A Survey of Modular Self-Reconfigurable (MSR) Robots

Baha Hasan

December 6, 2011

Abstract

A self-reconfigurable robot is a robot built from potentially many modules which are connected to form the robot. Each module has sensors, actuators, processing power, and means of communicating with connected modules. The robot autonomously changes shape by changing the way these modules are connected. Self-reconfigurable robots have a high degree of robustness, versatility, scale extensibility, and adaptability. This makes these robotic systems interesting to study. Since the late 1980’s, many systems of self-reconfigurable robots have been developed. This report gives a survey of many of these projects, and discusses several interesting features and capabilities of the robotic modules and structures. An overview of each robot system is presented, highlighting the most interesting aspects of the system. Following the survey of the various robot projects, a general discussion of self-reconfigurable robots is given, summarizing the main features and concerns of physical characteristics of self-reconfigurable robots and their modules, mechanisms of locomotion and reconfiguration, capabilities and applications of self-reconfigurable robots, and challenges for self-reconfigurable robot research.

1. Introduction

Reconfigurable robots are robots built from many independent homogenous or heterogeneous modules. Each module has actuators, sensors, processing power, memory, and means of communicating and connecting to neighbors. If a reconfigurable robot is autonomously able to change the way these modules are connected, the robot is a self-reconfigurable robot. Through reconfiguration of their modules, self-reconfigurable robots can change their shape and adapt to the specific task environment. Therefore, self-reconfigurable robots hold the promise to be a powerful robotic platform which enables us to explore the interaction between morphology and control [1]. Common examples of reconfigurability in action include transforming between snake shapes for moving through holes and legged locomotion for traversing rough terrain, and using reconfiguration for locomotion as shown in Fig.1 [2].
There are several ways of categorizing MSR robotic systems. One is based on the regularity of locations for attaching; lattice vs. chain vs. mobile, and another is based on the methods of moving between those locations; stochastic vs. deterministic [3]. Several systems exhibit hybrid properties [4].

**Lattice** A lattice based MSR system has modules arranged nominally in a 2D or 3D grid structure. For this category, there are discrete positions that a given module can occupy. In contrast to chain-based architectures where modules are free to move in continuous space, the grid based structure of lattice systems generally simplifies the reconfiguration process. Kinematics and collision detection are comparatively simple for lattice systems. An example is shown in Fig. 2.

**Chain** A chain based MSR system consists of modules arranged in groups of connected serial chains, forming tree and loop structures. Since these modules are typically arranged in an arbitrary point in space, the coordination of a reconfiguration is complex. In particular, forward and inverse kinematics, motion planning, and collision detection are problems that do not scale well as the number of modules increases. An example is shown in Fig. 3.

**Mobile** The mobile class of reconfiguration occurs with modules moving in the environment disconnected from other modules. When they attach, they can end up in chains or in a lattice. Examples of mobile reconfiguration devices include multiple wheeled robots that drive around and link together to form trains, modules which float in a liquid or outer space and dock with other modules.

**Stochastic** In a stochastic system, modules move in a 2D or 3D environment randomly and form structures by bonding to a substrate and/or other modules. Modules move in the environment in a passive state. Once a module contacts the substrate or another module, it makes a decision about whether it will bond to the structure or reject a bond. The time that it takes for the system to reach a desired configuration is probabilistically bounded. The reliance on environmental forces allows the mechanical actuation to be simplified as only bonding actuation is required internal to the module. An example is shown in Fig. 4.

**Deterministic** In deterministic MSR systems, modules move or are manipulated directly from one position to another in the lattice or chain. The positions of each module in the system are
known at all times. The amount of time it takes for a system to change from one configuration to another is determined. A module’s reconfiguration mechanism requires a control structure that allows it to coordinate and perform reconfiguration sequences with its neighbors.

Figure 2: Crystalline. The Crystalline system is a lattice style robot Developed by Rus et al. [5] at Dartmouth University.

Figure 3: PolyBot, Yim et al. developed the PolyBot chain-type MSR system at Palo Alto Research Center [6] (PARC, formerly Xerox PARC).

Figure 4: Stochastic 3D. A stochastic MSR robotic system has been demonstrated in both 2D and 3D by White et al. [7] at Cornell University.
Self-reconfigurable robots have characteristics which make them interesting in their own right, because these characteristics might push the limit for which kind of task-environments are suitable for robots. The following characteristics have been stressed in the literature:

- **Versatility.** The modules can be combined in different ways making the same robotic system able to perform a wide range of tasks [8].

- **Adaptability.** While the self-reconfigurable robot performs its task it can change its physical shape to adapt to changes in the environment [9].

- **Robustness.** Self-reconfigurable robots are made from many identical modules and therefore if a module fails it can be replaced by another [6].

- **Scale Extensibility.** The size of the robot system can be increased or decreased by adding or removing modules [10].

Self-reconfigurable robots can solve the same tasks as traditional robots, but it is probably not cost-effective to use a self-reconfigurable robot unless its unique functionalities are needed [6]. The versatility of these robots make them suitable in scenarios where the robots have to handle a range of tasks. The robots can also handle tasks in unknown or dynamic environments, because they are able to adapt to these environments. In tasks where robustness is of importance it might be desirable to use self-reconfigurable robots. Finally, their scale-extensibility means that the size of the robot can be adjusted to the size of the problem. Even though real applications for self-reconfigurable robots still are to be seen, a number of specific applications have been envisioned [6,8,9]: firefighting, search and rescue after an earthquake, battlefield reconnaissance, planetary exploration, undersea mining, and space structure building. Other possible applications include entertainment and service robotics.

2. **History and State of the Art**

The concept of modular self-reconfigurable robots can be traced back to the “quick change” end effector and automatic tool changers in computer-controlled machining centers in the 1970’s. Here, special modules, each with a common connection mechanism, were automatically interchanged on the end of an electro-mechanical or robotic arm [3]. The concept of applying a common connection mechanism to an entirely modular robot was introduced by Fukuda with the biologically-inspired CELlular roBOT (CEBOT) in the late 1980’s [11]. Here each CEBOT module is 18 x 9 x 5 cm and weighs approximately 1.1 kg. These units have independent processors and motors, and can communicate with each other to approach, connect, and separate automatically.

In the early 1990’s, modular reconfigurable robots were shown to have the ability to perform the task of locomotion. In 1994, Yim explored many statically stable locomotion gaits with Polypod. Polypod [11] is an MSR robot that is significantly lighter and smaller than CEBOT. A module by itself could not locomote, but through the collective behavior of the system of many modules it could move itself from place to place and achieve many different locomotion gaits [12] such as a slinky, caterpillar, or rolling track gait.
Through this work it became clear that controlling a system with a large number of modules is complex. Initial Polypod control used a gait control table to program simple gaits on a modular robot using prescribed motions. In addition to the complexity of coordinated control, the complexity of arbitrary configurations and the sequence of reconfigurations to attain those configurations quickly developed into an interesting computational problem.

Chirikjian and Murata developed lattice style configuration systems in [14, 15]. The lattice style robots have modules which sit on a lattice and make it easier to represent the configurations computationally. As a result this style of system quickly became popular among computational roboticists.

In the later 1990’s Rus [16] and Shen [17], also developed hardware but their larger contributions came in the distributed programming aspects. This included seminal trends in developing provable distributed algorithms [18] and decentralized control based on local communication [19]. Two of the areas of research include configuration self-recognition and kinematic planning of the motions for rearrangement between configurations.

One of the more interesting hardware platforms recently developed has been the modular transformer (MTRAN) series by Satoshi Murata et al. [20]. This system is a hybrid chain and lattice system. It has the best of both systems: the good task performance of a chain system mixed with the good reconfiguration performance of a lattice system.

There is a growing number of research groups actively involved in modular robotics research projects, as can been seen in the survey paper [5], a survey chapter in [6], and two special issues in robotics journals [7] and [8]. A number of algorithmic advances have complemented hardware development. See, for example, [6]–[10] and [12]–[14]. A brief overview of many of these projects is follows.

3. Control Systems for MSR

Since MSR systems are designed to be versatile, with numerous configurations for a set of modules, the problem of recognizing and choosing useful configurations is a central area of research. The organized control of modular structures is often a complex task, involving coordinated communication between modules (each which has a processor), central controllers, and in some cases, a human user [3].

The main characteristics of the MSR’s control systems are directly derived from the characteristics of self-reconfigurable robots. The following characteristics have been stressed in the literature:

- **Robustness.** The control algorithm should be robust to module failure and communication errors.

- **Adaptability.** The control system should exploit sensor input to adapt the robot to the environment.
- **Versatility.** The control system should enable the robot to perform many different tasks.

- **Scale Extensibility.** The control system should allow for changes in the number of modules.

- **Scalability.** The control system should be able to handle systems consisting of many modules.

The proposed control systems for self-reconfigurable robots can be divided into two main categories: centralized systems and distributed systems. In a centralized system the modules of the robot are controlled by a centralized host. In distributed systems a controller is running on each module. Distributed systems can be further sub-categorized into systems based on global or local information. We refer to distributed control systems based on local information as emergent control systems [1].

Centralized control systems are traditionally used to control robots, but are not suitable for self-reconfigurable robots, the reason being that if changes occur in modules of the robot or the environment a time-consuming re-planning process has to take place centrally before the robot can continue. Specifically, this means that centralized control systems do not handle the characteristics of robustness, adaptability, and scale-extensibility well. Furthermore, since the centralized host is responsible for controlling all modules, it will, with an increasing number of modules, be overloaded. This implies that the efficiency of the centralized system will drop and therefore it does not scale with an increase in the number of modules [1].

Distributed control systems are a step forward compared to centralized solutions. If global information is needed this has to be reliably communicated to the modules. This dependency makes systems dependent on global information less robust. Furthermore, scalability might also be an issue, because modules in large systems might spend substantial amounts of time waiting for global information to arrive. Because of the local nature of the controllers in emergent control systems it can be difficult to find local rules which make the system converge toward the desired global behavior [1].

The control systems proposed so far are mainly used for either locomotion or self-reconfiguration. Locomotion in lattice robots is achieved through a cluster-flow mechanism based on emergent control where modules from the back of the robot are moved to the front and through this motion produce locomotion. In chain-type robots the robot does not change shape; instead it locomotes using the actuators of the modules to produce locomotion gaits. There exist no emergent control algorithms for the control of locomotion of chain-type self-reconfigurable robots [1].

Self-reconfiguration algorithms exist for both chain-type and lattice-type robots. Chain-type self-reconfiguration is quite well understood, at least in theory, but this is not the case for lattice-type robots. In lattice-type self-reconfiguration the use of centralized systems is problematic, because the planning problem is computationally hard in addition to the general problems of centralized systems already mentioned. Distributing the problem does in some cases solve the scalability issue but if global information is used robustness is still an issue. Emergent solutions do not have
these problems because they are not based on planning, but emergent solutions often have other problems: emergent solutions are often not systematic, efficient, or guaranteed to converge [1].

3.1 Control Architectures

A design philosophy behind modular robots is that each module is very simple. A module by itself cannot achieve much, but modules arranged together in a system can achieve complex tasks such as manipulation and locomotion. Similarly, the control of a single module is usually simple whereas controlling a system of many modules becomes difficult very quickly. For the overall system, different control architectures have been implemented which we will describe in more detail [3].

In large part, the implementation of a control architecture depends on the communication structure upon which it is built. Communication between modules can be achieved through a global bus such as CANbus (Controller Area Network, a popular automotive and more recently robotics communications protocol) and/or locally using neighbor-to-neighbor communication such as infra-red (IR) emitter/detector pairs. Many systems use both (Polybot, CKBot, M-TRAN, CONRO and Superbot). Wireless communication is also possible which is architecturally similar to a global bus. In the YaMoR system [21], Bluetooth wireless is the sole means of inter-module communication. ATRON and Crystalline modules [22,5] use only local nearest neighbor IR communication.

As mentioned earlier, control architectures can be implemented in either a centralized or decentralized fashion. An example of centralized control architecture is implemented on [6]. Each module has its own controller that positions its local actuator. In addition, a master controller communicates to the module controllers to set local behaviors such as setting desired joint angles under position control. In other words, a designated unit sends commands to all the individual modules and synchronizes the action of the whole system. A simple method of implementing this control is to use a gait control table. The gait control table is an nxm matrix where m is the number of modules and n is the number of steps of the gait. Each cell in the table holds the desired joint angle for a module. Each column of angles corresponds to the sequence of joint angles for a given module. The controller steps through this table row by row and sends these angles to the corresponding module. Typically stepping through the table occurs at a specified rate, so the vertical axis can represent time. Each module takes the next desired joint angle in the table and interpolates in joint space. The time between steps sets the joint velocity so desired motions have C1 continuity in joint space.

Shen et al. propose a control that is based on biological hormone systems in [9]. The basic idea is that an inter-module “hormone” message is a signal that triggers different actions in different modules while leaving the low level execution of these actions to the individual modules. Based on this principle, Shen et al. designed a control mechanism that lies somewhere between master and master-less control in that typically one or more modules need to start the hormone messages. It reduces the communication cost for locomotion controls, yet maintains some degree of global synchronization and execution monitoring.

At its root, the hormone is a local message passing system where modules can receive, act on or change messages as they are passed from module to module. An advantage of this type of control
is that modules are treated identically without labels or identification numbers; instead the topology of a configuration is the differentiator and thus has a great bearing on the implementation. This lends itself well to simple locomotion control such as undulating gaits however, developing arbitrary motions can be more difficult to implement.

A fully decentralized planning system has been developed by Rus et al. In [18] an algorithm modeled after cellular automata is described. Cellular automata (CA) control uses local rules that are the same for all modules. A rule can be viewed as having a set of pre-conditions. If all those preconditions are satisfied, then a certain action is applied. For example, for a given cell, the pre-conditions could be whether a cell exists at a certain location, whether a cell does not exist at a certain location, and whether a cell is empty. If all preconditions are satisfied, the cell moves itself in a certain direction. Rather than having one master controller being in control of the whole system, modules think for themselves in a parallel distributed fashion. All modules run on the same rules and all modules are programmed with the same code. Just as the hormone method of control adds some complexity to the development of arbitrary motions, it is also difficult to do in the CA case.

An example of a hybrid architecture in which global as well as local communication is used is given in [23]. In this work the ability for a modular robot to repair itself is demonstrated by having the robot reassemble into one connected component after disassembly from a high energy event. As a system assembles itself, the connectivity of the robot changes many times. Having disparate disconnected pieces requires a level of decentralized control, however as the system comes together, the modules must act in a coordinated manner as well.

4. Examples Self-Reconfigurable Systems

4.1 Polypod (1993)

A chain-type MSR system. The Polypod robot, shown in Figure 5, was developed by M. Yim [24]. The Polypod consists of two types of module: segment and node. A segment has two connection plates at opposite ends and two degrees of freedom. The node modules have no degrees of freedom, but six connection plates and contain a battery.

Figure 5: A Polypod segment module (left). A Polypod node module (right).
4.2 Fracta (1994)

A lattice-type MSR system using 2D fracta [15] is capable of reconfiguration, transportation in two dimensions, and self-repairs. Three physical fracta have been built that are 125 mm in diameter and 160 mm in height, with a possibility for micro-scale fracta if the magnetic connections are replaced with electrostatic connections. Several fracta prototype modules are shown in Figure 6. Each fractum contains a microprocessor and an infrared optical communication channel. The individual fracta are homogeneous, autonomous, and use local relations to attach, detach, and cooperate with neighbors. The fracta are able to assemble into arbitrary three-dimensional structures and the structures can move as a whole, as well as discard a damaged part, thus performing self-repair. Physical experiments have demonstrated basic fracta movements. Simulations of self-assembling fracta have been performed, where fitness evaluation, diffusion, and activation determine the sequence of movements towards the desired configuration. Future work includes further developing self-repair and adapting to changes in the environment by changing shape.

Figure 6. Three Fracta Modules.

4.3 Metamorphic (1996)

Metamorphic robots [25] are two-dimensional, homogenous, lattice-based reconfigurable robots with hexagon or square-shaped modules, as illustrated in Figure 7. A module performs locomotion by rolling or sliding over neighboring modules. Hexagon modules roll over their neighbors while square modules slide over their neighbors in a vertical, horizontal, or diagonal direction. Besides climbing over adjacent modules, each module can perform basic computation and connect or disconnect from neighboring modules with mechanical hooks or electromagnets. Modules are controlled by an external processor. Experiments using a physical prototype tested basic linking and locomotion abilities. The metamorphic structures could function as a swarm of connected robots that act collectively. Possible applications are obstacle avoidance in highly unstructured and constrained environments, forming bridges and other structures, encircling objects (such as recovering space satellites), and performing inspections (e.g. of nuclear reactors).
4.4 MEL 3D-Units (1998)

The 3D-Unit robot [8] was the first three-dimensional lattice-type robot that was prototyped. Two units were built, each spanning 26.5 centimeters. Their shape is based on a regular hexagon. As depicted in Figure 8, a cube is in the center and an arm extends in each of the six directions. The modules are homogenous and are capable of changing their local connection, communicating with neighbors, and processing information. The connectors include a grasping structure with a key and keyhole mechanism. Each module contains sensors for position, angle, and contact. Two units must be moved together, where one is used as a pivot for the other. The modules can move on the plane or flip to the orthogonal plane. Physical tests involved rotational movement and lifting of modules. Simulations were performed on reconfiguring from a ladder shape to a tower shape. The simulations were successful for as many as 20 modules. The units of moveable type compare each reachable state to the goal and choose with a greater probability the move resulting in the smallest difference between the goal. The structures are also capable of self-repair. Possible real-world applications include use in hazardous or remote environments, such as space, deep sea, and radioactive environments. A limitation of this design is that some configurations are difficult to reach. Future plans involve constructing more complex structures. Two real hardware modules are shown in Figure 9.
4.5 The Robotic Molecule (1998)

The Molecule [16] is a 3D lattice-based robot with modules comprised of two atoms connected by a bond, as shown in Figure 10. The molecules exist in two versions, one with all female connectors and one with all male connectors. Connections between modules are established with electromagnets. Movement of these modules involves one atom rotating around the other atom. The basic types of motion are straight-line traversal and 90 degree convex (for climbing down) and concave (for climbing up) transitions to adjacent surfaces. Physical experiments have tested these modes of locomotion by using one prototype Molecule and a simulated lattice structure. Sixteen physical Molecules have been built, and the vision is to eventually create hundreds of these robots. The correctness of reconfiguration sequences has been verified with a Prolog simulator. The simulated structures are able to climb stairs, build a tower structure, and form a wall by tiling. Trajectory planning is implemented using graph search. A meta-module consisting of 16 Molecules with three levels of hierarchy has been constructed. This hierarchical structure allows the execution of polynomial-time planning algorithms. Physical experiments of locomotion modes have also been conducted on this meta-module. Future applications of Molecule-based structures may involve traversing different types of terrain, manipulating objects, and performing basic sensing functions. Work in the future includes various issues concerning general global motion planning algorithms. There are questions of the minimum number of Molecule robots that are required to satisfy the restrictions of known planning algorithms, as well as whether parallel transitions can speed up the planning and motion process. In Figure 11 two structures are shown built from simulated and real modules.
4.6 RIKEN Vertical (1998)

The “RIKEN Vertical” system, from the Institute of Physical and Chemical Research (RIKEN) [26], is a lattice –type 2D MSR system in the vertical plane. Modules have two degrees of freedom, one to control the extension/ extraction of the arm, and one to rotate the arm. Vertical robots have the ability to reconfigure against gravity, as in the case of climbing stairs like structures. Besides climbing stairs, another potential use for these robots would be to build a bridge structure for transporting cargo across a gap. Four prototype modules were built. Each has a cubic body with an edge length of 90 mm and a pair of arms. A schematic of the modules is given in Figure 12. Modules are connected with permanent magnets and a module can communicate with its neighbors. The arms of the modules perform rotating and sliding motions to change the bonding configuration of the modules. Movements are determined by local, minimum interactions between neighbors, very much like cellular automata. In a physical experiment, robots created a stairs-like structure and another robot was lifted to the top of the stairs. Simulations of creating and dismantling stairs were conducted using 15 modules. To build a stairs structure, a module can either move upper-left or left, and to dismantle the stairs, a module can either move lower-right or right. Future work involves building robots that contain processors and sensors.

Figure 12: RIKEN Vertical modules (left). RIKEN Vertical modules in a stair-structure (right).
4.7 I-Cubes (1999)

A lattice MSR project [27] consists of active links and passive cubes, as illustrated in Figure 13. The links are moveable and the cubes can be rotated, translated simultaneously in two directions, and act as a pivot joint for a moving link. The structures assembled from I-Cubes are potentially able to move over obstacles, climb stairs, traverse through tunnels and pipes, manipulate objects, form bridges and towers, and be utilized for space applications. Experiments using a physical prototype demonstrated basic link function and cube movement. A sequence of actions for climbing a step and building a tower were created manually with the aid of a graphical interface. The cube component has an edge length of 8 cm, although the desired length is 6 cm. Future millibots are a possibility. The connections are established by a cross or cone-shaped piece that locks into place. The modules are able to sense joint position. The structures move by means of joint rotations of the cubes as shown in Figure 14. The links are controlled externally by buttons or a graphical user interface. Future work involves constructing smaller and lighter cubes, constructing more links and cubes, enabling the modules to be autonomous, and devising motion planning schemes that combine learning and search techniques.

![Figure 13: I-Cube with a link moving from one cube face to another.](image)

![Figure 14: I-Cubes motion with four links and four cubes.](image)

4.8 Crystalline (2000)

Crystalline robots [5] are lattice homogenous square (cubic) modules that perform locomotion using expansion and contraction movements similar to muscles and amoeba. A prototype module appears in Figure 15. The modules can also connect and disconnect to other modules by means of a key and lock (channel) mechanism, and they contain position sensors. Modules are autonomous in that they contain their own processor and power supply. Ten physical modules have been built, and each is 7 inches tall and 2-4 inches wide, depending on whether it is contracted or expanded. Modules are able to perform locomotion and automated shape metamorphosis. Physical experiments have involved a module expanding and connecting with a neighbor, inchworm locomotion with two atoms, and performing atom relocation by propagating across a row in the crystal structure. Simulations have involved inchworm locomotion and reconfiguring a dog shaped structure into a couch-shaped structure, which was planned manually. Future work includes improving the hardware and developing distributed reconfiguration algorithms.
**4.9 POLYBOT (2000)**

A self-reconfigurable 3D robot named Polybot [6] is constructed from a chain structure consisting of segment and node modules. There are three generations (G1, G2 and G3) of PolyBot implemented, as illustrated in Figure 16. Some of the characteristics of these generations are summarized in table 4.1.

In Figure 16, a chain structure consisting of nine modules is displayed in G2. These structures are capable of several types of locomotion, including rolling (for flat terrain), earthworm motion (for tunnels and steps), and spider-like motion (for hills). Physical experiments have involved several modes of locomotion: earthworm locomotion, snake-like locomotion, a rolling track, caterpillar-like locomotion, cilia-like locomotion, 6 legged locomotion, slinky-like locomotion, and spider-like locomotion. Arm manipulation and ball balancing have also been demonstrated. Simulations of additional forms of locomotion include cartwheel locomotion, carrying an object while rolling, a rolling loop, and slinky locomotion on an x-y grid. Potential real-world applications for Polybot include planetary exploration, undersea mining, and search and rescue operations. Work in the future will involve increasing robustness, implementing self-repair functionality, and addressing the issue of the motion planning space becoming exponential in the number of modules.
Figure 16. Polybot generations.

<table>
<thead>
<tr>
<th>Specification</th>
<th>POLYBOT-G1</th>
<th>POLYBOT-G2</th>
<th>POLYBOT-G3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reconfigurability</td>
<td>Manual</td>
<td>Self</td>
<td>Self</td>
</tr>
<tr>
<td>Size cm</td>
<td>$7 \times 7 \times 7$</td>
<td>$11 \times 7 \times 6$</td>
<td>$5 \times 5 \times 4.5$</td>
</tr>
<tr>
<td></td>
<td>$5 \times 5 \times 4$ (G1v4)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Weight</td>
<td></td>
<td>416g</td>
<td>200g</td>
</tr>
<tr>
<td>CPU</td>
<td>Motorola 68HC11 PIC 16F877(G1v4)</td>
<td>Motorola power PC 555+ 1M external RAM</td>
<td>Motorola power PC 555+ 1M external RAM</td>
</tr>
<tr>
<td>Bus</td>
<td>RS232 RS485(G1v4)</td>
<td>2 CAN buses / Module</td>
<td>2 CAN buses / Module</td>
</tr>
<tr>
<td>Communication &amp; Control</td>
<td>50Hz PWM signal</td>
<td>Electrical SMA actuator</td>
<td>Electrical SMA actuator</td>
</tr>
<tr>
<td>Connection mechanism</td>
<td>Mechanical</td>
<td>SMA actuator</td>
<td>SMA actuator</td>
</tr>
<tr>
<td>Sensors</td>
<td>-----------</td>
<td>Infrared emitters and detectors, proximity sensor</td>
<td>Infrared emitters and detectors, proximity sensor</td>
</tr>
<tr>
<td>Motor</td>
<td>Servo-0.7Nm</td>
<td>Brushless DC Motor 5.6 Nm</td>
<td>MAXON pancake 21Nm/Kg</td>
</tr>
<tr>
<td>Power Supply</td>
<td>6V/0.5A Batteries (G1v4)</td>
<td>Two power buses: 8 V for electronics 12-24 V for Motor (High power requirement)</td>
<td>Single power bus: 35V</td>
</tr>
</tbody>
</table>

Table 4.1. Comparison between PolyBot three generations.
4.10 CONRO (2000)

CONRO (CONfigurable RObot) modules [17] are self-sufficient, autonomous, homogeneous, three-dimensional, and form into chain-like structures. Each module has a body and active and passive connectors, in addition to an infrared communication system and sensors and actuators. A pin/hole mechanism allows modules to connect. Modules can be assembled into snake and hexapod configurations. A photo of a physical module appears in Figure 17. Physical experiments have tested basic snake motion and a hexapod structure standing up. The structures can reconfigure based on the environment and current task. Small robots can merge and large robots can split. Many small robots can perform a task in parallel. Twenty physical modules with a length of 104 mm have been built. A work in progress is software to coordinate reconfiguration and locomotion of robots.

![CONRO modules](image1)

Figure 17 A close-up of a CONRO module (left). A hexapod built from CONRO modules (right). (USC Information Sciences Institute.)

4.11 Pneumatic (2002)

Pneumatic modules [28] are homogenous, three-dimensional, cubic-shaped structures, with pneumatic actuators consisting of flexible bellows, as illustrated in Figure 18. Motion and connection is achieved using compressed air from these bellow mechanisms. This scheme is inspired by animals such as worms and caterpillars with hydrostatic skeletons. The modules can perform rotational movements and stable contraction and elongation. Two prototype modules were built and basic rotational movement of these two modules was tested. Each module has an edge length of 20 centimeters. The goal is to scale up the size of these modules in order to utilize them for various space applications. Another possible application is to build a bridge structure and let a moving load pass through. Future plans are to improve the efficiency of the air supply and to install a power supply in each robot module.
Telecube modules [27, 29] are 3D cubic-shaped units that perform motion by expanding and contracting the sides of the cube, similar to the way tiles move in an 8-puzzle. Both contracted and expanded versions are shown in Figure 19. The modules are homogeneous with simple communication and infrared sensors that are able to gauge the extension of each side of the cube, read the contact sensor on each face, and determine whether or not they are connected to a neighbor. The connection mechanisms are implemented using permanent switching magnets. Two physical modules have been built, with another 5-20 planned. Each side of the module cube is 6 cm in the contracted state. Simulations have tested basic reconfiguration algorithms. The researchers have constructed meta-modules composed of 8 modules with additional modules embedded within. This construction simplifies reconfiguration and allows more types of configurations that otherwise would not be possible. Possible capabilities of the structures formed by Telecube modules include locomotion, object manipulation, sorting, interacting with other systems, and adapting to the current environment.

A reconfigurable robot composed of a hybrid of lattice and chain structures is M-TRAN (Modular TRANsformer) [20, 30, 31, 32]. Three types of prototype have been developed called M-TRAN I, II, III. Their specifications are listed in Table 4.2.

The modules are homogenous and consist of active and passive boxes that can attach, detach, rotate, and lift other modules. The assembled structures are able to manipulate objects, perform autonomous 3D locomotion resembling a 4-legged walker, caterpillar, an H-shape, or multiple walkers, and assume a tower structure. A 4-legged walker and a caterpillar structure constructed from M-TRAN modules are shown in Figure 20. Locomotion sequences are determined using genetic algorithms, a central pattern generator, and an automatic motion planner. Ten physical modules have been built, each of size 60 cubic mm for each of the two boxes. The modules contain connection surfaces with permanent magnets as shown in Figure 21.

There are sensors for position and orientation. Hardware with mechanical connectors and infrared sensors is being developed. Modules may be connected and function as a larger module, called a “meta-module,” which simplifies the reconfiguration problem. An interactive motion design interface using the Open GL library has been developed. The software uses a locomotion planner that combines global and local planning using database rules. Reconfiguration and climbing over obstacles have been demonstrated. Physical experiments have demonstrated basic reconfiguration and forward motion. The plan devised by the software is transferred to the hardware. A possible real-world application is search and rescue operations. Future plans involve building simpler and smaller modules, more transformations between structures, implementing search and learning techniques for motion generation, devising distributed control schemes for facilitating self-repair, and finding optimal configurations for a given task or environment.

Figure 20. M-TRAN modules forming a 4-legged walker (left) and caterpillar (right) [20].

Figure 21. M-TRAN module design [20, 32].
<table>
<thead>
<tr>
<th>Specification</th>
<th>M-TRAN I</th>
<th>M-TRAN II</th>
<th>M-TRAN III</th>
</tr>
</thead>
<tbody>
<tr>
<td>Size</td>
<td>66</td>
<td>60</td>
<td>65</td>
</tr>
<tr>
<td>Weight</td>
<td>440g</td>
<td>400g</td>
<td>420g</td>
</tr>
<tr>
<td>Master CPU</td>
<td>Basic stamp II</td>
<td>Neuron Chip</td>
<td>SH II (Renesus)</td>
</tr>
<tr>
<td>Bus</td>
<td>UART</td>
<td>LON Bus</td>
<td>CAN Bus</td>
</tr>
<tr>
<td>Salve CPU</td>
<td>2 × PIC</td>
<td>2 × PIC</td>
<td>3 × H8</td>
</tr>
<tr>
<td>Communication &amp; Control</td>
<td>Centralized Control</td>
<td>Global synchronization Distributed oscillator Control</td>
<td>Asynchronous Distributed oscillator Control</td>
</tr>
<tr>
<td>Wireless Communication</td>
<td>---------</td>
<td>RF command to one module</td>
<td>Two-way communication via Bluetooth</td>
</tr>
<tr>
<td>Connection mechanism</td>
<td>Permanent magnets and SMA actuator</td>
<td>Mechanical</td>
<td></td>
</tr>
<tr>
<td>Batteries</td>
<td>10 Proximity (IR) Inclinometer</td>
<td>Lithium ion</td>
<td>Lithium polymer</td>
</tr>
</tbody>
</table>

Table 4.2: Specifications of M-TRAN [32].


ATRON [33, 34, 35] robots are lattice-based structures consisting of homogeneous modules composed of two hemispheres, as shown in Figure 22. The modules are capable of sensing tilt, distance, and gravity; rotating around their equator; connecting and disconnecting with modules; sensing connected neighbors; recognizing dead modules, external objects, and nearby modules; communicating with neighbors; and detecting being lifted or tipped over. Infrared diodes are used for communication. Connectors use a point-to-point male/female hook scheme. A module is unable to move on its own; it must move with the aid of one of its neighboring modules. A group of modules can self-reconfigure in 3D. The researchers have considered control of metamodules. Potential applications for these robots are: production lines (packaging), assembly lines (sorting), cleaning and maintaining hazardous machinery, search and rescue efforts, nanorobotics, space applications, assisting the disabled, entertainment, and automatic construction.

---

Figure 22. (a) A single ATRON module. (b) Exploded view of the ATRON mechanics [35].
Approximately 100 physical modules based on LEGO®s or styrofoam have been constructed and each is 11 cm in diameter. The goal is to eventually build thousands to trillions of these modules on the millimeter or micrometer scale in the form of intelligent dust or nano-robots. Currently, scalability is unclear in terms of number of modules and it appears to be more difficult for lattice structures. The ATRON self-reconfigurable robot is capable of performing locomotion in several different ways. Distributed control strategies have been used to control locomotion. The two main categories are fixed topology locomotion (cars, walker, and snake) and locomotion by self-reconfiguration (cluster flow) [35]. This can be illustrated in Figure 23.

![Figure 23. Different configurations all consisting of 7 ATRON modules. (a) configuration capable of performing cluster walk. (b) four wheeled car. (c) snake configuration. The robot is able to autonomously reconfigure between any of these configurations that have different locomotion characteristics [35].](image)

### 4.15 SuperBot (2004, 2006)

Superbot robots [36] combine features and advantages of M-TRAN [20, 30, 31, 32], CONRO [17], and ATRON [33, 34, 35], as illustrated in Figure 24. These robots are three-dimensional and consist of both lattice and chain structures. Both a physical prototype and Open Dynamics Engine based simulations have been studied. The modules have three DOF (pitch, yaw, and roll) and can connect to each other through one of the six identical dock connectors. They can communicate and share power through their dock connectors. For high-level communication and control, the modules use a real-time operating system and the hormone-inspired control developed for CONRO as a distributed, scalable protocol that does not require the modules to have unique IDs [4]. Simulation experiments have been conducted on several modes of locomotion, including snake, caterpillar, insect, spider, rolling track, and H-walker.

Capabilities of a single real SuperBot module are moving, turning, sidewarding, maneuvering, traveling, reconfiguring, recovering from failure, and flipping. Other potential functions include climbing, lifting objects, and using tools. Possible real-world applications include transportation, exploration, construction, inspection, maintenance, resource utilization, and support for astronauts. Modules contain a position sensor and a 3D accelerometer for gravity. Future work will involve reconfiguring from one mode into another, and testing different terrains and travel distances. An issue to be addressed is the tradeoff between efficiency and adaptability of the robot structures.
4.16 Programmable Parts (2005, 2006)

The programmable parts project [37, 38] is based on the use of graph grammars for determining the behavior of the modules. The rules of the graph grammar correspond to chemical reactions. It is possible to design a rulebook so that parts are able to assemble into any desired structure [37]. The modules passively float on an air table and bind upon random collisions. The local grammar rules determine whether or not modules will stay bound or detach. Besides binding and detaching, modules are able to communicate with their neighbors. Connections between modules are made with permanent magnets. The physical modules can form 2D hexagonal structures that will eventually be capable of locomotion, self-repair, and transport. Potential real world applications of these structures include planetary exploration and mass production of 3D objects. A physical prototype has been built and simulations using Open Dynamics Engine have been developed. A few physical modules are shown in Figure 25. Six physical modules have been constructed and the simulation is capable of simulating hundreds of modules. In both physical experiments and simulations, the modules formed two-dimensional hexagon structures, but in the simulations, 50 modules (instead of 6) were used, and three different grammars were tested and compared. The physical modules are triangular-shaped with an edge length of 12 centimeters and a height of 4 centimeters. Future plans involve building 100 physical modules, scaling down the size of the modules, and exploring different grammars that will assemble modules into other shapes and define processes such as locomotion, self-repair, and transportation.

Odin [39, 40] is a heterogeneous hierarchical lattice-based robot composed of two different types of modules cylinder- shaped links and sphere-shaped joints. A prototype consisting of six links and four joints in a tetrahedron structure has been constructed, and this prototype is illustrated in Figure 26 (a). The links are 35 mm in diameter and 110 mm in length, and the joints are 50 mm in diameter. Connections between modules involve a lock and key mechanism. In the current revision of the system the joints are passive elements of the mechanical structure, allowing both power sharing and communication over RS- 485 among different link modules. Different functionalities, such as like passive structural support, sensing, actuation and power storage, are implemented by different kinds of link modules [40].

The modules use a hybrid global and local communication system. The electronics of the link modules consists of a general board common to all the different links and a specific one related to the different incorporated functionalities. The control software runs on the CPU of the general board, an Atmel AT91SAM7S micro-controller with 256Kbytes of program memory and 16 Kbytes of RAM [40].

Locomotion is based on distributed role-based control. Physical tests were conducted to demonstrate forward movement, but so far Odin is only capable of locomotion in two dimensions. Future work includes performing more complex tasks, installing a power supply on each robot, and building four types of link modules -- structure, actuation, power, and sensing. Additional modules will also be built, which will allow multi-level hierarchies to be created. There is a question of which level of the hierarchy to implement specific functionalities. These robots are actually not self-reconfigurable, but it might be possible to achieve self-reconfiguration at a higher level.

Figure 26. (a) The Odin reconfigurable robot in a tetrahedron configuration [39]. (b) Odin robot with 21 modules in a closely packed lattice (cubic closed-packed) configuration. The configuration consists of 8 telescoping link modules (black links), 6 rigid passive link modules (white links), and 7 joint modules. The links are connected to the joints with a flexible connector [40].
4.18 UBot (2009, 2011)

A reconfigurable homogenous robot composed of a hybrid of lattice and chain structures is UBot [41, 42]. Figure 27(a). UBot module is a cubic structure based on double rotational DOF, and has four connecting surfaces. UBot modules are designed to be compact, strong, flexible, and capable of performing efficient locomotion, self-reconfiguration and manipulation tasks. Each module has its own power and wireless communication module. A hook-type connecting mechanism with compact structure is designed— which is completely embedded in module— which works quick and reliable and has the function of self-lock after connected. The UBot module has enough torque and strength to lift four neighbouring modules. The modules are classified to active module and passive module by the function of connecting mechanism, as shown in Figure 27 (b). Active module is composed of one twin-rotation mechanism and four active connecting mechanisms. Passive module is composed of one twin-rotation mechanism and four passive connecting mechanisms. The overall size of both active and passive module is 80mm×80mm×80mm, the mass of active module is 350g, and the mass of passive module is 280g [41].

![Image](image_url)  
(a) Figure 27. (a) A worm-like robot of six UBot modules. (b) UBot modules: Active module (black) Passive module (White) [41].

The UBot system can accommodate different locomotion modes and self-reconfiguration such as snake-type, quadruped-walker and loop-type without increasing the hardware complexity of the robots; therefore, the system can avoid the inescapable tradeoff between the generality for adaptation and the specialty for high-performance for conventional robotic systems [42].

Future plans involve adding some sensors in the modules; therefore, the system can perceive external environment information in real time and make decision about which motion modes to choose and whether to deform. Also the plans include adding some assistant modules, such as: grabbing device or other execution modules to assist the system completing more tasks in real application [42]. Finally the software for motion planning and self-reconfiguration should be designed [41].

4.19 Sambot (2010)

The Sambot is a modular robot for both functions of self-assembly and self-reconfiguration. Combining advantages of mobile and chain-based reconfigurable robot [43]. Each Sambot module is a fully autonomous mobile robot, Figure 28, while multiple Sambot can construct a robotic structure such as snake-like robot or multi-legged crawller robot through self-assembly.
The robotic structure has the ability locomotion and reconfiguration this is can be illustrated in Figure 29.

Figure 28. The Sambot

Figure 29. The configuration of robotic structure assembled by multiple Sambot.

The Sambot is compact and flexible, the overall size is 80×80×102mm and the weight is 400g. Sambot has four degree of freedoms, including the autonomous movement of the two differential driving wheels, the autonomous rotate of the active docking surface around the central axle of the main body, and the opening and closing of the docking hooks. The main body of Sambot is composed of two symmetrical halves (left and right) and the rotating mechanism of the active docking surface. At the bottom of each of the two halves is installed a driving wheel, which is driven by two motors after being decelerated by the decelerator, and on each of the driving wheel is installed a photoelectric encoder, which feeds back the speed and angle of the rotating wheel. After testing, the driving speed of Sambot ranges from 0~20cm/s, and it can also drive in the reverse direction.

Sambot’s communication is divided into two phases: wireless communication in the dispersed state and CAN bus communication in the connection communication state. When the robots get to connect, they also complete electrical connection by CAN bus. So the connected-robots can obtain the control information or other robots’ state, and make corresponding actions. The robotic structure composed of multiple Sambot can have global communication through wireless communication; this allows making autonomous decisions and realizing distributed control.

Future research work mainly covers the following aspects: the first is to continue control algorithm and experiment of the self-assembly of multiple Sambots. The Second is to study the locomotion control of the robotic structure composed of multiple Sambots.
4.20 Cross-Ball (2011)

A new proposed homogenous lattice-based modular self-reconfigurable robot [44]. The major features of Cross-Ball include:

1. Several flexible reconfiguration capabilities, such as rotating, parallel and diagonal movements so that various 3D configurations can be built up.
2. A flexible and robust hardware platform and dedicated a motion controller works in a decentralized manner for MSRs using more complex self-reconfiguration algorithms and
3. The mobility of each individual module to simplify the configuration process under certain scenarios and potential applications to swarm robots.

The proposed Cross-Ball module, as shown in Figure 30 (a), is a sphere with 3-inch diameter. The module is in a ball shape to allow individual mobility and to be spatially efficient during self-reconfiguration. It consists of three main components: a rotary arm system, as shown in Figure 30 (b) and (c), and two halves of a sphere, where the arm system is connected to the two sphere halves.

![Figure 30. (a) The Cross-Ball module. The grey part is the rotary arm system with the main arm and two clasps. There are also two clasps on the sides of the module. (b) A semi-sphere of a Cross-Ball with its arm completely extended. (c) A semi-sphere of a Cross-Ball with its wheel extended [44].](image)

The attachment procedure for the Cross-Ball can be divided into stationary attachment and dynamic attachment. Stationary attachment means that modules will be connected to build up the target configuration unless new configuration is triggered. This can be achieved by 6 symmetric stationary attachments on 6 orthogonal directions. A stationary attachment part has 2 threaded ports with screws and 2 empty threaded ports, as shown in Figure 31. The screws are equipped with a vibration motor to rotate, retract and extend. The threaded ports can accept screws from other modules. For dynamic attachments, Cross-Ball is equipped with a rotary arm system, as and two independent clasps on two sides of module, by which self-reconfiguration motions can be executed. The clasps are equipped with electromagnets to easily attach to or repel from other modules because the poles of the electromagnet can be dynamically changed. When two clasps
are attached, they are restricted to move and rotate together. The dynamic attachment needs to power the electromagnets while being connected.

![Stationary attachments and the connected main arms](image)

**Figure 31.** Stationary attachments and the connected main arms [44].

Due to its unique and flexible mechanical design and the corresponding motion controller, the Cross-Ball modular robot is able to configure itself into various complex configurations, as shown in Figure 32.

![Configuration examples](image)

(a) a snake-like robot; (b) a vehicle-like robot; (c) a caterpillar-like robot; (d) a hexapod-like robot [44].

**Figure 32.** Some configuration examples that the Cross-Ball modular robots can build up.

Future work for this modular robot includes:
1. Build the prototype of real physical Cross-Ball modular robots.
2. Update and implement our high-level hierarchical morphogenetic controller on the prototype modules to form various configurations in real experiments.

### 5. Summary

The characteristics of the self-reconfigurable robots surveyed here are summarized in Tables 5.1 and 5.2. Table 5.1 summarizes the electrical characteristics and Table 5.2 summarizes the mechanical characteristics.
<table>
<thead>
<tr>
<th>Robot</th>
<th>CPU on-board</th>
<th>Power on-board</th>
<th>Sensors used for</th>
<th>Communication</th>
<th>Control Centralized (C) or Distributed (D)</th>
</tr>
</thead>
<tbody>
<tr>
<td>PolyPod</td>
<td>yes</td>
<td>no</td>
<td>force/torque</td>
<td>bus</td>
<td>C</td>
</tr>
<tr>
<td>Fracta</td>
<td>yes</td>
<td>no</td>
<td>none</td>
<td>inter-unit optical</td>
<td>D</td>
</tr>
<tr>
<td>Metamorphic</td>
<td>yes</td>
<td>no</td>
<td>n/a</td>
<td>n/a</td>
<td>C</td>
</tr>
<tr>
<td>MEL 3D-Units</td>
<td>no</td>
<td>no</td>
<td>joint Position</td>
<td>serial w. host</td>
<td>C</td>
</tr>
<tr>
<td>The Robotic Molecule</td>
<td>yes</td>
<td>no</td>
<td>Not reported</td>
<td>serial w. host</td>
<td>D</td>
</tr>
<tr>
<td>RIKEN Vertical</td>
<td>no</td>
<td>yes</td>
<td>none</td>
<td>radio w. host</td>
<td>C</td>
</tr>
<tr>
<td>I-Cubes</td>
<td>yes</td>
<td>yes</td>
<td>Joint position</td>
<td>serial w. host</td>
<td>C</td>
</tr>
<tr>
<td>Crystalline</td>
<td>yes</td>
<td>yes</td>
<td>Joint position</td>
<td>inter-unit optical</td>
<td>D</td>
</tr>
<tr>
<td>POLYBOT</td>
<td>yes</td>
<td>no</td>
<td>Joint position docking aid</td>
<td>CAN bus</td>
<td>D</td>
</tr>
<tr>
<td>CONRO</td>
<td>yes</td>
<td>yes</td>
<td>docking aid</td>
<td>Inter-unit optical</td>
<td>D</td>
</tr>
<tr>
<td>Pneumatic</td>
<td>n/a</td>
<td>n/a</td>
<td>n/a</td>
<td>n/a</td>
<td>n/a</td>
</tr>
<tr>
<td>Telecubes</td>
<td>n/a</td>
<td>n/a</td>
<td>docking</td>
<td>Inter-unit optical</td>
<td>D</td>
</tr>
<tr>
<td>M-TRAN</td>
<td>yes</td>
<td>no</td>
<td>Proximity (IR)</td>
<td>CAN Bus</td>
<td>D</td>
</tr>
<tr>
<td>ATRAN</td>
<td>yes</td>
<td>yes</td>
<td>distance, gravity, tilt, Joint position</td>
<td>IR communication</td>
<td>D</td>
</tr>
<tr>
<td>SuperBot</td>
<td>yes</td>
<td>no</td>
<td>Proximity (IR)</td>
<td>IR communication</td>
<td>D</td>
</tr>
<tr>
<td>Programmable Parts</td>
<td>yes</td>
<td>no</td>
<td>Proximity (IR)</td>
<td>IR communication</td>
<td>D</td>
</tr>
<tr>
<td>Odin</td>
<td>yes</td>
<td>no</td>
<td>none</td>
<td>Serial RS-485 Hybrid Communication</td>
<td>D</td>
</tr>
<tr>
<td>UBot</td>
<td>yes</td>
<td>no</td>
<td>none</td>
<td>Wireless Communication Serial w. host</td>
<td>D</td>
</tr>
<tr>
<td>Sambot</td>
<td>yes</td>
<td>yes</td>
<td>Proximity (IR) Accelerometer Gyroscope</td>
<td>Wireless communication CAN bus</td>
<td>D</td>
</tr>
<tr>
<td>Cross-Ball</td>
<td>n/a</td>
<td>n/a</td>
<td>Proximity (IR) Accelerometer</td>
<td>n/a</td>
<td>D</td>
</tr>
</tbody>
</table>

Table 5.1 Electrical characteristics of Modular self-reconfigurable Robotics.
<table>
<thead>
<tr>
<th>Robot</th>
<th>Class</th>
<th>Dimension</th>
<th>Actuat. DoF.</th>
<th>Connectors (Actuated)</th>
<th>Actuation</th>
<th>Attachment method</th>
</tr>
</thead>
<tbody>
<tr>
<td>PolyPod</td>
<td>Chain</td>
<td>3D</td>
<td>2</td>
<td>6(2)</td>
<td>Perpendicular, Parallel</td>
<td>Mech.</td>
</tr>
<tr>
<td>Fracta</td>
<td>lattice</td>
<td>2D</td>
<td>3</td>
<td>3(3)</td>
<td>Perpendicular</td>
<td>Electro Magnets</td>
</tr>
<tr>
<td>Metamorphic</td>
<td>lattice</td>
<td>2D</td>
<td>3</td>
<td>6(3)</td>
<td>Perpendicular, Parallel</td>
<td>Mech.</td>
</tr>
<tr>
<td>MEL 3D-Units</td>
<td>lattice</td>
<td>3D</td>
<td>6</td>
<td>6(6)</td>
<td>Rotational</td>
<td>Mech.</td>
</tr>
<tr>
<td>The Robotic Molecule</td>
<td>lattice</td>
<td>3D</td>
<td>4</td>
<td>10(10)</td>
<td>Rotational, Perpendicular</td>
<td>Mech.</td>
</tr>
<tr>
<td>I-Cubes</td>
<td>lattice</td>
<td>3D</td>
<td>3</td>
<td>2(2)</td>
<td>Rotational, Parallel</td>
<td>Mech.</td>
</tr>
<tr>
<td>Crystalline</td>
<td>lattice</td>
<td>2D</td>
<td>1</td>
<td>4(2)</td>
<td>Parallel</td>
<td>Mech.</td>
</tr>
<tr>
<td>POLYBOT</td>
<td>Chain</td>
<td>3D</td>
<td>1</td>
<td>2(2)</td>
<td>Perpendicular</td>
<td>Mech., SMA</td>
</tr>
<tr>
<td>CONRO</td>
<td>Chain</td>
<td>3D</td>
<td>2</td>
<td>4(1)</td>
<td>Perpendicular, Parallel</td>
<td>Mech., SMA</td>
</tr>
<tr>
<td>Pneumatic</td>
<td>lattice</td>
<td>3D</td>
<td>2</td>
<td>4(2)</td>
<td>Parallel, Rotational</td>
<td>Pneumatic, Mech.</td>
</tr>
<tr>
<td>Telecubes</td>
<td>lattice</td>
<td>3D</td>
<td>1</td>
<td>6(6)</td>
<td>Parallel</td>
<td>Switchable Perm. Magn.</td>
</tr>
<tr>
<td>M-TRAN</td>
<td>hybrid</td>
<td>3D</td>
<td>2</td>
<td>6(3)</td>
<td>Perpendicular, Rotational</td>
<td>Mech., SMA</td>
</tr>
<tr>
<td>ATRON</td>
<td>lattice</td>
<td>3D</td>
<td>1</td>
<td>4(4)</td>
<td>Rotational</td>
<td>Mech.</td>
</tr>
<tr>
<td>SuperBot</td>
<td>hybrid</td>
<td>3D</td>
<td>3</td>
<td>6(4)</td>
<td>Perpendicular, Parallel</td>
<td>SINGO</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Rotational</td>
<td>Switchable Perm. Magn.</td>
</tr>
<tr>
<td>Programmable Parts</td>
<td>stochastic</td>
<td>2D</td>
<td>2</td>
<td>3</td>
<td>Parallel</td>
<td>Perm. Magn.</td>
</tr>
<tr>
<td>Odin</td>
<td>lattice</td>
<td>3D</td>
<td>4 Joints 6 links (3)</td>
<td>12</td>
<td>Rotational</td>
<td>Mech.</td>
</tr>
<tr>
<td>UBot</td>
<td>lattice</td>
<td>2D</td>
<td>2</td>
<td>8(4)</td>
<td>Rotational</td>
<td>Mech.</td>
</tr>
<tr>
<td>Sambot</td>
<td>Mobile Chain</td>
<td>2D</td>
<td>4</td>
<td>4</td>
<td>Rotational</td>
<td>Mech.</td>
</tr>
</tbody>
</table>

Table 5.2 Mechanical characteristics of Modular self-reconfigurable Robotics.
6. Challenges for Modular Self-reconfigurable Robots

There are several limitations that challenge the researchers of many of these projects. These limitations include the fact that some of these robots are controlled externally, some types of configuration are difficult to reach, mechanical constraints of modules make transformations more difficult, and there is a tradeoff between efficiency and adaptability.

Future work in modular self-reconfigurable robotics involves increasing the number of modules, constructing smaller, simpler, and lighter modules, building modules that are autonomous in that they contain their own processor, power supply, and sensors, improving the module hardware, strengthening self-repair capabilities, increasing robustness, constructing more complex structures and performing more complex tasks, improving adaptation to the current environment, improvements in motion planning, finding the optimal configuration for a given task, testing different types of terrain and travel distances, experimenting with parallel reconfiguration, and implementing distributed reconfiguration algorithms.

7. Conclusions

In this report, many different systems of self-reconfigurable robots were presented. Several interesting features and abilities of these robotic systems were discussed. Common goals among the researchers of a number of these projects include constructing thousands to millions of millimeter or micrometer robotic modules, and developing robotic structures that are self-repairing, able to perform more complex tasks, and have the ability to adapt to a variety of tasks and environments. If the researchers are successful in implementing these desired features, the use of robotic structures can be extended to domains and applications that are currently unrealizable.
8. References


