包括视觉、力觉在内的多种反馈。在预测显示研究中, 人们试图把力觉以某种形式表现出来。在[50]中, 把力反馈信息转化为声音, 进行预测增强显示, 用于补偿遥加工 (Tele-Handling/Maching) 中的大时延。在[51]中, 基于传感信息分解的概念, 对视觉、力觉反馈信息进行同化, 从而在虚拟环境中融合了多种反馈信息。实验表明, 多种传感信息的融合有助于提高遥操作系统的性能。

预测显示依赖于操作环境和遥机器人的模型。在结构化环境下, 根据环境中物体的位置和几何模型可以实现三维图形显示。然而对于非结构化的、未知环境, 如何实现预测显示成为一个难题。解决的一个途径是基于图形和图像相结合的混合建模方法。加拿大多伦多大学的 ETC (Ergonomics in Teleoperation and Control) 研究中心研制了增强现实工具包 ARGOS[52, 53], 该工具包通过立体图形和立体图像叠加的方式, 允许操作员对非结构化环境进行部分建模, 从而提高遥操作系统的性能。在[54]中, 提出了一种完全基于图像的预测显示方法。该方法根据预先存储的图像, 建立相应的运动视觉函数, 通过视觉函数和图像插值的方法实现预测显示, 无需操作环境的模型和校正过程。然而, 这种方法的应用具有一定的局限, 只适合于小范围的精确运动,

且背景是静态的。在[55]中，通过纹理映射的方法，实现了部分未知环境的具有照片真实感的预测显示。图形图像相结合的混合建模方法，将是实现非结构化环境下预测显示的重要方法。

2.2.3 基于虚拟环境的人机接口

遥操作研究的目的是让操作器代替人完成各种灵巧的操作任务，这有赖于操作员和操作器之间充分的信息交换，其中的关键在于操作员能够获得充分的操作环境状态信息。为此，人们提出了遥现（Telepresence）的概念[56-58]。所谓遥现，是指向操作员提供有关操作现场的充分的而且高保真的传感信息，以至操作员具有身临操作现场的感觉。遥现可以视为遥操作中人机接口性能的一个标准。在大时延条件下，单纯依靠传感信息实现遥现是不可能的。虚拟现实技术通过计算机合成，向操作员提供沉浸感的虚拟环境显示，为大时延条件下实现遥现提供了一种可能。基于虚拟环境的人机接口在大时延遥操作中的应用得到了广泛的关注。JPL 实验室研制了一套遥机器人仿真系统 GLETS 作为操作员和遥操作机器人的接口系统[59]。在[47]中，让虚拟环境中机器人的动力学模型和真实机器人系统动力学模型保持一致，从而通过虚拟机器人的仿真运行实现了对真实机器人的有效控制。在[60]中，对头盔显示系统（Head Mounted Display, HMD）在遥操作中应用的效果进行了试验验证，试验结果表明，适当条件下，基于 HMD 的预测显示可以有效补偿时延的影响。

虚拟现实在遥操作中应用的前提是基于已知的环境和遥机器人的模型。在遥操作过程中，面临的一个问题是操作环境的实际状态和虚拟环境的状态间的偏差，如何消除二者间的偏差，保证虚拟环境仿真的准确性，是基于虚拟环境的遥操作系统中的一个关键问题。在[61]中，给出了一种基于传感器信息的虚拟环境误差校正策略，并通过实验验证了该方法在较小偏差情况下的有效性。在[62]中，提出了一个对虚拟环境的误差（包括几何误差和动力学误差）具有较强鲁棒性的遥操作系统。在该系统中，从手同时受位置和力两方面信号的控制，在从手和虚拟手之间引入了一种阻抗关系，控制从手的力信号强度正比于从手的轨迹位置误差，从而使得从手的控制对动力学模型的误差具有较强的鲁棒性；同时由于力控制的引入，使得系统对几何模型的误差具有较强的鲁棒性。

2.3 监控技术

2.3.1 监控的基本概念

从严格意义上，监控是指一个或多个操作员以间断方式对计算机进行编程，同时

连续接受计算机的反馈信息。计算机系统通过末端效应器和传感器与控制进程和任务环境形成闭合的控制回路。在不是很严格的意义上，监控泛指一个或多个操作员不断地对计算机系统编程并接受信息，计算机系统通过末端效应器和传感器与控制进程和任务环境相连。监控系统及其与手动系统的区别如图 2.1 所示：

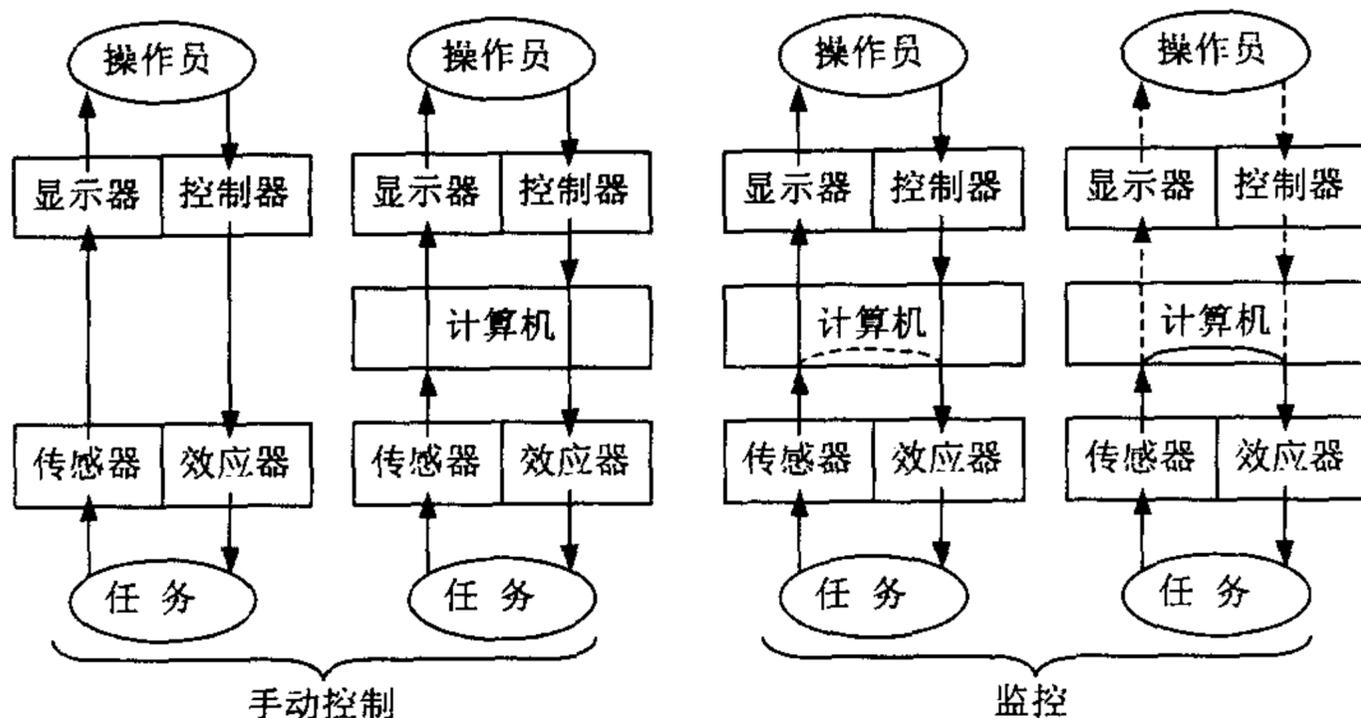


图 2.1 监控概念示意图

从图中可知，手动控制和监控的主要区别在于是否在操作员和任务环境之间存在计算机系统并形成局部闭合回路（在人与计算机之间或计算机与任务环境之间）。

监控用于大时延遥操作的基本思想是由人通过计算机接口对遥机器人进行编程，遥机器人对符号命令进行理解，并在与任务环境交互的过程中形成远端的局部闭合控制回路，从而将时间延迟排除于底层控制回路之外。在监控中，操作员除了对遥机器人编程外，还要不断监视遥机器人的运行，并以迭代方式更新与修改程序。从监控回路的角度，时延影响依然是存在的。要保证监控的有效性，需要三个方面的条件：1) 对给定的时延，子任务目标的“分量”相对于整个任务而言足够大，如果以平均执行时间度量“分量”，那么子任务的执行时间不应该过小；2) 远地的操作环境中的不可预知的因素的变化不应该太快，因为操作员编程的依据是对操作环境的状态的判断，如果缺乏准确的判断和预测，编程的合理性将难以保证；3) 遥机器人的性能是可靠的。遥机器人对符号命令的准确执行是监控的基础。

2.3.2 监控系统框架

在监控系统中，监控功能分为如下五个方面：

- 1) 规划。规划的含义包括对任务过程的理解、确定子任务目标，以及确定实现任务目标的途径三个方面。在对任务进程和环境理解的基础上，规划过程首先确定完成最终目标涉及的一系列子任务目标，这些子任务目标对遥机器人

而言必需是可理解的和可执行的；然后针对子任务目标确定任务执行途径，使得从初始状态可达到目标状态；

- 2) 示教或编程。编程是把规划的结果转化为计算机系统可以理解的指令的过程。具体的编程方法是由特定的机器人语言决定的。编程的结果是符号命令序列，遥机器人根据这些符号命令自主运行，直到程序被更新或操作员以手动控制的方式介入；
- 3) 监视。一旦符号命令程序交付遥机器人执行，操作员必须对其执行过程进行监视，通过各种反馈信息判断执行是否顺利进行，是否存在目标冲突或发生意外；
- 4) 介入。当遥机器人无法实现预定目标，或发生故障时，操作员需要通过更新、修改程序，或直接切换到手动控制方式，来干预执行过程。由于操作员往往是在遥机器人运行的动态过程中介入，因此保证介入过程中机器人运行的平稳是值得注意的问题。
- 5) 学习。监控器应该对任务执行过程中涉及的关键数据进行记录，用于后续的分析、模型数据库的更新等。

监控技术研究的一个基本问题是监控模型的建立。目前还无法建立一个统一的定量的监控模型。对于监控模型主要从三个角度对其进行定性描述：监控功能的嵌套循环结构、行为层次模型 (Levels of behavior)，以及生理行为定位 (Loci of biological activity)。监控的五项功能形成了三个嵌套的控制回路，如图 2.2 所示。

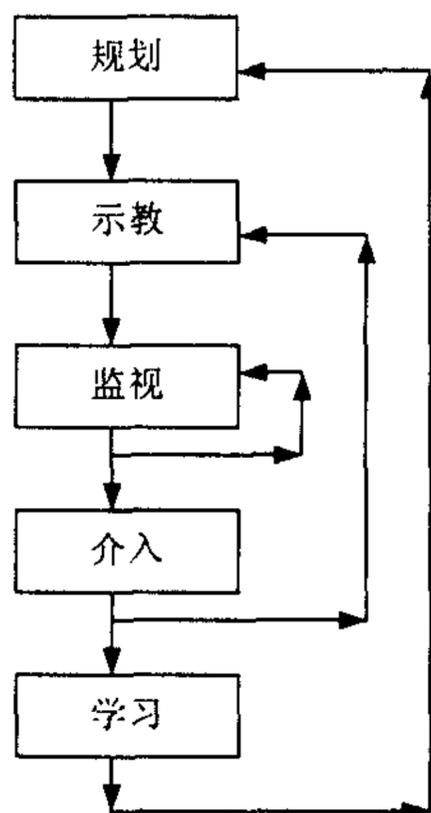


图 2.2 监控功能嵌套模型

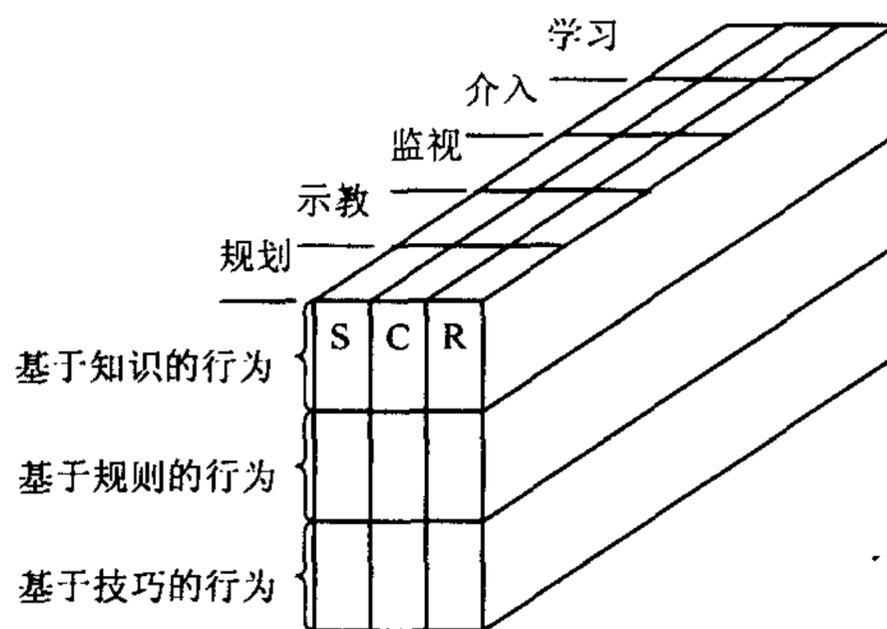


图 2.3 监控综合模型

在嵌套结构中，监视环节本身形成了一个回路。在操作员连续的监视过程中，随着兴趣点的转移，往往需要重新分配其注意力，或在不明显介入自动控制系统的情况

下进行微小的调整。在介入和示教之间形成了中间回路，这是因为，经过操作员对遥机器人控制的介入后，需要更新或修改原来的程序。在学习和规划之间形成了最外层的回路，因为对后续子任务的规划通常是基于对上一步的执行情况的了解。需要注意的是，对于三个不同的控制回路，其时间跨度也不相同。最里层的周期最短，最外层回路的时间跨度最大。不同的时间跨度意味着对操作员工作负荷的要求不同，显然，操作员的负荷主要集中在监视环节中。

按照 Rasmussen 的行为层次理论，可以把人的监控行为分为三个层次：基于技巧的行为 (skill-based behavior)、基于规则的行为 (rule-based behavior)，以及基于知识的行为 (knowledge-based behavior)。其中，监视一般对应于技巧层的行为；示教、介入，以及二者构成的回路中的其它部分，包括基于一定准则形成的计算机指令和任务协议等对应于规则层行为；而规划和学习作为高层的智能行为则属于知识层行为。

从生理行为定位的角度，可以把人的监控行为分为三种类型：感觉 (Sensory, S)、认知行为 (Cognitive activity, C)，以及反应功能 (Response or motor functions, R)。结合行为层次模型和生理定位模型,针对特定的任务，监控模型可以描述为如图 2.3 所示的三维矩阵形式。

2.3.3 注意力分配模型

监控系统是由人和计算机构成的混合系统。在给定系统功能要求的前提下，如何将各项功能在人和计算机之间进行分配，应该采取什么样的策略，依据什么样的准则，就是所谓人的注意力分配 (Human-Attention-Allocation) 问题。注意力分配问题是监控技术研究中的一类关键问题，具体涉及如下一些问题：

- 1) 人在给定任务中发挥哪些方面的作用。人拥有感觉、记忆、动作的能力，这些能力作为完成任务的资源是有限的，一般假设人在同一时间内只能做一件事。那么，当有多个任务请求时，需要将人的有限资源针对不同的任务进行分配；
- 2) 人应该承担哪些任务。从共享控制的角度，在同一时刻，人和计算机分别承担任务的不同方面。那么，对具体任务，人应该承担哪些方面的责任；
- 3) 完成任务所需的时间。把完成任务所需时间做为衡量监控系统性能的指标。监控系统的执行时间由两部分构成：人的编程时间和机器执行时间。研究表明，任务完成时间是由任务的复杂性决定的。
- 4) 如何确定交付机器自动执行的任务的尺度。在监控模式中，监控器向遥机器人发送符号指令，每条指令的执行过程对监控器而言是开环的。如果以执行时间度量指令的尺度，那么监控器需要确定在多长的时间范围内，遥机器人执行指令是可靠的，从而确定合理的指令时间跨度。指令时间过长，遥机器

人执行过程可能发生意外和故障；而指令时间过短，无疑将增加操作员的工作负担，因此需要确定优化的指令时间长度。一般指令时间长度决定于任务的复杂性、遥机器人的自主能力等多方面因素。

在[7]中，基于信号检测理论，提出了一个确定指令尺度的开环动作理论（Open-Loop Action Theory, OLAT）。OLAT给出了确定指令尺度问题的形式化描述，通过贝叶斯风险分析方法给出了确定指令尺度的概率准则。

2.4 遥编程技术

美国 Pennsylvania 大学的 Richard. Paul 在[64]中，针对遥操作中的大时延和有限带宽问题，提出了遥编程的概念。遥编程通过在操作员和遥机器人之间传递符号命令而非操作空间的控制信号，克服大时延的影响；通过虚拟现实技术使操作员获得即时的视觉和运动觉反馈，从而实现了操作员的连续操作。遥编程通过先进的人机接口技术实现了遥操作和监控两种策略的融合。

遥编程技术的有效性通过一系列的试验得到了验证。Richard. Paul 所在的 GRASP (General Robotics and Active Sensory Perception) 实验室与 NASA 的 JPL 的 TROPICS 实验室于 1994 年 8 月合作进行了陆地试验，取得了成功[65]。该试验表明，遥编程概念独立于具体的遥机器人控制结构和硬件系统，具有较好的适用性。此外，GRASP 与 WHOI (Woods Hole Oceanographic Institution) 的深水作业实验室合作进行了深水作业试验，取得了成功[66]。法国的 LAAS 实验室研制试验了 EDEN 遥编程系统[67]。在 ETSVII 卫星试验中，成功演示了在遥编程方式下执行空间作业任务。遥编程技术具有较广的应用前景。除了空间应用外，也可用于工厂自动化系统和深海作业。

遥编程涉及的关键技术主要有三个方面：人机交互界面、符号命令生成，以及虚拟环境误差校正。下面，从这三个方面简要介绍遥编程技术。

2.4.1 人机交互界面

虚拟环境首先要真实描述操作环境，不仅反映操作环境的静态几何模型，而且要进行碰撞检测，并在发生碰撞时，根据物理模型进行响应，反映操作环境的动态物理模型。除此之外，一种有效的人机交互手段是在虚拟环境中构造一些合成对象，辅助操作员对虚拟机器人进行控制，以便操作员更加方便、准确地实现其操作意图。合成对象是由计算机生成的与任务上下文相关的视觉和力觉信息线索 (Clues)。合成对象使界面不单单被动响应操作员的动作，而且可以主动引导操作员。常见的合成对象包括：虚拟墙、虚拟管道、合成支架 (Synthetic Fixture) 等。在[68]中，对合成支架进行了深入研究。合成对象的功能主要体现在以下几个方面：

- 1) 在操作过程中向操作员提供主动的视觉的和力觉信息，以辅助操作员更方便而准确地完成任务；
- 2) 帮助操作员进行误差诊断和误差校正；
- 3) 通过虚拟摄像机和放大镜等合成对象，对操作员的交互进行调整，可以使虚拟环境的响应更加自然，使界面更加适人化；
- 4) 一定程度上避免由于操作员的动作失准而产生歧义理解，从而增加系统对于模型误差的容错能力。

2.4.2 符号命令生成

具体的程序符号命令生成方法取决于机器人语言。机器人语言通常分为：动作级语言、对象级语言，和任务级语言三种。其中动作按照机器人是否同操作环境发生接触一般分为两类：一类是无接触状态下的自由运动；另一类是接触状态下的受约束运动。对于不同的语言，机器人命令的形式和所携带的信息也不同。对于动作级命令，一般包括以下几方面信息：a) 序号；b) 位置/力控制信息；c) 力和速度的约束信息；d) 成功条件[64]。在[67, 69]中，描述了任务级遥编程系统。对于任务级程序，一般包括以下信息：a) 附加的传感器信息；b) 任务上下文说明；c) 执行条件；d) 执行动作。不同的程序语言，取决于不同的任务描述方法。在[70]中，采用受控 Petri 网作为任务模型。在基于离散事件的模型描述中，任务执行过程表示为状态的变迁序列，在符号命令中包含了运动的下一个接触状态。该文通过实验表明，基于离散事件的模型描述，使系统较之时间参考的模型更具容错性。在[73-80]中，对符号命令的自动生成技术进行了研究。

2.4.3 误差校正

由于遥编程本质上是基于模型的操作，因此，在遥编程系统中，一个关键问题是如何克服由于操作环境的不确定性等因素带来的虚拟环境和实际环境的误差。在自由运动状态下，这种误差尚不甚明显，但在试图建立或保持某种接触状态时，这种误差就显得极为重要。误差主要来源于两个方面：一方面，遥机器人的运行相对于理想轨迹或预定目标必然存在偏差，这就造成了虚拟机器人和真实机器人之间的误差；另一方面误差来源于实际运动的操作对象和虚拟操作对象之间的误差。

在[71, 72]中，对误差校正进行了研究。在遥操作中，定义一个“意外情况”为超出预定序列以外的状态。与自主机器人不同的是，在遥编程中认为意外情况是无法预知并不可避免的，依靠操作员进行误差的诊断和校正而非寻求自主进行误差校正。遥编程的关键问题在于如何从意外情况中恢复执行而非尽量避免意外发生。在遥编程

系统中，一般采用单线程的思路进行误差校正，采取前向误差校正策略，试图在当前状态下继续任务。在意外状态下，根据操作员的动作生成误差校正命令，为此，操作员界面必须为每一个意外情况提供上下文和状态信息。为了有效地进行误差诊断和恢复，需要三个方面的条件：

- 1) 操作员界面须提供多种相互交叠的反馈信息；
- 2) 控制器必须能够判断出意外的发生并请求帮助；
- 3) 操作员必须能够对任务进行有效控制，能够介入误差校正。

2.5 碰撞检测技术

2.5.1 碰撞检测概述

虚拟环境要逼真地模拟真实的物理世界，不仅仅需要在几何形状、光照效果等方面进行逼真显示，还需要反映物体运动的物理规律，其中一个很重要的方面就是物体之间的相互作用。在真实的世界中，当两个物体相互接触时，它们一般不会穿透对方，而是在相互作用力下，改变各自的运动状态。在虚拟环境中，要反映物体间的相互作用，就要在赋予对象动力学特性的同时，检测出对象之间的相互作用关系，并根据相互作用关系控制对象的运动。如何反映物体间发生接触（或碰撞）时的相互作用，就是我们所说的碰撞问题。碰撞问题牵涉到碰撞检测和碰撞响应两部分内容。碰撞响应研究的是在物体发生碰撞后，如何运动的问题，基本属于动力学领域的研究内容。本文主要针对虚拟环境中物体间的碰撞检测问题进行研究。

碰撞检测（Collision Detection）又称为干涉检测（Interference Detection）。碰撞检测最直接的一个目的是判断两个物体是否发生接触（碰撞），而实际的应用往往还需要碰撞检测回答物体间接触的具体部位、物体间的距离，以及物体间何时发生碰撞等问题。碰撞检测的核心是空间干涉问题。对于空间干涉的研究，最早源于计算几何和机器人学领域[81-83]。在计算几何领域，主要是针对静态环境，判断两个几何形体是否相交，研究的重点是对复杂的几何形体进行准确的相交计算。计算几何领域的研究成果为碰撞检测在其它领域中的应用提供了基础。在机器人学领域，碰撞检测主要用于路径规划，给定两个物体及其预定的运动轨迹，碰撞检测的任务是判断物体沿运动轨迹运动的过程中是否发生碰撞。

目前，碰撞检测技术已经成为 CAD/CAM、机器人运动规划、计算机动画、虚拟环境等应用中的关键技术。在具体的面向机器人遥操作的虚拟环境中，在以下几方面需要进行碰撞检测：

- 1) 机器人编程。在操作员面向虚拟环境操纵虚拟机器人进行机器人编程的过程中，操作员需要知道机器人相对于操作对象的接触关系，以判断机器人是否

已经运动到期望的操作位置。基于碰撞检测，以视觉或力觉反馈的形式，向操作员报告机器人与操作对象的接触状态，将有助于操作员准确控制机器人的姿态，形成合理的控制指令。

- 2) 运动轨迹检验。对于规划好的运动指令，在实际交付机器人执行之前，需要检验在运动过程中，是否发生机器人与操作环境的意外碰撞。通过碰撞检测，可以帮助操作员对机器人的运动轨迹进行检验。
- 3) 任务预显示。在任务预显示过程中，除了要显示自由运动轨迹之外，更重要的是对机器人与操作对象发生接触的操作过程进行仿真显示，这涉及到复杂的动力学计算。准确地计算机器人与操作对象的接触状态，是计算相互作用力、进行动力学仿真的前提。

2.5.2 碰撞检测技术分类

有许多碰撞检测方法，包括：分层表示、几何推理、代数描述、空间分割、解析方法，以及优化方法等。具体采用什么方法由三方面因素决定：模型表示，检测问题类型，和仿真环境[86]：

1) 基于模型表示分类

在 CAD/CAM 和三维图形学中，有许多模型表示方法。大体上分为两类：多边形模型和非多边形模型。多边形模型分为结构化多边形模型和非结构化的多边形面片集合 (Polygon soup)；非多边形模型包括：实体构造模型、隐含曲面模型和参数曲面模型。

多边形模型是计算机图形学和仿真建模中最通用的一种模型，其中非结构化的多边形面片集合是用得最多的表示方法。在具体应用中，针对不同的模型表示有不同的碰撞检测方法。基于包围体的分层结构描述是针对多边形模型的碰撞检测的最重要的一种方法[102, 103, 105, 107]。

2) 基于检测问题类型

不同的应用需要解决不同类型的检测问题，大致有三种检测类型：

- 相交检测，检测物体间是否发生碰撞，以及碰撞发生的具体部位。主要用于计算机动画和基于物理的仿真；
- 距离检测，检测物体间的距离，当物体分离时，检测其间的欧几里得距离，当物体相互贯穿时，检测最小的分离距离。主要用于在机器人路径规划和动力学仿真中，计算相交时的作用力和惩罚函数；
- 碰撞预测，根据物体的位置和运动，预测物体间下一次可能的碰撞，又称作预估到达时间 (ETA, Estimated Time of Arrival)。碰撞预测使得可以在仿真中调节仿真步长。

3) 基于仿真环境分类

在实际应用中, 往往根据具体的仿真环境的特征选择合适的碰撞检测方法:

- 根据物体的多少: 两两相交问题 (Pair processing) 和多体相交问题 (NBody processing);
- 根据物体的运动: 静态碰撞检测和动态碰撞检测;
- 根据物体的形状: 刚体碰撞和变形体碰撞。

2.5.3 基本碰撞检测方法

在[87]中, 将碰撞检测方法主要分为四类: 静态碰撞检测、基于 4D 时空域 (Space-time) 求交的动态碰撞检测、基于扫描体 (Swept volume) 求交的动态碰撞检测, 以及基于离散采样的碰撞检测。下面简要介绍几种基本的碰撞检测方法。

1) 静态碰撞检测

静态碰撞检测是众多动态碰撞检测方法的基础。对于多边形模型, 多面体的凸性是影响静态碰撞检测方法的关键因素, 根据几何体的凸性的不同, 静态碰撞检测可以概括为三类方法: a) 凸多面体。两个凸多面体的相交计算, 可以通过线性编程技术 (linear programming), 以线性的时间复杂度实现[88]。经过适当的预处理, 两个凸多面体之间的相交计算可以以 $O(\log n \log m)$ 的复杂度完成, 其中 n 和 m 是两个凸多面体的顶点数[100, 109]; b) 具有凸多边形表面的多面体。两个多面体表面相交计算可以归结为表面上的边与另一个表面的相交计算。如果多面体的表面由凸多边形构成, 那么求交计算将可以大大简化; c) 一般多面体。针对一般多面体的相交检测方法主要有两类: 一类是先将多面体分解为若干凸多面体, 再对凸多面体进行相交检测; 另一类是直接对多面体进行检测[90, 91]。

2) 基于 4D 时空域 (Space-time) 求交的动态碰撞检测

这类方法的基本思想是把三维空间中连续运动物体的碰撞检测, 转化为 4D 时空域内压模体 (extruded volume) 的相交检测。所谓压模体是时空域内的点集, 描述了物体沿轨迹运动所占据的空间。两个连续运动的物体相交当且仅当它们的压模体相交。根据压模体求交方法主要针对实体构造模型 (CSG)。因为压模 (extrusion) 操作对集合的交、并、差运算是符合分配率的, 这一特性保证了物体及其压模体能够通过相同的布尔运算分别由构成物体的原型及原型的压模体得到。基于压模体的相交检测的关键在于两个方面: 一是压模体的计算, 对于复杂的运动轨迹, 压模体本身的计算相当复杂。在实际应用中, 往往假设物体作分段线性平移运动; 另一方面是压模体间的相交计算。为了提高压模体相交计算的效率, 一般采用包围体来近似压模体, 其中比较有代表性的是 Cameron 提出的 “S-Bounds” [89]。

3) 基于扫描体 (Swept volume) 求交的动态碰撞检测

所谓扫描体是指，在给定时间段内，运动的物体所占据的三维空间的点集。如果两个物体的扫描体不相交，那么这两个物体肯定不想交。然而，这只是判断两个物体不相交的充分而非必要条件，即扫描体相交而物体未必相交。为了使扫描体的相交计算成为判断物体之间相交的充分必要条件，需要计算两个物体间的相对运动扫描体，即假定其中一个物体是固定的，根据两者间的相对运动计算另一个物体的扫描体。扫描体的计算非常复杂，在实际应用中，往往被限制于处理简单的几何形体和简单的运动轨迹。为了简化扫描体的计算，一般采用分段凸壳逼近的方法，即首先对物体的全局运动产生一个扫描体的凸逼近，如果扫描体的凸逼近相交，那么将扫描体分段，再逐段计算凸逼近。这样的过程迭代进行，可以证明迭代产生的凸逼近将收敛于真实的扫描体。

4) 基于离散采样的碰撞检测

基于 4D 时空域的相交计算和基于 3D 扫描体的相交计算，是两种准确地进行连续运动的物体间的碰撞检测的思路。但是获得较高准确性的代价是 4D 压模体和 3D 扫描体的计算非常复杂，以致于它们在实际中只适用于比较简单的情况。为了适用于更加复杂的几何形体和运动轨迹的碰撞检测，一种简单的思路是通过在时间域上采样，把连续运动中的碰撞检测转化为离散的多个静态碰撞检测问题。这种思路的关键是采样间隔的确定，如果采样间隔过粗，可能会漏检，反之，采样间隔过密，将增加计算复杂性。因此，自适应采样是其中的关键技术[90]。

2.6 小结

本章对大时延遥操作技术和碰撞检测技术进行了综述。介绍了遥操作系统的体系结构，从双向遥操作和监控两个方面对大时延遥操作技术进行了介绍，介绍了遥编程技术，以及碰撞检测技术。

第三章 基于 Petri 网的大时延遥操作系统框架研究

3.1 概述

在这一章, 本文从建立灵活、合理的人机协调机制的需要出发, 提出了一个基于 Petri 网的监控式大时延遥操作系统框架。系统框架的特点在于提供了多层次介入机制和系统各部分功能的一致性系统模型, 其主要功能包括: 任务协调和任务监控。围绕系统框架的基本思想, 本文深入研究了基于 Petri 网的任务建模和任务协调两个方面。在任务建模方面, 研究了基于 Petri 网的简单任务建模方法以及基于分层 Petri 网描述的复杂任务建模方法。其中, 针对自由运动, 提出了一个基于路径关键点提取的动作级任务建模方法; 基于子任务的 Petri 网系统模型, 本文提出了一种通过变迁细化来建立任务的分层 Petri 网描述的方法, 给出了变迁细化的条件, 并通过定理证明变迁细化的有效性。在任务协调方面, 本文给出了一种根据任务空间的谓词描述获得相应的谓词/变迁网模型的方法, 并基于谓词/变迁网模型, 提出了一种任务协调方法, 详细讨论了其中的状态识别、动作识别等问题。

本章的第一节, 简要介绍了 Petri 网的背景及其在机器人领域的应用, 介绍了 Petri 网的基本概念; 在第二节, 阐述了基于 Petri 网的大时延遥操作系统框架的基本思想, 并给出了系统框架的概念描述; 第三节论述了基于 Petri 网的基本任务建模方法, 给出了路径关键点提取方法; 第四节论述了复杂任务建模方法, 首先介绍了分层 Petri 网的基本概念, 然后给出了具体的变迁细化方法; 在第五节, 介绍了基于 Petri 网的任务协调方法。最后是结论。

3.1.1 Petri 网概述

1962 年, 在 C. A. Petri 的博士论文中提出了 Petri 网的概念。随后, Petri 网在学术界和工业界得到了广泛关注, 其相关理论和应用的研究不断深入。在理论上, 其抽象、描述能力不断发展和增强, 从基本的条件/事件 (C/E) 网, 经过位置/变迁 (P/T) 网, 直到高级网 (包括: 谓词/变迁网和着色网); 从没有参数的网, 发展到赋时 Petri 网 (Timed Petri Nets) 和随机 Petri 网。发展至今, Petri 网已成为研究离散事件动态系统最有力的一种工具。在应用方面, 除了在性能评价和网络协议领域取得了成功外, 其应用领域已经扩展到软件工程、工业制造系统, 以及机器人等众多领域。

Petri 网以网络的形式描述系统中元素间的关系, 不仅提供了一种直观的图形模型表示方法, 同时也是严格定义的数学对象。Petri 网的优点在于善于描述系统行为中的

并发和冲突现象，同时提供了系统的分析方法，可以满足系统的静态结构分析和动态行为分析的需要。

在机器人领域，Petri 网主要被用于对机器人装配任务的建模和任务规划。在[139]中针对机器人装配任务，根据工件与环境的接触状态，建立了任务的 Petri 网模型，提出了包括过程建模、任务合成、运动控制以及轨迹规划在内的一套方法。[140]提出了一个通过与/或网描述机器人装配过程中的几何状态关系，并将其转化为 Petri 网，用于规划装配序列。在[40]中，以建立监控系统为目标，对机器人的搬运和装配作业进行了 Petri 网建模与控制的研究。在[141]中，利用 Petri 网理论对装配过程进行建模，并阐述了最优装配路径评判方法和生成方法。

Petri 网也被应用于智能机器人和遥机器人的研究中。在[142]中，论述了基于 Petri 网的智能机器人协调理论，并应用于运动机器人系统。其中将 Petri 网作为智能系统协调级模型，实现了包括分发器 (dispatcher)、路径规划、避障及轨迹控制，以及视觉系统等各部分协调器在内的功能的一致描述。在[143]中，不同类型的 Petri 网被用来进行定性和定量的系统性能分析。在[144]中，利用 Petri 网作为一致性模型，描述运动机器人的地图、任务过程，以及定性状态。此外，通过 Petri 网的逐步细化，实现了机器人基于有限前视策略 (Limited Lookahead Polices, LLP) 的监控。在[145]中，利用受控 Petri 网 (Controlled Petri Nets) 对遥机器人的 part-mating 任务进行建模，并在此基础上进行机器人的任务规划。在[146]中，探讨了应用 Petri 网来描述人的精神模型的方法。在[147]中，讨论了用谓词/事件网来建立智能机器人任务规划模型，还讨论了用 Petri 网为智能控制的协调级建立模型，并实现网络分解与综合。

3.1.2 Petri 网基本概念

[定义 3.1 Petri 网] 三元组 $N = (S, T; F)$ 称为 Petri 网，当且仅当下列条件满足：

- i) $S \cup T \neq \emptyset$ ，且有 $S \cap T = \emptyset$ ；
- ii) $F \subseteq (S \times T) \cup (T \times S)$ ；
- iii) $dom(F) \cup ran(F) = S \cup T$ ，其中：

$$dom(F) = \{x \in S \cup T \mid \exists y \in S \cup T, (x, y) \in F\},$$

$$ran(F) = \{y \in S \cup T \mid \exists x \in S \cup T, (x, y) \in F\}.$$

分别为 F 的定义域和值域。 S 和 T 分别称为 N 的位置 (place) 集和变迁 (transition) 集， F 为流关系 (flow relation)。 $X = S \cup T$ 称为 N 的元素集。

[定义 3.2 前置集、后置集] 设网 $N = (S, T; F)$ ， $x, y \in X$ ，

- i) ${}^*x = \{y \in X \mid (y, x) \in F\}$ 称为 x 的前置集或输入集；
- ii) $x^* = \{y \in X \mid (x, y) \in F\}$ 称为 x 的后置集或输出集；

[定义 3.3 Petri 网系统] 一个 Petri 网系统, 简称 P/T 系统, 表示为一个六元组 $\Sigma = (S, T, F, K, W, M_0)$, 其中:

- i) $N = (S, T; F)$ 是有向网;
- ii) $K : S \rightarrow \{1, 2, \dots\} \cup \{\omega\}$, 为 N 的容量函数, 其中 ω 表示无穷;
- iii) $W : F \rightarrow \{1, 2, \dots\}$ 为 N 上的权函数;
- iv) $M_0 : S \rightarrow \{0, 1, 2, \dots\}$ 为初始标识, 满足: $\forall s \in S : M_0(s) \leq K(s)$ 。

[定义 3.4 实施规则] 令 $\Sigma = (S, T, F, K, W, M_0)$ 是一个 P/T 系统。

- i) 函数 $M : S \rightarrow \{1, 2, \dots\}$ 叫做 Σ 的标识, 如果, $\forall s \in S : M(s) \leq K(s)$;
- ii) 一个变迁 $t \in T$ 在 M 下是可实施的, 当且仅当:

$$\forall s \in S : W(s, t) \leq M(s) \leq K(s) - W(t, s)$$
- iii) 如果 $t \in T$ 在 M 下是可实施的, 那么 t 可以实施并产生一个新的标识 M' , M' 可由下列方程给出:

$$\forall s \in S, M'(s) = M(s) - W(s, t) + W(t, s)$$

3.2 基于 Petri 网的大时延遥操作系统框架

3.2.1 系统框架的基本思想

为了最终实现灵活、合理的人机协调的目标, 系统框架需要解决如下两个方面的问题:

1) 提供操作员多层次介入机制

所谓多层次介入, 是对智能的分布层次而言。在分层递阶的智能系统体系结构中, 系统的功能按照智能水平的不同分布在多个层次上。多层次介入机制, 使得操作员可以根据操作任务和操作环境的复杂性, 以及机器人的自主能力, 适当调节其功能, 从而实现灵活的人机协调。

2) 支持系统各部分功能的一致性系统模型

合理的人机协调最终需要根据量化准则来实现功能的优化分配。为此, 需要解决包括: 建立人的精神模型、度量操作任务的复杂性, 以及度量机器人的操作能力等等方面的问题。就目前水平而言, 尚无法建立包括: 人、机器人、操作任务, 以及操作环境在内的定量的系统模型。然而, 要最终建立定量的系统模型, 首先要实现对系统各部分功能的一致性描述。

目前, 在大时延遥操作系统中, 多层次介入主要体现为监控模式和直接控制模式之间的切换。根据 Saridis 的智能机理论[16], 监控模式和直接控制模式分别实现了操作员在决策层和执行层的介入。然而, 对整个智能系统而言, 这两种模式仅仅代表了

两种极端的情况，不足以满足灵活的人机协调的需求。而且，在系统实现过程中，这两种控制模式分别对应由不同的模型表示的相互独立的控制模块，需要通过切换的方式实现多层次介入，而切换给保持控制的平稳性带来了困难。在共享控制方案中，目前主要解决的是在同一层次的人机协调问题（在执行层或在决策层）。在 Albus 的 NASREM 遥机器人系统参考模型[10]中，操作员可以介入系统的不同层次，实现不同的功能，体现了多层次介入的概念。然而，该模型并没有从人机协调的角度，对多层次介入进行系统地规范。该模型虽然实现了智能的逐层分解，但没有提供针对不同功能模块（包括：传感信息处理、世界建模、任务分解）的一致性模型描述。目前，大时延遥操作系统中各部分功能缺少一致的系统模型。综上所述，多层次介入和一致的系统模型是大时延遥操作中亟待解决的问题。

本文基于监控思想，建立大时延遥操作系统框架。在监控系统中，人的介入主要体现在两个方面：任务规划和任务监控。因此，多层次介入包括多层次任务规划和多层次任务监控两个方面。系统框架在功能上主要解决了任务规划和任务监控两个问题。系统框架的功能概念图如图 3.1 所示：

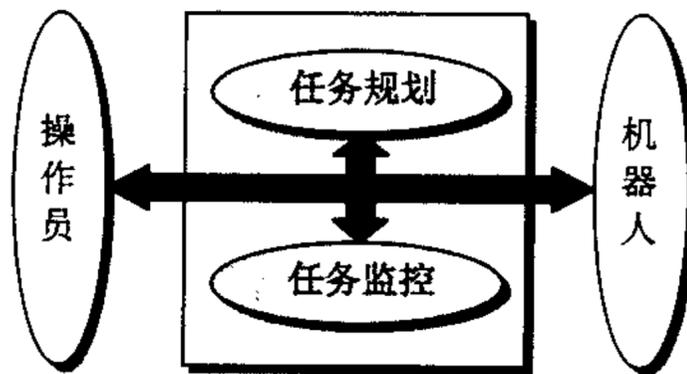


图 3.1 系统框架功能概念图

在逻辑层次，将大时延遥操作系统视为离散事件动态系统，本文采用 Petri 网形成大时延遥操作的系统模型，基于 Petri 网模型实现任务描述、规划和任务监控功能。

基于 Petri 网形成系统模型，主要因为 Petri 网提供了一种一致的有效的描述方法，使得在一致的模型基础上进行任务规划、系统性能分析，以及监控系统设计成为可能。具体表现在以下几个方面：

- 1) Petri 网可以以一种一致的形式描述遥操作系统不同逻辑层次的任务模型。Petri 网的模块式、分层建模方法，使其能够胜任对复杂的、分层结构的遥操作任务的描述。分层任务描述是实现多层次任务规划和任务监控的基础；
- 2) 通过对基本 Petri 网模型进行适当扩展，比如，利用计时 Petri 网、随机高级 Petri 网等扩展模型及相应的分析方法，可以对系统性能进行定量的分析；
- 3) 基于 Petri 网模型，已经发展了一套监控系统的分析和综合方法，可以用于遥操作任务监控系统的分析和设计。

在本文的绪论部分 1.3.2 中，论述了大时延遥操作面临的一个困难是对于连续操作的需求与间断控制方式间的矛盾。在系统框架中，通过基于 Petri 网的任务协调，

将操作员的连续规划动作转化为机器人的符号命令，从而实现任务规划功能。

所以，基于 Petri 网模型，可以实现监控式遥操作系统中包括：任务建模，任务协调，任务监控在内的各项功能。Petri 网与遥操作系统的关系如下图所示：

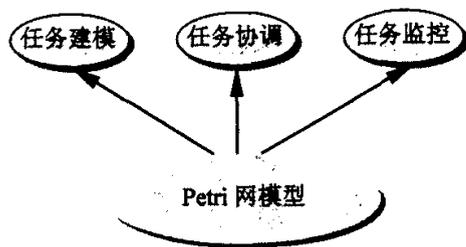


图 3.2 Petri 网在遥操作中应用示意图

3.2.2 系统框架描述

在系统框架中，本文根据机器人可接受的符号命令的抽象程度，将机器人的运动分为两个层次：动作级和任务级。关于动作和任务的具体含义定义如下：

[定义 3.5 动作] 动作是构成机器人运动行为的最基本元素，对应于机器人动作级语言中的动作命令，是机器人能够执行的最底层的符号命令。动作不可分解。对特定的机器人系统，动作集合表示为 $A_0 = \{a_1, a_2, \dots, a_n\}$ ，是有限集合。

[定义 3.6 任务] 任务泛指机器人的复杂运动行为，对应任务级语言中的符号命令。任务可以分解为子任务和动作的序列。任务具有如下形式：

- 1) 令 a 是一个动作，则 a 是一个任务；
- 2) 令 a 是一个动作， t_1 是一个任务，则 $t = at_1$ （或 $t = t_1a$ ）是一个任务；
- 3) 令 t_1, t_2 是任意两个任务，则 $t = t_1t_2$ 是一个任务。

根据上面的定义可知，通过递归定义，任务可以是非常复杂的。

对应机器人运动的两个层次，本文将系统框架中的元素（任务模型、任务协调、任务监控）分为两个层次。每个层次元素的定义如下：

[定义 3.7]

- 1) 动作级模型，描述机器人动作同状态间的转换关系。
- 2) 任务级模型，描述机器人任务同状态间的转换关系。
- 3) 动作级协调，将操作员的连续规划动作转化为机器人的动作序列。
- 4) 任务级协调，将操作员的连续规划动作转化为机器人的任务序列。
- 5) 动作级监控，以动作为对象进行监控。
- 6) 任务级监控，以任务为对象进行监控。

需要说明的是，上述对任务模型的划分仅是针对模型的概念划分，实际的

系统模型不限于两个层次,除了一层动作级模型外,可以有多层任务级模型。基于 Petri 网的系统框架结构示意图如下:

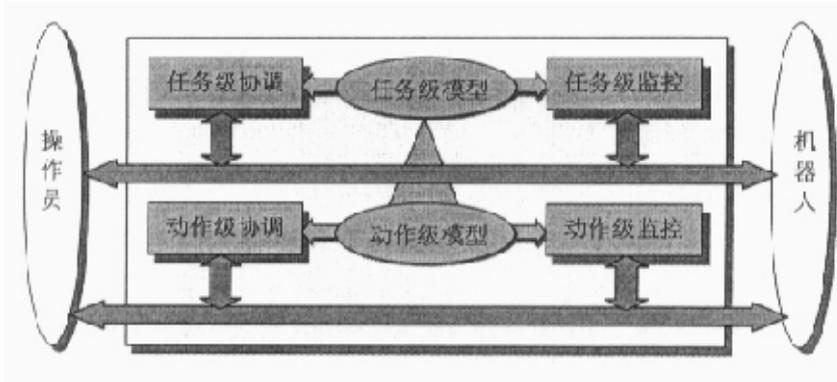


图 3.3 基于 Petri 网的系统框架示意图

3.3 基于 Petri 网的基本任务建模方法

3.3.1 任务建模基本概念

在应用 Petri 网描述机器人任务的过程中,主要需要解决三方面的问题: a) 状态集的确定,需要将机器人的状态空间抽象为有限的离散状态集合; b) 变迁集的确定,变迁代表导致机器人操作状态发生变化的事件,通常对应于机器人的运动行为; c) 状态与变迁的关系。本文所说的任务通常是指对包括操作状态、运动行为,以及二者相互关系的综合描述。围绕这三个问题,本文论述了基于 Petri 网任务建模的基本原理,给出了确定状态集和状态——变迁关系的基本原则和方法,并针对自由运动给出了具体的动作级任务建模方法。

在遥机器人系统任务的 Petri 网模型中,位置对应系统的状态,任务中的状态通常分为如下几类:

- 自由运动状态。机器人没有同操作环境发生接触、碰撞的自由运动过程中的状态,主要描述机器人的位置、姿态;
- 接触状态。机器人同操作环境发生接触的操作过程中的状态,主要描述接触关系,用于决定机器人的运动方向;
- 运动控制状态。描述机器人当前的控制方法和控制模式,比如关节空间/操作空间运动控制,位置控制模式/力控制模式;
- 约束状态。机器人同操作环境接触过程中,往往要对其施加速度约束、受力约束;
- 操作状态。泛指机器人执行任务过程中涉及的定性的逻辑层次状态,如:机

械手是空的、插件在插孔中等。操作状态可以由底层的离散状态或较低层的操作状态组合而成。

变迁一般对应机器人的运动行为。这里的运动行为对应机器人可执行的符号命令。运动行为大体可以分为两个层次：动作和任务(见定义 3.5、3.6)。对特定机器人系统，能够执行的预定义动作与任务为有限元集合，记为 A 。

将遥操作任务表示为如下形式：

$$\Gamma = (PN, M_0, A_T)$$

其中， $PN = (S, T, F)$ 是 Petri 网， M_0 是任务的初始状态标识， $A_T : T \rightarrow A$ ，是变迁到预定义动作或任务集合的映射，称之为变迁映射。

建立任务的 Petri 网模型的首要问题是确定任务的离散状态集合。对应于不同的抽象程度，可以确定不同层次的离散状态集合，本文把在特定抽象层次上确定离散状态集合的过程称之为状态抽象。设任务在不同层次的状态描述为 $\Omega = \bigcup_{i=1} S_i$ ，其中 S_i 为层次 i 上的状态集合。状态抽象需要满足以下原则：

- 1) $\forall i, S_i$ 是完备的。对逻辑层次 i ， S_i 包括了任务的全部状态描述；
- 2) $\forall s_1, s_2 \in S, (\exists t \in T, (s_1 \in t^* \wedge s_2 \in t^*) \vee (s_1 \in t^* \wedge s_2 \in t^*)) \Rightarrow (\exists a \in A, a = A_T(t))$ 两个状态间的变迁必须能够通过机器人可执行的动作或任务来实现。

按照逻辑层次深入的方向不同，状态抽象可以分为两类方法：自底向上的概括方法和自顶至下的细化方法。概括方法可以理解为对已有状态描述进行抽样或逻辑抽象的状态精简过程，一种典型的概括方法是对连续状态空间进行离散化而得到离散状态；细化方法则对应通常的问题分解的思路，把复杂问题逐步分解为相对简单的问题。

在状态抽象的基础上，Petri 网建模的主要问题集中于确定变迁集、变迁映射，以及状态——变迁关系。状态——变迁关系一般由以下几方面关系决定：

- 顺序关系。任务描述既是任务规划的依据，往往也反映任务规划的结果。对于一条规划好的路径，其中涉及的状态存在时间或逻辑上的前后顺序关系，这种顺序关系决定了状态之间以及状态及变迁间的关系；
- 因果条件关系。动作执行时附加的约束条件、任务的前提条件和结果条件，是确定状态——变迁关系的直接来源。

一般情况下，变迁直接对应具体的动作或任务，但是，变迁映射未必是一一对应关系。多个变迁可能对应同一个动作（任务），由于动作的初始条件不同，其导致的状态变化也不同。从状态抽象的原则可知，变迁映射满足如下的约束：

$$\forall t \in T, \exists a \in A, s.t. A_T(t) = a$$

3.3.2 动作级任务建模

在机器人的动作级建模中，需要确定机器人的动作及相应的操作状态之间的关系。其中，操作状态描述机器人的运动状态、约束条件，以及控制状态。根据机器人同操作环境是否发生接触，可以将运动状态分为两类：自由运动状态和接触运动状态。基于接触状态的任务建模在机器人装配作业中得到了大量的研究，在此不再赘述。本文主要对自由运动状态下的状态抽象和任务建模进行研究。

在自由运动过程中，机器人的动作通常由如下参数描述：起始点、目标点，以及运动速度。一个动作的执行结果，在三维操作空间中，一般是在起始点/目标点之间，或沿着固定方向的一段直线轨迹。在动作与动作之间，机器人处于静止状态。因此，动作级状态抽象将围绕如何确定运动轨迹及速度变化的关键点展开。本文通过确定运动轨迹的关键点来进行状态抽象。

· 如果将一个任务看作自由运动空间中一段连续运动轨迹，则在状态抽象的基础上，任务描述为由离散状态和动作组成的一个序列，其基本形式如下： $Task = s_0 a_1 s_1 a_2 \dots a_n s_n$ 。因此，任务 Petri 网模型的基本结构如图 3.4 所示。如果对动作附加约束条件，则在对应的变迁节点的前缀集中添加一个约束状态节点，如图 3.5

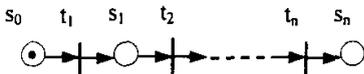


图 3.4 动作级任务基本结构图

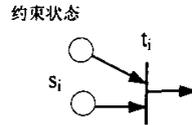


图 3.5 有约束变迁节点

所示。

3.3.3 路径关键点提取方法

三维空间中连续曲线的关键点提取问题描述如下：对三维操作空间中一给定的连续运动轨迹 $\{x = x(t), y = y(t), z = z(t)\}$ ，确定关键点集合：

$$P = \{(x_i, y_i, z_i, t_i) \mid x_i = x(t_i), y_i = y(t_i), z_i = z(t_i), i = 1, 2, \dots, n\}$$

其中， x, y, z, t 分别代表 X、Y、Z 坐标轴，以及时间轴上的坐标。实际应用中，曲线往往是以数字形式给出的，数字曲线为三维空间点的一个序列 $C = p_1 p_2 \dots p_n$ ， $p_i(x_i, y_i, z_i)$ 是三维空间中的点，这时，关键点集合 P 是数字曲线 C 的一个子集。如不加说明，本文后面所说曲线指数字曲线。

对于关键点，并没有明确的定义，但一般来说，所谓关键点有两种含义：一方面，关键点能反映曲线的几何特征，比如角点、拐点；另一方面，关键点可以看作用折线段逼近曲线时所必需的线段端点。根据关键点含义的不同，关键点提取大体有两种方法：一种是关键点检测方法，利用关键点的特征（比如曲率），根据特征判据直接检测关键点，一般关键点是曲率的局部极大值点；另一种方法是分段线性化方法，这种

方法试图用一系列的直线段逼近曲线,逼近过程由误差限控制,线性化后各直线段的端点可以作为曲线的关键点。研究表明,上述两种方法互有优缺点,存在一定的互补性。曲线关键点提取问题,在数字图像理解领域得到了深入的研究[156-158]。

本文在作者提出的多尺度曲线分段线性化算法[159]的基础上,给出一种三维空间连续曲线的关键点提取方法。在多尺度曲线分段线性化算法中,把数字曲线视为由若干圆弧连接而成,圆弧的连接点往往是曲线的关键点。每段圆弧的半径不同,对应的曲率也不同,因而表现出多种曲率尺度。对不同曲率的圆弧,按照一个统一的误差判据进行分段线性化,从而实现整个曲线的分段线性化。算法大致分为两步:首先根据曲率特征对曲线进行分割,分割后的每一段曲线上的点的曲率特征相近,可以近似看作一段圆弧;然后,对每一段曲线进行分段线性化,分段线性化的误差限根据曲率特征确定。这种方法把关键点检测和分段线性化的思想融合在一起,因而表现出较好的效果。该算法具体到三维空间曲线关键点提取问题中,有以下问题需要解决:

- 1) 曲率的计算。原算法中的曲率是针对二维空间的数字曲线的,在三维空间中曲率需重新定义;
- 2) 线性化逼近误差的计算。原算法中的误差是通过计算曲线同逼近直线段围成的闭合区域的面积得到,在三维空间中误差需要重新定义。

下面将围绕上述问题进行论述:

1. 曲率计算

对三维空间连续曲线,曲率计算公式为:

$$k = \sqrt{\frac{(\dot{x}^2 + \dot{y}^2 + \dot{z}^2)(\ddot{x}^2 + \ddot{y}^2 + \ddot{z}^2) - (\dot{x}\ddot{x} + \dot{y}\ddot{y} + \dot{z}\ddot{z})^2}{(\dot{x}^2 + \dot{y}^2 + \dot{z}^2)^3}} \quad (\text{以 } t \text{ 为参数})$$

对于数字曲线,数字曲率应根据相应变量的数值微分计算得到,当采用前向差分,有:

$$\begin{cases} \dot{x}_i = x_i - x_{i-1} \\ \ddot{x}_i = \dot{x}_i - \dot{x}_{i-1} = x_i - 2x_{i-1} + x_{i-2} \end{cases}$$

直接计算曲率存在两方面的困难:一是公式中含有大量的乘法运算,计算量较大;二是计算数值微分对噪声比较敏感。为此,本文给出一种简单的间接曲率计算方法。首先,对曲线的形式作一些限定,本文限定曲线是分段平面的,所谓分段平面是指在局部区域,曲线段是平面的,这里不对平面方程作限定;然后,把曲线近似视为由平面圆弧连接而成。由于在同一段圆弧内曲率处处相等,且是圆弧半径的倒数,所以,本文将空间曲率计算转化为平面圆弧半径的计算。而圆弧半径可以通过弦高和弦长计算得到。对于曲线 C 上一个给定的点 p_i ,在 p_i 的邻域内计算圆弧半径。为此,首先给出邻域的概念:

[定义 3.8 k-邻域] p_i 的 k -邻域定义为 C 上的一个点集,记为 $R_k(p_i)$, k 为邻域长

度。

$$R_k(p_i) = \{p_j \mid p_j \in C, 0 \leq (i-k) \leq j \leq (i+k) \leq n, i \leq n\}$$

p_i 点所在圆弧的半径通过在其 k -邻域内统计弦长和弦高计算得到。如图 3.6 所示, 其中, h 为弦高, l 为弦长, 存在如下关系:

$$\begin{cases} l = 2R \sin \theta \\ h = R - R \cos \theta \end{cases}$$

可以推得:

$$R = \frac{l^2}{8h} + \frac{h}{2}$$

弦长 l 可以通过计算 k -邻域两个端点间的距离得到。弦高可以通过计算 p_i 与邻域两个端点的中点间的距离得到。

$$\begin{cases} l = |p_{i-k} p_{i+k}| \\ h = |p_i p_i'|, p_i' = \frac{p_{i-k} + p_{i+k}}{2} \end{cases}$$

2. 误差计算

给定一段曲线 $C = p_1 p_2 \dots p_n$, 用直线段 $L = \overline{p_1 p_n}$ 逼近该曲线的误差定义为曲线上各点到直线段的距离的和, 令曲线上点 p_i 到直线 L 上的投影为 p_i^L , 满足:

$$p_i^L \in L \wedge \overline{p_i p_i^L} \perp L,$$

则误差为:

$$E = \sum_{i=2}^{n-1} |p_i p_i^L|$$

下面以一个具体的例子说明机器人轨迹的关键点提取及生成的 Petri 网任务模型。机器人的运行轨迹及其关键点如图 3.7 所示:

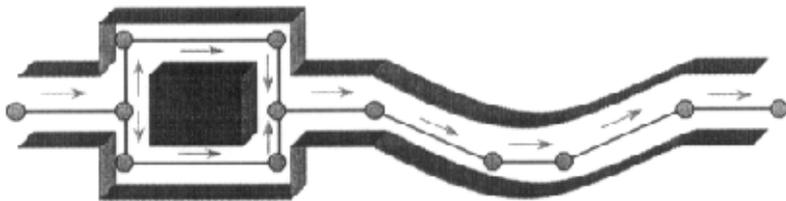


图 3.7 机器人轨迹及关键点图

其中, 黑线部分代表机器人的运动路径, 圆点代表路径上的关键点。由上述运动轨迹所生成的 Petri 网任务模型如图 3.8 所示。

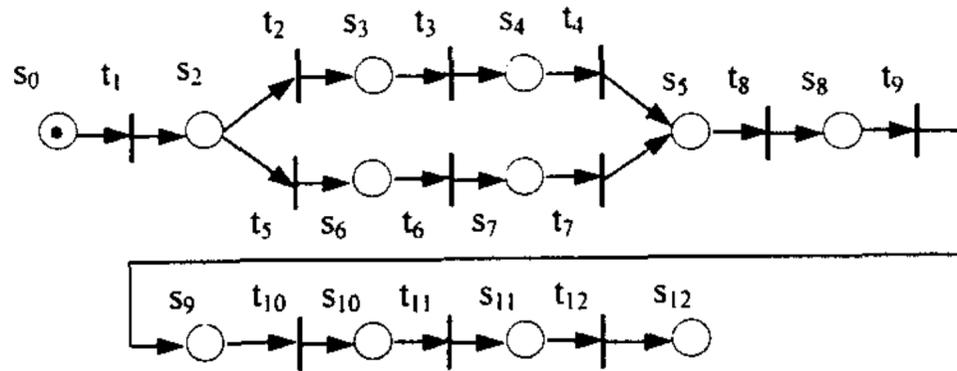


图 3.8 自由运动任务 Petri 网模型图

3.4 基于 Petri 网的分层任务建模方法

本文通过分层 Petri 网来描述任务的分层模型。建立分层 Petri 网模型主要有两种方法：细化和抽象。通过逐层细化（或逐层抽象），可以自上至下（或自下而上）地建立任务的分层描述。本文建立任务的 Petri 网模型的基本思想是先分别建立不同逻辑层次的子任务的 Petri 网模型，在遵守共同的约束的前提下，子任务建模过程是相互独立的；在此基础上，通过一定的约束条件，在不同逻辑层次的子任务模型间建立变迁细化关系，从而实现任务的 Petri 网模型的分层描述。这种方法的优点是，类似于面向对象的程序设计方法，在对子任务模型的输入输出接口进行规范的基础上，可以实现子任务模型在不同任务描述中的重用。

在[148]和[149]中，分别给出了关于分层 Petri 网和细化的形式化描述。本文首先介绍了分层 Petri 网相关的基本概念。为了保证子任务模型间能够实现细化，本文首先给出了状态约束条件，然后，本文将 Petri 网系统的状态标识进行分类，并在此基础上给出变迁细化条件；对满足变迁细化条件的两个子任务模型，本文以一个定理的形式给出了建立二者间变迁细化关系的方法，并对方法的正确性进行了证明。最后，以插孔任务为例，说明了建立分层 Petri 网模型的过程。

3.4.1 分层 Petri 网基本概念

分层 Petri 网的构造和描述是建立在“细化”变换基础上的，细化实质上是变换前后两个 Petri 网之间的映射，描述了 Petri 网之间的节点及连接弧的对应关系。在[148]中给出了分层 Petri 网的形式化描述，本文援引其中相关的基本概念如下：

[定义 3.9 网射] 设 $N = (S, T; F)$ 是一个 Petri 网， $X = S \cup T$ ， $X' \subseteq X$ 是 N 中元素的子集。则有：

- a) X' 在 N 中的边界定义为：

$\mathbf{bd}_N(X') := \{x \in X' \mid \exists y \in X \setminus X' : (x, y) \in F \vee (y, x) \in F\}$ (其中符号“ \setminus ”代表集合间的差运算)

X' 在 N 中的环境定义为: $\mathbf{env}_N(X') := \mathbf{bd}_N(X \setminus X')$

b) 设 $N_1 = (S_1, T_1; F_1)$ 和 $N_2 = (S_2, T_2; F_2)$ 是 Petri 网, $f: X_2 \rightarrow X_1$ 是一个映射, 则 (N_2, N_1, f) 称为一个网射, 当且仅当:

i) $f(F_2) \subseteq F_1 \cup id_{X_1}$, id_{X_1} 表示 X_1 中的单个元素;

ii) $\forall t \in T_2 \forall s \in {}^*t \cup t^* : f(t) = f(s) \vee (f(t) \in T_1 \wedge f(s) \in S_1)$ 。

其中, $f((x, y)) = (f(x), f(y))$ 。i) 表明映射 f 保持了除节点标识外的网结构;

ii) 表明映射 f 尽可能保持了节点的类型 (位置或变迁)。

c) 在网射 (N_2, N_1, f) 中, N_2 称为是 N_1 的一个细化, 而 N_1 则称为是 N_2 的一个抽象。对 $x \in X_1$, N' 是 N_2 的子网, 如果满足 $X' = f^{-1}(x)$, 且有 $F' = F_2 \cap X'^2$, 则称 N' 是 x 的细化, x 是 N' 的前辈 (predecessor)。

d) 一个网射 (N_2, N_1, f) 称为一个商, 当且仅当 f 在节点和弧上是满射, 即:

i) $f(X_2) = X_1$; ii) $F_1 \setminus f(F_2) = \emptyset$

[定义 3.10 细化序列] 设 N_1, \dots, N_n 为 Petri 网, 在映射 f_1, \dots, f_{n-1} 下, (N_{i+1}, N_i, f_i) , $i = 1, \dots, n$, 是商。则称 $V = (N_1, f_1, \dots, f_{n-1}, N_n)$ 为一个细化序列。

如果在每个细化序列中增加一个新网, 以及从 N_1 到新网的映射 f_0 , 新网只包含单一的元素“ \perp ”, 则网络映射 f_0, \dots, f_{n-1} 在节点集合 $X_V \cup \{\perp\}$ 上形成了一个树形结构, 其中树根为 \perp 。树结构是节点集合上的偏序。令“ $<$ ”表示由“前辈”关系导出的树上的偏序, 即 $x < y$ 表示“ x 是 y 的前辈”; $x \leq y : \Leftrightarrow x < y \vee x = y$ 。树根 \perp 总是在偏序关系下的最小元素, 树上的叶节点是最大元素。则以 x 为根节点的子树的叶节点集合为:

$$\mathbf{leaf}(x) := \{y \in x^+ \mid y^+ = \{y\}\},$$

其中 $x^+ := \{z \mid x \leq z\}$, 是 x 的后辈集合, 即 x 的子树上的节点集合。

[定义 3.11 分层 Petri 网] 设 $(S, T; F)$ 是一个 Petri 网, 映射 $f: X \rightarrow X \cup \{\perp\}$, $X = S \cup T$ 。称 $HN = (S, T; F, f, \perp)$ 是一个分层 Petri 网, 当且仅当:

(HN1) 无向图 $G_{HN} = (X \cup \{\perp\}, \{(x, y) \mid f(x) = y \vee f(y) = x\})$ 是连通的;

(HN2) $F \subseteq \mathbf{leaf}(\perp) \times \mathbf{leaf}(\perp)$;

(HN3) $\forall x \in S : \mathbf{bd}_N(\mathbf{leaf}(x)) \subseteq S$; $\forall x \in T : \mathbf{bd}_N(\mathbf{leaf}(x)) \subseteq T$ 。

其中, (HN1) 表明映射 f 在节点集合 $X \cup \{\perp\}$ 中形成了一个树结构, “ \perp ” 是树根。

分层 Petri 网是通过“细化”过程建立的。一种典型的 Petri 网细化方法是用一个子网替换 Petri 网的一个位置 (或变迁) 节点, 同时, 子网保持同被替换元素的前置集和后置集的连接。一个细化过程主要由两部分构成: 建立 Petri 网中的元素与其细化子网元素间的对应关系; 建立一个元素的前置集和后置集与其细化子网间的连接关

系。对一个有标识的 Petri 网系统的细化由细化系统描述[149]。

[定义 3.12 细化系统] 一个 Petri 网细化系统定义为一个 5 元组, $\mathfrak{R} = (M_0, \mathbf{M}, \rho, \phi, \psi)$, 其中:

- M_0 是一个带有初始标识的位置/变迁网, $M_0 = (S_0, T_0; F_0, m_0)$;
- \mathbf{M} 是带有标识的子网集合, $\mathbf{M} = \{M_1, \dots, M_k\}$;
- ρ 是细化函数, 将 S_0 中的元素 (位置细化) 和 T_0 中的元素 (变迁细化) 与 \mathbf{M} 中的子网对应起来, $\rho: S_0 \cup T_0 \rightarrow \{1, \dots, k\}$; 每个位置 $s \in S_0$ 被细化为子网 $M_{\rho(s)}$, 如果 $s \in \text{dom}(\rho)$, 否则 s 保持为一个单一的位置; 每个变迁 $t \in T_0$ 被细化为子网 $M_{\rho(t)}$, 如果 $t \in \text{dom}(\rho)$;
- ϕ 和 ψ 是 (输入/输出) 接口函数, 定义每个位置 (或变迁) 的前置/后置集与其细化之间的连接关系。对每个 $s \in S_0$, 如果 $s \in \text{dom}(\rho)$, 则 $\phi(s): T_0 \rightarrow 2^{S_{\rho(s)}}$ 且有 $\psi(s): T_0 \rightarrow 2^{S_{\rho(s)}}$; 类似的, 对每个 $t \in T_0$, 如果 $t \in \text{dom}(\rho)$, 则 $\phi(t): S_0 \rightarrow 2^{T_{\rho(t)}}$, 且有 $\psi(t): S_0 \rightarrow 2^{T_{\rho(t)}}$ 。

3.4.2 变迁细化方法

在建立分层 Petri 网模型过程中, 按照细化的元素类型不同, 可以把细化变换分为位置细化和变迁细化两类。本文主要采用变迁细化的方式。这是因为, 变迁细化同机器人编程的思路更加吻合。对给定的机器人语言, 构造复杂的机器人程序的一个主要方式是通过子程序嵌套实现的, 每个子程序实现特定的状态转换功能, 对应于任务的 Petri 网模型中的变迁元素。

在基于子任务的 Petri 网模型建立分层任务模型的过程中, 面临的一个问题是: 如果子任务模型的类型和规范不一致, 则将难以通过细化建立分层 Petri 网模型。这里所谓的规范是指 Petri 网的元素 (位置和变迁) 的含义及其定义方法。不同逻辑层次的子任务模型对应的 Petri 网类型可能不同, 比如, 动作级任务通常用条件/事件网模型来描述, 而任务级的模型可能用高级网如谓词/变迁网来描述更有效。在操作的不同阶段, 子任务的规范很可能不同, 比如, 在自由运动阶段, 位置节点代表路径上的关键点, 而在接触运动过程中, 位置节点则代表机械手与操作环境的接触状态。因此, 必须对子任务模型的类型和规范进行约束, 以保证能通过细化建立分层模型。考虑到高级网在特定语义下, 可以转化为等价的条件/事件网 (C/E 网), 本文假定对应不同子任务的 Petri 网模型类型相同——都是 C/E 网。在此基础上, 需要对作为位置元素的系统状态的定义作必要的限制, 以保证细化后子网同环境间能建立合理的输入输出关系。

对那些由机器人运动所产生的操作状态，基本要求是状态是具体的。具体的含义有两个方面：一方面是相对于抽象状态、复合状态而言的，状态必须是相对某种测量手段可观测的元状态；另一方面，具体是同机器人的基本动作相联系的，即每个状态必须是能够通过基本动作可实现的或者是引发基本动作的条件。据此，本文给出状态约束条件如下：

[定义 3.13 状态约束条件] 设分层 Petri 网 $HN = (S, T; F, f, \perp)$ ，则对 $\forall s \in S$ ，满足：

- i) $\forall t \in {}^{\circ}s, \exists x \in \text{bd}_N(\text{leaf}(t)), s \in x^{\circ}$;
- ii) $\forall t \in s^{\circ}, \exists x \in \text{bd}_N(\text{leaf}(t)), s \in {}^{\circ}x$ 。

这里，元素的前置集、后置集都是通过连接关系 F 确定的。

· 对满足上述状态约束的两个子任务 Petri 网模型，通过建立变迁细化关系实现任务的分层描述。为了保证变迁细化关系的存在，本文给出一个变迁细化条件。

设有两个 Petri 网系统 $PN_1 = (S_1, T_1; F_1, M_1^0)$ ， $PN_2 = (S_2, T_2; F_2, M_2^0)$ ，其中 M_1^0, M_2^0 分别为 PN_1 和 PN_2 的初始状态标识。这里，假定 PN_2 对应子网模型，要根据 PN_2 实现对 PN_1 中某个变迁的细化。考虑到两个事实：a) 变迁的激发需要其输入集中的节点都含有一个托肯；b) 变迁激发后，该变迁节点处于静止状态。本文根据这两点事实形成变迁细化条件。首先，本文将 Petri 网系统的状态标识分为初始状态标识、终止状态标识和中间状态标识三类。初始状态标识即为 M_0 。

[定义 3.14 终止状态标识] 对 Petri 网系统 $PN = (S, T; F, M_0)$ ，称一个可达状态标识 M 为终止状态标识，如果满足：

$$\forall t \in T, \exists s \in {}^{\circ}t, M(s) = 0。$$

在终止状态标识下，没有一个变迁能够被激发，或者说 Petri 网系统的运行遇到终止状态标识而停止。终止标识集合记为 M_e 。一个可达状态标识如果不是初始或终止状态标识，则为中间状态标识。对一个状态标识 M ，令位置节点集合 $SS_M = \{s \in S \mid M(s) = 1\}$ 。

对 PN_1 中的某个变迁节点 $t \in T_1$ ，如果要根据 PN_2 实现对 t 的细化，需要满足如下的变迁细化条件：

[定义 3.15 变迁细化条件]

- i) $X_1 \cap X_2 = {}^{\circ}t \cup t^{\circ}$;
- ii) $\exists t' \in T_2, {}^{\circ}t' = {}^{\circ}t$ ，且有： ${}^{\circ}t = SS_{M_0^2}$;
- iii) $\forall M \in M_e^2$ ，满足 $t^{\circ} \subseteq SS_M$ 。

其中，条件 i) 保证两个子网的元素间具有必要的独立性，即除 t 的输入输出集外不能有重复元素；条件 ii) 保证 t 的输入集将启动细化子网运行，并且子网启动的行

为与变迁 t 的一致。这种一致性是指子网启动运行，当且仅当 t 的输入集都含有托肯；条件 iii) 保证细化子网的运行结果将使 t 的输出集中的节点拥有托肯，从而保证子网同变迁 t 的运行结果的一致性。

易知如果 $t \in T_1$ 对 PN_2 满足变迁细化条件，则有 $\bullet t \cup t^\bullet \subseteq S_2$ 。变迁细化条件是根据 PN_2 对 $t \in T_1$ 进行细化的一个充分条件。在满足变迁细化的条件下，通过本文提出的一个定理给出一种在两个 Petri 网系统模型间建立变迁细化的方法，简称变迁细化方法。

[定理 3.1] 给定两个 Petri 网系统 $PN_1 = (S_1, T_1; F_1, M_1^0)$ ， $PN_2 = (S_2, T_2; F_2, M_2^0)$ 。对某个变迁 $t_1 \in T_1$ ， PN_2 相对 PN_1 及 t_1 ，满足变迁细化条件。

令根据 PN_2 生成的子网 $N' = (S', T'; F')$ 如下：

$$S' = S_2 - \bullet t_1 \cup t_1^\bullet; \quad T' = T_2; \quad F' = F_2 \cap X'^2, X' = S' \cup T'.$$

则存在一个网射 (N_1, N_3, f) ，实现由 N' 对 t_1 的细化，且该网射是一个商，其中 $N_1 = (S_1, T_1; F_1), X_1 = S_1 \cup T_1$ 。

[证明] 由变迁细化条件 i) 可知， $X_1 \cap X' = \emptyset$ ，因此，本文可以根据 N_1 和 N' 构造 Petri 网 N_3 如下：

$$\text{令 } S_3 = S_1 \cup S', \quad T_3 = (T_1 - \{t_1\}) \cup T',$$

$$F_3(x, y) = \begin{cases} F_1(x, y), x, y \in X_1; \\ F'(x, y), x, y \in X'; \\ F_2(x, y), (x \in \bullet t_1 \wedge y \in T') \vee (x \in T' \wedge y \in t_1^\bullet); \\ \text{没有定义, 对其它 } (x, y). \end{cases}$$

根据 N_3 的定义，有 $\forall x \in X_3, x \in X_1 \vee x \in X'$ 。定义网络映射 $f: X_3 \rightarrow X_1$ 如下：

$$\forall x \in X_3, f(x) = \begin{cases} x, x \in X_1 \\ t_1, x \in X' \end{cases}$$

根据 $F_3(x, y)$ 的定义，对 $(x, y) \in F_3$ ，当 $x \in \bullet t_1 \wedge y \in T'$ 时， $f((x, y)) = (x, t_1) \in F_1$ ；当 $x \in T' \wedge y \in t_1^\bullet$ 时， $f((x, y)) = (t_1, y) \in F_1$ 。因此，有 $f(F_3) \subseteq F_1 \cup \{t_1\}$ ，符合网射定义 [定义 3.9] 中的条件 i)；

$$\forall t \in T_3, \forall s \in \bullet t \cup t^\bullet, \begin{cases} f(t) = t \in T_1 \wedge f(s) = s \in S_1, t \in T_1 \\ f(t) = f(s) = t_1, t \in T' \wedge s \in S' \\ f(t) = t_1 \in T_1 \wedge f(s) = s \in S_1, t \in T' \wedge s \in S_1 \end{cases}, \text{符合网射定义中}$$

条件 ii)。

因此， (N_3, N_1, f) 构成一个网射。

$$\forall x \in X_1, \begin{cases} \text{当 } x \neq t_1 \text{ 时, } x \in X_3, f(x) = x, \\ \text{当 } x = t_1 \text{ 时, } \exists x \in X' \subset X_3, f(x) = t_1, \end{cases}, \text{ 所以有: } f(X_3) = X_1;$$

要证商定义条件 ii), 只要证 $\forall (x, y) \in F_1$, 满足: $(x, y) \in f(F_3)$, 分三种情况:

(1) $x, y \neq t_1$

有: $x, y \in X_3, (x, y) \in F_3, \because f(x) = x \wedge f(y) = y, \therefore (x, y) = f((x, y)) \in f(F_3)$;

(2) $x = t_1, y \neq t_1$

有: $y \in t_1^*$, 根据变迁细化条件 iii), 在 PN_2 中, $\forall M \in M_e^2, t_1^* \subseteq SS_M$,

$\therefore \exists t \in T_2 = T' \subseteq T_3, s.t., (t, y) \in F_2$,

$\therefore (t, y) \in F_3, f(t) = t_1 \wedge f(y) = y, f((t, y)) = (t_1, y) = (x, y)$

(3) $x \neq t_1, y = t_1$

有: $x \in t_1^*$, 根据变迁细化条件 ii), 在 PN_2 中, $\exists t' \in T_2, t' = t$,

$\therefore \exists t \in T_2 = T' \subseteq T_3, s.t., (x, t) \in F_2$,

$\therefore (x, t) \in F_3, f(t) = t_1 \wedge f(x) = x, f((x, t)) = (x, t_1) = (x, y)$

所以有 $\forall (x, y) \in F_1 \Rightarrow (x, y) \in f(F_3)$, 即 $F_1 \setminus f(F_3) = \emptyset$ 。因此, (N_3, N_1, f) 是一个商。因此, 根据 PN_1 和 PN_2 , 可以建立变迁细化。□

3.4.3 应用实例

下面以插孔任务为对象说明构造任务分层描述的方法。插孔任务一般分为两个步骤: 搜孔和插孔。在搜孔阶段, 机器人的运动主要由自由运动以及带有受力约束的接触运动构成, 往往处于位置控制模式; 而在插孔阶段, 接触运动过程由力控制实现。本文将二维平面情况为例简单说明插孔过程。

把机器人的自由运动空间划分为两个部分, 如下图所示:

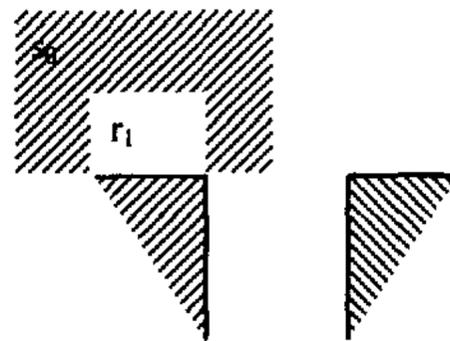


图 3.9 运动空间分割图

其中阴影所围空白区域 r_1 代表插孔任务前插件的预备位置, 而该区域外的自由运动区域形成自由运动的起始状态 s_0 。插件与插孔间的接触状态由接触特征对来描述。接触特征由顶点、边特征构成。插件与插孔的特征如图 3.10 所示:

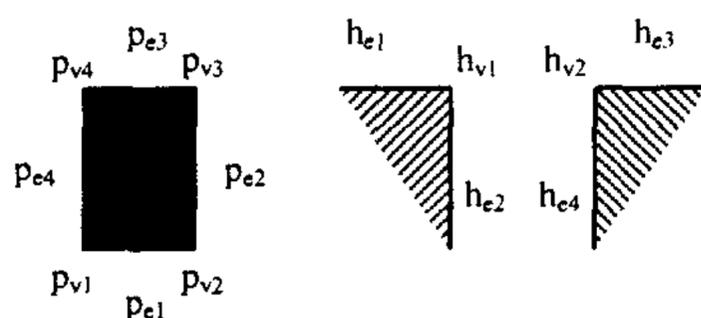


图 3.10 接触特征示意图

其中, p_{v_i} 代表插件的顶点特征, p_{e_i} 代表插件的边特征; h_{v_i} 代表插孔的顶点特征, h_{e_i} 代表插孔的边特征。

插孔任务相关的接触状态描述如下图所示:

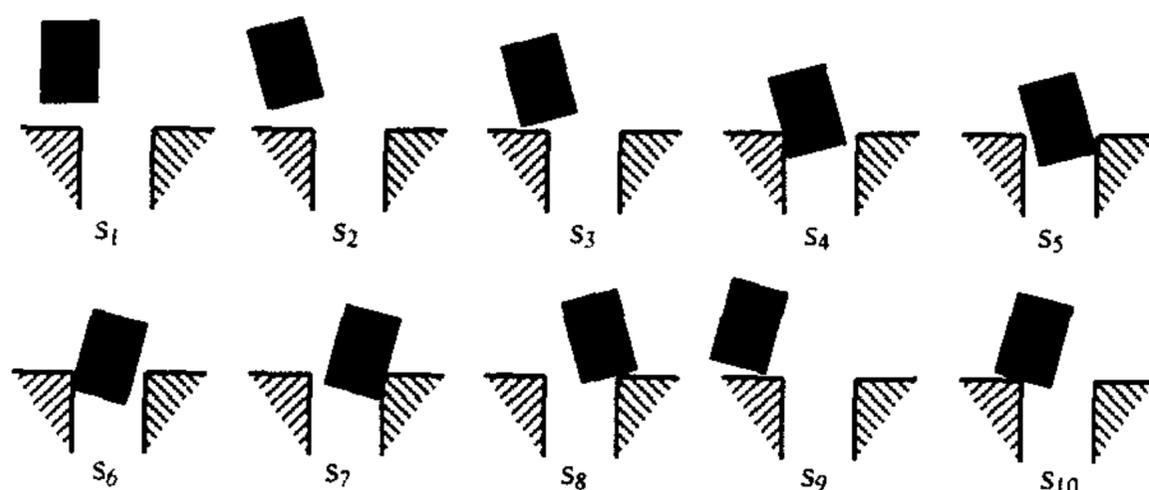


图 3.11 接触状态示意图

其中, s_1 是插孔任务的预备状态, 当插件特征 (p_{v_1} 或 p_{v_2}) 进入区域 r_1 时, 即为预备状态; s_2 是插孔任务的调整状态, 在插孔任务前, 需要调整插件的姿态以便搜孔。以下各状态皆为接触状态, 分别由接触特征对描述如下:

$$s_3 : (p_{v_1}, h_{e1}); \quad s_4 : (p_{e4}, h_{v1}); \quad s_5 : (p_{v_2}, h_{e4}); \quad s_6 : (p_{v1}, h_{e2});$$

$$s_7 : (p_{v_2}, h_{e2}); \quad s_8 : (p_{e1}, h_{v2}); \quad s_9 : (p_{v_2}, h_{e1}); \quad s_{10} : (p_{e1}, h_{v1})。$$

插孔任务的状态集合为 $S = \{s_i\}, i = 0, 1, \dots, 10$ 。

搜孔子任务的一般过程为: 首先从初始位置运动到预备位置, 然后调整姿态, 向下运动到同操作台接触, 搜索孔的位置, 搜孔子任务的正常结束条件是插件局部进入插孔内 (由状态 s_4, s_5, s_6, s_7 表征)。这里, 令 s_4 为搜孔子任务的正常结束状态。搜孔子任务和插孔任务的 Petri 网系统模型分别如下:

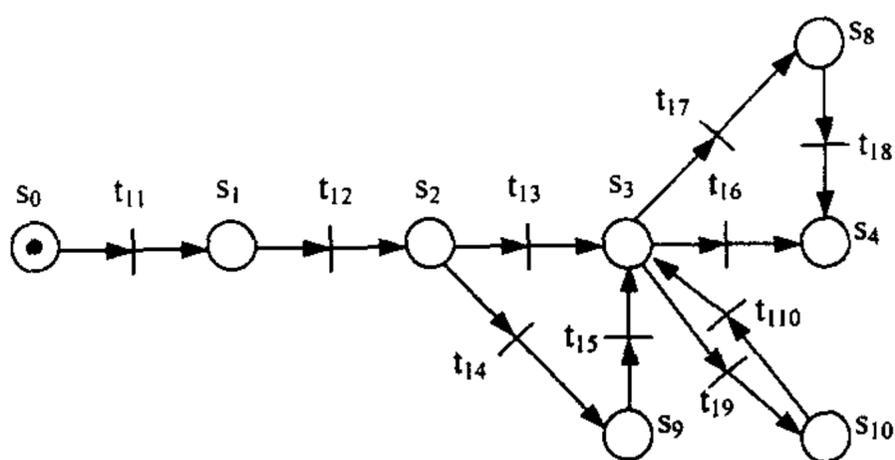


图 3.12 搜孔子任务 Petri 网系统模型图

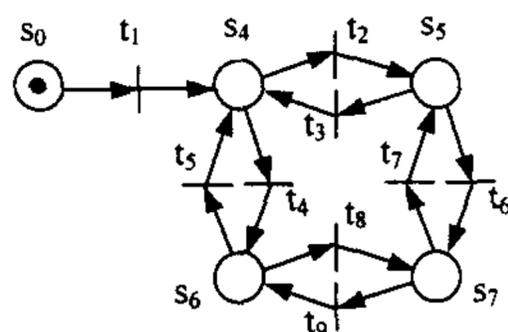


图 3.13 插孔任务 Petri 网系统模型图

其中 t_1 代表搜孔子任务。可以看出，搜孔子任务模型相对于插孔任务模型满足变迁细化条件，经细化后，插孔任务的 Petri 网模型如下：

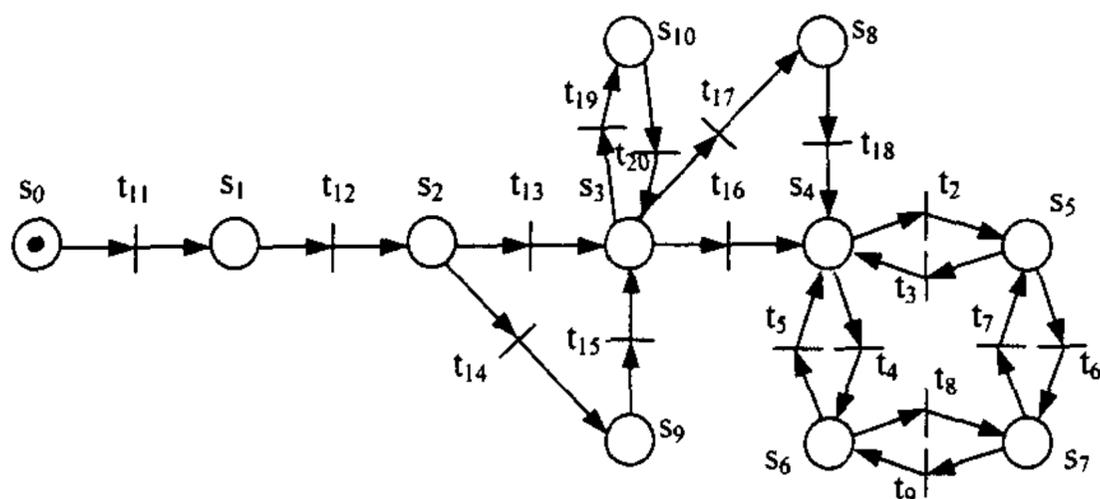


图 3.14 插孔任务细化模型图

3.5 基于 Petri 网的遥操作任务协调

本文中所谓任务协调，有别于 Saridis 智能机理论中协调级的功能，其主要功能是实现操作员的任务规划功能，将操作员的规划行为转化为机器人任务。在本文大时延遥操作系统框架（见 3.1 节）中，操作员的规划行为是通过操作员操纵虚拟机器人在虚拟环境中进行连续的运动和操作给出的。而最终的机器人任务是指机器人能够理解并执行的动作序列。

任务协调的基本思想是首先基于任务空间的谓词描述构造相应的谓词/变迁网（Predicate/Transition Nets，简称为 Pr/T 网）模型，在此基础上，根据操作员的操作导致的操作状态的变化，进行状态识别。机器人的动作由 Pr/T 网中相应的输入状态和输出状态确定，因此根据变化前后的操作状态，进行动作识别，从而可以根据操作员的连续操作获得动作序列。为了满足进一步分析的需要，根据动作序列，将任务描述为基本 Petri 网的形式。任务协调方法所涉及的关键问题包括以下几个方面：

- 识别虚拟机器人及虚拟环境的状态;
- 根据虚拟机器人及虚拟环境的前后状态变化, 确认相应的机器人动作;
- 根据动作序列形成任务描述。

在[150]中给出了谓词/变迁网的形式描述, 本文首先引用了该文中关于谓词/变迁网的基本概念。然后, 给出了任务空间的相关定义, 在此基础上, 建立任务空间的谓词描述, 阐述了由任务空间的谓词描述导出 Pr/T 网描述的方法; 针对结构化环境下的装配作业任务, 论述了状态识别方法, 给出了判别状态的谓词条件, 论述了利用任务空间的上下文信息减少状态识别的计算量的方法; 给出了动作识别条件; 论述了基于任务空间的 Pr/T 模型和任务的动作序列生成任务的 C/E 网模型的方法。最后, 本文针对插孔任务, 描述了任务协调的应用实例。

3.5.1 谓词/变迁网基本概念

谓词/变迁网是由 Genrich 和 Lautenbach 提出的[150], 其目的是对个体动态变化的属性及其相互关系提供一种形式化的描述。谓词/变迁网所描述的对象是由个体及其相应的函数和关系构成的结构。在数理逻辑中, 一个结构定义为 $\mathfrak{R} = (D; f_1, \dots, f_k; R_1, \dots, R_n)$, 其中, D 是个体的非空集合, 称为论域。 f_i 是论域 D 上的函数, R_j 是定义在 D 上的关系。一般, 在谓词/变迁网中, 结构中可变的的关系对应于位置, 而静态的关系及函数, 或者说它们的名字作为注解。下面给出谓词/变迁网相关的定义:

[定义 3.16 面向语言的结构] 一个结构 \mathfrak{R} 是面向语言 L 的, 或者说是一个 L -结构, 如果 L 中的每个运算符 $f^{(m)}$ 都表示 \mathfrak{R} 中的一个 m 元函数, 记为 $f_{\mathfrak{R}}$, 而且 L 中的每个谓词 $P^{(n)}$ 都表示 \mathfrak{R} 中的一个 n 元关系, 记为 $P_{\mathfrak{R}}$ 。

对于一个动态的 L -结构, 本文记表示所有关系的谓词集合为 Π , 可以划分为两个部分, 一部分是静态谓词集合, 记为 Π_s , 一部分是可变的谓词集合, 记为 Π_v 。

[定义 3.17 符号和] 对于给定的一阶谓词语言 L , 对每个 $n \geq 0$, 一类 n 元的符号和 $LC^{(n)}$ 定义为:

- 常量 0 属于 $LC^{(n)}$;
- 如果 v_1, \dots, v_n 是 L 中的项, 那么 n 元组 $\langle v_1, \dots, v_n \rangle$ 属于 $LC^{(n)}$;
- 如果 l_1, l_2 属于 $LC^{(n)}$, 那么 $(l_1 + l_2)$ 属于 $LC^{(n)}$;
- 如果 l 属于 $LC^{(n)}$, 并且 z 是一个非负整数, 那么 zl 属于 $LC^{(n)}$;
- $LC^{(n)}$ 不包括其它表达式。

对于 $n \geq 0$, 所有 $LC^{(n)}$ 集合的并记为 LC 。

[定义 3.18 谓词/变迁网] 设 L 为一阶谓词语言, 令 L_s 表示只涉及 Π_s (表示静态

关系的谓词集合)的子语言。那么一类有注解的标识网 PRT_i 定义为, $MN = (N, A, M^0)$, 其中, N 是一个 Petri 网, A 是语言 L 中的注解, M^0 是标识。谓词/变迁系统记为 Pr/T 系统。

1. N 是一个 Petri 网, $N = (S, T; F)$;
2. A 是 N 的注解, $A = (A_N, A_S, A_T, A_F)$, 其中
 - a) $A_N = \mathfrak{R}$ 是一个面向 L_s 的一阶结构, 称为 MN 的支撑;
 - b) $A_S : S \rightarrow \Pi_v$, 是一个在位置集合 S 和可变谓词集合 Π_v 间的双射, 对 $s \in S$, 若 $A_S(s)$ 是一个 n 元谓词, 就说 s 是 n 元谓词;
 - c) $A_T : T \rightarrow f_D$, 其中, f_D 是 D 上的公式集, 对 $t \in T$, $A_T(t)$ 只包含运算符和静态谓词;
 - d) $A_F : F \rightarrow LC$, 是一个从弧的集合 F 到语言 L 的项的符号和集合 LC 的映射, 即对某个弧 $(x, y) \in F$, 无论是从位置 s 流入或流出 ($x = s$ or $y = s$), 如果 n 是注解 s 的谓词的索引, 那么, $A_F(x, y)$ 属于 $LC^{(n)}$ 。
3. $M^0 : S \rightarrow LC$, 是位置的标识, 是从位置 $s \in S$ 到 n 元常量的符号和的映射, 如果 n 为注解 s 的谓词的索引, 那么, $M^0(s)$ 属于 $LC^{(n)}$, 并且 $M^0(s)$ 的值, $\mathfrak{R}(M^0(s))$ 是一个系数为 0 或 1 的线性组合。

[定义 3.19 可行替换] 设置换 $\alpha = (v_1 \leftarrow d_1, v_2 \leftarrow d_2, \dots, v_n \leftarrow d_n)$, 则 α 为标识 M 下变迁 t 的可行替换的充分必要条件是:

- i) $A_T(t) : \alpha = TRUE$,
- ii) 对所有 $s \in {}^*t$, $A_F(s, t) : \alpha \leq M(p)$ 。

[定义 3.20 变迁规则] 设 M 为 Pr/T 系统的一个标识, 则:

- i) 变迁 $t \in T$ 在 M 有发生权的充分必要条件是存在 M 下的可行替换 α ; 记为 $M[t : \alpha]$;
- ii) 若 $M[t : \alpha]$, 其后继标识 M' 为: 对所有 $s \in S$,

$$M'(s) = M(s) - (A_F(s, t) : \alpha) + (A_F(t, s) : \alpha)$$

3.5.2 任务空间模型

本文为了描述任务空间模型, 首先明确如下一些基本定义:

[定义 3.21 个体] 个体是指操作环境中与机器人操作相关的单个实体, 包括动态的实体如: 机器人、操作对象, 以及静态的实体如操作台等。

[定义 3.22 操作状态] 操作状态是对操作过程中某一时刻, 对个体的状态和个体间相互关系的全面的定性描述。

[定义 3.23 动作] 动作是机器人能够理解并执行的基本命令单元。在规划过程中, 动作是改变操作状态唯一的直接来源。同时, 一个动作可以由其所导致的操作状态的变化所完全描述。一个机器人所能支持的动作是有限的。

[定义 3.24 任务] 一个任务是对一个动作序列以及与动作序列相关的操作状态变迁过程的综合描述。

[定义 3.25 任务空间] 任务空间由操作状态集合、动作集合, 以及操作状态和动作之间的因果关系所构成的综合描述。任务空间蕴含了可能发生的机器人动作序列。

在一阶谓词描述的任务空间中, 机器人的动作由前提条件和目标条件所决定。参考人工智能中经典的 STRIPS 任务规划系统, 一个动作的发生, 将导致前提条件中一部分谓词不能成立, 同时使一部分新的谓词成立。这种谓词的状态的变化恰好可以通过谓词/变迁网的运行来描述。因此本文首先建立任务空间的谓词描述, 然后由谓词描述导出相应的 Pr/T 网描述。

本文用一阶谓词逻辑作为描述任务空间的语言。设一阶谓词语言为 L 。则任务空间中的个体对应于 L 中的常量, 在 L 的变量集合 V 中, 每个元素代表任务空间中的个体。

对 $\forall n \geq 0$, $P^{(n)}$ 代表 L 中的 n 元谓词集合, $P = \bigcup_{n \in N} P^{(n)}$ 是 L 的谓词集合。 P 中的元素对应任务空间中个体的状态(一元谓词对应个体状态)以及个体间的相互关系。

一个操作状态通常是由一组状态和关系联立构成, 这在谓词语言 L 中对应于一个合取公式。本文限制描述一个操作状态的公式为文字的合取式, 而且其中的任何变量都假定由存在量词量化了。

一个动作把一个操作状态变为另一个操作状态。在语言中对应于一条蕴含规则, 操作状态的变迁由规则的前件和后件描述。本文采用 STRIPS 规划系统中的 F 规则的形式来描述一个动作。在 F 规则中, 一个动作由三部分描述: 第一部分是先决条件公式, 先决条件作为操作状态的描述, 由文字的合取组成, 且变量都假定由存在量词量化; 第二部分是一个叫删除表的文字表, 表示原来的操作状态中不再成立的部分, 这里假定删除表中的所有自由变量都是先决条件中的变量; 第三部分是加添公式, 对应于蕴含规则的后件, 是由文字的合取式构成。

设一阶谓词语言 L 上的非空个体集合为 D , 则定义在 L 上的一个关系结构是一个二元组, $R = (D, P)$, P 为 L 上的谓词集合。设 F 规则集合为 Γ , 则任务空间由定义在语言 L 上的关系结构 R 及 Γ 完全描述, 可以描述为二元组 $T = (R, \Gamma)$ 。

根据任务空间的谓词语言描述得到相应的 Pr/T 网描述。任务空间中个体的状态和个体之间的关系对应于 Pr/T 的位置, 预先定义的动作对应于 Pr/T 的变迁, 系统的操作状态由 Pr/T 的标识描述, 而每一个动作对应的前件和后件则由相应的弧来描述。

3.5.3 状态识别

在任务空间的 Pr/T 网模型基础上, 任务协调需要从虚拟机器人的连续运动过程, 识别出离散的操作状态, 本文称这一过程为状态识别。在讨论具体的状态识别方法之前, 主要针对机器人装配作业任务, 作如下假设:

- 1) 任务空间是结构化的, 任务空间中个体的几何位置和形状是已知的;
- 2) 任务空间中个体的状态及相互关系由个体的位置及个体间的接触状态完全确定;

对于结构化的环境, 可以根据个体的几何模型及其在空间中的位置、姿态, 通过碰撞检测确定接触状态, 从而判断相对关系。对任务空间中的谓词集合 P , 状态识别的过程就是确定 P 的元素全部为真的子集 P' 的过程, 即 $P' = \{p \in P \mid p = TRUE\}$ 。本文规定, 一个谓词是否为真, 是由所涉及的个体的特征面是否发生接触来决定的, 称这些判定谓词的接触条件为谓词条件。本文把一个谓词条件表达成若干子条件的合取的形式, 每个子条件为所涉及的两个个体中相应的特征面是否发生接触的条件, 下面给出谓词条件的形式定义。

[定义 3.26 谓词条件] 对 n 元谓词 $p(O_1, O_2, \dots, O_n)$, O_i 为所涉及的个体, 其成立的条件为:

1. $C = C_1 \wedge C_2 \wedge \dots \wedge C_i \wedge (\neg C_{i+1}) \wedge (\neg C_{i+2}) \wedge \dots \wedge (\neg C_m)$;
2. $C_i = \exists(u, v) Contact(f_u^{i1}, f_v^{i2}), f_u^{i1} \in F^{i1}, f_v^{i2} \in F^{i2}, i = 1, 2, \dots, m$ 。

其中, F^{i1}, F^{i2} 分别为对应于 C_i 的两个个体的特征面集合。

以插孔任务为例, 分别对谓词 PEGINHOLE(p, h) 和 HOLEEMPTY(h) 给出其谓词条件。插件和插孔的特征面示意图如下:

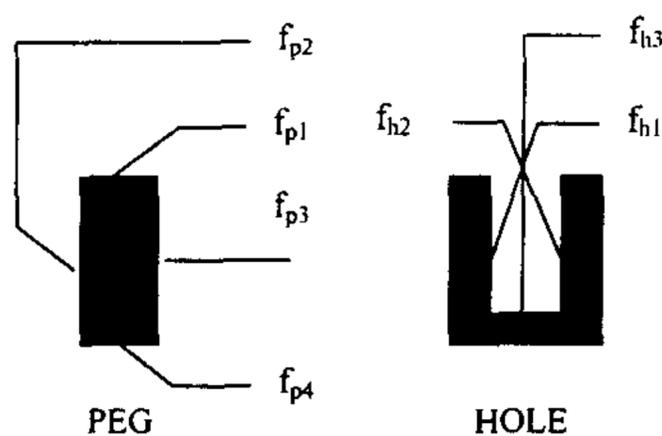


图 3.15 插孔任务特征面示意图

PEGINHOLE(p, h):

$$C = C_1 \wedge C_2,$$

$$F^{1p} = F^{2p} = \{f_{pi}\}, i = 1, 2, 3, 4, F^{1h} = \{f_{h1}, f_{h3}\}, F^{2h} = \{f_{h2}, f_{h3}\}$$

HOLEEMPTY(h):

$$C = \neg C_1,$$

$$F^{lp} = \{f_{p_i}\}, i = 1, 2, 3, 4, \quad F^{lh} = \{f_{h_1}, f_{h_2}, f_{h_3}\}$$

从计算复杂性的角度考虑, 通过基于几何模型间的碰撞检测来确定接触状态, 进而确定相对关系的方法, 对于复杂的任务空间是难以接受的。因此, 减少计算复杂性成为状态识别的关键问题。状态识别的计算复杂性由三个方面决定: 待判定的谓词的数量、每个谓词涉及的两两检测的个体的数量, 以及两个个体间待检测的特征面的数量。对给定的谓词, 个体的特征面是确定的, 而特征面间的碰撞检测属于计算几何范畴的问题, 需要通过专门的算法来解决, 在此不加赘述。因此, 本文主要从谓词数量和个体数量两个方面来考虑减少状态识别的复杂性。主要的思想是基于任务空间模型, 利用上下文相关信息尽量减少检测的范围。

在任务空间的 Pr/T 网系统模型中, 可以根据系统运行的每一步, 确定下一步可能的可达标识, 下一步可达标识集合确定了状态识别的谓词及其个体范围。对给定状态标识 M , 设满足引发条件的变迁集合为 $T_M = \{t_1, t_2, \dots, t_n\}$, 则待检测的谓词集合为:

$$P = \{A_s(s) \mid s \in \bigcup_{i=1}^n (t_i^* - \cdot t_i)\},$$

其中, $A_s(s)$ 为从位置到谓词的映射 (见定义 3.18), $t_i^*, \cdot t_i$ 分别代表变迁 t 的后置集和前置集。设在状态标识 M 下的可行替换集合为 $\{\alpha_i\}, i = 1, 2, \dots, l$, 对给定 n 元谓词 $p(v_1, v_2, \dots, v_n)$, 令 $D_p(\alpha) = \{d \mid \exists i, d \text{ 是在 } \alpha \text{ 下置换 } p \text{ 中变量 } v_i \text{ 的个体}, i = 1, \dots, n\}$, 则待检测的个体集合为:

$$D_p = \{d \mid \exists i, d \in D_p(\alpha_i), i = 1, \dots, l\}$$

3.5.4 动作识别

假设每个动作由其前件和后件唯一确定, 那么, 在理想情况下, 识别出当前状态后, 应该可以确定当前的动作。但是, 问题在于状态的变化并不一定同预定义的动作的结果一一对应, 由于任务空间模型通常考虑的是正常条件下的状态变迁, 因此, 系统当前状态可能不是相对给定任务空间模型的可达状态标识。称非可达状态标识的出现为系统异常。导致系统异常的原因包括:

- 1) 任务建模中忽略了的中间状态;
- 2) 操作员误操作导致的错误状态。

考虑到操作员可能进行的非线性规划过程, 区分中间状态和错误状态对任务协调器而言难度很大, 因此, 本文不赋予任务协调对系统异常进行判断和响应的功能, 而是把这部分功能留给操作员来完成。为了保证系统的正常运行, 需要对操作员的操作

作必要限制,使得系统的运行遵从动作识别条件。

[定义 3.27 动作识别条件] 对给定的系统初始状态标识 M_0 , 当前动作的初始状态标识 M_1 , 以及当前标识 M 。本文称动作是可识别的, 如果 M 满足如下条件:

- i) M 是从 M_0 可达的, 即 $M_0 \rightarrow M$;
- ii) M 是从 M_1 可达的, 即 $M_1 \rightarrow M$ 。

条件 i) 说明系统只对可达状态标识进行动作识别, 本文称满足而条件 i) 的状态为正常状态; 条件 ii) 限制操作员必须保证相邻的两个正常状态之间存在可达关系。

3.5.5 任务生成

在动作识别的基础上, 任务协调将操作员的一系列操作转化为一个动作序列, 每一个动作对遥机器人而言是可执行的。为了满足对性能分析和任务监控的需要, 还需要给出针对任务的 Petri 网模型描述。

设任务空间的 Pr/T 网系统模型为 $PN = (N, A, M^0)$, 其中 $N = (S, T; F)$ 。一个动作序列 $p_1 p_2 \dots p_n$, 在映射 $p = A_T(t)$ 下, 可以看作是对 PN 中的变迁序列 $t_1 t_2 \dots t_n$ 的一个实例。令当前任务的相关变迁集合, $T = \{t \mid \exists i, t = t_i, i = 1, 2, \dots, n\}$, 本文基于 T 生成当前任务的 C/E 网模型描述, 简称任务模型。任务模型的生成大体分为两个步骤: 首先, 将 PN 展开为一个基本网, $PN' = (\hat{S}, \hat{T}, \hat{F}, \hat{c}_0)$, 其中, 每个 $s \in \hat{S}$ 是 $A_S(s), s \in S$ 的一个实例, 而 $t \in \hat{T}$ 是对 PN 中变迁的一个可行实例; 第二步, 对 PN' 进行精简, 得到 C/E 网系统 $\sum(PN) = (B, E; F', c_0)$ 。关于实例化和精简, 都在[150]中给出了相应的形式化描述, 下面援引如下:

[定义 3.28 变迁实例] 设 $t \in T$, α 是对 t 的一个可行替换, 则 t 的 α -实例, 记为 $t:\alpha$, 是一个集合对 $(\cdot(t:\alpha), (t:\alpha)\cdot)$, 其中的元素为闭合的原子公式。

一个公式 $P(d_1, d_2, \dots, d_n) \in \cdot(t:\alpha)$, 当且仅当存在一个位置 $s \in S$, $(s, t) \in F$, 且有:

- i) P 是 s 的一个注解, $P = A_S(s)$;
- ii) 在符号和 $A_T(s, t):\alpha$ 中, $d \in \{d_1, d_2, \dots, d_n\}$ 的系数为 1;

一个公式 $P(d_1, d_2, \dots, d_n) \in (t:\alpha)\cdot$, 当且仅当存在一个位置 $s \in S$, $(t, s) \in F$, 且有:

- i) P 是 s 的一个注解, $P = A_S(s)$;
- ii) 在符号和 $A_T(t, s):\alpha$ 中, $d \in \{d_1, d_2, \dots, d_n\}$ 的系数为 1;

给定动作序列 $p_1 p_2 \dots p_n$ 及其对应的变迁集合 T , 可以构造基本网 $PN' = (\hat{S}, \hat{T}, \hat{F}, \hat{c}_0)$ 如下 (在此, 不对动作及其变迁区分):

- i) $\hat{T} = \{p \mid \exists i, p = p_i, i = 1, 2, \dots, n\}$, 为动作序列中动作构成的集合;

- ii) $\hat{S} = \bigcup_{i \in \hat{I}} (\cdot \hat{i} \cup \hat{i} \cdot)$, 为动作对应的变迁实例构成的集合;
- iii) $\hat{F} = \bigcup_{i \in \hat{I}} (\cdot \hat{i} \times \{\hat{i}\} \cup \{\hat{i}\} \times \hat{i} \cdot)$;
- iv) \hat{c}_0 由 M^0 产生: 若使 $\hat{s} = P(d_1, d_2, \dots, d_n)$ 属于 \hat{c}_0 , 当且仅当存在一个位置 $s \in S$, 其注解为 $A_s(s) = P$, 并且在 $M^0(s)$ 中, $d \in \{d_1, d_2, \dots, d_n\}$ 的系数为 1。

在 PN' 的基础上进行简化, 可以获得最终的任务模型 $\sum(PN) = (B, E; F', c_0)$ 。简化依据下面等价关系的概念。

[定义 3.29 等价关系]

- 1) 称两个元素 $\hat{x}, \hat{y} \in \hat{X} = \hat{S} \cup \hat{T}$ 是等价的, 记为 $\hat{x} \sim \hat{y}$, 当且仅当 $\cdot \hat{x} = \cdot \hat{y}$ 和 $\hat{x} \cdot = \hat{y} \cdot$;
- 2) 设 $\hat{x}_1 \sim \hat{x}_2$ 且 $\hat{y}_1 \sim \hat{y}_2$, 则有 $(\hat{x}_1, \hat{y}_1) \in \hat{F} \Leftrightarrow (\hat{x}_2, \hat{y}_2) \in \hat{F}$, 和 $\hat{x}_1 \in \hat{c}_0 \Leftrightarrow \hat{x}_2 \in \hat{c}_0$;
- 3) $\sum(PN) = (B, E; F', c_0)$ 是对 $PN' = (\hat{S}, \hat{T}, \hat{F}, \hat{c}_0)$ 的等价化简。

3.5.6 应用实例

下面以插孔任务为例, 说明任务协调方法。

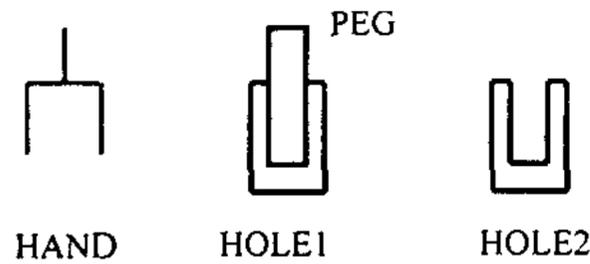


图 3.16 插孔任务

如图 3.16 所示, 在任务空间中的个体包括: 机械手 (HAND), 插件 (PEG), 插孔 1 (HOLE1), 和插孔 2 (HOLE2)。

$D = \{HAND, PEG, HOLE1, HOLE2\}$ 。

操作状态由下列谓词描述:

- HANDEEMPTY: 机械手是空闲的;
- PEGFREE(p): 插件是自由的 (未被机械手抓持);
- HOLDING(p): 机械手抓持插件 p;
- PEGINHOLE(p, h): 插件 p 位于插孔 h 中;
- HOLEEMPTY(h): 插孔 h 中没有插件。

因为在这个任务中, 机械手是唯一的, 所以关于机械手的状态和关系谓词都省略了机械手的变量。机械手的动作以 STRIPS 形式的 F 规则描述如下, 其中先决条件记

为 PRE, 删除表记为 DEL, 添加公式记为 ADD:

- 1) grasp(p)
PRE: HANDEEMPTY, PEGFREE(p)
DEL: HANDEEMPTY, PEGFREE(p)
ADD: HOLDING(p)
- 2) release(p)
PRE: HOLDING(p)
DEL: HOLDING(p)
ADD: HANDEEMPTY
- 3) part(p, h)
PRE: HOLDING(p), PEGINHOLE(p, h)
DEL: PEGINHOLE(p, h)
ADD: HOLEEMPTY(h)
- 4) mate(p, h)
PRE: HOLDING(p), HOLEEMPTY(h)
DEL: HOLEEMPTY(h)
ADD: PEGINHOLE(p, h)

任务空间的 Pr/T 模型如下:

表 3.1

位置	谓词	变迁	动作
P ₁	HANDEEMPTY	t ₁	grasp(p)
P ₂	PEGFREE(p)	t ₂	release(p)
P ₃	HOLDING(p)	t ₃	part(p, h)
P ₄	PEGINHOLE(p, h)	t ₄	mate(p, h)
P ₅	HOLEEMPTY(h)		

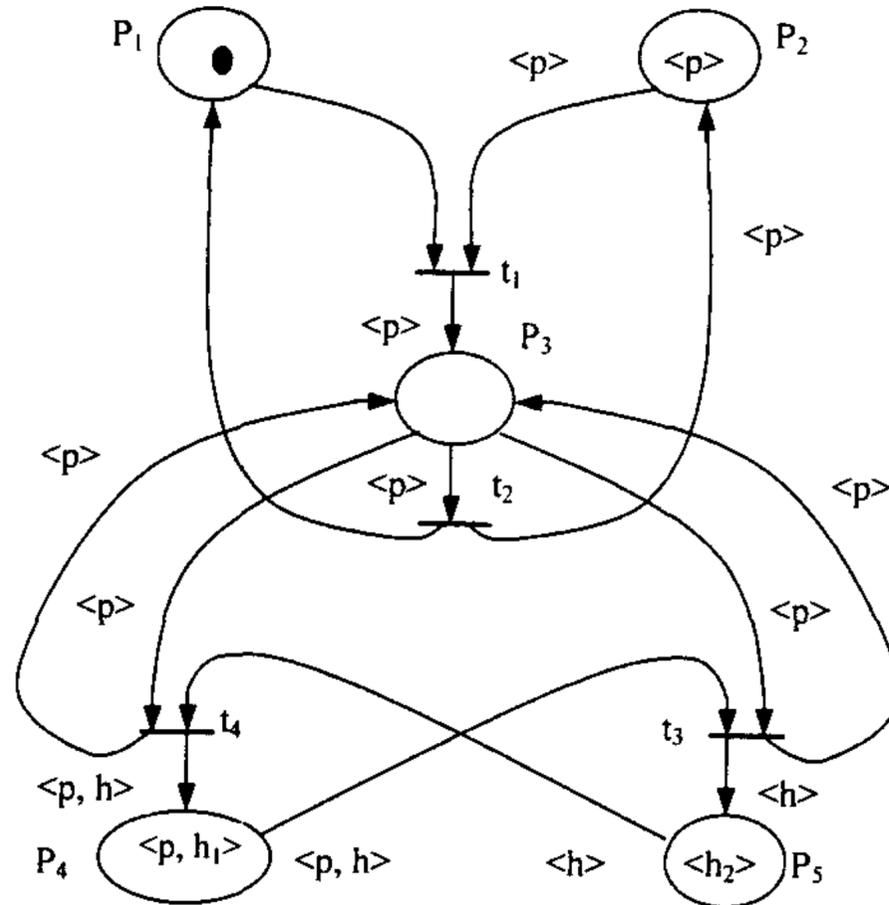


图 3.17 任务空间 Pr/T 网模型

插孔任务过程如下图所示:

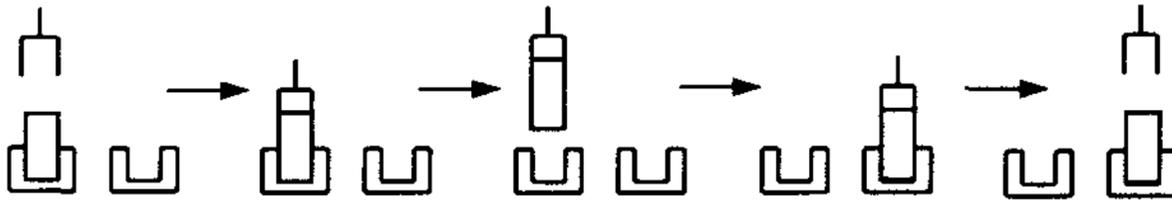


图 3.18 插孔任务过程

根据上述过程，产生动作序列如下：

{grasp(PEG), part(PEG, HOLE1), mate(PEG, HOLE2), release(PEG)}

生成的 Petri 网任务模型如下：

表 3.2

位置	谓词	变迁	动作
P ₁	HANDEEMPTY	T ₁	grasp(PEG)
P ₂	PEGFREE(PEG)	T ₂	release(PEG)
P ₃	HOLDING(PEG)	T ₃	part(PEG, HOLE1)
P ₄	PEGINHOLE(PEG, HOLE1)	T ₄	mate(PEG, HOLE2)
P ₅	HOLEEMPTY(HOLE1)		
P ₆	HOLEEMPTY(HOLE2)		
P ₇	PEGINHOLE(PEG, HOLE2)		

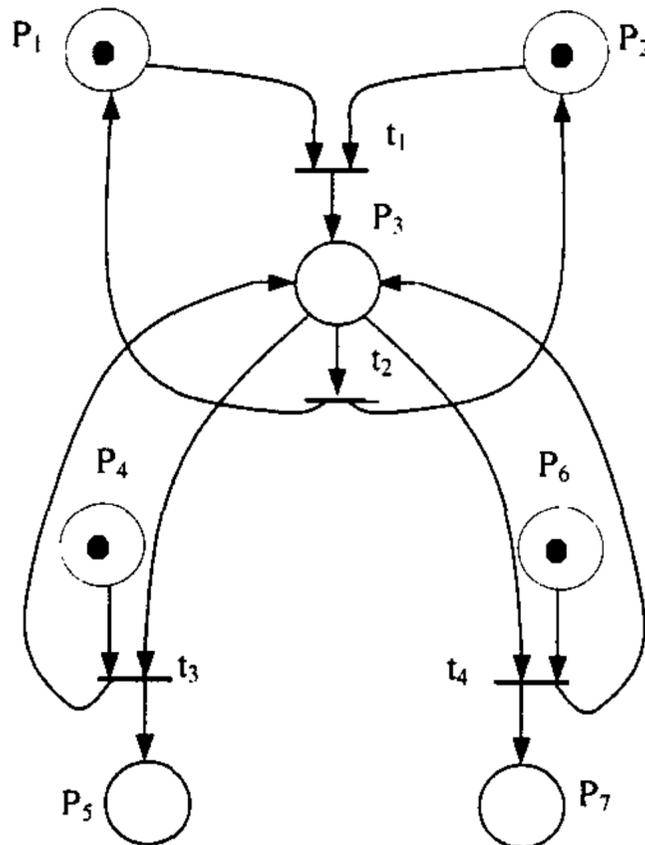


图 3.19 任务 C/E 网模型

3.6 小结

本章提出了一个基于 Petri 网的监控式大时延遥操作系统框架。该框架的特点在于通过基于 Petri 网的分层任务模型，提供了操作员多层次介入机制，其主要功能包

括：任务协调和任务监控。在系统框架的思想指导下，本文深入研究了基于 Petri 网的任务建模和任务协调方法。针对自由运动，提出了一个基于路径关键点提取的动作级任务建模方法，该方法为动作级遥编程提供了基础；提出了一种通过基于子任务的 Petri 网系统模型的变迁细化来建立任务的分层 Petri 网描述的方法。通过插孔任务实例说明了分层 Petri 网任务建模方法。提出了一种基于任务空间的谓词/变迁网模型的任务协调方法，对其中的状态识别、动作识别，以及任务形成等问题进行了研究，该方法为任务级遥编程提供了基础。最后，通过插孔任务实例对任务协调方法进行了说明。

第四章 基于遥编程的大时延遥操作系统

本章在遥编程概念的基础上，给出了一个基于动作级遥编程的大时延遥操作实验系统，并介绍了地面环境的大时延遥操作仿真试验。试验结果表明，在几秒（试验中设为 3 秒）时延条件下，系统能够平稳地完成高精度遥操作任务。

本章首先介绍了遥编程的基本原理，在此基础上，给出了大时延遥操作系统的设计思想。系统从功能上主要分为三个部分：人机接口系统、通信系统，以及遥机器人系统。本章分别对这三个部分进行了详细介绍。最后，描述了大时延遥操作仿真试验——插方孔试验，给出了试验结果。

4.1 遥编程基本原理

在遥编程系统中，利用对遥操作环境的先验知识构造虚拟环境，操作员面向虚拟环境进行操作，同时获得即时的视觉和运动觉反馈；系统监视操作员的动作，并将操作员的动作转化为符号命令程序，发往遥机器人执行；遥机器人接受命令，半自主地连续执行，并不断向操作员报告执行状态，当检测到误差或意外情况发生时，遥机器人自动停机，并等待操作员的误差恢复命令。

在遥编程系统中，存在两个反馈回路，一个是在操作员的主控站，另一个在遥机器人工作站。在操作员主控站，通过计算机仿真技术和虚拟现实技术，使操作员获得即时反馈信息，从而进行连续操作。主控站的另一项功能是根据操作员的动作，自动生成符号命令程序，符号命令的抽象程度，取决于遥机器人的智能程度，可以是自由位置控制的动作级命令，可以是接触作业时的受约束运动命令，也可以是任务级的程序；遥机器人是具有较高执行性能和一定容错能力的半自主机器人，能够连续执行主控站的命令，并检测当前的运行状态，当由于主控站的虚拟模型和实际模型出现偏差而造成执行误差，或由于操作环境的不确定性发生意外时，遥机器人会自动停机并向主控站报告。遥编程概念图见图 4.1。事实上，遥编程系统一般是工作在连续操作状态，在控制过程中不受时间延迟的影响，只在发生意外，需要进行误差恢复时受时间延迟的影响。

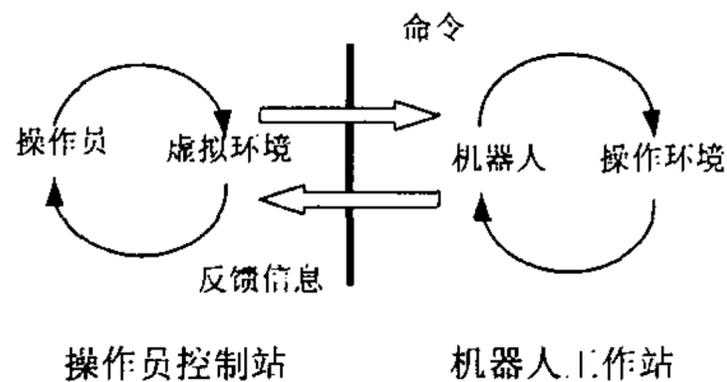


图 4.1 遥编程系统概念结构图

根据遥机器人编程语言的不同，遥编程主要分为三类：

- 1) 示教式编程，编程的结果是一段连续的运动轨迹。系统记录虚拟机器人运动中每一个时刻的位置、速度等信息，遥机器人根据这些信息，将完全复现虚拟机器人的运动。
- 2) 动作级编程，编程的结果是动作级语句。动作级语句描述机器人运动的基本动作，如：“x 关节转动 x 角度”，“末端沿直线运动到 x 位置”，等等。对于每一条动作语句，系统记录其起始位置和终止位置，识别出相应的动作类型，并确定动作语句的参数。遥机器人根据动作的类型和参数，自行规划连续运动轨迹。
- 3) 任务级编程，一个任务是由动作级语句，以及相关的参数设置语句，条件判断语句等构成的，包括特定的分支、循环结构的程序段。对于一个任务，需要指定其起始条件和终止条件。遥机器人首先对任务进行解释，然后通过任务协调，以自主的方式执行任务。

根据任务的复杂性和遥机器人的自主能力的不同，不同的遥编程方式各有其适用范围。一般地说，对于简单的任务采用示教方式和动作级编程比较方便快捷，而对于接触操作等比较复杂的任务，对位置控制和力控制精度要求较高，采用任务级编程更加合适。

4.2 系统设计思想

在这一节，首先对大时延遥操作系统的总体设计思想和系统功能结构进行了介绍，然后，针对遥编程涉及的两个关键问题：程序自动生成和虚拟环境误差校正，进行了初步的研究，并给出了本文的解决方法。

4.2.1 分段复合控制方案

为了有效克服大时延对遥操作系统的影响，并确保遥操作的连续性和准确性。本文提出了一种分段复合控制方案。分段复合控制的基本思想是在遥操作的不同阶段，根据对操作精度和平稳性的要求不同，采取不同的控制策略，从而提高系统完成作业任务的综合性能。本文根据操作器与操作对象的相对位置关系，把操作空间划分为三个区域：

- 1) 自由运动区，它由操作器趋近操作对象过程中的大范围自由运动空间构成。操作器在这个区域一般不会发生同操作对象的碰撞，对实际运动轨迹的控制精度要求不高，因而允许以较快的速度连续运动；
- 2) 准操作区，在这个区域中，操作器在小范围内向操作对象逼近，可能发生同

操作对象的碰撞。在准操作区内，操作器的运动轨迹需要严格控制，对速度的要求低于对安全性的要求；

- 3) 操作区，它是操作器相对操作对象很近，往往发生接触的一个小区域，操作器将在这个区域完成操作作业。在这个区域，运动空间往往是有约束的，除了对位置控制精度的要求，还包括对力控制精度的要求。在操作区，意外、故障随时可能发生，操作的安全性十分重要。

在上述三个区域，根据操作器运动方式的不同，本文对每个区域内的运动采取不同的控制方式：

- 1) 自由运动区——在线控制。在自由运动区，操作员连续控制虚拟机器人，系统根据操作员的动作连续生成控制指令并即时发送给遥机器人执行，从而实现虚拟机器人和遥机器人之间的主从控制。遥机器人处于位置控制方式，以较快速度运行；
- 2) 准操作区——离线控制。在准操作区，解除虚拟机器人和遥机器人之间动作的同步。操作员针对虚拟机器人进行连续操作，并利用预显示和碰撞检测技术，对已生成指令的运动轨迹进行检验，如果不理想，可以取消并重新生成，直到指令得到确认，才交付给遥机器人执行。机器人一般处于受约束运动控制状态，以较慢速度运行；
- 3) 操作区——单步监控。操作员每次发送一条命令，命令中附加严格的速度条件和结束条件。通过遥现和增强现实技术，使操作员能监视遥机器人的运动并准确判断机器人的运行状态。机器人一般处于力控制模式。

本文基于遥编程概念，研制了一个大时延遥操作实验系统。系统从功能结构上主要分三个部分：人机接口系统，通信系统，和遥机器人系统。如下图所示：

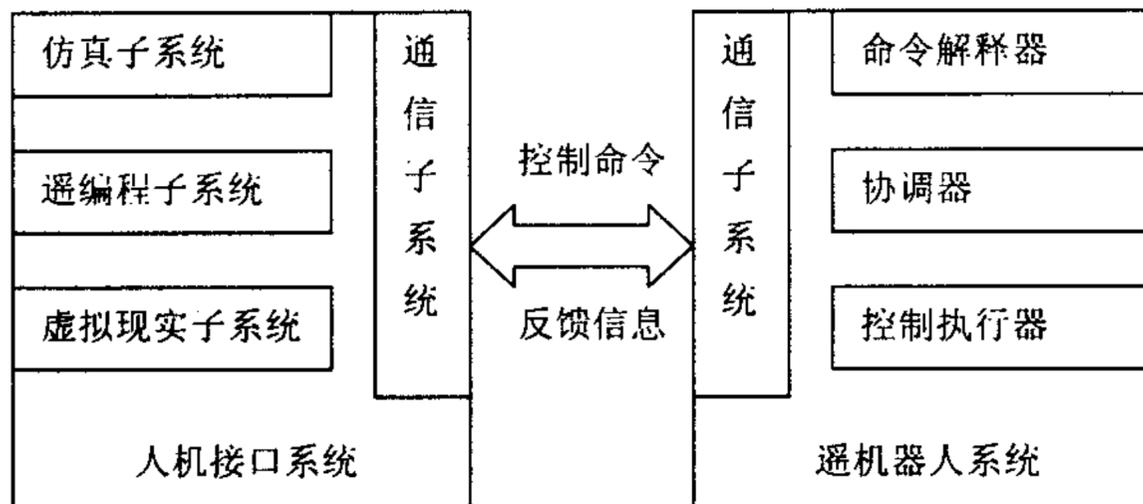


图 4.2 遥操作系统结构

其中，人机接口系统实现操作员交互界面和程序自动生成功能，通讯系统负责控制命令和反馈信息的传输，遥机器人系统是一个基于离散事件反馈的局部自主机器人系统。本文根据结构化的操作环境模型构造了虚拟环境，基于机械手的运动学模型构

造了虚拟机器人。在仿真子系统中集成了遥机器人的运动学模型、命令解释功能以及路径规划功能。在遥操作过程中，操作员面向虚拟环境连续地控制虚拟机器人，系统根据操作员的动作生成机器人动作控制语句。程序自动生成功能是基于动作级语言实现的。通信系统可以将控制语句连续地发送给遥机器人，同时连续地接收反馈信息。遥机器人系统接收控制语句后通过命令解释、协调，自主运行，并检测运行状态，当发生意外时可以自动停机，并报告状态。

4.2.2 程序自动生成方法

在动作级遥编程中，自动生成动作语句序列的过程，由轨迹分段和动作匹配两部分构成。所谓轨迹分段是指将虚拟机器人的连续运动轨迹沿时间轴进行分割。在此基础上，根据每一段轨迹的特点选择适当的动作语句，实现动作匹配，比如当轨迹只涉及位置变化时，选择平移动作，而当轨迹只涉及姿态变化时，选择旋转动作。动作匹配方法取决于具体的动作级语言。程序自动生成中最关键的问题是如何进行轨迹分段。

在本文的第三章（见 3.3.3 节），给出了一个路径关键点提取方法。路径关键点提取的思想可以用于轨迹分段。然而，轨迹分段同任务建模中的路径关键点提取问题有所不同的是：a) 轨迹分段是在虚拟机器人运动过程中实时进行的，对轨迹分段而言，只有当前时刻及其以前的轨迹信息；b) 除了分段线性逼近的误差因素外，轨迹分段的准则要考虑多方面的因素。确定轨迹分段的准则需要考虑两个方面的因素：

1) 轨迹段的相对控制误差

轨迹段的相对控制误差定义为： $E_1 = e_c / L$ ，其中 L 为轨迹段的长度， $e_c = f(e_r, e_t)$ ，是由机器人控制精度 e_r 和任务精度要求 e_t 决定的控制误差。一般 $e_r \leq e_t$ ，取 $e_c = e_r$ 。相对控制误差越小越好。

2) 轨迹段的线性逼近误差

线性逼近误差（记为 E_2 ）是由自动生成的动作语句决定的机器人理想运动轨迹和虚拟机器人的运动轨迹间的误差。一般，希望遥机器人的运动轨迹尽可能准确地再现虚拟机器人的运动轨迹，因此线性逼近误差越小越好。通过缩短轨迹段的长度（或采样间隔），可以减小 E_2 。

可以看出，对轨迹段的相对控制误差和线性逼近误差的要求是互为矛盾的。轨迹分段的准则应该体现二者间的权衡。上述讨论没有对虚拟机器人的运动轨迹进行限制，而实际上，虚拟机器人的运动轨迹往往不是任意的。本文通过对虚拟机器人的运动轨迹进行限定，实现了一种简单的轨迹分段方法。

假定虚拟机器人的运动轨迹本身是分段线性的，在每一段内速度基本保持一致。

实践表明,上述假定具有一定的合理性。因此,本文的轨迹分段是通过检测速度的变化实现的。机器人的速度 \bar{V} 对应操作空间中的点,速度的变化由两个速度间的距离表征 $dV = \|\bar{V}_1 - \bar{V}_2\|$ 。轨迹分段是沿时间轴对轨迹的速度进行扫描和聚类的过程。对当前轨迹段,设轨迹段的平均速度为 \bar{V}_m ,当前时刻速度为 \bar{V}_i ,当 $dV = \|\bar{V}_i - \bar{V}_m\| \geq \varepsilon$ (ε 为速度变化门限),结束当前轨迹段,否则,更新 \bar{V}_m ,继续轨迹段的生长。

4.2.3 虚拟环境误差校正方法

虚拟环境的误差主要来源于两个方面:一方面,遥机器人的运行相对于理想轨迹或预定目标存在偏差,这就造成了虚拟机器人和真实机器人之间的误差;另一方面误差来源于实际运动的操作对象和虚拟操作对象之间的差异。根据误差来源的不同,虚拟环境的误差校正分为两个方面:虚拟机器人的误差校正和虚拟操作对象的误差校正。

1) 虚拟机器人的误差校正

通过光电码盘等传感器,可以直接测量得到遥机器人的状态信息,对于机械手,其状态可由关节角或末端位置来表达。设遥机器人的状态矢量为 X ,虚拟机器人的状态为 \tilde{X} ,则虚拟机器人相对于遥机器人的误差为: $e = \|\tilde{X} - X\|$,当误差超过预定的阈值, $e \geq T$,将虚拟机器人的状态更新为遥机器人的状态。

2) 虚拟操作对象的误差校正

同虚拟机器人不同的是,操作对象的位姿一般无法通过传感器直接测量得到,需要利用视频图像,通过目标定位技术来获取。对于虚拟对象的误差校正可以通过单目计算机视觉定位的方法实现。假设在结构化环境下,物体的几何信息和特征结构都是已知的,摄像机已经标定。设待求物体的位姿齐次矩阵为 M ,取物体上非共面的4个特征点 p_1, p_2, p_3, p_4 ,已知特征点的齐次坐标为 $P_i = (x_i, y_i, z_i, 1)$,特征点在图像中对应的像素点的齐次坐标为 $Q_i = (u_i, v_i, 1)$,则存在如下关系:

$$Z_{ci} \cdot Q_i = C \cdot M \cdot P_i,$$

其中 C 是摄像机对应的投影矩阵, Z_{ci} 为 p_i 相对于摄像机的深度信息, $i=1,2,3,4$ 。将四个线性方程联立求解,可以确定物体相对于世界坐标系的位姿矩阵 M 。

在本文所涉及的试验中,操作环境基本是静态的。因此,误差校正主要围绕虚拟机器人进行。

4.3 人机接口系统设计

基于遥编程和分段复合控制的思想，在人机接口系统（HCI）中主要实现了以下功能：

- 1) 实现操作员控制接口，操作员通过操作界面进行在线方式或离线方式的编程；
- 2) 符号命令生成，根据操作员的动作生成机器人动作语句；
- 3) 对机器人及其工作环境进行仿真，仿真主要针对机械手进行运动学仿真，仿真结果通过虚拟环境显示出来。
- 4) 对遥机器人操作环境进行遥现，反馈信号主要来源于传感器信号，遥现将利用反馈信号实现增强现实显示。

下面，首先介绍 HCI 的软件结构，然后分别对其中关键部分的实现进行介绍。

4.3.1 系统软件结构

根据系统的功能需求，HCI 的软件结构主要包括以下对象：人机界面对象、虚拟机器人对象、机械手运动学仿真对象、图形系统、编程部件，以及通讯部件。

- 1) 人机界面对象，主要包括四个子部分：手控制器、图形界面、编程界面、通讯界面。通过手控制器操作虚拟机器人；通过图形界面实现虚拟操作环境的视觉反馈和真实操作环境的遥现；通过编程界面进行离线编程；通过通讯界面实现同遥机器人系统的通讯；
- 2) 虚拟机器人对象。虚拟机器人驱动图形系统中的虚拟机器人的几何模型，内嵌机械手运动学仿真对象。有三种运动方式：直接操纵方式、命令仿真方式、反馈遥现方式；
- 3) 图形系统，主要实现两个功能：虚拟机器人运动预显示，真实机器人遥现；
- 4) 编程部件，有两种编程方式：a) 根据虚拟机器人的运动自动编程；b) 通过编程界面手动编程；
- 5) 通讯部件，实现双向的信息传输。

在 HCI 中，操作员直接与人机界面对象进行交互，通过人机界面对象实现对其它各部件的控制。虚拟机器人部件在不同的模式下，接受三个方面的控制：手控制器、编程部件和通信部件。编程部件接收虚拟机器人的状态信息生成机器人控制语句，并交付给通信部件。通信部件实现控制语句的发送和反馈信息的接收。HCI 的软件结构如图 4.3 所示。

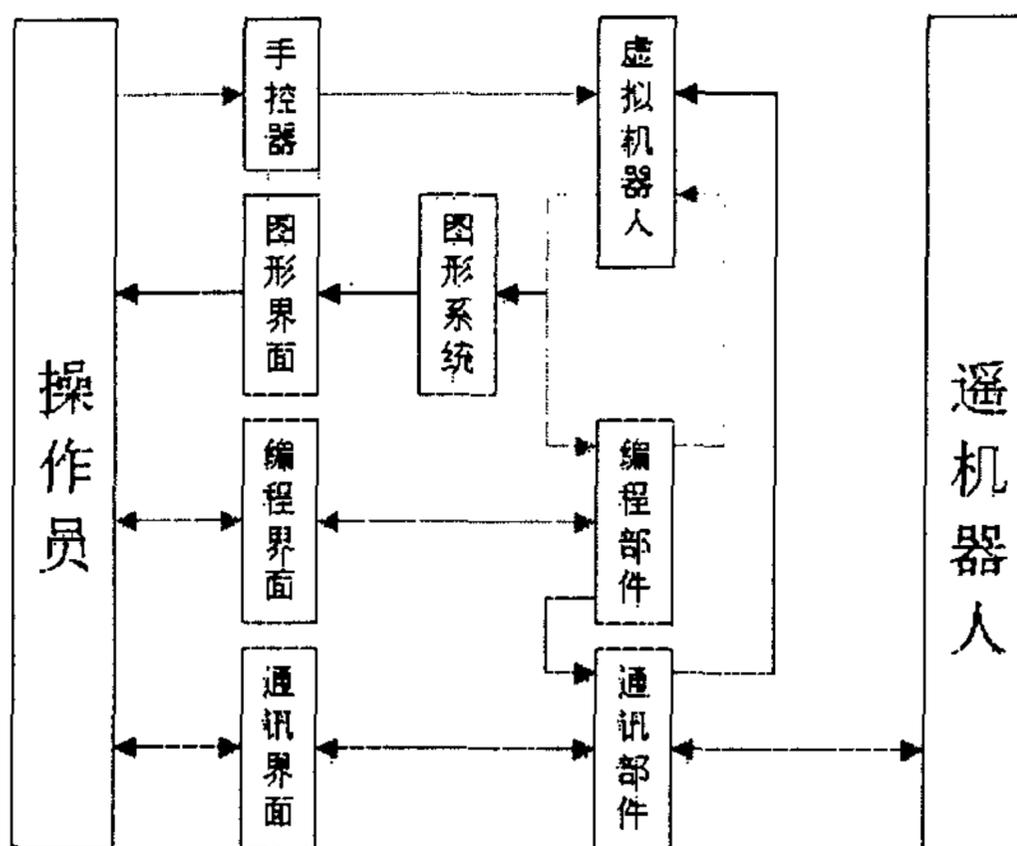


图 4.3 HCI 软件结构图

4.3.2 虚拟环境的构造

在虚拟环境中，构造了操作环境的几何模型：包括操作台、操作工件等，同时构造了遥机器人的几何模型。由手控器给出机器人的关节或末端位置，通过运动学计算驱动机器人的几何模型，从而实现虚拟机器人的运动。在此基础上，本文的工作主要包括以下内容：

- 1) 实现多种运动控制方式。在虚拟环境中，操作员可以选择虚拟机器人的关节或者末端进行控制，从而实现关节、末端两种控制方式；
- 2) 碰撞检测。本文采用基于分层结构的包围盒检测方法，实现了虚拟机器人和操作环境（包括操作工件）之间的快速碰撞检测。碰撞检测的作用一方面用以辅助操作员的操作，通过在碰撞时给出提示来弥补操作员的观察准确性的不足，从而避免明显的操作失误；另一方面在命令预显示过程中，对碰撞情况报警，有助于对命令合理性的检验；
- 3) 虚拟坐标。为了辅助操作员对操作方向进行控制，防止操作员在视点漫游过程中“迷失”，本文构造了虚拟坐标等合成对象，可以根据需要随时给出包括：世界坐标系、工作坐标系、机器人基坐标系和控制单元的局部坐标系在内的多种坐标系的坐标轴。

4.3.3 虚拟机器人的构造

本文根据 PUMA562 机械手的运动学模型构造虚拟机器人。PUMA562 机械手的

连杆参数如下:

表 4.1 连杆参数表

连杆 i	a_{i-1} (mm)	α_{i-1}	d_i (mm)	θ_i	关节变量范围
1	0	0	0	0	-160~160
2	0	-90	144.06	0	-205~45
3	431.815	0	0	0	-45~225
4	-20.41	90	432.63	0	-110~170
5	0	-90	0	0	-100~100
6	0	90	0	0	-266~266

PUMA562 机械手的各连杆坐标系确定如下:

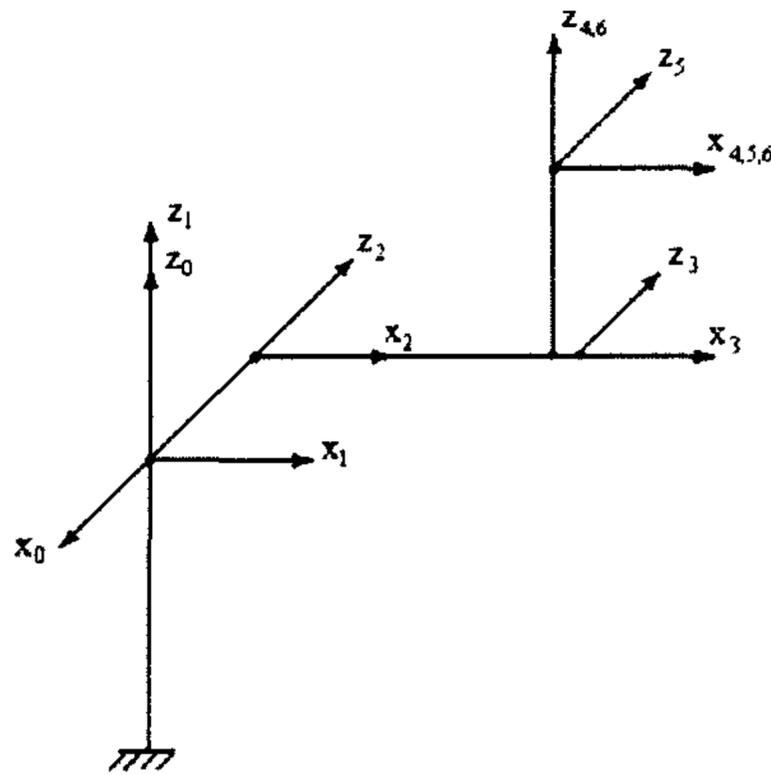


图 4.4 PUMA562 运动学模型

4.3.4 虚拟机器人的运动控制

通过人机交互界面实现了对虚拟机器人的多种运动控制方式,运动控制主要有以下几个方面:

- 1) 运动部件,通过运动部件的选择,可以分别对机械手各个关节和工具进行控制;
- 2) 运动坐标系,可以使机械手分别在世界坐标系、基坐标系、工作坐标系、机械手局部坐标系内进行运动;
- 3) 运动方向,可以约束机械手在不同的轴向或平面上进行运动;
- 4) 运动方式,可以约束机械手进行平移或旋转运动;

操作员可以根据需要将以上四个方面进行不同的组合,从而实现多种不同的控制

方式，提高了操作员操作的方便性。不同控制元素的组合遵从下面的规则：

- 1) 在关节控制状态（运动部件是某个关节）下，只能在关节局部坐标系内绕 Z 轴作旋转运动；
- 2) 在末端控制状态（运动部件是工具）下，可以在任何坐标系下，在任何轴向或平面作旋转和平移运动。

4.3.5 编程子系统

系统根据特定时间段内虚拟机器人的起始和结束状态，结合当前的运动控制状态，自动生成动作语句。此外，通过附加的命令面板，使操作员可以手工输入、编辑各种语句。

语句的构成基于本实验室开发的 OSRL 动作级语言，主要分为三类：

- 1) 动作语句，包括关节运动和末端运动两类。每一条动作语句的主体内容包括：动作标识号和动作参数。不同的动作拥有不同的参数结构。可以对动作语句附加约束条件，包括：速度约束条件和力约束条件；
- 2) 参数语句，通过参数语句可以设置机器人的运行速度、调整机器人的控制模式（位置控制模式和力控制模式），以及向机器人发送监控事件；
- 3) 任务调用语句，通过调用预先开发好的任务，操作员可以实现对机器人特定任务的监控。

在对轨迹分段的基础上，根据当前的运动控制状态，确定语句的类型；根据机械手的初始状态和终止状态，确定语句的参数。假设：在一条语句生成过程的自始至终，系统控制状态保持一致（中间没有关节控制和末端控制的切换）。那么，确定语句类型的规则如下：

- 1) 关节运动指令：如果当前处于关节控制状态，则生成关节运动语句；
- 2) 末端运动指令：如果当前处于末端控制状态，则生成末端运动语句。

具体语句生成规则如下：

- 1) MOVE：如果终止状态和初始状态相比，有两个以上关节角发生变化，则生成该语句。由终止状态下的关节角构成参数；
- 2) DRIVE：如果终止状态和初始状态相比，只有一个关节角发生变化，则生成该语句。由发生变化的关节及其终止角度构成参数；
- 3) DRAW：如果终止状态相对初始状态，只有位置发生变化，则生成该语句。由终止状态的位置构成参数；
- 4) ROTATE：如果终止状态相对初始状态，只有姿态发生变化，则生成该语句。由终止状态的姿态构成参数；

- 5) MOVES: 如果终止状态相对初始状态, 位置和姿态都发生了变化, 则生成该语句。由终止状态的位姿构成参数。

4.3.6 编程界面设计

编程界面需要实现以下功能:

- a) 面向机器人语言, 实现控制语句的生成;
- b) 实现机器人参数的设定, 包括: 控制系统参数设定和控制模式转换 (力/位置控制模式的转换);
- c) 实现对机器人局部自主任务的调用;
- d) 记录并显示所生成的控制语句。

根据以上的功能需求, 编程界面的软件功能如下:

- a) 生成语句。包括两种方式: 自动生成方式和手工添加方式;
- b) 编辑语句。对已有动作级语句进行修改, 实现 with 和 until 语句;
- c) 管理语句。实现语句的删除、预览和交付执行。

编程子系统的概要设计如下:

1) 语句的定义和结构。

语句包括实语句和空语句两种。实语句是指 OSRL 中的语句结构。空语句指不包括任何信息的语句结构。实语句分三种类型: 机器人动作控制语句、参数设置语句 (赋值语句) 和任务调用语句。动作控制语句由三部分构成: 动作语句主体、速度条件和结束条件。

每一条动作控制语句都包含一个机器人的起始状态和一个终止状态。在语句序列中, 前后相邻的两条动作控制语句, 前一条的终止状态和后一条语句的起始状态相同, 称之为动作的连续性, 所有关于语句的操作都不能破坏动作连续性原则。

2) 语句管理规则

语句序列由实语句和空语句构成。一个语句序列中只能而且必须在结尾处包含一个空语句。针对语句序列中的语句有三种基本操作: 选择、添加和修改。此外, 还包括预览和交付执行。

- a) 语句序列中的每一条语句都可以处于被选择或未被选择两种状态。在同一时刻只有一条语句可以处于被选择状态。缺省状态下, 结尾空语句处于被选择状态。添加、修改操作不影响语句的选择状态。删除当前语句后, 选择状态自动向后移动一条语句。预览和交付执行操作一定程度上同当前的选择状态无关, 也不影响当前的选择状态。
- b) 添加语句操作将在选中语句之前插入一条新的实语句。动作语句和任务调用语句只能添加于空语句之前, 参数设置语句可以添加在任意语句之前。目前,

任务调用语句是不可预测的，所以任务调用语句只能在结尾空语句之前，也就是说，任务调用语句后不能插入任何其它语句。

- c) 除结尾空语句外，其它被选择的语句都可能被删除。动作语句只有处于结尾空语句之前的才能被删除。可以删除任意位置的参数设置语句。
- d) 除任务调用语句和空语句外，其它语句都可能被修改。可以修改任意位置的参数设置语句。动作语句的修改分两个部分：动作参数和附加条件（until 和 with 字段）。任意位置动作的附加条件都可以被修改，只有结尾空语句前的动作语句的动作参数可以被修改。

预览和交付执行功能同基本操作不同的是，将使遥编程系统进入某种状态，可以在后台持续执行一段时间，而不是即刻在前台完成的。

3) 预览功能定义

预览将通过仿真系统和图形系统，预先显示给定范围内语句的执行过程，以便进行路径验证和碰撞检测。预览对语句序列没有影响。在预览状态下，将禁止除选择语句外的其它操作。

4) 交付执行功能定义

交付执行功能将通过通信子系统，将给定范围的语句发送给遥机器人。已交付的语句将从语句序列中删除。

5) 与虚拟机器人通信

与虚拟机器人的通信内容主要有：

- a) 向虚拟机器人传送语句，以供仿真预览；
- b) 向虚拟机器人传送选中语句的初始状态；
- c) 接收虚拟机器人的状态，作为自动生成语句的终止状态。

4.3.7 多种图形显示模式

在操作员界面，实现了多种图形显示模式，其中包括：遥编程显示模式、任务预览显示模式，以及操作环境遥现模式。

- 1) 遥编程模式，操作员控制虚拟机器人进行连续运动；
- 2) 任务预览显示模式，根据遥机器人系统模型构造了机器人仿真器，仿真器的主要功能包括：命令解释、轨迹规划、运动学计算。通过仿真运行，虚拟机器人可以执行已经生成的语句序列，演示机械手的理想运动轨迹；
- 3) 遥现模式，假定在操作环境中，物体的几何形状保持不变，只有位置和姿态变化。因此，本文根据来自操作现场的机器人和操作对象的位姿信号，以图形的方式重构操作环境，从而实现操作现场的视觉遥现。通过遥现，操作员可以对操作现场进行监控，并在出现意外情况下进行误差诊断。

4.3.8 系统硬件实现

人机接口系统是基于微机平台实现的。由 PII400 和 Tornado3000 三维图形加速卡组成微机图形工作站；由 V8 头盔和 FASTRAK 定位跟踪器实现立体显示和视点漫游。由 SPACE MOUSE 三维鼠标作为手控器，输入控制信号。

4.4 通信系统设计

本文基于 TCP/IP 协议，以客户/服务器模式建立操作员控制站和遥机器人的通讯。本文在通讯系统的设计上主要有如下特点：

- 1) 实现逻辑上独立的两个信道，一个用于传输控制命令和反馈状态报告，本文称之为命令信道；一个用于传输操作现场的传感器信息，本文称之为数据信道。两个信道的通信方式有所差异，命令信道传输的特点是传输内容具有突发性，传输单元的长度不定；数据信道的传输具有连续性，且数据单元长度相同。针对两个信道的不同特点，本文分别设计了两套通信协议；
- 2) 通过缓存技术，实现了命令的连续发送和接收。根据直接控制方式和离线编程的需要，命令信道需要能够连续地传输多条命令语句，而遥机器人只能逐条接收命令语句执行，因而出现了遥机器人对命令语句的执行速度和命令语句传输速度之间的匹配同步问题。本文通过在命令的接收端设立缓存区，寄存连续接收的多条命令，遥机器人以管道的方式读取缓存区中的命令。

在这一部分，本文将介绍通信协议的设计，操作员主控站的通信部件设计，以及遥机器人工作站的通信服务器设计。

4.4.1 通信协议设计

通讯系统设计中，一个关键问题是命令协议的设计。命令通信有以下几个特点：

- 1) 通信内容的主体是字符串
- 2) 通信内容的长度可变
- 3) 通信内容以命令语句为单位，可以由多条语句构成

为了满足命令通信的需要，本文设计了如下的通信协议：

- 1) 通信正文：由若干控制语句构成的字符串。
- 2) 信头：为控制通信的特定结构，说明通信正文的长度，标识一次通信的结束等。
- 3) 通信单元：包括一个信头，以及相应的通信正文。通信单元的传输过程如下：发送方先发送信头，接收方根据信头的信息，判断通信正文的长度，和一次

通信的结束，再接收正文。

- 4) 一次通信：一次通信由若干通信单元构成。在一次通信过程中，服务器连续接收通信单元，不发送响应信息。在最后一个通信单元中标识一次通信结束，服务器对本次通信内容进行处理（交付机器人执行命令），然后向操作员控制站返回应答信息。

4.4.2 遥机器人通信服务器设计

通信服务器需要完成如下功能：

- 1) 从主控站接收控制语句
 - 接收通信控制、运动控制语句（包括动作控制语句、参数设置语句和任务调用语句）等多种语句
 - 能够同时接收多条运动控制语句
 - 服务器在连续接收多条控制语句过程中，需要控制语句流量
- 2) 向主控站反馈信息
 - 应答通信控制信息
 - 报告每一条语句的执行状况
 - 连续发送机器人的状态数据
- 3) 向机器人传送命令语句
 - 每次只能传送一条语句。发送一条语句后，必须等待反馈信息，判断语句执行完毕后，再发送下一条语句
 - 根据机器人的反馈状态控制命令的传送
- 4) 从机器人获取反馈信息
 - 获取语句执行状况反馈
 - 获取状态数据

根据上述需求，通信服务器概要设计如下：

- 1) 通过管道同机器人命令解释程序通信。实现向机器人发送控制语句。通过两个半双工管道实现服务器和机器人的双向通信。
 - 服务器读取机器人命令解释程序的输出，获得应答信息，利用阻塞方式，实现服务器和机器人的同步。
 - 服务器写出一条语句后，直到获得应答信息，判断上一条语句执行结束后，才进行下一轮通信。
- 2) 通过共享存储去获取机器人的状态数据，包括关节角、位姿、夹具的开合状态，和机器人的运行状态。
- 3) 通过 socket 实现服务器和主控站的通信。

- 在控制信息通信中，把主控站作为客户端。主控站可以连续发送多条控制语句。
- 服务器根据头语句中的容量信息，判断是否接收其后的控制语句。
- 以语句为单位，向主控站报告机器人语句执行状况。对来自主控站的语句进行编号，报告内容包括编号，通过语句编号使能在主控站对具体语句的状态进行判断。
- 语句执行状况报告内容包括：语句结束状态，机器人操作状态。服务器需要在每条语句结束后，进行状态判断。

4.4.3 人机接口系统通信部件设计

通信部件具有如下功能：

- 1) 通过人机交互的方式建立通信，主要有以下内容：
 - 交互方式建立同遥机器人的连接
 - 交互方式注册/注销操作
- 2) 发送控制语句
 - 能够连续发送多条语句
- 3) 接收反馈信息，反馈信息分为两种：应答信息和遥测数据。
 - 应答信息是在建立通信和发送控制语句过程中，同遥机器人的交互信息。应答信息中包含了遥机器人状态的定性的报告，应答信息的传输是交互式的非连续的；
 - 遥测数据是由遥机器人连续发送，占有一条独立的信道。

4.5 遥机器人系统

系统从功能上主要由三部分构成：语言解释器，将机器人语言程序解释为控制指令；协调器，实现路径规划；控制执行器，在实现了关节空间伺服控制以及操作空间的力/位置混合伺服控制的基础上，实现了离散事件反馈控制器，从而实现了在自由运动空间和约束空间的局部自主控制（控制器具体设计可参见[160, 40]）。

遥机器人系统硬件上主要由 PUMA560 机械手、一台 Delta 3000 工控机和传感器组组成。Delta3000 工控机的配置是一块 MVME 197 作为系统的主板，其上运行 Unix 系统，另有两块 MVME 162 目标控制板，目标板上都运行实时操作系统 pSOS+，完成机械手的控制任务。传感器主要包括六维力传感器和机械手本体的光电码盘、电位器，分别给 Delta3000 工控机提供机械手的力和位置信息。

4.6 试验及结果

为了验证基于遥编程的大时延遥操作系统的性能,进行了以插方孔为代表的大时延遥操作试验。通过试验,主要考察系统在如下几方面的性能:

- 1) 自由运动情况下,系统的运行效率;
- 2) 受约束运动情况下,系统的位置控制精度;
- 3) 接触作业情况下,系统的控制精度和平稳性。

为了模拟大时延,本文在通信线路中人为设置了3秒的时间延迟,上行通路和下行通路各有3秒延迟。试验的遥操作任务是插方孔。插孔和插件的间隙配合精度为:0.02 ~ 0.04mm。插孔试验的大致过程是:在第一阶段,首先,机器人从初始位置运动到插件上方附近,然后机械手末端夹具对准插件,然后抓住插件并把插件从初始插孔中拔出;在第二阶段,机械手抓持插件运动到目标插孔上方附近,然后向下运动搜孔,当搜孔成功并调整好插件的姿态后,向下插孔。

针对上述三个方面的性能,本文分别进行了如下几项遥操作试验:

- 1) 趋近试验。以在线遥编程的方式,控制机械手从初始位置趋近插件,在这一过程中,操作员连续控制虚拟机器人运动,编程子系统自动生成一系列动作语句并即时发送给遥机器人执行。除非遥机器人发出误差警告,操作员并不监视遥机器人的运动过程;
- 2) 抓持试验。以离线编程方式,控制机械手使夹具对准插件并抓住插件,机械手处于位置控制模式;
- 3) 插拔试验。采取监控方式,以遥现方式监视机械手自主完成将插件从插孔中拔出和插孔的过程;
- 4) 综合试验。遵从分段复合控制的思想,完成整个插孔任务。

上述四项试验的结果如下:

- 1) 趋近试验中,操作员控制站平均发送动作语句数为4条,每条指令的平均运行时间为1s,虚拟机器人完成趋近过程所需平均时间为4s。遥机器人完成趋近过程相对虚拟机器人而言滞后约3s,总用时约7s。遥机器人的误差报警次数为0。如果对每条指令以单步方式发送,则预计运行时间为28s $((3+1+3)*4=28)$ 。这说明,在自由运动状态下,通过在线遥编程,大大提高了遥机器人的运行效率。
- 2) 抓持试验中,通过碰撞检测、预测显示等手段,对准及抓持的成功率为100%。这说明,在离线编程方式下,系统具有较高的位置控制精度。
- 3) 插拔试验,通过调用预先开发的插孔(拔孔)子任务,拔孔的成功率为95%,插孔的成功率为90%。

- 4) 综合试验, 通过对在线遥编程控制、离线编程控制和单步监控的综合运用, 系统成功地完成了插孔试验。

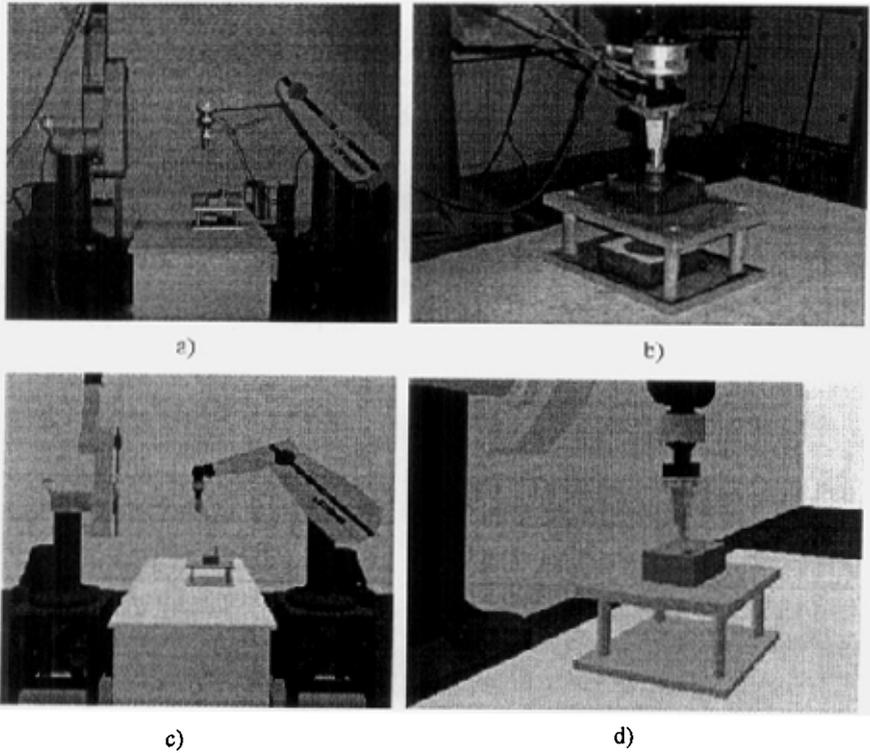


图 4.5 遥操作试验演示图

遥操作试验演示如图 4.5 所示, 其中 a)、b) 两幅图为真实操作环境图像, 而 c)、d) 两幅图为虚拟环境图。

4.7 小结

本文对遥编程技术进行了初步的研究, 实现了一个基于遥编程概念的大时延遥操作试验系统。本文详细介绍了系统的设计思想和各分系统的功能实现。本文介绍了插方孔大时延遥操作试验。试验结果表明, 本文研制的遥操作系统有效克服了大时延的影响, 能够完成较高精度的遥操作作业任务。同时, 本文的试验验证了基于遥编程概念的遥操作技术为大时延遥操作提供了一条有效的途径。本文研制的大时延遥操作系统于 2000 年底通过了“863”课题验收, 操作精度达 $0.02\text{mm} \sim 0.05\text{mm}$, 获得了评审委员会的高度评价。

第五章 快速碰撞检测技术研究

在这一章, 本文从空间遥科学实验的虚拟现实仿真的实际需要出发, 着重对物体数量众多的动态环境中的快速碰撞检测技术进行了研究。在借鉴已有的多体碰撞检测方法的基础上, 本文主要研究了基于空间分割的碰撞检测方法。首先, 针对物体均匀分布的情况, 提出了一个均匀空间分割的计算模型, 在此基础上, 给出了一个快速碰撞检测算法, 对算法的计算复杂性进行了分析, 并通过实验对算法的性能进行了验证; 对物体空间分布不均匀的情况, 本文提出了一类非均匀空间分割方法, 该方法由两部分构成, 首先基于空间分布密度, 进行粗略的非均匀空间分割, 然后在此基础上, 对每个单元格再施加均匀空间分割。在此框架下, 本文提出了两种非均匀空间分割方法, 基于投影直方图的空间分割方法和基于聚类的自适应空间分割方法。

5.1 概述

5.1.1 研究背景

空间遥科学试验仿真面临的一个困难就是难以在地面仿真空间的微重力环境。利用虚拟现实仿真, 为解决这一困难提供了一种途径。然而, 对于大数据量的遥科学实验, 虚拟现实仿真面临计算瓶颈的限制, 将难以满足实时可视化表达的需要。在微重力环境下的微粒沉淀仿真中, 为了实现对微粒沉淀过程的可视化, 需要将每个微粒作为实体, 仿真其动态运动过程。这里, 除了自由状态下的动力学计算外, 还必须计算微粒间的相互作用, 主要体现为碰撞及其响应。对于数以千计的微粒, 其间的碰撞检测成为系统的主要计算瓶颈。本文对于多体碰撞检测研究的目的就是为了解决由碰撞检测带来的计算瓶颈问题。

5.1.2 多体碰撞检测技术综述

设环境中存在 n 个运动物体, 如果对每个物体, 都要与其它 $(n-1)$ 个物体进行两两碰撞检测, 那么 n 个运动物体间的碰撞检测的计算复杂性为 $O(n^2)$, 当 n 较大时, 这将成为系统计算的瓶颈。加速多体碰撞检测可以从两个方面考虑: 一方面是利用物体的运动参数 (如速度、加速度), 通过预估碰撞时间, 在时间域上减少不必要的碰撞检测的次数; 另一方面是利用物体间的空间相关性, 减少相距较远的物体间的两两碰撞检测。基于这两种思想, 加速多体碰撞检测的方法主要有三种: 时序调度方案

(Scheduling Scheme)、基于排序的扫描排除方法 (Sorting-based sweep and prune), 以及基于空间分割的方法 (Spatial subdivision)。

1. 时序调度方案

已知物体运动速度、加速度的上界, 可以对物体间的最早碰撞时间 (碰撞时间下界) 进行估算。时序调度方法首先计算两两物体间的最早碰撞时间, 构造一个碰撞时间队列, 队列根据碰撞时间下界进行排序。碰撞检测只在队列的头元素所指示的最早碰撞时刻进行, 这样在不会漏掉实际的碰撞的前提下, 避免了在最早碰撞时刻之前进行不必要的碰撞检测, 同时, 碰撞检测只针对达到最早碰撞时刻的少数物体进行, 有效减少了参与碰撞检测的物体的数量。队列根据发生的关键事件 (如发生碰撞) 进行更新[85, 93]。

时序调度方案的问题在于, 维持一个最早碰撞时刻的队列需要较大的内存开销, 同时, 当碰撞发生频繁, 甚至在最小时间段内有多个物体发生碰撞时, 算法在碰撞检测和队列更新上的计算复杂性较高。另外, 时序调度方案需要预知物体运动的速度、加速度上限, 这对有些应用是无法获得的。

2. 基于排序的扫描排除方法

在 Cohen 等人研制的 I-COLLIDE 碰撞检测系统中[95], 采用了基于排序的扫描排除方法。算法的基本思想是用动态大小的轴向对齐包围盒 (Axis Aligned Bounding Box, 简称为 AABB) 作为物体的近似, 通过判断 AABB 是否接近, 来决定是否对两个物体进行准确的碰撞检测, 对于 AABB 不相交的物体, 可以肯定不会发生碰撞, 因此排除于碰撞检测之列, 从而减少参与碰撞检测的物体的数量。

算法的关键是如何快速地判断物体的 AABB 是否相交。基本的思路是将 AABB 按相邻关系进行排序, 然后以扫描的方式对相邻的 AABB 进行求交。由于在三维空间难以进行排序, 因此算法采用投影的方式进行降维处理。根据投影方式的不同, 具体有两种扫描排除方法:

- 1) 一维扫描排除方法。将物体的 AABB 向三个坐标轴 (x, y, z) 投影, AABB 的轴向投影是坐标轴上的一段区间, 然后将投影区间分别在三个轴向上进行排序, 沿排序的投影进行扫描, 判断投影区间是否交叠。可以证明, 两个 AABB 在三维空间相交当且仅当它们在三个坐标轴上的投影区间都交叠。当判断两个物体的 AABB 相交后, 再进行准确的干涉检测。扫描排除过程的计算量线性依赖于物体的数量。通常对 n 个区间进行排序的计算复杂性为 $O(n \log n)$, 但是当物体分布稀疏、运动速度相对较慢时, 可以利用时间相关性, 在前一帧排序的基础上采用插入排序或冒泡排序法, 从而可以把排序的计算复杂性降低为 $O(n)$ 。一维扫描排除方法的问题在于, 当空间中物体的分布较密, 且在两邻两帧之间, 物体的运动幅度较大时, 一维投影区间会出现大量的交叠,

而且排序的计算复杂性较高。

- 2) 2D 扫描排除方法。将物体的 AABB 向三个坐标平面 (xy, yz, zx) 投影, 投影为长方形区域。一般, 相对于一维投影区间, 二维投影区域相互交叠的较少, 从而当物体分布比较密集时, 效率较高。通过区间树 (范围树) 结构实现投影区域的排序和查找, 构造区间树的计算复杂性为 $O(n \log n)$ 。基于范围树, 一个长方形区域的查找的计算复杂性为 $O(\log n + k)$, 其中 n 为投影区域的数量, k 为相交的区域数目, 所以, 全部投影区域的相交测试的计算复杂性为 $O(n \log n + k)$ 。

3. 基于空间分割的方法

空间分割技术最初被用于光线跟踪算法[126-131]。在复杂场景的真实感绘制中, 光线跟踪算法必须要进行大量的光线与空间物体之间的求交运算, 求交运算成为影响光线跟踪算法效率的瓶颈。采用空间分割技术, 把复杂场景分割成离散的体元 (volume), 每个体元只包含较少数量的物体, 当光线经过某个体元时, 求交计算只对体元中的物体进行, 从而可以大大减少求交计算的次数。在复杂场景应用中, 空间分割提供了一种有效的空间搜索方法, 不仅用于光线跟踪, 而且可以用于碰撞检测等方面。已有的空间分割方法大致可以分为两类: 均匀空间分割 (uniform space subdivision) 和自适应空间分割 (adaptive spatial subdivision)。

基于空间分割来加速碰撞检测的基本思想是考虑到碰撞只在邻近的物体之间发生, 对于某一个物体, 要检测出与其发生碰撞的物体, 只需考虑该物体邻域范围内的物体, 而对于邻域外的物体则肯定不会发生碰撞。为此, 对物体所占据的空间进行分割, 分割成一系列的方体单元格 (voxel), 碰撞检测只在同处于一个单元格内的物体之间进行。在碰撞检测应用中, 常用的空间分割方法包括: 均匀空间分割、八叉树 (Octree) 分割, 二叉树 (BSP) 分割, 以及 Brep 索引。

1) 均匀空间分割

均匀空间分割把整个空间分割成大小相等的单元格 (voxel 或 cell), 把每一个物体放入它所占据的单元格中。在碰撞检测过程中, 对每一个物体, 需考虑与其相交的所有单元格 (最多八个)。均匀空间分割比较适合于物体尺寸相近, 分布均匀的情况。均匀空间分割的特点是物体的邻域检索效率很高。由于各单元格的尺寸相同且排列规则, 因此易于存储和检索, 当物体分布比较稀疏时, 采用散列表形式存储, 能获得 $O(n)$ 的存储效率和常数的检索效率。均匀空间分割遇到的关键问题是确定适当的单元格尺寸, 如果单元格的尺寸选择不当, 算法的计算开销会较大。对于空间中物体分布不均, 大小不同的情况, 均匀空间分割往往难以确定一个最优的单元格尺寸。

Turk[161]面向分子对接 (molecular-docking) 仿真, 提出了两种基于均匀空间分割的碰撞检测算法。实践表明, 采用均匀空间分割、散列表检索技术, 可以有效地解

决多体碰撞检测问题。采用均匀空间分割的存储检索结构，在每个计算周期中的更新代价较低。在 D. J. Kim, L. J. Guibas, 和 S. Y. Shin 提出的针对多个运动球体的快速碰撞检测方法中[94]，采用了一种分层均匀空间分割方法。该方法对相同尺寸的球体，根据均匀分布模型进行均匀分割，但考虑到由于存在不同尺寸的球体所导致的实际分布不均，采用分层结构，将不同尺寸的物体进行分离。该方法同时还采用了基于事件驱动的时序调度方法来控制碰撞检测的采样时间。

2) 八叉树分割

八叉树分割是一种被广泛采用的自适应空间分割方法。首先，把整个空间作为一个沿坐标轴向对齐的方体，分别沿着每个坐标轴的方向将该方体均匀二分，形成大小相等的八个单元格。对于每个单元格，如果单元格中的物体数量较少，则可以方便地进行各物体间的碰撞检测；如果单元格中的物体数量较多，超出了预定的门限，则以同样的方式迭代进行分割。通过八叉树分割可以根据单元格中物体分布的复杂性，自适应地进行逐层细分。八叉树分割以一种分层结构揭示了空间中物体在不同分辨率下的分布。

Moore 和 Wilhelms [84]将八叉树用于碰撞检测之中，解决多体碰撞检测问题。在实际应用中，八叉树结构不仅可以用于多体碰撞检测，也可用于加速两两物体间的碰撞检测。同均匀空间分割相比，八叉树分割能较好的适应空间中物体的尺寸、位置分布不均的情况。但是，由于采用分层结构，当物体分布较密时，八叉树结构深度较大，这将增加区域检索的计算复杂性。而且，八叉树结构比较复杂，维护更新代价较高。尽管八叉树结构可以在每个计算周期进行更新，当空间中存在大量的运动物体时，八叉树结构的更新计算量将很大。

3) 二叉树分割

二叉树分割作为一种自适应空间分割方法，与八叉树分割有许多相似之处。同样采用分层结构，二叉树分割通过一个平面将根节点分为两个部分，分割平面未必是沿轴向对齐的，分割后的两个部分也未必是尺寸相同的。二叉树分割的原则是两个部分的复杂性相近，对于多面体而言，分割后两个部分的多边形数目相近。二叉树分割一般用于复杂的多面体模型，用以加速两两物体间的碰撞检测。对于多体碰撞检测问题，二叉树分割难以适用。

5.1.3 主要研究内容

基于对已有多体碰撞检测方法的分析，本文研究的基本思路是不对物体的运动速度等信息作要求，主要利用物体间的空间相关性来加速碰撞检测。具体地，以空间分割作为加速的基本手段，同时，在单元格内的碰撞检测采用扫描排除法。本文不采用

时序调度方法主要是从碰撞检测的通用性和独立性方面来考虑的。将碰撞检测从应用计算过程分离出来，作为单独的模块，将有助于通过分布式计算提高整个系统的计算效率。考虑到空间分割方法可以同诸如扫描排除法等加速算法相结合，具有较强的包容性，且其利用空间相关性的概念十分简捷，因此，本文集中于对基于空间分割的快速碰撞检测方法进行研究。具体包括以下两方面的内容：

1. 均匀空间分割方法 (Uniform Space Subdivision, USS)

首先，为了说明 USS 对加速碰撞检测的意义，本文分别针对计算复杂性为 $O(n^2)$ 的原始碰撞检测方法（不采取任何加速手段的两两检测）和一类复杂性为 $O(n \log n)$ 的方法，给出了两个定理，证明了采用 USS 方法，可以在原算法基础上进一步加速。

USS 需要解决的一个关键问题是单元格尺寸的确定。以往的 USS 确定单元格尺寸主要是基于经验，缺少定量的性能指标依据。在[94]中，给出了一个计算单元格尺寸的模型，但模型是基于物体的速度信息给出的。本文主要利用物体的分布密度，通过一个三维空间的点分布模型，给出一个计算单元格尺寸的方法。在 I_COLLIDE 算法中，扫描排除是基于插入排序实现的，插入排序的计算效率受物体运动的时间相关性变化的影响较大，因此，本文在单元格内采用一种基于 AVL 排序的扫描排除法。给出了一个快速碰撞检测算法。本文在对该算法进行复杂性分析的基础上，通过与 I_COLLIDE 算法进行对比实验，验证了算法的效率。

2. 非均匀空间分割方法 (NonUniform Space Subdivision, NUSS)

为了适应物体非均匀分布的情况，本文对非均匀空间分割方法进行了深入研究。基本的思路是，首先根据分布密度对空间进行非均匀分割，分割后的每个单元格内的物体分布密度基本均匀；然后对每个单元格按照前述的均匀空间分割方法进行分割。所以，NUSS 的关键是如何将空间按分布密度进行分割。为此，本文提出了两种基于分布密度的 NUSS 方法：

1) 基于轴向投影直方图的方法

首先在三个坐标轴建立物体的投影的直方图，轴向直方图反应了轴向的分布密度；然后，在每个轴向，通过区域生长算法实现轴向分割；最后，通过三个轴向分割的合成实现空间分割。

2) 基于聚类的自适应空间分割

为了使分割直接反映物体的空间分布密度，首先，对物体进行基于局部密度的聚类，聚类将根据距离和密度的一致性对物体进行分类；然后，根据聚类的结果进行分割。一般的聚类方法，是通过检测样本点间的距离的一致性进行聚类的。对基于密度的聚类问题而言，密度信息不是直接给出的，因此，必须对样本的局部密度进行测量。本文在已有基于密度的聚类算法的基础上，给出了一种新的聚类算法。

聚类对物体集合形成了一个划分，根据聚类结果进行空间分割的过程，实际上是

通过一定的空间包围区域对类簇的形状进行近似的过程，空间包围区域越细致，其对类簇的近似越准确，但相应的计算和存储复杂性越高，反之，则准确性越低，计算和存储也越简单。因此，需要在分割的准确性和简单性方面进行权衡。本文通过包围盒来形成空间分割，给出了具体的分割方法。

5.2 基于均匀空间分割的快速碰撞检测方法

5.2.1 基本原理

均匀空间分割的基本思想是对整个场景空间 Δ ，分别沿 x 、 y 、 z 轴进行等间隔分割，形成 n_x 、 n_y 、 n_z 个部分，由此形成大小相等的一系列单元格，称为体元 (volume or voxel)。均匀空间分割的关键是体元大小的确定，体元过大，对于密度较大的场景，体元中的物体数目较大，就会降低碰撞检测的效率；反之，体元过小，均匀空间分割可能产生数目众多的体元，增加了算法的内存开销。因此，必须根据具体的应用情况，优化确定体元的大小。一个经常被采用的原则是 $\sqrt[3]{n}$ 原则，即对于预定的体元数目 n ，采用如下公式：

$$n_x = n_y = n_z = \sqrt[3]{n}。$$

均匀空间分割比较适合于场景中的物体呈均匀分布的情况，而对于物体密度不均的情况，用均匀分割方法，往往难以找到一个优化的体元大小，以至算法难以在时间效率和空间效率间取得平衡。同以 Octree 为代表的分层结构的自适应空间分割方法相比，均匀空间分割方法产生的数据结构简单，体元的查找比较快捷。在碰撞检测中，均匀空间分割是用的最多的一种空间分割方法。均匀空间分割同扫描排除等其它快速碰撞检测方法可以相互结合，能进一步提高碰撞检测的效率。为说明均匀空间分割对加速多体碰撞检测的意义，本文分别针对不采取任何加速手段的原始碰撞检测方法和一类复杂性为 $O(n \log n)$ 的方法，给出两个定理。

设参与碰撞检测的物体的数量为 n ，通过均匀空间分割将整个物体空间分成 m 个单元格，每个单元格中包含的物体数为 n_i 。

如果不采用任何加速方法， n 个物体间的碰撞检测计算复杂性为 $O(n^2)$ 。通过均匀空间分割，每个单元格中物体的碰撞检测的计算复杂性为 $O(n_i^2)$ 。

[定理 5.1]: 令 $F_0 = n^2$ ， $F_s = \sum_{i=1}^m n_i^2$ ，其中， $n = \sum_{i=1}^m n_i, n_i \geq 0, m > 1$ ，则有：

$$F_s \leq F_0, \text{ 且当 } n_1 = n_2 = \dots = n_m = \frac{n}{m} \text{ 时, } F_s = \frac{n^2}{m}, \text{ 取得最小值。}$$

证明：对于 $F_s \leq F_0$ ，很容易证明，不再赘述。对 F_s 的最小值，通过拉格朗日求

极值的方法加以证明。令：

$$f(n_1, n_2, \dots, n_m, \lambda) = \sum_{i=1}^m n_i^2 + \lambda \cdot (\sum_{i=1}^m n_i - n), \text{ 则有:}$$

$$\begin{cases} \frac{\partial f}{\partial n_i} = 2n_i + \lambda \\ \frac{\partial f}{\partial \lambda} = \sum_{i=1}^m n_i - n \end{cases}, \text{ 令 } \frac{\partial f}{\partial n_i} = 0, \frac{\partial f}{\partial \lambda} = 0, \text{ 则可得: } \begin{cases} \lambda = -\frac{2n}{m} \\ n_i = -\frac{\lambda}{2} = \frac{n}{m} \end{cases},$$

因此, F_s 在 $n_1 = n_2 = \dots = n_m = \frac{n}{m}$ 时, 取得最小值 $F_{s, \min} = \frac{n^2}{m}$ 。□

假设存在某种快速碰撞检测的方法使得 n 个物体间的碰撞检测的计算复杂性为 $O(n \log n)$ (如扫描排除法), 则经过均匀空间分割, 每个单元格中物体碰撞检测的计算复杂性为 $O(n_i \log n_i)$ 。

[定理 5.2]: 令 $F_s = \sum_{i=1}^m n_i \cdot \log n_i$, $F_0 = n \cdot \log n$, 其中, $n = \sum_{i=1}^m n_i, n_i > 0, m > 1$, 则

有:

$$F_s \leq F_0, \text{ 且当 } n_1 = n_2 = \dots = n_m \text{ 时, } F_s = n \cdot \log\left(\frac{n}{m}\right), \text{ 取得最小值。}$$

证明: $\because n_i \leq n, \therefore \log n_i \leq \log n$, 所以有:

$$F_s \leq \sum_{i=1}^m n_i \log n = \log n \cdot \sum_{i=1}^m n_i = n \cdot \log n = F_0$$

令 $f(n_1, n_2, \dots, n_m, \lambda) = \sum_{i=1}^m (n_i \cdot \log n_i) - \lambda \cdot (\sum_{i=1}^m n_i - n)$, 则有:

$$\begin{cases} \frac{\partial f}{\partial n_i} = \log n_i - \lambda \\ \frac{\partial f}{\partial \lambda} = \sum_{i=1}^m n_i - n \end{cases}, \text{ 令 } \frac{\partial f}{\partial n_i} = 0, \frac{\partial f}{\partial \lambda} = 0, \text{ 则可得: } \begin{cases} \lambda = \log\left(\frac{n}{m}\right) \\ n_i = \frac{n}{m} \end{cases},$$

因此, F_s 在 $n_1 = n_2 = \dots = n_m = \frac{n}{m}$ 时, 取得最小值 $F_{s, \min} = n \cdot \log\left(\frac{n}{m}\right)$ 。□

从上面两个定理可以看出, 采用均匀分割方法可以降低碰撞检测的计算复杂性, 当物体服从均匀分布时, 均匀空间分割将取得最佳效果, 这说明, 均匀空间分割方法比较适合于均匀分布的情况。

影响均匀空间分割方法性能的一个关键因素是单元格的数目 m , 当只考虑一次性碰撞检测的效率时, 似乎 m 越大越好。然而, 随着 m 的增加, 系统的存储效率和结构的更新效率将降低。在均匀空间分割中, 每一个体元 (单元格) 都要保存一个与其相交的物体的列表。当 m 较大时, 这将造成很大的存储开销。此外, 如果在每个体元的碰撞检测中采用快速算法 (如扫描排除法), 那么往往需要维持一个复杂的数据结

构（如二叉树），对于动态的环境，必须考虑更新体元内数据结构的计算复杂性。所以， m 的确定，或者说体元的大小的确定需要综合考虑静态碰撞检测效率、存储效率，以及数据更新效率等因素。

在这一节中，首先基于空间特征点分布模型，提出了一种均匀空间分割中单元格尺寸的计算方法。在此基础上，提出了一种结合均匀空间分割和扫描排除方法的快速碰撞检测算法（USSCD）。对 USSCD 的算法复杂性进行了分析。最后，以微粒沉淀仿真试验为平台，实现并验证了 USSCD，并将其同 I_COLLIDE 进行了比较。

5.2.2 均匀空间分割计算模型

考虑到对于物体的数量众多、物体的大小比较均匀、单个物体的体积相对于整个物体空间较小的情况（比如微粒沉淀仿真、分子运动仿真），在考虑物体的空间分布时，重要的是物体的位置信息，而具体尺寸和形状并不十分重要。因此，本文将物体抽象为三维空间中的点，用特征点（如中心、局部坐标系原点）来描述。下面，本文用三维空间特征点分布模型来分析多体碰撞检测的计算复杂性，从而导出确定均匀空间分割单元格大小的方法。

在前面的分析中，采用均匀空间分割方法进行多体碰撞检测，计算复杂性由两个方面构成：单元格内的碰撞检测计算复杂性和单元格数据更新的计算复杂性。二者都取决于底层的碰撞检测算法（单元格内的多体碰撞检测算法）。例如，当单元格内物体间的碰撞检测不采用任何加速算法，完全通过两两碰撞检测实现，那么单元格内碰撞检测的计算复杂性为 $O(n^2)$ ，单元格数据结构更新的复杂性就是点的区域查找的复杂性，对于均匀空间分割而言是常数。在这里，本文基于底层碰撞检测的计算复杂性为 $O(n \log n)$ 的一类算法（如扫描排除方法）进行考虑。

为了计算单元格更新的计算复杂性，需要统计移出/移入单元格的物体的数量，为此，本文引入单元格边界域的概念。所谓单元格的边界域，是指包围单元格的边界（由各个面构成）的空间区域。单元格是由分别平行于三个坐标平面的 6 个面构成，对应于单元格某个面的边界域是以该平面为中心，以边界带宽（边界带宽与物体的尺寸相关）为厚度的长方体区域。如图 5.1 所示。（图中阴影部分为边界域）

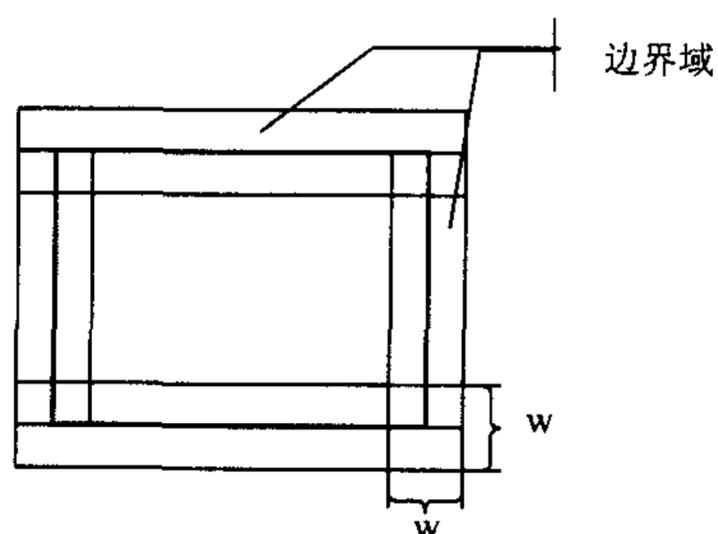


图 5.1 边界域示意图

在没有物体的运动速度信息的情况下，本文用分布于单元格的边界域内的物体的数量作为移出/移入的物体数量的上限。

设一个三维空间域为 D_0 , D_0 上的点分布密度为 $p(v)$, $v(x, y, z)$ 为空间中的一点。设 D_0 上的一个划分为 D_1, D_2, \dots, D_m , 满足:

$$D_0 = \bigcup_{i=1}^m D_i, \text{ 且 } \forall i, j, i \neq j, D_i \cap D_j = \phi. \text{ 其中 } m \text{ 为单元格的数目。}$$

D_0 上的点分布为 N_0 , 每个单元格 D_i 上的点分布为 $N_i = \int_{D_i} p(v) dv, i=1, 2, \dots, m$, 满足:

$$N_0 = \sum_{i=1}^m N_i$$

设单元格 D_i 的边界域为 D_{bi} , 则其边界域上的点分布为 $N_{bi} = \int_{D_{bi}} p(v) dv$ 。

令 D_0 上的碰撞检测计算量为 F , 单元格内部碰撞检测计算量为 F_i , 单元格更新计算量为 F_u , 则有:

$$F = F_i + F_u, F_i = \sum_{i=1}^m F_{ii}, F_u = \sum_{i=1}^m F_{ui}$$

其中, D_i 上的碰撞检测计算量为 F_{ii} , D_i 的更新计算量为 F_{ui} , $i=1, 2, \dots, m$ 。假设, 在单元格内采用某种碰撞检测算法, 使得:

$$F_{ii} = N_i \cdot \log N_i, \text{ 则有: } F_i = \sum_{i=1}^m N_i \cdot \log N_i$$

$$F_{ui} = O(N_{bi} \cdot \log N_i), \text{ 则有: } F_u = \sum_{i=1}^m N_{bi} \cdot \log N_i$$

$$\text{可得: } F = \sum_{i=1}^m (N_i + N_{bi}) \cdot \log N_i \quad (5.1)$$

这里, F_{ui} 的计算是基于排序二叉树的检索得到的。

假设 D_0 内的点服从均匀分布, 即: $\forall v(x, y, z) \in D_0, p(v) = \delta_0$ 。不失一般性, 假设 D_0 为正方体, D_0 的尺寸为 L , 单元格的数目为 m , 单元格的尺寸为 l , 边界带宽为 w (w 由物体的平均尺寸决定), 单元格的体积为 v_i , 边界域的体积为 v_{bi} , 则有:

$$v_i = l^3 = v_0 / m, v_{bi} = 6l^2 \cdot w, v_0 = L^3, m = L^3 / l^3, l = L / m^{1/3}$$

$$N_i = v_i \cdot \delta_0 = N_0 / m, N_{bi} = v_{bi} \cdot \delta_0, N_0 = v_0 \cdot \delta_0$$

根据公式 (5.1), 有:

$$F = \sum_{i=1}^m (N_i + N_{bi}) \log N_i = m \cdot (l^3 + 6l^2 w) \cdot \delta_0 \cdot \log(l^3 \cdot \delta_0)$$

可推得:

$$F = N_0 \cdot (1 + 6w/l) \cdot (3 \log l + \log \delta_0), \quad (5.2)$$

$$F = N_0 \cdot \left(1 + \frac{6wm^{\frac{1}{3}}}{L}\right) \cdot (\log N_0 - \log m),$$

对 (5.2) 进行求导:

$$F'(l) = \frac{3N_0}{l^2} (l + 6w - 6w \log l - 2w \log \delta_0)$$

令 $G(l) = l + 6w - 6w \log l - 2w \log \delta_0$, 可得:

$G'(l) = 1 - 6w/l$, 当 $l = 6w$ 时, $G(l)$ 取得最小值:

$$G_{\min} = 2w(6 - 3 \log 6w - \log \delta_0)$$

这里有两种情况:

1) $6 - 3 \log 6w - \log \delta_0 \geq 0$, 此时, $F'(l) \geq 0$, $F(l)$ 随 l 单调递增, 本文令 l 取最小值: $l_{\min} = 2w$ (5.3)

2) $6 - 3 \log 6w - \log \delta_0 < 0$, 此时, $G(l)$ 有两个零点, 易知, 在 $G(l)$ 的第二个 (较大的) 零点处, $F(l)$ 取极小值。本文可以通过数值计算方法求得该零点 l_0 , l_0 满足: $l_0 \geq 6w$, 取:

$$l = l_0 \quad (5.4)$$

5.2.3 快速碰撞检测算法 USSCD

在 USSCD 中, 单元格内的碰撞检测采用类似于 J. D. Cohen 和 M. C. Lin 的扫描排除法 (Sweep and Prune) [95]。首先将物体的 AABBs 向三个坐标轴 (x, y, z) 进行投影, 对于凸多面体, 其轴向投影为一段区间, 分别对三个轴向的投影区间的端点进行排序, 然后, 依序对投影区间端点进行扫描, 完成交叠检测。所不同的是, USSCD 采用了不同的排序算法和数据结构。在 J. D. Cohen 和 M. C. Lin 的算法中, 采用了冒泡和插入排序算法, 基于物体运动具有较强的时间相关性的假设, 冒泡和插入排序算法可以获得接近线性的计算复杂性。但是, 当物体运动的时间相关性较差时, 冒泡法和插入排序法在最坏情况下的计算复杂性可退化为 $O(n^2)$ 。USSCD 采用基于平衡二叉树 (AVL 树) 的二分插入排序法, 这种方法保证算法的计算复杂性为 $O(n \log n)$ 。

在 USSCD 中, 以轴向对齐包围盒 (AABB) 作为物体的包围体, 空间分割针对这些 AABB 进行。设包围体的最大边长为 R_{\max} , 则令边界带宽 $w = R_{\max}$ 。当物体分布密度比较稀疏时, 根据公式 (5.3), 取 $l = l_{\min} = 2w = 2R_{\max}$, 这时, 保证每个单元格在轴向上至少可以容纳两个物体。而当物体分布比较密集时, 根据公式 (5.4), 通过数值计算的方法确定一个优化的单元格尺寸。在实际应用中, 问题空间未必是立方体, 而往往是长方体, 对此, 基于空间特征点分布模型, 按照相同的思路可以确定优化的

单元格的大小, 这里不再赘述。为了讨论方便, 这里只以立方体的情况为例介绍算法。需要指出的事, 根据公式 (5.3) 或 (5.4), 最终获得的单元格的数量 m 未必是整数, 在实际应用中, 需要根据单元格大小的最优值, 确定一个接近最优的单元格尺寸, 使得沿各个轴向的分割为整数分割, 即 $m = \left\lceil \frac{L}{l} \right\rceil$ 。建立物体空间的坐标系如下: 坐标系的各个轴向分别与 x, y, z 轴对齐, 坐标系的原点取为物体空间立方体的一个顶点, 使得整个物体空间处于坐标系的第 I 相限:

$$D = \{v(x, y, z) \mid 0 \leq x \leq L, 0 \leq y \leq L, 0 \leq z \leq L\}$$

此时, 每个单元格表示为 $d(d_x, d_y, d_z)$, 其中 $0 \leq d_x, d_y, d_z \leq m$, 使得:

$$d(d_x, d_y, d_z) = \{v(x, y, z) \mid d_x l \leq x \leq (d_x + 1)l,$$

$$d_y l \leq y \leq (d_y + 1)l,$$

$$d_z l \leq z \leq (d_z + 1)l\}$$

在预处理阶段, 对于每个单元格, 计算出所有与其相交的 AABBs, 并形成特定的数据结构 (平衡二叉树)。一个 AABB 与单元格相交的充分必要条件是, AABB 在三个轴向上的投影都与单元格的投影区间相交。

在扫描排除方法中, 需要对每个单元格中的物体进行排序。本文通过构造 AVL 树实现排序过程。设一个 AABB 在 X 轴上的投影区间为 $[L_x, R_x]$, 其中 L_x 和 R_x 是投影区间的边界点, 算法将这些边界点按大小进行排序, 形成序列, 这个序列以 AVL 树的结构表示。

在每个计算周期, 由于物体的运动, 单元格中的数据结构需要进行更新, 导致单元格内数据更新的事件有三种, 对应的操作各不相同:

- 1) 原有物体移出单元格, 对应数据结构的删除操作;
- 2) 新物体移入单元格, 对应数据结构的插入操作;
- 3) 物体投影的相对位置发生变化, 对应数据结构的排序调整操作。

数据更新的思路是, 在每个计算周期, 首先重新分配每个 AABB 给对应的单元格, 然后在每个单元格重新建立数据结构。算法主要包括两个步骤: 物体分配和物体排序。物体分配的任务是重新确定各个物体的 AABB 的归属。分配过程以单元格为单位依次进行, 在每个单元格中, 如果检测到某个物体不再属于当前单元格, 则将该物体的投影从 AVL 树中删除, 同时计算与物体相交的单元格, 如果物体与一个新的单元格相交 (在前一个计算周期不相交), 则将物体标志为该单元格的新的元素以待排序。为了方便物体与单元格之间关系的查询, 需要对每个物体维持一个与之相交的单元格的列表。分配过程结束以后, 对每个单元格中的物体进行排序 (通过投影排序)。因此, 对于一个单元格, 其数据更新主要通过对原有的 AVL 树进行删除、调整, 和插入操作完成。

5.2.4 算法复杂性分析

在预处理阶段，要进行物体的分配和单元格内 AVL 树的构造。由于采用 AABB 作为包围体，AABB 与单元格的相交计算量为常数。一个物体最多与 8 个单元格相交，因此分配一个物体的计算复杂性为常数。完成全部 n 个物体的分配的计算复杂性为 $O(n)$ 。对于含有 n_i 个物体的单元格，构造其中的 AVL 树的过程实际就是对 n_i 个元素的二分插入排序的过程，其计算复杂性为 $O(n_i \log n_i)$ ，根据定理 5.2，有：

$$\sum_{i=1}^m n_i \log n_i \leq n \log n,$$

因此，完成所有单元格的 AVL 树构造的计算复杂性为 $O(n \log n)$ 。

在碰撞检测过程中，更新一个单元格的数据结构的计算量由平衡二叉树上插入和删除操作构成。在含有 n 个节点的 AVL 树中，一个节点插入、删除的计算复杂性是 $O(\log n)$ 。在一个计算周期中，碰撞检测的计算复杂性根据公式 (5.1) 计算得到。

5.2.5 实现与对比实验

为了验证 USSCD 算法的性能，本文分别实现了 USSCD 和 I_COLLIDE，并以微重力环境下的微粒沉淀试验仿真为背景，进行了一系列的实验，基于试验结果对两种算法的性能进行了比较分析。

由于 USSCD 主要面向物体均匀分布的情况，因此，在实验中令物体随即地均匀分布在一个立方体空间中，而且物体的尺寸基本相同。本文采用球体作为碰撞检测对象。虚拟环境如下图所示：

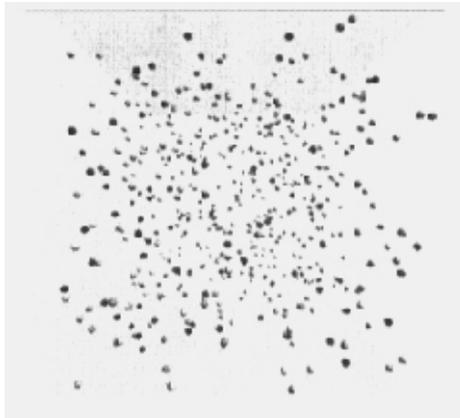


图 5.2 微粒实验仿真环境

试验的一个主要目的是考察 USSCD 的计算复杂性。在 USSCD 中，提高碰撞检

测效率的手段主要包括两个方面：基于 AVL 的排序扫描和空间分割。为了考察这两个方面的实际效果，在实验中将 USSCD 和 I_COLLIDE，以及未采用空间分割的 AVL 排序扫描法（简称为 SCD）进行了比较。考虑到影响碰撞检测效率的两个重要因素：物体的空间分布密度和物体的平均运动速度。分布密度反映了物体之间的空间相关性，而运动速度反映了物体间的时间相关性。在实验中，分别考察了分布密度和运动速度变化对上述三个算法（USSCD、SCD、I_COLLIDE）效率的影响。

本文主要进行了三项实验：

- 1) 复杂性测试实验。在物体的分布密度和运动速率保持不变的情况下，测试碰撞检测效率同物体数量变化之间的关系；
- 2) 空间相关性测试。在物体的数量和运动速度保持不变的情况下，测试碰撞检测效率同物体空间尺寸变化之间的关系；
- 3) 时间相关性测试。在物体的数量和分布密度保持不变的情况下，测试碰撞检测效率同物体运动速率变化之间的关系。

在实验中，分布密度是由物体的数量同空间的体积的比决定的。设物体的数量为 N ，物体空间的尺寸为 L ，则分布密度表示为： $\delta = N/L^3$ 。在实验中，让物体以一定的速率 SPEED，在随机方向上进行运动。碰撞检测算法的效率由一段时间内的平均帧频率 FPS 来度量。设在 Δt 时间内，碰撞检测执行了 n 次，则 $FPS = n/\Delta t$ 。这里， Δt 只包含碰撞检测部分的计算时间，不包括诸如绘图等过程的时间。

三项测试的实验数据如下表所示：

表 5.1 复杂性测试 ($\delta = 0.0005, SPEED = 0.5$)

N \ FPS	16	32	64	128	256	512	1024	2048
USSCD	217.62	93.67	46.69	21.61	10.61	5.24	2.50	1.14
SCD	376.13	152.11	62.27	23.62	9.24	2.63	0.98	0.26
ICOLLIDE	1471.43	431.83	154.21	43.93	11.16	1.99	0.53	0.10

表 5.2 空间相关性测试 ($N = 256, SPEED = 0.5$)

L \ FPS	16	23	64	128	256
USSCD	5.877	9.819	11.404	9.742	4.476
SCD	2.565	4.964	8.547	12.641	15.317
ICD	4.199	8.097	11.216	13.039	14.205

表 5.3 时间相关性测试 ($N = 512, L = 100$)

SPEED \ FPS	0.1	0.2	0.4	0.8	1.6	3.2	6.4	12.8
USSCD	217.62	93.67	46.69	21.61	10.61	5.24	2.50	1.14
SCD	376.13	152.11	62.27	23.62	9.24	2.63	0.98	0.26
ICOLLIDE	1471.4	431.83	154.21	43.93	11.16	1.99	0.53	0.10

首先, 根据复杂性测试数据, 分别将 USSCD 同 SCD 和 I_COLLIDE 进行比较, 如下图所示:

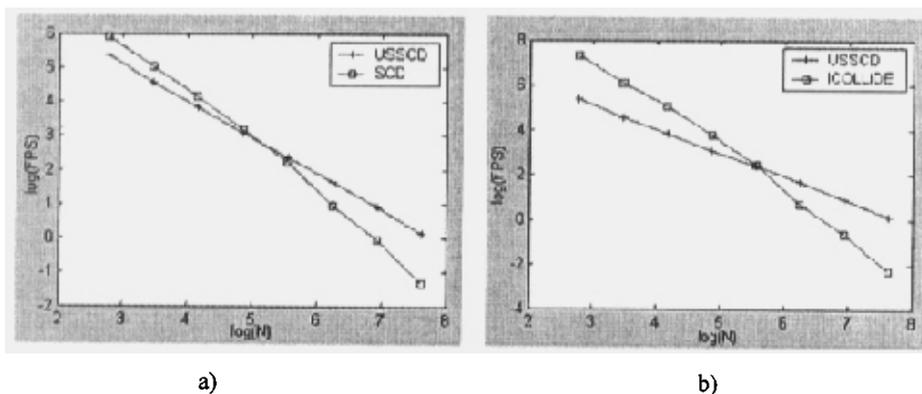


图 5.3 复杂性效果分析图

在图 a)和 b)中, 横坐标是物体数量 N 的对数, 纵坐标是帧频率 FPS 的对数。从图 a)可以看出, 当 N 比较小时, SCD 的速度要高于 USSCD, 而随着 N 的增加, 两种算法的速度都呈下降趋势, 而 SCD 下降的速度较 USSCD 要快。这说明, 当 N 较小时, 空间分割所带来的计算开销大于其在每个单元格内所产生的效益, 因而降低了碰撞检测的效率; 而当 N 较大时, 空间分割所带来的单元格内的效益明显高于其引入的计算开销, 所以较之 SCD 表现出较高的效率。如果近似将两条曲线视为直线, 则从直线的斜率可以看出, USSCD 的计算复杂性要优于 SCD; 在图 b)中, USSCD 同 I_COLLIDE 的关系同图 a)中相似, 所不同的是, I_COLLIDE 算法的速度随 N 的增加而下降的速度更快。USSCD 的计算复杂性优于 I_COLLIDE。这一方面是由于空间分割的作用, 另一方面是由于采用 AVL 排序方法的缘故。

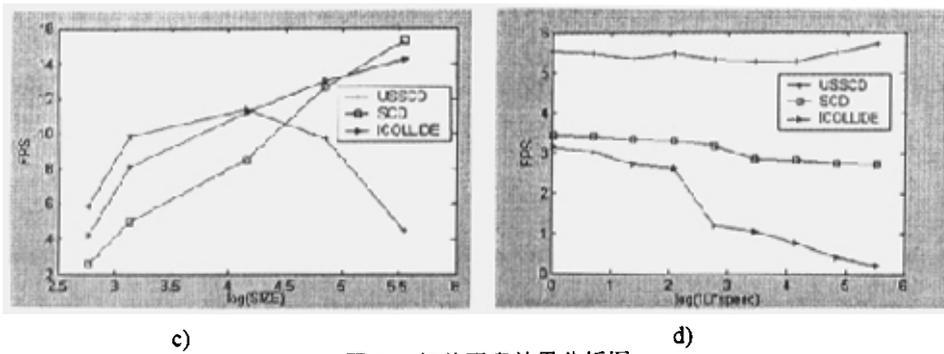


图 5.4 相关因素效果分析图

在图 c)中, 比较了物体分布密度变化对三种算法效率的影响。其中, 横坐标是物体空间尺寸的对数, 纵坐标是 FPS。在物体数量一定的情况下, 空间尺寸越大则分布密度越小。从图中可以看出, 在一定范围内, 密度的减小将增加 USSCD 的速度。但密度小到一定程度后, USSCD 的速度将随着空间尺寸的增加而降低。这是因为, 在运动速度一定的情况下, 密度越小, 对排序算法而言, 物体的相对位置变化越少, 因而减少了排序的计算开销; 同时, 物体之间的距离的增加, 将降低物体间碰撞的概率, 因此减少了扫描过程的计算开销。密度减小, 从排序和扫描两个方面提高了碰撞检测的效率。但是, 密度小到一定程度后, 空间分割的单元格尺寸不变, 随着空间尺寸的增加, 单元格的数目呈指数倍增加 (单元格数量同空间体积成正比), 由于单元格增加而引入的计算开销降低了 USSCD 的效率。由于没有单元格增加的负面影响, SCD 和 I_COLLIDE 的速度将随着密度的减小而提高。

在图 d)中, 比较了物体运动速度变化对三种算法效率的影响。其中, 横坐标是物体运动速度的对数, 纵坐标是 FPS。从图中可以看出, 随着物体运动速率的增加, SCD 和 I_COLLIDE 的效率都不同程度地降低, 而 USSCD 则没有明显的变化。物体运动的方向是随机的, 因此, 物体运动速率的增加减少了物体运动的时间相关性。图 d)说明, USSCD 算法效率基本不受物体运动的时间相关性的影响。而 I_COLLIDE 算法由于采用了递增式的碰撞检测, 算法效率依赖于物体运动的时间相关性, 当相关性较低时, 算法效率明显降低。排序算法在一定程度上受物体运动时间相关性的影响, 但是对于基于 AVL 的排序算法而言, 这种影响不大。

5.2.6 小结

实验表明, 当待检测物体数量较多时, USSCD 的碰撞检测效率相对 I_COLLIDE 算法较高。同时, 对同种扫描排除方法, 空间分割表现出优越性。USSCD 算法的效率基本不受物体运动的时间相关性的影响。但是对物体分布密度较低的情况, USSCD 的效率将随着物体空间增大而降低。

5.3 基于非均匀空间分割的快速碰撞检测方法

5.3.1 概述

当考虑空间中物体的非均匀分布的情况,单纯的均匀空间分割的方法将表现出局限性。经典的非均匀空间分割算法如:八叉树(Octree)分割、二叉树分割(BSP)等,分割算法和分割结果都比较简单,算法效率较高,在光线跟踪等方面中都取得了比较成功的应用。在光线跟踪中,空间分割的目的是尽量减少与光线进行相交计算的物体的数量,因此空间分割的一个主要任务是控制每个单元格中物体分布的复杂性,或者说控制单元格中物体的数量。非均匀空间分割面临的一个主要问题是如何在单元格中物体的数量和空间区域查找效率之间进行折衷。由于在光线跟踪中通常针对的是静态景物,因此,空间分割方法很少考虑空间结构更新的计算开销。八叉树和二叉树在控制单元格中物体的数量方面都取得了较好的结果,但它们是通过复杂的分层结构来实现空间分割的。对于碰撞检测问题而言,复杂的分层结构在运动物体的碰撞检测中遇到的主要问题是结构更新的代价较高。此外,复杂分层结构的区域查找效率相对于均匀空间分割而言较低。因此,在碰撞检测中,除了单元格中物体分布的复杂性,必须同时考虑空间区域查找效率,以及空间分割结构更新的计算复杂性。考虑到分割的性能将直接影响碰撞检测的效率,通过性能较好的分割来提高后续碰撞检测的效率,这促使本文在预处理阶段采用比较复杂的分割方法,牺牲分割的效率来换取分割的性能。

非均匀空间分割的一条基本原则是空间分割应该反映物体的空间分布。在物体分布稀疏的区域进行比较粗略的分割,而对分布密集的区域,应该给予比较细致的分割。对于碰撞检测应用,如何进行空间分割应该结合具体的应用特点。在本文中,着重针对具有以下特点的碰撞检测问题进行研究:

- 参与碰撞检测的物体数量众多,以至一般的两两碰撞检测无法满足实时计算的需要,这促使本文研究基于空间分割的快速多体碰撞检测方法;
- 物体的大小比较一致,在考察物体的空间分布时,可以只考虑物体的位置,而对物体的大小和形状可以忽略;
- 物体在空间中的分布结构具有时移不变性,即在仿真过程中物体的空间分布基本保持不变。这里对于空间分布的限制并没有限制物体的具体运动。

上述特征主要基于两点考虑:一是在诸如微观粒子仿真试验等应用环境,单个物体相对于整个空间而言较小;二是,在某些应用中,物体的运动具有一定的区域性,整体空间表现出比较稳定的非均匀性结构。基于上述特征,可以将物体由三维空间中相应的特征点(如物体的中心)来表示,从而基于三维空间点分布模型对物体的空间

分布进行分析并形成非均匀空间分割。根据空间分布的时移不变性，在预处理阶段进行非均匀空间分割，并在碰撞检测过程中保持空间分割不变。

为了使非均匀空间分割在反映物体的非均匀分布结构的同时，具有均匀空间分割在空间区域查找效率方面的优点。本文将非均匀空间分割分为两个层次，在实现上对应于两个步骤。首先，根据物体的空间分布密度，进行粗略的非均匀空间分割，分割后的单元格都是轴向对齐的方体，但每个单元格在三个轴向上的边长不同，不同单元格的大小也不同。经过粗略的非均匀空间分割，每个单元格中的物体基本呈现均匀分布；在分割的第二步，基于前面的非均匀空间分割，对每个单元格进行均匀空间分割，具体的均匀空间分割方法参照上一节的论述。非均匀空间分割如下图所示：

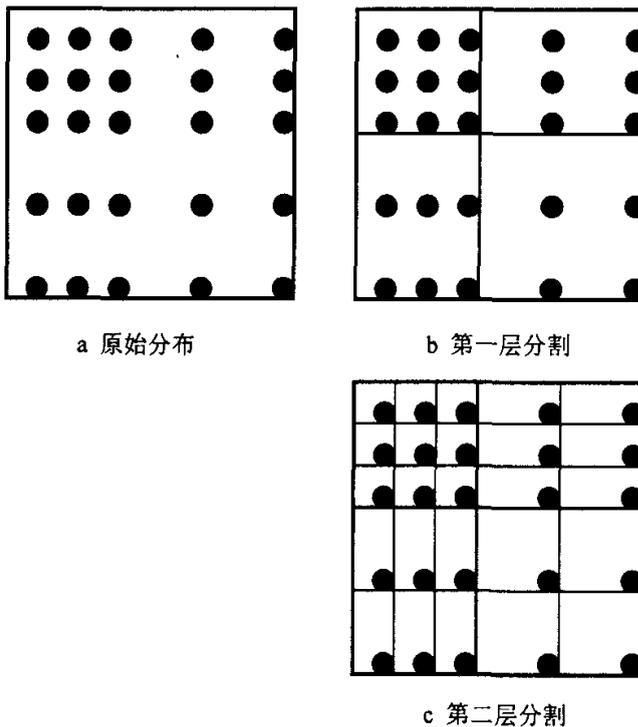


图 5.5 非均匀分割示意图

在本节余下的内容中，本文将着重阐述两种非均匀空间分割方法，首先介绍基于投影的非均匀空间分割方法（PBNUSS）；第二个非均匀空间方法是基于聚类的方法，本文提出了一个基于局部密度的聚类算法(HDBC)，在 HDBC 的基础上提出了一种基于聚类的自适应空间分割方法（CBASS）。

5.3.2 基于投影的非均匀空间分割

非均匀空间分割的目的是检测三维空间中物体的分布密度，根据分布密度的一致

性原则进行分割，形成若干单元格，每个单元格基本服从均匀分布。然而，直接检测密度均匀的空间区域是比较复杂的工作。如果将每个单元格限定为轴向对齐的长方体，那么对于一个密度均匀的长方体区域，其中物体在三个轴向（X, Y, Z）上的边缘概率分布应该也是均匀的；反之，如果在三个轴向上的边缘概率分布都是均匀的，那么该长方体区域中的物体服从均匀分布。基于上面的考虑，本文通过轴向投影的方法获得快速的非均匀分割。

基于轴向投影的非均匀空间分割（Projection Based Non-Uniform Spatial Subdivision, PBNUSS）的基本思想是将三维空间中的样本点分别向坐标系的三个轴向（X, Y, Z）进行投影，根据轴向投影建立投影直方图，用投影直方图表征分布密度，在直方图基础上检测符合密度一致性条件的区域，在轴向形成非均匀分割，再将三个轴向上的分割进行合成，形成三维空间中的分割。设样本点 p 在 X 轴上的投影为 p_x ，则样本点集 $P = \{p_1, p_2, \dots, p_N\}$ 在 X 轴上的投影形成的集合为 $P_x = \{p_{1x}, p_{2x}, \dots, p_{Nx}\}$ 。根据 P_x 在 X 轴的分布形成的分割为 D_x ，

$$D_x = \bigcup_{i=1}^{m_x} D_{ix}$$

同样，在 Y 轴和 Z 轴形成的分割为 D_y 、 D_z ，则样本点集 P 在三维空间中形成的分割为 D，满足：

$$D = D_x \otimes D_y \otimes D_z$$

基于投影的非均匀空间分割如下图所示：

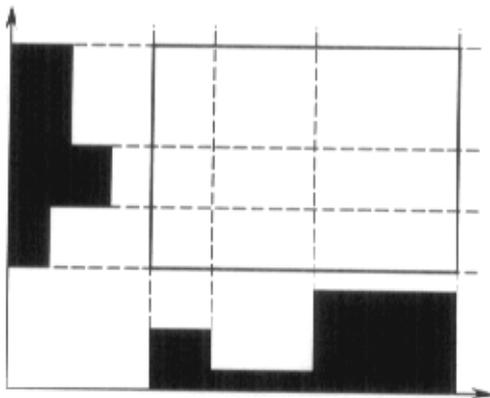


图 5.6 PBNUSS 示意图

其中阴影区域代表轴向直方图；

虚线代表轴向分割

PBNUSS 主要分为三个步骤：

- 1) 构造投影直方图，首先对一维空间进行采样离散化为一系列的区间，然后以这些区间为单位，统计位于各区间上的物体的数量，构造直方图。

- 2) 基于投影直方图进行区域分割, 对一维投影直方图进行区域分割的方法可以参考图像处理中的一些经典方法(如: 边缘检测、区域生长、区域合并)。这里的关键是分割参数的确定, 分割参数用于衡量分割区域的密度一致性, 直接影响分割的精细程度。本文主要采用基于区域生长的分割方法进行研究。

- 3) 对轴向分割进行合成, 形成三维空间非均匀分割。

1. 构造直方图

对于均匀采样, 构造直方图的关键是采样间隔的确定。在 PBNUSS 中, 设物体的数量为 n , 物体空间长度为 L , 令平均每个采样区间所包含的物体数量是 δ , 则采样间隔定为:

$$l = \delta \cdot L / n$$

这里, δ 是预定的参数, δ 的大小将影响随后的区域分割的效果。如果 δ 过小, 则由投影直方图描述的分布密度曲线将表现出较低的信噪比; 反之, δ 过大, 则投影直方图不能充分反映密度变化的细节。

2. 基于区域生长的分割

区域生长方法在数字图像处理和计算机视觉领域中, 是一种基本的图像分割方法。区域生长方法的基本原理是利用区域内元素的性质的相似性, 或者说一致性。对数字图像而言, 一个有意义的区域, 其中像素灰度值往往具有较强的一致性。区域生长方法基于一定的一致性准则, 从某一个种子(核)开始, 不断向邻域扩展, 当邻域中的点符合一致性准则, 比如其灰度值与当前区域的平均灰度值之差小于一定的阈值, 则将其吸收入当前区域; 否则作为边界点等待其它区域的吸收。通过不断地吸收新的像素, 区域逐渐生长, 直到无法继续扩展, 区域生长结束。通过初步的区域生长形成的图像分割中, 相邻区域的平均灰度可能比较接近, 因此需要对性质相近的相邻区域根据一致性准则进行合并。合并过程往往是一个迭代的过程, 直到所有相邻区域无法再进行合并, 区域生长结束。

在 PBNUSS 中, 区域生长过程就是沿坐标轴对各区域进行扫描和合并的过程。设一维投影空间为 $R = [0, L]$, 经过离散化后, R 被分割为 m 个区域 $\{R_1, R_2, \dots, R_m\}$, $R_i = [l_i, r_i]$, 其中, l_i, r_i 分别是 R_i 的左右边界点, 满足:

$$l_1 = 0, r_1 = l_2, \dots, r_i = l_{i+1}, \dots, r_m = L, i = 1, \dots, m$$

R_i 的平均密度为 $D_i = N_i / L_i$, 其中, L_i 为 R_i 的长度, N_i 为 R_i 中的样本点数。设一致性准则为:

$$f(D_i, D_j) \leq T, T \text{ 为预定的阈值。}$$

区域生长算法如下:

- 1) 从 R_1 开始, 当前区域的密度为 $D = D_1$, 当前邻域索引为 $index = 2$ 。

2) 检测 R_{index} ，如果 $f(D_{index}, D) \leq T$ ，则：

- 将 R_{index} 并入当前区域
- 更新当前区域密度

否则：

- 当前区域生长结束
- 新的生长区域设为 R_{index} ，当前区域的密度为 $D = D_{index}$

3) $index = index + 1$ ，如果 $index \leq m$ ，则返回到 2)。

4) 本轮区域生长结束。

一轮区域生长结束后，如果发生了区域合并，就进行下一轮的合并，这个过程迭代进行，直到所有区域无法进行合并为止。

5.3.3 基于局部密度的聚类算法 HDBC

当空间中物体的分布呈现出明显的结构时，也就是说，物体都集中在一些高密度区域，而在其它区域的分布则比较稀疏，要使得非均匀空间分割能充分反映出空间分布结构，一个自然的想法是根据空间中物体的自然分类来分割。因此，本文通过聚类来检测空间中的自然分类，然后基于聚类来进行非均匀空间分割。

在聚类问题中，一个基本概念是类簇 (clusters) 的定义，一般，类簇被描述为空间中的连续区域，每个区域由密度较高的点构成，不同高密度区域之间由低密度区域分隔。同时，类簇内的样本点之间的相似性大于类簇之间的样本点的相似性。

设 X 为样本集合， $X = \{x_1, x_2, \dots, x_N\}$ ，其中 N 为样本点的数目。定义 X 上的一个 m -聚类 \mathfrak{R} ， \mathfrak{R} 将 X 分割为 m 个子集 (类簇) C_1, C_2, \dots, C_m ，各子集满足如下条件：

- i) $C_i \neq \phi, i=1, \dots, m$
- ii) $\bigcup_{i=1}^m C_i = X$
- iii) $C_i \cap C_j = \phi, i \neq j, i, j = 1, \dots, m,$

聚类算法就是在样本集合 X 的多种可能的分割中形成有意义的聚类的计算过程。按照相似性判据的不同，聚类算法可以分为：基于距离 (Location-based) 的聚类和基于密度 (Density-based) 的聚类。基于距离的聚类的基本思想是以距离作为样本间相似性的度量，同一类簇的样本间的距离应该较小，不同类簇的样本间距离则较大；基于密度的聚类则认为自然的分类应该是密度较高的区域，类簇之间由密度较低的区域分隔。因为密度信息需要通过在样本点的邻域计算得到，所以基于密度的聚类又被称为基于局部的聚类 (Locality-based clustering)。传统的聚类算法如 k -平均算法，大都

是基于距离的算法。基于距离聚类的特点是比较适合于类簇内的样本分布比较紧致，类簇的形状为球形或椭球形区域的情况。而其缺点是难以满足任意形状分布的情况。同基于距离的聚类相比，基于密度的聚类算法可以适用于任意形状类簇的情况，不受类簇分布形状的限制。

基于密度的聚类算法面临的一个关键问题是如何有效地计算样本点的局部密度。局部密度往往是由样本点的近邻来描述，比如最近邻距离， k -近邻平均距离，给定邻域内的样本数等，所有这些方法都依赖区域查找技术，即在给定区域查找要求的样本点。比较典型的基于密度的聚类算法有：DBSCAN 算法[133]和 DBCLASD 算法[132]。

在实际应用中，具体采用那种聚类算法取决于问题的具体特点和要求。在面向碰撞检测的非均匀空间分割中，聚类问题的特点是：

- 在三维空间中聚类；
- 期望的类簇的数目是未知的，缺少足够的先验知识；
- 样本点的数量巨大，有必要限制算法的存储效率和计算效率；

在本文的非均匀空间分割方法中，对聚类方法的要求有别于一般的聚类算法，主要有两方面的要求：首先，形成紧致的分类，同基本聚类方法一样，依据距离，对聚集的点簇形成自然的分类，这里，因为空间分割最终形成的分割单元为轴向对齐的方体，所以可以不考虑分布区域为特殊性状（如环形，嵌套等）的聚类；其次，由于非均匀空间分割要保证子空间密度的均匀性，因此，形成一个类簇的样本点，不仅距离接近，而且分布密度要比较均匀。这里，度量样本间的相似性有两个判据：距离（欧氏空间距离）和分布密度，距离可以通过样本点本身的位置计算得到，但分布密度是无法依靠样本点本身获得的。如何依据密度进行分类，是面向非均匀空间分割的聚类算法的难点。在本文中，根据样本点的邻近关系来确定局部密度。基于密度的聚类算法，首先要根据样本点的邻近点计算局部密度；然后，在密度一致性条件约束下，进行聚类。

本文首先通过近邻距离给出局部密度（Local Density, LD）定义。设样本点集为 S ，样本点 $p \in S$ ，则 p 的有关局部密度的定义如下：

[定义 5.1]: (k -近邻距离), p 的 k -近邻距离, 表示为 $ND_k(p)$, 是 p 与其第 k 个近邻点的距离:

$$ND_k(p) = \text{dist}(p, q), p, q \in S, q \text{ 是 } p \text{ 的第 } k \text{ 个近邻点, 即:}$$

$$\exists (k-1) \text{ 个点 } q_i, \text{ 满足: } \text{dist}(p, q_i) \leq \text{dist}(p, q), q_i \in S - \{p\}, i = 1, \dots, k-1$$

[定义 5.2]: (k -近邻密度) p 的 k -近邻密度, 表示为 $LD_k(p)$, 是 p 的 k 个近邻距离的平均:

$$LD_k(p) = \frac{1}{k} \sum_{i=1}^k ND_i(p)$$

由以上定义可见，局部密度测量的关键是对样本点的近邻的查找，即如何快速有效地确定当前点的第 k 个近邻。近邻的确定通过区域查找 (Region queries) 来完成，也就是，通过检查当前点邻域范围内的点来确定第 k 个近邻点。一般，需要预先建立一个区域查找树 (如： k -D 树，八叉树， R^* 树等) 来提高区域查找的效率。在 HDBC 中，基于八叉树进行区域查找。八叉树的构造这里不再赘述。对于要求的 k -近邻密度，令八叉树的每个叶子节点至多包含 k 个样本点。这样，对于样本点 p ，其 k -近邻必然位于 p 所处的叶子节点的上一层节点及其相邻的八个兄弟节点之中。

给定一个种子，本文根据密度一致性原则，通过扩展的方法形成一个类簇。扩展的方式参照 DBCLASD 算法。不同的是在 DBCLASD 中，通过 χ^2 检验保证类簇的密度一致性，而本文则通过限定相对均值误差来保证密度一致性。在此，本文给出 HDBC 相关的定义如下：

设样本空间为 D ，样本集 S 为 D 的一个子集，样本点 p 是样本集 S 的一个成员。

[定义 5.3]: (k -邻域) 点 p 的 k -邻域，表示为 $NND_k(p)$ ，是以 p 为中心， p 的 k -近邻距离 $ND_k(p)$ 为半径的球域：

$$NND_k(p) = \{q \mid dist(p, q) \leq ND_k(p), q \in D\}$$

[定义 5.4]: (直接 k -邻域可达) 样本集 S 中的一点 q ， $q \in S$ ，相对于点 p 是直接 k -邻域可达的，如果 q 满足：

$$q \in NND_k(p)$$

[定义 5.5]: (k -邻域可达) 样本集 S 中的一点 q ， $q \in S$ ，相对于点 p 是 k -邻域可达的，如果存在一条路径 $q_1, q_2, \dots, q_n, q_1 = p, q_n = q$ ，满足： q_{i+1} 相对于 q_i 是直接 k -邻域可达的，其中， $i=1, \dots, n-1$ 。

[定义 5.6]: (k -邻域连接) 样本集 S 中的两点 $p, q, p, q \in S$ ，是 k -邻域连接的，如果存在一个点 r ，满足： p 和 q 都相对于 r 是 k -邻域可达的。

设样本集 C 是 S 的一个子集，则 C 的平均密度 $MD(C)$ 是 C 中元素的平均 k -近邻距离：

$$MD(C) = \frac{1}{n} \sum_{i=1}^n ND_k(p_i), p_i \in C$$

[定义 5.7]: (密度一致性条件) 称 C 中的元素是密度一致的，如果对 $\forall p \in C$ ，满足条件：

$$Dist(ND_k(p), MD(C)) \leq \delta, \text{ 其中, } \delta \text{ 是密度一致因子。}$$

[定义 5.8]: (类簇) 一个类簇 C 是样本集 S 的一个非空子集，满足如下条件：

i) C 中的元素是密度一致的；

- ii) C 是最大的, 即对 $\forall p, q$, 如果 $p \in C$, q 相对于 p 是 k -邻域可达的, 且 q 相对于 C 符合密度一致条件, 则 $q \in C$;
- iii) 对 $\forall p, q \in C$, p 和 q 是 k -邻域连接的 (连接性)。

聚类算法描述如下:

步骤 1: 对给定的参数 k , 建立邻域查找八叉树;

步骤 2: 计算每个点的 k -近邻密度;

步骤 3: 循环进行如下操作, 直到整个样本集搜索结束:

step3.1: 按顺序搜索样本集合, 选择未分类的样本点作为新类簇的种子, 初始化新类簇:

- 1) 将种子加入类簇的待扩展队列;
- 2) 计算类簇的平均密度 MD

step3.2: 重复进行如下操作, 直到待扩展队列为空:

- 1) 从待扩展队列提取一个待扩展点 p
- 2) 对 p 进行密度一致性检验, 如果 p 满足密度一致性条件, 则将 p

加入类簇:

- a) 标记 p 属于当前类簇
- b) 更新类簇的 MD
- c) 将 p 的 k -邻域中的新的 (没有标记的) 样本点加入待扩展队

列

步骤 4: 聚类结束

5.3.4 基于聚类的自适应空间分割

基于聚类的自适应空间分割方法 (Clustering Based Adaptive Spatial Subdivision, CBASS) 通过两个步骤实现场景空间的自适应分割: 首先, 对物体进行基于密度的聚类, 然后根据聚类结果形成非均匀空间分割。

由于基于密度的聚类, 其最终结果是相互分离的、任意形状的区域, 不是严格意义上的空间分割, 必需将聚类结果转化为一个便于后续分割及操作的空间分割。因此, 如何由聚类结果获得期望的空间分割是 CBASS 面临的一个重要问题。

基于密度的聚类方法将产生任意形状的空间区域, 同时这些空间区域只是整个场景空间的子集, 并非一个空间分割。基于这些空间区域产生空间分割, 需要考虑两点:

- 最终的空间分割应该尽可能准确地反映聚类所揭示的物体的分布;
- 空间分割应该具有尽可能简单的结构, 以便于后续的分割、体元查找, 及体元更新操作;

而这两点实际是互相矛盾的。本文假定最终的体元是方向与坐标轴对齐的长方

体。形成空间分割的过程实际是用不同尺寸的体元对空间区域的离散逼近过程。为了追求准确性，体元的分布要尽量反映聚类区域的形状，或者说要刻画出区域的边界。由于不同区域的密度不同，体元的尺寸不同，所以准确性要求增加了空间分割的复杂性，同时空间分割的复杂性势必增加后续操作的复杂性。

由于自适应空间分割的目的就是要提高碰撞检测的效率，所以基于自适应空间分割的操作必须足够简单。本文简化自适应空间分割的方法是用轴向对齐的包围盒来近似聚类形成的空间区域，再基于包围盒形成空间分割。包围盒由类簇中样本点的空间坐标在轴向上的最大最小值形成。采用 AABB 的方法带来的一个问题是，相邻区域的 AABB 可能出现交叠，为了形成空间分割，必须对 AABB 的交叠区域进行分配。一个简单的分配方法是将交叠区域划分给其中的某一个区域；另一种方法是把交叠区域作为一个独立的新的区域，新的区域的平均密度是两个类簇的平均密度的平均。

包围盒交叠情况如图 5.7 所示。其中 i) 图中当包围盒 a 和 b 交叠时，将交叠区域归 a，同时根据 a 的边界对 b 进行分割形成三个区域：b、b1 和 b2。ii) 图中当包围盒 a 和 b 交叠时，将交叠区域作为新的区域 c，同时对 a 和 b 进行分割。

包围盒之间的空白区域根据包围盒的顶点坐标进行分割，同包围盒一起形成最终的空间划分。根据包围盒形成空间分割如图 5.8 所示：

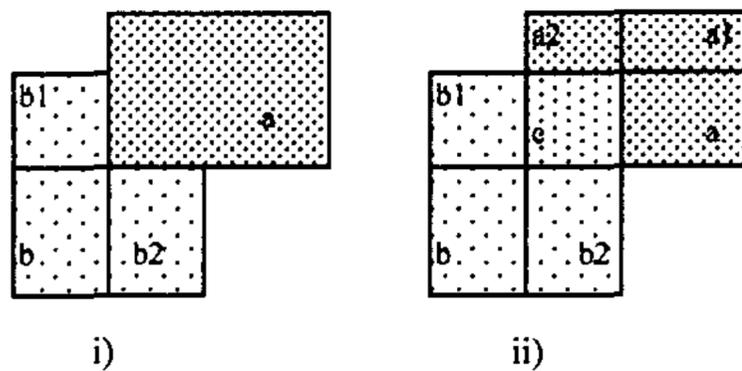


图 5.7 包围盒交叠情况示意图

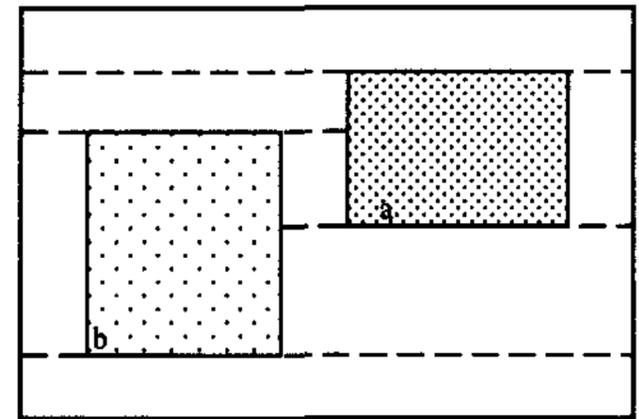


图 5.8 包围盒分离情况示意图

5.4 小结

在这一章，本文基于空间分割方法对多体快速碰撞检测技术进行了研究。针对物体均匀分布的情况，提出了一个基于均匀空间分割的快速碰撞检测算法 USSCD。USSCD 的特点在于：1) 通过一个空间点分布模型实现了单元格尺寸的优化；2) 算法结合了基于 AVL 排序的扫描排除法。在对算法的计算复杂性进行理论分析的基础上，通过实验对算法的性能进行了验证。试验结果表明，在均匀分布情况下，当待检测物体数量较多时，USSCD 的碰撞检测效率较高，优于 I_COLLIDE 算法。同时，算法的效率基本不受物体运动的相关性的影响。但是对空间尺寸较大而且物体分布密度

较低的情况，USSCD 的效率较低。

对物体空间分布不均匀的情况，本文提出了一类非均匀空间分割方法，该方法通过两层分割实现非均匀空间分割。同以往的自适应空间分割方法相比，该方法产生的分割结构比较简单，更适合于碰撞检测的需要。本文提出了两种具体的非均匀空间分割方法，基于投影直方图的空间分割方法 PBUNSS 和基于聚类的自适应空间分割方法 CBASS，在 CBASS 中采用了一个新的基于局部密度的聚类算法 HDBC。

第六章 结论与展望

6.1 结论

以空间遥机器人应用为背景, 本文从理论和实践两个方面对大时延遥操作进行了深入研究。克服大时延影响的一个关键是建立灵活的人机协调机制, 为此, 本文研究了基于 Petri 网的大时延遥操作系统框架, 深入研究了基于 Petri 网的任务建模方法和任务协调方法。遥编程通过结合监控技术和基于虚拟现实的人机接口技术, 为大时延遥操作提供了一种有效途径。本文实现了一个基于遥编程的大时延遥操作试验系统, 进行了典型了遥操作试验。本文以遥科学实验为背景, 对基于空间分割的多体快速碰撞检测技术进行了深入研究, 并对部分算法的性能通过试验进行了验证。

本文提出了一个基于 Petri 网的大时延遥操作系统框架。该框架的特点在于通过 Petri 网模型; 在逻辑层次上, 为框架所涉及的各项功能, 包括: 任务协调、任务监控, 提供了一致的描述方法。本文提出了一个基于路径关键点提取的动作级任务建模方法, 从而在理论上, 解决了基于 Petri 网的基本任务建模问题。该方法可以用于动作级遥编程。本文提出了一个通过对子任务的 Petri 网模型进行变迁细化, 建立任务的分层 Petri 网模型的方法, 该方法解决了基于 Petri 网的复杂任务建模问题。本文提出了一个基于 Petri 网的任务协调方法, 该方法根据任务空间的谓词/变迁网描述, 实现了从操作员的连续动作到任务的命令序列的转化。通过任务协调, 实现了操作员连续操作和遥机器人监控的结合, 可以用于任务级遥编程。

本文实现了一个动作级遥编程系统, 并基于遥编程建立了一个大时延遥操作试验系统。该系统的特点在于针对任务不同阶段的特点, 通过分段复合控制方案提高了系统的整体性能; 基于虚拟现实技术建立了先进的人机接口, 实现了包括遥编程、任务预显示和操作现场遥现等在内的多种功能。本文通过典型的大时延遥操作仿真试验, 对系统的性能进行了验证。试验结果表明, 在大时延条件下, 系统能够成功完成高精度遥操作任务, 在保证操作的平稳性的同时, 系统的执行效率高于单步监控方式。

本文分别对基于均匀空间分割和基于非均匀空间分割的快速碰撞检测算法进行了深入研究。提出了一种新的基于均匀空间分割的快速碰撞检测算法 USSCD, 该算法与经典的 ICOLLIDE 算法进行了一系列的对比试验, 试验结果表明, 在物体均匀分布的情况下, 随物体数量的增多, USSCD 表现出明显的优势。提出了一类非均匀空间分割方法, 该方法通过两个步骤实现空间分割: 首先, 基于物体分布密度进行粗略的非均匀空间分割; 然后, 对每个子空间进行均匀分割。该方法的特点在于, 在解决

物体分布不均情况下的空间分割的同时,兼顾了物体组织结构的简单性,从而能够较好地满足碰撞检测的需要。在该方法的指导思想下,本文提出了两种非均匀空间分割方法:一种是基于投影的非均匀空间分割方法。在该方法中,通过物体的轴向投影直方图表征空间分布密度,通过区域生长实现空间分割。该方法的特点在于通过对轴向投影的分割形成三维空间的分割,简单直观;另一种方法是基于聚类的自适应空间分割方法。该方法首先通过一个基于局部密度的聚类算法对物体进行聚类,然后根据聚类结果实现空间分割。该方法通过聚类反映物体分布密度的不同,概念直观,然而,由于聚类过程本身的计算复杂性较高,因此使得预处理阶段的计算开销较大。该算法适合于非常复杂的、而且物体的非均匀分布在运行过程中基本保持不变的环境。

6.2 未来的研究方向

就建立灵活的人机协调机制的目标而言,本文的研究只是一个开始。要将整个智能系统的众多功能合理地人机间进行分配,需要一个涵盖不同层次功能的统一的系统框架,建立一致的定量的系统模型。优化的功能分配方案,应该建立在对包括操作员、遥机器人,以及任务环境在内的多方面因素的定量分析的基础上。Petri 网模型在一定层次上为实现这一目标提供了一种可能。基于 Petri 网可以实现一致的任务描述、建立系统的监控理论,以及对系统性能的定量分析。因此,将 Petri 应用于大时延遥操作还需要深入研究。然而, Petri 网本质上是一种离散事件动态模型,定位于逻辑层次,对于系统的连续动态特性, Petri 网具有局限。所以,研究结合离散事件动态模型和连续动态模型的混合动态系统,具有重要的意义。

遥编程为克服大时延提供了一种有效的途径。遥编程涉及的关键技术需要进一步的研究。程序自动生成的关键是针对连续运动的轨迹分段。轨迹分段需要考虑机器人的控制误差以及对连续轨迹的逼近误差等多方面因素。优化的轨迹分段方法亟待建立。由于遥编程本质上是基于模型的控制方法,其有效性取决于虚拟环境和真实环境间的一致性。因此,虚拟环境的误差校正方法需要深入地研究。

碰撞检测一直以来是虚拟现实应用中的关键问题,也是没有完全解决的问题。虚拟现实走向实用,需要准确、高效,而且简单的碰撞检测方法。然而,目前在碰撞检测方面的研究,大多是针对具体的应用环境,碰撞检测方法往往是应用领域相关的(依赖于具体的模型表示、状态信息)。研究相对通用的、独立的、高效的碰撞检测方法具有重要的意义。

致谢

这里，首先要表达我对导师贺汉根教授衷心的感谢和敬意。贺老师深厚的理论基础、严谨的治学态度，以及敏捷的思维，将一直是我学习的目标。贺老师不仅具有较高的学术素养，而且作风踏实，常常亲身科研的第一线，这种理论和实践相结合的作风深深影响着包括我在内的众多学生。在这四年的学习过程中，一直承蒙贺老师的悉心指导。从基础理论学习到科研实践，从具体的工程问题到论文的写作，在我科研素质提高的一一点一滴之中，都渗透着贺老师的谆谆教诲。不仅在学习方面，在为人处世方面也时时感受到贺老师的教诲，积极进取、不记名利、诚以待人，贺老师的这些人生态度和品格，是我人生修养的榜样。贺老师在生活中对学生的关心，使远离父母的我，时常能感受到父母般的关怀。这里，不仅代表我个人，也代表我的父母，再次向贺老师表达衷心的感谢。

在遥操作研究过程中，本人在机器人方面的基础比较薄弱，幸而得到了韦庆、李杰、张辉三位博士的热心帮助，才能以尽快的速度掌握机器人方面的相关知识，保证了课题研究的顺利完成。这里，向他们表达衷心的感谢。这里还要感谢吴涛同学，他以其扎实的数学基础和热心的帮助，使本人能够顺利解决许多理论分析和实验分析问题。感谢安向京、陈璟两位博士，他们在本文的写作过程中，都提供了许多宝贵的意见。此外，一并感谢实验室的史美萍教员、吴晓莉同学、彭晓军同学、卢晓军同学等，他们为本文的顺利完成营造了一个友好、互助、轻松的良好氛围。

这里，还要感谢先父和远方的母亲，他们多年来为了我的成长倾注了大量的心血，无私而温暖的亲情，是我面对挫折和困难重新振作的力量来源。遗憾的是，由于多年远在他乡求学，少有机会侍奉双亲左右、聊尽孝心。无以为报，谨以此文告慰先父在天之灵，给远方的母亲，以及其他关心我的亲人们一些安慰。

最后，感谢我的妻子，周沐女士。在本文的写作过程中，她在本职的教学和科研工作之余，无怨地承担了绝大部分家务。正是她的支持和鼓励，使我一次次走出彷徨、摆脱劳累。如果我能够有所成就，其中必有她的一份期望和辛劳。

参考文献

1. Tsutomu Iwata, Mitsushige Oda, Ryoichi Imai, "Next Generation Space Robot", N90-29899. p245-252.
2. 孙增圻, "空间机器人国外发展状况及我国的发展设想", 空间机器人及遥科学技术研讨会论文集, 1999年8月16-18日, 山东, 烟台, p1-20.
3. 王旭东, 梁斌, 吴宏鑫, 杨保华, "面向21世纪的我国遥科学空间机器人发展的建议", 空间机器人及遥科学技术研讨会论文集, 1999年8月16-18日, 山东, 烟台, p21-29.
4. Dave Iavery, "Robotics Technology Developments in the United States Space Telerobotics Program", N95-14943, p1-19.
5. P. Putz, "Space robotics in Europe: A Survey", Robotics and Automation System, Vol. 23, 1998, p3-16.
6. 洪炳熔, 等, "空间机器人及其发展趋势", 中国有色金属学报, 1995.10, p26-35.
7. Thomas B. Sheridan, "Telerobotics, Automation, and Human Supervisory Control". MIT press.
8. P. Backes, M. Long, R. Steele, "The modular telerobot task execution system for space telerobotics", Proceedings of IEEE International Conference on Robotics and Automation, 1993, p524-530.
9. G. Hirzinger, J. Heindl, K. Landzettel, "Predictive and Knowledge-based Telerobotic Control concepts", Proceedings of IEEE International Conference on Robotics & Automation, 1989, p1768-1777.
10. J. S. Albus, R. Quintero, R. Lumia, "Overview of NASREM: The NASA/NBS Standard Reference Model for Telerobot Control System Architecture", PB94-194560/HDM, P1-9.
11. J. S. Albus, "4-D/RCS: A Reference Model Architecture for Demo III", Proceedings of the 1998 IEEE ISIG/CIRA/ISAS Joint Conference, 1998, p634-639.
12. Yangsheng Xu, Ben Brown, Shigeru Aoki, Tetsuji Yoshida, "A Space Station Robot Walker and its Shared Control Software", N94-30543, P123-130.
13. R. J. Anderson, "Autonomous, Teleoperated, and Shared Control of Robot Systems", Proceedings of the 1996 IEEE, International Conference on Robotics & Automation, April 1996, p2025-2032.
14. R. J. Anderson, D. L. Shirey, W. D. Morse, "A Telerobot Control System for Accident Response", Proceedings of the 2000 IEEE, International Conference on Robotics & Automation, San Francisco, CA, April 2000, p2000-2006.

15. H. D. Stevens, E. S. Miles S. J. Rock R. H. Cannon, "Object-Based Task-Level Control: A Hierarchical Control Architecture for Remote Operation of Space Robots", N94-30560, p264-273.
16. D. R. Lefebvre, G. N. Saridis, "A Computer Architecture for Intelligent Machines", Proceedings of the 1992 IEEE International Conference on Robotics and Automation, Nice, France, May 1992, p2745-2750.
17. L. Michaloski, Paul G. Backes, Ronald Lumia, "Integration of sensor feedback and teleoperation into an open architecture standard", SPIE Vol. 2589, P206-217.
18. L. Conway, R. A. Volz, and M. W. Walker, "Teleautonomous systems: Projecting and Coordinating Inteligent Action at a Distance", IEEE Trans. On Robotics and Automation. Vol. 6, No. 2, April. 1990, p146-158.
19. C. Guo, T. J. Tarn, N. Xi, A, K, Bejczy, "Fusion of Human and Machine Intelligence for Telerobotic Systems", IEEE International Conference on Robotics and Automation, 1995, p3110-3114.
20. B. S. Graves, "A Generalized Teleautonomous Architecture Using Situation-based Action Selection", Dissertation, 1995.
21. S. Graves, R. Volz, "Action Selection in Teleautonomous Systems", IEEE International Conference on Intelligent Robots and Systems v 3 1995. IEEE, Piscataway, NJ, USA,95CB35836. p 14-19.
22. K. M. Brien, "Task-level Control for Networked Telerobotics", Master thesis, 1996.
23. K. Brady, T. J. Tarn, "Internet-based remote teleoperation", Proceedings of the 1998 IEEE International Conference on Robotics and Automation, May 1998, p65-70.
24. R. L. Williams, F. W. Harrison, D. I. Soloway, "Shared Control of Multiple-Manipulator, Sensor-Based Telerobotic Systems", Proceedings of the 1997 IEEE International Conference on Robotics & Automation, April 1997, p962-967.
25. P. Aigner, B. J. McCarragher, "Modeling and Constraining Human Interactions in Shared Control Utilizing a Discrete Event Framework", IEEE Trans. On Systems, Man, and Cybernetics—Part A: Systems and Humans, Vol. 30, No. 3, May 2000, p369-379.
26. Austin, D.J.; McCarragher, B.J., "A unifying discrete event framework for mobile robots", 1998 IEEE International Conference on Systems, Man, and Cybernetics, 1998. Volume: 1 , p782 -787.
27. McCarragher, B.J.; Hovland, G.; Sikka, P.; Aigner, P.; Austin, D. , "Hybrid dynamic modeling and control of constrained manipulation systems" IEEE Robotics & Automation Magazine Vol. 42, June 1997 p27 -44.
28. B. J. McCarragher, "Petri Net modeling for Robotic Assembly and Trajectory Planning", IEEE Trans. On Industrial Electronics, Vol. 41, No. 6, Dec 1994, p631-640.
29. 李华忠, "基于虚拟现实的空间机器人共享控制系统", 博士论文, 1999.
30. G. Hirzinger, B. Brunner, J. Dietrich, and J. Heindl, "Sensor-Based Space Robotics-ROTEX and Its Telerobotic Features", IEEE Trans. On Robotics and

- Automation. Vol. 9, No. 5, Oct. 1993, p649-663.
31. T. B. Sheridan, "Space Teleroperation Through Time Delay: Review and Prognosis", IEEE Trans. On Robotics and Automation. Vol. 9, No. 5, Oct. 1993, p592-606.
 32. J. K. Rosenblatt, C. E. Thorpe, "Combining multiple goals in a behavior-based architecture", IEEE International Conference on Intelligent Robots and Systems v 1 1995. IEEE, Piscataway, NJ, USA,95CB35836. p 136-141.
 33. H. I. Christensen, P. Pirjanian, "Theoretical Methods for Planning and Control in Mobile Robotics", International Conference on Knowledge-Based Intelligent Electronic Systems, 1997. KES '97. Proceedings., 1997. Vol. 1 , p81-86
 34. M. Huber, "A Hybrid Architecture for Hierarchical Reinforcement Learning", Proceedings of the 2000 IEEE International Conference on Robotics and Automation, April 2000, p3290-3295.
 35. M. Huber, "A Hybrid Discrete Event Dynamic Systems Approach to Robot Control", Technical Report #96-43, University of Massachusetts.
 36. M. Lindström, A. Orebäck, H. I. Christensen, "DERRA: A Research Architecture for Service Robots", Proceedings of the 2000 IEEE International Conference on Robotics & Automation, San Francisco, CA, April 2000, p3278-3283.
 37. D. Mali, "Tradeoffs in Making the Behavior-Based Robotic Systems Goal-Directed" Proceedings of the 1998 IEEE International Conference on Robotics & Automation, Leuven Belgium, May 1998, p1128-1133.
 38. You Song, etc, "Share Control in Intelligent Arm/HandTeleoperatedSystem", Proceedings of the 1999 IEEE International Conference on Robotics & Automation, May 1999, p2489-2494.
 39. 蔡自兴, "递阶智能控制系统的一般结构", 控制与决策, Vol. 14, No. 6, Nov. 1999, p642-647.
 40. 李杰, "监控式多机器人协调控制技术的研究", 博士论文, 1999.
 41. 陈璟, "智能机器人动力学整合机制以及监控自主控制器研究", 博士论文, 1999.
 42. Blake Hannaford, "Stability and Performance tradeoffs in bi-lateral telemanipulation", Proceedings of IEEE International Conference on Robotics & Automation, 1989, p1764-1767.
 43. J. S. McLaughlin, B. D. Staunton, "Performance Constraints and Compensation for Teleoperation With Delay", N90-29881, p63-72.
 44. Dale A. Lawrence, "Stability and Transparency in Bilateral Teleoperation", IEEE Trans. On Robotics and Automation. Vol. 9, No. 5, Oct. 1993, p624-637.
 45. Sukhan Lee, and Hahk Sung Lee, "Modeling, Design, and Evaluation of Advanced Teleoperator Control Systems with Short Time Delay", IEEE Trans. On Robotics and Automation. Vol. 9, No. 5, Oct. 1993, p607-623.
 46. Bob Bicker & Ow Sin Ming, "Shared Control in Bilateral Telerobotics Systems", SPIE

-
- Vol. 2351 Telemanipulator and Telepresence Technologies, 1994, p200-206.
47. Kazuhiro KOSUGE, Koji TAKEO and Toshio FUKUDA, "Unified Approach for Teleoperation of Virtual and Real Environment-Manipulation Based on Reference Dynamics", Proceedings of IEEE International Conference on Robotics & Automation, 1995, p938-942.
 48. A. K. Beiczy, W. S. Kim, and S. C. Venema, "The Phantom Robot: Predictive Displays for Teleoperation with Time Delay", Proceedings of IEEE International Workshop Intelligent Robots System, IROS, 1990. p546-550.
 49. W. S. Kim, P. S. Schenker, A. K. Bejczy, S. Leake, and S. Ollendorf, "An Advanced Operator Interface Design with Preview/Predictive Displays for Ground-Controller Space Telerobotic Servicing", SPIE Vol. 2057, p96-107.
 50. M. Mitsuishi, T. Hori, and T. Nagao, "Predictive, Augmented and Transformed Information Display for Time Delay Compensation in Tele-Handling/Machining", Proceedings of IEEE International Conference on Robotics & Automation, 1995, p45-52.
 51. B. J. Nelson, and P. K. Khosla, "Integrating Force and Vision Feedback Within Virtual Environments for Telerobotic Systems", Proceedings of IEEE International Conference on Robotics & Automation, 1997, p1588-1593.
 52. Anu Rastogi, "Design of an Interface for Teleoperation in Unstructured Environments using Augmented Reality Displays", Master Thesis, 1996. University of Toronto, Toronto, Canada.
 53. Paul Milgram, Anu Rastogi, Julius J. Grodski, "Telerobotic Control Using Augmented Reality" Proceedings 4th IEEE International Workshop on Robot and Human Communication (ROMAN'95) 5-7 July, 1995, Tokyo.
 54. Martin Jägersand, "Image Based Predictive Display for Tele-Manipulation", Proceedings of IEEE International Conference on Robotics & Automation, 1999, p550-556.
 55. M. Barth, T. Burkert, C. Eberst, N. O. Stöffler, and G. Färber "Photo-Realistic Scene Prediction of Partially Unknown Environments for the Compensation of Time Delays in Telepresence Applications" Proceedings of the 2000 IEEE International Conference on Robotics & Automation San Francisco, CA. April 2000, 3132-3137.
 56. R. L. Smith, D. J. Gillan, and M. Stuart, "Telepresence for space: The state of the concept", N90-25526. p223-228.
 57. T. B. Sheridan, "Human Factors Considerations for Remote Manipulation", N95-14949, p1-24.
 58. J. Garin, J. Matteo, and V. A. Jennings, "Telepresence and Telerobotics", N89-19873, p411-420.
 59. F. S. Schebor and J. L. Turney, "Real-time, Interactive, Visually Updated Simulator System for Telepresence", N92-22683, p60-70.
-

60. A. L. Gregory Tharp, L. F. Stephen Lai, and L. Stark, "Some of What One Needs to Know about Using Head-Mounted Displays to Improve Teleoperator Performance", IEEE Trans. On Robotics and Automation. Vol. 9, No. 5, Oct. 1993, p638-647.
61. A. Kheddar, K. Tanie, and P. Coiffet, "Detection of discrepancies and sensory-based recovery for virtual reality based telemanipulation systems", Proceedings of IEEE International Conference on Robotics & Automation, 1998, p2877-2883.
62. Y. Tsumaki and M. Uchiyama, "A Model-Based Space Teleoperation System with Robustness against Modeling Errors", Proceedings of the 1997 IEEE International Conference on Robotics & Automation, 1997, p1594-1599.
63. Y. Wakita and S. Hirai, "Hierarchical Control of a Visual Monitoring System for Telerobot Tasks", Proceedings of the 1993 IEEE/RSJ International Conference on Intelligent Robots and Systems, 1993, p321-326.
64. Richard Paul, Thomas Lindsay, Craig Sayers, "Time Delay Insensitive Teleoperation", Proceedings of the 1992 IEEE/RSJ International Conference on Intelligent Robots and Systems, Raleigh, NC July 7-10, 1992, p247-254.
65. Stein, Matthew R.; Paul, Richard P.; Schenker, Paul S.; Paljug, Eric D., "Cross-country teleprogramming experiment", IEEE International Conference on Intelligent Robots and Systems v 2 1995. IEEE, Piscataway, NJ, USA. p 21-26.
66. Sayers, Craig P.; Lai, Angela; Paul, Richard P., "Visual imagery for subsea teleprogramming", IEEE International Conference on Robotics and Automation v 2 1995. IEEE, Piscataway, NJ, USA, 95CB3461-1. p 1567-1572.
67. Jérôme Perret Christophe Proust Rachid Alami, "How to Tele-Program a Remote Intelligent Robot", IEEE/RSJ/GI International Conference on Intelligent Robots and Systems v 1 1994. IEEE, Piscataway, NJ, USA. p 397-404.
68. Sayers, Craig; Paul, Richard, "Synthetic fixturing", Advances in Robotics, Mechatronics and Haptic Interfaces American Society of Mechanical Engineers, Dynamic Systems and Control Division (Publication) DSC v 49, 1993. Publ by ASME, New York, NY, USA. p 37-46.
69. M. Y. Kaczor, Ch. Laugier, E. Mazer, "TELE-ACT: A Task Level Teleprogramming System", Proceedings of 1993 IEEE/RSJ International Conference on Intelligent Robots and Systems Yokohama, Japan July 26-30, 1993, p310-313.
70. Cho, Young-Jo; Kotoku, Tetsuo; Tanie, Kazuo, "Discrete-event-based planning and control of telerobotic part-mating process with communication delay and geometric uncertainty", IEEE International Conference on Intelligent Robots and Systems v 2 1995, IEEE, Piscataway, NJ, USA. p 1-6.
71. Stein, Matthew R.; Sayers, Craig P.; Paul, Richard P., "Recovery from task execution errors during time delayed teleoperation", IEEE/RSJ/GI International Conference on Intelligent Robots and Systems v 1, 1994. IEEE, Piscataway, NJ, USA. p 547-554.
72. Matthew R. Stein and Richard P. Paul, "Kinesthetic Replay for Error Diagnosis in Time

- Delayed Teleoperation”, SPIE Vol. 1833 Telemanipulator Technology (1992), p278-287.
73. H. Friedrich, J. Holle R. Dillmann, “Interactive Generation of Flexible Robot Programs”, Proceedings of 1998 IEEE International Conference on Robotics & Automation, Leuven, Belgium May 1998, p538-543.
74. John E. Lloyd, Jaffrey S. Beis, Kinesh K. Pai, David G. Lowe, “Model-based Telerobotics with Vision”, Proceedings of the 1997 IEEE International Conference on Robotics and Automation Albuquerque, New Mexico, April 1997.
75. J. R. Chen, B. J. Mcarragher, “Programming by Demonstration – Constructing Task Level Plans in a Hybrid Dynamic Framework”, IEEE International Conference on Robotics and Automation. IEEE, San Francisco, CA 2000. p 1402-1407.
76. J. R. Chen, B. J. Mcarragher, “Robot Programming by Demonstration – Selecting Optimal Event Paths”, IEEE International Conference on Robotics and Automation. IEEE, Leuven Belgium, May 1998. p 518-523.
77. Sing Bing Kang, “Robot Instruction by Human Demonstration”, Dissertation, 1994.
78. Donald R. Myers, ” An Approach to Automated Programming of Industrial Robots” Proceedings of the 1999 IEEE International Conference on Robotics& Automation Detroit, Michigan l May 1999, p3109-3114.
79. S. MÜNCH, J. KREUZIGER, M. KAISER, R. DILLMANN, “Robot Programming by Demonstration (RPD) - Using Machine Learning and User Interaction Methods for the Development of Easy and Comfortable Robot Programming Systems”, Proceedings of the 24th International Symposium on Industrial Robots (ISIR '94).
80. Masayuki Tsuda, Hiroyuki Ogata, Yoshito Nanjo, “Programming Groups of Local Models from Human Demonstration to Create a Model for Robotic Assembly” Proceedings of the 1998 IEEE International Conference on Robotics & Automation Leuven, Belgium. May 1998. p530-537.
81. D. Baraff, “Analytical Methods for Dynamic Simulation of Non-penetrating Rigid Bodies”, Gomputer Graphics, Vol. 23, No. 3, July 1989, p223-230.
82. D. Baraff, “Curved Surfaces and Coherence for Non-penetrating Rigid Body Simulation”, Computer Graphics, Vol. 24, No. 4, August 1990, p19-28.
83. V. V. Kamat, “A Survey of Techniques for Simulation of Dynamic Collision Detection and Response”, Computer & Graphics, Vol.17, No.4, 1993, p379-385.
84. M. Moore, and J. Wilhelms, “Collision Detection and Response for Computer Animation”, Computer Graphics, Vol. 32, No. 4, August 1988, p289-298.
85. B. Mirtich, and J. Canny, “Impulse-based simulation of rigid bodies.” Proceedings of ACM Interactive 3D Graphics, Monterey, CA, 1995.
86. M. C. Lin, and S. Gottschalk, “Collision detection between geometric models: a survey”, In Proceedings of IMA Conference on Mathematics of Surfaces, 1998, Vol. 1, p602-608.

87. P. Jiménez, F. Thomas, and C. Torras, "3D Collision Detection: A Survey", *Computer & Graphics*, Vol. 25, No. 2, April 2001, p269-285.
88. Ming Lin and John Canny. "A fast algorithm for incremental distance calculation" *Int. Conf. Robotics & Automation*, pages 1008–1014, Sacramento, April 1991.
89. S. A. Cameron, "Collision detection by four-dimensional intersection testing", *IEEE Trans. Robotics Automat.* Vol.6, No. 3, 1990, p291-302.
90. Gilbert, Elmer, Johnson, Daniel, Keerthi, S. "A Fast Procedure for Computing the Distance Between Complex Objects in Three-dimensional Space" *IEEE Journal of Robotics and Automation*, pp. 193-203, April 1988.
91. Quinlan, Sean. "Efficient Distance Computation between Non-Convex Objects" *IEEE Int. Conference on Robotics and Automation*, pp. 3324-3329, 1994.
92. M. Held, J. Klosowski, and Joseph S. B. Mitchell. "Real-time collision detection for motion simulation within complex environments". In *Proc. ACM SIGGRAPH'96 Visual Proceedings*, page 151, 1996.
93. M. C. Lin, "Efficient Collision Detection for Animation and Robotics", PhD thesis, Department of Electrical Engineering and Computer Science, University of California, Berkeley, December 1993.
94. D. J. Kim, L. J. Guibas, and S. Y. Shin, "Fast Collision Detection Among Multiple Moving Spheres", *IEEE Trans. On Visualization and Computer Graphics*, Vol. 4, No. 3, 1998, p230-242.
95. Cohen, Jonathan D.; Lin, Ming C.; Manocha, Dinesh; Ponamgi, Madhav, "I-COLLIDE: An interactive and exact collision detection system for large-scale environments", *Proceedings of the Symposium on Interactive 3D Graphics 1995*. ACM, New York, NY, USA. p 189-196.
96. Hubbard, P.M., "Interactive collision detection", *Proceedings of IEEE 1993 Symposium on Research Frontiers in Virtual Reality*, 1993. p24-31.
97. Hubbard, Philip M., "Collision detection for interactive graphics applications", *IEEE Transactions on Visualization and Computer Graphics* v 1 n 3 Sep 1995. p 218-230.
98. D. E. Johnson, and Elaine Cohen, "Bound Coherence for Minimum Distance Computations", *Proceedings of 1999 IEEE International Conference on Robotics & Automation*, 1999, p1843-1848.
99. E. Larsen, S. Gottschalk, M. C. Lin, and D. Manocha, "Fast Distance Queries with Rectangular Swept Sphere Volumes", *Proceedings of 2000 IEEE International Conference on Robotics & Automation*, 2000, p3719-3762.
100. S. Cameron, "Enhancing GJK: Computing Minimum and Penetration Distances between Convex Polyhedra", *Proceedings of 1997 IEEE International Conference on Robotics & Automation*, 1997, p3112-3117.
101. A. Wilson and E. Larson and D. Manocha, and M. C. Lin, "Partitioning and Handling Massive Models for Interactive Collision Detection", *EUROGRAPHICS'99*, Vol. 18,

- No. 3, 1999, C319-C329.
- 102.Klosowski, James T.; Held, Martin; Mitchell, Joseph S.B.; Sowizral, Henry; Zikan, Karel, "Efficient collision detection using bounding volume hierarchies of k-DOPs", IEEE Transactions on Visualization and Computer Graphics v 4 n 1 Jan-Mar 1998. p 21-36.
- 103.G. Barequet, B. Chazelle, L. J. Guibas, J. S. B. Mitchell, A. Tal, "BOXTREE: A Hierarchical Representation for Surfaces in 3D", EUROGRAPHICS'96, Vol. 15, No. 3, 1996, C387-C396.
- 104.M. K. Ponamgi, D. Manocha, and M. C. Lin, "Incremental Algorithms for Collision Detection Between Polygonal Models", IEEE Trans. On Visualization and Computer Graphics, Vol. 3, No.1, 1997, p51-64.
- 105.S. Gottschalk, M. Lin, and D. Manocha, "OBBTree: A Hierarchical structure for rapid interference detection", In Proc. Of ACM SIGGRAPH'96, 1996, p171-180.
- 106.A. C. Nearchou, and N. A. Aspragathos, and D. P. Sofotassios, "Reducing the Complexity of Robot's Scene for Faster Collision Detection", Journal of Intelligent and Robotic Systems, Vol. 26, 1999, p79-89.
- 107.I. J. Palmer, and R. L. Grimsdale, "Collision Detection for Animation using Sphere-Trees", Computer Graphics Forum, Vol.14, No. 2, 1995, p105-116.
- 108.张宇, 马丽红, 余英林, "空间模型碰撞检测的向量符号分析方法与实现", 中国图像图形学报, Vol.3, No. 6, Jun. 1998, p481-484.
- 109.任世军, 洪炳熔, 孟庆鑫, "判断两个凸多面体是否相交的一个快速算法", 软件学报, 2000, 11 (4), 563-568.
- 110.Hyun Suk Kim; Heedong Ko; Kunwoo Lee; Chong-Won Lee, "A collision detection method for real time assembly simulation", Proceedings., IEEE International Symposium on Assembly and Task Planning, 1995. p387 -392.
- 111.Smith, A.; Kitamura, Y.; Takemura, H.; Kishino, F., "A simple and efficient method for accurate collision detection among deformable polyhedral objects in arbitrary motion", Proceedings of Virtual Reality Annual International Symposium, 1995. p136 -145.
- 112.P. M. Hubbard, "Collision Detection for Interactive Graphics Application", PhD Thesis, Department of Computer Science, Brown University, April 1995.
- 113.Tzafestas, C.; Coiffet, P., "Real-time collision detection using spherical octrees: virtual reality application", 5th IEEE International Workshop on Robot and Human Communication, 1996., p500-506.
- 114.P. G. Xavier, "Fast Swept-Volume Distance for Robust Collision Detection", Proceedings of IEEE Interantional Conference on Robotics and Automation, 1997. p1162-1169.
- 115.Chuan-Jun Su; Lin Fu-Hua; Xiao-ke Zhang, "An efficient collision detection methodology for virtual assembly", 1998 IEEE International Conference on Systems,

- Man, and Cybernetics, 1998. Vol.1, p360 –365.
- 116.Zachmann, G., “Rapid collision detection by dynamically aligned DOP-trees”, Virtual Reality Annual International Symposium, 1998. Proceedings., IEEE 1998, p90-97.
- 117.D. E. Johnson, and E. Cohen, “A Framework for Efficient Minimum Distance Computations”, Proceedings of IEEE International Conference on Robotics and Automation, 1998, p3678-3684.
- 118.C. G. Ong, E. G. Gilbert, “The Gilbert-Johnson-Keerthi Distance Algorithm: A Fast Version for Incremental Motions”, Proceedings of IEEE International Conference on Robotics and Automation, 1997. p1183-1189.
- 119.Baciu, G.; Wingo Sai-Keung Wong; Hanqiu Sun, “RECODE: an image-based collision detection algorithm”, Computer Graphics and Applications, 1998. Pacific Graphics '98. Sixth Pacific Conference on, p125 –133.
- 120.J. T. Klosowski, “Efficient Collision Detection for 3D Graphics and Virtual Environments”, PhD Thesis, State University of New York at Stony Brook, May 1998.
- 121.P. Jiménez, C. Torras, “Benefits of Applicability Constraints in Decomposition-free Interference Detection between Nonconvex Polyhedral Models”, Proceedings of IEEE International Conference on Robotics and Automation, 1999. p1856-1862.
- 122.Gregory, A.; Lin, M.C.; Gottschalk, S.; Taylor, R., “A framework for fast and accurate collision detection for haptic interaction”, Virtual Reality, 1999. Proceedings., IEEE. p38 –45.
- 123.S. Redon, A. Kheddar, and S. Coquillart, “An Algebraic Solution to the Problem of Collision Detection for Rigid Polyhedral Objects”, Proceedings of IEEE International Conference on Robotics and Automation, 2000. p3733-3738.
- 124.Maria del C. Amdzquita Benftez Kamal K. Gupta, Binay Bhattacharya “EODM - A Novel Representation For Collision Detection”, Proceedings of IEEE International Conference on Robotics and Automation, 2000. p3727-3732.
- 125.S. Gottschalk, “Collision Queries using Oriented Bounding Boxes”, PhD Thesis, Department of Computer Science, Chapel Hill, 2000.
- 126.A. Glassner, Space subdivision for fast ray tracing. IEEE Computer Graphics & Applications, Oct. 1984, p15-22.
- 127.A. Fujimoto, T. Tanaka, and K. Iwata, “Arts: Accelerated ray-tracing system”, IEEE Computer Graphics & Applications, 1986, p16–26.
- 128.J. Cleary and G. Wyvill, “Analysis of an algorithm for fast ray tracing using uniform space subdivision”, The Visual Computer, No.4, 1988, p65–83.
- 129.A. Glassner, “Spacetime ray tracing for animation”, IEEE C. G. & A., Vol. 8, No. 2, 1988. p60–70.
- 130.D. Jevans and B.Wyvill, “Adaptive voxel subdivision for ray tracing”, in Proc. of Graphics Interface '89, June 1989, p164–72.
- 131.F. Cazals, G. Drettakis, and C. Puech, “Filtering, Clustering and Hierarchy

- Construction: a New Solution for Ray-Tracing Complex Scenes”, *Computer Graphics Forum*, Vol.14, No. 3.
- 132.Xu, Xiaowei, Martin Ester, Hans-Peter Kriegel and Jörg Sander. (1998). “A Distribution-Based Clustering Algorithm for Mining in Large Spatial Databases”, *Proceedings of the 14th International Conference on Data Engineering (ICDE 98)*. Orlando, Florida. pp. 324-331.
- 133.Ester, Martin, Hans-Peter Kriegel, Jörg Sander and Xiaowei Xu. (1996). “A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise”. *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining (KDD-96)*. Portland, Oregon, August 1996.
- 134.Wang, Wei, Jiong Yang, and Richard Muntz. (1999). “STING+: An Approach to Active Spatial Data Mining”, *Proceedings of the International Conference on Data Engineering (ICDE99)*. Sydney, Australia. pp. 116-125.
- 135.Ng, Raymond T. and Jiawei Han. (1994). “Efficient and Effective Clustering Methods for Spatial Data Mining”. *Proceedings of the 20th Very Large Databases Conference (VLDB 94)*, Santiago, Chile. pp 144-155.
- 136.Zhang, Tian, Raghu Ramakrishnan, and Miron Livny. (1996). “BIRCH: An Efficient Data Clustering Method for Very Large Databases”. *Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data*, Montreal, Canada, 1996. pp. 103—114.
- 137.P. Bajcsy, N. Ahuja, "Location and Homogeneity Based Hierarchical Clustering", *IEEE Trans. Pattern Anal. Machine Intell.* vol. 20, 1998, p1011-1015.
- 138.D. Fasulo. An analysis of recent work on clustering algorithms. Technical Report UW-CSE-01-03-02, University of Washington, 1999.
- 139.B. J. McCarragher, “Petri Net Modelling for Robotic Assembly and Trajectory Planning”, *IEEE Trans. On Industrial Electronics*, Vol. 41, No. 6, Dec, 1994, p631-640.
- 140.T. Cao, A. C. Sanderson, “AND/OR Net Representation for Robotic Task Sequence Planning”, *IEEE Trans. On Systems, Man, and Cybernetics*, Vol. 28, No. 2, May, 1998, p204-218.
- 141.张伟军, 彭商贤等, “基于 PETRI 网的机器人装配作业建模及装配路径由化方法的研究”, *机器人*, Vol. 20, No. 3, May. 1998, p207-213.
- 142.F. Y. Wang, etc., “A Petri-Net Coordination Model for an Intelligent Mobile Robot”, *IEEE Trans. On Systems, Man and Cybernetics*, Vol. 21. No.4, July, 1991, p777-789.
- 143.P. Lima, etc., “Petri Nets for Modelling and Coordination of Robotic Tasks”, *Proceedings of 1998 IEEE International Conference on Systems, Man, and Cybernetics*, 1998. Vol.1, p190-195.
- 144.Keigo Kobayashi and Toshimistu Ushio, “An Application of LLP Supervisory Control with Petri Net Models in Mobile Robots”, *Systems, Man, and Cybernetics*, 2000 IEEE International Conference, vol.4, p3015 –3020.

145. Y. J. Cho, T. Kotoku, and K. Tanie, "Discrete-Event-Based Planning and Control of Telerobotic Part-Mating Process with Communication Delay and Geometric Uncertainty", *Proceedings of the 1995 IEEE LRSJ International Conference on Intelligent Robots and Systems (IROS)*, Pittsburgh, Pennsylvania, August 1995, pp. 1-6 (Vol. 2).
146. M.; Rauterberg, S.; Schluep, M., Fjeld, "How to model behavioural and cognitive complexity in human-computer interaction with Petri nets", *Robot and Human Communication, 1997. RO-MAN '97. Proceedings., 6th IEEE International Workshop on*, p320 -325
147. 刘巧光, 蔡自兴, "基于 Petri 网的智能控制结构建模", *中南工业大学学报*, Vol. 26, No. 2, April. 1995, p236-240.
148. R. Fehling, "A Concept of Hierarchical Petri Nets with Building Blocks", *Advances in Petri Nets 1993, Lecture Notes in Computer Science*, p148-168.
149. W. M. Zuberek, I. Bluemke, "Hierarchies of Place/Transition Refinements in Petri Nets", *Proceedings of the 5-th IEEE International Conference on Emerging Technologies and Factory Automation*, Kauai, Hawaii, November, 1996, pp.355-360.
150. H. J. Genrich, "Predicate/transition nets", *Lecture Notes in Computer Science*, 1986, p207-247.
151. S. Ramaswamy, "Hierarchical Time-Extended Petri Nets (H-EPNs) for Integrated Control and Diagnostics of Multilevel Systems", *Dissertation*, 1994.
152. W. M. Zuberek, "Petri Nets in Hierarchical Modeling of Manufacturing Systems", *IFAC Conf. On Control System Design (CSD'2000) Special Session on Petri Nets in Design, Modeling and Simulation of Control Systems*; Bratislava, Slovakia, 18-20 June 2000. p287-292.
153. X. He, and C. H. Yang, "Structured Analysis Using Hierarchical Predicate Transition Nets", 1992, p212-217.
154. 袁崇义, "Petri 网原理", 电子工业出版社, 1998。
155. 蒋昌俊, "Petri 网理论与方法研究综述", *控制与决策*, Vol. 12, No. 6, Nov. 1997, p631-636.
156. Roberg, J., "A Data Reduction Algorithm for Planar Curves", *Comput Vision Graphics Image Process*, 1985, 29(2). P168-195.
157. Wu J. S., Leou J. J., "New Polygonal Approximation Schemes for Object Shape Representation", *Pattern Recognition*, 1993, 26(4), p471-484.
158. Fischler M. A., Wolf H. C., "Locating Perceptually Salient Points on Planar Curves", *IEEE Trans on PAMI*, 1992, 16(2). P113-129.
159. 李焱, 刘肖琳, "多尺度分段线性化算法", *模式识别与人工智能*, Vol.10, No.2, 1997, p127-132.
160. 韦庆、李杰, "基于事件反馈的机器人监控技术及在大时延遥操作中的应用", 空

间机器人及遥科学技术研讨会, 1999.8。

161.G. Turk, "Interactive collision detection for molecular graphics". Master's thesis, Computer Science Department, University of North Carolina at Chapel Hill, 1989.

研究工作及成果

作者在攻读博士学位期间以主要参加者，参与了如下课题的研究：

1. 国家自然科学基金资助项目“VR 技术在大时延空间机器人中的应用”（项目编号：68685009，1997-1999）。已经验收结题。评价为优。
2. 国家 863-2 空间机器人与遥科学项目“VR 技术在遥科学中的应用”（1999.1-2000.12）。已经验收结题。获高度评价。
3. 国防科技重点实验室基金试点项目“基于 VR 技术的遥控操作系统技术研究”（2000.12-2002.12）。在研。
4. “十五”国防预研项目“基于虚拟现实的维修性与维修工作分析及评价系统”（2001-2005）。在研。

作者在攻读博士学位期间发表了如下论文：

1. Li Yan Wei Qing He Hangen, “An Virtual Environment Integrated Teleoperation System”, Proceedings of The International Conference on Modeling and Simulation in Distributed applications. September 25-27, 2001, Changsha, China, p433-437.
2. 李焱, 贺汉根, “一个面向遥科学实验的遥操作实验系统”, 宇航学报, 2001 年, No. 6.
3. 李焱, 贺汉根, “应用遥编程的大时延遥操作技术”, 机器人, Vol.23, No.5, 2001.
4. 李焱, 吴涛, 贺汉根, “应用虚拟现实的遥操作机器人技术”, 国防科技大学学报, 2001 年, No. 5.
5. 李焱, 贺汉根, “应用遥编程的月球探测机器人遥操作技术”, 第二届月球探测技术研讨会论文集, 2001 年 1 月, 北京.
6. 李焱 戴斌 贺汉根, “应用遥编程技术的大时延遥操作系统”, 中国 2000 年机器人学大会 (CCR'2000), 2000.
7. 段红, 李焱, 王跃峰, “遥科学实验中的虚拟现实仿真技术研究”, 中国国防科学技术报告, 2000 年.
8. 李焱, 贺汉根, “虚拟现实技术在大时延遥操作中的应用”, 空间机器人及遥科学技术研讨会论文集, 1999 年 8 月, 烟台.