

文章编号 : 1002-0446(2004)01-0058-05

一种多机器人协作控制方法*

孟 伟, 洪炳镛, 韩学东

(哈尔滨工业大学计算机科学与技术学院, 黑龙江 哈尔滨 150001)

摘 要: 提出了一种基于 Petri 网模型的多个移动机器人协作控制方法. 该方法主要包括两个模块: 高层控制模块(HLCM)和低层控制模块(LLCM). HLCM 负责任务分配和再规划, LLCM 实现单个机器人的控制逻辑. 利用可达树对 Petri 网模型中的死锁进行检测, 并给出了消除死锁的方法. 仿真实验证明了该方法的有效性.

关键词: 多机器人系统; 协作; Petri 网模型; 死锁

中图分类号: TP24 **文献标识码:** B

A COORDINATION CONTROL METHOD FOR MULTI-ROBOT

MENG Wei, HONG Bing-rong, HAN Xue-dong

(School of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001, China)

Abstract: A control method for multi-robot coordination is presented, and it consists of two levels. The first level is High Level Control Module (HLCM), in which task assignment and replanning is performed. The second level is Low Level Control Module (LLCM), which completes the logical control of each robot. Reachability tree is used to check deadlock in the Petri net model and the method to eliminate deadlock is also presented. Experiment results prove the effectiveness of this method.

Keywords: multi-robot system; coordination; Petri net model; deadlock

1 引言 (Introduction)

随着机器人应用领域和范围的不断扩展, 单机器人在某些方面已经很难达到人们期望的要求, 多机器人协调与合作作为一种新的机器人应用形式日益引起国内外学术界的兴趣与关注. 多机器人通过协调合作可以完成单机器人难以完成的复杂任务, 而且可以提高工作效率, 即使当机器人工作环境发生变化或系统局部发生故障时, 多机器人之间仍可通过本身具有的协调与合作关系完成预定的任务. 从研究的角度看, 多机器人协调研究比单机器人增加了许多新问题, 主要有^[1-3]: (1) 复杂协调任务的描述; (2) 同一工作空间中多机器人协调和集中; (3) 多机器人协调系统的自适应控制; (4) 多机器人协调系统的负载分配; (5) 以多传感器为基础的数据检测和障碍描述; (6) 多机器人协调系统的建模和控制结构的模式; (7) 多机器人坐标间的标定; (8) AI 技术和控制系统与多机器人协调系统的结合. 为完成同一

任务, 多个机器人协作应具有鲁棒性、灵活性、低代价和高性能的优点.

这些系统可用于材料处理^[6]、扫描工作^[4]、卫星探测^[7]和微空间的工作^[5]. 然而, 由于随着机器人数量的增加, 机器人之间的交互的复杂度呈指数增长, 因而很难分析单个机器人的作用, 也很难设计机器人的协作控制算法.

本文提了一种多机器人协调控制的方法, 通过任务分解和协作使得多个机器人完成同一任务. 该方法基于 Petri 网^[8], 受局部传感器信息驱动, 主要包含两个模块: 高层控制模块和低层控制模块. 高层控制模块完成多个移动机器人的全局控制规划, 即机器人如何协作, 包括任务分配和任务更改规划; 低层控制模块完成单个机器人的局部控制规划, 包括作业规划和死锁检测及消除. 仿真实验证明了该方法的有效性.

* 基金项目: 国家自然科学基金资助项目(69985002); 国家 863 计划资助项目(2001 AA422270).
收稿日期: 2003-07-01

2 多机器人协作控制模型 (Multi-robot coordination control model)

多机器人的协作控制模型如图 1 所示. HLCM 负责任务的分配与再调整, LLCM 负责每个机器人的任务执行, 它们之间通过通讯模块进行交互.

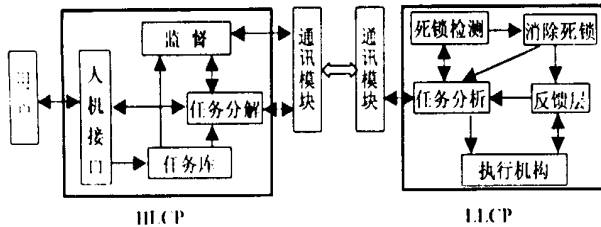


图 1 多机器人协调控制模型

Fig.1 Multi-robot coordination control model

2.1 高层控制模块

为了完成同一目标, 需要多个机器人协调合作, 高层控制模块主要完成任务分配和突发情况下机器人的任务再规划. 高层控制模块不仅考虑总体任务的完成, 还要考虑每个机器人的状态. 在该模块中利用 Petri 网建立一个由多个子任务构成的总任务描述模型. HLCM 的设计过程可以描述如下:

Step1: 将总任务分解为 n 个子任务, 即 $W_{obj} = \sum_{i=1}^n W_{subi}$.

Step2: 为子任务 W_{subi} 分配库所 $p_i, i=1, 2, \dots, n$.

Step3: 根据子任务的时序和优先级, 决定各个库所间的优先权关系, $\beta_1 > \beta_2 > \dots > \beta_n$.

Step4: 利用连接算子 \otimes , 将库所连结成代数式 Z 进行描述.

Step5: 利用代数式 Z 与 Petri 网间的映射关系, 建立特定任务目标的高层控制模块的 Petri 网模型.

Step6: 确定初始状态 u_0 . 将每个机器人作为一个 token, 机器人的标号赋给 token, token 放入到 Petri 网模型初始库所中.

Step7: 为了控制子任务的更改或完成, 可以增加突发库所 p_k . 如果需要改变或完成子任务, 分配一个 token 给库所 p_k , 即可完成某个对应的子任务或者跳到相应的处理库所, 控制命令传给相关的机器人.

Step8: 根据子任务的序列给库所和变迁加上标号.

通过上述步骤建立高层控制模块的模型, 其中, 如果与子任务对应的库所中有 token, 则子任务分配给带有相应标号的机器人, 通过带标号的 token 和它所在的库所来监视机器人的状态.

2.2 低层控制模块

低层控制模块负责完成单个机器人的控制规划, 包括作业规划和死锁的检测及消除. 机器人进行规划时除了要考虑高层控制模块分配的任务, 还需要考虑与其它机器人的交互以及子任务的特点.

为单个机器人设计控制逻辑的主要过程描述如下:

Step1: 如果传感器是打开的, 相应的库所中就生成一个 token.

Step2: 利用 Petri 网的 AND/OR 逻辑生成合成库所.

Step3: 根据机器人的分类状态决定左轮和右轮的速度. 每个命令对应着一个库所 (p_i) 和一个变迁 (T_i).

Step4: 根据机器人分类状态的重要程度来决定时间的优先级.

Step5: 根据事件的重要程度给每个变迁分配一个数字 $T_m (m=1, 2, \dots)$.

Step6: 将只有一个 token 的库所与所有的命令变迁相连, 防止一次选择两个以上命令.

在给变迁分配优先级时, 级别越高分配的数字 T_m 越小. 越先被触发的变迁分配的数字越小. 因此, 即使发生冲突, 通常也能产生一个唯一的控制命令.

该过程根据子任务的数量从 Step1 到 Step6 反复执行, 构成了低层控制模块.

3 死锁检测及消除 (Checking and eliminating of deadlock)

机器人在执行实际任务过程中, 外界环境因素和机器人本身因素都可能导致机器人发生错误, 陷入死锁状态. 死锁可以理解为机器人在一点反复执行同一动作, 如果不对机器人的某些参数进行调整, 就可能导致子任务无法完成. 为了便于对机器人的状态进行监控, 采用了基于可达树的死锁检测及消除的方法.

3.1 死锁检测

可达树是 Petri 网分析的方法之一, 表示 Petri 网的可达标识集, 对 Petri 网的性能分析十分重要. 每个 Petri 网对应一棵可达树, 但由于 Petri 网的可达集可能是无限的, 因而可能会产生无限的可达树. 为此, 可以利用下面三类节点来限制可达树的规模: 端点、重复节点、覆盖节点.

设 u 为可达树的根, 即待处理节点, 构造可达树的算法可以描述如下:

Step1: 设 x 为待处理的边界点, 其标识为 u_x .

Step2:若在可达树中已经存在另一个节点 y ,且 $u_x = u_y$,则 x 为重复节点.

Step3:若在节点 x 的标识 u_x 下,没有任何一个变迁是使能的,即对 $\forall t \in T, \delta(u_x, t)$ 无定义,则 x 为一个端点.

Step4:如果在标识 u_x 下, $\exists t \in T$,使得 $\delta(u_x, t)$ 有定义,则产生一个新的节点 z ,其标识 u 可以通过以下步骤来确定:

(1) 如果 $u_z(p) = \omega$,其中 $p \in P, u_z(p) = \omega$.

(2) 如果从初始标识 u_0 到 u_x 的有向路径上存在节点 y ,其标识 u_y 满足 $u_y < \delta(u_x, t)$,并且 $\exists p \in P, u_z(p) < \delta(u_x, t)(p)$,则 $u_z(p) = \omega$.

(3) 否则, $u_z(p) < \delta(u_x, t)(p), p \in P$.

Step5:把节点 z 作为节点 x 的叶节点,并设为边界点,节点 x 设为内部节点.

Step6:如果所有的节点都已归入端点、重复节点或内部节点,则过程结束;否则,转向 Step1;

在 Petri 网 $C = \langle P, T, I, O, u_0 \rangle$ 中,若存在可达标识 u 和依次使能的变迁序列 $\sigma \in T^*$,则状态函数定义为: $\delta(u, \sigma) = u'$,其中 u 可按如下方式递推: $\delta(u, t\lambda) = \delta(\delta(u, t), \lambda), \lambda \in T^*$. 设 x 为 Petri 网可达树中的叶节点(包含端点和重复节点),它所关联的标识为 u_x ,则基于可达树的死锁检测方法可以分两个情况进行讨论:

情况1:如果 x 是端点,则 Petri 网有死锁,死锁标识为 u_x ,导致死锁的运行过程为 σ , σ 是根节点到 x 的有向路径.

情况2:如果 x 为重复节点,则:

(1) 消除中 u_x 对死锁特性无影响的 ω 分量;

(2) 如果 u_x 中不含 ω 分量,则可按如下方法验证 u 是否为死锁标识: $\forall t \in T$,检查 $\delta(u_x, t)$ 是否有定义,若对于 $\forall t \in T, \delta(u_x, t)$ 都没有定义,则 Petri 网有死锁, u_x 为死锁标识.导致死锁的运行过程是根节点到 x 的有向路径.

(3) 如果 u_x 中有 ω 分量,则按如下步骤进一步分析:

• 计算 $u', u'', \Delta u$;

• 如果 $\forall p \in P$,都使得 $\Delta u(p) \geq 0$,则广义标识 u_x 中不含有可达的死锁标识,也就是说 Petri 网无死锁;

• 如果 $\exists p \in P$,满足 $\Delta u(p) < 0$,记 u_d' 为死锁标识,则 Petri 网有死锁,否则 Petri 网无死锁.

通过上述方法可以检测出 Petri 网中是否存在死锁现象,可以避免机器人在实际执行中发生死锁.

3.2 消除死锁的方法

由于死锁产生的原因通常是逻辑冲突或事件优先级不正确,动态的改变控制命令可以校正该错误.

校正死锁错误的过程如下所示:

Step1:在控制模型中的库所集 P 中寻找只有一个 token 的库所 p_i .

Step2:如果记忆的库所只有一个,转 Step5;否则,转 Step3.

Step3:检查命令变迁,即检查与库所相连的命令,其中不包括优先级最高的库所.关于命令,利用表1中所给的权值根据(1)式计算死锁系数 β .

$$\beta = \frac{(\text{下个运动状态})^2}{\text{当前运动状态}} + (\text{与前一状态方向变化})^2 \quad (1)$$

Step4:将具有最高优先权的命令变迁 T_j 与具有最大 β 的命令变迁 T_k 相交换.如果新的事件优先级与前面的相同,从记忆的合成库所中删掉具有最大 β 的命令变迁的输入库所.重复 Step2 到 Step4.

Step5:随机改变事件优先权,直到优先权与先前不同.

Step6:根据新的事件优先权修改控制模型,通过触发已经激活的变迁,选择控制命令.

Step7:执行所选择的控制命令.一旦机器人移动,返回修改前的控制模型.

该过程解决了死锁问题,这样就保证了机器人如果出现死锁现象,也可以通过动态方式进行解决,使得机器人的行为更为流畅.如表1所示.

表1 所选下个命令的权值

运动状态	向前	避障	停止并寻找目标
权值	1	3	5
与前一状态方向变化	停止	相同	相反
权值	1	2	5

4 实验及分析 (Experiment and analysis)

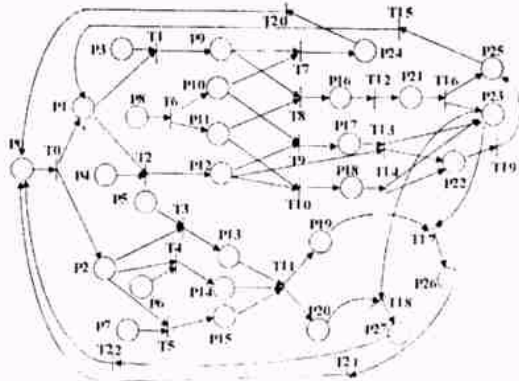
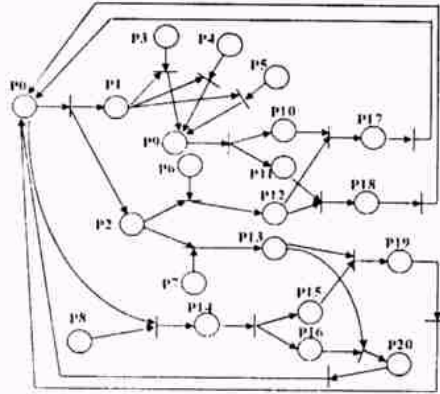
实验中,两个机器人跟踪同一运动目标.任务目标包括两个子任务,即跟踪运动目标任务和跟踪其它机器人任务.为了有效地搜索对象,当机器人无法跟踪运动目标时,可以通过其它机器人的信息来判断目标是否丢失,若未丢失,可以通过跟踪其它机器人的方式跟踪目标.移动机器人不能准确地知道它目前的位置,和另一个机器人的位置.速度和移动方向,因此机器人必须通过局部传感器信息来推断另

一个机器人的状态和运动目标的状态.为了解决逻辑冲突,事件的优先权由高到低依次为:有障碍、避障、目标在某个方向、前进/左转/右转、寻找目标、传感器无数据.事件优先权的次序越小,变迁的标号就越小.

两个机器人跟踪同一运动目标的高层控制模块模型如图 2(a)所示,低层控制模块模型如 2(b)所示.

示.

当两个机器人被障碍物阻挡,无法发现运动目标时,根据控制模型,机器人会在原地进行运动目标的搜索,但由于障碍的遮挡,它们总是无法看到目标,最终机器人无法前进,陷入死锁状态.利用上面提出的消除死锁的方法,就可以解决该问题.表 2 给出了消除死锁时动态调整优先权的过程.



(a)

(b)

图 2 双机器人跟踪目标任务的高层控制模块和低层控制模块

Fig.2 HLCM and LLCM of tracking moving objects with two robots

表 2 事件优先权的调整

Table 2 Adjustment of priority of events

事件优先权	1	2	3	4
初始状态	7	20	11	18
第一次	7	11	20	18
第二次	11	7	18	20
第三次	11	18	7	20

实验结果如图 3 所示.其中,实线为 1 号机器人的行走路线,虚线为 2 号机器人的行走路线,出发时 1 号在 2 号的前面.运动目标的行走路线为起点 \rightarrow A \rightarrow B \rightarrow C \rightarrow D \rightarrow 终点.

图 4(a)和图 4(b)分别给出了 1 号和 2 号机器人与运动目标的轨迹差值图,可以看出整体跟踪效果较好,只是在运动目标绕过障碍物时,机器人丢失目标,导致轨迹差别较大.在这两处由于后面的机器人首先找到了运动目标,因而两个机器人的跟踪顺序发生了变化.图 4(c)给出了两个机器人的轨迹差值图,可以看出,从丢失目标到重新跟踪的过程中两个机器人的距离差值最大.

5 结论 (Conclusion)

本文提出了一种利用局部感知信息的多移动机器人协作控制方法,给出了多机器人的协调控制模型以及死锁检测及消除的方法.该方法利用 Petri 网将控制模型可视化,清楚地表示了命令冲突和机器人的状态,通过改变控制命令模式和使用命令变迁标号,动态地生成控制命令,很容易检查控制命令的重复或删除.另外,与命令变迁关联的数字很容易表示事件的优先权.通过控制命令模式和变迁标号的再分配,可以动态地改变控制命令,从而有助于开发通用的错误校正方法.

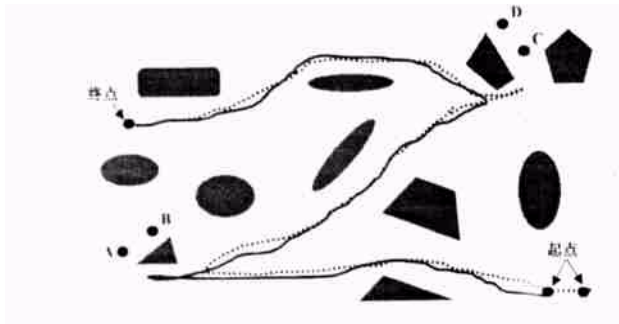


图 3 实验结果

Fig.3 Experiment result

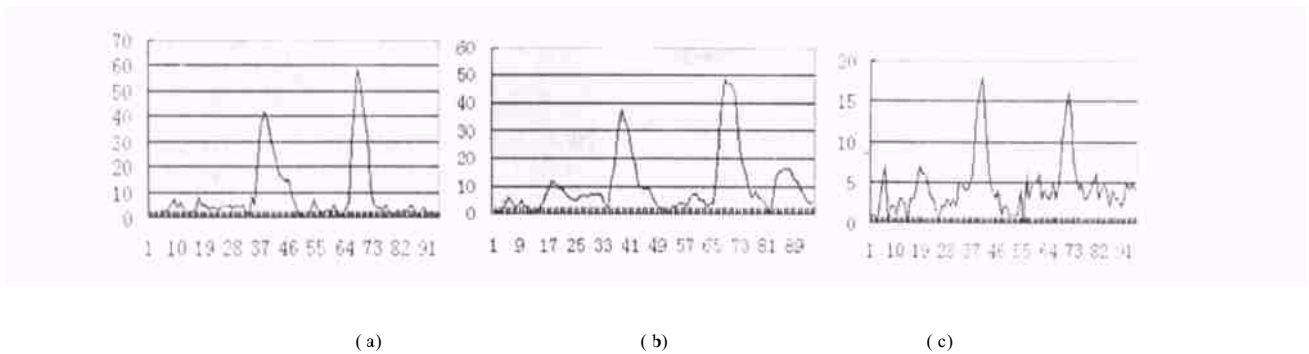


图 4 轨迹误差

Fig.4 Trajectory error

参考文献 (References)

- [1] Yamauchi B. Decentralized coordination for multirobot exploration[J]. *Robotics and Autonomous Systems*, 1999, **29**(1) : 111 - 118 .
- [2] Koivo A J, Bekey G A. Report of workshop on coordinated multiple robot manipulators : planning, control, and application[J]. *Robotics and Automation*, 1998, **14**(1) : 91 - 93 .
- [3] Kazerooni H. Compliance control and unstructured modeling of cooperation robots[A]. *Proceedings of 1988 IEEE International Conference on Robotics and Automation*[C]. Philadelphia : 1998, vol .1 . 510 - 515 .
- [4] Arai T, Kurabayashi D, Ota J, *et al.* Motion planning for cooperative sweeping with relocating obstacles[A]. *Proceedings of IEEE International Conference on Systems, Man and Cybernetics* [C]. Beijing : 1996, vol .2 : 1513 - 1518 .
- [5] Dario P, Carrozza M C, Lencioni L, *et al.* A micro robotic system for colonoscopy[A]. *Proceedings of IEEE International Symposium on Computational Intelligence in Robotics and Automation*[C]. 1997, vol .2 : 1567 - 1572 .
- [6] Kube C R, Zhang H. The use of perceptual cues in multi-robot box-pushing[A]. *Proceedings of IEEE International Conference on Robotics and Automation* [C]. Minneapolis : 1996, vol .3 : 2085 - 2090 .
- [7] Miller D P. Multiple behavior-controlled micro-robots for planetary surface missions[A]. *Proceedings of IEEE International Conference on Systems, Man and Cybernetics*[C]. 1990. 289 - 292 .
- [8] 袁崇义. Petri 网原理[M]. 北京 : 电子工业出版社, 1998 .

作者简介:

- 孟 伟 (1974-), 女, 博士生. 研究领域: 智能机器人, 智能控制.
- 洪炳熔 (1937-), 男, 教授, 博士生导师. 研究领域: 空间机器人, 虚拟现实, 机器人足球.
- 韩学东 (1973-), 男, 博士生. 研究领域: 多智能体系统, 智能控制, 机器人足球.

(上接第 6 页)

参考文献 (References)

- [1] 西上雅郎, 水岛敏文. Glass roof cleaning robot system ' Canadian Crab' [J]. *Journal of the Robotics Society of Japan*, 1992, **10**(5) : 40 - 42 .
- [2] Briones L, Bristamante P, Serna M A. Robicen : a wall-climbing pneumatic robot for inspection in nuclear power plants[J]. *Robotics & Computer Integrated Manufacturing*, 1994, **11**(4) : 287 - 292 .
- [3] Luk B L, Collie A A, Billingsley J. Robug II : an intelligent wall climbing robot [A]. *Proceedings 1991 IEEE International Conference on Robotics and Automation*[C]. 1991, 3 . 2342 - 2347 .
- [4] Nishi A. A biped walking robot capable of moving on a vertical wall[J]. *Mechatronics*, 1992, **2**(5) : 543 - 554 .
- [5] 刘淑霞, 王炎. 爬壁机器人技术的应用[J]. *机器人*, 1999, **21**(2) : 193 - 199 .
- [6] 王巍, 宗光华. 一种新型机器人的机构稳定性分析[J]. *机器人*, 1999, **21**(7) : 642 - 648 .

作者简介:

- 张厚祥 (1974-), 男, 博士生. 研究领域: 自主移动机器人技术, 气动控制技术.