



(12) 发明专利申请

(10) 申请公布号 CN 102814813 A

(43) 申请公布日 2012. 12. 12

(21) 申请号 201210203241. 7

(51) Int. Cl.

(22) 申请日 2012. 05. 04

B25J 9/16 (2006. 01)

(30) 优先权数据

61/482, 808 2011. 05. 05 US

(71) 申请人 范努克机器人技术美国有限公司

地址 美国密执安州

(72) 发明人 H·D·麦吉 张天禄 P·斯旺森

陶建明 萧笛 黄浩祥 郑少佳

蔡启庚

(74) 专利代理机构 中国专利代理(香港)有限公

司 72001

代理人 马永利 卢江

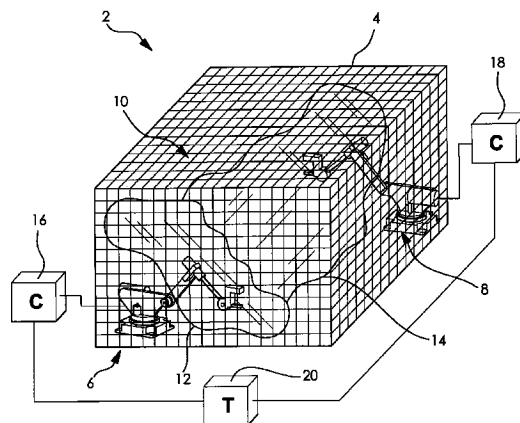
权利要求书 2 页 说明书 11 页 附图 16 页

(54) 发明名称

在多机器人系统中自动防止死锁的方法和系统

(57) 摘要

本发明涉及在多机器人系统中自动防止死锁的方法和系统。一种用于在包含多个机器人的工作间中控制避免碰撞和死锁的系统和方法自动地确定潜在的死锁状况,并识别出避免这些状况的方式。通过在执行具有潜在死锁状况的运动之前确定无死锁运动语句来消除死锁状况。对无死锁运动语句的确定可以离线、在正常执行之外进行,或是它可以在正常产生执行期间进行。如果有充足的 CPU 处理时间可用,那么在正常产生执行期间的确定提供了响应于动态状况的最大灵活度,例如 I/O 时序或外部事件或序列的时序上的变化。为使得 CPU 影响最小,该确定可以离线地进行,其中能够分析编程序列的多个排列并找到最优执行序列。



1. 一种用于防止具有公用工作空间的机器人对死锁的方法,所述机器人中的每一个都被相关程序控制,其中当程序被并发执行时,在相关程序的执行的一部分期间,所述机器人中的每一个占用公用工作空间的至少一部分,所述方法包括以下步骤:

在程序的并发执行期间,识别出由机器人占用的公用工作空间的部分;

识别出在其中所述公用工作空间的部分重叠的至少一个干扰区域;

分析该至少一个干扰区域并识别出会在何处发生两个机器人的至少一个死锁状况;以及

通过在执行导致至少一个死锁状况的机器人的任何运动之前自动地确定和执行至少一个无死锁运动语句,在程序的执行期间避免至少一个死锁状况。

2. 根据权利要求 1 的方法,包括在程序的正常产生执行期间施行所述确定和执行。

3. 根据权利要求 1 的方法,其中所述至少一个无死锁运动语句被添加到所述程序的至少一个中。

4. 根据权利要求 1 的方法,其中通过修改所述程序的至少一个中的至少一条指令来提供所述至少一个无死锁运动语句。

5. 根据权利要求 1 的方法,其中通过对所述程序的至少一个中的指令重新排序来提供所述至少一个无死锁运动语句。

6. 根据权利要求 1 的方法,其中通过控制独立于至少一个程序中的程序指令的各程序中的至少一个的执行来提供所述至少一个无死锁运动语句。

7. 根据权利要求 1 的方法,包括用体素模型来表示公用工作空间的被占用部分,并且比较所述各体素模型以识别出至少一个干扰区域。

8. 一种计算机可读介质,其包含能够由计算机执行以施行防止具有公用工作空间的机器人对死锁的方法的指令,所述机器人中的每一个都由相关程序控制,其中当程序并发执行时,在相关程序的执行的一部分期间,所述机器人中的每一个占用公用工作空间的至少一部分,所述方法包括:

在程序的并发执行期间,识别出由机器人占用的公用工作空间的部分的指令;

识别出在其中所述公用工作空间的部分重叠的至少一个干扰区域的指令;

分析所述至少一个干扰区域并识别出会在何处发生两个机器人的至少一个死锁状况的指令;以及

通过在执行导致所述至少一个死锁状况的机器人的任何运动之前自动地确定和执行至少一个无死锁运动语句,在程序的执行期间避免所述至少一个死锁状况的指令。

9. 根据权利要求 8 所述的方法,包括在程序的正常产生执行期间施行所述确定和执行的指令。

10. 根据权利要求 8 所述的方法,其中所述至少一个无死锁运动语句被添加到所述程序的至少一个中。

11. 根据权利要求 8 的方法,其中通过修改所述程序的至少一个中的至少一条指令来提供所述至少一个无死锁运动语句。

12. 根据权利要求 8 的方法,其中通过对所述程序的至少一个中的指令重新排序来提供所述至少一个无死锁运动语句。

13. 根据权利要求 8 的方法,其中通过控制独立于至少一个程序中的程序指令的各程

序的至少一个的执行来提供所述至少一个无死锁运动语句。

14. 根据权利要求 8 的方法,包括用体素模型来表示公用工作空间的被占用部分以及比较各体素模型以识别出至少一个干扰区域的指令。

15. 一种防止具有公用工作空间的多个机器人死锁的方法,所述机器人中的每一个都被相关程序控制,其中当所述程序并发执行时,在相关程序的执行的一部分期间,所述机器人的每一个都占用公用工作空间的至少一部分,所述方法包括以下步骤:

在程序的并发执行期间,识别出由机器人占用的公用工作空间的部分;

为至少两个机器人识别出在其中所述公用工作空间的部分重叠的至少一个干扰区域;

分析所述至少一个干扰区域并识别出会在何处发生所述至少两个机器人的至少一个死锁状况;以及

通过执行导致所述至少一个死锁状况的所述至少两个机器人的任何运动之前自动地确定和执行至少一个无死锁运动语句,在程序的执行期间避免所述至少一个死锁状况。

在多机器人系统中自动防止死锁的方法和系统

[0001] 相关申请的交叉引用

[0002] 本申请为 2008 年 5 月 21 日提交的序列号为 12/124,430 的共同未决美国专利申请的部分延续。

[0003] 本申请要求 2011 年 5 月 5 日提交的序列号为 61/482,808 的美国临时专利申请的权益。

技术领域

[0004] 本发明通常涉及用于控制多个机器人的系统,以及用于控制针对多个机器人的运动干扰避免的方法。

背景技术

[0005] 物体在空间中的移动为典型制造环境中的必要任务。越来越多地使用机器人来实现必要的移动。但是,当多个物体正在移动时,在这些物体之间会存在潜在的干扰。如果至少两个物体同时共享同一空间,例如当这些物体关于公用参照系具有相同坐标时,就会存在干扰。

[0006] 随着现代工业机器人以相当高的速率移动,机器人之间的干扰可导致碰撞,并对机器人和由机器人操作的工作件产生不希望的损坏。碰撞可导致制造过程中的昂贵的停机时间。因此,期望避免这样的碰撞。

[0007] 现有技术的系统和方法已被用于试图最小化干扰和碰撞。但是,现有技术的系统和方法存在一些不足。典型地,仅仅关于预定的干扰区域或“静态空间”对工具中心点(TCP)进行检查。针对多个机器人,难以直接或有效地防止其干扰或碰撞。并且,难以关于多个移动机器人的静态坐标系来指定干扰空间。任何干扰空间都不仅是机器人运动路径的函数,而且还是运动速度的函数。在试图处理当两个或更多个机器人请求同时移动到公用空间时的死锁情况时,也存在困难。

[0008] 现有技术系统还试图防止机器人的 TCP 在关于其世界坐标系的固定空间中发生碰撞。当(具有多个控制器的)多个机器人在任务执行期间共享公用或“干扰”空间时,每个控制器必须等待直到没有机器人在公用空间中为止。然后控制器可以接下来发出运动控制指令,以允许机器人移动。该过程也被称为“等待和移动”过程,其通常会增加工作循环时间。但是,难以有效地根据固定坐标系指定干扰空间,因为干扰空间不仅是机器人运动路径的函数而且还是运动速度的函数。当多于一个机器人请求同时移动到公用空间时,它会产生死锁情况,在这种情况下没有一个机器人可以移动,因为它们都在彼此等待。

[0009] 现有技术系统还试图用球体和圆柱体来对机器人建模。这些系统在移动过程中实时地预测机器人的未来位置。因为这些系统并不确定机器人在一段时间所占用的累计空间,因此这些系统必须在机器人运动期间经常性地执行比较。这些系统对工作间内的所有机器人的模型进行逐个元件地比较。这种比较在计算上非常昂贵,并且随着机器人以及用于建模机器人和加工的元件的数量增加花费将以指数方式增长。因为比较是实时进行的,

所以当检测到即将发生碰撞时,这些系统通常必须停止涉及即将发生碰撞的所有机器人,且自动编程操作必须终止。当机器人属于不同的控制器时,这些系统变得更加困难,因为它们需要在控制器之间进行实时传送的大量的信息。现有技术系统还试图利用 I/O 握手机制以避免干扰。在本发明中,无需使用 I/O PLC。

[0010] 一个已知系统和方法公开在受让人的共同未决国际申请号 PCT/US2007/066638 中,通过引用将其整体结合在此。该系统和方法包括“动态空间检查”系统,其中对机器人操作的效率最大化,并使得多个机器人的干扰或碰撞的可能性被最小化。由每个控制器控制的机器人仅在用户定义的动态空间中工作,由此避免碰撞。但是,动态空间检查系统通常仅保护 TCP 不进入用户定义的用直线围着的空间。

[0011] 另一用于避免机器人碰撞的已知方法报告在 Pollack 等人的美国专利号 5,150,452 中。该方法包括创建包含期望机器人移动的碰撞地图。通过以逐个字节地方式将机器人地图和世界地图以逻辑异或操作结合起来,以及其后将碰撞地图和世界地图以逻辑异或操作结合,后面是将碰撞地图和世界地图以逻辑或操作结合起来,将期望机器人的初始位置从“世界”地图移除。碰撞由或和异或结合的任何比特位置上的差异来指示。该方法提供了二维 x-y 投影和一维高度上的碰撞检测,但是不允许三维、实时的碰撞检测。

[0012] 用于在机器人和一个或多个障碍物之间的碰撞发生之前检测该碰撞的另一已知方法在 Greenspan 等人的美国专利号 5,347,459 中描述。该机器人由体素化工作空间中的球体建模。工作空间中的每个体素被指定一个值,该值对应于其与最近障碍物的距离。如果球体中心的体素值在体素上小于其它球体的半径,那么确定即将发生碰撞。但是该方法仅仅保护了单个机器人臂。该机器人也仅由球体建模,因此致使对机器人的关键过程路径的保护不充分。

[0013] 存在对一种控制针对多个机器人的运动干扰避免的方法和系统的持续需要。期望该系统和方法提供三维和实时的碰撞检测、事先与机器人系统进行机器人运动通信、预定无碰撞的编程轨迹、并保护关键过程路径。

[0014] 当程序或任务处于下述这样一种状态时将发生死锁的情形:任何程序或任务的连续顺序执行都将导致在与该程序或任务有关的一个或多个机器人之间产生干扰并且没有程序或任务可以在没有干扰的情况下以顺序方式被处理。

[0015] Chaffee 等人的美国专利号 7,114,157 描述了一种通过以设定顺序获取资源的避免死锁方法。虽然该方法可以避免死锁,但是如果所设定的无死锁顺序还未被确定,则其无法提供允许操作的能力。另外,如果所请求的顺序在规定的无死锁顺序之外,则它不允许可能是无死锁的操作。

[0016] Chang 等人的美国专利申请公开号 2009/0326711 描述了一种自动或手动地利用自动区(zone)避免死锁的方法。Chang 提议使用优先级值来防止死锁的方法。Chang 不提议在不使用优先级值的情况下防止死锁的方法。

发明内容

[0017] 与该即时公开一致地,惊人地公开了一种控制针对多个机器人的运动干扰避免的系统和方法,其中该系统和方法在不使用 I/O 握手机制的情况下,提供了三维和实时的碰撞检测、事先与机器人系统的机器人运动通信、无碰撞编程轨迹的预定、以及关键过程路径

的保护。

[0018] 在一个实施例中,提供防止具有公用工作空间的机器人对死锁的方法,每个机器人由相关程序控制,其中当程序并发地执行时,每个机器人在该相关程序的执行的一部分期间,占用该公用工作空间的至少一部分,所述方法包括以下步骤:在程序并发执行期间,识别出被机器人所占用的公用空间的部分;识别出至少一个干扰区域,在该干扰区域中公用工作空间的部分重叠;分析所述至少一个干扰区域并识别出会在哪里出现两个机器人的至少一个死锁状况;以及通过在执行导致至少一个死锁状况的任何机器人运动之前自动地确定和执行至少一个无死锁运动语句,而在程序执行期间避免所述至少一个死锁状况。

[0019] 在进一步的实施例中,提供一种计算机可读介质,其包含可由计算机执行以施行防止具有公用工作空间的机器人对死锁的方法的指令,每个机器人由相关程序控制,其中,当程序被并发执行时,每个机器人在所述相关程序的执行的一部分期间占用公用工作空间的至少一部分,所述方法包括:在程序并发执行期间,识别出被机器人占用的公用空间的部分的指令;识别出至少一个干扰区域的指令,在该干扰区域中所述公用工作空间的部分重叠;分析至少一个干扰区域并识别出两个机器人的至少一个死锁状况会出现在哪里的指令;以及通过在导致至少一个死锁状况的任何机器人运动的执行之前自动地确定和执行至少一个无死锁运动语句,而在程序执行期间避免所述至少一个死锁状况的指令。

[0020] 在另一实施例中,一种防止具有公用工作空间的多个机器人死锁的方法,每个机器人由相关程序所控制,其中,当所述程序并发执行时,每个机器人在所述相关程序的的执行一部分期间占用公用工作空间的至少一部分,所述方法包括以下步骤:在程序并发执行期间,识别出被机器人所占用的公用空间的部分;识别出至少一个干扰区域,在该干扰区域中针对至少两个机器人的公用工作空间的部分重叠;分析至少一个干扰区域并识别出至少两个机器人的至少一个死锁状况会出现在哪里;以及通过在导致至少一个死锁状况的至少两个机器人的任何运动的执行之前自动地确定和执行至少一个无死锁运动语句,而在程序执行期间避免所述至少一个死锁状况。

[0021] 根据本发明的系统和方法无需事前知晓无死锁序列。该系统和方法将自动地确定潜在的死锁状况,并识别出避免这些状况的方法。该系统和方法不需要优先级值来防止死锁。该系统和方法防止了干扰状况,并同时防止了死锁状况存在的可能性。

附图说明

[0022] 当根据附图考虑时,从下面优选实施例的详细描绘中,以上所述的以及本发明的其它优点对于本领域技术人员来说将变得容易显而易见,其中:

[0023] 图 1 示意了根据本公开的示例性机器人系统,其具有在工作间中操作的第一机器人和第二机器人;

[0024] 图 2 是描绘根据本公开的干扰检查自动区方法的过程流程图;

[0025] 图 3 示意了在图 1 中描绘的第一机器人和第二机器人的其中一个的体素化模型;

[0026] 图 4 是在图 3 中描绘的体素化模型的等距视图;

[0027] 图 5 示意了在图 1 中描绘的第一机器人和第二机器人的其中一个的放大视图,其具有由重叠在第一机器人和第二机器人的其中一个之上的体素化球体和圆柱体形成的体素化模型;

- [0028] 图 6 示意了在图 5 中描绘的由多个体素形成的体素化球体和体素化圆柱体；
- [0029] 图 7 示意了根据本发明的无死锁矩阵和相关的程序列表；
- [0030] 图 8 示意了根据本发明的干扰矩阵和相关的程序列表；
- [0031] 图 9 示意了将多个死锁区域结合为单个无死锁区域的路径和相关表格；
- [0032] 图 10 示意了根据本发明的不具有和具有死锁预防的几个机器人程序执行序列；
- [0033] 图 11 是根据本发明的死锁避免序列的流程图；以及
- [0034] 图 12 是根据本发明用于确定潜在死锁区域的流程图。

具体实施方式

[0035] 以下描绘本质上仅为示例性，并不意图限制本公开、应用或使用。应当理解的是，贯穿附图，相应的参考数字同样指代相似的或相应的部件和特征。关于所公开的方法，所提出的步骤本质上仅为示例性，并且因此，并非必要或关键。

[0036] 图 1 是示意控制多个机器人之间的碰撞避免的机器人系统 2 的示意图。作为非限制性的示例，机器人系统 2 可以是具有至少两个机器人的车身修理厂机器人系统、喷水切割机器人系统、激光焊接机器人系统、电弧焊接机器人系统、以及涂漆机器人系统中的一个。也可根据需要，采用其他的具有多个机器人的机器人系统 2。

[0037] 机器人系统 2 包括工作间 4，其限定了第一机器人 6 和第二机器人 8 被配置为在其中进行操作的包封。第一机器人 6 和第二机器人 8 被配置为选择性地占用设置在工作间 4 中的至少一个公用空间 10。虽然只示例了第一机器人 6 和第二机器人 8，但应当认识到，机器人系统 2 可具有多于两个机器人，而不会背离本公开的范围和精神。

[0038] 第一机器人 6 可在第一机器人 6 沿着第一编程路径移动期间占用公用空间 10 的第一部分 12。第二机器人 8 可在第二机器人 8 沿着第二编程路径移动期间占用公用空间 10 的第二部分 14。第一和第二部分 12、14 同样被称为“自动区”或“自动区域”。每个编程路径包括一个或多个自动区，并且每个自动区被预处理并且从一个或多个运动语句中得到。例如，第一和第二编程路径可由在具有多个运动语句的指令序列中的一个或多个运动语句控制。技术人员应当理解的是，第一和第二部分 12、14 可能重叠，因此产生在不使用控制第一和第二机器人 6、8 运动的干扰避免方法的情况下发生机器人碰撞的可能性。

[0039] 第一和第二机器人 6、8 由至少一个控制器 16、18 控制。在图 1 所示的实施例中，第一机器人 6 被第一控制器 16 控制，并且第二机器人 8 被第二控制器 18 控制。该至少一个控制器 16、18 适用于与电功率的源（未示出）进行电通信。控制器 16、18 可以执行驻留在控制器 16、18 内的指令序列，例如计算机程序。在其他的实施例中，指令序列可以驻留在与控制器 16、18 通信的计算机可读介质或存储器上。

[0040] 机器人系统 2 可进一步包括本领域已知的其他组件，例如被配置为连接各种系统组件的网络媒体（未示出）、可编程逻辑设备（未示出）、以及与至少一个控制器 16、18 进行电通信的教示设备 20 中的至少一个。在一个特定实施例中，教示设备 20 可包括监视器，并被配置为图形化地表示第一机器人 6、第二机器人 8、公用空间 10 的第一部分 12、以及公用空间 10 的第二部分 14，以便机器人系统 2 的操作者根据需要进行观看。教示设备 20 可包括用于发起指令序列以轻柔推机器人的装置。

[0041] 根据本公开的示例性指令序列在图 2 中示出。该指令序列包括干扰检查自动区方

法 200。干扰检查自动区方法 200 首先包括初始步骤 202,其在工作间 4 中提供至少一个公用空间 10。初始步骤 202 之后是第一确定步骤 204 以及第二确定步骤 206。第一确定步骤 204 包括:确定在第一机器人 6 沿着第一编程路径移动的期间占用的公用空间 10 的第一部分 12。第二确定步骤 206 包括:确定在第二机器人 8 沿着第二编程路径移动的期间占用的公用空间 10 的第二部分 14。应当认识到,第一和第二部分 12、14 可自动地确定,即,没有必要如本领域所已知的那样,使用户指定每个机器人的操作区。第一部分 12 和第二部分 14 然后在比较步骤 208 中进行比较,以确定在它们之间是否存在重叠 210。

[0042] 作为非限制性的示例,第一和第二确定步骤 204、206 可以在具有第一和第二机器人 6、8 的机器人系统 2 的初始操作期间基本上实时地实施。该初始操作可在禁闭(lockdown)模式下对第一和第二机器人 6、8 实施,以避免其任何潜在的碰撞。在另一示例中,第一和第二确定步骤 204、206 可通过执行第一和第二编程路径的离线运行来实施,例如,以识别出可能由机器人 6、8 在实时操作期间所占用的公用空间 10 的第一和第二部分 12、14。例如在第一和第二确定步骤 204、206 期间识别的第一和第二部分 12、14 可进一步保留在存储器上,并且第一和第二机器人 6、8 在后续操作期间重新使用。应当理解的是,机器人运动的编程路径或轨迹从而被预处理并且被保留,以防止其碰撞的发生。

[0043] 响应于重叠 210 的存在,选择第一和第二机器人 6、8 的第一和第二移动步骤 212、214。如果在第一部分 12 和第二部分 14 之间存在重叠 210,则相信机器人 6、8 很可能发生碰撞,并选择第一移动步骤 212。第一移动步骤 212 包括第一机器人 6 沿着第一编程路径和第二机器人 8 沿着第二编程路径中的仅一个移动。如果在第一部分 12 和第二部分 14 之间不存在重叠 210,那么相信机器人 6、8 不可能发生碰撞,并选择第二移动步骤 214。第二移动步骤 214 包括使第一机器人 6 沿着第一编程路径移动并且使得第二机器人 8 沿着第二编程路径移动。

[0044] 第一和第二确定步骤 204、206 可由本领域已知的任何措施实施。在根据本公开的一个特定实施例中,第一和第二确定步骤 204、206 可包括用具有至少一个体素 302 的至少一个体素模型 300 表示公用空间 10、第一部分 12 和第二部分 14 中的至少一个的步骤,例如在图 3 至 6 中示例示出的那样。体素 302 为表示三维空间中的规则网格上的值的体积元素。例如,比较步骤 208 可包括确定在表示第一部分 12 的第一体素模型 300 和表示第二部分 14 的第二体素模型 300 之间是否存在重叠 210。体素模型 300 通常由多个体素 302 表示,其逼近各种组件的形状,例如,第一和第二机器人 6、8 的基底、臂、以及工具。

[0045] 参考图 3 和 4,示例性的体素模型 300 可包括在沿 X 轴、Y 轴和 Z 轴的坐标处设置的多个体素。体素模型 300 被配置为描绘由第一机器人 6 和第二机器人 8 的至少一个所占用的公用空间 10 的三维体积。体素模型 300 可以例如是控制器 16、18 或别的计算机可读介质或存储器产生和保存到控制器 16、18 或别的计算机可读介质或存储器的数据文件的形式。在某些实施例中,体素模型 300 是动态的,并包括与机器人移动序列相关的多个体素模型 300。动态体素模型 300 可用于描绘在第一机器人 6 和第二机器人 8 中的至少一个的移动期间所占用的公用空间 10 的三维体积。每个自动区从体素模型 300 的多个快照的累积/叠印得到,以一个或多个 ITP 间隔来获得每个快照。

[0046] 现在参考图 5 至 6,第一和第二机器人 6、8 的体素模型 300 可包括至少一个体素化球体 500 和体素化圆柱体 502,以表示由第一和第二机器人 6、8 所占用的三维体积。如此处

进一步描述的那样,体素化球体 500 和体素化圆柱体 502 可表示各种组件,例如第一和第二机器人 6、8 的基底和臂。体素化球体和圆柱体 500、502 通常包括相对彼此设置为逼近第一和第二机器人 6、8 的常规形状和边界的多个体素 302。

[0047] 本领域普通技术人员应当理解的是,体素化球体 500 和圆柱体 502 可由任何适当的措施产生。例如,体素化球体 500 可通过首先将表示机器人 6、8 组件的示例性球体投影在 X-Y 平面上来逼近。然后识别出包围该球体的最小的圆柱体“A”。然后相同的球体被投影至 Y-Z 平面上,并识别出包围该球体的最小的圆柱体“B”。相同的球体然后被投影至 Z-X 平面上,并且识别出包围该球体的最小圆柱体 C”。然后,通过识别出圆柱体“A”、“B”和“C”的交集来逼近体素化球体 500。

[0048] 在另一实施例中,通过首先找到包围表示机器人 6、8 组件的球体的最小立方体(box)来逼近体素化球体 500。该立方体具有体积占用“A”。然后识别出在立方体“A”内并在球体之外的体素体积占用“B”。然后通过从体积“A”中减去体积“B”来逼近体素化球体 500。

[0049] 备选地,可以通过识别出示例性球体和 X-Y 平面的交集来逼近体素化球体 500。该交集包括沿着 Z 轴的球体中心,并形成相交的圆形板“A1”。该圆形板“A1”被指定任意高度,并且随后识别该交叉的圆形板“A1”的体素占用。进一步,通过将示例性球体和 X-Y 平面的交集沿着 Z 轴移位来定位圆形板“A2”、“A3”...“A(n)”,并且类似地识别出其体积占用。然后通过将针对示例性球体的圆形板所识别的每个体素占用进行合并来逼近该体素化球体 500。

[0050] 通过识别出左半球/整个球体的体素占用“A”,识别出右半球/整个球体的体素占用“B”,以及识别出在左半球和右半球之间的多个圆形板的体素占用“C”,来逼近体素化圆柱体 502。然后创建垂直于基底线的基底圆形板。该基底线连接在圆柱体的端部的两个半球。随后的圆形板可以通过沿着 Z 轴移位基底圆形板来得到。备选地,可以计算基底线的体素占用,并且随后移位基底线,以填充两个半球之间的整个区域。随后,通过合并每个体素占用“A”、“B”和“C”来逼近体素化圆柱体 502。

[0051] 一种用于逼近体素化圆柱体 502 的备选方法包括,首先识别出能够包围表示机器人组件的示例性圆柱体的最小立方体。该立方体具有体积占用“A”。然后识别出位于立方体之内并在该圆柱体之外的体素体积占用“B”。随后,通过用体素体积占用“A”减去体素占用“B”来逼近体素化圆柱体 502。本领域普通技术人员应当理解,可以根据需要采用适用于逼近表示机器人 6、8 组件的体素化球体 500 和圆柱体 502 的其他措施。

[0052] 体素化还可以从可表示机器人、臂装饰物、以及加工的任何任意 CAD 表面或体积类似地确定。

[0053] 体素化是一种非常有效的用于表示由机器人运动部分占用的空间的方法。体素化过程具有一些计算开销来创建体素化空间,但是运行时间组件是非常有效的。一旦空间被体素化,不论运动路径期间由机器人所占用的公用空间多复杂或多大,体素化空间的最大存储需求是固定的。对体素化空间之间干扰的运行时间检查是非常有效的。虽然优选实施例提供了由体素化区域表示的干扰检查自动区,但是可以使用任何一种对机器人在运动路径期间所占用的空间的体积或表面表示的方法。

[0054] 进一步的实施例在编程运动、由轻推发起的运动,或者由其它装置发起的运动期

间避免碰撞。它们还可保护移动机器人免受与工作间中的固定机器人或其他固定物体或其他指定区域进行碰撞。针对编程运动,该体素化过程、或其它干扰检查自动区建模过程可以在离线期间或在测试运行期间发生,并且体素化数据被有效地存储以便于后续的检索。当如从新程序、机器人轻推或从由其它装置发起的运动确定一个新的运动路径时,该过程还可实时地发生。在这些情况下,指令序列可通过使用教示设备或发起运动的其他装置来确定。不允许开始实际运动直到该过程完成为止,并且与该干扰检查自动区和其它占用的干扰检查自动区没有干扰。

[0055] 在进一步的实施例中,公用空间 10 的第一部分 12 和第二部分 14 中的至少一个可包括加工空间和由附接在第一机器人 6 和第二机器人 8 中的至少一个的工具所占用的装饰物组件(例如 ArcTool 送丝器)。确定重叠 210 的步骤还可包括,将第一部分 12 和第二部分 14 的至少一个与由非机器人障碍物(未示出)所占用的公用空间 10 的第三部分进行比较。该障碍物可以根据需要由用于定义公用空间 10 的尺寸、形状和位置的参数集合来手动指定。在另一示例中,方法 200 可进一步包括确定由障碍物占用的公用空间 10 的第三部分的步骤。应当理解的是,由非机器人障碍物占用的公用空间的第三部分也可以根据需要由类似的体素模型 300 中的体素 302 表示。

[0056] 本公开的方法 200 可进一步包括第一机器人 6 已经沿着第一编程路径移动和第二机器人 8 已经沿着第二编程路径移动的至少一种情况之后确定公用空间 10 的非占用部分的步骤。公用空间 10 的非占用部分然后可以根据需要被释放给顺着另外编程路径或轨迹的另一机器人使用。在进一步的实施例中,方法 200 可包括将公用空间 10 的第一部分 12 和第二部分 14 的坐标传送给至少一个控制器 16、18 以进一步根据本公开的方法 200 来使用的步骤。控制器 16、18 然后可以如此处所述被采用来实施第一和第二确定步骤 204、206 中的至少一个,并将该坐标转换为表示第一和第二部分 12、14 的体素模型 300。

[0057] 技术人员应当意识到,一旦确定在公用空间 10 的第一和第二部分 12、14 之间是否存在重叠,第一机器人 6 和第二机器人 8 的各自移动可由优先级值来确定。作为非限制性的示例,第一和第二机器人 6、8 的优先级值可基于先进先出系统。在另一示例中,第一和第二机器人 6、8 中的一个可以总是具有比第一和第二机器人 6、8 中的另一个更高的优先级值。该更高的优先级值可以基于针对机器人系统 2 中机器人 6、8 的优先级的预定用户设置。在进一步的示例中,优先级值可基于公用空间 10 中非占用部分的可用性来选择。技术人员可以根据需要向第一和第二机器人 6、8 指定期望的优先级值。

[0058] 在进一步的实施例中,方法 200 进一步包括针对可能导致死锁状况的重叠 210 的存在而分析多个编程路径的步骤。随后可以根据需要调整多个编程路径的操作序列,以防止死锁状况的发生。操作序列可以手动或自动地调整,例如,根据需要,基于优先级值进行。

[0059] 应当意识到的是,本公开的干扰检查自动区系统 2 和方法 200 有利地提供了优于传统多臂机器人系统的竞争性优势。系统 2 和方法 200 提供了三维且实时的碰撞避免。系统 2 对编程的需求最小,因为系统 2 避免对指定或教示干扰区的需要。产品的停机时间也因此被减少。

[0060] 本领域普通技术人员还应当意识到,体素化球体 500 和圆柱体 502 的使用最小化了通常与产生机器人模型典型相关的处理需求。第一和第二机器人 6、8 的移动事先与系统 2 进行通信,例如,通过离线产生体素模型 300。同样地,保留无碰撞的第一和第二机器人 6、

8 的编程路径或轨迹。处理路径为在应用过程（例如切割、焊接、碾磨、布线、喷漆、分发、或别的类似过程）发生期间的运动序列。一旦过程路径启动，就在没有中断的情况下完成整个过程路径，这通常是重要的。还应当意识到第一和第二机器人 6、8 的关键过程路径也根据本方法 200 被同样地保护。

[0061] 根据需要，系统 2 支持在相同控制器 16、18 内和越过多个控制器 16、18 的多机器人臂碰撞避免。系统 2 和方法 200 进一步提供了简化的机器人配置和防止对如使用如某些现有技术系统和方法所实施的 I/O 握手协议的需要。干扰检查自动区系统 2 和方法 200 简化了多臂干扰检查的配置，并防止多机器人臂死锁。

[0062] 干扰预防和死锁预防紧密相关。干扰发生在当两个机器人试图占用相同的物理空间时。干扰防止的一个方法包括检查正在或将被所有机器人在当前运动指令中占用的空间，并在干扰发生之前使一个或多个机器人停止。这种类型的干扰防止具有一个主要缺陷，即在一个或多个已停止的机器人当前占用另一机器人所需要或将需要的空间时，非常容易发生死锁，并且该已停止的机器人不能继续，因为其继续被另一机器人阻挡。

[0063] 另一干扰防止方法是通过在机器人程序中程序化地插入进入和退出区命令而手动建立干扰区。这些命令可以提供与 PLC 或其他机器人的通信，以便促使每次只允许一个机器人臂占用指定区。对于单个区，存在较少关注。主要问题可能是优先级；如果区在变成可用时被多个机器人等待则谁可首先得到该区中的一个。但是，针对多个区，很容易发生死锁。例如，如果每个机器人当前占用一个区并且下一个运动处于由其它机器人占用的区中，则容易寻求邻近区。在此情况下，每个机器人将永久等待其它机器人放弃期望的区。死锁状况将发生。

[0064] 根据本发明的系统和方法通过在执行具有潜在死锁状况的运动之前确定无死锁运动语句来消除以上情形中的死锁状况。无死锁运动语句的该确定可以离线进行、在正常执行之外进行，或它在正常产生执行期间进行。如果有充足的 CPU 处理时间可用，那么在正常产生执行期间的确定提供了响应于动态状况的最大灵活度，例如 I/O 时序或外部事件或序列的时序上的变化。为使得 CPU 影响最小，该确定可以离线地进行，其中能够分析编程序列的多个排列并找到最优执行序列。

[0065] 根据本发明的系统和方法无需事先知晓无死锁序列。该系统和方法将自动地确定潜在的死锁状况并识别出避免这些状况的方式。该系统和方法不需要优先级值来防止死锁。该系统和方法防止干扰状况并同时防止存在死锁状况的可能性。

[0066] 在图 7 中所示的是针对将由第一机器人运行的“原始程序 A”列表 401 和将由第二机器人运行的“原始程序 B”列表 403 的行号 (line numbers) 的“无死锁矩阵”400。矩阵 400 中的阴影单元表示了定义无死锁区域的上升 (ascending) 凸壳。大部分阴影单元表示了从一个程序的行“i-1”至行“i”和在另一程序中从行“j-1”至行“j”的路径上的行进期间具有干扰的程序行对。需要剩下的单元来完成该上升凸壳。在矩阵 400 中的单元可被指定为 (i, j) 的格式，其中“i”为程序“A”行号并且“j”为程序“B”行号。因此，干扰单元为 (3, 6 ; 3, 7 ; 4, 5 ; 4, 6 ; 4, 7 ; 5, 4 ; 5, 5 ; 5, 6 ; 5, 7) 并且剩下的单元为 (3, 4 ; 3, 5 ; 4, 4)。

[0067] 例如，如果程序“A”在行 3 上并且程序“B”在行 5 上，则存在死锁状况，因为如果“A”试图移动到行 4，则将与行 5 上的“B”产生干扰，并且如果“B”试图移动到行 6，那么将与行 3 上的“A”产生干扰。因此，没有机器人可以继续。通过完成从“A”的行 3 至 5 和“B”

的行 4 至 7 的死锁区,每次只有一个程序“ A”或“ B”能够占用该区。这在示为“新程序 A”列表 402 和“新程序 B”列表 404 的示例程序中识别为区 [1]。

[0068] 图 8 示出了一个示例,其中“干扰矩阵”405 可被再分为三个无死锁区:区 [1];区 [2];和区 [3]。该干扰单元为 (3,3;3,4;3,5;3,6;3,7;4,5;4,6;4,7;5,4;5,5;5,6;5,7;6,6;6,7)。区 [1] 包括单元 (3,3;3,4;3,5;3,6;3,7),区 [2] 包括单元 (4,5;4,6;4,7;5,4;5,5;5,6;5,7),并且区 [3] 包括单元 (6,6;6,7)。上升凸壳通过作为虚拟干扰的单元 (4,4) 而创建,以使得可以产生三个上升凸壳,并且总区域以上升方式凸出。这三个区然后被合并至“新程序 A”列表 406 和“新程序 B”列表 407 中。不会存在下面这种情况,即在任何无干扰行对组合中,至少一个程序可在不产生干扰的情况下前进到下一行。

[0069] 图 9 示出了具有多个干扰区域以及可能存在死锁的两个死锁区域的程序对的示例。“程序 A”描绘从行 3 至行 15 的路径 409。“程序 B”410 描绘从行 9 至行 21 的路径。程序具有第一死锁区域 412 和第二死锁区域 411,如在路径和表 413 中所示出的那样。制作单个死锁区域的简单方法是将所有的干扰区域和中间无干扰区域结合至如表 414 所示的单个区域中,该区域本质上将是无死锁的。

[0070] 图 10 示出了分别运行“程序 A”和“程序 B”的机器人 6、8 的五个序列。在最左侧的第一序列中,机器人 6 运行“程序 A”,机器人 8 不移动。在其右侧的下一序列中,机器人 8 运行“程序 B”,机器人 6 不移动。在中间序列中,机器人 6、8 并发地运行各自的程序并且发生了碰撞。其右侧的下一序列中,示出机器人 6、8 利用引起死锁的现有技术的干扰避免特征并发地运行各自程序。在最右侧的序列示出了机器人 6、8 利用根据本发明的死锁防止并发地运行各自程序,使得在不发生干扰的情况下完成了两个程序。

[0071] 以上描述的系统和方法将自动地添加和 / 或修改程序语句,以防止碰撞和管理潜在的死锁二者。在某些情况中,所发出的自动等待命令可能会使得针对给定任务的整个循环时间增加。在不能接受增加的循环时间的情况下,根据本发明的系统和方法可以推荐或自动创建重新排序的路径,这将使得机器人可以运行整个程序而不停止。换句话说,如果存在两个机器人将同时占用相同空间的路径部分,则该系统和方法将自动地推荐一条新的路径序列(设想以不同的顺序焊接点),从而可以消除这种情况。程序指令的重新排序的该特征具有用于油漆店应用的潜力。

[0072] 如果在程序中发生条件化执行语句,那么分析可包括序列执行的所有组合。或者,为简单起见,包含在条件化分支区域中的程序片段可独立处理。典型地,机器人程序员将教示这样的条件化区域,以使得该区域的进入和退出路径与其它机器人没有干扰,并且以使得所述独立处理是可行的。但是,如果条件化区域的进入或退出路径与其他机器人有干扰,则应当分析条件化区域和非条件化区域的多个组合以保证无干扰并且无死锁。

[0073] 该方法的一个优势在于,可消除死锁状况的相同机制也可消除在正常程序运动期间的干扰。因此,在正常程序运动期间可以不对机器人-机器人干扰进行实时检查。这可以节约大量的 CPU 利用率,从而允许处理时间能用于其它目的,诸如更快的内插率或更短的内插时间。在误差状况期间或当程序未运行时,一种减少使用 CPU 的干扰检查(诸如在序列号 12/124,430、题为“Multi-Arm Robot System Interference Check via Three Dimensional Automatic Zones”的共同未决美国专利申请中提出的)可用于防止干扰。

[0074] 由程序所占用的空间可以为物理空间、时间、程序序列、行号、程序或行的完成率、

或与机器人、程序或系统有关的状态变化的函数。该干扰区域和死锁区域通过定义占用空间的相同函数关联起来。

[0075] 避免死锁的方法可以以多种方式处理。最简单的方式是扩大干扰区域,以使得它们可以防止如图 9 中所示的死锁和干扰。扩大干扰区域以使得对于整个程序仅存在一个干扰区域将保证系统不会死锁,但是如果循环时间是重要的,那么可使用某些类别的序列控制和 / 或多个干扰区域。另一种方式是提供干扰区域之外的执行序列,以使得死锁被防止。这可以通过使得第一机器人在该第一机器人继续之前等待第二机器人完成可能存在死锁状况的程序执行的某些部分来简单地进行。

[0076] 存在确定占用空间几种方式。最简单的方式是占用空间纯粹是机器人贯穿程序执行的过程中所占用的累积空间。当多个机器人具有以此方式来确定的其占用空间时,则干扰区域可以简单地是作为每个机器人的累积占用空间的交集的空间。每次只允许一个机器人进入该空间将防止干扰和死锁二者。

[0077] 另一种方式是确定作为时间的函数的占用空间。在时间上的每个点处,存在由每个机器人占用的空间。通过将占用空间表示为特定间隔上的时间整合,占用空间可被表示为与时间有关的一系列离散空间。该干扰区域是在相同时间或时间间隔的机器人相交。可通过每次只允许一个机器人在干扰区域中来避免干扰。因为相关的时间由于序列控制而变化,所以时间基础必须被调整为在已应用序列控制之后与新时间相关联。

[0078] 针对以上的基于时间的方法,所述时间可以被表示为时间间隔数 (number)。在此方式中总的经过时间变得不重要,并且干扰区域可被表示为间隔数的矩阵。如果在特定的间隔状态存在干扰,那么必须不允许该状态出现,因为可能会在该间隔状态中发生碰撞。

[0079] 类似地,程序序列可被表示为行号、完成百分比、程序状态等等。针对任何这种序列化表示,可存在作为所选参数的函数的所定义的对干扰的表示。避免干扰的目标是避免其中存在干扰的函数集合的状态。对于可以被表示为序列化的离散间隔或序列化的状态的函数,这可被表示为如图 8 中示出的干扰矩阵。干扰矩阵包含干扰区域。当然,矩阵仅是表示数据的一种方式,但是矩阵可以为更加视觉化的表示。通常矩阵为稀疏的,因此任何稀疏矩阵技术也可用于该表示。

[0080] 状态定义 (例如 I/O 状态或过程或其它状态) 可被用来定义区域。只要该状态具有序列化含义或可被定义为表示与机器人位置和占用空间相关的序列化活动,则它们可以被用于干扰和死锁区域的参数表示。

[0081] 一旦干扰矩阵已经确定,一种死锁防止方法就会避免序列号单调增加的任何“陷阱”,其中干扰将阻挡向前序列。通过使得干扰矩阵上升凸出,避免了死锁和干扰二者。这种上升凸出干扰矩阵可被称为死锁矩阵。该死锁矩阵包含图 7 中所示的死锁区域。与干扰相似,该死锁矩阵仅是表示数据的一种方式。

[0082] 一旦死锁矩阵已被确定,就存在多种避免死锁和干扰的措施。最简单的措施是在程序中插入指令来调整执行时序,由此避开死锁矩阵的死锁区域。向程序添加指令具有这样的优点,即可通过简单地检查程序来确定执行时序的确切性质。另一种措施允许在实际程序修改之外确定时序。

[0083] 干扰和死锁区域的参数定义提供了避免干扰和死锁的通用方法。对区域的该表示允许通过各种因素来控制各种程序。

[0084] 同样,为了减少 CPU 使用率,可更新实际程序以包含区和序列信息,因此在程序中包含所有用来避免死锁状况的必要信息。这消除了对程序序列、干扰和潜在的死锁状况进行实时检查的需要。

[0085] 图 11 是根据本发明系统和方法的死锁避免序列的流程图。在步骤 420 中,程序被教示给机器人控制器 16、18。在步骤 421 中,确定占用空间。然后,在步骤 422 中确定干扰区域。潜在的死锁区域在步骤 423 中被确定。在步骤 424 中,确定避免死锁的程序执行需求。最后,在步骤 425 中提供如以上所述的避免死锁的机制。

[0086] 步骤 423 在图 12 中更加详细地示出。在步骤 426 中,存在识别出的占用空间和所有干扰区域。在步骤 427 中,检查每个干扰区域组合。在决策步骤 428 中进行对“存在死锁状况吗?”的检查。如果输出为“否”,则该步骤 423 在步骤 429 中结束。如果输出为“是”,则步骤 430 提供防止死锁发生的措施并且该方法返回至步骤 427。

[0087] 当为了示例该发明的目的而已经示出某些代表性实施例和细节时,进行各种变化而不背离该公开的范围,对于本领域技术人员来说是显而易见的,这将进一步在后面的所附权利要求中进行描述。

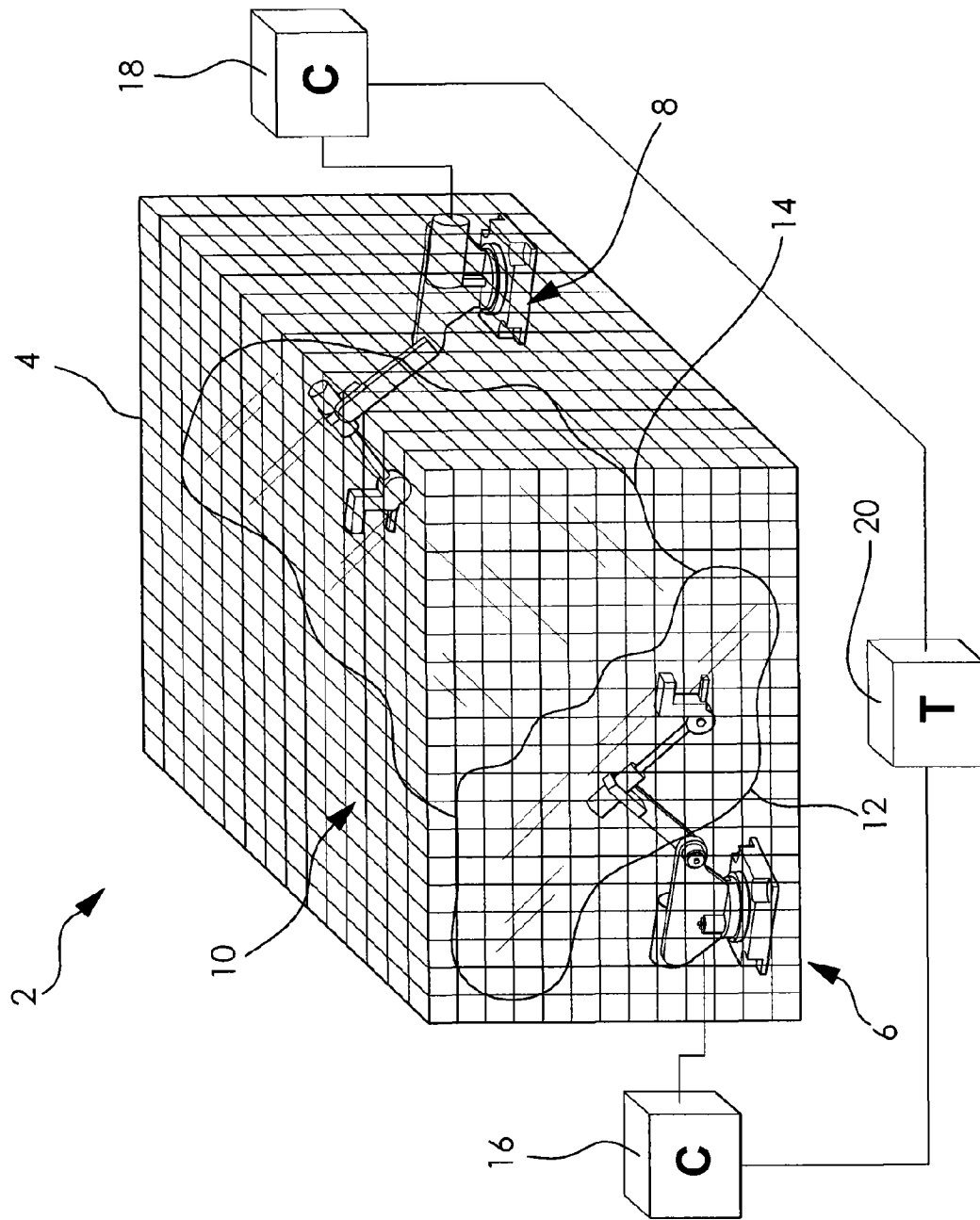


图 1

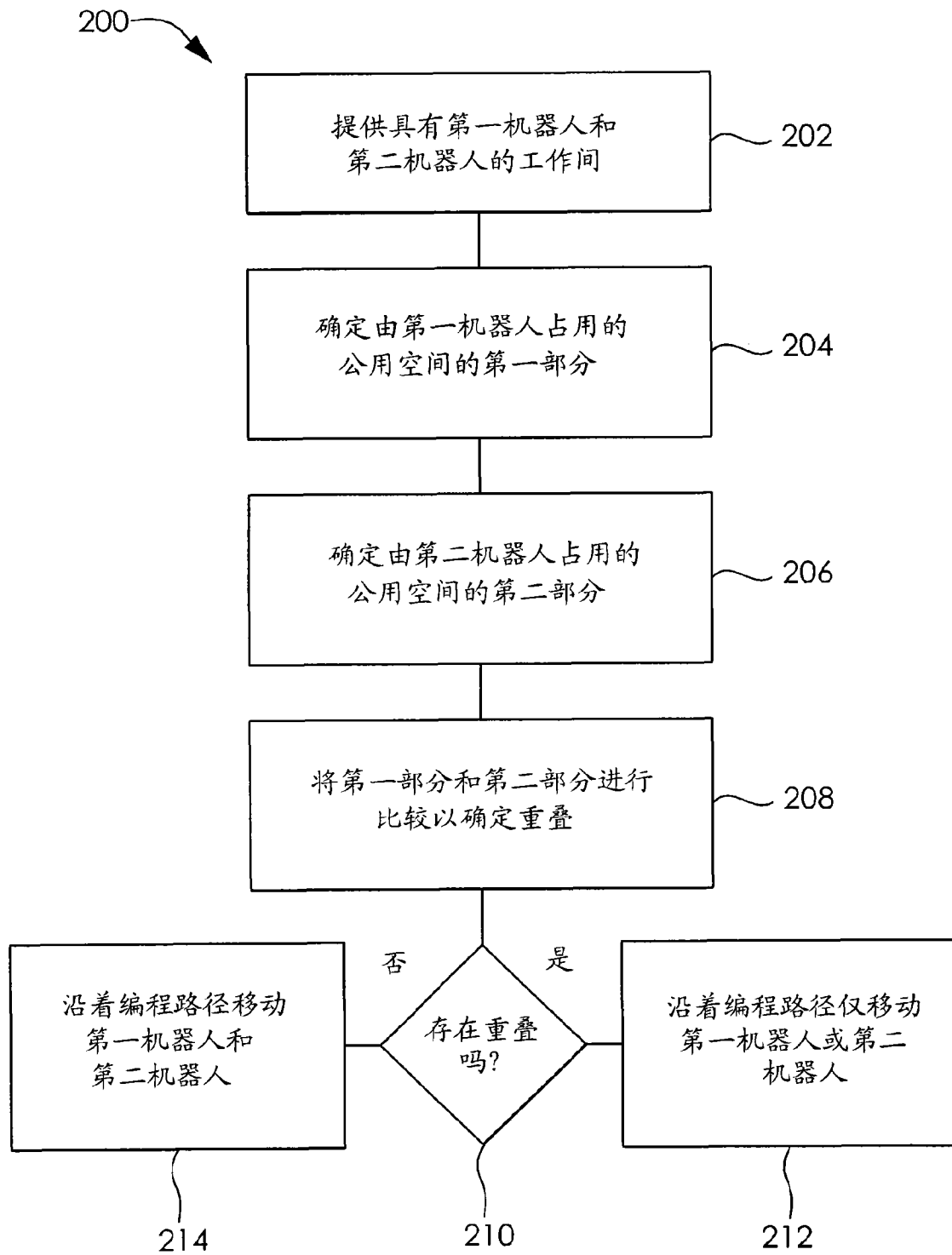


图 2

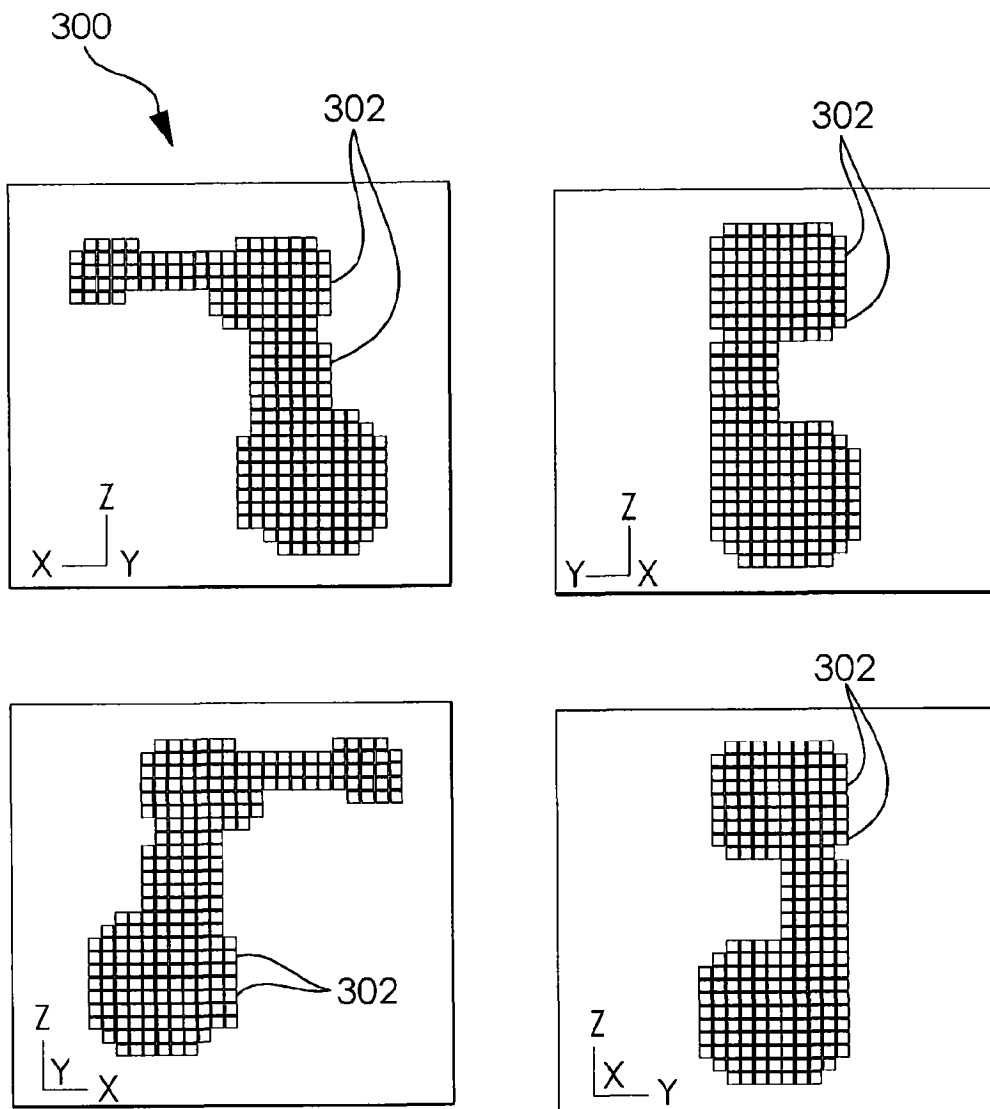


图 3

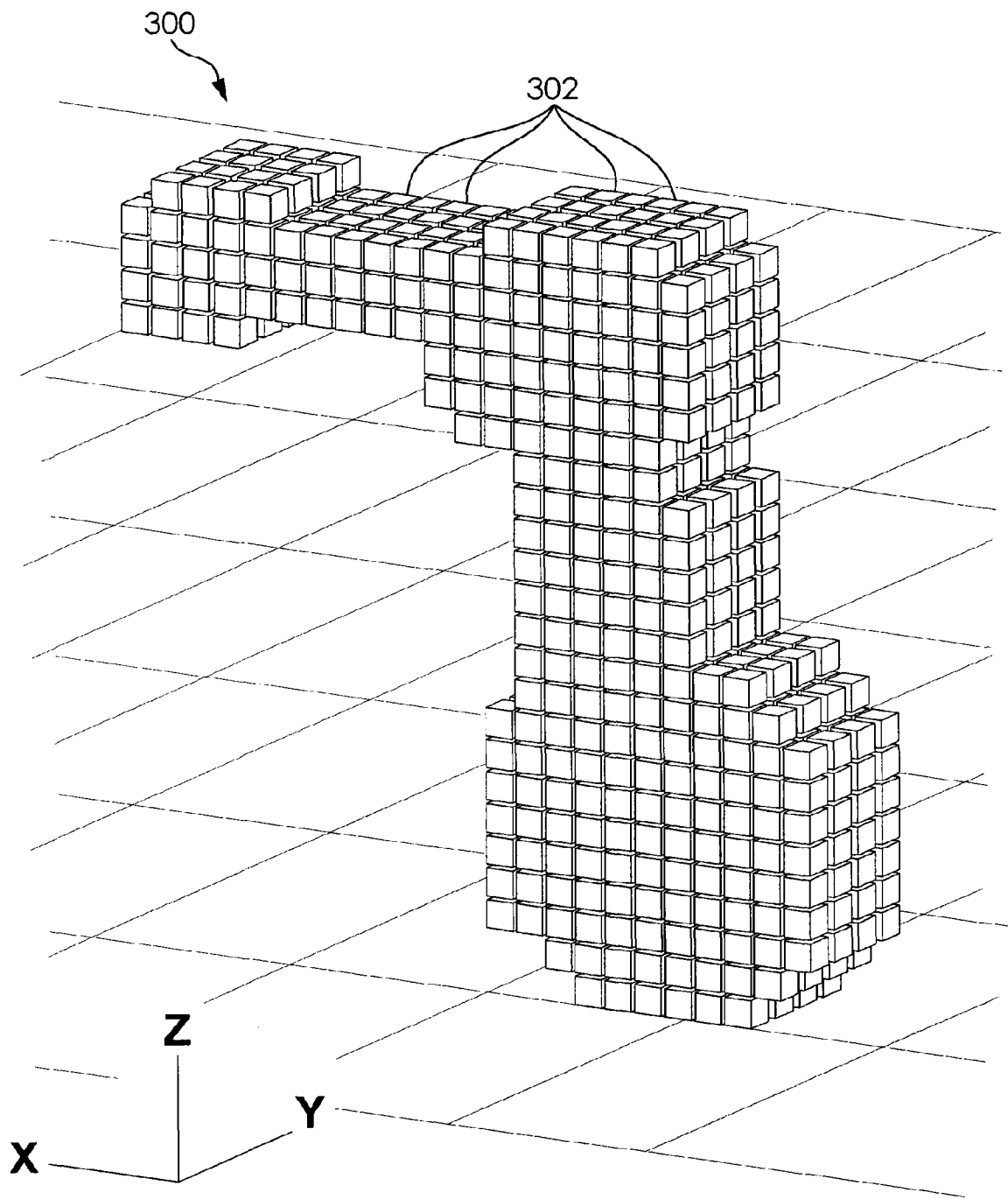


图 4

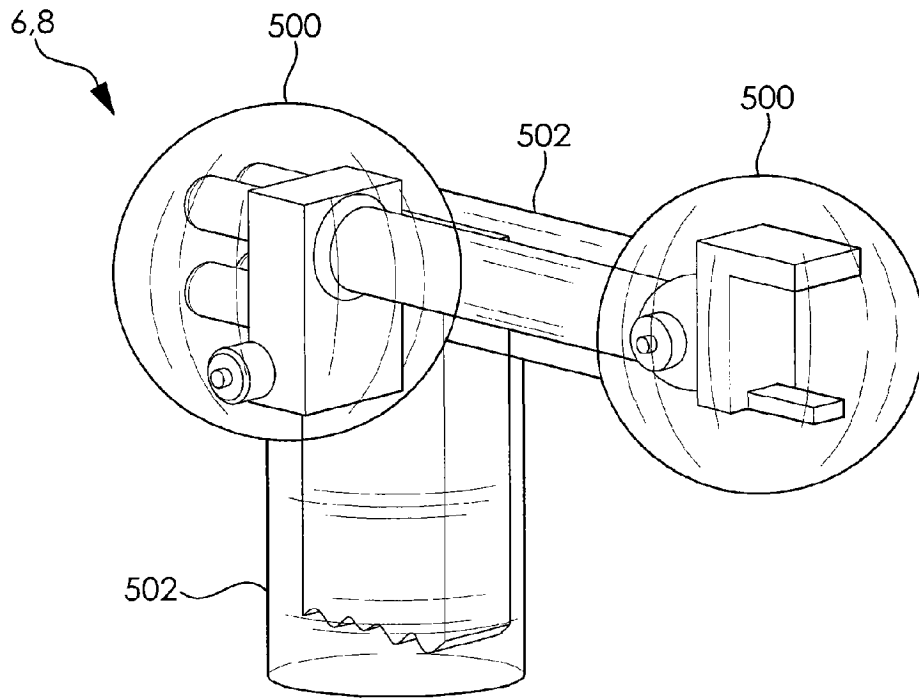


图 5

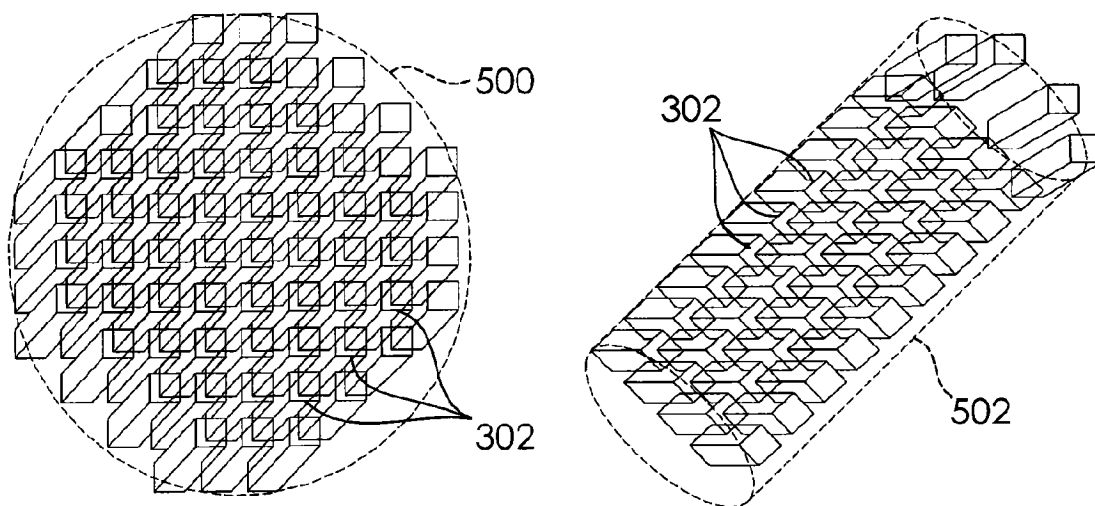


图 6

无死锁矩阵											
		程序B行号									
		1	2	3	4	5	6	7	8	9	10
程序A行号	1	0	0	0	0	0	0	0	0	0	0
	2	0	0	0	0	0	0	0	0	0	0
	3	0	0	0	1	1	1	1	0	0	0
	4	0	0	0	1	1	1	1	0	0	0
	5	0	0	0	1	1	1	1	0	0	0
	6	0	0	0	0	0	0	0	0	0	0
	7	0	0	0	0	0	0	0	0	0	0
	8	0	0	0	0	0	0	0	0	0	0
	9	0	0	0	0	0	0	0	0	0	0
	10	0	0	0	0	0	0	0	0	0	0

400

原始程序A	
1	L P[1] 2000MM/SEC FINE
2	L P[2] 2000MM/SEC FINE
3	L P[3] 2000MM/SEC FINE
4	L P[4] 2000MM/SEC FINE
5	L P[5] 2000MM/SEC FINE
6	L P[6] 2000MM/SEC FINE
7	L P[7] 2000MM/SEC FINE
8	L P[8] 2000MM/SEC FINE
9	L P[9] 2000MM/SEC FINE
10	L P[10] 2000MM/SEC FINE

401

图 7

新程序A	
1	L P[1] 2000MM/SEC FINE
2	L P[2] 2000MM/SEC FINE
3	进入区 [1]
4	L P[3] 2000MM/SEC FINE
5	L P[4] 2000MM/SEC FINE
6	L P[5] 2000MM/SEC FINE
7	退出区 [1]
8	L P[6] 2000MM/SEC FINE
9	L P[7] 2000MM/SEC FINE
10	L P[8] 2000MM/SEC FINE
11	L P[9] 2000MM/SEC FINE
12	L P[10] 2000MM/SEC FINE

402

原始程序B	
1	L P[1] 2000MM/SEC FINE
2	L P[2] 2000MM/SEC FINE
3	L P[3] 2000MM/SEC FINE
4	L P[4] 2000MM/SEC FINE
5	L P[5] 2000MM/SEC FINE
6	L P[6] 2000MM/SEC FINE
7	L P[7] 2000MM/SEC FINE
8	L P[8] 2000MM/SEC FINE
9	L P[9] 2000MM/SEC FINE
10	L P[10] 2000MM/SEC FINE

403

图 7(继续)

404

新程序B	
1	L P[1] 2000MM/SEC FINE
2	L P[2] 2000MM/SEC FINE
3	L P[3] 2000MM/SEC FINE
4	进入区 [1]
5	L P[4] 2000MM/SEC FINE
6	L P[5] 2000MM/SEC FINE
7	L P[6] 2000MM/SEC FINE
8	L P[7] 2000MM/SEC FINE
9	退出区 [1]
10	L P[8] 2000MM/SEC FINE
11	L P[9] 2000MM/SEC FINE
12	L P[10] 2000MM/SEC FINE

图 7(继续)

干扰矩阵											
		程序B行号									
		1	2	3	4	5	6	7	8	9	10
程序A行号	1	0	0	0	0	0	0	0	0	0	0
	2	0	0	0	0	0	0	0	0	0	0
	3	0	0	1	1	1	1	1	0	0	0
	4	0	0	0	0	1	1	1	0	0	0
	5	0	0	0	1	1	1	1	0	0	0
	6	0	0	0	0	0	1	1	0	0	0
	7	0	0	0	0	0	0	0	0	0	0
	8	0	0	0	0	0	0	0	0	0	0
	9	0	0	0	0	0	0	0	0	0	0
	10	0	0	0	0	0	0	0	0	0	0

405

原始程序A	
1	L P[1] 2000MM/SEC FINE
2	L P[2] 2000MM/SEC FINE
3	L P[3] 2000MM/SEC FINE
4	L P[4] 2000MM/SEC FINE
5	L P[5] 2000MM/SEC FINE
6	L P[6] 2000MM/SEC FINE
7	L P[7] 2000MM/SEC FINE
8	L P[8] 2000MM/SEC FINE
9	L P[9] 2000MM/SEC FINE
10	L P[10] 2000MM/SEC FINE

401

图 8

新程序A	
1	L P[1] 2000MM/SEC FINE
2	L P[2] 2000MM/SEC FINE
3	进入区 [1]
4	L P[3] 2000MM/SEC FINE
5	退出区 [1]
6	进入区 [2]
7	L P[4] 2000MM/SEC FINE
8	L P[5] 2000MM/SEC FINE
9	退出区 [2]
10	进入区 [3]
11	L P[6] 2000MM/SEC FINE
12	退出区 [3]
13	L P[7] 2000MM/SEC FINE
14	L P[8] 2000MM/SEC FINE
406	15 L P[9] 2000MM/SEC FINE
	16 L P[10] 2000MM/SEC FINE

图8(继续)

无死锁矩阵											
		程序B行号									
		1	2	3	4	5	6	7	8	9	10
程序A行号	1	0	0	0	0	0	0	0	0	0	0
	2	0	0	0	0	0	0	0	0	0	0
	3	0	0	1	1	1	1	1	0	0	0
	4	0	0	0	1	1	1	1	0	0	0
	5	0	0	0	1	1	1	1	0	0	0
	6	0	0	0	0	0	1	1	0	0	0
	7	0	0	0	0	0	0	0	0	0	0
	8	0	0	0	0	0	0	0	0	0	0
	9	0	0	0	0	0	0	0	0	0	0
	10	0	0	0	0	0	0	0	0	0	0

407

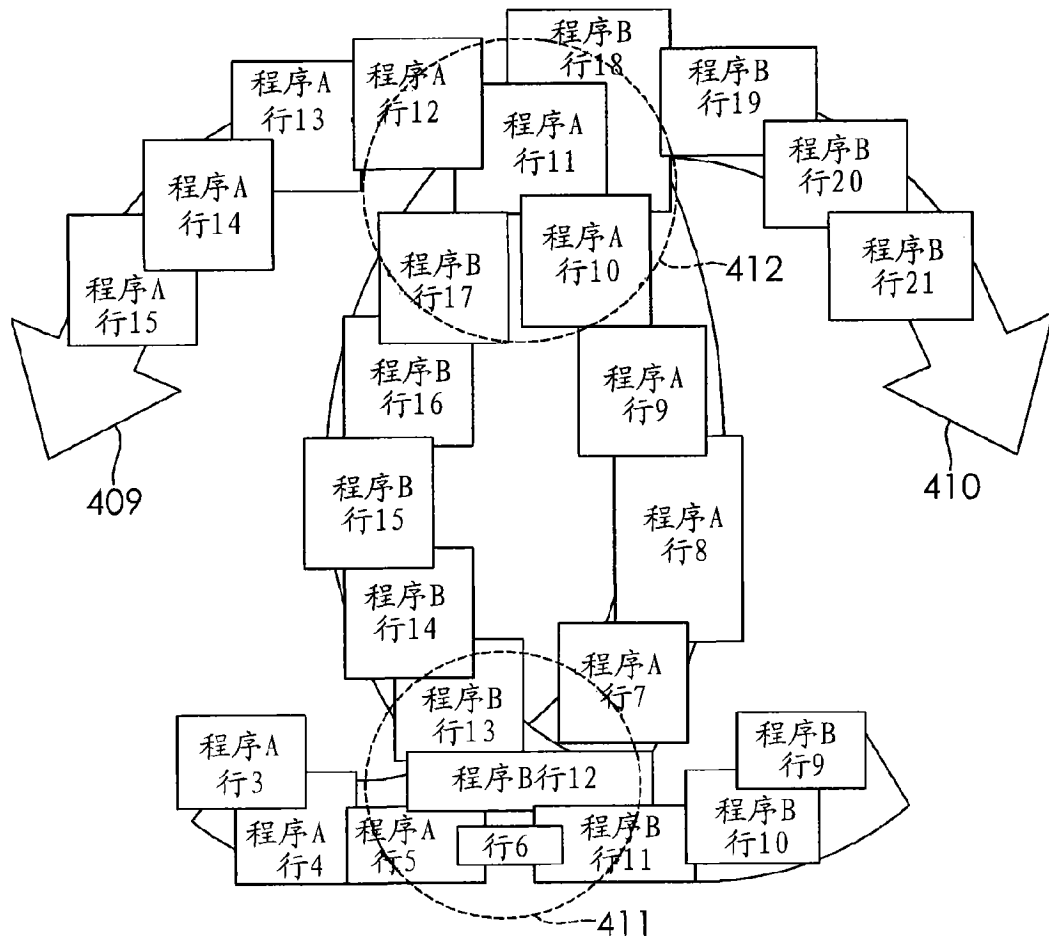
原始程序B	
1	L P[1] 2000MM/SEC FINE
2	L P[2] 2000MM/SEC FINE
3	L P[3] 2000MM/SEC FINE
4	L P[4] 2000MM/SEC FINE
5	L P[5] 2000MM/SEC FINE
6	L P[6] 2000MM/SEC FINE
7	L P[7] 2000MM/SEC FINE
8	L P[8] 2000MM/SEC FINE
9	L P[9] 2000MM/SEC FINE
10	L P[10] 2000MM/SEC FINE

403

图8(继续)

新程序B	
1	L P[1] 2000MM/SEC FINE
2	L P[2] 2000MM/SEC FINE
3	进入区 [1]
4	L P[3] 2000MM/SEC FINE
5	进入区 [2]
6	L P[4] 2000MM/SEC FINE
7	L P[5] 2000MM/SEC FINE
8	进入区 [3]
9	L P[6] 2000MM/SEC FINE
10	L P[7] 2000MM/SEC FINE
11	退出区 [1]
12	退出区 [2]
13	退出区 [3]
14	L P[8] 2000MM/SEC FINE
408	L P[9] 2000MM/SEC FINE
16	L P[10] 2000MM/SEC FINE

图 8(继续)



死锁区域		程序A	程序B
413-1	开始行	10	17
	结束行	12	18
413-2	开始行	5	11
	结束行	6	13
死锁区域		程序A	程序B
414-1	开始行	5	11
	结束行	12	18

图 9

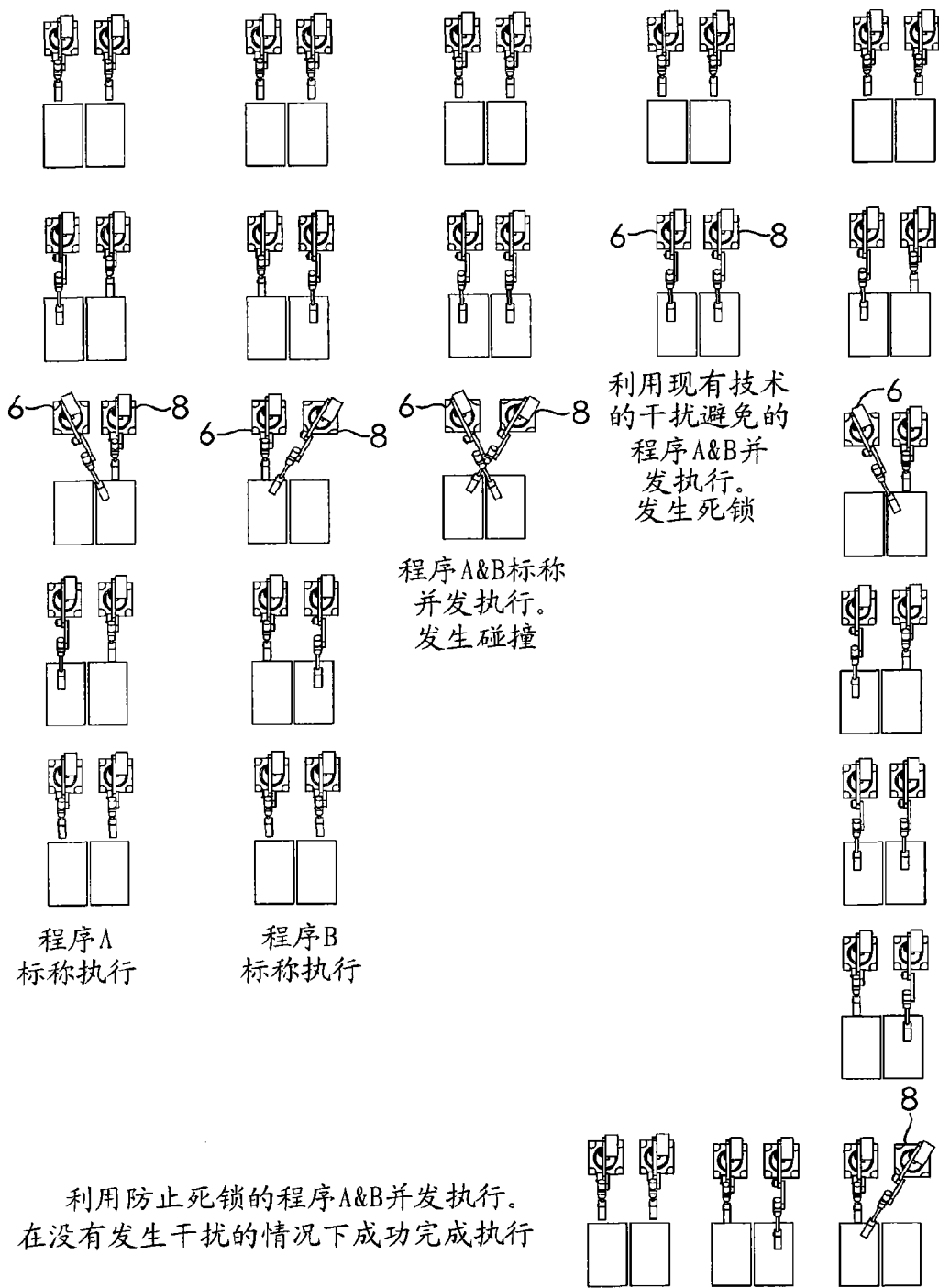


图 10

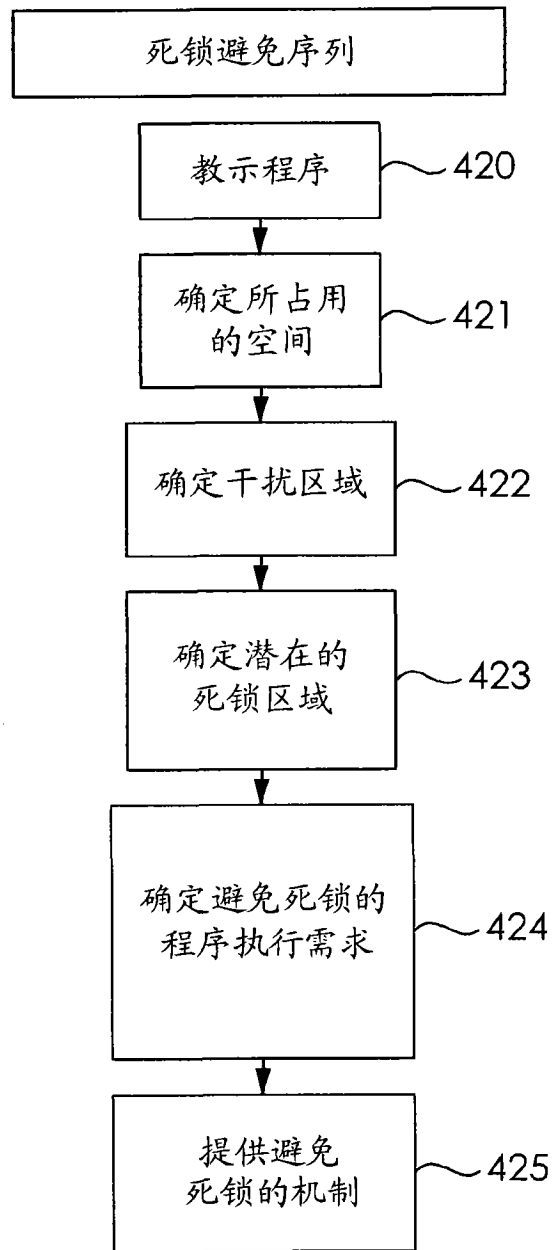


图 11

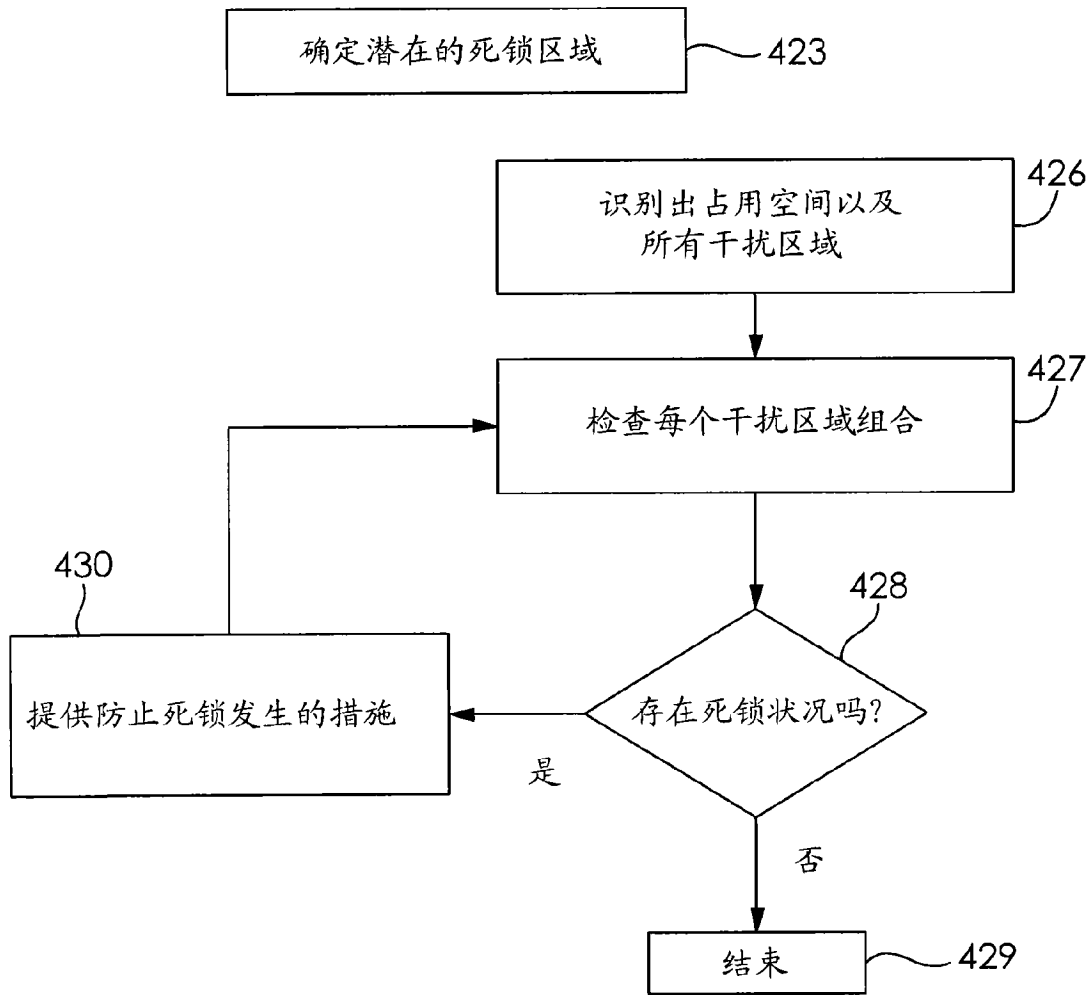


图 12