Artificial Intelligence 223 (2015) 27–64

Contents lists available at ScienceDirect

Artificial Intelligence

www.elsevier.com/locate/artint

Modular robotic systems: Methods and algorithms for abstraction, planning, control, and synchronization

Hossein Ahmadzadeh, Ellips Masehian *

Tarbiat Modares University, Jalalé-Alé-Ahmad Highway, Tehran, 14115-143, Iran

ARTICLE INFO

Article history:
Received 6 June 2014
Received in revised form 5 January 2015
Accepted 6 February 2015
Available online 12 February 2015

Keywords:
Modular robotic systems
Solution methods and algorithms
Self-reconfiguration
Gait
Flow
Self-assembly

ABSTRACT

While expected applications of Modular Robotic Systems (MRS) span various workspaces, scales, and structures, practical implementations of such systems lag behind their potentials in performing real-world tasks. Challenges of enhancing MRS capabilities not only are limited to designing reliable, responsive, and robust hardware, but also include developing software and algorithms that can effectively fulfill tasks through performing fundamental functions like shape-formation, locomotion, manipulation, etc. Thus, MRS solution methods must be able to resolve problems arising from the tightly-coupled kinematics of interconnected modules and their inherent limitations in resources, communication, connection strength, etc. in performing such functions through domain-specific operations including *Self-reconfiguration*, *Flow*, *Gait*, *Self-assembly*, *Self-disassembly*, *Self-adaptation*, *Grasping*, *Collective actuation*, and *Enveloping*. Despite the large number of developed solution methods, there is no inclusive and updated study in the literature dedicated to classifying, analyzing, and comparing their specifications and capabilities in a systematic manner. This paper aims to fill in this gap through reviewing 64 solution methods and algorithms according to their application in each operation and by investigating their capabilities in (1) modeling and simplifying MRS problems through *Abstraction methods*, (2) solving MRS problems through *Solution* and *Control methods*, and (3) coordinating actions of modules through *Synchronization methods*. Challenging issues of each solution approach along with their advantages and weaknesses are also analyzed and open problems and improvement outlooks are mentioned. Overall, this paper aims to investigate the research areas in MRS algorithms that have been evolved so far and to explore promising research directions for the future.

1. Introduction

Robots were invented with the vision of helping humans do their tasks, especially 4D (Dirty, Dangerous, Difficult, and Dull) tasks, more comfortably. The conventional approach in designing robots has been to design their hardware and software in conformance with the tasks they are supposed to do. Conventional robots can perform certain tasks accurately, however they are not very flexible and adaptive, and thus applications consigned to them heavily depend to their physical structure on the one hand and their controller capabilities on the other hand. The concept of morphologically variable robots firstly was introduced by Toshio Fukuda in 1985 with the name CEBOT (an abbreviation for 'cellular robotic system'),

* Corresponding author.
E-mail address: masehian@modares.ac.ir (E. Masehian).

http://dx.doi.org/10.1016/j.artint.2015.02.004

a name which was later changed to *Modular Robotic System*, or *Modular Robot* [55,54]. Modular robots were introduced as a remedy to flexibility and adaptability limitations of fixed-body, monolithic robots, comprising a class of robotic systems that can change their shape in conformance to the task and environmental conditions through assuming various morphologies. A modular robot consists of several units with few degrees of freedom (DOFs) called *modules* which are usually equipped with connection mechanisms to cooperatively connect to or detach from each other in order to create complex structures and configurations with many DOFs such as a snake to slither into narrow tunnels, a loop to traverse flat terrains, or a hexapod to navigate rough terrains. Developing practical Modular Robotic Systems (MRS) is tied with numerous hardware and software challenges which according to recent statements by Stoy and Kurokawa [159] and Fitch et al. [52], must (1) be agile enough to reconfigure in a timely manner, (2) be sufficiently robust and fault-tolerant, (3) be scalable to many-module configurations, (4) be reliable in constructing solid and robust structures, (5) exhibit self-reconfiguration for an extended period of time, (6) deal with uncertainty in the environment and sensed data, (7) be able to handle communication unreliability among modules, and (8) deal with limitations of mechatronic devices in terms of resources, sensing accuracy, actuation power, battery life, etc.

While existing review or survey papers on modular robotics have mainly tackled the architecture and hardware aspects of modular robots, in this paper, we particularly focus on solutions to the challenging problems arisen when developing software components for modular robots. In other words, we study the algorithms and solution methods that have been developed in the context of modular robotics for tackling problems that emerge when modular robots perform tasks through some fundamental functions such as *shape-formation* to form a desired configuration from an initial configuration, *locomotion* for moving from a place to another, *manipulation* for physical interaction with the objects, *supporting* and *balancing* for shoring up unstable objects, etc. Instead of categorizing solution methods to these problems merely by their underlying technical and theoretical aspects, we have organized them according to their contribution toward performing nine basic operations performable by modular robots, namely (1) Self-reconfiguration, (2) Flow, (3) Gait, (4) Self-assembly, (5) Self-disassembly, (6) Self-adaptation, (7) Grasping, (8) Collective actuation, and (9) Enveloping. While there is no guideline that prescribes to consider these operations as basic and underlying, we have deduced such a categorization based on reports on various experiments, success stories, and recommendations in the MRS literature. The rationale behind is that these low-level operations can serve as building blocks for generating high-level behaviors such as shape-formation, locomotion, manipulation, supporting and balancing. For example, reaching a desired shape (shape-formation) can be accomplished through either Self-reconfiguration or Self-assembly basic operations depending on hardware and software capabilities of the modules and the task-specific parameters.

The solution methods for achieving the abovementioned nine operations must address some domain-specific issues that make development of planners/controllers for modular robots very challenging. For example, planning for self-reconfiguration of a modular robot is proved to be NP-complete as it has been reduced to known NP-complete problems like PSAT [65] or 3-PARTITION [78]. Thus, employment of *search-based* methods, which are conventional in Artificial Intelligence, is not straightforward in modular robotics as they need to explicitly represent the state-space and then search it for a solution. In fact, search-based methods usually suffer from intractable configuration-space sizes due to exponential growth of the branching factor in the graph representation of the state-space with the increase of the number of modules. Moreover, the tightly coupled kinematics of the connected modules within a configuration not only limits the number of possible actions of each module, but also urges development of such controllers that avoid taking actions that may lead to undesirable conditions in the structure of modular robot. Examples of undesirable states are *fragmentation* of modular robots into multiple parts, *overcrowding* the structure of modular robot by several modules which intend to enter the same lattice position [196], *hollow* configurations in which modules are trapped in a hole or tunnel within the body, and *solid* configuration in which outer modules cannot find a path toward interior of the robot's body [158]. Such problems can be alleviated to some extent through using *control-based* and *agent-based* approaches that plan for reconfiguration in distributed manner based on local information available in modules. However, their underlying methods must mainly concern with keeping the *connectedness* of the modular robot during reconfiguration steps, considering *convergence* to the desired shape or behavior as a result of local interaction between modules, maintaining *adaptability* to the environmental changes, and exhibiting *robustness* to module failures. These challenges get even worse when modular robots operate in conditions of unpredictable events, sensor noise, uncertainty, and actuator imperfection. Under such circumstances, classical engineering approaches fail to function efficiency, while it can be observed that biological systems, despite their relatively simple interactions, can handle such complex situations efficiently in an autonomous and decentralized manner. Therefore, some *bio-inspired* solution methods are devised in the MRS context motivated from self-organization property of multicellular organisms with the aim of emulating self-organizing behaviors of natural systems by modular robots. Overall, the abovementioned challenges have been treated by various solution methods that can be studied under general categories of search-based, control-based, agent-based, bio-inspired, and other intelligent approaches.

Despite the large number of algorithms devised for solving MRS-related problems, there is no explicit and comprehensive categorization of methods in the literature for identifying their specifications, strengths, weaknesses, as well as application contexts and related challenges. Existing reviews, surveys, or books on modular robotics have mainly tackled the architecture, mechanical, and hardware aspects of modular robots, with less emphasis on implemented solution methods. Murata and Kurokawa [117] studied architecture of modular robots and formally classified them into *lattice*, *chain*, and *hybrid* classes. Yim et al. [195] studied modular robots merely from the hardware point of view, and categorized reconfiguration methods according to the source of module motion into *deterministic* and *stochastic* reconfiguration categories. Butler and