

## 移动型模块化机器人的高效重构规划方法

胡亚南<sup>1,2</sup>, 马书根<sup>1</sup>, 李斌<sup>1</sup>, 王明辉<sup>1</sup>, 王越超<sup>1</sup>

(1. 中国科学院沈阳自动化研究所机器人学国家重点实验室, 辽宁 沈阳 110016; 2. 中国科学院大学, 北京 100049)

**摘要:** 基于穷举思想的模块化机器人重构规划方法的计算复杂度与模块数呈阶乘关系, 难以适用于模块较多的场合. 为此, 本文提出一种与模块数呈线性复杂度的高效重构规划方法. 将重构规划问题视为最优控制问题, 通过求解 Hamilton-Jacobi-Bellman 方程得到定义在状态空间上的值函数和最优控制律. 值函数的吸引域决定了各模块对应的最优目标, 而应用控制律能够得到不同状态到达最优目标的最优运动轨迹. 这样既避免了考虑不同构形模块间的多种对应关系所产生的组合爆炸, 同时也能得到各模块的满足运动学约束的最优轨迹. 仿真试验验证了方法的可行性和计算效率.

**关键词:** 模块化机器人; 重构规划; 运动学; 最优控制; HJB 方程

**中图分类号:** TP242

**文献标识码:** A

**文章编号:** 1002-0446(2016)-04-0467-08

### An Efficient Reconfiguration Planning Method for Mobile Type Modular Robots

HU Yanan<sup>1,2</sup>, MA Shugen<sup>1</sup>, LI Bin<sup>1</sup>, WANG Minghui<sup>1</sup>, WANG Yuechao<sup>1</sup>

(1. State Key Laboratory of Robotics, Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang 110016, China;

2. University of Chinese Academy of Sciences, Beijing 100049, China)

**Abstract:** The reconfiguration planning methods for modular robots based on the exhaustion idea have factorial time complexity with respect to the number of modules, which are difficult to be applied to the cases containing large numbers of modules. To solve this problem, an efficient reconfiguration planning method is proposed, which has linear time complexity with respect to the number of modules. The reconfiguration planning problem is viewed as an optimal control problem. By solving the Hamilton-Jacobi-Bellman equation, the value function and optimal control law defined on the state space are obtained. The domain of attraction of the value function determines the optimal goal for the individual modules, and the optimal trajectories to the optimal goals at different states can be obtained by applying the optimal control law. Thus, the combination explosion caused by calculating the corresponding relations between modules of two configurations can be avoided, and the optimal trajectories of the individual modules that satisfy their kinematic constraints can be obtained at the same time. Simulation results validate the feasibility and efficiency of the proposed method.

**Keywords:** modular robot; reconfiguration planning; kinematics; optimal control; Hamilton-Jacobi-Bellman equation

## 1 引言 (Introduction)

由于技术的进步和实际需求的推动, 机器人的应用场景逐渐从静态的简单结构化环境扩展到动态的复杂非结构化环境. 复杂且多变的环境特征要求机器人有更强的适应能力和容错能力, 这给机器人的结构设计和运动控制造成了挑战, 但同时也催生了新的设计思想和控制策略. 模块化机器人的出现符合这一趋势, 因此受到越来越多的关注<sup>[1]</sup>. 模块化机器人由大量结构和功能相对简单的模块组成. 借助自身的连接装置, 多个模块能够组成一个整体, 而通过改变模块间的连接关系和相对位置,

模块化机器人能够变换为具有不同功能的多种构形, 进而满足多种任务的需求. 根据模块间的几何结构, 模块化机器人可以分为 3 种基本类型: 栅格型、链型和移动型<sup>[2]</sup>. 栅格型和链型模块化机器人的模块一般无法独立地运动或运动能力有限, 各模块必须连接形成群体构形、相互协作才能完成有效的运动. 而移动型模块化机器人的每个模块都可视为自主的移动机器人, 能够借助环境独立运动.

在遇到新的环境和任务时, 模块化机器人需要改变自身的构形以适应当前环境, 这一过程又称为重构. 重构规划的任务是生成各模块的控制量和运动轨迹从而实现重构. 在重构规划中, 目标构形一

般由人指定或由上级控制层给出,但各模块的运动目标和运动轨迹均需要由重构规划算法生成。

与传统机器人的运动规划相比,模块化机器人的重构(运动)规划面临两个挑战。首先,每个模块作为一个独立的机器人,都需要对其运动进行规划,导致构形空间的维度高。而一般规划算法的复杂度与维度呈指数关系。其次,在规划中不仅需要考虑模块自身以及各模块间的约束关系,还要考虑当前模块对整体结构的依赖和影响。针对这些问题,人们根据模块化机器人的特点提出了多种解决方法。例如,为了避免对大量模块同时进行规划,将部分模块作为一个整体进行规划,这些模块被称为宏模块。Bhat<sup>[3]</sup>基于宏模块的思想提出了分层的方法实现栅格型模块化机器人的重构。Vassilvitskii<sup>[4]</sup>将宏模块推广到包含自由模块,并使用宏模块作为“骨架”建立期望的构形。而为了处理模块间复杂的约束,基于规则的策略被提出。Yoshida<sup>[5]</sup>提出了一种分层规划方法,同样适用于栅格类型。其中全局层负责生成各模块的运动路径以实现整体构形的运动,而局部层基于规则库对各模块的运动进行详细规划以跟踪全局层的运动路径。Dumitrescu<sup>[6]</sup>基于一组预先定义的运动规则提出了一种分布式运动规划方法,并对在特定规则下重构的可达性进行了分析。其研究同样针对栅格型机器人。Shen<sup>[7]</sup>受生物荷尔蒙的启发,针对链型机器人提出一种模块间的通信规则。各模块根据规则协同运动实现机器人整体的运动或重构。

目前的重构规划研究主要针对栅格型和链型模块化机器人,而对移动型模块化机器人的研究较少。本文针对移动型模块化机器人的重构问题提出一种具有线性时间复杂度的规划方法,克服了基于穷举思想的方法在模块较多时计算量大的缺陷,使其能够用于包含大量模块的重构任务。将最优控制理论中的动态规划原理推广到重构问题,通过求解最优控制问题同时得到各独立模块的最优目标和运动轨迹,并使用改进的避撞方法消除轨迹间的干涉,使模块的运动满足几何和运动学约束。仿真结果验证了方法的有效性和计算效率。

## 2 移动型模块化机器人:轮手一体机器人 (Mobile type modular robot: wheel-manipulator robot)

对于移动型模块化机器人,其单模块具有一定的独立移动能力,能够实现对大范围环境的分布式探索;而多个模块连接成为一个群体后,其运动能

力能够超越单模块的限制,实现在困难环境中的穿行及作业。鉴于这类模块化机器人独特的运动性能和环境适应能力,研制了轮手一体机器人<sup>[8]</sup>。轮手一体机器人的单个模块由5自由度机械臂和三角形履带轮体组成,如图1(a)所示。轮手一体机器人的机械臂位于轮体的一侧,其外形不对称且与运动方向相关。轮体和机械臂均配备连接机构。机械臂在机器人移动时用于控制转向;在重构时,其作为主动连接机构与其它模块上的连接机构对接组成期望的群体构形,如图1(b)所示的三角构形。

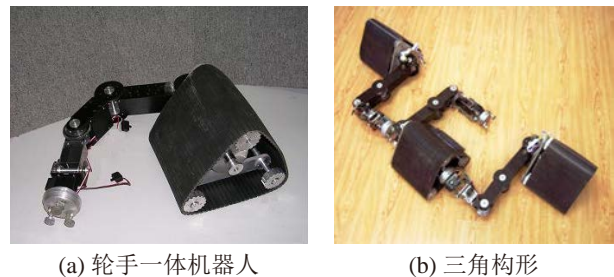


图1 轮手一体机器人及其群体构形

Fig.1 Wheel-manipulator robot and the group configuration

在做转向运动时,轮手一体机器人能够通过调整机械臂关节的转角改变其转向半径。精确地预测机器人的转向需要考虑复杂的动力学模型以及履带的受力情况。本文为了简化问题,采用受非完整约束的小车模型对其运动进行描述,如图2(a)所示。因此,机器人的运动方程可以表示为

$$\begin{cases} \dot{x} = v \cos \theta \\ \dot{y} = v \sin \theta \\ \dot{\theta} = \frac{v}{L} \tan \phi \end{cases} \quad (1)$$

其中,  $(x, y)$  是轮体中心在全局坐标系中的位置,  $\theta$  是轮体相对全局坐标系  $X$  轴的转角,  $v$  是机器人的运动速度,  $\phi$  是机械臂末端导轮的偏转角,  $L$  是轮体中心到导轮的距离,如图2(b)所示。

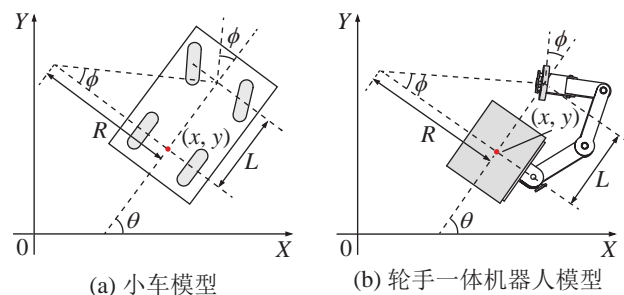


图2 轮手一体机器人的等效模型

Fig.2 The equivalent model of the wheel-manipulator robot

### 3 重构规划方法的基本思想 (Basic idea of the reconfiguration planning method)

假设机器人重构的环境为平坦的地面, 且不存在静态障碍物. 下面介绍重构的实现步骤和重构规划方法的基本思想.

#### 3.1 重构步骤

由于移动型模块化机器人 (例如轮手一体机器人) 的运动通常受到约束, 其模块需要足够的自由空间机动以改变其方向. 而在群体构形中, 各模块间的距离较小, 直接对其运动进行规划很容易发生碰撞导致规划耗时较长. 为此定义过渡状态, 并将重构过程分解为 3 个步骤<sup>[9]</sup>. 为便于表述, 以图 3 所示的重构任务为例进行详细分析.

重构任务的初始构形为串形构形, 而目标构形为三角构形. 第 1 步, 串形构形中的各模块机械臂手爪断开与其它模块的连接, 各模块的轮体做简单的直线移动, 移动到安全位置后停止 (图 3(a)–(b)). 移动到安全位置的目的是给机器人足够的机动空间. 这些安全位置定义为过渡状态 (transition state). 对于目标构形也定义类似的过渡状态. 第 2 步, 已经处于独立状态的各模块由串形构形的过渡状态运动到三角构形的过渡状态 (图 3(b)–(c)). 第 3 步, 各模块执行简单的直线运动, 到达最终的目标状态, 机械臂与其它模块上的连接装置对接组成目标构形完成整个重构任务 (图 3(c)–(d)). 注意机械臂的相对位置对完成重构有重要影响, 因此在重构规划中不可忽略.

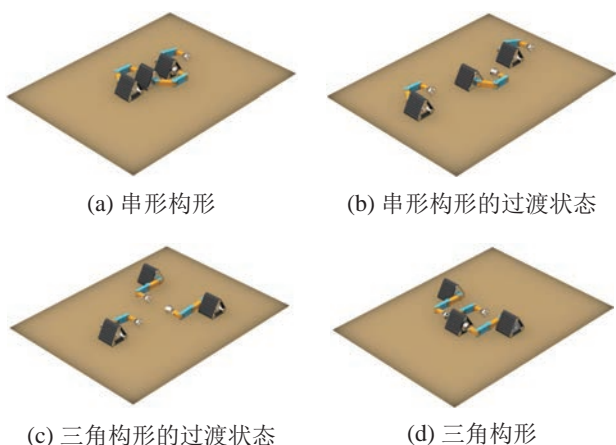


图 3 重构步骤的例子

Fig.3 Example illustrating the reconfiguration steps

在上述步骤中, 由于第 1 和第 3 步涉及的运动较简单且各模块的运动目标明确, 不存在模块的对应关系选择和复杂的运动规划. 而第 2 步较为复杂, 也是实现重构的关键环节, 既需要选择各模块

的运动目标, 也需要计算其运动轨迹. 因此, 后文的研究只针对第 2 步中存在的问题.

#### 3.2 最优控制理论的启发

实现重构的第 2 步需要解决两个问题. 首先, 每个模块各自的运动目标 (即目标构形中各模块的过渡状态, 后文称为子目标) 未知, 需要得到各模块对应的子目标, 称为目标分配问题; 其次, 生成各模块到达子目标的运动轨迹, 称为轨迹生成问题. 显然, 不同的目标分配方式通常导致不同的运动轨迹, 进而产生不同的代价. 模块化机器人重构时, 在实现组成目标构形的前提下通常要求满足某一性能指标最优, 例如运行时间最短或能量消耗最小. 模块化机器人的模块受到体积限制, 自身携带的能量有限, 因此考虑以能量作为优化指标更有实用意义.

为了得到能量最小的目标分配方式, 需要各模块的运动代价, 而各模块的运动代价取决于为其分配的子目标. 因此, 目标分配与轨迹生成之间存在相互依赖的关系. 为得到最优的重构方案, 可以采用穷举法, 即尝试模块与子目标间所有可能的分配并分别计算其运动轨迹进而求得代价. 但穷举法的计算量为  $O(n!)$ , 其中  $n$  为参与重构的模块数量. 实际存在的目标分配方式随模块数量急剧增加, 如图 4 所示. 当模块数为 9 时, 可能的分配方式已经有 362 880 种. 因此穷举法只适用于模块数量较少的情况 ( $n < 6$ ), 模块数较多时计算所有的分配方式显然将非常耗时.

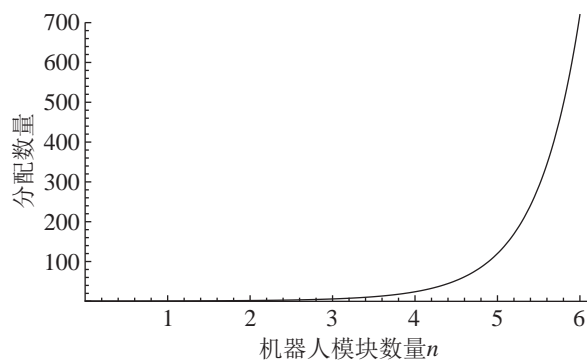


图 4 分配数量与模块数的关系

Fig.4 Relation between the number of assignments and the number of modules

在最优控制理论中, 解决单个目标下的最优控制问题可以借助动态规划原理. 得到的解包含两部分: 值函数和最优控制律. 值函数是机器人状态的函数, 其含义为机器人在当前状态下到达目标的最优代价值; 而最优控制律为当前状态下机器人以最优的代价到达目标应采取的控制策略. 如果将最优



控制问题推广到多目标的情况，则值函数定义了每个目标的吸引域。状态位于某个目标吸引域中的机器人到达该目标是最优的。于是可以得到不同状态到不同目标的最优分配关系，而机器人的最优轨迹只需应用最优控制律于运动方程并积分即可得到。如果将重构中目标构形的过渡状态（子目标）定义为状态空间中的目标，就可以应用动态规划原理求解重构规划问题。但此时得到的是单机机器人在不同状态下的最优控制解，而没有考虑多机器人的几何约束。如果直接用于重构规划，尤其是对于存在运动学约束的机器人，各模块沿最优轨迹运动时可能发生碰撞。为此，在得到各模块的最优子目标及最优轨迹后需要利用避撞方法对其运动进行协调。

为便于理解，下面以2维平面内运动不受约束的机器人重构问题为例介绍基于动态规划原理的重构方法的基本思想。假设目标构形包含2个子目标，而附近环境中存在3个独立模块。需要从中选择2个模块完成重构，且应使机器人总的移动距离最小。模块与子目标存在的分配方式共有  $3! = 6$  种，图5展示了其中2种分配方式（如图5中箭头所示）。

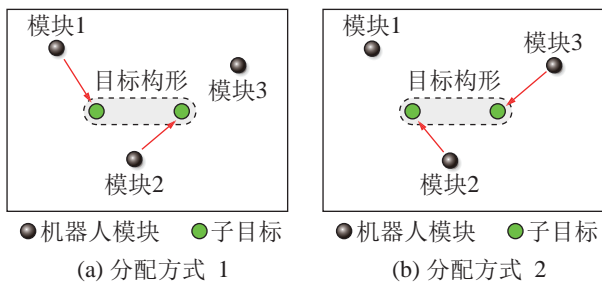


图5 两种目标分配方式  
Fig.5 Two ways of goal assignment

根据规定的代价指标（距离），动态规划的计算过程为以子目标的代价（距离为0）为初始值，逐步向外扩展得到附近环境中所有点到（任一）子目标最近的距离。计算得到各点的最优距离值，其图像如图6(a)所示。假设机器人的控制量为速度。在计算值函数的过程中同时得到了机器人的最优控制策略，即在每个位置处都定义了一个最优速度向量，使机器人以最小的代价（距离）到达最近的子目标，如图6(b)所示。在子目标吸引域内存在多个模块时，只需要选择代价最小（距离子目标最近）的模块即可（即模块2和模块3）。因此，最优的目标分配和运动轨迹能够同时得到。

当某个子目标的吸引域内没有机器人模块时重构将无法完成，如图7(a)所示。为此，提出添加虚

拟障碍物的方法。注意虚拟障碍物并不对应环境中的真实障碍物，其目的是影响值函数（的吸引域）从而保证重构任务能正确完成。为便于计算，虚拟障碍物的形状可采用简单的几何形状，例如圆弧或圆环。建立虚拟障碍物的方法是，首先确定距离子目标最近的模块，假设其距离为  $r_m$ ，如图7(b)所示。虚拟障碍物表示为以产生冲突的子目标为中心的圆环。圆环的半径  $r_o$  应大于  $r_m$ ，以保证能将最近的模块包含进来，但其半径不能过大，否则将失去限制吸引域的意义。本文采用  $r_o = r_m + \max(h_x, h_y)$ ，其中  $(h_x, h_y)$  是数值求解最优控制问题时各维度的分辨率。假设目标构形中的模块距离大于  $r_o$ ，则总能保证每个虚拟障碍物只包含1个子目标。

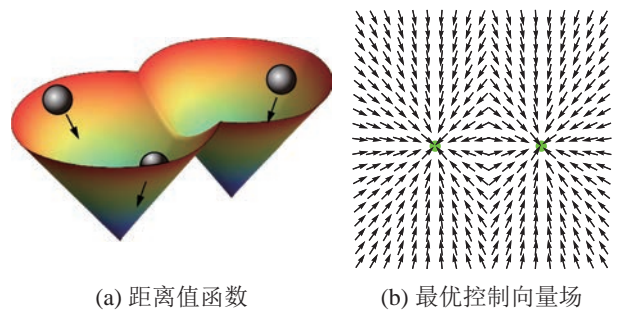


图6 值函数与最优控制向量场  
Fig.6 Value function and the feedback control vector field

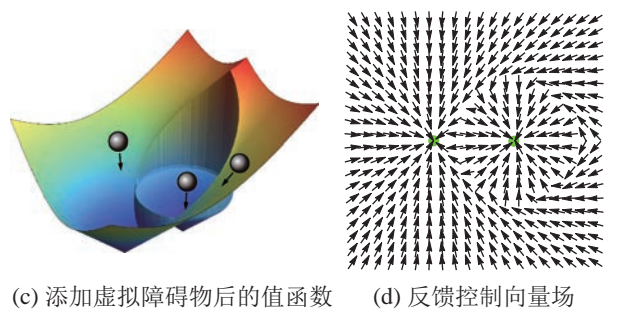
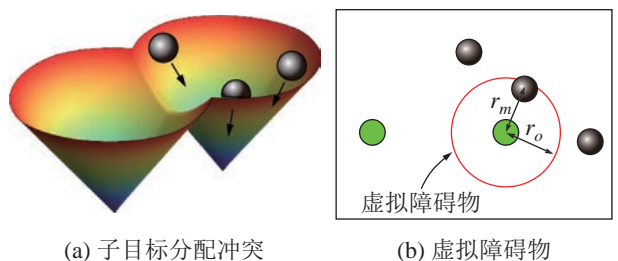


图7 子目标分配冲突消除  
Fig.7 Conflict elimination of subgoals assignment

设置虚拟障碍物后需重新求解最优控制问题，更新后的值函数图像如图7(c)所示。受到虚拟障碍物的影响，值函数的吸引域被改变，同时最优控制向量场也随之改变，使得每个子目标的吸引域内都至少有一个模块。

#### 4 基于最优控制的目标分配 (Goal assignment based on optimal control)

机器人的运动特性反映在其运动方程上, 为将 3.2 节中的思想应用于运动受约束的移动型模块化机器人, 根据运动方程定义最优控制问题. 为此, 将机器人的运动方程 (1) 改写为

$$\begin{cases} \dot{x} = v \cos \theta \\ \dot{y} = v \sin \theta \\ \dot{\theta} = s|v|/\rho \end{cases} \quad (2)$$

其中,  $\rho$  是机器人的最小转向半径,  $s$  是转向控制量, 其范围是  $s \in [-1, 1]$ ,  $s = 0$  表示机器人做直线运动. 由于轮手一体机器人的移动速度较慢, 其能量消耗与移动距离近似成正比. 因此将移动距离作为优化指标, 其与机器人的速度无关, 故考虑机器人以单位速度运动, 因此式 (2) 中  $v \in \{\pm 1\}$ . 此时距离最小等价于时间最短.

不考虑机械臂的姿态, 机器人的状态定义为向量  $\mathbf{x} = (x, y, \theta)$ . 默认状态下, 机器人前进时机械臂位于轮体右侧. 机器人的运动轨迹  $\mathbf{x}(t)$  由时间  $t$  参数化. 对任一状态  $\mathbf{x}$ , 定义轨迹集合  $\Omega(\mathbf{x})$  为所有以  $\mathbf{x}$  为起点且满足运动方程 (2) 的轨迹. 给定环境中的子目标状态集合  $X$  后, 可以定义状态  $\mathbf{x}$  的值函数  $V(\mathbf{x})$  为

$$V(\mathbf{x}) = \min\{T : \mathbf{x}(t) \in \Omega(\mathbf{x}), t \in [0, T], \mathbf{x}(T) \in X\} \quad (3)$$

其物理意义是以状态  $\mathbf{x}$  为起点, 终点位于子目标状态的满足式 (2) 的轨迹中代价最小的轨迹对应的代价 (时间). 根据动态规划原理, 式 (3) 可表示为

$$V(\mathbf{x}) = \min\{V(\mathbf{x}(\Delta t)) + \Delta t\} \quad (4)$$

调整式 (4) 中的项, 并除以  $\Delta t$  得到

$$\min\left\{\frac{V(\mathbf{x}(\Delta t)) - V(\mathbf{x})}{\Delta t} + 1\right\} = 0 \quad (5)$$

令  $\Delta t \rightarrow 0$ , 可得

$$\min\{\nabla V \cdot \dot{\mathbf{x}} + 1\} = 0 \quad (6)$$

其中,  $\nabla V = (V_x, V_y, V_\theta)$  是值函数的梯度向量, “ $\cdot$ ” 表示内积. 将运动方程 (2) 代入式 (6) 得到

$$\min_{\substack{v \in \{\pm 1\} \\ s \in [-1, 1]}} \{V_x v \cos \theta + V_y v \sin \theta + V_\theta s / \rho + 1\} = 0 \quad (7)$$

式 (7) 即 Hamilton-Jacobi-Bellman (HJB) 方程. HJB 方程是 1 阶双曲型偏微分方程, 其解是值函数

$V$ . 但 HJB 方程在传统意义下的解 (即处处连续可微的解) 不存在. 为此, 学界定义了弱解, 其中常用的弱解是黏性解. 由于黏性解的定义不满足符号对称性, 为求解 HJB 方程 (7), 需要将其符号反转改写为

$$-\min_{\substack{v \in \{\pm 1\} \\ s \in [-1, 1]}} \{V_x v \cos \theta + V_y v \sin \theta + V_\theta s / \rho + 1\} = 0 \quad (8)$$

式 (8) 是关于状态的非线性方程, 而非线性 HJB 方程一般难以得到其解析解, 因此通常寻求其数值解. 为此, 首先将状态空间离散化. 在  $(x, y, \theta)$  空间建立分辨率为  $(h_x, h_y, h_\theta)$  的均匀网格, 并令  $V_{ijk}$  表示位于网格节点  $(i, j, k)$  处的值函数的数值近似<sup>[9]</sup>. 随后将式 (8) 在网格节点上离散化. 式 (8) 的等价表达为

$$-|V_x \cos \theta + V_y \sin \theta| - |V_\theta| / \rho - 1 = 0 \quad (9)$$

将上式记为

$$H(\mathbf{x}, \nabla V) = 1 \quad (10)$$

其中,  $H(\mathbf{x}, \nabla V) = -|V_x \cos \theta + V_y \sin \theta| - |V_\theta| / \rho$  称为汉密尔顿函数.

文 [10] 给出了汉密尔顿函数的单调数值离散格式, 但直接用于式 (10) 较繁琐. 这里采用文 [11] 中的方法, 先固定控制量再按照文 [10] 中的单调格式离散化, 最后选取值最小的  $V_{ijk}$  作为数值近似解.

由于  $x$  和  $y$  方向的值函数的偏导数相互独立, 因此可以单独处理. 根据文 [10], 对形如  $au_x$  项的离散结果等价于逆风格式, 即

$$au_x = \begin{cases} a \frac{u_i - u_{i-1}}{h_x}, & a > 0 \\ a \frac{u_{i+1} - u_i}{h_x}, & a < 0 \end{cases} \quad (11)$$

其中,  $u_x$  是函数  $u$  在  $x$  方向的偏导数,  $a$  是独立于  $u_x$  的常量或变量. 逆风格式满足单调性, 其形式简洁, 比直接离散  $H(\mathbf{x}, \nabla V)$  更方便.

根据逆风格式, 当  $v = 1$  时,  $-v \cos \theta V_x$  的离散结果为

$$-v \cos \theta V_x = |\cos(kh_\theta)| \frac{V_{i+\xi_k, j, k} - V_{ijk}}{h_x} \quad (12)$$

其中,  $\xi_k = \text{sgn}(\cos(kh_\theta))$ ,  $\text{sgn}()$  为符号函数.

此时  $-v \sin \theta V_y$  的离散结果为

$$-v \sin \theta V_y = |\sin(kh_\theta)| \frac{V_{i, j+\eta_k, k} - V_{ijk}}{h_y} \quad (13)$$

其中,  $\eta_k = \text{sgn}(\sin(kh_\theta))$ .

当  $v = -1$  时,  $-v \cos \theta V_x$  的离散结果为

$$-v \cos \theta V_x = |\cos(kh_\theta)| \frac{V_{i-\xi_k, j, k} - V_{ijk}}{h_x} \quad (14)$$

同理,  $-v \sin \theta V_y$  的离散结果为

$$-v \sin \theta V_y = |\sin(kh_\theta)| \frac{V_{i, j-\eta_k, k} - V_{ijk}}{h_y} \quad (15)$$

最后考虑  $\theta$  方向,  $|V_\theta|$  可以直接离散化, 得到

$$|V_\theta| = \max \left\{ \frac{V_{ijk} - V_{i, j, k+1}}{h_\theta}, \frac{V_{ijk} - V_{i, j, k-1}}{h_\theta}, 0 \right\} \quad (16)$$

将在不同的控制量下得到的离散结果, 即式 (12) ~ (16) 分别代入  $-V_x v \cos \theta - V_y v \sin \theta + |V_\theta| / \rho - 1 = 0$ . 根据控制量和  $|V_\theta|$  的不同, 分为以下 4 种情况:

1)  $v = 1, |V_\theta| \neq 0$

$$V_{ijk} = \frac{cV_{i+\xi_k, j, k} + sV_{i, j+\eta_k, k} + d \min(V_{i, j, k \pm 1}) + 1}{c + s + d} := V_1 \quad (17)$$

2)  $v = 1, |V_\theta| = 0$

$$V_{ijk} = \frac{cV_{i+\xi_k, j, k} + sV_{i, j+\eta_k, k} + 1}{c + s} := V_2 \quad (18)$$

3)  $v = -1, |V_\theta| \neq 0$

$$V_{ijk} = \frac{cV_{i-\xi_k, j, k} + sV_{i, j-\eta_k, k} + d \min(V_{i, j, k \pm 1}) + 1}{c + s + d} := V_3 \quad (19)$$

4)  $v = -1, |V_\theta| = 0$

$$V_{ijk} = \frac{cV_{i-\xi_k, j, k} + sV_{i, j-\eta_k, k} + 1}{c + s} := V_4 \quad (20)$$

其中,  $c = |\cos(kh_\theta)| / h_x$ ,  $s = |\sin(kh_\theta)| / h_y$ ,  $d = 1 / (h_\theta \rho)$ .

最后, 选择最小值作为数值近似解  $V_{ijk}$ , 即

$$V_{ijk} = \min(V_1, V_2, V_3, V_4) \quad (21)$$

非线性代数方程 (21) 给出了每个网格节点处的  $V_{ijk}$  与相邻节点值  $V_{i \pm 1, j, k}$ ,  $V_{i, j \pm 1, k}$ ,  $V_{i, j, k \pm 1}$  的关系. 位于网格边界处的节点由于缺少某些方向上的相邻节点, 需要利用外插方法得到其相邻节点的值. 本文选择快速扫描方法 (fast sweeping method) 求解方程组 (21), 其具体实现方法可见文 [12].

## 5 基于 ORCA 的避撞策略 (Collision avoidance strategy based on ORCA)

求解最优控制问题能够得到单个机器人模块到达其最优子目标的最优轨迹. 但由于未考虑其它模块, 因此在多模块同时运动时无法保证相互间无干涉. 为此, 需要采用避撞算法对模块的运动进行协调. 最优互利避撞方法 (optimal reciprocal collision avoidance, ORCA) [13] 是一种高效的多机器人避撞方法, 其建立在速度障碍物的概念上. 速度障碍物定义为在一定时间  $\tau$  以内使 2 个机器人发生碰撞的相对速度的集合. 其几何形状为速度空间中顶点位于原点的截断圆锥, 如图 8(b) 所示. ORCA 方法假设机器人外形均为圆形, 如果机器人的外形是其它形状则可以采用最小外接圆作为近似.

对于 2 个可能发生碰撞的机器人, 只要将其相对速度移出速度障碍物, 即可在时间  $\tau$  以内避免碰撞. 相对速度的改变量应越小越好, 因为机器人通常沿预定轨迹运动, 这意味着每个机器人以最小的轨迹偏离实现避撞. 最小的改变量可以通过将相对速度投影到速度障碍物来实现, 这样得到了一个向量  $\mathbf{u}$ , 如图 8(c) 所示.

ORCA 的基本思想是将 2 个机器人原来的速度都叠加  $\mathbf{u}/2$ , 从而得到最优避撞速度集合. 如果每个机器人的速度都位于自己的最优避撞速度集合内, 它们一定不会在时间  $\tau$  以内发生碰撞. 上述集合可以表示为线性约束, 如果将与期望速度的距离作为优化目标, 便可以利用高效的线性规划方法求得最优的避撞速度.

ORCA 方法最初用于运动不受约束的移动机器人. 对于存在运动学约束的机器人 (例如轮手一体机器人), 下面提出一种简单的扩展, 使其生成避撞的控制量. 其基本思想是让机器人的速度跟随 ORCA 方法得到的避撞速度, 算法如 nonholonomic-ORCA 所示.

假设每个模块有唯一的标识 (记为 ID). 函数 ORCA() 用于生成机器人的避撞速度向量  $\mathbf{v}_{ID}^{aw}$ , 如图 8(d) 所示. 速度  $\mathbf{v}_{ID}^{prefer}$  是机器人沿最优轨迹运动时的速度向量. 函数 VectorAngle() 计算两个向量的角度  $\varphi$ , 其范围限制在区间  $(-\pi, \pi]$  内. 变量 direction 表示机器人的运动方向. 如果  $\varphi \in (-\pi/2, \pi/2]$ , 说明 ORCA 计算得到的避撞速度向量向机器人当前速度向量  $\mathbf{v}_{ID}$  的投影与  $\mathbf{v}_{ID}$  同向, 此时机器人无需改变运动方向, 即 direction = 1; 反之, 机器人需要改变运动方向, 即令 direction = -1. 机器

人新的控制量  $(v^{new}, \phi^{new})$  采用比例控制的思想, 其分别与  $direction$  和  $\phi$  成正比以实现跟踪, 其中正数  $k_v$  与  $k_\phi$  为比例系数。

nonholonomic-ORCA Algorithm	
1.	$v_{ID}^{new} = ORCA(v_{ID}^{prefer});$
2.	$\phi = VectorAngle(v_{ID}^{new}, v_{ID});$
3.	<b>if</b> $-\pi/2 < \phi \leq \pi/2;$
4.	$direction = 1;$
5.	<b>else;</b>
6.	$direction = -1;$
7.	<b>end if</b>
8.	$v^{new} = k_v \cdot direction \cdot \ v_{ID}^{new}\ ;$
9.	$\phi^{new} = k_\phi \cdot \phi;$

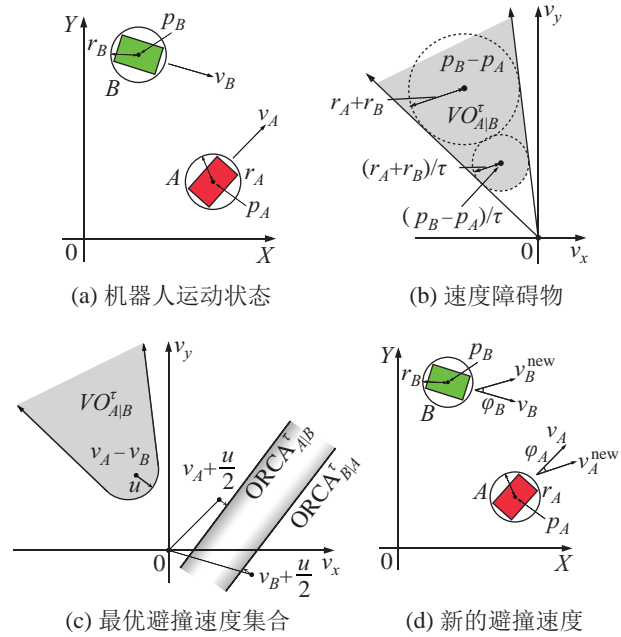


图 8 ORCA 避撞方法  
Fig.8 ORCA collision avoidance method

### 6 仿真 (Simulation)

下面通过仿真对提出的重构规划方法的有效性进行验证。仿真中离散网格借助 Armadillo C++ 线性代数库 [14] 实现。计算结果被符号计算软件 Mathematica 调用, 其负责运行 ORCA 避撞算法并显示重构规划结果。仿真采用的参数见表 1。

仿真实例考虑 2 种重构情况: 第 1 种情况下模块的初始状态组成为确定的构形, 且初始构形的模块数与目标构形相同; 在第 2 种情况下, 初始模块均为独立状态, 但独立模块数大于目标构形中的模块数。

表 1 仿真参数 Tab.1 Simulation parameters	
符号	数值
$\rho$	0.3 m
$h_x$	0.05 m
$h_y$	0.05 m
$h_\theta$	0.0634 rad
tolerance	0.1

实例 1 属于第 1 种类型, 任务是由串形构形变换为三角形。首先串形构形中的各模块相互分离, 随后移动到过渡状态。各模块在初始构形及目标构形对应的过渡状态之间的运动轨迹如图 9(a) 所示。虚线显示的圆是机器人的最小包围圆, 用于 ORCA 算法计算避撞速度。最后, 各模块调整手臂姿态并平移组成三角形 (图 9(b))。

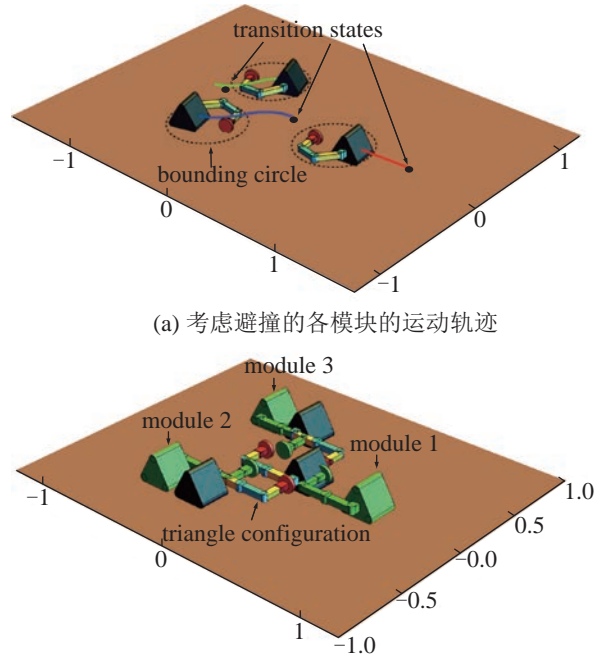


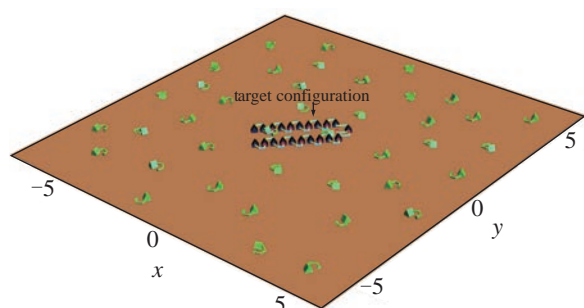
图 9 两种整体构形间的变换重构  
Fig.9 Reconfiguration between two group configurations

下面考虑第 2 种重构情况: 环境中有 40 个独立模块 (显示为绿色以区别于目标构形中的模块), 其状态为随机生成, 而目标构形由 21 个模块组成, 如图 10(a) 所示。为实现重构, 需要从独立模块中选择 21 个模块组成目标构形。各模块到其子目标的运动状态如图 10(b) 所示。

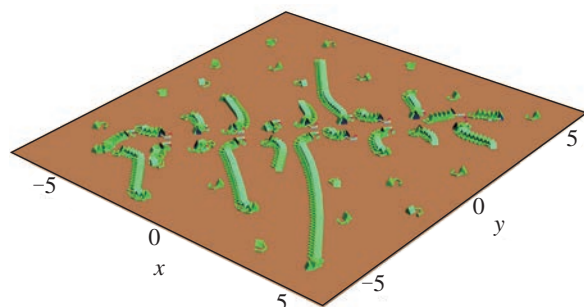
为验证方法的复杂度, 以包含不同模块数 (2 ~ 21) 的串形构形为目标构形进行仿真 (初始模块与目标构形的模块数相同)。每种情况运行 5 次并计算其平均运行时间。重构规划算法的计算分为两



部分, 第一部分求解 HJB 方程, 第二部分计算模块的避撞轨迹. HJB 方程求解的计算量与模块数量无关, 其只取决于离散网格的数量. 仿真采用的网格各维度离散数量为 100. 在计算环境为 Windows 7 OS、2.5 GHz CPU、4 GB RAM 时的平均计算时间为 102 s; 计算模块的避撞轨迹主要取决于 ORCA 算法, 其最坏时间复杂度为  $O(n)$ , 实际计算时间如图 11 所示. 可以看到实际运行时间与模块数量  $n$  基本呈线性关系.



(a) 初始模块以及目标构形



(b) 各模块的自动状态

图 10 独立模块的重构

Fig.10 Reconfiguration of independent modules

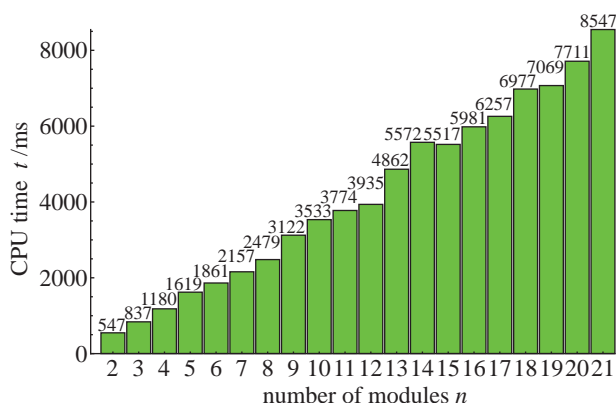


图 11 计算避撞轨迹所需的时间

Fig.11 Time used for the computation of the collision-free trajectories

## 7 结论 (Conclusion)

重构规划是实现模块化机器人自主控制和环境适应能力的重要环节. 计算最优的重构方案面临组

合爆炸问题, 传统方法只能处理模块数量较少的情况. 本文提出的重构规划方法通过求解最优控制问题得到各模块到目标构形的最优目标和最优轨迹, 随后利用改进的避撞方法得到各模块的无碰撞运动轨迹. 其总体计算复杂度与模块数量呈线性关系, 因此可用于模块数量大的重构任务. 虽然最优控制问题的计算次数与模块数无关, 但由于求解动态规划依赖数值方法, 因此对于高维的状态空间计算量较大. 对于简单的重构任务, 传统方法的效率可能更高. 下一步尝试将其推广到机器人受到更复杂的约束的情况, 例如环境中存在障碍物以及考虑动力学约束.

## 参考文献 (References)

- [1] 曹燕军, 葛为民, 张华瑾. 一种新型模块化自重构机器人结构设计与仿真研究[J]. 机器人, 2013, 35(5): 568-575, 606.  
Cao Y J, Ge W M, Zhang H J. Structure design and simulation analysis of an innovative modular self-reconfigurable robot - 360bot[J]. Robot, 2013, 35(5): 568-575,606.
- [2] Yim M, Shen W M, Salemi B, et al. Modular self-reconfigurable robot systems[J]. IEEE Robotics & Automation Magazine, 2007, 14(1): 43-52.
- [3] Bhat P, Kuffner J, Goldstein S, et al. Hierarchical motion planning for self-reconfigurable modular robots[C]//IEEE/RSJ International Conference on Intelligent Robots and Systems. Piscataway, USA: IEEE, 2006: 886-891.
- [4] Vassilvitskii S, Kubica J, Rieffel E, et al. On the general reconfiguration problem for expanding cube style modular robots [C]//IEEE International Conference on Robotics and Automation. Piscataway, USA: IEEE, 2002: 801-808.
- [5] Yoshida E, Murata S, Kamimura A, et al. A motion planning method for a self-reconfigurable modular robot[C]//IEEE/RSJ International Conference on Intelligent Robots and Systems. Piscataway, USA: IEEE, 2001: 590-597.
- [6] Dumitrescu A, Suzuki I, Yamashita M. Motion planning for metamorphic systems: Feasibility, decidability, and distributed reconfiguration[J]. IEEE Transactions on Robotics and Automation, 2004, 20(3): 409-418.
- [7] Shen W M, Salemi B, Will P, et al. Hormone-inspired adaptive communication and distributed control for CONRO self-reconfigurable robots[J]. IEEE Transactions on Robotics and Automation, 2002, 18(5): 700-712.
- [8] He X Y, Ma S G, Li B, et al. Study on turn motion of child rovers of a reconfigurable planetary rover system[C]//IEEE/RSJ International Conference on Intelligent Robots and Systems. Piscataway, USA: IEEE, 2005: 1407-1412.
- [9] Hu Y N, Ma S G, Li B, et al. Reconfiguration planning for wheel-manipulator robots[C]//IEEE International Conference on Cyber Technology in Automation, Control and Intelligent Systems. Piscataway, USA: IEEE, 2015: 529-534.
- [10] Osher S, Fedkiw R. Level set methods and dynamic implicit surfaces applied mathematical sciences[M]. New York, USA: Springer, 2003.



