中国科学: 信息科学 2018 年 第 48 卷 第 2 期: 143–176
SCIENTIA SINICA Informationis
《中国科学》杂志社
SCIENCE CHINA PRESS
评述

自重构模块化机器人路径规划方法综述

刘永进[1,2,†*], 余旻婧[1†], 叶子鹏[1], 王昌凌[3]

1. 清华大学计算机科学与技术系, 北京 100084, 中国
2. 清华信息科学与技术国家实验室, 北京 100084, 中国
3. Department of Design Engineering & TU Delft Robotics Institute, Delft University of Technology, Delft 2600 AA, The Netherlands
* 通信作者. E-mail: liuyongjin@tsinghua.edu.cn
† 同等贡献

收稿日期: 2017–07–12; 接受日期: 2017–09–25; 网络出版日期: 2018–01–24

国家重点研发计划 (批准号: 2016YFB1001200) 和国家自然科学基金创新研究群体项目 (批准号: 61521002) 资助

摘要 自重构模块化机器人指由一系列通用模块组装而成、能够根据所处环境或任务的变化依靠模块间的相互通信和自主运动重组为另一种适应新环境、新任务构型的机器人, 对它的研究近年来已成为学术界和工业界关注的热点. 模块化机器人的自重构过程在数学建模上, 可以归约到在其所能表达的构型空间中的两个构型之间找到一条满足某种条件的最优路径. 本文围绕自重构模块化机器人的路径规划方法, 从单个模块的运动方式、实现运动所需的模块硬件结构、多个模块整体间的组织方式、构型空间表示方法以及路径规划算法这 5 个层次, 对国际上已有代表性工作进行总结分析, 并在此基础上指出现有工作中存在的问题与挑战, 提出可能的发展方向并对以后的工作作出展望, 为研究人员系统性地了解自重构模块化机器人路径规划方法的现状以及开展后续研究提供了思路.

关键词 模块化机器人, 自重构, 路径规划, NP 完全问题, 近似算法, 计算几何

1 引言

在全球制造模式面临变革以及以数字化、智慧化制造与新型材料的应用为标志的第三次工业革命的浪潮下, 机器人技术与产业得到了广泛的关注. 美国在 2011 年由国家科学基金会牵头实施了国家机器人计划, 2012 年美国国家科学基金会、美国国家航空航天局、美国国立卫生研究院和美国农业部 4 家联邦机构联合宣布资助大学研究人员开展机器人研究. 欧盟于 2014 年启动了民用机器人研发计划 SPARC, 同 euRobotics 协会下属 180 多家公司和研发机构将共同投资超过 28 亿欧元研发民用机器人. 我国国家自然科学基金委员会也在 2016~2017 连续两年发布了共融机器人基础理论与关键

引用格式: 刘永进, 余旻婧, 叶子鹏, 等. 自重构模块化机器人路径规划方法综述. 中国科学: 信息科学, 2018, 48: 143–176, doi: 10.1360/N112017-00154
Liu Y J, Yu M J, Ye Z P, et al. Path planning for self-reconfigurable modular robots: a survey (in Chinese). Sci Sin Inform, 2018, 48: 143–176, doi: 10.1360/N112017-00154

© 2018《中国科学》杂志社

技术研究重大研究计划项目指南. 目前, 随着微处理器、传感器等核心硬件和大规模并行计算等核心软件的飞速发展, 机器人技术正处于变革期的临界点, 有望出现井喷式的增长.

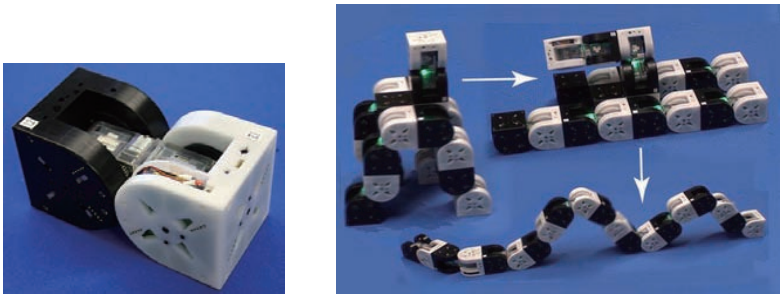
现有的机器人从组织结构上可以分为两种类型: 固定构型机器人和模块化机器人. 固定构型机器人指为特定的工作 (如高精度的工业生产、大量的重复性工业作业等) 设计、具有固定结构的机器人. 在工作阶段, 固定构型机器人的结构不再发生改变. 与之相反, 模块化机器人可以按照环境和任务的不同自适应地改变构型. 这里, 模块化是指在对机器人的目标功能分析后, 将整体结构分解并设计生产出一系列通用模块或标准模块. 将机器人整体分解成通用模块进行研究、生产, 可以降低设计、调试和维护的成本和难度. 与此同时, 相比于固定构型机器人, 模块化机器人通过增减某些模块或者对现有模块进行重新组合, 可以由当前构型迅速变换为另一种适应新环境、新任务的构型. 近年来, 模块化机器人已成为学术界和工业界关注的热点.

模块化机器人又可以进一步分为两种类型: (1) 由不同尺寸和功能的模块组装而成的模块化机器人; (2) 由相同模块组装而成的模块化机器人, 其中每一个模块都是一个封装了特定功能的物理独立单元. 前一种模块化机器人通常可分解分为关节模块、连杆模块、末端执行器模块等类型, 不同类型的模块完成的功能不同, 不能相互替代. 这种类型的机器人通过多种模块的不同组装形式构成不同的形态, 可以完成诸如喷漆、焊接、爬壁等不同任务, 已经较为广泛地应用在工业上. 后一种模块化机器人由于每个模块完全相同, 在重构过程中任意两个模块都能相互替换. 根据是否能够自动完成重构, 后一种模块化机器人可进一步分为非自重构和自重构两种. 非自重构机器人是指需要借助外力才能实现重构的机器人, 而自重构机器人可以自主控制模块, 根据所处环境或任务的变化通过模块间的相互通信和自主运动重组到目标构型. 对自重构机器人而言, 其每个模块均可以视为一个基本的机器人单元, 搭载着驱动器、传感器、电源、连接元件和通信元件, 具有独立的处理能力. 本文中所讨论的模块化机器人, 特指这类由相同模块组成的自重构模块化机器人.

与其他类型的机器人相比, 自重构模块化机器人具有以下独特性质:

- 鲁棒性. 因为该类型机器人由大量相同的模块组成, 所以具有一定程度的冗余. 这种冗余使得模块化机器人具有很高的鲁棒性. 对传统机器人而言, 某个部件或软件出现故障, 往往会导致整个机器人系统的崩溃, 维护修理成本非常高. 但模块化机器人由相同的模块组成, 用冗余的模块替换发生故障的模块就能够快速地实现自我修复.
- 多功能性. 在诸如深水作业或者外太空探险等应用场景中, 目标任务多样, 但每次行动所能携带的装备非常有限. 模块化机器人 (例如水下 Hydra 机器人 [1]) 可以利用相同的模块, 通过自重构方式变换相应的工作构型, 从而能够完成不同任务.
- 适应性. 其自重构功能和自我修复功能决定了其具有很强的适应性, 在一些复杂的工作场景中 (比如携带摄像机进入震后灾区寻找幸存的生命时), 模块化机器人可以变换最佳的运动方式适应不同的地形. 举例来说, 变换成蛇形运动以便通过空洞缝隙, 变换成轮式运动通过沼泽地区, 变换成多足运动通过崎岖不同的山地等 (图 1). 目前模块化机器人已经实现的运动方式包括行走、攀爬、滚动、蛇形运动等.

这些特性使得自重构模块化机器人在一些新兴的应用环境中具有广阔的应用前景. 从二十世纪九十年代以来, 这种模块化机器人在学术界受到越来越多的关注. 设计和研发自重构模块化机器人涉及到多个学科领域, 包括计算机、电子、自动化、机械和材料学科等. 本文从计算机学科的角度, 对自重构模块化机器人设计中的一个关键要素 “自重构过程中的路径规划” 进行总结分析, 并总结阐述其研究进展. 自重构过程中的路径规划不仅与算法设计密切相关, 还与模块的硬件实现和模块间的运动方式有着密切的关联.



(a) (b)

图 1 (网络版彩图) 日本产业技术综合研究所的 M-TRAN III 模块化机器人 [2]. 图片由 Haruhisa Kurokawa 教授提供

Figure 1 (Color online) The M-TRAN III modular robot in national institute of advanced industrial science and technology (AIST), Japan [2]. (a) A single M-TRAN III module; (b) the reconfiguration from a quadruped robot to snake robot. Both robots are made up of the same number of modules. Photos are courtesy of professor Haruhisa Kurokawa

本文围绕自重构模块化机器人路径规划方法, 从单个模块的运动方式、实现运动所需的模块硬件结构、多个模块整体间的组织方式、构型空间表示方法以及路径规划算法 5 个层次进行系统性的介绍, 在此基础上说明现有工作中存在的问题与挑战, 提出可能的发展方向并对以后的工作作出展望. 本文的组织如下: 文章分为 8 个部分, 其中第 1 节为引言, 第 2 节介绍自重构机器人模块可能具有的不同运动方式, 第 3 节介绍自重构化机器人模块的主要硬件结构与作用, 第 4 节介绍自重构模块化机器人模块整体组织形式, 第 5 节介绍模块化机器人的形状和构型的定义并给出构型空间的下界, 第 6 节重点介绍现有自重构机器人的路径规划算法, 第 7 节总结现有研究中存在的问题与挑战, 并提出自重构机器人可能的发展方向, 第 8 节对全文进行总结.

2 单个模块的运动方式

模块化机器人的自重构路径规划方法与单个模块所能完成的运动密切相关, 不同的运动方式意味着不同的物理约束, 进而对应不同的路径规划算法. 概括性来说, 单个模块的运动自由度越大, 路径规划算法的设计越灵活. 这一节将针对现有模块的运动方式展开介绍.

2.1 单个模块的几何形状

自重构模块化机器人根据模块中是否具有多个能够互相运动的子模块, 主要可分为双子型模块和单一型模块两类.

• 双子型模块. 记模块的几何外形为 Φ. 双子型模块在几何外形上可以分解为两个不互相重叠的基本几何单元 Φ_1 和 Φ_2, 即 $\Phi = \Phi_1 \bigcup \Phi_2$ 且 $\Phi_1 \bigcap \Phi_2 = \emptyset$. 每个几何单元对应一个子模块, 子模块间通过连接件相连, 不可分离但可以相互运动. 由于两个子模块间保持固定连接, 不需要类似挂钩等复杂的连接器. 根据不同的运动方式, 两个子模块可以是相同的 (如 University of California Davis 提出的 iMobot [3]), 也可以是不同的 (日本产业技术综合研究所的 M-TRAN 系列 [2,4,5] 和 University of Southern California 的 SuperBot [6]). 双子型模块中两个子模块间的相互运动通常包括中心铰接型 (2.2 小节) 和边铰接型 (2.3 小节). 图 2 中给出了两种双子型模块的例子.

• 单一型模块. 该类型模块一般作为一个整体做刚体运动. 由这种类型模块组成的模块化机器人, 每两个模块间都可以相互分离, 需要时也可以相互连接. 这种类型模块的典型代表包括 Massachusetts

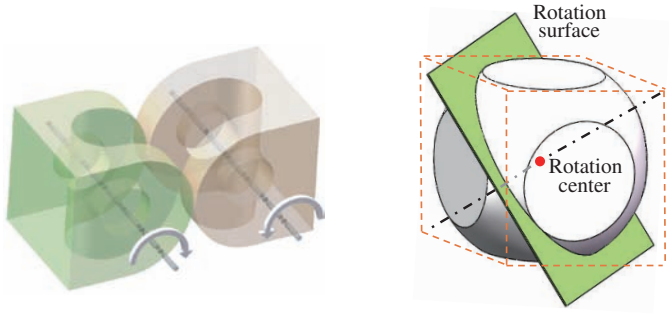


图 2 (网络版彩图) 双子型模块

Figure 2 (Color online) Double-unit modules. (a) M-TRAN III module [2] is an edge-hinged double-unit module (Subsection 2.3). It consists of two (i.e., female and male) semi-cylindrical cubes. Two cubes are connected by a linker. Each cube can independently rotate with respect to the linker and the rotation angle is ranged from $-90°$ to $90°$; (b) the CellRobot in KEYi technology incorporated consists of two symmetrically hemispherical units. By rotating around the center of the module, each unit can relatively move with respect to each other (Subsection 2.2)

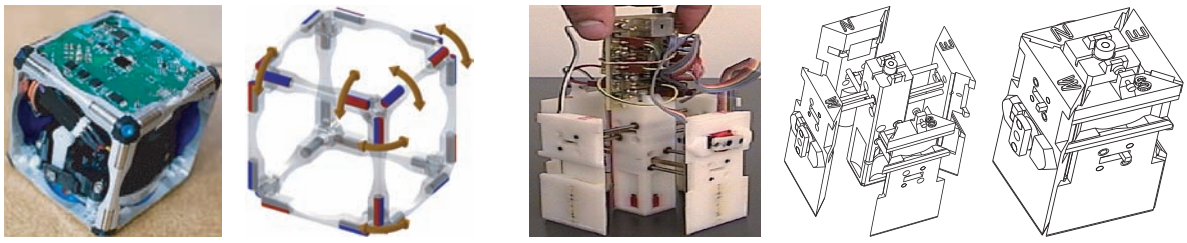


图 3 (网络版彩图) 单一型模块. 图片由 Daniela Rus 教授和 John Romanishin 博士提供

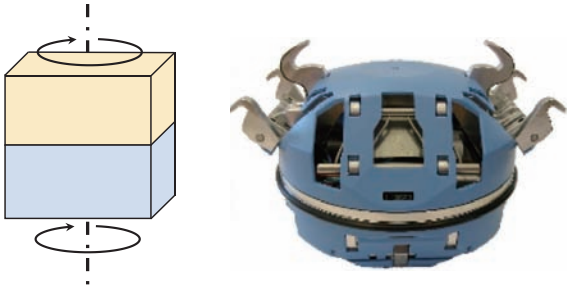
Figure 3 (Color online) Single-unit modules. (a) M-Blocks [4] modules locomote by pivoting about edges, and the movements are driven by a torque generated by rapidly decelerating an internal flywheel. The permanent magnets embedded in edges are free to rotate to allow 2–4 modules to form structural magnetic bonds. (b) crystalline [5] modules move by expanding and contracting and the length of the side when expanded is two times that of contraction. Photos are courtesy of professor Daniela Rus and doctor John Romanishin

Institute of Technology (MIT) 提出的 M-Blocks [7] 和 Dartmouth College 提出的 Crystalline [8]. 模块与模块间的常见运动包括绕边旋转型 (2.4 小节) 和伸缩型 (2.5 小节). 图 3 给出了两种单一型模块的例子.

双子型模块的两个子模块间可以相互旋转, 根据子模块间是否具有外部可见的连接件又可以进一步分为两类. 在日本产业技术综合研究所的 M-TRAN 系列 [2, 4, 5] 这一经典的、具有外部可见连接件的双子型模块中, 其子模块采用半圆柱立方体结构, 如图 2(a) 所示, 这种结构可以看作半圆柱体和长方体的结合体, 半圆柱体形用来构造圆弧面, 长方体一侧主要用于实现模块间的相互连接, 与此同时长方体的规则形状有利于实现多个模块的稳定堆砌. 北京可以科技有限公司的商业化 CellRobot 模块[1)] 是另外一种典型的双子型模块. 如图 2(b) 所示, 每个 CellRobot 模块分为两个半球形子模块, 两个子模块之间能够相互旋转, 旋转接触面为平面, 两个子模块间没有外部可见的连接件. 每个半球子模块上有 4 个接口用于和其他模块连接, 半球旋转时可以带动与其连接的其他模块一起旋转.

单一型模块的几何外形通常为立方体型. 美国 MIT 提出的 M-Blocks 模块是一种典型的绕边旋转

1) "CellRobot". http://www.keyirobot.com/.



(a) (b)

图 4 (网络版彩图) 中心铰接类型 I. 图片由 Henrik Hautop Lund 教授提供

Figure 4 (Color online) Central-point-hinged type I. (a) The rotate axis is the line connecting the centers of two faces opposite with each other; (b) Atron [10] proposed by University of Southern Denmark modules are in this type. Photos are courtesy of professor Henrik Hautop Lund



(a) (b) (c)

图 5 (网络版彩图) 中心铰接类型 II. 图片由 Swiss Federal Institute of Technology in Lausanne 生物机器人实验室提供

Figure 5 (Color online) Central-point-hinged type II. (a) The rotation axis is the diagonal of bounding box. (b) Molecube modules [11] proposed by Cornell University cannot move relative to each other. (b) Roombots [12, 13] modules proposed by Swiss Federal Institute of Technology in Lausanne (EPFL) are able to move relative to each other. Photos are courtesy of BioRob, EPFL

型单一型模块. 如图 3(a) 所示, M-Blocks 模块的每条边上安装有柱形磁体, 通过柱形磁体的自由转动来调节磁极位置, 保证旋转过程中两模块间能保持边连接. 美国 Dartmouth College 提出的 Crystalline [9] 是典型的伸缩型单一型模块. 如图 3(b) 所示, Crystallline 在二维方向上可以伸展和收缩, 伸展后边长为收缩的两倍, 通过模块自身的伸缩可以带动所连接的相邻模块进行移动.

模块不同的几何外形使得模块间具有不同相对运动形式. 对于模块化机器人而言, 主要可以分为以下 4 种运动方式: 中心铰接型 (2.2 小节)、边铰接型 (2.3 小节)、绕边旋转型 (2.4 小节) 和伸缩型 (2.5 小节).

2.2 中心铰接型

中心铰接型是双子型模块的一种常见运动方式. 该类型的模块由相互对称的两部分组成, 以模块的中心为旋转中心, 子模块间能发生相对旋转. 模块间通过机械挂钩的伸出与收回实现连接或分离. 该类型模块化机器人的自重构过程由每个模块内部两子模块间的相对旋转实现. 该类型的机器人又可以根据旋转轴的不同进一步分为两种类型.

- 类型 I. 旋转轴为立方体相对的两个平行面中心连线. 如图 4(a) 所示. 单个模块沿着离立方体上

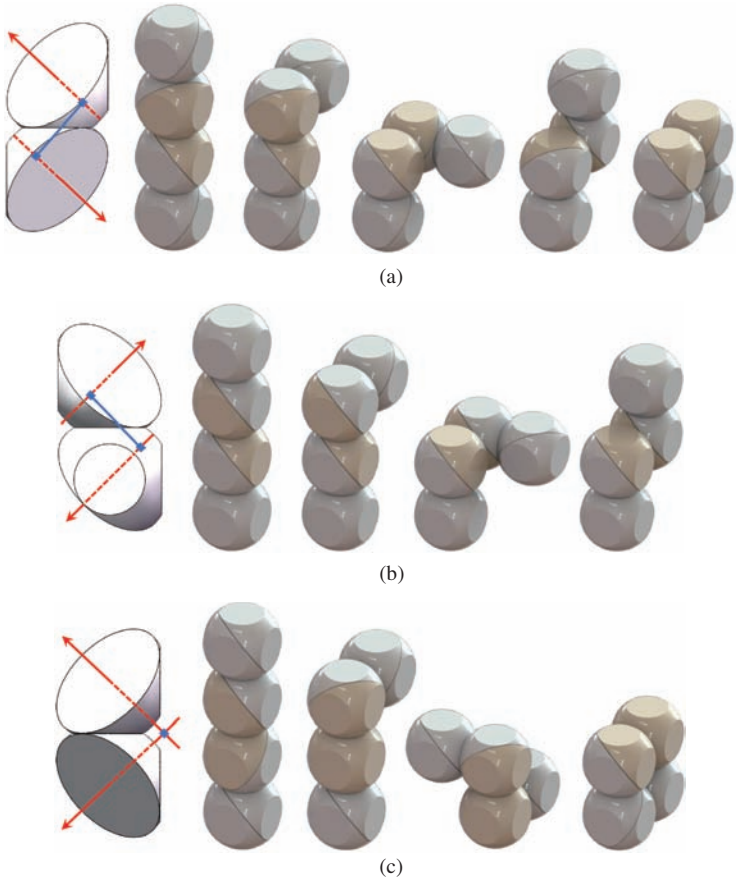


图 6 (网络版彩图) 对于中心铰接型模块 (类型 II), 根据两个相邻模块间各自内部旋转轴的不同相对位置关系, 可以形成两轴斜交、两轴平行和两轴正交 3 种几何构型 [12]

Figure 6 (Color online) Possible grid-reconfigurations with two Roombots because of the different relationship between rotate axes [12]. (a) Skew: 5 options; (b) parallel, 4 options; (c) orthogonal, 4 options

下两个平面等距的平面被分成了两个子模块, 相互旋转的接触面即为该平面. 典型例子包括 University of Southern Denmark 提出的 ATRON 模块 [10], 如图 4(b) 所示.

• 类型 II. 旋转轴为立方体的体对角线. 如图 5(a) 所示. 模块沿对角线平分面分成了两个对称的子模块, 相互旋转的接触面即立方体对角线的平分面. 根据两个相邻的模块间能否发生相对旋转, 又可以进一步分成两种类型. 第 1 类是任意两个相邻的模块间均不能发生相对旋转运动, 如 Cornell University 提出的 Molecube 模块 [11] (图 5(b)) 和北京可以科技有限公司的 CellRobot 模块 (图 2(b)), 这类机器人的自重构只能通过单个模块内部子模块间的相对旋转带动与之连接的模块进行运动来实现; 第 2 类是两个相邻的模块间可以发生相对旋转运动, 如 Swiss Federal Institute of Technology in Lausanne (EPFL) 提出的 Roombots [12,13], 如图 5(c) 所示, 这种类型比第 1 类多了一个旋转自由度, 其旋转轴如图中蓝色线段所示. 第 2 种类型的模块根据相邻模块各自内部旋转轴间的相对位置关系不同, 可以构成不同的形状类型, 如图 6 所示, 两轴斜交、平行和正交时, 所能形成的组合构型数分别 5 种、4 种和 4 种.

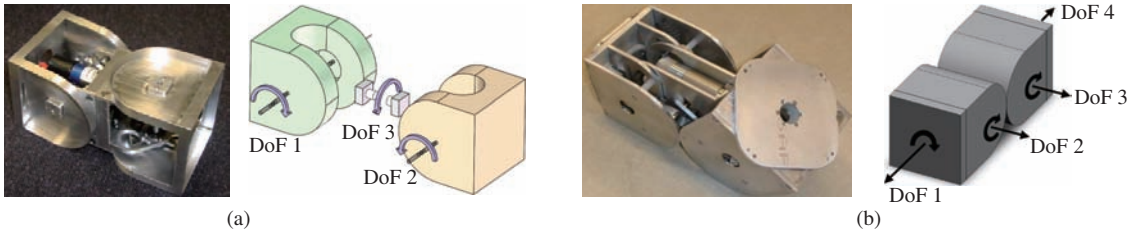


图 7 (网络版彩图) 双子模块类型中的边铰接型. 图片由 Wei-Min Shen 教授提供

Figure 7 (Color online) The edge-hinged type. (a) The sub-modules of SuperBot [6] can rotate around the link (DoF 1 and 2) and with each other (DoF 3); (b) the sub-modules of iMobot [3] can rotate around the link (DoF 2 and 3), and the modules can rotate with each other (DoF 1 and 4). Photos are courtesy of professor Wei-Min Shen

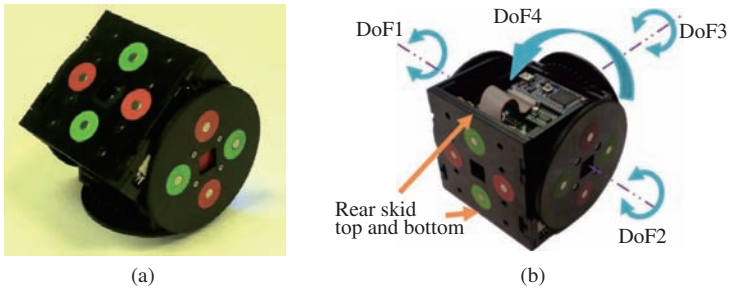


图 8 (网络版彩图) SMORES 模块及其运动方式 [14]. SMORES 模块的具有 4 个旋转自由度, 除了两侧的轮子分别具有旋转功能外 (自由度 1 和 2), 连接面上也构建有可旋转的圆盘, 通过磁铁连接后, 两个模块可形成一个整体, 将轮子的中心连线视为旋转轴即可实现与双子模块类型类似的旋转运动 (自由度 3 和 4). 图片由 Mark Yim 教授提供

Figure 8 (Color online) (a) SMORES modules and (b) their motion modes [14]. Each module has two wheels (DoF 1 and 2), and the connected face is a rotatable disk. Once the connection of two modules is made, there are DoF 3 and 4. Photos are courtesy of professor Mark Yim

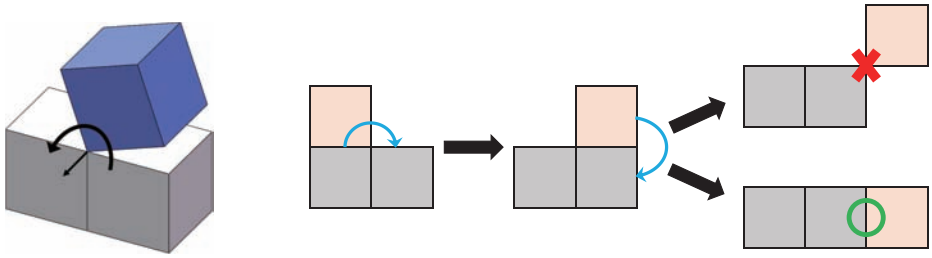
2.3 边铰接型

边铰接型同时存在双子模块和单模块类型中. 它多见于双子模块类型中, 该类型中两个子模块由连接件相连, 两个子模块均能与连接件做相对旋转运动, 运动范围在 $-90°$ 到 $+90°$ 之间. 典型的边铰接型模块包括 M-TRAN (图 2(a))、SuperBot (图 7(a))、iMobot (图 7(b)) 等. 这些模块的不同之处在于, 如图 2(a) 所示, M-TRAN 系列模块 [2,4,5] 只有两个运动自由度, 即两个子模块分别与连接件做相对旋转运动. SuperBot 的子模块间增加了一个旋转自由度, 如图 7(b) 中所示的自由度 3, 而 iMobot 在相邻的两个模块间增加了旋转自由度, 如图 7(b) 所示的旋转轴 1 和 2.

一种单模块类型的边铰接型机器人为 University of Pennsylvania 提出的 SMORES [14], 如图 8 所示, SMORES 模块的具有 4 个旋转自由度, 除了两侧的轮子分别具有旋转功能外 (自由度 1 和 2), 连接面上也构建有可旋转的圆盘, 通过磁铁连接后, 两个模块可形成一个整体, 将轮子的中心连线视为旋转轴即可实现与双子模块类型类似的旋转运动 (自由度 3 和 4). SMORES 模块化机器人的自重构过程由模块间的相对旋转运动实现.

2.4 绕边旋转型

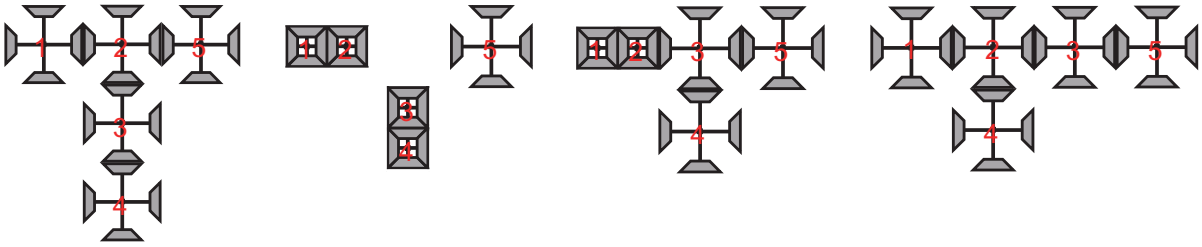
该类型的每一个模块通常以一个整体做刚体运动. 机器人自重构过程中, 两个相邻模块以一条相



(a) (b)

图 9 (网络版彩图) 绕边旋转型模块的工作方式

Figure 9 (Color online) Pivoting modules. (a) The modules move by pivoting about edges. (b) Modules in this type cannot control the angle of rotation accurately because of lacking in connector. They cannot stop at the position showed in red cross and the only way to stop pivoting is other modules block the way, as show in green circle



(a) (b) (c) (d)

图 10 (网络版彩图) 伸缩型模块化机器人的构型变换原理

Figure 10 (Color online) A sketch map of reconfiguration by expanding and contracting. Modules 1, 2, 3, 4 of (a) were contracted, and the configuration is transformed into (b). Then modules 3 and 4 were expanded, transformed into (c). Then modules 1 and 2 were expanded, transformed into (d). Because the modules are all the same, the reconfiguration between (a) to (d) can be regarded as that module 4 moved to the position of module 5

邻边作为旋转轴, 一个模块绕着旋转轴相对另一模块做旋转运动, 如图 9(a) 所示. 图 3(a) 中所示的 M-Blocks 模块为一种典型的绕边旋转型模块. 两个模块间的相对旋转运动的产生机理为, M-Blocks 模块借助每个模块内部飞轮的加速或减速, 将不同方向的动量传至外壳, 驱使模块产生运动. 由于模块边线内部具有磁铁, 因此模块翻转过程中仍能与相邻的模块保持边连接, 从而使得运动以绕着相邻边旋转的方式发生. 绕边旋转型模块的旋转运动比边铰接型模块灵活, 但由于模块之间没有机械连接机构, 因此无法准确地控制翻转角度, 只能通过与目标位置相邻的模块的阻拦使其停止翻转, 如图 9(b) 所示. 2015 年 MIT 在 M-Blocks 模块的基础上提出了 3D M-Blocks [15], 相比于老版模块只能实现一个方向上的翻转, 新版的 3D M-Blocks 能实现 6 个方向上的翻转.

2.5 伸缩型

每个伸缩型模块都具有空间伸展和收缩的功能. 比如图 3(b) 中所示的 Crystalline 伸缩型模块 [8], 伸展时的每条边长度为收缩时的两倍. 伸缩型模块的移动是由与之相连的模块的伸缩操作完成的. 相比于之前提到的几种类型的模块化机器人只能在模型的外表面翻转移动, 伸缩型模块化机器人能够在模型内部通过不同模块间的伸展和收缩实现模块移动, 完成构型变换, 如图 10 所示.

图 3(b) 中所示的美国 Dartmouth College 提出的 Crystalline [8], 可以视为是一种二维的伸缩型模块, 因为它只有 x 和 y 两个维度上的伸缩操作. 美国 Palo Alto 研究中心提出的 Telecubes 模块 [16], 则



(a) (b) (c)

图 11 (网络版彩图) Telecubes 模块 [16] 的三维伸缩操作. 图片由 Mark Yim 教授提供

Figure 11 (Color online) The 3D expand and contract operation of Telecubes [16]. (a) CAD model of Telecubes module; (b) the physical prototype of Telecubes module expanded; (c) the physical prototype of Telecubes module contracted. Photos are courtesy of professor Mark Yim

是一种三维的伸缩型模块, 它可以在 x, y 和 z 3 个维度上进行伸缩操作, 如图 11 所示.

3 模块的硬件结构

自重构模块化机器人的每个模块均可以视为一个基本的机器人单元, 搭载着驱动器、连接器、计算与通信元件、传感器和电源, 具有独立的处理能力. 第 2 节中介绍的单个模块的不同运动方式正是由于其具有不同的硬件结构. 本节着重介绍这些模块的硬件部件.

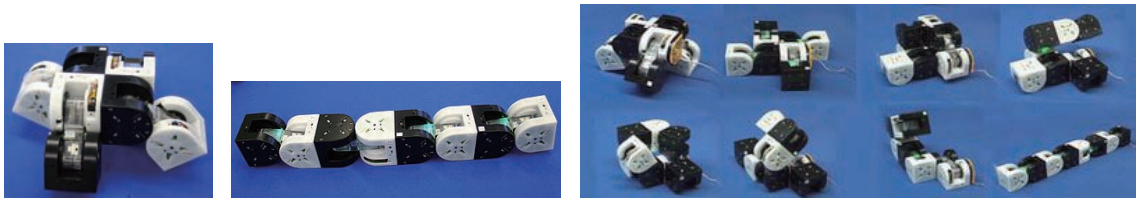
3.1 驱动器

驱动器是模块的动力单元, 主要具有以下两个功能.

(1) 驱动单个模块产生运动, 并使得模块化机器人完成自重构过程; 多数自重构机器人的单个模块具有自主移动的能力, 通过感知周围环境, 寻找其他模块并与之连接. 在一些构型下, 单个模块移动时可以不借助其他模块的帮助, 这样就降低了自重构过程的难度. 模块自主移动的能力与水平高低对模块设计的复杂度有很大影响, 较高的自主移动能力需要更多的驱动器, 因此设计过程中通常需要在模块结构复杂度和自主移动能力高低两者之间进行权衡. 此外, 在模块化机器人的自重构过程中 (图 12(c)), 组成机器人的多个模块, 按照一定顺序, 通过驱动器驱动单个模块逐个进行弯曲、扭转、收缩或扩张等操作, 移动到某个预定目标位置并改变模块间的连接关系. 更多的关于自重构过程的详细内容见第 5 节.

(2) 驱动整个模块化机器人产生运动: 在一些自重构机器人中, 有的单个模块可以凭借自身动力驱动整个机器人, 比如在机器人底部的 Roombots 模块通过自身轮子的滚动, 驱使整个模块化机器人向某个方向进行运动 (图 13). 另外一些自重构机器人只有在多个模块组成一个整体、并且在多个模块协调运动时, 机器人才能产生期望的运动. 比如由 4 个 M-TRAN 模块 [2] 组成的四足机器人和蛇形机器人, 都需要每个模块按照一定的模块之间协调运动规律进行弯曲或扭转操作, 才能完成四足行走或蛇型转弯等运动 (图 12(a) 和 (b)).

驱动器类型的选择, 取决于它所需要完成的功能, 一般考虑尺寸大小、产生力矩的大小、控制难易程度和成本等因素. 表 1 中列出了一些常用驱动器类型的分类. 除了常见的电机驱动器外, 形状记忆



(a) (b) (c)

图 12 (网络版彩图) 由 4 个 M-TRAN 模块 [2] 组成的 (a) 四足构型和 (b) 蛇型构型. 需要每个模块按照一定的模块之间协调运动规律进行弯曲或扭转操作, 才能完成整个机器人的四足行走或蛇型转弯等运动. (c) 两个构型之间的自重构过程. 图片由 Haruhisa Kurokawa 教授提供

Figure 12 (Color online) The (a) four-legged walkers and (b) snake forms assembled by four M-TRAN modules [2]. Each module should bend or rotate by the rules to walk or crawl. (c) The automatic metamorphosis between configurations. Photos are courtesy of professor Haruhisa Kurokawa



(a) (b)

图 13 (网络版彩图) (a) Roombots 模块采用滚动运动时, 可以搬运家具; (b) 由多个相同 Roombots 模块 [12, 13] 组成的椅子构型, 最底部与地面接触的模块进行滚动运动时, 可以带到整个构型向某个方向进行运动. 图片由 EPFL 生物机器人实验室提供

Figure 13 (Color online) (a) The rolling Roombots modules can be used to moving furniture; (b) a Roombots [12, 13] chair. The rolling bottom modules enable the chair to move. Photos are courtesy of BioRob, EPFL

表 1 驱动器类型的比较 [20]

Table 1 Comparison of different actuators used in reconfigurable modular robots [20]

Type	Torque	Size	Control complexity	Cost
Brushless DC motor	Large	Medium	Complex	High
Brush DC motor	Medium	Large	Easy	High
DC stepping motor	Medium	Large	Moderate	Moderate
Shape memory alloy	Small	Small	Complex	Low

合金在不同的温度下能产生的膨胀或收缩力, 由于形状记忆合金占用的空间非常小, 所以也被用于模块化机器人中 [17~19].

3.2 连接器

每个模块内部都有连接器, 它用于实现模块之间的连接与分离, 模块化机器人的自重构即通过改变模块间的连接关系达到改变机器人构型的目的. 此外, 在双子型模块中, 有时会有一个连接件. 连接件和这里讨论的连接器是不同的两个部件: 连接件用于连接两个子模块, 而连接器用于连接不同的

模块.

模块中的连接器通常具有以下特性 [20].

• 体积小. 在三维自重构的模块化机器人中, 每个模块通常有 6∼8 个连接器, 因此连接器的大小对模块体积有很大影响, 连接器越小, 模块的体积也能相应减小.

• 连接或分离迅速. 模块化机器人的自重构可以分为 3 个步骤: 分离 (单个或数个模块与相邻模块分离)、移动 (单个或数个模块的移动)、再连接 (模块达到新位置后重新与相邻模块再连接). 提高模块间连接和分离的速度, 可以加速自重构过程.

• 连接强度高. 模块化机器人的整体结构强度很大程度上取决于连接器的连接强度, 如果连接器非常脆弱, 则可能在高应力下断裂使得机器人中的模块分离.

• 鲁棒性和容错性能好. 在自重构过程中, 大量连接器需要反复的连接和分离, 因此, 连接器操作的鲁棒性非常重要. 此外由于模块移动过程中的定位精度问题, 在连接过程中也要求模块间的连接和分离操作具有一定的容错能力.

连接器的硬件选择主要可以分为以下 4 种类型.

(1) 机械连接器. 挂钩是一种常用的机械连接器类型, 在 PolyBot 系列机器人 [21,22] 中, 由公连接器的挂钩通过驱动插入母连接器的凹槽中, 闩结构落至相应位置防止挂钩脱出, 分离时打开闩结构, 释放挂钩; 在 M-TRAN 系列机器人 [2,4,5] 中, 则由公连接器的挂钩直接钩住母连接器. University of Southern California 提出的 SINGO 连接器 [23] 运用能自主分离的两个相互连接的挂钩, 解决了一旦公连接器失效则无法分离的问题. 由于连接和分离过程需要将挂钩放置于正确的位置和朝向, 因此对精度的要求很高. 由于多数模块化机器人对连接器的强度要求较高, 所以多数连接器都采用机械结构. 虽然机械连接器强度高, 但随之带来的是复杂度高、所占空间大、放置难度大等问题.

(2) 磁性连接器. 最简单的磁性连接器是在两个需要连接的模块表面上放置永磁体的相反磁极, 这种方法较容易实现, 但分离时需要提供至少与磁力相等大小的力使两个模块分离. 一些模块化机器人系统中会采用电磁体, 通过控制电流使得模块间连接与分离. 由于磁力与模块间距离成反比, 仅仅依靠磁力来连接模块, 相对于机械连接器来说磁力的连接强度较弱, 但相比机械连接器, 磁性连接器所占体积比较小. 由 University of Pennsylvania 最近提出的 SMORES-EP 模块使用电磁体连接器 [24] 代替早期版本的 SMORES 中的永磁体连接器 [14].

(3) 静电力连接器. 工作原理与磁性连接器类似, 对两个需要连接的模块表面充电形成不同的磁极, 两者相互吸引形成连接 [25]. 与磁性连接器一样, 静电力连接器的连接强度较弱, 可以考虑用在弱重力或无重力环境中, 比如水下或太空等对连接强度要求不高的环境中.

(4) 尼龙搭扣连接器. 指在两个需要连接的模块表面上分别放置尼龙钩带和尼龙绒带, 通过施加轻压, 就能产生较大的扣合力和撕揭力. 相比其他几种连接器类型更加简单经济, 缺点是必须通过人工进行分离. YaMoR [26] 是典型的使用尼龙搭扣连接器的机器人.

3.3 计算和通信元件

由于每个模块都是一个具有独立功能的机器人单元, 所以模块内部都安装有计算和通信元件. 从简单的微控制器到 32 位复杂指令集微处理器都已经应用在自重构机器人模块中. 举例来说, CONRO 模块 [27] 使用了 1 个简单的 Basic Stamp 可编程式微控制器; Fractum 模块 [28,29] 使用了 1 个 ZILOG 公司制造的 8 位微处理器 Z80; Crystalline 模块 [8] 使用了 1 个 ATMEL 公司制造的 AT89C2051 单

片微处理器[2)]; M-TRAN II 模块[5] 使用了 3 个 PIC 精简指令集微控制器和 1 个 Echelon 公司制造的 TMPN3120 神经元芯片; M-TRAN III 模块[2] 使用了 1 个 32 位 HD64F7047 微处理器、1 个 16 位 HD64F3687 和 2 个 16 位 HD64F3694 微处理器, 均为 Renesas 公司制造; SuperBot 模块[6] 使用了 1 个 Atmel 公司制造的 8 位低功耗 CMOS 微处理器 MEGA128; ATRON 模块[10] 使用了 2 个 Atmel 公司的 MEGA128L 微处理器; Molecube 模块[11] 使用了 2 个 Atmel 公司制造的 8 位微处理器 ATMEGA16; Polypod 模块[30] 使用了 1 个美国 Motorola 公司制造的 32 位 68000 型中央处理器. 将来不排除有功能更加强大、可以安装小型机器人操作系统的微处理器单元被安装到自重构机器人模块中.

模块化机器人中的多个模块, 可以通过光学、电子或无线通信方式形成一个通信系统, 每个模块中的微处理器通过通信系统连接成为一个网络. 影响机器人运作性能的因素不仅包括单个微处理器的性能, 还包括通信系统的性能. 模块化机器人中的通信系统工作方式主要可以分为以下 5 种[20].

(1) 中心通信. 很多模块化机器人都采用中心通信方式, 在这种方式下, 每个模块中的驱动器直接与一台主机通过线缆或者以无线通信方式相连. 如 UBot[31, 32]、SMORES-EP 模块[33] 等组成的机器人由中央主机发送指令控制每个模块中的驱动器和连接器. 但这种通信方式的鲁棒性较差, 一旦主机崩溃, 整个系统都会崩溃.

(2) 亚符号通信. 对结构非常简单的模块化机器人而言, 每个模块可以通过感知相邻模块的行为或者环境的变化进行间接通信, 而不需要专用的通信硬件, 可以降低硬件结构的复杂度. 这种通信方式类似蚁群中每个蚂蚁通过感知局部邻域其他蚂蚁的活动, 如 EPFL 提出的 SWARM-BOTS[34], Harvard University 提出的 Swarm[35], University of Minnesota 提出的 MinDart[36] 等类似语言符号进行通信的方式, 称为亚符号通信.

(3) 全局通信. 是指所有模块间能直接通过共同的通信介质在整个构型间通信, 比如 CKBot[37] 中布置在整个模块化机器人当中的数据总线. 当分属不同部位的模块间需要紧密合作时, 这种不依赖物理距离的特性就十分重要. 全局通信方式的局限性在于, 因为通过全局总线传递的信号中不包含信号发送模块相对于接受模块的位置信息, 所以全局通信方式无法获取模块间的局部拓扑连接信息.

(4) 局部通信. 如果需要了解模块化机器人局部的模块间拓扑连接信息, 就需采用局部通信方式. 最常见的局部通信是相邻模块间的红外通信, 如 CONRO 模块[27]、ATRON 模块[38] 等, 这种方式不依赖于物理连接 (比如挂钩相连) 就能互相感知对方并进行通信. 此外, 也可以通过模块中连接器的电力连接实现局部通信, 这种方式能够增加通信带宽, 但缺点在于模块连接前无法通信, 也无法判断是否有其他的模块可供连接.

(5) 多模式通信. 在一些模块化机器人中同时使用了全局通信和局部通信这两种通信方式, 例如 M-TRAN 模块[4] 同时使用了基于红外的局部通信方式和基于总线的全局通信方式.

3.4 传感器

自重构模块化机器人的一个显著特点是其具有自适应性, 能根据环境或者任务的不同而灵活的改变构型. 只有装备传感器的模块化机器人才能灵敏的检测周围环境的变化或者交互式的接受指令实现自重构. 可以装载在模块上的传感器种类非常多, 常见的包括利用视觉传感器 (即微型相机) 判断是否具有障碍物, 利用温度和湿度传感器感知环境的温度和湿度变化, 利用扭力传感器防止驱动器损坏, 利用压电式传感器测量模块所受到的动态压力, 以及利用倾斜度传感器判断每个模块的朝向等. 单个

2) 包括 2k 字节可反复擦写的只读程序存储器, 128 字节的随机数据存储器, 通用 8 位中央处理器和 Flash 存储单元.

模块可以配备多种类型的传感器, 如北京航空航天大学提出的 Sambot 模块, 同时具有红外传感器、陀螺仪和加速度计, 用于给出机器人的姿态和速度信息 [39].

3.5 电源

电源用于给机器人中的每个模块供电. 根据实际需要可以选择自带电源或者外接电源. 如果机器人所处环境复杂, 应选择自带电源. 自带电源的机器人部分直接使用电池[3)], 优点是工艺简单, 但需要定时将电池取出充电或更换电池; 有部分机器人使用板载电源, 板载电源相较于直接使用电池复杂, 但充电过程较为简单, 如 Harvard University 提出的 Kilobot 机器人 [40], 充电时只需将两个引脚分别放置在电压差为 6 v 的两个导电线上即可. 如果机器人位置较为固定, 则可以选择使用外接电源通过线缆给模块供电, 如 University of Southern Denmark 提出的 Deformatron robot [41], 哈尔滨工业大学提出的 HitMSR [42] 等. 这种方式可以避免使用板载电源所需考虑的电池位置及充电时间等问题, 并且可以减轻模块重量, 减小模块设计的难度. 除此以外, 也有部分机器人只让少数模块携带电源, 通过模块间的连接机构传递电力为其他模块供电, 或者在充电过程中只连接少数模块与充电器相连, 再通过这些模块将电力传至整个机器人.

4 自重构机器人的模块组织形式

自重构机器人通过将多个相同的模块按照一定的连接顺序组织在一起形成构型来完成相应的任务. 不同的目标任务可以通过 3 种不同的途径来完成: (1) 构型变换, 即从一个初始构型 (比如蛇形) 变换为另一个目标构型 (比如多足形); (2) 运动, 即机器人保持构型不变, 通过多个模块间的协调运动 (比如改变两个模块间关节或连接处的角度) 使整个机器人产生期望的运动; (3) 操控物体, 即与待操作物体进行交互 (比如支撑或拾取). 以上这些途径中, 针对完成任务所需功能的不同, 自重构机器人的模块组织形式多种多样, 从模块连接的空间维度上可以划分为二维和三维 (4.1 小节), 从自重构过程中模块的运动方式上可以划分为链式、逐个移动式和混合式 (4.2 小节).

4.1 根据空间维度划分

根据模块连接的空间维度, 自重构机器人可以划分为二维和三维. 二维自重构机器人中的模块只能沿着 x 和 y 两个方向 (或这两个方向的线性组合) 连接在一起, 而三维自重构机器人中的模块可以沿着 x, y 和 z 3 个方向 (或这 3 个方向的线性组合) 进行连接. 早期的自重构机器人大多数是二维. 设计二维自重构机器人要比设计三维自重构机器人简单, 因为 z 方向 (通常为高度方向) 没有模块连接, 可以用于挂载电源、线缆等元件. 二维自重构机器人通常为水平放置, 所以在模块组织和自重构时不需要克服重力来移动模块, 可以采用功率较小的驱动器, 同时连接器等机械结构所需强度也比三维自重构机器人要小. 由于二维自重构机器人只在 x 和 y 两个方向张成的平面内连接, 所以它的模块一般采用如正三角形、矩形和正六边形这些适合平面填充的几何形状. 典型的二维自重构机器人包括 Metamorphic [43, 44], Fracta [28], SWARM-BOT [45], YaMoR [26] 等.

近年来三维自重构机器人开始不断出现. 对于三维自重构机器人而言, 由于需要在 z 这个高度维度上进行模块连接和自重构, 它对驱动器的功率和连接器的强度强度都有很高的要求. 三维自重构机

3) Alberto. “Dtto”. https://hackaday.io/project/9976-dtto-explorer-modular-robot.

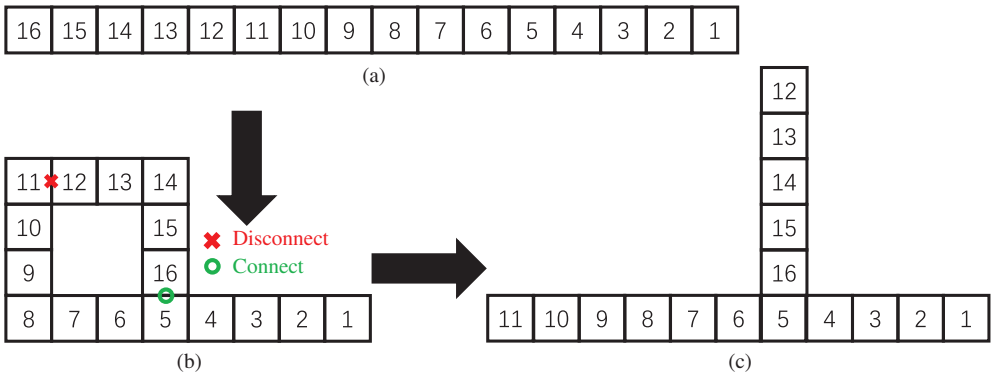


图 14 (网络版彩图) 链式自重构过程

Figure 14 (Color online) The chain-typed self-reconfiguration. (a) Modules 9–16 are regarded as a chain, and it can be fold through the cooperation among modules; (b) module 16 moves to connect module 5, then module 11 disconnects module 12; (c) modules 12–16 are regarded as a new chain

器人的模块外形以立方体、八面体和十二面体这些适合三维空间填充的几何形状为主. 典型的三维自重构机器人包括 Molecule [46], PolyBot [47], Telecubes [16], Miche [48] 等.

4.2 根据自重构过程中模块的运动方式划分

不同的自重构机器人外形差异较大, 但根据自重构过程中模块的运动方式可以被分为 3 种类型: 链式、逐个移动式和混合式.

• 链式自重构. 这种类型的机器人在自重构过程中, 局部多个模块可以形成一个链状结构并作为一个整体在机器人中进行移动. 举例来说, 如图 14 所示, 可以多个顺序连接的模块作为一个链状结构, 通过该链状结构中所有模块驱动器的联合协调运动, 将链状结构的末端模块移动到一个新的位置; 末端模块在此新的位置与相邻的模块进行连接, 同时链状结构中原先与机器人相连的模块断开连接, 实现整个链状结构在机器人中的移动过程. 这种类型的自重构过程每次可以移动多个模块, 但每次移动时需要通过链状结构的末端模块与其他模块连接成环, 再在所需位置断开连接以实现自重构. 由于连接过程需要末端模块与相应模块对齐, 因此该过程要求多个模块驱动器的协同工作以实现链的弯曲和末端模块的精确定位, 对模块驱动器的控制精度和多个模块间的同步通信都要求很高, 自重构过程缓慢且复杂.

• 逐个移动式自重构. 这种类型的自重构过程中, 每次只有一个模块驱动器在工作. 比如对于中心铰接型、边铰接型和绕边旋转型模块, 一个驱动器工作对应每次只有一个模块在机器人中移动 (如图 15 所示). 对于伸缩型模块, 一个驱动器工作可能带动多个模块发生移动, 但构型变换的效果还是可以视为是单个模块从一个位置变换到了另一个位置 (如图 10 所示). 逐个移动式自重构对模块驱动器的控制精度和模块间的同步通信要求相较链式自重构低, 因此这种类型被广泛地采用.

• 混合式自重构. 该类型机器人在自重构过程中的每一步, 可以根据需要选择链式自重构或逐个移动式自重构. 这使得自重构的变换过程更加灵活, 同时也增加了自重构算法设计的难度. 与链式自重构一样, 混合式自重构对模块驱动器的定位精度和模块间通信要求很高.

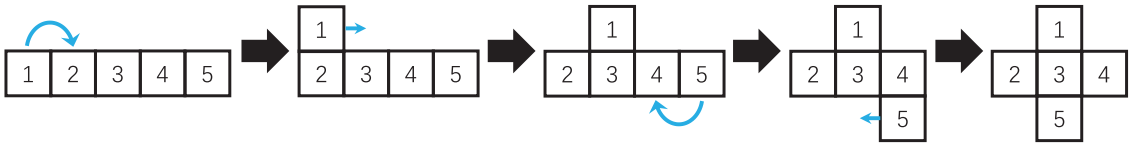
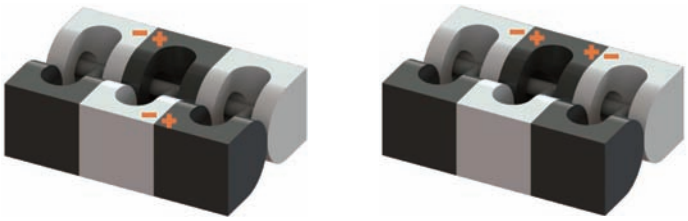


图 15 (网络版彩图) 逐个移动式自重构过程: 该过程中每次只能移动一个模块

Figure 15 (Color online) The one-by-one-type self-reconfiguration. Only one module moves in each step



(a) (b)

图 16 (网络版彩图) 3 个 M-TRAN 模块组成的两个不同构型. 一个 M-TRAN 模块有一个公子模块 (黑色) 和一个母子模块 (白色). 图中一对 $(+,-)$ 表示一个公连接器的挂钩插入一个母连接器的凹槽中形成连接, 没有 $+$ 或 $-$ 符号标记的公或母子模块仍然可以自由旋转并移动到新的位置. (a) 和 (b) 构型虽然形状相同, 但由于连接方式不同, 所以是两个不同的构型.

Figure 16 (Color online) Two configurations composed by two M-TRAN modules. A M-TRAN module consists of female cube (white) and male cube (black). (+,−) represents that the hook in male cube is inserted into the slit in female cube of another module. The male and female cube without '+,−' is free to rotate. Configurations (a) and (b) are different because their connection methods are different

5 模块化机器人的形状、构型与构型空间

在定义模块化机器人自重构以及讨论自重构算法之前, 我们需要介绍几个模块化机器人的相关概念. 模块化机器人的形状是指整体机器人的外形, 由其所有组成模块的位置和朝向所决定. 模块化机器人的构型包括形状和拓扑连接关系两部分. 即使两个模块化机器人的形状相同, 如果模块间的连接关系不同, 这两个机器人的构型也不相同. 5.1 小节介绍描述构型的两种数学表示方法. 一个由 n 个模块组成的机器人, 它所能表达的所有构型称之为构型空间. 自重构则是指从构型空间中一个构型变换到另一个构型的过程. 在一个构型上执行一步自重构操作 (包括 4.2 小节中的 3 种方式) 可以得到另外一种构型, 在这两个构型之间连一条边, 那么自重构过程就是在构型空间中两个构型之间找到一条路径. 构型空间的大小是衡量不同类型模块的构型表达能力的一个重要指标, 也是度量自重构算法难易程度的一个重要指标. 5.2 小节介绍刻画构型空间大小的数学枚举方法. 由于构型数与模块数 n 的呈指数关系, 对于一个较大的 n, 枚举方法很难在一个可以接受的时间内计算出结果, 因此只能用理论下界对其作出估算. 5.3 小节将介绍构型空间理论下界的计算方法.

5.1 构型的表示方法

相同形状的模块化机器人可能具有不同的构型, 因此由相同数量模块组成的模块化机器人的构型数大于形状数. 图 16 给出了一个例子. 为了区分不同的构型, 常用的数学表示方法包括以下两种.

- 加权图表示法. 将构型中的每个模块看作一个结点, 如果两个模块相互连接, 则对应的两个结点间有边相连. 两个不同形状 (图 17(a) 和 (b)) 或两个形状相同但构型不同 (图 16) 的模块化机器人可能有相同的图表示法 (图 17(c)). 因此, 需要记录连接面的信息并作为图中边的权重, 这种加权图表示

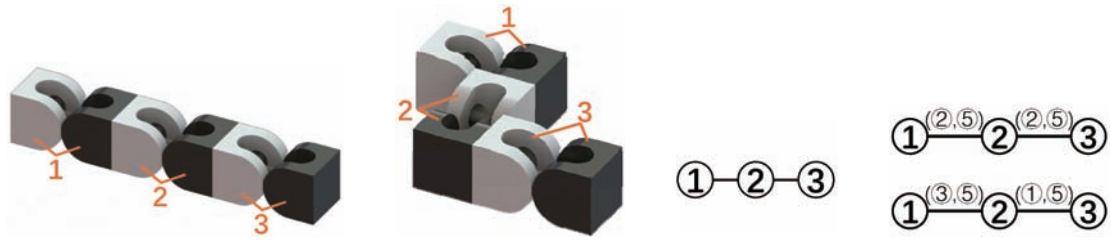


图 17 (网络版彩图) **(a) 3 个 M-TRAN 模块组成的构型 1; (b) 同样 3 个模块组成的另一种不同的构型 2; (c) 将构型中的每个模块看作一个结点, 如果两个模块相互连接, 则对应的两个结点间有边相连. 这种图表示方法无法区分构型 1 和 2, 它们的图表示是相同的; (d) 在图中的每条边上记录连接面的信息并作为图中边的权重, 这种加权图表示法能够有效地区分构型 1 和 2**

Figure 17 (Color online) (a) The configuration 1 is composed of three M-TRAN modules; (b) the configuration 2 is composed of three M-TRAN modules; (c) graph representation for the configuration. Each module is regarded as a node. There is an edge between two nodes if two modules are connected; (d) weighted digraph representations for two configurations. This method separates configurations 1 and 2 easily

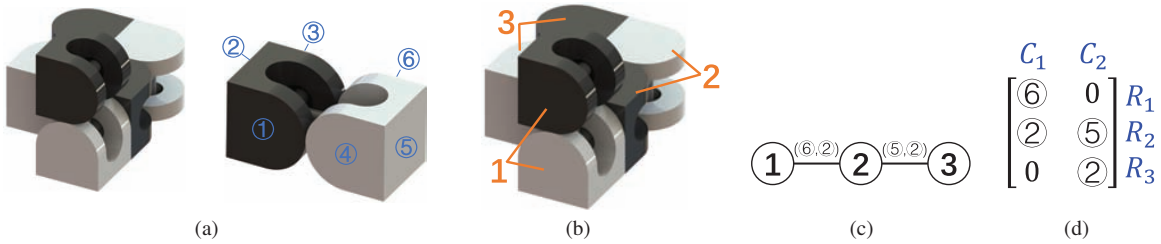


图 18 (网络版彩图) **(a) M-TRAN 模块中各个面的唯一标示. (b) 一种由 3 个 M-TRAN 模块所组成的构型. (c) 该构型的加权图表示. (d) 加权图对应的矩阵表示. 构型中 3 个模块共有两个连接面, 分别对应矩阵的 C_1 和 C_2 两列. 每列中只有两个非零单元, 分别对应所连接的两个模块. 比如 C_1 列中第 1 和第 2 两个单元非零, 表示第 1 个连接面连接第 1 和第 2 个模块. 非零单元中的数字表示对应模块连接的是哪一个面, 比如 C_1 列中第 1 和第 2 两个单元分别是 ⑥ 和 ②, 表示第 1 个连接面是由第 1 个模块的 ⑥ 面和第 2 个模块的 ② 面组成**

Figure 18 (Color online) (a) Each face is uniquely identified with an ID; (b) a configuration of three M-TRAN modules; (c) the weighted graph representation of configuration in (b); (d) the matrix representation of configuration in (b). Two columns of the matrix correspond to the two joint faces in (b). Three rows of the matrix correspond to three modules. Two non-zero units in each column correspond to the id of the faces which are connected to other modules. For example, column $C1$ means the face 6 of module 1 and the face 2 of module 2 are connected

法能够有效地区分不同的构型 (图 17(d)).

- 矩阵表示法. 对于由 n 个模块组成的构型, 定义一个 $n \times m$ 的矩阵 M, m 为连接面 (或关节) 的数量, 每个单元 $M_{i,j}$ 表示第 i 个模块是否与第 j 个连接面相连, 以及连接的具体是哪一个面 (图 18). 可以证明, 这种矩阵表示法与加权图表示法等价, 两者可以互相推导.

5.2 构型空间大小与枚举方法

给定一个数量为 n 的模块集合, 通过模块之间的不同连接方式可以得到不同构型的模块化机器人. 构型的数目越多, 说明这 n 个模块所能表达的形态越多. 构型空间是模块化机器人所能表达的所有构型的集合. 构型空间的大小或基数是该集合中所有构型的数量. 由于每个模块的摆放位置、朝向和连接关系是有限的, 对于一个较小的 n, 可以通过枚举的方法获得构型空间的大小.

枚举的方法可以用一个简单的排列组合模型来刻画. 我们以一种简化的立方体模块为例, 它的

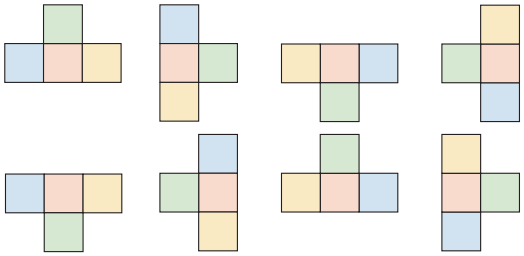


图 19 (网络版彩图) 由镜像和旋转操作得到的 8 种构型, 虽然几何位置或形状不同, 但都属于同一种构型

Figure 19 (Color online) Eight configurations which are mirrored or rotated from each other belong to same configuration

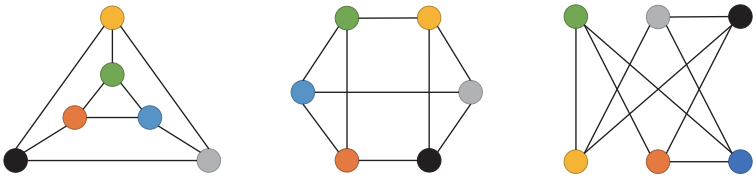


图 20 (网络版彩图) 3 个图虽然几何位置和形状均不相同, 但都是同构的

Figure 20 (Color online) Three graphs above are isomorphic

6 个面中任何一个面都可以采用无性别连接器和另外一个立方体模块相连. 首先从 n 个模块中随机挑选一个, 再从剩下的 $n-1$ 个模块中随机挑选一个与第 1 个模块连接, 共有 $\binom{n}{1}\times\binom{n}{2}\times 6\times 6$ 中可能性. 枚举法就是通过人工摆放出这些具体位置, 再去掉重复的构型得到所有的构型. 这里重复的构型是指一种构型可以通过平移、旋转、镜像或者综合以上几种方法得到一系列同构构型, 图 19 给出了一个例子. 如果采用 5.1 小节中的加权图表示方法表达构型, 那么同构构型可以通过图同构的概念来描述. 图同构指的是对于两个图 G_1 和 G_2, 其中 $G_1=(V_1,E_1)$ 和 $G_2=(V_2,E_2)$, 如果存在一个双射 $m:V_1\to V_2$, 对所有 $x,y\in V_1$, 使得 $e=(x,y)\in E_1$ 当且仅当 $(m(x),m(y))\in E_2$, 则称 G_1 和 G_2 是同构的, 记为 $G_1\cong G_2$. 图 20 给出了一个同构图的示例.

为了简化枚举构型空间的难度, 可以采用唯一构型签名的方法作为标识符来表征该构型的同构构型集合. 该标识符将每种构型都对应到一组排列好的模块数组, 模块的排序顺序如下: 假设有 M_1 和 M_2 两个模块, 空间坐标分别为 (x_1,y_1,z_1) 和 (x_2,y_2,z_2), 如果 $x_1<x_2$, 则 $M_1<M_2$. 如果 $x_1=x_2$, 则依次比较 y 和 z 方向上的坐标值. 每个模块被抽象为 6 面都有连接器的立方体结构, 连接后均能绕轴旋转 0, 90, 180 和 270 度. 通过将构型旋转或镜像得到所有的同构构型, 二维的有 8 种 (4 种旋转变换 $\times$ 2 种镜像变换, 如图 19 所示), 三维有 24 种 (二维的种类 $\times$ 3 个维度). 计算出所有同构构型所对应的模块数组后, 选取其中最小的数组作为该同构构型集合的构型签名, 其排序方式如下. 记一种配置 C_i 是一个由模块构成的有序数列, 按照模块的全序关系从小到大进行排序, 其中的元素 C_i^j 为该配置的第 j 个模块, 配置间的偏序关系定义为

$$C_1<C_2\leftrightarrow\forall k\in\{1,2,\ldots,n\}|C_1^k<C_2^k,$$

其中 n 为构型中的模块总数.

IT University of Copenhagen 和优傲机器人公司 (Universal Robots) 的研究人员 [49] 对几种不同类型模块的构型空间进行了枚举. 举例来说, 4 个 M-TRAN 模块组合能表达的构型数有 41627711 种, 而 3 个 SuperBot 模块组合能表达的构型数有 3003064 种. 对于一种简化的立方体模块, 模块间采用

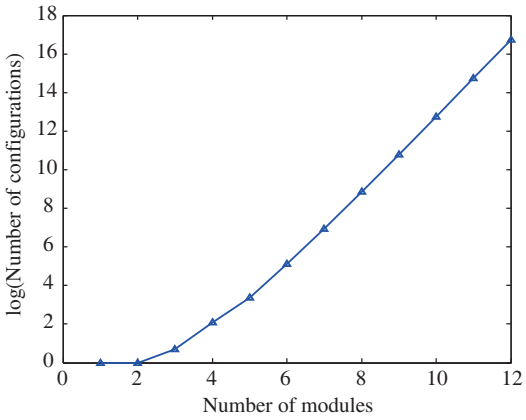


图 21 (网络版彩图) 模块为简化的立方体、模块间采用简化的无性别连接器时, 构型数与模块数量的关系. 横轴为模块数, 纵轴为构型数的对数, 其结果近似为一条直线. 拟合函数的表达式为 $f(n) = \mathrm{e}^{1.62n-3.7}$. 由此可以看出随着模块数的增加, 构型数呈指数级增长

Figure 21 (Color online) The relationship between number of configurations and number of modules. The horizontal axis is the number of modules, and the vertical axis is the logarithm of the configuration number. The result is approximately a straight line. The expression of the fitting function is $f(n) = \mathrm{e}^{1.62n-3.7}$, which indicates that the number of configuration increases exponentially with the increase of module number

简化的无性别连接器时, 文献 [49] 中给出了模块数从 1 到 12 时所能表达的构型数. 我们把这些数据拟合成一张对数图, 如图 21 所示, 其中横轴为模块数, 纵轴为构型数的对数, 以模块数 n 为自变量, 其构型数 $f(n)$ 的函数拟合结果为 $f(n) = \mathrm{e}^{1.62n-3.7}$. 由该拟合结果可以看出, 随着模块数的增加, 构型数呈指数级增长, 由 12 个立方体模块组成的机器人, 其构型数已经可以达到 18598427 之多. 所以即使在模块数量很少的情况下, 模块化机器人所能表达的构型空间也是非常大的. 因此当模块数量较大时, 精确枚举出所有的构型非常困难甚至无法实现, 只能用理论下界值对其做出估计.

5.3 构型空间的理论下界

20 世纪 60 年代, 美国数学家、New Mexico State University 的 Frank Harary 教授对相同类型的单元所能构成的形状数量问题进行了研究, 并在文献 [50] 中把它列为图枚举研究领域的一个尚待解决的问题. Harary 用两种不同的集合 $S(n)$ 和 $T(n)$ 来表示由 n 个相同模块组成的形状数目. 在 $S(n)$ 中, 形状相同但朝向不同的形状视为同一形状, $S(n)$ 集合中元素个数记为 $s(n)$. 在 $T(n)$ 中, 形状相同但朝向不同的形状视为不同, $T(n)$ 集合中元素个数记为 $t(n)$. 举例来说, 图 19 中所示的 8 个图形, 在 $S(n)$ 中视为同一图形, 但在 $T(n)$ 中视为不同图形. 显而易见, $s(n)$ 和 $t(n)$ 之间的关系为

$$\frac{t(n)}{8} < s(n) < t(n).$$

Eden 在文献 [51] 中给出了 $s(n)$ 的上下界, 即 $(3.14)^n < s(n) < 4^n$. 在 Eden 工作的基础上, Klarner 和 Rivest [52] 在 1973 年证明了以下结论:

$$\text{极限} \lim_{n\to\infty}(s(n)^{\frac{1}{n}}) = \theta \quad \text{存在, 且} \quad 3.87 < \theta < 4.65.$$

这就说明, 随着模块数量的增长, 构型空间可能的大小也呈指数级的上升. 具体而言, 对 n 个三角形单元, $s(n) > 2.13^n$; 对 n 个正方形单元, $s(n) > 3.87^n$; 对 n 个六边形单元, $s(n) > 4^n$. 这里的下界只考虑

了模块的相对位置和方向, 并未考虑连接器的位置以及可能具有的不同性别, 如果一并考虑连接器的这些信息, 数目会更多.

6 自重构路径规划算法

模块化机器人的自重构可以看成是在构型空间中, 在两个构型之间找到一条满足某种条件的最优路径 (6.1 小节). 具体定义为模块化机器人在模块数量保持不变的情况下, 从初始构型变换到目标构型的全部模块运动过程. 与之对应, 自重构路径规划问题是指如何规划并实现该自重构过程的问题. 如 4.2 小节所述, 链式自重构方式对模块驱动器的控制精度和模块间通信都有很高的要求, 自重构过程缓慢且复杂, 在实际物理实现的模块化机器人中并不常见. 所以本节主要针对逐个移动式自重构路径规划方法进行阐述. 该问题可以用如下的数学模型来描述:

自重构路径规划问题. 给定一个初始构型 I 和一个目标构型 G, I 和 G 由相同数量的 n 模块组成. 模块的运动满足以下条件:

- 每次只移动一个模块;
- 模块移动时, 所有模块仍然是连通的.

对于构型 I 中的所有模块, 寻找这样一个模块移动序列, 使得按照此序列中的顺序依次移动 I 中的相应模块, 可以重构得到目标构型 G. 该模块移动序列称为自重构路径.

由于自重构过程的时间成本和能源消耗均可以用模块移动的步数来刻画, 自重构路径规划通常对应到求解一个优化问题.

自重构路径规划优化问题. 在所有可能的自重构路径中, 模块移动步数最少的路径称为最优路径. 自重构路径规划优化问题即为寻找最优路径问题.

6.2 小节中, 我们论述自重构路径规划优化问题是 NP 难问题, 所以在多项式时间内寻找最优解是不可能的, 只能采用启发式策略寻找最优解的近似解. 6.3 小节中我们介绍几种常见的启发式度量, 它们均适用于各类模块化机器人. 此外, 还针对几种特殊的模块运动类型, 包括伸缩型模块 (6.4 小节)、滑动型模块 (6.5 小节) 和元模块 (6.6 小节) 3 类, 分别介绍它们特有的自重构路径规划算法.

6.1 图遍历与最短路径搜索方法

一个简单直接的自重构路径规划方法是用 5.2 小节介绍的方法, 将 n 个模块所能组成的所有构型枚举出来, 并表征为一个连通图 $G_n = (V_n, E_n)$:

- 每个顶点 $v \in V_n$ 对应一种构型;
- 如果能够通过将 v_i 中的某个模块移动一步得到 v_j, 则 v_i 和 v_j 间连一条边 $e \in E_n$.

显而易见, 构型 v_p 到构型 v_q 的每条自重构路径对应图 G_n 中顶点 v_p 和 v_q 间的一条路径. 所以经典图论的相关图遍历方法 (比如深度、广度优先算法) 都可以应用到求解自重构路径规划优化问题中. 相应的, 最优自重构路径对应顶点 v_p 和 v_q 在图 G_n 的最短路径, 可以由经典的 Dijkstra 算法求解 [53].

以上基于图遍历与最短路径搜索的方法虽然简单, 但由 5.3 小节可知, n 个模块所形成的构型空间的复杂度下界为 $\Omega(\mathrm{e}^n)$. 这种指数级的复杂度很难在实际中得到应用. 于是研究人员开始尝试寻找多项式时间复杂度的路径规划方法. 然而, 在 6.2 小节中, 我们阐述说明链式自重构方式下, 模块化机

器人的自重构路径规划优化问题是 NP 难问题, 所以只能通过 6.3 小节中介绍的启发式度量来给出具有多项式时间复杂度的近似解.

6.2 链式自重构路径规划优化问题是 NP 难问题

5.3 小节中已经指出, n 个相同模块的可以组装出 C^n 种不同的构型, 其中 C 为大于 3 的常数. 一方面, 这说明模块化机器人可以表达的形状范围很大, 具有很强的潜在变形能力. 另一方面, 这也意味着模块化机器人自重构路径规划方法的复杂度很高. 但 $\Theta(\mathrm{e}^n)$ 复杂度的搜索空间并不一定意味着该空间中的所有搜索算法都是指数级的, 比如在一个 n 节点的无向图中, 两个节点之间可能存在 $O(\mathrm{e}^n)$ 条简单路径; 但是搜索该两点间最短路径的时间复杂度仅为 $O(n^2)$. 所以研究人员也曾试图构建多项式时间复杂度的自重构路径规划优化方法. 2010 年, 文献 [54] 证明了链式模块化机器人的自重构路径规划优化问题是 NP 难问题. 这意味着寻找最优解是不现实的, 研究人员于是开始转向提出各种近似优化方法. 下面, 为清晰的介绍文献 [54] 中的成果, 我们首先简要回顾与 P/NP 问题有关的几个定义 [55].

P 类问题. 所有可以在多项式表达的时间内解决的问题.

NP 问题. 所有可以在多项式时间内验证解是否正确的判定问题.

P 类问题和 NP 问题是否恒等 (P=NP?), 是理论计算机科学领域至今仍然没有解决的问题. 虽然还没有明确的答案, 但大部分研究人员认为 NP 问题不太可能存在多项式时间的算法解.

NP 完全问题. 所有满足以下条件的判定问题:

- 它是一个 NP 问题;
- 其他的 NP 问题都可在多项式时间内归约成它.

这里, NP 问题 L_1 可归约成 NP 问题 L_2, 记为 $L_1 \preceq L_2$, 是指存在一个规约映射 f, 满足:

- f 将问题 L_1 的每一个实例都映射到问题 L_2 的一个实例;
- f 满足: $x \in L_1$ 当且仅当 $f(x) \in L_2$, $\forall x$.

如果映射 f 可以在多项式时间内计算完成, 则称 L_1 可在多项式时间内归约成 L_2, 记为 $L_1 \preceq_{\mathrm{P}} L_2$.

NP 难问题. 如果所有的 NP 问题可以在多项式时间内归约到某个问题, 则该问题称为 NP 难问题.

因为 NP 难问题不一定是 NP 问题, 所以 NP 难问题的解的复杂度不会低于 NP 完全问题.

由以上定义可知, 需要证明一个判定问题是否是 NP 完全问题或一个优化问题是否是 NP 难问题, 一个常用的技巧是找到一个已知的 NP 完全问题, 把它归约到需要证明的问题上. 介绍 P/NP 问题的专著 [55] 中, 收集了超过 300 个 NP 完全问题和 NP 难问题, 其中三划分问题已被证明是 NP 完全问题. 针对链式自重构模块化机器人, 文献 [54] 中给出了一个实例映射, 在多项式时间内将三划分问题归约到了自重构路径规划优化问题, 所以链式模块化机器人的自重构路径规划优化问题是 NP 难问题.

6.3 启发式搜索和启发式度量

对于 NP 难的模块化机器人自重构路径规划优化问题, 研究人员采用了多种启发式方法寻找多项式时间内的最优解的近似解. 对于 6.1 小节中所述的图遍历方法, 由于图中具有 $O(\mathrm{e}^n)$ 个节点, 寻找两个节点 (分别对应初始构型和目标构型) 间的最短路径所用的 Dijkstra 算法的时间复杂度也为 $O(\mathrm{e}^n)$. 与 Dijkstra 算法向节点四周毫无方向的均匀搜索不同, 启发式搜索能够对每个可能经过的节点使用估

价函数进行评估, 得到最有可能通向目标节点的方向, 再从这个方向继续搜索直到达到目标节点, 这种方法可以节省大量的无效搜索路径, 从而提高搜索效率.

启发式搜索的一个经典算法是 A* 搜索. 在模块化机器人的路径规划应用中, 其估价函数 $f(x)=d(x)+h(x)$ 的具体表达为, $d(x)$ 代表从初始构型到当前构型 x 的实际距离, $h(x)$ 为在最优路径中从当前构型 x 到目标构型的估计距离. 搜索过程中维护 open 和 close 两个列表, 其中 open 表记录待考察的节点, close 表记录已访问过的节点. 每次根据估价函数选择 open 表中 $f(x)$ 值最大的节点, 将其从 open 中删除并放入 close 中, 检查与其相邻的节点, 如果这些节点可达并且不在 open 中则将这些结点放入 open, 如果已经存在于 open 中, 则考察通过该选定节点到达这些节点的 $d(x)$ 值是否更小, 如果是则更新 $f(x)$, $d(x)$ 并将父节点改为该选定节点, 如果不是则不做操作. 不断循环以上过程, 直到将目标构型对应的节点加入 open 中. 记从当前构型 x 到目标构型的真实距离为 $d_g(x)$. A* 搜索的估价函数具有以下特点 [56].

(1) 当 $h(x)=d_g(x)$, 搜索将严格按照最短路径进行, 搜索效率最高;

(2) 当 $h(x)<d_g(x)$, 搜索范围变大, 搜索效率降低, 但可以找到最优解;

(3) 当 $h(x)>d_g(x)$, 搜索范围比第 2 种情况小、速度快, 但不能保证找到最优解.

对于模块化机器人路径规划问题而言, 需要定义一种度量来计算图中任意两个节点间的距离. 度量设计的好坏直接决定了启发式搜索的效率. 记 A 和 B 为任意两个由 n 个模块组成的构型. 任何一种度量 $\tilde{d}$ 均需要满足如下条件:

- 非负性. $\tilde{d}(A,B)\geqslant 0$;
- 同一性. $\tilde{d}(A,B)=0\Longleftrightarrow A\bigcap B=n\Longleftrightarrow A=B$;
- 对称性. $\tilde{d}(A,B)=\tilde{d}(B,A)$;
- 三角不等式. $\tilde{d}(A,C)\leqslant\tilde{d}(A,B)+\tilde{d}(B,C)$, 其中 C 为任意一种构型.

美国 The Johns Hopkins University 的研究人员在文献 [57] 提出了几种可用于模块化机器人自重构路径规划的度量, 包括重叠性度量、最小移动步数度量和最优分配度量.

重叠性度量. 定义为 A 和 B 构型中位置不重合的模块数量, 记为 $\beta_{\text{overlap}}=n-|A\bigcap B|$. β_{overlap} 代表最少需要移动的模块数目. 对于两个完全相同的构型来说, $A\bigcap B=n$, 因此 $\beta_{\text{overlap}}=0$. 但重叠性度量只能表示两个构型中非重合的模块数量, 不能反映为了实现构型间的变换实际所需要的步数, 两种不同的构型变换可能具有相同的重叠性度量值, 但实际步数相差很大. 如图 22 所示, 其中 $\beta_{\text{overlap}}(a,b)=\beta_{\text{overlap}}(a,c)=1$, 但实际模块移动的步数明显后者要大于前者.

最小移动步数度量. 定义为从构型 A 变换到构型 B 所需的最小移动步数, 记为 $\beta_{\min}$. 文献 [57] 中证明 $\beta_{\min}$ 满足度量所需的 4 个条件. 但计算两个构型间的最小移动步数通常是一件非常困难的事情. 一个简单的近似方法是先计算两个构型间的重合模块, 剩下的只需要考虑移动那些非重合的模块. 记 A 和 B 构型的模块位置集合分别为为 S_A 和 S_B, 重合模块集合 $S_O=S_A\cap S_B$. 记 $S_1=S_A\setminus S_O$, $S_2=S_B\setminus S_O$. 给 S_1 和 S_2 中的每一个模块赋予一个数字标识, 用该模块到 S_O 的最小移动距离来表征. 将 S_1 集合中标识的最大模块对应到 S_2 中标识最小的模块, 并用 Dijkstra 算法计算最短路径得到该模块的最小移动步数. 如此迭代循环, 直到 S_1 和 S_2 集合为空. 除了以上这种近似方法, 其他能够建立 S_1 和 S_2 集合中模块对应关系的算法都可以应用. 但这种度量的计算量非常大, 当模块数量较大时, 非常耗时.

最优分配度量. 给定一个双射 $f:A\to B$, 它将 A 构型中的模块 a_i 映射到 B 构型中的模块 b_j. 记 $f(A,B)=\sum_{1\leqslant i,j\leqslant n}d_{ij}m_{ij}$, 其中 d_{ij} 为模块 a_i 和 b_j 间的 Hausdorff 距离, $m_{i,j}$ 为一个 Boolean 变量, 计算方法如下: 若当前构型中的模块 a_i 映射到新构型中的 b_j 位置, 则 $m_{i,j}=1$, 否则为 0. 使用

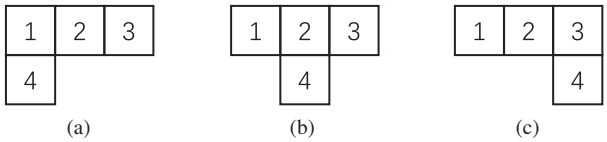


图 22　重叠性度量. (a), (b) 和 (a), (c) 中重叠的模块均为 1, 2, 3, 因此重叠性度量值均为 1, 但从 (a) 重构为 (b) 只需要 1 步, 从 (a) 重构为 (c) 需要 2 步

Figure 22　The overlap metric. Modules 1, 2, 3 are overlapped in (a), (b) and (a), (c), so both the overlap metrics are 1. However, transformation from (a) to (b) needs only one step while (a) to (c) needs two steps

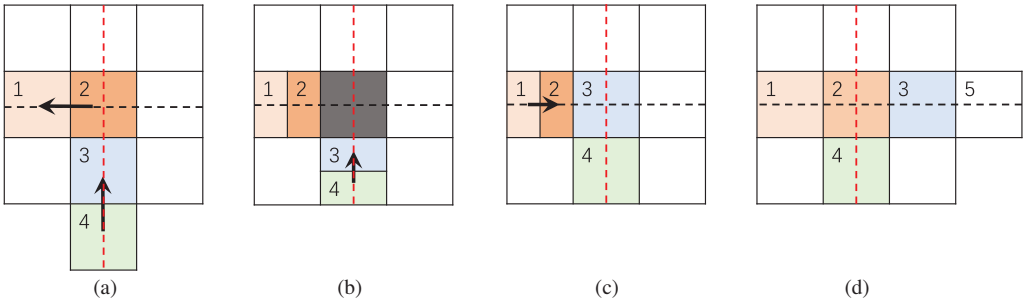


图 23　(网络版彩图) 伸缩型模块化机器人的构型变换原理

Figure 23　(Color online) The principle of expanded-and-extracted modules. Module 1, 2, 3, 4 of (a) were contracted, and the configuration is transformed into (b). Then modules 3 and 4 were expanded, transformed into (c). Then modules 1 and 2 were expanded, transformed into (d). Because the modules are all the same, the reconfiguration between (a) to (d) can be regarded as that module 4 moved to the position of module 5

Hungarian 算法 [58] 可以求出所有匹配中的最小 $f(A,B)$ 值, 即为最优分配度量值.

6.4　伸缩型模块的自重构路径规划

6.3 小节中介绍的启发式自重构路径规划方法, 适用于第 2 节中介绍的各种模块类型. 除了通用方法外, 对于具有特殊运动形式的模块, 还可以设计专用的方法. 本小节我们介绍伸缩型模块的自重构路径规划方法.

除了模块化机器人必须具有的、模块间的自动连接和断开功能之外, 伸缩型模块还具有伸展及收缩功能, 这 4 个基本功能构成了伸缩型模块进行自重构的基础. 其他类型的模块自重构过程中只能在机器人的表面运动, 但伸缩型模块通过伸展与收缩操作可以实现在机器人的内部空间内移动. 这种特殊运动方式, 使得伸缩型模块对完成自重构所需自由空间的限制很小, 同时模块的 "重定位" 比其他类型模块相对容易, 不需要将某个特定模块逐步移动到特定位置, 只需将目标位置的相邻模块移至目标位置即可实现构型的变换. 一个简单的二维重定位过程如图 23 所示. 首先分别做出通过起始模块位置 L_o (图 23(a) 中模块 4 位置) 指向机器人内部并平行于坐标轴的辅助线 (红色), 以及通过目标模块位置 L_d (图 23(d) 中模块 5 位置) 指向机器人内部并平行于坐标轴的辅助线 (黑色), 两条辅助线的交点所在的模块位置设为 L_i. 将紧邻 L_i 的在红色虚线上朝向起始模块的两个模块收缩合并为 M_1 使得初始模块进入机器人内部, 将 L_i 位置的模块及紧邻 L_i 的在黑色虚线上反向目标模块的模块收缩合并为 M_2, 空出 L_i 位置 (图 23(b) 中深灰色区域). 将 M_1 伸展, 将模块 3 移动到 L_i 位置, 将 M_2 伸展, 将最右侧白色块推出, 实现重定位. 三维上的重定位方法在原理上与二维情况完全相同.

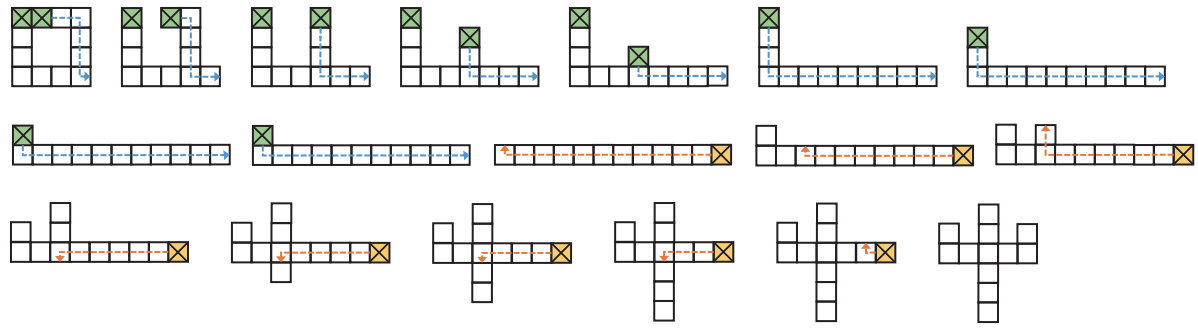


图 24 (网络版彩图) 融化生长自重构算法示例. 先将原始“口”字构型 (左上角) 变换为中间构型“—”字型, 再变换成为目标的“叉”构型 (右下角)

Figure 24 (Color online) An example of “melt-grow” algorithm. The original shape are reconfigured to a linear form, then reconfigured to the target shape

对于单个伸缩型模块在机器人中的重定位操作, 上述重定位方法仅适用于机器人外形为凸多边形或凸多面体的情况, 其单个模块重定位的操作均可以在 $O(1)$ 常数时间内实现. 机器人外形为凹多边形或凹多面体时, 单个模块重定位操作的时间复杂度为 $O(k)$, 其中 k 为凹多边形或凹多面体边界上凹点的数目. 其重定位算法可以归纳如下. 记 C' 为任意一个外形为凹多边形或凹多面体的机器人构型, s 为 C' 表面上一个需要移动模块的初始位置, g 为该模块需要在 C' 上移动到的目标位置. 首先建立 C' 中所有模块的连通图 $G(C')$. 在 $O(t)$ 时间内能在 $G(C')$ 中找到一条从 s 到 g 的路径 L, t 为表达路径的折线段中拐点的数目, 易知 $t = O(k)$. L 由多条直线段首尾相接连接而成. 沿着 s 所在的 L 中的直线段方向将两个模块压缩到一个模块位置 s', 再沿着路径 L 中线段顺序传递 s', 直到传递 g 所在的线段上, 最后释放 s' 并将模块伸展推出到 g 位置, 该过程时间复杂度为 $O(t)$. 图 10(a)∼(c) 3 个图说明了沿着辅助线方向从初始位置到目标位置模块的压缩传递过程.

基于单个伸缩型模块的重定位算法, 文献 [8] 中进一步提出了一种针对二维的 Crystalline 伸缩型模块的融化生长 (melt-grow) 自重构算法. 其核心思想是先将原始构型 S 和目标构型 G 都变换到一种中间构型 I, 再对 G 到 I 的自重构过程取逆. 该算法的时间复杂度为 $O(n^2)$, 其中 n 为模块数. 该算法基于的最小单位为一个集合 Grain(4), 这个集合在二维和三维情况下分别对应一个 4×4 和 $4\times4\times4$ 模块组合. 算法中所采用的中间构型 I 在理论上可以有多种选择, 为了清晰地介绍算法的本质, 本文统一用直线段这个二维形状作为中间构型 I. 融化生长算法由以下 4 个部分组成.

(1) 对于任何一种构型 C, 以 4×4 模块组合为基本单位 (称为一个晶粒) 将 C 转换为一个晶粒连通图, 记为 GCG(C);

(2) 计算中间构型的结构 I, 比如一条直线段;

(3) (融化) 将原始构型 S 通过一个迭代过程变换为 I. 首先将连通图 GCG(C) 中的所有割点对应的晶粒作为候选晶粒集合. 在每轮迭代中, 从候选晶粒集合中选择一个晶粒, 将其移动到该晶粒在 I 中最近的一个需要填充的位置上. 不断重复此操作直到变换完成. 同理, 将目标构型 G 变换为 I;

(4) (生长) 将目标构型 G 到构型 I 的变换过程取逆, 即每次在 I 中寻找合适的晶格位置并将其移动到 G 中的对应位置, 不断重复直到变换完成.

图 24 中给出了该算法的一个简单实例.

与二维伸缩型模块 Crystalline [8] 类似, 三维伸缩型模块 Telecubes [16] 在自重构过程中也采用元模块 (即一个 $2\times2\times2$ 模块组合) 为最小单位. 两者区别在于, Crystalline 模块化机器人中元模块每条

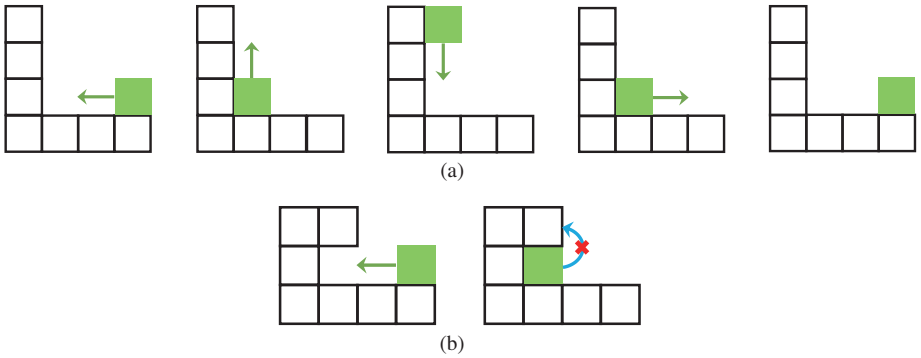


图 25 (网络版彩图) (a) 滑动型模块能在凹形表面上往返运动; (b) 滑动型模块不能在凸形表面上做翻转运动

Figure 25 (Color online) (a) The sliding module is allowed only sliding movement across other modules' surfaces; (b) the sliding module is not allowed rotate around other modules

边上的 4 个模块必须同时运动, 但 Telecubes 模块化机器人中, 元模块内部的模块仍能独立地做伸缩运动. 在 Telecubes 模块化机器人自重构过程中, 对于每个需要移动的模块 M, 要求即使在构型中去除 M, 剩下的模块仍然保持连通. 然后使用 PacMan 算法 [59] 计算出所需路径, 再将其分解为 Telecubes 模块能完成的运动方式从而实现构型间的自重构.

6.5 滑动类型模块的自重构路径规划

6.4 小节介绍的伸缩型模块自重构, 允许模块在构型内部传递, 可以充分利用模块化机器人的内部物理空间. 但该自重构方法在原理上要求一个模块在做伸缩操作时, 可以带动多个 (原理上数量不限) 模块进行移动. 但 Crystalline 模块选择的驱动器为乐高公司生产的迷你电机, 扭矩为 1.4 N/cm, 其功率大小无法带动多个模块一起运动. 同样, Telecubes 模块使用的驱动器 Maxon EC06/GP06, 最多可以同时推动 3 个模块运动. 所以从实际的物理约束出发, 两个伸缩型模块的组合可以在构型表面做滑动运动. 同时, 其他一些类型的模块, 如 MIT 提出的 EM-Cube [60] 等模块也采用滑动方式运动. 本小节主要介绍滑动类型模块的自重构路径规划方法.

这里模块的滑动运动是指模块只允许在其他模块的表面上滑动而不能进行翻转运动. 在这种滑动运动的限制下, 一个模块只能在机器人外形的凹形表面上做往返运动, 如图 25 所示. 注意到如果模块化机器人的构型退化为一维或者二维形状, 因为模块不能进行翻转 (即不能同时改变 x, y 和 z 3 个坐标), 所以再也不可能重构出一个三维构型. 因此, 需要设计特殊的自重构路径规划方法.

针对滑动类型模块, 日本电信电话公司通信科学实验室的研究人员 [61] 提出了一种基于三维压缩和伸展的自重构路径规划方法. 为了描述模块的位置, 首先定义 Descartes 坐标系, X 和 Y 轴在水平面上, Z 轴垂直于水平面, 每个模块由坐标 (x_i, y_i, z_i) 唯一表示. 所有模块被分为 3 类: 壁、杆和体.

- 设置常数 x_{thresh} 和 y_{thresh}, 使得对于初始构型 S 中的所有模块有 $x_i \leqslant x_{\text{thresh}}, y_i \leqslant y_{\text{thresh}}$;
- 壁由满足如下条件的模块组成: $x_i = x_{\text{thresh}}$, $y_i < y_{\text{thresh}}$;
- 杆由满足如下条件的模块组成: $x_i = x_{\text{thresh}}$, $y_i = y_{\text{thresh}}$;
- 体由满足如下条件的模块组成: $x_i < x_{\text{thresh}}$, $y_i < y_{\text{thresh}}$.

算法依赖于一种三维压缩操作, 它可以视为沿着 xy 方向挤压构型, 直到构型中所有的孔洞都被挤出, 成为实心形状. 具体操作如下: 首先建立两个辅助面 $X_{\text{wall}}(x)$ 和 X_{final}, $X_{\text{wall}}(x)$ 是通过 X 轴上坐标为 x 的点, 并垂直于 X 轴的平面, X_{final} 是 $X_{\text{wall}}(x_{\text{thresh}}+1)$ 平面. 将 x 遍历 $x_{\min}$ 与 x_{thresh} 间

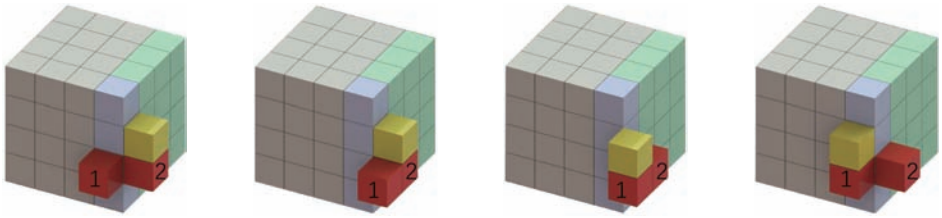


图 26 (网络版彩图) 种子模块帮助其他模块从一平面运动至另一平面 [62]. 红色为种子模块, 黄色为待转运工作模块

Figure 26 (Color online) The seed modules (red) help work module (yellow) move to another surface [62]. (a) Initial state. Seed module 1 is on the target surface while seed module 2 and work module are on the original surface; (b) seed module 1 slides on the target surface to connect the seed module 2; (c) work module slides from seed module 2 to 1; (d) seed module 1 and work module slide together to target surface

的所有值, 得到一系列的辅助面 $X_{\text{wall}}(x)$. 使用这些辅助面将所有模块往 x 轴正方向移动直至没有空隙. 同理, 在 y 方向上也执行同样的压缩操作. 压缩完成后, 构型中没有孔洞等空隙. 此外, 需利用记录的压缩过程以便后续进行逆操作实现压缩前构型的恢复.

基于三维压缩操作, 算法的整体思路如下. 在初始构型 S 和目标构型 G 之间构造 3 种三维中间构型 M_s, M_p 和 M_g. (1) M_s 为 S 经过三维压缩操作后的构型, 从 S 到 M_s 的压缩的过程记为 $S \to M_s$; (2) M_g 为 G 经过三维压缩操作后的构型, 记录压缩过程 $G \to M_s$, 并取逆过程 $M_s \to G$ 用于后续的伸展操作; (3) M_p 是从 M_s 到 M_g 的过渡构型. 基于这 3 种中间构型, 自重构方法为先将输入模型 S 压缩为 M_s, 再对 M_s 进行置换操作并重构 M_p 构型, 再重构变换为 M_g, 最后通过伸展操作变换为目标构型 G. 具体的过渡过程 $M_p \leftarrow M_s \leftarrow M_g$ 需要借助一种构型变换操作: 如图 26 所示, 首先选择两个模块作为种子模块, 一个种子模块在杆所在平面上, 另一个种子模块在壁所在平面上, 种子模块用于帮助其他模块从一个平面移动到另一平面, 弥补模块不能进行边界上翻转的缺陷. 然后按照体、壁、杆的顺序选择其中的模块借助种子模块运动到目标位置, 目标位置的选择原则为: 若目标构型中的杆部分尚未完成则移动填充至杆部分, 依次为壁部分和体部分, 构型变换完成后, 将种子模块还原至构型内部.

文献 [61] 中证明, 只要所有的三维中间构型包括 M_s, M_p 和 M_g 需要满足以下 4 个条件, 则基于三维压缩和伸展的自重构路径一定存在.

(1) 构型中任何一个模块 $x_i \leqslant x_{\text{thresh}}$, $y_i \leqslant y_{\text{thresh}}$;

(2) 对于任意一个模块 $x_i < x_{\text{thresh}}$, 均存在与之相邻的模块满足 X 坐标为 $x_i + 1$;

(3) 对于任意一个模块 $y_i < y_{\text{thresh}}$, 均存在与之相邻的模块满足 Y 坐标为 $y_i + 1$;

(4) 所有满足 $x_i = x_{\text{thresh}}$ 和 $y_i = y_{\text{thresh}}$ 的模块构成一个杆 P.

易知通过三维压缩操作得到的 M_s 和 M_g 一定满足这些条件.

6.6 元模块的自重构路径规划

6.4 小节介绍了伸缩型模块的专用自重构算法. 专用算法充分考虑了所研究模块的运动特性, 能够有效提高自重构路径规划方法的效率. 但目前除了 6.3 小节中的通用算法外, 还没有针对边铰接型和中心铰接型模块的专用算法.

奥地利和西班牙的研究人员在文献 [62] 提出了一种巧妙的元模块思想, 可以将少数不是伸缩型的模块通过组合形成一个具有伸缩型模块运动功能的元模块. 基于元模块, 已有的伸缩型模块自重构路



(a) (b) (c)

图 27 (网络版彩图) M-TRAN 模块构成的元模块 [62]

Figure 27 (Color online) The meta-module composed of M-TRAN modules [62]. (a) The expanding state of meta-module, which includes six arms assembled by 12 M-TRAN modules; (b) central blocks; (c) the contracting state of meta-module

径规划算法 (包括融化 – 生长算法 [8] 或 PacMan 算法 [59]) 都可以用来实现自重构. 这种基于元模块的方法适用于边铰接型模块, 如 M-TRAN 系列、SuperBot、SMORES 等, 也适用于中心铰接型模块, 如 Molecube 模块和 Roombots.

在数据库领域, 元数据 (meta-data) 和元模型 (meta-model) 是指关于数据的数据和关于模型的模型. 元模块 (meta-module) 借鉴了这种 “元” 的思想. 对于边铰接型模块而言, 以 M-TRAN 为例, 如图 27(a) 所示, 使用 12 个 M-TRAN 模块组成的元模块在伸展收缩的时候不会发生碰撞并且始终保持连接. 该元模块能实现伸展、收缩、连接、分离等伸缩型模块所具有的全部功能. 该元模块由 6 个臂组成, 分别平行于 x, y 和 z 轴. 每个臂由两个模块组成, 在收缩过程中, 每个臂两端的模块需保持各自的朝向与伸展时相同. 对于中心铰接型模块而言, 与边铰接型类似, 同样包含 6 个臂, 分别平行于 x, y 和 z 方向, 它们在伸展收缩的时候不会发生碰撞并且始终保持连接. 同样, 该元模块具有伸缩型模块的全部功能, 但其伸展和收缩的过程不能在一个平面中完成.

如 6.5 小节所述, 伸缩型模块有着其自身的物理限制, 比如驱动器的功率不足无法在实际操作中同时推动算法所要求的模块个数时, 另一种思路是将少数 4 个伸缩型模块再次组成元模块, 该元模块具有滑动型模块的所有功能. 通过调用 6.5 小节中的滑动型模块自重构算法, 可以实现在物理约束条件下的模块化机器人自重构.

7 存在的问题、挑战与可能的发展方向

模块化机器人的自重构路径规划方法研究, 还处于方兴未艾的阶段. 基于前文的综合介绍, 本节从关键理论研究和基于硬件约束的算法设计两个方面, 进一步指出一些存在的问题和挑战, 为自重构路径规划方法研究方法的发展提供一些思路.

(1) 关键理论研究. 模块化机器人自重构路径规划算法的复杂度, 与构型空间大小和规划问题是否 NP 难密切相关. 第 5 节中对构型空间大小和理论下界估计进行了讨论. 但目前已有的这些工作, 出于数学建模的考虑, 都对所研究的模块类型和运动方式进行了简化. 本文在 7.2 和 7.1 小节指出两个这方面的挑战, 具体包括 (a) 构型空间的紧下界与更有效的启发式搜索策略和 (b) 逐个移动式自重构路

径规划是否 NP 难.

(2) 基于硬件约束的算法设计. 模块化机器人的自重构与单个模块所能实现的运动密切相关 (第 2 节), 它取决于模块的硬件结构 (第 3 节). 目前已有的自重构路径规划方法研究 (第 6 节), 对不同模块硬件结构带来的特殊性考虑还不充分, 本文在 7.3, 7.4 和 7.5 小节指出这方面的 3 个挑战, 具体包括满足物理约束的自重构路径规划、适用于 M-TRAN 系列特殊运动方式的自重构路径规划方法研究和柔性模块化机器人研究.

7.1 构型空间的紧下界与更有效的启发式搜索策略

5.3 小节中已经介绍并讨论了模块机器人构型空间的理论下界, 但这些已有的理论下界值只考虑了模块的相对位置和方向, 而没有考虑模块中连接器的位置和可能具有的不同性别类型 (比如 M-TRAN 模块中分为公、母两个子模块). 将这些因素一并考虑的时候, 机器人的构型数目会显著增多. 不同于已有基于简化模块模型的理论下界值, 尝试给出真实模块模型 (包括考虑连接器位置和性别等更多实际因素) 构型空间的紧下界值, 是一个有待探索的问题.

考虑了更多包括连接器位置和性别等实际因素后计算出的构型空间理论下界值, 明显会比 5.3 小节中简化模型给出的下界值更大. 这在原本就规模庞大的构型空间中寻找从初始构型到目标构型中的转换难题上又新添了更大的难度, 因此寻找更有效的启发式策略就十分必要, 希望能够更有效地剪枝、过滤启发式搜索中可能存在的大量无效搜索路径, 从而进一步提高搜索效率.

7.2 逐个移动式自重构路径规划是否 NP 难?

6.2 小节中论述总结了链式模块化机器人的自重构路径规划问题是 NP 难问题. 但链式自重构的自由度和复杂度明显高于逐个移动式自重构, 并且逐个移动式自重构这种方式更能满足常见模块的各种物理约束, 因此回答逐个移动式自重构路径规划是否 NP 难具有非常重要的意义. 如果答案是否定的, 那么逐个移动式自重构优化问题就存在多项式时间内的全局最优解, 这将对路径规划方法研究产生深刻的影响.

如果我们将链的长度约束到 1, 那么逐个移动式自重构可以看成是链式模块化机器人自重构的一个特例. 在理论计算机科学领域, 一个为人们所熟知的事实是一个 NP 难问题的特例不一定也是 NP 难. 比如经典的旅行推销员问题是著名的 NP 难问题, 但是它有多个很有意义的特例都是 P 问题 [63]. 但是它的逆命题是成立的, 即如果一个问题的特例是 NP 难, 那么这个问题本身是 NP 难的. 所以, 如果能够证明逐个移动式自重构优化问题是 NP 难, 那么它也对应的是链式模块化机器人自重构是 NP 难的一个新证明.

7.3 满足物理约束的自重构路径规划

第 6 节中介绍的大多数路径规划方法, 并没有完全的考虑实际使用模块所受到的物理约束, 因而在实际应用中会因为存在物理约束使得原有的路径规划方法并不适用. 举例来说, 由于多数模块本身的驱动器功率、旋转角度或连接器强度等设计原因, 在物理上只能从一个位置运动至邻近的部分位置, 而无法在任意方向上移动, 或者像 6.4 小节中伸缩型模块那样带动任意多个模块一起运动. 此外, 如果模块移动过程中所经过某些位置的相邻位置被其他模块占用, 则会使本身具有运动能力的模块无法移动或者需要经过更复杂的运动规划实现移动. 6.6 小节中介绍的元模块是解决这些问题的一个思路, 因为相比于单个模块的自由度, 元模块由于含有多个模块而增大了运动自由度, 这些额外增加的自由度

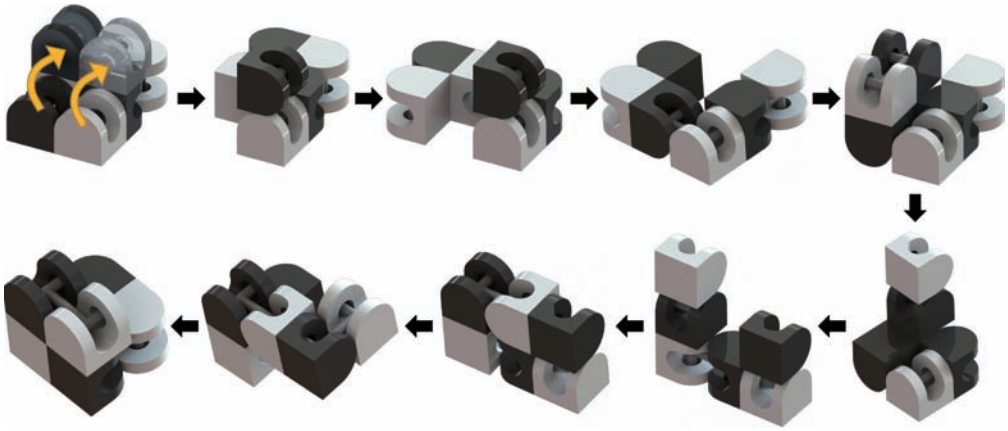


图 28 (网络版彩图) 一个由 4 个 M-TRAN 模块组成的机器人自重构过程. 左上角所示的是初始构型, 左下角所示的是目标构型. 初始构型与目标构型间的差异用橙色箭头指出, 只有一个模块的位置发生了变化. 仅仅调整这样一个模块的位置, M-TRAN 机器人就需要进行 9 步移动, 过程非常繁琐 [61]

Figure 28 (Color online) An example of self-reconfiguration of 4 M-TRAN modules. Upper-left is original shape and lower-left is the target shape. The difference between them is shown by orange arrow. It needs 9 moves to change only one module's position [61]

在一定程度上减少了运动限制, 简化自重构问题. 例如, 单个伸缩型模块驱动器功率不足以在实际自重构过程同时推动多个模块来实现算法, 则可以采用将少数几个伸缩型模块组成元模块, 该类型元模块满足 6.5 小节中提到滑动型模块的运动要求, 通过使用滑动型模块的自重构算法, 可以实现机器人的自重构. 此外, 针对专门的模块硬件约束来设计专用的路径规划方法也是将来可能的一个发展方向.

在实际应用中, 不仅需要考虑来自机器人模块本身存在的物理限制, 也需要考虑来自机器人所处工作环境的外部限制. 模块化机器人一个重要特征是需要根据环境任务的不同改变构型, 因此复杂外部环境的约束也是模块化机器人在自重构路径规划中的一个重要考虑因素.

举例来说, 在外太空或深水作业环境中, 重力的影响可以忽略不计. 在这些特殊环境中, 驱动器功率不足的问题并不存在, 而模块化机器人自重构过程中的连接 – 分离 – 重连接操作是能耗的最主要来源. 在此条件下, 为了节约能源及减小构型变换的复杂性, 自重构路径规划问题应重点考虑如何使用最少次数的连接 – 分离 – 重连接操作实现构型变换. 另外一个可能的应用场景是当外界工作环境是崎岖不平的路面时, 模块化机器人在自重构过程中, 不仅需要考虑每个模块移动的自身运动, 同时还要考虑每个中间状态对整个机器人的重心造成的影响, 避免侧翻等意外事故的发生; 当工作空间非常狭小时, 同时需要考虑自重构过程如何尽可能的在模块化机器人内部进行, 这样可以尽可能少地占用外部空间.

7.4 适用于 M-TRAN 的自重构路径规划算法

2.3 小节中提到的 M-TRAN 系列是一类在学术界和工业界都受到广泛关注的重要双子型模块. 如图 2 所示, 该模块由公、母两个子模块和一个连接件组成, 模块与模块之间的连接器也有公母的性别之分. 双子型模块类型中的其他模块如 SuperBot, iMobot 等, 它们的运动方式都很类似. 在 7.1 小节中, 我们已经说明考虑 M-TRAN 这样的具有公母子模块的机器人类型会使得构型空间大小的理论下界显著提高. 图 28 给出了一个由 4 个 M-TRAN 模块组成的机器人自重构过程, 其中橙色箭头从模块的原始位置指向目标位置. 由图中可以看出, 即使是简单的改变 4 个模块中的 1 个模块位置, 其所需

步骤也非常繁琐. 因此有必要像伸缩型模块 (6.4 小节) 和滑动模块类型 (6.5 小节) 一样, 考虑满足实际物理约束的专用自重构路径规划算法.

目前已有的专门针对伸缩型模块和滑动类型模块的自重构路径规划算法. 由于模块的运动方式不同, 无法直接适用于 M-TRAN 模块. 6.6 小节中介绍的元模块自重构方法, 将 12 个 M-TRAN 模块封装成一个伸缩型元模块, 从而可以利用滑动类型模块的自重构路径规划算法. 该方法虽然可行, 但仍然没有充分考虑 M-TRAN 模块的自身所特有的运动方式, 专门针对 M-TRAN 模块来设计自重构路径规划算法是一个迫切需要解决的问题.

7.5 柔性自重构模块化机器人

目前所有模块化机器人研究中均使用刚性材料进行制造. 在面对复杂工作环境时, 比如建筑物坍塌受到撞击或运动空间过于狭小时受到挤压, 都会对机器人模块产生由刚性碰撞带来的机械损害, 造成失效. 在机器人研究领域, 越来越多的研究人员注意到生物的柔性与灵活性, 提出了仿生柔性机器人的概念. 模块化机器人的一个将来可能发展方向是与柔性机器人研究结合起来, 在机器人模块的设计上采用柔性材料, 从而能够减轻碰撞所产生的作用力, 提高模块的安全性.

具体而言, 未来可以考虑将具有高强度、高弹性、柔软、防水等特性的柔性材料应用于模块化机器人的制造, 使其能够减弱碰撞带来的损害, 并且可以在各种环境中运动更加灵活. 与此同时, 柔性材料的使用使得模块化机器人的自重构能力进一步增强, 在诸如变形金刚等三维实体变形应用中能够更好地接近模仿的真实物体, 同时将其应用于水下环境探测时, 能够最大程度地实现无干扰观察拍摄等优势.

8 总结

本文围绕自重构模块化机器人的路径规划方法, 从单个模块的运动方式、完成运动所需的硬件结构、多个模块间的组织方式、构型空间表示方法和路径规划算法 5 个层次进行对国际上已有代表性工作进行总结分析.

- **单个模块的运动方式.** 分为中心铰接型、边铰接型、绕边旋转型和伸缩型, 不同的运动方式相应的路径规划算法所需满足的物理约束不同. 中心铰接型模块由相互对称的两部分组成, 以模块的中心为旋转中心, 子模块间能相对旋转; 边铰接型模块旋转轴为子模块相对面的中心连线, 两个子模块均能与连接件相对运动; 绕边旋转型模块一般作为整体做刚体运动, 自重构过程中, 模块间相对旋转的中心轴为模块边线; 伸缩型模块具有伸展、收缩功能, 伸展时长度为收缩时的两倍, 单个模块通过伸缩带动与之相连的模块运动.
- **完成运动所需的硬件结构.** 主要包括驱动器、连接器、传感器、电源和通信元件, 不同的硬件结构决定了不同的运动方式. 驱动器使得单个模块运动, 也可以通过多个模块的驱动器共同作用完成机器人行走等运动及自重构过程; 连接器用于实现模块间的连接与分离, 改变其模块间的连接关系从而改变构型; 传感器用于帮助模块化机器人检测周围环境的变化或者接受指令; 电源用于给机器人供电; 通信元件用于模块间传递与接受信息.
- **多个模块间的组织方式.** 从空间维度上可分为二维和三维两种, 根据机器人自重构过程模块的运动方式可分为链式、逐个移动式和混合式 3 种: 链式自重构机器人在自重构过程中, 局部多个模块形成链状结构并作为整体在机器人中进行移动; 逐个移动式自重构机器人在自重构过程中, 每次只有一

个模块驱动器在工作; 混合式自重构类型机器人在自重构过程中, 可以根据需要选择链式自重构或逐个移动式自重构.

• **构型空间表示.** 由固定数量模块组成的机器人能组成的所有构型. 模块化机器人的形状是指整体机器人的外形, 由其所有组成模块的位置和朝向所决定, 构型则包括形状和拓扑连接关系两部分. 构型的表示方法通常分为加权图表示法和矩阵表示法. 当模块数量较大时, 精确枚举出所有的构型非常困难甚至无法实现, 只能用理论下界值对其做出估计. 随着模块数量的增长, 构型空间可能的大小呈指数级的上升.

• **路径规划算法.** 算法中最为简单直接的是基于图遍历与最短路径搜索的方法, 但模块数量较多时, 构型空间非常大造成图遍历的方法不可行, 因此需要使用启发式搜索来给出具有多项式时间复杂度的近似解, 本文列举了包括伸缩型模块、滑动型模块及元模块等模块类型的自重构路径规划算法. 值得注意的是, 已有的算法中有较多算法并没有充分考虑模块的物理实现, 因此还需要针对不同的模块运动特性, 进一步设计更加高效的自重构算法, 以解决在模块数量较多情况下的模块化机器人自重构问题.

在总结以上已有工作的基础上, 我们从关键理论研究和基于硬件约束的算法设计两个方面, 指出现有工作中存在的不足、问题与挑战, 并提出了今后可能的发展方向, 为研究人员开展进一步的研究提供了思路.

致谢 感谢 Swiss Federal Institute of Technology in Lausanne 生物机器人实验室提供本文中使用的 Roombots 图片, 美国 Massachusetts Institute of Technology 计算机科学与人工智能实验室 Daniela Rus 教授和 John Romanishin 博士提供了 M-Blocks 和 Crystalline 图片, 美国 University of Southern California Polymorphic 机器人实验室 Wei-Min Shen 教授提供了 SuperBot 图片, 美国 University of Pennsylvania GRASP 实验室 Mark Yim 教授提供了 SMORES 和 Telecubes 图片, 日本产业技术综合研究所的 Haruhisa Kurokawa 教授提供了 M-TRAN III 图片, 以及 Technical University of Denmark Henrik Hautop Lund 教授提供了 ATRON 图片.

参考文献

1 Østergaard E H, Christensen D J, Eggenberger P, et al. Hydra: from cellular biology to shape-changing artefacts. In: Proceedings of the 15th International Conference on Artificial Neural Networks, Poland, 2005. 275–281
2 Kurokawa H, Tomita K, Kamimura A, et al. Distributed self-reconfiguration of M-TRAN III modular robotic system. Int J Robot Res, 2008, 27: 373–386
3 Ryland G G, Cheng H H. Design of iMobot, an intelligent reconfigurable mobile robot with novel locomotion. In: Proceedings of IEEE International Conference on Robotics and Automation, Anchorage, 2010. 60–65
4 Murata S, Yoshida E, Kamimura A, et al. M-TRAN: self-reconfigurable modular robotic system. IEEE/ASME Trans Mech, 2002, 7: 431–441
5 Kurokawa H, Kamimura A, Yoshida E, et al. M-TRAN II: metamorphosis from a four-legged walker to a caterpillar. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Las Vegas, 2003. 2454–2459
6 Salemi B, Moll M, Shen W M. SUPERBOT: a deployable, multi-functional, and modular self-reconfigurable robotic system. In: Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems, Beijing, 2006. 3636–3641
7 Romanishin J W, Gilpin K, Rus D. M-blocks: momentum-driven, magnetic modular robots. In: Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems, Tokyo, 2013. 4288–4295

8 Rus D, Vona M. Crystalline robots: self-reconfiguration with compressible unit modules. Auton Robot, 2001, 10: 107–124

9 Rus D, Vona M. A physical implementation of the self-reconfiguring crystalline robot. In: Proceedings of the IEEE International Conference on Robotics and Automation, San Francisco, 2000. 1726–1733

10 Jorgensen M W, Ostergaard E H, Lund H H. Modular ATRON: modules for a self-reconfigurable robot. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Sendai, 2004. 2068–2073

11 Zykov V, Chan A, Lipson H. Molecubes: an open-source modular robotics kit. In: Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems Workshop, Self-Reconfigurable Robotics, San Diego, 2007. 3–6

12 Sproewitz A, Billard A, Dillenbourg P, et al. Roombots-mechanical design of self-reconfiguring modular robots for adaptive furniture. In: Proceedings of IEEE International Conference on Robotics and Automation, Kobe, 2009. 4259–4264

13 Spröwitz A, Moeckel R, Vespignani M, et al. Roombots: a hardware perspective on 3D self-reconfiguration and locomotion with a homogeneous modular robot. Robot Auton Syst, 2014, 62: 1016–1033

14 Davey J, Kwok N, Yim M. Emulating self-reconfigurable robots-design of the SMORES system. In: Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems, Vilamoura, 2012. 4464–4469

15 Romanishin J W, Gilpin K, Claici S, et al. 3D M-Blocks: self-reconfiguring robots capable of locomotion via pivoting in three dimensions. In: Proceedings of IEEE International Conference on Robotics and Automation, Seattle, 2015. 1925–1932

16 Suh J W, Homans S B, Yim M. Telecubes: mechanical design of a module for self-reconfigurable robotics. In: Proceedings of the IEEE International Conference on Robotics and Automation, Washington, 2002. 4095–4101

17 Ikuta K. Micro/miniature shape memory alloy actuator. In: Proceedings of the IEEE International Conference on Robotics and Automation, Cincinnati, 1990. 2156–2161

18 Yoshida E, Kokaji S, Murata S, et al. Miniaturization of self-reconfigurable robotic system using shape memory alloy actuator. J Robotic Mech, 2000, 12: 96–102

19 Yoshida E, Murata S, Kokaji S, et al. Micro self-reconfigurable modular robot using shape memory alloy. J Robotic Mech, 2001, 13: 212–218

20 Stoy K, Brandt D, Christensen D J, et al. Self-Reconfigurable Robots: an Introduction. Cambridge: MIT Press, 2010. 63–91

21 Yim M, Zhang Y, Roufas K, et al. Connecting and disconnecting for chain self-reconfiguration with PolyBot. IEEE/ASME Trans Mech, 2002, 7: 442–451

22 Yim M, Eldershaw C, Zhang Y, et al. Self-reconfigurable robot systems: PolyBot. J Robotic Soc Jpn, 2003, 21: 851–854

23 Shen W M, Kovac R, Rubenstein M. SINGO: a single-end-operative and genderless connector for self-reconfiguration, self-assembly and self-healing. In: Proceedings of IEEE International Conference on Robotics and Automation, Kobe, 2009. 4253–4258

24 Tosun T, Davey J, Liu C, et al. Design and characterization of the EP-Face connector. In: Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems, Deajeon, 2016. 45–51

25 Karagozler M E, Campbell J D, Fedder G K, et al. Electrostatic latching for inter-module adhesion, power transfer, and communication in modular robots. In: Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems, San Diego, 2007. 2779–2786

26 Moeckel R, Jaquier C, Drapel K, et al. Exploring adaptive locomotion with YaMoR, a novel autonomous modular robot with Bluetooth interface. Ind Robot, 2006, 33: 285–290

27 Castano A, Shen W M, Will P. CONRO: towards deployable robots with inter-robots metamorphic capabilities. Auton Robot, 2000, 8: 309–324

28 Murata S, Kurokawa H, Kokaji S. Self-assembling machine. In: Proceedings of the IEEE International Conference on Robotics and Automation, San Diego, 1994. 441–448

29 Tomita K, Murata S, Kurokawa H, et al. Self-assembly and self-repair method for a distributed mechanical system. IEEE Trans Robotic Autom, 1999, 15: 1035–1045

30 Yim M. New locomotion gaits. In: Proceedings of the IEEE International Conference on Robotics and Automation,

San Diego, 1994. 2508–2514
31 Zhao J, Cui X D, Zhu Y H, et al. A new self-reconfigurable modular robotic system UBot: multi-mode locomotion and self-reconfiguration. In: Proceedings of IEEE International Conference on Robotics and Automation, Shanghai, 2011. 1020–1025
32 Zhao J, Tang S F, Zhu Y H, et al. A modular self-reconfigurable robot based on universal joint. Robot, 2010, 32: 608–613 [赵杰, 唐术锋, 朱延河, 等. 基于万向式关节的模块化自重构机器人. 机器人, 2010, 32: 608–613]
33 Jing G Y, Tosun T, Yim M, et al. An end-to-end system for accomplishing tasks with modular robots. In: Proceedings of Robotics Science and Systems, Michigan, 2016
34 Mondada F, Pettinaro G C, Guignard A, et al. SWARM-BOT: a new distributed robotic concept. Auton Robot, 2004, 17: 193–221
35 Rubenstein M, Cornejo A, Nagpal R. Programmable self-assembly in a thousand-robot swarm. Science, 2014, 345: 795–799
36 Rybski P E, Larson A, Veeraraghavan H, et al. Performance evaluation of a multi-robot search & retrieval system: experiences with MinDART. J Intell Robot Syst, 2008, 52: 363–387
37 Sastra J, Bernal-Heredia W G, Clark J, et al. A biologically-inspired dynamic legged locomotion with a modular reconfigurable robot. In: Proceedings of ASME Dynamic Systems and Control Conference, Michigan, 2008. 1467–1474
38 Østergaard E H, Kassow K, Beck R, et al. Design of the ATRON lattice-based self-reconfigurable robot. Auton Robot, 2006, 21: 165–183
39 Wei H X, Wang T M. Configuration analysis and self-assembly control for modular swarm robots. J Mech Eng, 2010, 46: 100–108 [魏洪兴, 王田苗. 模块化群体机器人构型分析与自组装控制. 机械工程学报, 2010, 46: 100–108]
40 Rubenstein M, Nagpal R. Kilobot: a robotic module for demonstrating behaviors in a large scale (2^{10} units) collective. In: Proceedings of the IEEE International Conference on Robotics and Automation Workshop, Modular Robotics: State of the Art, Anchorage, 2010. 47–51
41 Stoy K. The deformatron robot: a biologically inspired homogeneous modular robot. In: Proceedings of the IEEE International Conference on Robotics and Automation, Orlando, 2006. 2527–2531
42 Zhang Y H, Zhao J, Zhang L, et al. Novel modular self-reconfigurable robot system. J Mech Eng, 2006, 42: 175–178 [张玉华, 赵杰, 张亮, 等. 新型模块化可重构机器人系统. 机械工程学报, 2006, 42: 175–178]
43 Chirikjian G S. Kinematics of a metamorphic robotic system. In: Proceedings of the IEEE International Conference on Robotics and Automation, San Diego, 1994. 449–455
44 Pamecha A, Chiang C J, Stein D, et al. Design and implementation of metamorphic robots. In: Proceedings of the ASME Design Engineering Technical Conference and Computers in Engineering Conference, Irvine, 1996
45 Mondada F, Pettinaro G C, Guignard A, et al. SWARM-BOT: a new distributed robotic concept. Auton Robot, 2004, 17: 193–221
46 Kotay K, Rus D, Vona M, et al. The self-reconfiguring robotic molecule. In: Proceedings of the IEEE International Conference on Robotics and Automation, Leuven, 1998. 424–431
47 Yim M, Duff D G, Roufas K D. PolyBot: a modular reconfigurable robot. In: Proceedings of the IEEE International Conference on Robotics and Automation, San Francisco, 2000. 514–520
48 Gilpin K, Kotay K, Rus D, et al. Miche: modular shape formation by self-disassembly. Int J Robot Res, 2008, 27: 345–372
49 Stoy K, Brandt D. Efficient enumeration of modular robot configurations and shapes. In: Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems, Tokyo, 2013. 4296–4301
50 Harary F. Unsolved problems in the enumeration of graphs. Publ Math Inst Hungar Acad Sci, 1960, 5: 63–95
51 Eden M. A two-dimensional growth process. In: Proceedings of the 4th Berkeley Symposium on Mathematics, Statistics, and Probability. Berkeley: University of California Press, 1961. 223–239
52 Klarner D A, Rivest R L. A procedure for improving the upper bound for the number of n-ominoes. Canad J Math, 1973, 25: 585–602
53 Cormen T, Leiserson C, Rivest R. Introduction to algorithms. Cambridge: MIT Press, 1990. 527–531
54 Hou F, Shen W M. On the complexity of optimal reconfiguration planning for modular reconfigurable robots. In: Proceedings of IEEE International Conference on Robotics and Automation, Anchorage, 2010. 2791–2796

55 Michael R G, David S J. Computers and intractability: a guide to the theory of NP-completeness. B Am Math Soc, 1980, 3: 898–904
56 Russell S, Norvig P. Artificial Intelligence: a Modern Approach. Egnlewood Cliffs: Prentice-Hall, 1995. 25–27
57 Pamecha A, Ebert-Uphoff I, Chirikjian G S. Useful metrics for modular robot motion planning. IEEE Trans Robotic Autom, 1997, 13: 531–545
58 Papadimitriou C H, Steiglitz K. Combinatorial Optimization: Algorithms and Complexity. Mineola: Dover Publications, 1998. 248–255
59 Butler Z, Byrnes S, Rus D. Distributed motion planning for modular robots with unit-compressible modules. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Maui, 2001. 790–796
60 An B K. Em-cube: cube-shaped, self-reconfigurable robots sliding on structure surfaces. In: Proceedings of IEEE International Conference on Robotics and Automation, Pasadena, 2008. 3149–3155
61 Kawano H. Full-resolution reconfiguration planning for heterogeneous cube-shaped modular robots with only sliding motion primitive. In: Proceedings of IEEE International Conference on Robotics and Automation, Stockholm, 2016. 5222–5229
62 Parada I, Sacristán V, Silveira R I. A new meta-module for efficient reconfiguration of hinged-units modular robots. In: Proceedings of IEEE International Conference on Robotics and Automation, Stockholm, 2016. 5197–5202
63 Burkard R E, Deineko V G, van Dal R, et al. Well-solvable special cases of the traveling salesman problem: a survey. SIAM Rev, 1998, 40: 496–546

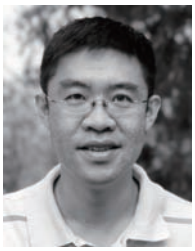
Path planning for self-reconfigurable modular robots: a survey

Yong-Jin LIU*[1,2,†], Minjing YU[1,†], Zipeng YE[1] & Charlie C.L. WANG[3]

1. *Department of Computer Science and Technology, Tsinghua University, Beijing* 100084*, China*;
2. *Tsinghua National Laboratory for Information Science and Technology, Beijing* 100084*, China*;
3. *Department of Design Engineering & TU Delft Robotics Institute, Delft University of Technology, Delft* 2600 AA*, The Netherlands*
* Corresponding author. E-mail: liuyongjin@tsinghua.edu.cn
† Equal contribution

Abstract Self-reconfigurable modular robots (SRMRs) are a special type of robots that can change their shapes and functions according to different tasks and environments. Such a robot is usually constructed using connected modules, each of which can encapsulate a simple function independently and also communicate with each other. Complex tasks can be completed by those connected modules collaboratively. In recent years, SRMRs have attracted considerable attention from both the academia and industry because of their versatility and flexibility. The path planning problem for the transformation of an SRMR is an important but not a well-solved problem, which can be considered as finding an optimal path in the configuration space where every point represents a feasible configuration of the SRMR. To provide a systematic overview of this research, we review the existing approaches considering five different aspects of SRMRs, including the type of motion on a single module, hardware for different motions, connectivity between modules, representation of a configuration space, and path planning algorithms. Aiming at motivating more research into SRMRs, the problems in existing approaches are analyzed and challenges in future work are summarized at the end of this paper.

Keywords modular robots, self-assembly, path searching, NP-complete problems, approximation algorithms, computational geometry



Yong-Jin LIU was born in 1977. He received his B.Eng. degree from Tianjin University, China, in 1998, and Ph.D. degree from Hong Kong University of Science and Technology, Hong Kong, China, in 2004. He is currently an associate professor with the TNList, Department of Computer Science and Technology, Tsinghua University, China. His research interests include computational geometry, modular robots, pattern analysis, computer graphics, and computer vision.



Zipeng Ye was born in 1996. He is currently a Ph.D. student with TNList, Department of Computer Science and Technology, Tsinghua University, China. He received his B.Eng. degree from Tsinghua University, China, in 2017. His research interests include computational geometry, modular robot, and computer graphics.



Minjing YU was born in 1991. She is currently a Ph.D. student with TNList, Department of Computer Science and Technology, Tsinghua University, China. She received her B.Eng. degree from Wuhan University, China, in 2014. Her research interests include modular robot, cognitive computation, computer graphics, and computer vision.



Charlie C.L. Wang was born in 1975. He received his Ph.D. degree in mechanical engineering from Hong Kong University of Science and Technology in 2002. Currently, he is a full professor and the chair of advanced manufacturing at Delft University of Technology, The Netherlands. His research interests include geometric computing, computational design, robotics, and advanced manufacturing.