

MRCS 中机器人控制体系框架结构*

范 永 谭 民

(中国科学院自动化研究所 北京 100080)

摘 要 针对机器人协作系统中任务协作和运动协调的两个过程,考虑机器人自主性和协作性的要求,设计一种分层的机器人控制体系结构,其中包括协作规划层 协调规划层和行为控制层。该控制体系采用分层 模块式结构,有利于功能的扩充。

关键词 机器人控制体系,分层结构,多机器人协作系统

分类号 TP 24

Architecture of Robotic Control System in MRCS

Fan Yong, Tan Min

(Institute of Automation, Chinese Academy of Sciences)

Abstract A layered architecture of robotic control system in MRCS (Multi-Robot Cooperation System) was designed. In accordance with the process of task cooperation and motion coordination in MRCS, this architecture was divided into three layers: cooperation-planning layer, coordination-planning layer and behavior-control layer. This architecture can meet the demand of MRCS, and is easy to expand functions.

Key words architecture of robotic control system, layered architecture, multi-robot cooperation system

1 引 言

目前,多机器人协作控制已引起普遍重视^[1-3]。这是由于单个机器人在信息的获取、处理及控制能力等方面存在局限性,难以适应复杂的工作任务及多变的工作环境,而多机器人协作系统(MRCS)却具有单个机器人无法比拟的优越性(如并行性、柔性、鲁棒性等)。于是人们考虑通过多机器人的协作来弥补机器人个体能力的不足,扩大完成任务的能力范围以及提高完成任务的效率。

80 年代后期以来,国外 MRCS 的研究已在理论、实践等方面取得了很大进展,并建立了一些仿真、实验系统,如 CEBOR^[4,5]、SWARM^[6]、AC-TRESS^[7]、GOER^[8]、ALLIANCE^[9] 和 Army-ant^[10] 等系统。国内在这方面的研究也逐渐引起重视,如中科院沈阳自动化所研究的 DAMAS 系统^[11],就是针对装配作业建立的分布协作式多机器

人系统。

在 MRCS 中,机器人控制系统是机器人的核心部分,同时也决定了机器人的协作能力。本文面向 MRCS,提出一种适合多机器人协作执行任务要求的分层机器人个体控制体系结构,其中包括协作规划、协调规划和行为控制 3 个层次。

2 MRCS 中机器人个体控制体系的框架结构

多机器人系统的研究主要是研究机器人的协作控制与协调控制。文献[3]明确指出,在多机器人系统中,协作与协调是两个不同而又相互联系的概念:协作是研究高层的组织与协作机制问题,主要研究多机器人之间的协作机制,多机器人系统的组织形式等;而协调则着重研究机器人之间协作关系确定后具体的运动控制问题。因此,多机器人系统的协作和协调是两个不同的过程,协作过程是一个任务级的协调过程,而协调过程则是一个动作级的协调过

程。有别于多机器人协调系统,MRCS 的研究涉及到以下几个方面^[12]: 1) 机器人系统结构; 2) 群体体系结构; 3) 社会协作规则。

理解多机器人之间协作与协调的关系以及 MRCS 系统的要求,将有利于进行 MRCS 中机器人个体控制体系结构的设计。

由于机器人控制系统是机器人的核心部分,决定了 MRCS 中机器人的协作能力,因此 MRCS 要求机器人不仅具备内部管理、轨迹规划及控制功能,而且具备信息交互、分析推理及组织管理等功能。基于 MRCS 的要求,机器人个体控制系统的设计应考虑感知能力、局部规划能力、通讯能力、任务分解能力、任务分配能力、学习能力以及控制能力等问题^[12]。

考虑 MRCS 中机器人的自主性和协作性要求,协作与协调的关系以及 MRCS 系统的开放性、动态组织的特点,同文献[13]一样,本文的机器人个体控制体系的框架结构也设计成一个三层结构形式,但层次的划分及每层的功能有别于文献[13]设计的 Agent。

如图 1 所示,本文的机器人控制体系结构包括协作规划层(CpPL)、协调规划层(CdPL)以及行为控制层(BCL)。

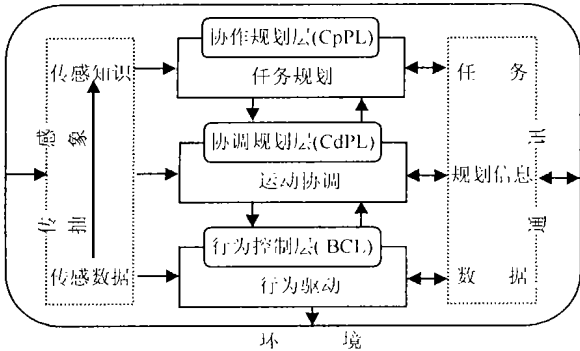


图 1 MRCS 系统中机器人控制体系的框架结构

2.1 协作规划层(CpPL)

如图 2 所示,协作规划层是为满足机器人的任务协作要求而设计的。在 MRCS 中,许多复杂的任务需要由多个机器人协同完成,而协作层赋予机器人协作能力和组织能力等。协作规划层主要实现任务的承接、分解、分配,以及机器人之间的通讯、协商等功能的管理。机器人在承接一个任务后,要根据任务的类型、大小以及自身能力,判定是独立执行任务还是组织其他机器人协作执行。若决定协作执行时,则将任务分解成若干子任务,按某种协议与其他机

器人磋商,制定联合行动计划;若决定独立执行,则将任务交给协调规划层处理。

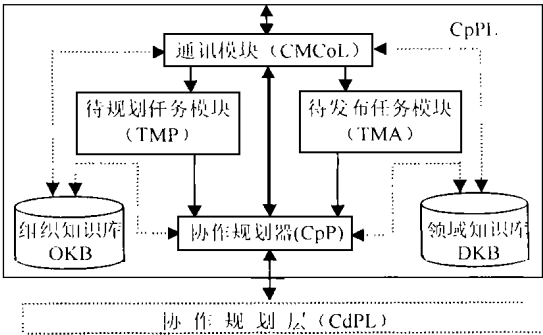


图 2 协作规划层功能结构

CpPL 的各个模块功能如下:

- 1) 协作层通讯模块(CMCoL): 负责机器人之间任务级的信息传递,如任务的承接、发布等;
- 2) 待规划任务模块(TMP): 存放来自用户或其他机器人的招标任务;
- 3) 待发布任务模块(TMA): 存放协作规划模块中任务分解后需其他机器人协作执行的任务;
- 4) 组织知识库(OKB): 组织模块中存放社会规则、谈判策略以及组织结构重组等方面的知识;
- 5) 领域知识库(DKB): 领域问题求解知识,包括规划任务的规则、前提数据等;
- 6) 协作规划器(CpP): 是协作规划层的核心,根据领域知识将承接的任务进行分解(若不需分解则将任务交给协调层处理),基于确定的协商策略和协商协议,采用协商的方式实现机器人之间任务的动态分配。

协作规划算法如下:

Step1: 任务承接: $R_i \leftarrow \text{Task} - \text{description}$, 其中 R_i 表示机器人, $\text{Task} - \text{description}$ 表示对任务的描述。

Step2: 任务分解: 采用专用算法或通用算法(如与 / 或解图)对任务进行分解(若不需分解则转 Step6),并根据子任务的串 / 并关系规划子任务的执行顺序。

Step3: 任务规划: 对每一子任务 $\text{Task} - \text{description}$, 规划所需机器人的类型和个数,表示为一个三元组: $(\text{Robot}, \text{Task} - \text{description}, \text{Dependence})$, 其中 Dependence 表示机器人和任务之间的关系,如: (R_i, T_i, ϕ) 表示 R_i 独立执行任务 T_i , $(R_i, T_i, (R_{jk}, T_{jk}))$ 表示机器人 R_i 是联合任务 T_i 的协调者,它将任务 T_i 分解为 m 个子任务 T_{jk} ,

并由相应的机器人 R_{jk} 协作执行。

Step4: 任务分配: 采用协商方法实现任务的动态分配。协商(Negotiation) 可以表示为一多元组^[13], 设 D 为一协商域, 则

$$\text{Negotiation} = (\mathcal{R}, \rho, N, U, P, S)$$

其中

$\mathcal{R} = \{\text{Robot}_1, \dots, \text{Robot}_k\} (k \geq 2)$ 为一机器人

群体;

$R = \{r_1, \dots, r_l\} (l \leq k)$ 为一角色(Role) 集;

ρ 为角色分配函数, 实现机器人的角色分配,

$\rho(\text{Robot}) = r, \text{Robot} \in \mathcal{R}, r \in R$;

$N \subseteq D$ 为协商集;

$U = \{u_1, \dots, u_k\}, u_i: N \rightarrow R$ 是 Robot_i 的效用函数;

$P = (K, \{\pi: R \times K \rightarrow 2^R \mid r \in R\})$ 表示协商协议, 其中, $K = \{k_1, \dots, k_m\}$ 表示通讯原语(Primitives) 的集合, π 表示一种映射关系, 实现从通讯原语到考虑协议中特定的角色所容许反映的转换;

$S = \{\alpha: P \times R \times K \times 2^D \times U \rightarrow K \times N \mid 1 \leq i \leq k\}$ 表示一协商策略集合。

Step5: 若任务分配失败, 则转 Step3。

Step6: 任务规划完毕。

在任务规划过程中, 任务分解粒度的大小将直接影响规划的效率: 任务分解的粒度过小, 会导致任务规划的复杂化, 以及任务分配时通讯的巨大开销; 而任务分解的粒度过大, 则会降低任务分配的成功率。因此, 在选择任务分解方法或设计任务分解算法时应充分注意这一问题。

2.2 协调规划层(CdPL)

如图 3 所示, 协调规划层是为解决机器人之间协作关系确定后具体的运动控制问题。协作层确定了各机器人的子任务后, 机器人根据当前目标(任务)、自身状态、以往经验、传感信息等来规划自身行为。另外, 在动作执行过程中负责检查冲突和采取有效措施消解冲突。协调层中无法处理的问题(如: 子任务的实现、冲突的消解) 则返回协作层, 由协作规划器重新规划。

CdPL 各个模块的功能如下:

1) 协调层通讯模块(CMcdL): 负责机器人之间运动规划信息的传递;

2) 建模模块(MM): 根据传感信息和通讯信息对外部环境建模, 目的是对其他机器人的运动状态做出预测, 以作为规划自己动作的参考。其中建模模

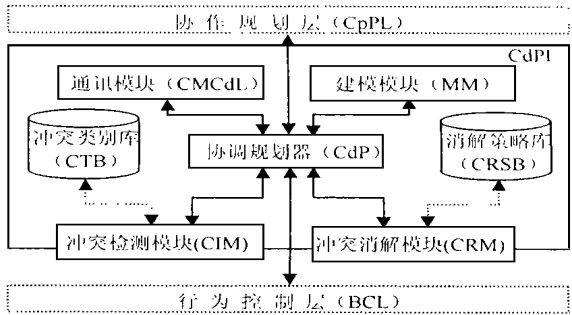


图 3 协调规划层功能结构

型包括世界模型和其他机器人的状态模型;

3) 冲突检测模块(CIM): 随着时间的推移及环境的变化, 原来的计划在执行过程中可能会产生冲突, 该模块则是用来检测冲突的发生, 判断冲突发生的类型以及推测冲突发生的原因等;

4) 冲突类别库(CTB): 存放各种冲突类型, 如目标冲突、资源冲突等;

5) 冲突消解模块(CRM): 根据冲突类型提供冲突消解的策略;

6) 消解策略库(CRSB): 存放冲突消解策略;

7) 协调规划器(CdP): 是协调层的核心, 负责机器人运动协调的局部规划, 根据任务(目标)、自身状态及外部环境信息, 规划自身的行为, 产生近期的运动序列, 并负责机器人协调规划层之间的协商, 实现机器人之间冲突的消解。

协调规划算法如下:

Step1: 规划阶段: 对于 CpPL 给定的任务采取相应的规划方法(如 C 空间法、人工势场法等), 考虑传感信息及来自其他机器人的信息, 进行运动规划, 将运动规划序列送行为控制层;

Step2: 信息发布: 规划完成后, 向其他机器人发布运动规划信息, 表示为

$$\text{Plan-information} = (\text{Goal}, \text{Resource})$$

其中, Goal 为规划的运动目标, Resource 为所需资源。资源信息表示为

$$\text{Resource}(R_i, T_i) = \{\text{resource}_{i_1}, \dots, \text{resource}_{i_n}\}$$

其中

$$\text{resource}_{i_j} = (r\text{-name}_{i_j}, r\text{-time}_{i_j}, r\text{-type}_{i_j})$$

$r\text{-name}$ 表示资源名称, $r\text{-time}$ 表示占用资源时间, $r\text{-type}$ 表示占用资源方式(独占、共享);

Step3: 冲突检测: 根据传感信息和其他机器人发布的规划信息, 判定冲突发生的可能性以及冲突的类型;

Step4: 冲突消解: 提出冲突消解策略, 基于协商的方式(协商不仅能实现任务的动态分配, 也可以用于冲突消解) 制定冲突消解计划, 若冲突无法解决, 则转 Step1, 若任务无法实现, 则将信息反馈到协作规划层;

Step5: 执行冲突消解计划, 规划运动序列, 产生行为控制层运动控制命令, 若冲突没有解决, 则转 Step4;

Step6: 协调规划结束。

显然, 协调规划层保障了机器人的在线运动规划以及冲突的实时解决, 这也是设计协调规划层的目的所在。但冲突消解的效率在很大程度上依赖于冲突类别和冲突策略库中知识的完备性。

2.3 行为控制层(BCL)

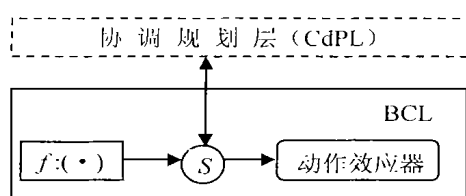


图4 行为控制层功能结构

行为控制层的设计有两个目的: 一是为了使机器人对紧急情况或简单任务做出迅速反应(如避障), 它直接由传感信息映射到某种行为, 基本上不做推理或根据简单规则直接推理; 二是执行协调规划层产生的运动控制命令, 产生相应的动作。

通过反应产生的运动具有最高优先级, 动作模块应立即执行, 并将从协调规划层送来的动作中断。如果发生中断, 协调规划模块将决定是否重新进行规划。图4中, $f:(\cdot)$ 表示从传感信息到运动控制的映射关系, 可以由神经网络、分类器及规则推理等实现^[14]。动作效应器是根据 $f:(\cdot)$ 或协调规划器的结果进行相应处理(例如坐标变换、运动学计算等), 实现动作执行。

3 结 语

本文针对机器人协作系统中协作与协调之间的关系, 设计一种分层机器人控制结构。它不仅能实现机器人协作系统中机器人所必须具备的内部管理、轨迹规划及控制功能, 而且具备信息交互、分析推理、任务分配以及组织管理等功能, 满足机器人个体的自主性和协作性要求, 以及 MRCS 系统的开放性、动态组织等特点。由于该控制体系采用分层、模块式结构, 因此有利于系统的设计、维护及功能的扩

充。

参 考 文 献

- 1 Cao Y, Fukunaga A, Kahng A. Cooperative mobile robotics: Antecedents and directions. *Autonomous Robotics*, 1997, 4: 7- 27
- 2 Agah A. Robot teams, human workshops and animal sociobiology: A review of research on natural and artificial multi-agent autonomous systems. *Advanced Robotics*, 1996, 10(6): 523- 545
- 3 王越超, 谈大龙. 协作机器人学的研究现状与发展. *机器人*, 1998, 20(1): 69- 75
- 4 Fukuda T, Nakagawa S A. Dynamically reconfigurable robotic system. In: *Int Conf on Ind Electronics, Control and Instrumentation*. Cambridge, 1987. 588- 595
- 5 Fukuda T, Iritani G. Construction mechanism of group behavior with cooperation. In: *IEEE/RSJ IROS*. Pittsburgh, 1995. 535- 542
- 6 Jin K, Liang P, Beni G. Stability of synchronized distributed control of discrete swarm structures. In: *IEEE ICRA*. San Diego, 1994. 1033- 1038
- 7 Asama H, Matsumoto A, Ishida Y. Design of an autonomous and distributed robot system: ACTRESS. In: *IEEE/RSJ IROS*. Tsukuba, 1989. 283- 290
- 8 Caloud P, Choi W, Latombe J *et al.* Indoor automation with many mobile robots. In: *IEEE/RSJ IROS*. Tsuchiura, 1990. 67- 72
- 9 Parker L E. ALLIANCE: An architecture for fault tolerant cooperative control of heterogeneous mobile robots. In: *IEEE/RSJ IROS*. Munich, 1994. 776- 783
- 10 Cem Ünsal. Self-organization in large population of mobile robots. M S Thesis, 1993
- 11 陈仁际. 基于 Agent 的分布协作式多机器人系统协作控制研究及实现. 哈尔滨工业大学博士学位论文, 1999
- 12 谭民. 机器人团体协作与控制的研究. 见: 863 计划自动化领域发展战略研讨会论文集. 1999. 185- 190
- 13 Jorg P Muller. The design of intelligent agents: A layered approach. Springer, 1996. 45- 123
- 14 史忠植. 高级人工智能. 北京: 科学出版社, 1997. 212- 248

作 者 简 介

范永男, 1971年生。中国科学院自动化研究所博士生。研究方向为机器人团体协调与控制, 机器人控制器开发等。

谭民男, 1962年生。中国科学院自动化研究所研究员, 博士生导师。研究领域为机器人可靠性设计, 机器人团体协调与控制, 机器人控制器开发等。