ANNUAL REVIEWS

Annual Review of Control, Robotics, and Autonomous Systems

Modular Reconfigurable Robotics

Jungwon Seo,[1] Jamie Paik,[2] and Mark Yim[3]

[1]Department of Mechanical and Aerospace Engineering and Department of Electronic and Computer Engineering, The Hong Kong University of Science and Technology, Hong Kong, China; email: junseo@ust.hk

[2]Reconfigurable Robotics Laboratory, École Polytechnique Fédérale de Lausanne (EPFL), 1015 Lausanne, Switzerland; email: jamie.paik@epfl.ch

[3]General Robotics, Automation, Sensing, and Perception (GRASP) Laboratory, University of Pennsylvania, Philadelphia, Pennsylvania 19104, USA; email: yim@seas.upenn.edu

Annu. Rev. Control Robot. Auton. Syst. 2019. 2:63–88

First published as a Review in Advance on November 2, 2018

The *Annual Review of Control, Robotics, and Autonomous Systems* is online at control.annualreviews.org

https://doi.org/10.1146/annurev-control-053018-023834

Keywords

cellular and modular robots, distributed robot systems, multirobot systems, reconfigurable robots

Abstract

This article reviews the current state of the art in the development of modular reconfigurable robot (MRR) systems and suggests promising future research directions. A wide variety of MRR systems have been presented to date, and these robots promise to be versatile, robust, and low cost compared with other conventional robot systems. MRR systems thus have the potential to outperform traditional systems with a fixed morphology when carrying out tasks that require a high level of flexibility. We begin by introducing the taxonomy of MRRs based on their hardware architecture. We then examine recent progress in the hardware and the software technologies for MRRs, along with remaining technical issues. We conclude with a discussion of open challenges and future research directions.

1. INTRODUCTION

Modular reconfigurable robot (MRR) systems are made up of many repeated modules (or units) that can be rearranged or can rearrange themselves into different configurations depending on the task the robot is to solve at the time. Note that the meaning of the term configuration in reconfigurable robotics (and this article) is commonly generalized to incorporate the connectivity of the modules (which module is connected to which, represented as an adjacency matrix, a linked list, and the like) into the conventional robotics definition of the term, which refers to just the pose of the robot (the full set of the joint angles of the robot). The term reconfiguration thus also refers to the process of changing connectivity.

Modularity in engineering refers to the compartmentalization of elements, often facilitating manufacture, assembly, modification, and repair. Complex systems are often built up from modular blocks. An alternative to this modular approach is designing the whole system together, which can be called an integrated design approach. While integrated systems are often more difficult to design, they can be optimized for higher performance than the modular approach. For robotics, the choice between modular and integrated architectures can have a large impact on the range of application as well as cost or performance.

In product design, modularity can be categorized into three subtypes (1):

- Slot architecture: Each of the interfaces between components is of a different type, so that the various components in the product cannot be interchanged. Many products and most robots are slot-modular. For example, a blow dryer may have modular elements (a heating coil, a frame, and an impeller); a robot arm will have motors, a transmission, rigid links, sensors, and computation.
- Bus architecture: There is a common bus to which the other physical components connect via the same type of interface. A laptop computer with USB ports forms a common interface in which modular elements can be added or rearranged.
- Sectional architecture: All interfaces are of the same type, and there is no single element to which all the other components attach—i.e., there is no base component. The assembly is built up by connecting the components to each other via identical interfaces. Sectional sofas are an example, as are LEGO brick systems.

While many aspects of modularity are applicable to robotics, sectional architectures are the ones that we focus on in modular reconfigurable robotics. The variability in the system is greatly enhanced by sectional modularity. A fourth type of modularity in robotics can be added by considering each robot in a group of disconnected mobile robots (a swarm) as a module. These systems share some of the task allocation issues, but that they are not physically connected greatly changes the ability to apply forces on the environment and changes the constraints for control.

MRR systems have three areas of promise (2):

- Versatility: The systems typically have many redundant degrees of freedom (DOFs) and can adapt their configurations to suit a wide range of tasks.
- Robustness: The redundancy and self-reconfiguration can be used for self-repair, increasing robustness.
- Low cost: Repeated modules mean that economies of scale can be used to reduce the cost of those modules.

Unfortunately, only one of the three—versatility—has been proven to date. The literature includes examples of MRR systems performing hundreds of locomotion and manipulation tasks (3, 4). However, these research prototypes are not very robust, and their costs typically range from several hundred to several thousand dollars (USD) for each module.

MRR systems can be classified into several architectural groups based on the geometrical arrangement of the units:

- Lattice reconfiguration architectures: Lattice reconfiguration architectures have units that are arranged in a regular, three-dimensional pattern, such as a cubic crystal lattice or cannonball packing. These systems exploit this regularity to ease the computational aspects of reconfiguration.
- Chain architectures: Chain architectures are characterized by units that form serial chains. These chains are often connected to form a tree or closed-chain loops. Through articulation, chain architectures can potentially reach any point or orientation in space, and they are therefore more versatile. Generally, however, they are more demanding to represent and analyze computationally and more difficult to control.
- Mobile architectures: Mobile architectures have units that use the environment to maneuver around and can hook up to form complex chains, lattices, or a number of secondary robots that can perform swarm-like behaviors.

A fourth MRR class has started to emerge: reconfigurable trusses. These systems do not fit into the lattice, mobile, or chain style (serial chains are not an inherent part of the system). Instead, prismatic members form parallel truss structures that can reconfigure, changing the topology of the network. In addition, hybrid systems can often exploit the best properties of multiple different architectures.

These architectural groups are elaborated with examples in Section 2, along with trade-offs in design. Sections 3 and 4 discuss recent advances and challenges in the hardware and software technologies for MRRs, respectively. Section 5 discusses open problems in modular reconfigurable robotics. Since this article is intended to provide a review of reconfigurable robotics research for those who are interested in performing research, our discussion focuses on design insights and important research directions, rather than surveying many existing systems and demonstrations. The reader is expected to have a background in engineering and robotics research.

2. OVERVIEW

Dozens of groups have constructed many versions of MRRs, with many approaches for programming them. More than 1,800 papers and a book (3) have been written on reconfigurable robotics at the time of this writing. Many of the early papers were covered in a previous survey (4).

2.1. Architecture

Here, we give examples of MRR systems in each architectural class. The intent is not to be exhaustive, but to provide early representative examples.

2.1.1. Mobile architecture. The first work toward an MRR system can be traced to Fukuda et al. (5) and the cellular robotic system (CEBOT), which can be considered a mobile reconfiguration system. The primary contribution of this work was a probe and drogue docking mechanism that was tested on a wheeled robot driving to and docking with a mating module.

Mobile reconfiguration architectures also take advantage of air and water mobility. One example is a reconfigurable floating structure system called the Tactically Expandable Maritime Platform (TEMP), which was described by Paulos et al. (6). Their work emphasized the systemic issues of dozens of modules attaching in the open seas, featuring large amounts of uncertainty in sensing and control. A recent system called ModQuad (7) has demonstrated midair self-assembly (**Figure 1**). In this case, mechanical docking is enabled by magnets on the corners of a cubic

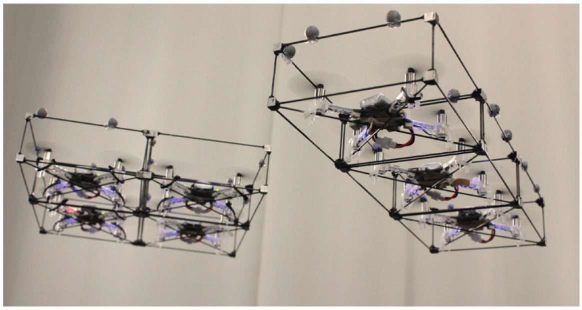

Figure 1

ModQuad: Four modules form a square and three modules form a linear structure, which are assembled midair inside a motion capture environment. Figure reproduced from Reference 7 with permission.

frame around small quad-rotor unoccupied aerial vehicles. Because of the extreme weight budget requirements, the frame and docking mechanisms must be as minimal as possible. The midair assembly of arbitrary planar shapes has been demonstrated, but an undocking mechanism, which would enable disassembly, has yet to be presented.

2.1.2. Lattice reconfiguration architecture. One of the first important works in lattice reconfiguration was the crystalline robots shown in **Figure 2** (8). These robots helped to start the local-rule-based algorithmic approaches grounded in a physical robot with the concomitant constraints that arise from physical systems.

A critical shared aspect of the lattice reconfiguration architectures is that reconfiguration is a local process. When planning motion and determining whether a module can move from one position to another, lattice systems need only check a fixed set of locally neighboring positions

Figure 2

Lattice reconfigurable crystalline robots, which reconfigure by translating modules through expansion. Photo by Daniela Rus's laboratory, reproduced from Reference 8 with permission.

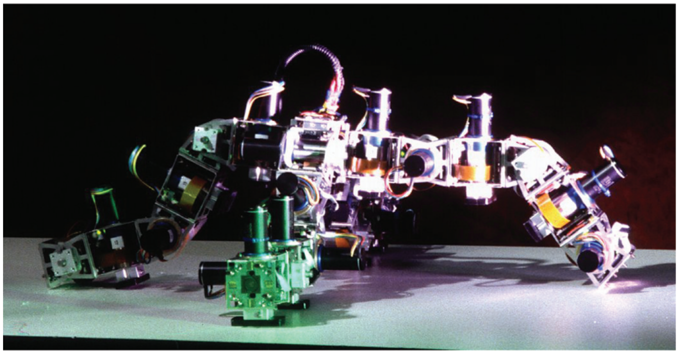

Figure 3

The PolyBot chain reconfiguration system, with 24 modules arranged in a four-legged configuration. Figure reproduced from Reference 9 with permission.

for the existence or absence of modules. All other architectures have global constraints. This implies that the computational time complexity for the reconfiguration of the lattice systems can be independent of the number of modules, which is not the case for other architectures. In addition, the control for each module need only be concerned with providing local motion of one module relative to another, which can simplify module design.

While ModQuad (**Figure 1**) might be another example of a system that forms modules into a lattice, it is different from the lattice reconfiguration architecture in that ModQuad modules become rigidly attached and maneuver through the environment, whereas lattice modules maneuver on other modules.

2.1.3. Chain architecture. The PolyBot system (2), shown in **Figure 3**, is one of the earliest chain architecture systems. It is composed of two types of modules: one that has one articulated DOF with two connection plates, and one that has zero DOFs with six connection plates as the faces of a cube. The one-DOF modules can be attached end to end to form snake-like articulations. The zero-DOF modules are used as structural nodes in tree-like configurations.

Whereas the mobile and lattice architectures do not form serial chains, this architecture can form traditional robot arms with full six-DOF control of the distal end of the arm. As a result, these types of systems have performed a variety of traditional robot arm tasks, whereas the mobile and lattice architectures have focused more on forming different shapes.

2.1.4. Hybrid architectures. One of the most impactful designs in hybrid architectures was the Modular Transformer (M-TRAN) series of systems, shown in **Figure 4** (10). Its module is composed of two linked cubes. These linked cubes can form serial chains for manipulation or locomotion tasks and reconfigure in the same way that lattice systems do. The design was also elegant in the attempt to minimize the required DOFs to achieve both chain tasks and lattice reconfiguration.

Another important hybrid system is the Self-Assembling Modular Robot for Extreme Shape-Shifting (SMORES) system (11), shown in **Figure 5**, which has modules at a lower granular level than M-TRAN: Individual modules are one cube instead of two attached cubes. One significant contribution in the SMORES design is the ability to emulate the function of M-TRAN and most other MRR systems. An important aspect of the design that enables this emulation builds on the observation that many reconfigurable systems are based on a cube shape, can be arranged in a cubic lattice, and are able to rotate about a major axis through the center of the cube.

Figure 4

The Modular Transformer (M-TRAN) III hybrid architecture system, composed of modules with two degrees of freedom (one white and one black element). Photo courtesy of Satoshi Murata, reproduced with permission; copyright 2005 National Institute of Advanced Industrial Science and Technology (AIST).

2.1.5. Truss architecture. A truss reconfiguration system whose nodes are reconfigurable, called the variable topology truss (VTT) system, is shown in **Figure 6** (12). The system can metamorphose by changing the length of the truss members and the degree of each node (the number of the truss members incident to it). The advantage of this architecture is that, similarly to the trusses used in constructing buildings or bridges, the systems can be made very large and strong while being efficient in the material used.

Representing the truss architecture as a graph enables a graph-theoretic analysis computationally. However, the physical constraints that arise from having elements with finite, nonnegligible

Figure 5

Two Self-Assembling Modular Robot for Extreme Shape-Shifting (SMORES) modules attached to emulate one Modular Transformer (M-TRAN) III module. The red and green circles on each face indicate the polarity of the magnets for docking. Figure reproduced from Reference 11 with permission.

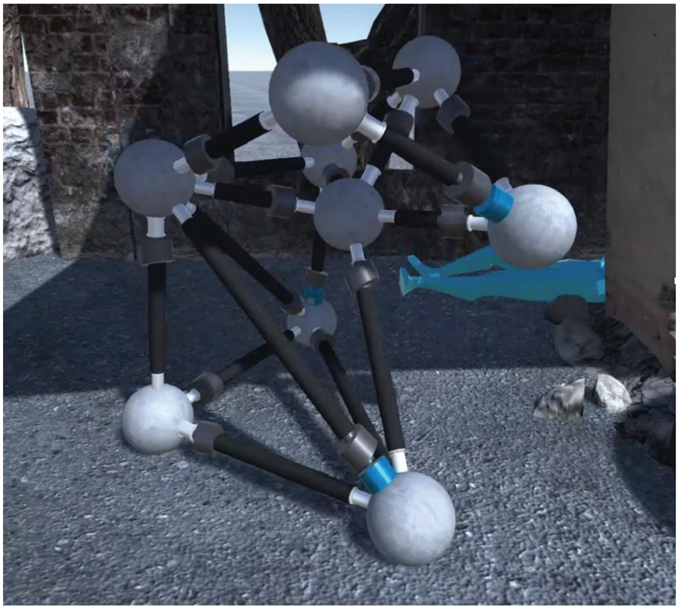

Figure 6

Variable topology truss system with 18 members and 8 nodes. Photo courtesy of Eugene Park, reproduced with permission.

sizes complicate the analysis of these systems. For instance, when the lengths of the members are changed, the angles between the nodes also change; however, the physical size (e.g., diameter) of the members prevents the angle between two members sharing a node from becoming too small. This limitation must be accounted for in any motion planning algorithm. Although the truss architecture has not been considered one of the major classes of reconfigurable systems in previous surveys (13), some relevant early examples are the variable geometry truss systems (14–16), in which truss members are variable but not automatically reconfigurable.

2.2. Trade-Offs in Design

There are a variety of aspects to consider when it comes to designing MRR systems. The hardware and software design issues in MRR systems are highly intertwined, while in conventional robotic systems they are often examined separately. Interestingly, many of the more difficult implementation issues occur on the software side.

2.2.1. Manual reconfiguration versus self-reconfiguration. Automatic self-reconfiguration is the main focus of this article, though many aspects of manual (or human-assisted) reconfiguration systems are the same. Even systems reconfigured by humans can be made self-reconfigurable or robot-assisted by replacing the human with a robot arm.

2.2.2. Generality versus specificity. When thinking about the goals of many autonomous robot systems from an academic viewpoint (versus a purely industrial application), interesting questions arise from the goal of making the systems as general as possible. Academically, the more general or broadly applicable a technology is, the greater its contributions to science and engineering and its impact on the world tend to be. However, industrial applications require proving the ability to achieve specific performance levels. From an MRR point of view, this generality applies to an even larger degree. Since this approach inherently focuses on the building-block level, the promise of versatility overlaps with generality.

2.2.3. Multiplying effect of *n* modules. Determining which features to include in a module has an interesting effect on module design. Robot designers realize that any change to one module is repeated *n* times and thus can have a major impact on a full system. In some cases, this can lead to feature creep, where adding extra features leads to complex modules with high cost and often low reliability.

To counter these trends, the designers of the Claytronics system (17) decided to adopt a minimalistic approach they call the ensemble axiom: A robot should include only enough functionality to contribute to the desired functionality of the ensemble. Their end goal was to create and program very large numbers of modules (millions) at a very small size (less than 1 mm on a side).

2.2.4. Compactness. Another consequence of the multiplying effect of module design results in an increased emphasis on compactness. Tasks often have a characteristic workspace that establishes required robot parameters, such as link lengths. Whereas modules can often be added to satisfy larger workspaces, increasing versatility to apply to tasks in smaller workspaces is limited by the module size. Another way to view this for a given workspace, and a set of modules that span that workspace, is that halving the module size (length of a linear side) enables eight times as many modules to fit in a given volume. This increases the number of DOFs and thus the space of solutions for any given task.

3. HARDWARE AND DESIGN: ADVANCES AND CHALLENGES

One key characteristic of MRRs is the ability to adapt their hardware structure to suit a given task or environment. In this section, we discuss how this poses unique challenges and has affected three major aspects of the hardware design of MRR systems:

- Actuation: While off-the-shelf actuators are prevalent for MRR systems, an increasing number of customized and smart-material-based actuators can be used to overcome geometric design restrictions and address the compactness design goal. For self-reconfiguration, latches can employ dedicated actuators for additional versatility in the assembly of each module.
- Sensing: The ultimate goal of MRR systems is to adapt and react to different environments and tasks. Doing so requires that the modules be aware not only of themselves but also of their assembled state and their interactions with the perceived environment.
- Structure and connection: The ability of modules to attach and detach in the presence of uncertainty is one of the unique aspects of MRR systems compared with traditional robots. Additionally, making a system modular and self-reconfigurable leads to an inherent loss in strength (stiffness, ultimate tensile strength, etc.) as the number of assembled modules increases.

3.1. Actuation

MRR modules typically have two types of actuators: a large actuator, which moves modules relative to other modules or to the environment, and smaller latch actuators, which enable or disable the bond between modules.

A dimensionless measure of the main actuator is the characteristic strength, which is defined as the number of modules that can be supported in a cantilever fashion under gravity. This measure is important because it describes the largest serial chain that a system can support without breaking.

Note that it ultimately also depends on the bonding strength (important for lattice systems) and the module mass. Characteristic strength is typically on the order of 2 to 5.

MRR systems gain task space and capability by increasing the number of modules in a system. Therefore, the power density of the main actuator is critical to achieving a higher characteristic strength and to the ability of the module to support its own weight and apply forces on the environment. MRR systems can use several different actuation technologies, depending on the number and type of joints in each module. Many use off-the-shelf actuators, such as DC motors (18–21), stepper motors (22, 23), and servomotors (19, 24). There are also customized actuators for achieving unconventional movements, including a three-DOF pneumatic actuator (25, 26) and an electromagnetic inertial actuator (27). These actuators provide mobility to an independent module or to the rest of the structure for changing the configuration (21, 23, 27).

Not all motors need to be embedded on the structure: Remote actuation sources have been demonstrated to be effective for launching reconfigurations. Platform setups that induce vibrations (28, 29), a magnetic field (30), or a thermal gradient (31) can also initiate single and/or multiple sequences of motion. While such global actuation induces reconfigurations without limiting the number of controllable modules, controllable DOFs of individual modules in a compact packaging are crucial for practical assembly tasks involving multiple modules. When the power and torque density of the motor are critical for the overall reconfiguration space, customized actuators using functional smart materials could provide new design freedom to MRR systems. In robotics, there are examples of such actuators using electroactive polymers (32), piezoelectric crystals (33, 34), shape memory polymers (35, 36), and shape memory alloys (SMAs) for their high force, torque, or strain-to-weight-ratio characteristics [for example, 1-N force output from a 20-mg actuator (37)]. Various forms of SMA actuators have been applied in lightweight MRR systems.

3.1.1. Shape memory alloy actuators. SMA actuators (most often made of a nickel–titanium alloy) enable some of the most compact actuation mechanisms important for MRR systems. The memory effect of SMA actuators comes from the crystal shape transition between the martensite and austenite states, where each state is defined by a predefined temperature threshold. To activate this phase transition, Joule heating is often used to directly apply the current to the SMA actuators. This is feasible for spring or thin ribbon actuators but is not efficient for other forms of SMA actuators that produce higher forces. Otherwise, an additional customized heater layer that targets a specific zone of the SMA is used (38, 39). MRR systems use various types of SMA actuators, including torsional (40) or folding (41) actuators in addition to the most prevalent spring/string actuators (42). Ribbon-type linear SMA actuators can also be composed of modular blocks called unit cells (43) that can extend, rotate, and bend.

Most linear actuators are created by winding SMA wires to achieve a larger range of motion beyond the typical maximum 5% strain rate. This winding process requires maintaining equal tension on the wire throughout, including during the curing process. An alternative actuator type uses SMA sheets. Due to their planar geometry, they can be fabricated by laser-cut patterning (44) or by etching techniques (45, 46). This process is repeatable, fast, and flexible and creates actuators that can be directly annealed without needing to secure them onto another element. Furthermore, as these actuators are planar, they can easily be integrated into prototypes by adding an actuation layer in combination with a heating layer. This type of SMA can pave the way for an assembly-free fabrication process. The manufacturability of SMA actuators in various forms brings advantages to modular design. Laser-cutting technology enables flexible, plug-and-play, and no-gearbox designs of sheet-type SMA actuator modules (37, 41) for multicrease self-folding robotic origami sheets (47) with minimal mechanical assembly effort. The choice of actuation dictates the mechanical performance of the MRR, while the customizability, compactness, light weight, and high force output make SMA actuators highly attractive for robotic applications. These features are especially beneficial for MRRs, as each module requires the self-contained and integrated design of several mechanisms for motion, coupling, and reconfiguration.

3.1.2. Bonding actuation. Ideally, the connection plates transfer information and power, but at a minimum, they must physically anchor one module to the next. These mechanisms are often hooks that are activated by DC motors (18, 19, 21, 48, 49) or use permanent magnets (11), electromagnets (50), and/or electropermanent magnets (51). While SMA actuators' maximum torque or force output may be too low to be used as the main actuator on heavier MRR modules, they are an attractive choice for activating latches because of their high force density and direct drive that requires no additional transmission. Earlier work on spring SMA-based connection mechanisms demonstrated compact female–male retractable pin connectors to lock and unlock several modules (52) arranged in lattice or chain reconfigurations that employ magnetic connectors (53). A similar approach with spring SMA actuators has been proposed for automatic coupling of modular origami robots (23) by retracting and releasing a sliding shaft with rotary teeth that, when coupled with stationery teeth, allows connected modules to rotate relative to one another. For self-assembly inside a fluid, the fluid flow can be used (much like magnets) to apply forces on the modules as well as form pressure bonds to hold structures together (54). More recently, a robot that melts (or remelts) plastic around the rim of the robot to join two modules has resulted in bonds that could support loads of more than 5 kg in 100 repeated trials (55).

3.2. Sensing and Communication

The sensing and communication in MRRs are often homogeneous over distributed modules. Because the systems are expandable, the reach of their sensory resources needs to be scalable. We can separate sensing into two categories: internal and external. The internal sensing modalities are for monitoring the states of each module locally. The external sensing modalities are involved with observing the environment outside the robot system.

3.2.1. Internal sensing. As MRR systems are characterized by large numbers of independent modules and their DOFs, the principal sensing requirements concern the control of active and passive DOFs. The majority of MRR systems use rotary joints, although some also have linear joints. The states of these joints are monitored by mixing and matching rotary encoders (19, 21, 22), potentiometers (11, 18, 56), Hall-effect sensors (20), or various strain sensors. One example that maximizes compactness—one of the important MRR design goals—is PaintPots (57). PaintPots utilize conductive paint to build potentiometers into the structure of a robot module at a relatively low cost. Some MRR systems are also equipped with sensors such as accelerometers that can measure module orientation, which can be critical for startup and initialization procedures (18, 56, 58).

Another critical sensing function for MRRs is to help guide docking processes, especially for chain and mobile reconfiguration. Chain systems utilize inverse kinematics with the internal sensors mentioned above, but often the errors are too large. Onboard cameras (59) or arrangements of LEDs and photosensors (9) can be integrated for better state estimation.

3.2.2. External sensing and communication. External sensing can be used to detect other modules as well as the environment. This is crucial for reconfiguration and global tasks, in which modules need to recognize each other and communicate directly or indirectly with their neighbors. More specifically, external sensing involves identifying who and where a neighboring module is (18), as well as detecting any interaction between them by using, for example, a touch switch (21) or spring-loaded contact (60, 61). Once a physical contact is made, the coupling is commonly used for direct communication between modules through electrical contacts (19, 20, 62) or induction (51). A bus system can simplify global communication but is limited to a group of modules in direct contact. Serial communication between neighboring modules can overcome the communication bus limits for scalability but increases the complexity of individual modules, and the system may still require some form of global communication. Wireless communication has been implemented both globally and locally using infrared (18, 21, 27, 56, 58), Bluetooth (18), and Wi-Fi (63), greatly improving the adaptability and autonomy of the overall system. Although it adds a considerable amount of overhead, wireless communication opens the possibility of sharing sensory information and enabling a more distributed approach to various functions. Broadcast messages are particularly useful because they are not limited to an address space. Initial studies have implemented distributed sensing of the environment using bump switches (22) as well as mapping using onboard sensors (64), and combining these with an external sensing system seems to be effective for achieving dedicated tasks through either mobile (65) or global systems (6). In heterogeneous systems, some of the sensing tasks can be offloaded to dedicated sensing modules (65, 66).

Historically, most homogeneous systems have had limited onboard computation and sensing. While the advantages of the homogeneous approach have not been rigorously evaluated, a hierarchical approach with a centralized sensor and environmental understanding seems to be more effective. Because of the complexity of creating an MRR system, a considerable amount of research has focused on functionality, system integration, and control, either physically or in a simulation. However, environmental sensing and awareness are crucial for a system to carry out tasks in real-world scenarios and fulfill the promise of versatility.

3.3. Distributed Control Architecture

Typically, the distribution of computational resources tends to complicate the programming and control of a system compared with a single centralized resource. However, including computational resources on each module means that the available memory and computational cycles scale with the number of modules. A variety of control architectures could implement the software functions expected from MRRs. Homogeneously distributed approaches are elegant and interesting from a computer science perspective, ranging from bioinspired control (67) to control of millions of modules in simulation (68) and nearly a thousand physically (69). The challenge tasks demonstrated include self-repair (70), locomotion (71), and manipulation (72), all of which are still at a relatively low level.

From a practical point of view, an approach that takes advantage of the inherent hierarchical physical arrangements of computational elements might work better for higher-level tasks that involve combining tasks of manipulation, locomotion, and environmental sensing. The hierarchy can include centralizing activities such as sensing and interpreting the environment, as well as reasoning about the appropriate configuration for the conglomerate while distributing local activities and controlling individual DOFs or motions of subgroups whose actions are tightly coupled (e.g., a robot arm). Section 4.1.1 describes an example of such a divide between centralized and distributed control, which was more efficient than either fully central or fully distributed control (65).

3.4. Structural Strength

The utility of the conglomerate shapes that MRRs can form is an important issue. The DARPA Programmable Matter program (73) [not to be confused with the programmable matter efforts of the Claytronics project (74)] focused on developing useful structures, often using a wrench as an example shape: Could an MRR system form a wrench that has enough strength to turn a bolt?

The strength (e.g., stiffness) of a conglomerate is a function of the module materials as well as the bonding stiffness between the modules. Typically, the bonding mechanism is significantly weaker than the module materials. White et al. (75) presented a strength analysis using the 6×6 stiffness matrices of systems composed of both soft and rigid components.

An alternative approach to creating mechanically stronger MRR systems is to change the architecture of the system from attaching elements serially to inherently parallel mechanisms, such as truss architectures. Parallel structures are naturally stronger than serial ones, but the strength of the full system is still dependent on the strength of the constituent truss members. In the VTT case (12), the spiral zipper mechanism used to form the tubes of each member has an optimal strength-to-weight ratio for beams under compression (76).

While structural weakness is typically the main issue with MRR conglomerates, sometimes structural flexibility is required. For example, in the DARPA Tactically Expandable Maritime Platform (TEMP) project (6), which uses an assemblage of large floating structures in the open ocean, the waves from the sea state combined with the inertia of the modules can induce very large forces and moments. Complying with the waves can reduce these forces and moments so that the structure can actually survive.

An important aspect of robot arm systems coupled with stiffness is precision. To obtain highly precise positioning, the structures of robot arms must be stiff. While there are some examples of modulating the individual joint stiffness to increase precision in grasping, the lack of stiffness in an MRR system leads to difficulties in docking with itself, imposing design constraints on the connection mechanism.

3.5. Connection Mechanisms

Although there are a wide variety of MRR systems, ranging from rotating cube-shaped modules in lattice systems to extending prismatic beams in truss systems, all self-reconfiguring systems share a few common aspects. One is the ability to physically attach and detach from other elements of the system, using an interface that we call the connection plate. Two modules can join and become rigidly attached when two connection plates physically lock (mate) together. Mating has several important characteristics:

- Gender: A connection plate can be male, female, hermaphroditic, or genderless. Male plates usually have a physical feature that protrudes, and female plates have the negative feature to that protrusion. Note that male/female primarily denotes a polarity; magnetic connectors (as in the SMORES system shown in **Figure 5**) can be considered gendered male or female even without protrusions, labeling north or south arbitrarily as male or female. Hermaphroditic connectors have both male and female features and tend to be the most common because all of the connection plates can be the same, which enables any two connection plates to mate. Genderless connectors have no discrete polarity (as hermaphroditic connectors do) or mating geometric features.
- Approach direction: The docking process between two connection plates assumes that there is a specific approach direction. This is typically a translation parallel to the normal of the plane that characterizes the mating face—for example, perpendicular to the faces of the ModQuad, crystalline, and SMORES modules. Note that while the vast majority of approach directions are simple translations, it is possible to design a connection plate with a complex approach direction, such as the motion of a screw (still one-DOF motion, but coupled translation and rotation).
- Latching: Every docking connector needs to form a rigid connection between the modules, which includes preventing motions that back out along the approach direction. This can be done with a latch, which can be passive or active. The difference is whether actuators

are added to the connection plate specifically to latch or unlatch. Passive latches utilize the main actuators to cause some form of unidirectional holding action (e.g., magnets or bistable snap-through latches). Undocking for these latches must occur either through the main actuation or through some other external mechanism (e.g., a human hand or a robot gripper).

- Compactness: As with modules in general, the compactness of the docking mechanism is an important design trade-off. High reliability or the ability to tolerate large errors often comes at the cost of space. For example, the docking mechanism of the ATRON system (77) consumes more than half the space inside each module (78), although it is considered arguably one of the most reliable docking mechanisms.

Given the difficulties with precision positioning (Section 3.4), many systems aim to maximize the ability to dock in the presence of uncertainty. Thus, a metric to measure how much uncertainty a docking system can tolerate yet still dock can be used as a figure of merit.

The area of acceptance (AA) is defined as "the range of possible starting conditions for which mating will be successful" (79, p. 1227). In other words, AA is the full set of possible errors (for translation, rotation, and the two in combination) that two docking connectors can have positionally relative to each other yet still successfully dock together.

3.5.0.1. Zero-rotation area of acceptance. When all rotational DOFs are constrained, we call the set of positions that align successfully the zero-rotation area of acceptance (ZRAA). When two modules approach in the z direction translationally, ZRAA can be represented as a finite region in the x–y plane. This gives us a relatively simple, quick picture of the acceptance potential of the given connector. A similar idea was described by Nilsson (80). **Table 1** summarizes some analytically determined ZRAA values.

Table 1 Zero-rotation area of acceptance (ZRAA) metrics, normalized to the characteristic length of the face

System	Normalized ZRAA sum
GENFA connector (81)	0.00353
PolyBot (71)	0.00503
M-TRAN III (18)	0.00592
JHU (82)	0.00592
I-Cubes (83)	0.0187[a]
CONRO (84)	0.0425[a]
Vacuube (85)	0.0555
X-CLAW (86)	0.0649
ACOR (unpaired) (87)	0.0711
SINGO connector (88)	0.306
DRAGON (89)	0.353
AMOUR (90)	1.57
3-D X-Face (79)[b]	2.00

Abbreviations: ACOR, active connector for robotic systems; AMOUR, Autonomous Modular Optical Underwater Robot; CONRO, Configurable Robot; GENFA, Genderless and Fail-Safe; JHU, Johns Hopkins University; M-TRAN, Modular Transformer; SINGO, single end operative.

[a]Estimated values.

[b]The 3-D X-Face is not a full connector.

3.5.0.2. Full area of acceptance. The full AA is obtained by taking all of the possible DOFs into account. For two-dimensional connectors, the full AA is represented with two parameters, for one translational and one rotational DOF. For three-dimensional connectors, five parameters are needed to represent the two translational and three rotational DOFs. In the most general case, it is difficult to develop an analytical model to estimate AA. However, representation of contact between two objects in arbitrary dimensions has been studied extensively in motion planning in configuration space obstacles. We can conveniently use these representations with sampling-based methods to estimate AA (91).

4. SOFTWARE AND CONTROL: ADVANCES AND CHALLENGES

Ultimately, the vision of MRR systems includes fully autonomous self-reconfiguration. Both the conventional AI/robotic issues, ranging from semantic understanding to motion control, and the MRR-specific concerns need to be addressed. From a software control point of view, we can break the MRR-specific issues down into two parts: task-shape matching and reconfiguration planning and control.

The highest-level activity in MRR scenarios is to understand the task or situation and then match a configuration that can perform that task or behave appropriately in that situation (Section 4.1). The system must then determine how to metamorphose its current configuration into the desired one (Section 4.2). This problem can be divided into two parts: one that focuses on the connectivity arrangement and one that determines how to move the modules in a collision-free manner despite the many DOFs in a typical MRR system.

Many of the specific complications in developing software for reconfigurable robots are derived from the architecture of the hardware. For example, some latching systems require docking approaches from specific directions, and most systems have a limited number of modules that can be suspended in a cantilever fashion under gravity. These types of constraints along with module geometry constraints can drastically change the implementation of software to control reconfiguration or to perform tasks. However, MRR systems still have a set of common classes of software tasks.

4.1. Task–Shape Matching

The task–shape matching problem with MRRs is concerned with determining which of the possible connectivity arrangements of an MRR system may be the most suitable for a given task, which in the most general case can be any robotic locomotion or manipulation activity. Developing a formalism to rigorously describe the tasks that we want robots to perform is an active research area. One approach to user-friendly, high-level task description is to use linear temporal logic formulas (92). At a relatively low level, particularly in the robotic manipulation and locomotion literature (93), robotic tasks are given as a control objective for motion, force, hybrid motion–force, or impedance control.

In the MRR context, it is critical to understand the space of all the possible connectivities of a given robot system. This is nontrivial because the search space and the number of constraints on module connectivity grow exponentially with the number of modules (94). In many cases, such constraints may also be complex and global. For example, a docking event between two modules may be affected by the presence of another module that is not directly involved with that process (95). Constructing the database of robot connectivities becomes more complex if we additionally take into account what each connectivity is suitable for—i.e., its functions and behaviors. Here, problems worth further investigation include understanding what connectivities may function identically (due to structural symmetry, for example). Such high-level information can help us figure out the structure of the search space better and query the database more efficiently (96).

Given a database of feasible robot connectivities and the descriptions of a desired task, a variety of approaches to task–shape matching can be considered. Chen & Burdick (97) formulated the problem as a discrete optimization procedure with a task-oriented objective function. Morrow & Khosla (98) proposed a sensorimotor primitive layer as a tool for bridging the robot hardware system and the tasks. Paredis et al. (99) presented the concept of a rapidly deployable system, which features a paradigm of software assembly that has the potential to facilitate the generation of control software as part of a solution to task–shape matching. Farritor & Dubowsky (100) proposed a hierarchical selection process as a way to systematically search for the space of robot connectivities. Although these approaches may be effective when the search space is relatively small, the highly complex and nonlinear nature of the general case remains a great challenge.

4.1.1. Example: an end-to-end system for accomplishing tasks with modular robots. Jing et al. (101) presented an end-to-end approach to task–shape matching. This paper features a system for accomplishing complex tasks with MRRs. The system comprises a high-level mission planner, a large robot connectivity–behavior library, a design and simulation tool for populating the library with new connectivities and behaviors, and MRR hardware (SMORES; see **Figure 5**). At the highest level, a target task is specified in an abstract manner, in terms of the task environment and desired behavioral properties—for example, "if the robot is moving in a tunnel, maintain a maximum height of 3 units." The high-level mission planner of the system then selects robot connectivities and behaviors from the library that fulfill all the requested functionalities. Finally, the specified mission is compiled and sent to the robot hardware to complete the task. **Figure 7** shows a wide range of robot connectivities synthesized by the system, which were applied to the demonstrated challenge task, cleaning the top of a table. A high-fidelity physics simulator and a more exhaustive connectivity–behavior library (as presented in 102) can further improve the tool chain.

4.2. Reconfiguration Planning and Control

A key ability of MRR systems that differentiates them from normal robot systems is the determination and execution of a course of action that includes connectivity changes to switch from a starting configuration to another one. This problem is referred to as reconfiguration planning and control.

Although it may be possible to regard the problem as a variant of the general robot motion planning and control problem, which has been studied in robotics for many decades (103–106), the problem is distinct from the traditional approaches in that the connectivity of the modules

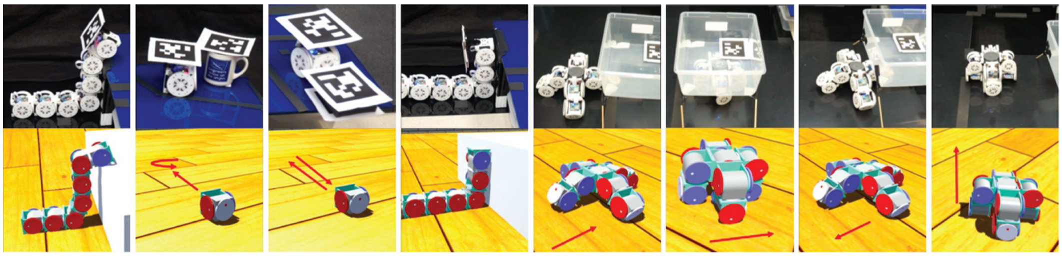

Figure 7

A set of robot configurations assembled with Self-Assembling Modular Robot for Extreme Shape-Shifting (SMORES) modules (11) to perform a user-specified task—cleaning the top of a table. For example, the leftmost panel shows a snake-shaped connectivity to climb up the table. Figure reproduced from Reference 101 with permission.

(or the topology of the robotic structure) also changes as needed. Challenges here include the possibly massive number of modules and their DOFs, which necessitates analyzing data in high-dimensional spaces (often referred to as the curse of dimensionality). Standard motion planning techniques that work well with a handful of dimensions and have guarantees of finding a solution can fail with an excessive number of DOFs, whose configuration space can be very complex topologically.

The main advantage of the lattice reconfiguration system over other types of architectures is that planning and control can be simplified through discretization. Both the connectivity of the system and the control actions that each module can take can be represented as a discrete set. Reconfiguration planning is then reduced to searching for a sequence of discrete motions in which the modules move to the adjacent lattice cells on the surface of the robotic structure. Collision detection, which is typically the most time-consuming part of motion planning, is also facilitated by the simplification that modules nominally sit on lattice positions, and changes in module position occur along discretized paths, by successively moving into adjacent lattice cells. The local nature of collision detection also paves the way to formulate motion planning in a distributed fashion. For those who are interested more in the computational aspects of reconfigurable robotics, the lattice reconfiguration architectures thus tend to be the most popular platform.

The reconfiguration of the chain architecture systems features changes in the topology of the linkage, for example, between open and closed kinematic chains. There are a wide range of fundamental issues regarding this process, such as determining what types of linkages have connected connectivity spaces such that reconfiguration can be possible between any two feasible connectivities (for more history and details on this research area, see 107). If it turns out that two configurations can reach each other, the next task is to find a collision-free path connecting them. Such a query can be addressed by modern robot motion planning software [for example, MoveIt! (**http://moveit.ros.org**)], but it can be computationally expensive for highly redundant kinematic chains with a complex topology featuring multiple holes, which can be easily imagined in realistic scenarios. The high computational cost stems from the fact that detecting collisions in linkages is a global issue in which all the DOFs of the system need to be taken into account. One approach to lowering the cost is to freeze the mobility of a subset of modules in order to reduce the dimensionality of the search space.

For the computed reconfiguration plans to be physically executable, the static and dynamic constraints of the robot and environment need to be taken into account, an approach referred to as kinodynamic motion planning. For example, in the chain architecture systems, the payload of a module is always upper bounded, so there is a limit on the number of modules that it can cantilever. What makes kinodynamic planning for the chain-type systems more challenging is that the inertial properties of the system change each time a reconfiguration occurs.

Generally, reconfiguration can be performed more efficiently via a common intermediate configuration, sometimes called a canonical configuration (108). One interesting research question is to find an optimal canonical configuration in terms of, for example, minimizing the sum of the control efforts of all the modules. In the chain architecture systems, an obvious choice for the canonical configuration is a simple open kinematic chain (108). A good research direction would be to further investigate what advantages canonical configurations can provide and how they can be implemented with real hardware systems.

In the examples below, we look at the practices of reconfiguration in each architecture.

4.2.1. Example: reconfiguration of lattice-type systems. The problem studied by Werfel & Nagpal (109) is concerned with algorithmic issues regarding decentralized construction with a

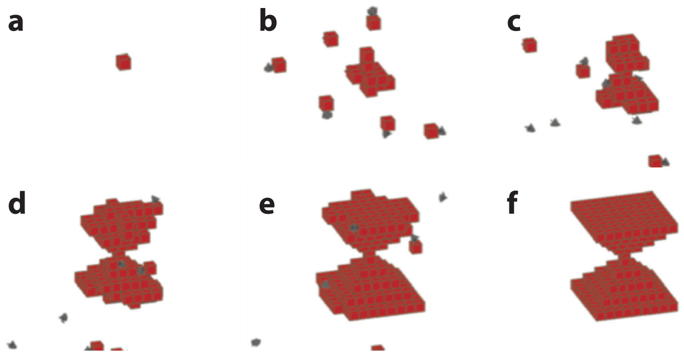

Figure 8

Decentralized construction of an hourglass-shaped structure. Figure adapted from Reference 109 with permission.

bipartite modular system composed of passive blocks to embody the structure and active robots to carry the blocks on the surface of the structure (**Figure 8**). Planning for connectivity changes is performed at the block level (the blocks are assumed to have some computing power). The blocks share a common coordinate frame and the description of a desired structure. They then determine at which of their faces additional blocks should be allowed to attach according to the local-scale rules for growing the structure in a greedy manner and preventing any dead-end sites that cannot be physically reached by the blocks. Motion planning and control strategies for the mobile robots include random movement in which the robots move arbitrarily on the surface of the growing structure and a more systematic approach—featuring, for example, gradient descent following a numerical gradient signaled by the blocks. In these approaches, there is a trade-off between the cost of communication and robot control effort.

Figure 9 shows a lattice reconfiguration system presented by Romanishin et al. (110) named 3-D M-Blocks. The system is capable of dynamic reconfiguration. Each module is able to exert both forward and backward torques, generated by the momentum of the flywheel inside, about three orthogonal axes, which enables it to pivot around an adjacent module bonded magnetically.

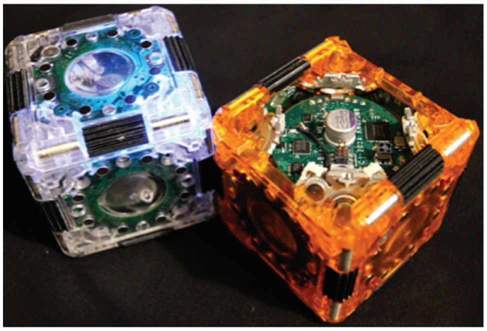

Figure 9

Two M-Block modules, which are able to move dynamically relative to each other. Photo by Daniela Rus's laboratory, reproduced from Reference 110 with permission.

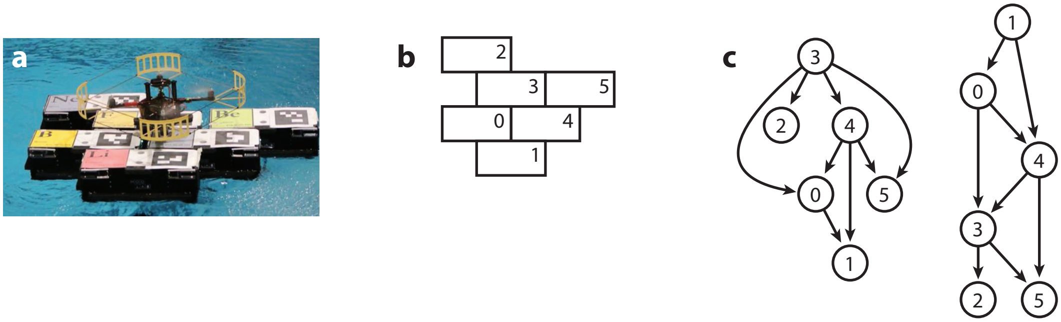

Figure 10

(*a*) A floating quadrotor landing pad assembled with six boats. (*b*) The target structure schematic. (*c*) Two different assembly plans for the target structure, represented as directed acyclic graphs. For example, the plan on the left begins by occupying site 3. Figure adapted from Reference 6 with permission.

4.2.2. Example: self-assembly of modular robotic boats. Paulos et al. (6) presented the self-assembly of large teams of autonomous modular robotic boats (**Figure 10**). The cuboid-shaped boats are designed to dock only along their long sides and form a regular pattern that looks like the common brick wall. Each robot is capable of holonomic locomotion on water. The robots can thus be considered a mobile reconfiguration system. The connectivity planning algorithms presented by Seo et al. (111, 112) parse a blueprint for a target structure and return an assembly sequence that specifies an order to assemble the structure in $O(m)$ time, where m is the size of the target structure (that is, the number of the dock sites of the target structure). The resulting plan facilitates motion planning for the modules in the sense that an open dock site is not flanked by two other modules already assembled in the structure, which will form a narrow gap that can essentially block off the open site. In addition, if a target structure covers a simply connected area with no hole (as in **Figure 10**), it is possible to perform module assembly in a parallel, distributed manner.

Based on the outcome of the connectivity portion of reconfiguration planning, the motion planning scenario described by Paulos et al. (6) is concerned with a large pool of available modular robotic boats and open docking sites around the perimeter of the growing structure. This task necessitates a highly scalable motion planner that can handle the plurality of the mobile robots and the candidate docking positions, which may change frequently as the robots dock to the structure in a decentralized manner. The trajectory planning algorithm described by Paulos et al. (6) builds on Dijkstra's algorithm (113) and runs in polynomial time—$O(n^3)$ time, where n is the number of modules.

4.2.3. Example: planning and control for chain reconfiguration systems. Casal (114) presented a planner for purely kinematic chain reconfiguration. This work features a connectivity planner utilizing the canonical configuration approach (108) and a motion planner using a sampling-based motion planning approach.

There are other approaches that represent chain systems as trees (acyclic graphs) for connectivity planning without going through a canonical configuration (115, 116). These approaches can be distributed and make some assumptions about the nonisomorphism of serial chains.

Şucan et al. (117) presented a kinodynamic motion planning approach applicable to MRR systems that combines a sampling-based motion planner and a physics-based simulator. The method is demonstrated with the topology of an open kinematic chain. The problem of more general kinodynamic reconfiguration involved with the change of module connectivity remains open.

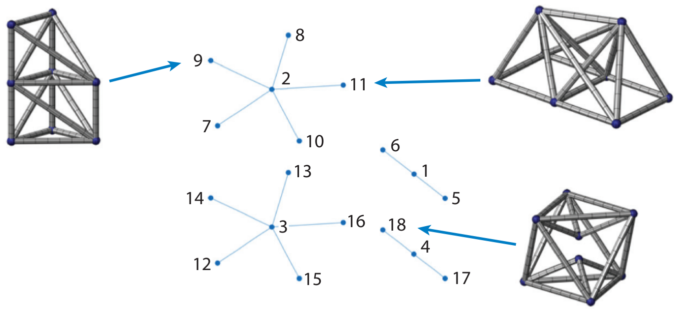

Figure 11

The topology neighbor graph for reconfigurable 18-member trusses. For example, to reconfigure from topology 9 to topology 11, the variable topology truss must first reconfigure into topology 2. Topology 18 cannot reconfigure into topology 11 because there is no path in the graph. Figure adapted from Reference 12 with permission.

4.2.4. Example: reconfiguration of variable topology truss systems. A unique aspect of the truss systems is the constraints on the types of topologies that can be obtained. The reconfigurability of a VTT can be described using the topology neighbor graph of the VTT (12). Each node in the topology neighbor graph represents a unique truss connectivity. Two nodes are connected if a single topological reconfiguration step takes a configuration in the first connectivity to a configuration in the second connectivity. Admissible connectivities and topological reconfiguration steps are determined by the constraints of the physical system. For example, the VTT system requires statically determinate structures (more precisely, infinitesimal rigidity) (12). Merging and splitting of nodes are allowed because each joint between two members is partial; however, the merging of edges in a graph of the connectivity is not allowed because they are made up of truss members that cannot share the same space. It is interesting that, given the physical constraints for the VTT system, at least 18 members are required before topological reconfiguration can occur. **Figure 11** shows the neighbor graph that indicates the 19 nonisomorphic connectivities and the reachability of each. As more members are added to a system, the number of nonisomorphic connectivities grows exponentially (a 24-member system has more than 10,000 connectivities), and listing these connectivities in a brute force manner becomes intractable.

5. FUTURE DIRECTIONS

There are many examples and demonstrations of MRRs with many (dozens of) modules performing simple locomotion and manipulation tasks as well as self-reconfiguration. However, MRR systems have not yet fulfilled their promise of versatility, robustness, and low cost. When we say versatile, the implication is that the variety of tasks performed by the robot will actually be useful tasks. To be useful, the systems may need to fulfill all three promises simultaneously.

5.1. Scaling Numbers

While early work on the computational aspects of MRRs focused on large numbers of modules in simulation, in practice there have rarely been physical systems with more than several dozen modules. Rubenstein et al. (69) described a system with 1,000 modules and demonstrated practicalities such as how to program all the modules and how to recharge their batteries. However, this system had loosely coupled mobile elements moving in a plane as opposed to rigidly coupled structures, which tend to have more utility. At this stage, the important future question may not be what the largest number of modules is, but rather what the right number of modules is that can be useful for a set of tasks.

In terms of the robustness element of the three areas of promise, a larger number of modules increases the probability of failure of individual components. If the modules are tightly coupled and dependent on each other, the likelihood of failure drastically increases, so there is a stronger incentive to optimize this number.

5.2. Scaling Size

The early work of Fukuda et al. (5) established the dream of miniature robots that are injected into a human body to self-assemble and repair a human heart. While miniaturized robots moving through a body and capable of self-assembly are being developed (118), the practical usefulness of miniature MRR systems is still in question. Self-assembly has been demonstrated, but self-reconfiguration of repeated modular elements in the MRR style is less clear.

On the other end, scaling up the size to fit a given task is an interesting problem. This again relates to finding the optimal number of modules. The truss architectures seem promising in this respect, as the size is adjustable without increasing the number of modules.

5.3. Scalable Computation

Task-level computation is the highest level of computational problem. Ultimately, exploiting the self-reconfiguration capability in an autonomous system will require an understanding of tasks and the environment in which the tasks are to be done. While work has begun on task–shape matching, developing a better formalism for characterizing tasks at fine granularity (more general) will enable researchers to develop algorithms to match capabilities to subtasks and mechanisms for integrating subtasks together to achieve complex behaviors.

Reconfiguration planning solutions for a variety of systems have been shown to be somewhat effective. Several even have the ideal characteristics of being complete as well as distributed. However, we have yet to see demonstrations of these algorithms on a physical system doing something useful. This may require the use of hybrid MRR systems. The truss reconfiguration robot systems promise to have a hardware structure that solves many of the usefulness problems of earlier systems (structural strength, number of modules, and scalability of size) but present many significant hardware and software design complexities.

Future applications and requirements, such as space travel, may also drive the development of MRR systems. These systems need to be extremely compact and multifunctional. For extended-stay extraterrestrial habitation, self-repair is also important, so potential future MRR systems could be well suited for this application.

DISCLOSURE STATEMENT

The authors are not aware of any affiliations, memberships, funding, or financial holdings that might be perceived as affecting the objectivity of this review.

LITERATURE CITED

1. Eppinger SD, Ulrich KT. 1995. *Product Design and Development*. New York: McGraw-Hill
2. Yim M, Duff DG, Roufas KD. 2000. PolyBot: a modular reconfigurable robot. In *ICRA 2000: IEEE International Conference on Robotics and Automation*, Vol. 1, pp. 514–20. New York: IEEE

3. Stoy K, Brandt D, Christensen DJ, Brandt D. 2010. *Self-Reconfigurable Robots: An Introduction*. Cambridge, MA: MIT Press
4. Yim M, Shen WM, Salemi B, Rus D, Moll M, et al. 2007. Modular self-reconfigurable robot systems. *IEEE Robot. Autom. Mag.* 14(1):43–52
5. Fukuda T, Nakagawa S, Kawauchi Y, Buss M. 1989. Structure decision method for self organising robots based on cell structures-CEBOT. In *1989 IEEE International Conference on Robotics and Automation*, Vol. 2, pp. 695–700. New York: IEEE
6. Paulos J, Eckenstein N, Tosun T, Seo J, Davey J, et al. 2015. Automated self-assembly of large maritime structures by a team of robotic boats. *IEEE Trans. Autom. Sci. Eng.* 12:958–68
7. Saldaña D, Gabrich B, Li G, Yim M, Kumar V. 2018. ModQuad: the flying modular structure that self-assembles in midair. In *2018 IEEE International Conference on Robotics and Automation*, pp. 691–98. New York: IEEE
8. Rus D, Vona M. 2001. Crystalline robots: self-reconfiguration with compressible unit modules. *Auton. Robots* 10:107–24
9. Yim M, Zhang Y, Roufas K, Duff D, Eldershaw C. 2002. Connecting and disconnecting for chain self-reconfiguration with PolyBot. *IEEE/ASME Trans. Mechatron.* 7:442–51
10. Murata S, Yoshida E, Kamimura A, Kurokawa H, Tomita K, Kokaji S. 2002. M-TRAN: self-reconfigurable modular robotic system. *IEEE/ASME Trans. Mechatron.* 7:431–41
11. Davey J, Kwok N, Yim M. 2012. Emulating self-reconfigurable robots - design of the SMORES system. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4464–69. New York: IEEE
12. Spinos A, Carroll D, Kientz T, Yim M. 2017. Variable topology truss: design and analysis. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2717–22. New York: IEEE
13. Yim M, White P, Park M, Sastra J. 2009. Modular self-reconfigurable robots. In *Encyclopedia of Complexity and Systems Science*, ed. RA Meyers. New York: Springer. **https://doi.org/10.1007/978-0-387-30440-3_334**
14. Miura K, Furuya H, Suzuki K. 1985. Variable geometry truss and its application to deployable truss and space crane arm. *Acta Astronaut.* 12:599–607
15. Hamlin GJ, Sanderson AC. 1997. TETROBOT: a modular approach to parallel robotics. *IEEE Robot. Autom. Mag.* 4(1):42–50
16. Chirikjian GS. 1994. Hyper-redundant manipulator dynamics: a continuum approximation. *Adv. Robot.* 9:217–43
17. Kirby BT, Aksak B, Campbell JD, Hoburg JF, Mowry TC, et al. 2007. A modular robotic system using magnetic force effectors. In *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2787–93. New York: IEEE
18. Kurokawa H, Tomita K, Kamimura A, Kokaji S, Hasuo T, Murata S. 2008. Distributed self-reconfiguration of M-TRAN III modular robotic system. *Int. J. Robot. Res.* 27:373–86
19. Spröwitz A, Moeckel R, Vespignani M, Bonardi S, Ijspeert AJ. 2014. Roombots: a hardware perspective on 3D self-reconfiguration and locomotion with a homogeneous modular robot. *Robot. Auton. Syst.* 62:1016–33
20. Liedke J, Matthias R, Winkler L, Wörn H. 2013. The collective self-reconfigurable modular organism (CoSMO). In *2013 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*. New York: IEEE. **https://doi.org/10.1109/AIM.2013.6584059**
21. Wei H, Chen Y, Tan J, Wang T. 2011. Sambot: a self-assembly modular robot system. *IEEE/ASME Trans. Mechatron.* 16:745–57
22. Baca J, Hossain S, Dasgupta P, Nelson CA, Dutta A. 2014. ModRED: hardware design and reconfiguration planning for a high dexterity modular self-reconfigurable robot for extra-terrestrial exploration. *Robot. Auton. Syst.* 62:1002–15
23. Belke CH, Paik J. 2017. Mori: a modular origami robot. *IEEE/ASME Trans. Mechatron.* 22:2153–64
24. Zhao M, Anzai T, Shi F, Chen X, Okada K, Inaba M. 2018. Design, modeling, and control of an aerial robot dragon: a dual-rotor-embedded multilink robot with the ability of multi-degree-of-freedom aerial transformation. *IEEE Robot. Automat. Lett.* 3:1176–83
25. Robertson MA, Paik J. 2017. New soft robots really suck: vacuum-powered systems empower diverse capabilities. *Sci. Robot.* 2:eaan6357

26. Agarwal G, Robertson MA, Sonar H, Paik J. 2017. Design and computational modeling of a modular, compliant robotic assembly for human lumbar unit and spinal cord assistance. *Sci. Rep.* 7:14391
27. Romanishin JW, Gilpin K, Claici S, Rus D. 2015. 3D M-Blocks: self-reconfiguring robots capable of locomotion via pivoting in three dimensions. In *2015 IEEE International Conference on Robotics and Automation*, pp. 1925–32. New York: IEEE
28. Gilpin K, Knaian A, Rus D. 2010. Robot pebbles: one centimeter modules for programmable matter through self-disassembly. In *2010 IEEE International Conference on Robotics and Automation*, pp. 2485–92. New York: IEEE
29. White PJ, Yim M. 2010. Reliable external actuation for full reachability in robotic modular self-reconfiguration. *Int. J. Robot. Res.* 29:598–612
30. Diller E, Pawashe C, Floyd S, Sitti M. 2011. Assembly and disassembly of magnetic mobile micro-robots towards deterministic 2-D reconfigurable micro-systems. *Int. J. Robot. Res.* 30:1667–80
31. Tolley MT, Felton SM, Miyashita S, Aukes D, Rus D, Wood RJ. 2014. Self-folding origami: shape memory composites activated by uniform heating. *Smart Mater. Struct.* 23:094006
32. White PJ, Latscha S, Schlaefer S, Yim M. 2011. Dielectric elastomer bender actuator applied to modular robotics. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 408–13. New York: IEEE
33. Zhang X, Zhang G, Nakamura K, Ueha S. 2011. A robot finger joint driven by hybrid multi-DOF piezoelectric ultrasonic motor. *Sens. Actuators A* 169:206–10
34. Yan S, Zhang F, Qin Z, Wen S. 2006. A 3-DOFs mobile robot driven by a piezoelectric actuator. *Smart Mater. Struct.* 15:N7–13
35. Hines L, Arabagi V, Sitti M. 2012. Shape memory polymer-based flexure stiffness control in a miniature flapping-wing robot. *IEEE Trans. Robot.* 28:987–90
36. Bae WG, Choi JH, Lee SH, Kang D, Jung KW, Suh KY. 2010. Centering mechanism for micro vessel robot using micropatterned shape memory polymers. In *2010 3rd IEEE RAS and EMBS International Conference on Biomedical Robotics and Biomechatronics*, pp. 594–98. New York: IEEE
37. Zhakypov Z, Huang JL, Paik J. 2016. A novel torsional shape memory alloy actuator: modeling, characterization, and control. *IEEE Robot. Autom. Mag.* 23(3):65–74
38. Zhakypov Z, Paik J. 2018. Design methodology for constructing multimaterial origami robots and machines. *IEEE Trans. Robot.* 34:151–65
39. Firouzeh A, Paik J. 2017. Soft actuation and sensing towards robot-assisted facial rehabilitation. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 306–13. New York: IEEE
40. Sheng J, Gandhi D, Gullapalli RP, Simard JM, Desai JP. 2017. Development of a meso-scale SMA-based torsion actuator for image-guided procedures. *IEEE Trans. Robot.* 33:240–48
41. Paik JK, Wood RJ. 2012. A bidirectional shape memory alloy folding actuator. *Smart Mater. Struct.* 21:065013
42. Pulnev S, Nikolaev V, Priadko A, Rogov A, Chikiryaka A, Nikanorov S. 2009. Shape memory effect and linear actuators based on Cu-Al-Ni single crystals. *Bull. Russ. Acad. Sci. Phys.* 73:1398
43. Torres-Jara E, Gilpin K, Karges J, Wood RJ, Rus D. 2010. Compliant modular shape memory alloy actuators. *IEEE Robot. Autom. Mag.* 17(4):78–87
44. Kohl M, Just E, Pfleging W, Miyazaki S. 2000. SMA microgripper with integrated antagonism. *Sens. Actuators A* 83:208–13
45. Mineta T, Mitsui T, Watanabe Y, Kobayashi S, Haga Y, Esashi M. 2001. Batch fabricated flat meandering shape memory alloy actuator for active catheter. *Sens. Actuators A* 88:112–20
46. Whitney JP, Sreetharan PS, Ma KY, Wood RJ. 2011. Pop-up book MEMS. *J. Micromech. Microeng.* 21:115021
47. Paik J, Byoungkwon A, Rus D, Wood RJ. 2011. *Robotic origamis: self-morphing modular robot*. Paper presented at the 2nd International Conference on Morphological Computation, Venice, Italy, Sept. 12–14
48. Kamimura A, Kurokawa H, Yoshida E, Tomita K, Kokaji S, Murata S. 2004. Distributed adaptive locomotion by a modular robotic system, M-TRAN II. In *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Vol. 3, pp. 2370–77. New York: IEEE

49. Liedke J, Wörn H. 2011. CoBoLD—a bonding mechanism for modular self-reconfigurable mobile robots. In *2011 IEEE International Conference on Robotics and Biomimetics*, pp. 2025–30. New York: IEEE
50. Kotay K, Rus D, Vona M, McGray C. 1998. The self-reconfiguring robotic molecule. In *1998 IEEE International Conference on Robotics and Automation*, Vol. 1, pp. 424–31. New York: IEEE
51. Tosun T, Davey J, Liu C, Yim M. 2016. Design and characterization of the EP-Face connector. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 45–51. New York: IEEE
52. Yoshida E, Murata S, Kokaji S, Kamimura A, Tomita K, Kurokawa H. 2002. Get back in shape! *IEEE Robot. Autom. Mag.* 9(4):54–60
53. Murata S, Yoshida E, Tomita K, Kurokawa H, Kamimura A, Kokaji S. 2000. Hardware design of modular robotic system. In *2000 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Vol. 3, pp. 2210–17. New York: IEEE
54. White P, Zykov V, Bongard JC, Lipson H. 2005. Three dimensional stochastic reconfiguration of modular robots. In *Robotics: Science and Systems I*, ed. S Thrun, GS Sukhatme, S Schaal, pp. 161–68. Cambridge, MA: MIT Press
55. Swissler P, Rubenstein M. 2018. FireAnt: a modular robot with full-body continuous docks. In *2018 IEEE International Conference on Robotics and Automation*, pp. 6812–17. New York: IEEE
56. Brandt D, Christensen DJ, Lund HH. 2007. ATRON robots: versatility from self-reconfigurable modules. In *2007 International Conference on Mechatronics and Automation*, pp. 26–32. New York: IEEE
57. Tosun T, Edgar D, Liu C, Tsabedze T, Yim M. 2017. PaintPots: low cost, accurate, highly customizable potentiometers for position sensing. In *2017 IEEE International Conference on Robotics and Automation*, pp. 1212–18. New York: IEEE
58. Salemi B, Moll M, Shen WM. 2006. SUPERBOT: a deployable, multi-functional, and modular self-reconfigurable robotic system. In *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3636–41. New York: IEEE
59. Groß R, Bonani M, Mondada F, Dorigo M. 2006. Autonomous self-assembly in a swarm-bot. In *Proceedings of the 3rd International Symposium on Autonomous Minirobots for Research and Edutainment (AMiRE 2005)*, ed. K Murase, K Sekiyama, T Naniwa, N Kubota, J Sitte, pp. 314–22. Berlin: Springer
60. Kernbach S, Schlachter F, Humza R, Liedke J, Popesku S, et al. 2011. Heterogeneity for increasing performance and reliability of self-reconfigurable multi-robot organisms. arXiv:1109.2288 [cs.RO]
61. Lyder A, Garcia RFM, Stoy K. 2010. Genderless connection mechanism for modular robots introducing torque transmission between modules. In *Proceedings of the ICRA Workshop on Modular Robots: State of the Art*, pp. 77–81. New York: IEEE
62. Parrott C, Dodd TJ, Groß R. 2014. HiGen: a high-speed genderless mechanical connection mechanism with single-sided disconnect for self-reconfigurable modular robots. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3926–32. New York: IEEE
63. Pacheco M, Fogh R, Lund HH, Christensen DJ. 2015. Fable II: design of a modular robot for creative learning. In *2015 IEEE International Conference on Robotics and Automation*, pp. 6134–39. New York: IEEE
64. Magnenat S, Philippsen R, Mondada F. 2012. Autonomous construction using scarce resources in unknown environments. *Auton. Robots* 33:467–85
65. Tosun T, Daudelin J, Jing G, Kress-Gazit H, Campbell M, Yim M. 2017. Perception-informed autonomous environment augmentation with modular robots. arXiv:1710.01840 [cs.RO]
66. Zhu Y, Zhao J, Cui X, Wang X, Tang S, et al. 2013. Design and implementation of UBot: a modular self-reconfigurable robot. In *2013 IEEE International Conference on Mechatronics and Automation*, pp. 1217–22. New York: IEEE
67. Shen WM, Salemi B, Will P. 2002. Hormone-inspired adaptive communication and distributed control for CONRO self-reconfigurable robots. *IEEE Trans. Robot. Autom.* 18:700–12
68. Fitch R, Butler Z. 2008. Million module march: scalable locomotion for large self-reconfiguring robots. *Int. J. Robot. Res.* 27:331–43
69. Rubenstein M, Cornejo A, Nagpal R. 2014. Programmable self-assembly in a thousand-robot swarm. *Science* 345:795–99
70. Tomita K, Murata S, Kurokawa H, Yoshida E, Kokaji S. 1999. Self-assembly and self-repair method for a distributed mechanical system. *IEEE Trans. Robot. Autom.* 15:1035–45

71. Yim M, Duff D, Roufas K. 2000. PolyBot: a modular reconfigurable robot. In *ICRA 2000: IEEE International Conference on Robotics and Automation*, Vol. 1, pp. 514–20. New York: IEEE
72. Yim M, Reich J, Berlin AA. 2000. Two approaches to distributed manipulation. In *Distributed Manipulation*, ed. KF Böhringer, H Choset, pp. 237–61. Boston: Springer
73. Zakin M. 2007. *The next revolution in materials*. Presentation at DARPA's 25th Systems and Technology Symposium (DARPATech), Anaheim, CA, Aug. 7. **http://archive.darpa.mil/DARPATech2007/proceedings/dt07-dso-zakin-materials.pdf**
74. Goldstein SC, Campbell JD, Mowry TC. 2005. Programmable matter. *Computer* 38:99–101
75. White PJ, Posner ML, Yim M. 2010. Strength analysis of miniature folded right angle tetrahedron chain programmable matter. In *2010 IEEE International Conference on Robotics and Automation*, pp. 2785–90. New York: IEEE
76. Collins F, Yim M. 2016. Design of a spherical robot arm with the spiral zipper prismatic joint. In *2016 IEEE International Conference on Robotics and Automation*, pp. 2137–43. New York: IEEE
77. Stoy K, Christensen DJ, Brandt D, Bordignon M, Schultz UP. 2009. Exploit morphology to simplify docking of self-reconfigurable robots. In *Distributed Autonomous Robotic Systems 8*, ed. H Asama, H Kurokawa, J Ota, K Sekiyama, pp. 441–52. Berlin: Springer
78. Østergaard EH, Kassow K, Beck R, Lund HH. 2006. Design of the ATRON lattice-based self-reconfigurable robot. *Auton. Robots* 21:165–83
79. Eckenstein N, Yim M. 2014. Area of acceptance for 3D self-aligning robotic connectors: concepts, metrics, and designs. In *2014 IEEE International Conference on Robotics and Automation*, pp. 1227–33. New York: IEEE
80. Nilsson M. 1999. *Symmetric docking in 2D: a bound on self-alignable offsets*. Paper presented at the International Association of Science and Technology for Development (IASTED) Conference on Robotics and Applications, Santa Barbara, CA, Oct. 28–30
81. Fu G, Menciassi A, Dario P. 2011. Development of a genderless and fail-safe connection system for autonomous modular robots. In *2011 IEEE International Conference on Robotics and Biomimetics*, pp. 877–82. New York: IEEE
82. Kutzer M, Moses M, Brown C, Scheidt D, Chirikjian G, Armand M. 2010. Design of a new independently-mobile reconfigurable modular robot. In *2010 IEEE International Conference on Robotics and Automation*, pp. 2758–64. New York: IEEE
83. Unsal C, Khosla PK. 2000. Solutions for 3D self-reconfiguration in a modular robotic system: implementation and motion planning. In *Sensor Fusion and Decentralized Control in Robotic Systems III*, ed. GT McKee, PS Schenker, pp. 388–401. SPIE Proc. Vol. 4196. Bellingham, WA: Soc. Photo-Opt. Instrum. Eng.
84. Castano A, Behar A, Will P. 2002. The CONRO modules for reconfigurable robots. *IEEE/ASME Trans. Mechatron.* 7:403–9
85. Garcia R, Hiller J, Stoy K, Lipson H. 2011. A vacuum-based bonding mechanism for modular robotics. *IEEE Trans. Robot.* 27:876–90
86. Cong JJ, Fitch R. 2011. *The X-CLAW self-aligning connector for self-reconfiguring modular robots*. Paper presented at the 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) Workshop on Reconfigurable Modular Robotics: Challenges of Mechatronic and Bio-Chemo-Hybrid Systems, San Francisco, Sept. 25
87. Badescu M, Mavroidis C. 2003. Novel active connector for modular robotic systems. *IEEE/ASME Trans. Mechatron.* 8:342–51
88. Shen WM, Kovac R, Rubenstein M. 2009. SINGO: a single-end-operative and genderless connector for self-reconfiguration, self-assembly and self-healing. In *2009 IEEE International Conference on Robotics and Automation*, pp. 4253–58. New York: IEEE
89. Nilsson M. 2002. Connectors for self-reconfiguring robots. *IEEE/ASME Trans. Mechatron.* 7:473–74
90. Vasilescu I, Varshavskaya P, Kotay K, Rus D. 2005. Autonomous Modular Optical Underwater Robot (AMOUR) design, prototype and feasibility study. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pp. 1603–9. New York: IEEE

91. Eckenstein N, Yim M. 2017. Modular robot connector area of acceptance from configuration space obstacles. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3550–55. New York: IEEE
92. Kress-Gazit H, Fainekos GE, Pappas GJ. 2009. Temporal-logic-based reactive mission and motion planning. *IEEE Trans. Robot.* 25:1370–81
93. Mason MT. 2001. *Mechanics of Robotic Manipulation*. Cambridge, MA: MIT Press
94. Chen IM, Burdick JW. 1998. Enumerating the non-isomorphic assembly configurations of modular robotic systems. *Int. J. Robot. Res.* 17:702–19
95. Nguyen A, Guibas LJ, Yim M. 2000. Controlled module density helps reconfiguration planning. In *Algorithmic and Computational Robotics: New Directions*, ed. BR Donald, K Lynch, D Rus, pp. 23–36. Natick, MA: AK Peters
96. Park M, Chitta S, Teichman A, Yim M. 2008. Automatic configuration recognition methods in modular robots. *Int. J. Robot. Res.* 27:403–21
97. Chen IM, Burdick JW. 1995. Determining task optimal modular robot assembly configurations. In *Proceedings of 1995 IEEE International Conference on Robotics and Automation*, Vol. 1, pp. 132–37. New York: IEEE
98. Morrow JD, Khosla PK. 1995. Sensorimotor primitives for robotic assembly skills. In *Proceedings of 1995 IEEE International Conference on Robotics and Automation*, Vol. 2, pp. 1894–99. New York: IEEE
99. Paredis CJJ, Brown HB, Khosla PK. 1996. A rapidly deployable manipulator system. In *Proceedings of IEEE International Conference on Robotics and Automation*, Vol. 2, pp. 1434–39. New York: IEEE
100. Farritor S, Dubowsky S. 2001. On modular design of field robotic systems. *Auton. Robots* 10:57–65
101. Jing G, Tosun T, Yim M, Kress-Gazit H. 2016. An end-to-end system for accomplishing tasks with modular robots. In *Robotics: Science and Systems XII*, ed. D Hsu, N Amato, S Berman, S Jacobs, chap. 25. N.p.: Robot. Sci. Syst. Found.
102. Şucan IA, Kruse JF, Yim M, Kavraki LE. 2008. Reconfiguration for modular robots using kinodynamic motion planning. In *ASME 2008 Dynamic Systems and Control Conference*, pp. 1429–31. New York: Am. Soc. Mech. Eng.
103. Murray RM, Li Z, Sastry SS. 1994. *A Mathematical Introduction to Robotic Manipulation*. Boca Raton, FL: CRC
104. LaValle S. 2006. *Planning Algorithms*. Cambridge, UK: Cambridge Univ. Press
105. Latombe JC. 1991. *Robot Motion Planning*. Int. Series Eng. Comput. Sci. Vol. 124. New York: Springer
106. Lynch KM, Park FC. 2017. *Modern Robotics*. Cambridge, UK: Cambridge Univ. Press
107. Demaine ED, O'Rourke J. 2008. *Geometric Folding Algorithms: Linkages, Origami, Polyhedra*. Cambridge, UK: Cambridge Univ. Press
108. Yim M, Goldberg D, Casal A. 2000. Connectivity planning for closed-chain reconfiguration. In *Sensor Fusion and Decentralized Control in Robotic Systems III*, ed. GT McKee, PS Schenker, pp. 388–401. SPIE Proc. Vol. 4196. Bellingham, WA: Soc. Photo-Opt. Instrum. Eng.
109. Werfel J, Nagpal R. 2008. Three-dimensional construction with mobile robots and modular blocks. *Int. J. Robot. Res.* 27:463–79
110. Romanishin JW, Gilpin K, Claici S, Rus D. 2015. 3D M-blocks: self-reconfiguring robots capable of locomotion via pivoting in three dimensions. In *2015 IEEE International Conference on Robotics and Automation*, pp. 1925–32. New York: IEEE
111. Seo J, Yim M, Kumar V. 2013. Assembly planning for planar structures of a brick wall pattern with rectangular modular robots. In *2013 IEEE International Conference on Automation Science and Engineering*, pp. 1016–21. New York: IEEE
112. Seo J, Yim M, Kumar V. 2016. Assembly sequence planning for constructing planar structures with rectangular modules. In *2016 IEEE International Conference on Robotics and Automation*, pp. 5477–82. New York: IEEE
113. Cormen TH, Leiserson CE, Rivest RL, Stein C. 2009. *Introduction to Algorithms*. Cambridge, MA: MIT Press
114. Casal A. 2002. *Reconfiguration planning for modular self-reconfigurable robots*. PhD Thesis, Stanford Univ., Stanford, CA

115. Hou F, Shen WM. 2008. Distributed, dynamic, and autonomous reconfiguration planning for chain-type self-reconfigurable robots. In *2008 IEEE International Conference on Robotics and Automation*, pp. 3135–40. New York: IEEE
116. Dutta A, Dasgupta P, Nelson C. 2018. Distributed configuration formation with modular robots using (sub)graph isomorphism-based approach. *Auton. Robots*. In press. **https://doi.org/10.1007/s10514-018-9759-9**
117. Şucan IA, Kruse JF, Yim M, Kavraki LE. 2008. Kinodynamic motion planning with hardware demonstrations. In *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1661–66. New York: IEEE
118. Nagy Z, Oung R, Abbott JJ, Nelson BJ. 2008. Experimental investigation of magnetic self-assembly for swallowable modular robots. In *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1915–20. New York: IEEE