Evolution of PolyBot:

A Modular Reconfigurable Robot

David G. Duff, Mark Yim and Kimon Roufas

3333 Coyote Hill Rd, Palo Alto, CA 94304

Abstract

Modular, self-reconfigurable robots show the promise of great versatility, robustness and low cost. This paper presents examples and issues in realizing those promises. PolyBot *is a modular, self-reconfigurable system that is being used to explore the hardware reality of a robot with a large number of interchangeable modules. Three generations of PolyBot have been built over the last three years which include ever increasing levels of functionality and integration.* PolyBot *has shown versatility, by demonstrating locomotion over a variety of terrain and manipulating a variety of objects.* PolyBot *is the first robot to demonstrate sequentially two topologically distinct locomotion modes by self-reconfiguration.* PolyBot *has raised issues regarding software scalability and hardware dependency and as the design evolves the issues of low cost and robustness are being addressed while exploring the potential of modular, self-reconfigurable robots.*

1. Introduction

Modular robotic systems are those systems that are composed of modules that can be disconnected and reconnected in different arrangements to form a new configuration enabling new functionalities. There have been a variety of modular reconfigurable systems as there are many aspects of robot systems that can be modular and reconfigurable. These include: manual reconfiguration [10] and automatic reconfiguration [5], homogenous and heterogeneous modules [4].

The systems addressed here are automatically reconfiguring, hardware systems that tend to be more homogenous than heterogeneous. That is that the system may have different types of modules but the ratio of the number of module types to the total number of modules is low. Systems with all of these characteristics are called *n-modular* where *n* refers to the number of module types and *n* is small typically one or two, (e.g. a system with two types of modules is called 2-modular). The general philosophy is to simplify the design and construction of components while enhancing functionality and versatility through larger numbers of modules. Thus, the low heterogeneity of the system is a design leverage point getting more functionality for a given amount of design. The analog in architecture is the building of a cathedral from many simple bricks. In nature, the analog is complex organisms like mammals, which have billions of cells, but only hundreds of cell types.

Modular *self-reconfigurable* robot systems can also reconfigure (re-arrange) their own modules. There are a growing number of modular reconfigurable robotic systems that fit the n-modular profile [5][7][8][9][11][12][14][20]. These systems claim to have many desirable properties including versatility (from many configurations), robustness (through redundancy and self-repair) and low cost (from batch fabrication). However, the practical application outside of research has yet to be seen. While the number of modules has been large in simulation, the physical implementation of these systems has rarely had more than 10 modules. Section 2 explores these desirable properties and examines some of the issues that need to be addressed before n-modular systems with large numbers of modules can be made practical. Section 3 presents the PolyBot system design, which has been designed to explore the versatility issue. An overview of the functionality is presented along with some programming strategies.

2. Three Promises of N-modular Systems

Modular reconfigurable robotic systems that are composed of many modules have three promises. They promise to be versatile, robust, and low cost [4][7][9][13]. However, there are important issues that must be addressed before these promises can be realized.

2.1 Versatility

Modular reconfigurable robots with many modules have the ability to form a large variety of shapes with large numbers of degrees of freedom (DOF). The robot may change its shape to suit different tasks. A classical example scenario was introduced in [15] and on the Internet in 1994 [17]. It shows, in a purely kinematic simulation, a robot using three different modes of locomotion depending on the terrain type; rolling type for efficiency and speed over flat terrain, earthworm type to slither through obstacles, and over large steps and finally a spider form to stride over uncertain hilly terrain. Three images from this visualization are shown in Figure 1.

a)

b)

c)

Figure 1: Polypod simulation showing reconfiguration, a) using efficient rolling track gait, b) using obstacle crossing earthworm gait, c) using stable spider gait.

One measure of the versatility of a modular system may be the number of isomorphic configurations that are capable by a given system [2]. For many systems, this number grows exponentially with the number of modules. Another measure may be the number of DOF in the system. This also grows with the number of modules though linearly in this case.

However, it is not clear that large numbers of modules will lead to increased versatility. Even if many configurations and motions are possible, systems must have methods for planning and controlling the motion to take advantage of these configurations. Computational time complexities in planning and control often grow exponentially with the number of modules. In most cases, the computational resources also grow, though linearly, with the number of modules as each module often carries some computational resource itself. For the promise of versatility to come to fruition, methods of exploiting the distributed computational resource and strategies for dealing with the exponential size of many of the spaces will need to be developed.

2.1.1 Applications

Even though the versatility gives the capability to do a large set of specific tasks it is not necessarily reasonable to use the technology for that task. It is usually the case that tools made specifically for a task are cheaper and more efficient at that specific task than the versatile tools capable of doing many different tasks. For example, an adjustable wrench can be used for tightening a variety of bolts, but a box wrench specifically designed for a particular bolt will work more reliably (with less chance of stripping the head) and cost less.

Applications in which the n-modular systems excel are those in which versatility is critical. Typically, these are situations in which a) some information about the environment is not known a priori or b) situations in which the robot needs to perform multiple tasks for which having a number task specific mechanisms would exceed operational requirements. When the environment is not well understood a system cannot be designed specifically for a task, since the task that is needed is not known. Examples of such applications include planetary exploration, undersea mining, search and rescue and other tasks in unstructured unknown environments. In situations in which the robot needs to perform multiple tasks it may be more appropriate to have a general purpose system which can be reconfigured to each task rather than several task specific systems. The adjustable wrench is again a good example. When there is a large variety of bolts to tighten and the performance of the adjustable wrench is acceptable, an adjustable wrench can be less expensive and lighter weight than an entire set of box wrenches.

Finally, robotic systems, like other tools should be evaluated in the context of life cycle cost of a total system. General purposes tools are frequently justifiable when instantiations of the same tool can be used for multiple applications since training, repair and replacement inventory and development time are all reduced

2.2 Reliability

Another result of being modular and self-reconfigurable is the ability of the system to repair itself [7]. When a system has many identical modules and one fails, any module can replace it. As the number of modules increases, the redundancy also increases. Having redundancy does not necessarily increase the robustness of the system. More modules mean that there are more modules that can fail. If a system has millions of modules, it is likely that many of them will not be working properly.

To employ compensation requires the understanding of the failure modes of the modules and the construction of algorithms, configurations and designs tolerant to failure of some percentage of the modules.

There are two basic strategies to increase robustness to failing modules. The first is to use the redundancy of a system and global feedback to compensate for local errors of individual modules. The classical feedback control view would be that the failed module inserts some disturbance into the system and the global control of the system compensates for the introduced error. The second strategy is sometimes called self-repair [4][7]. In some instances, it may be appropriate to eject a failed module (detach it) from the system and replace it with a working module from a non-critical position. If a module fails in such a way that the ability to detach is also lost, the working modules that are attached directly to the failed module may detach and carry the failed module away.

2.3 Low Cost

Low cost is the hardest promise for modular robotics to deliver. The cost savings are indirect and may require computation over the life cycle of several systems to support. In the context of a single task it is less expensive to develop a single purpose machine than to develop and build a multi-purpose machine. But this argument could as easily be made against a drill press or a conventional industrial robot arm. Each of these devices would cost more than a task specific machine if one didn't get the benefit of all the previous development and economies of scale that had already preceded them. One of the general tenets of the modular reconfigurable approach is that versatility comes from the programming of the devices. Hence, rather than developing unique hardware and then programming it for a given robotic task, the problem is instead reduced to (re)programming the existing versatile hardware. The broad utility of this method will require the development of programming tools to facilitate and simplify programming.

As the flexible automation industry discovered in the 1980's, the cost of programming (and reprogramming) systems is often more than the cost of the hardware, thus reducing the value of the flexible nature of the hardware. The extreme versatility of n-modular systems requires a new paradigm in programming.

3. PolyBot

PolyBot is a modular reconfigurable robot system composed of two types of modules, one called a segment and one called a node. The segment module has 1 DOF and 2 connection ports. The node module is rigid with no internal DOF and 6 connection ports. So far, the systems have concentrated on addressing the versatility issue. Future generations will address the promises of robustness and low cost.

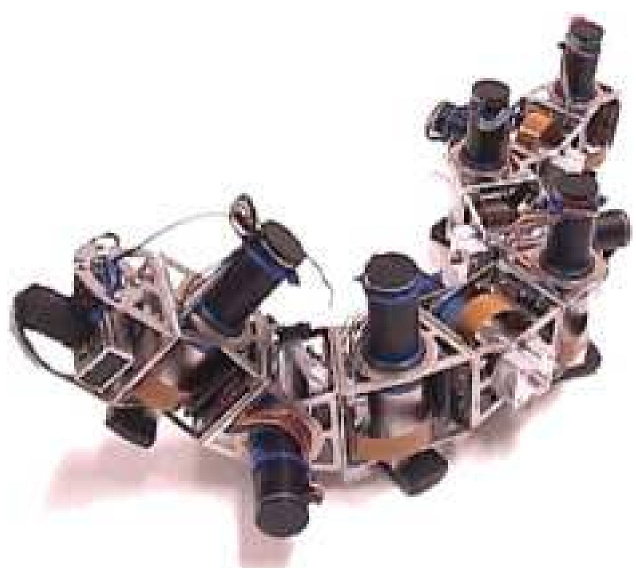

Figure 2: Nine G2 modules attached together in a snake configuration. A micro video camera with transmitter and 9V battery is attached at the front.

3.1 PolyBot design

The design philosophy behind PolyBot is that each module is very simple and that by itself cannot do very much. In combination with many others a more complex system can be built to achieve more complex tasks. Another design goal for PolyBot is that each module should fit within a cube 5cm on a side.

Three PolyBot systems have been built and experimented with. The first is called generation 1 (G1) which is a simple quickly made prototype with hobby RC servos.

The structure was built using laser-cut plastic parts. Up to 32 modules were bolted together and controlled via gait control tables with off board computing. Generation 2 (G2) functionality adds self-reconfiguration capability, additional strength and on-board computing. G3 is currently in development. It is much more compact than G2 and adds a brake/ratchet to the main actuation.

G2 is pictured in Figure 2. The segment module can be divided into three subsystems: 1) connection plate (shown in Figure 3), 2) sensing, computation and communication, and 3) structure and actuation,.

3.1.1 Connection Plate

Each segment has two connection plates. The connection plate serves two purposes. One is to attach two modules physically together. The other is to attach two modules electrically together as both power and communications are passed from module to module.

PolyBot allows two connection plates to mate in 90 degree increments allowing two modules to act together in-plane or out-of-plane. This multi-way attachment requires the electrical connectors to be both hermaphroditic as well as 4 times redundant.

These connectors were custom made as no commercial hermaphroditic connectors could be found with large enough current capacity and high enough density (1mm pitch). The connection plate consists of 4 grooved pins along with 4 chamfered holes as shown in Figure 3. An SMA actuator rotates a latching plate that catches the grooves in the four pins from a mating connection plate.

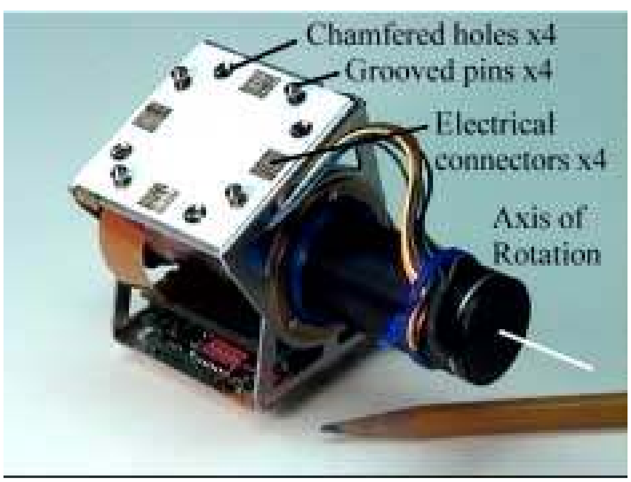

Figure 3: A G1 module showing the connection plate with 4 pins, 4 mating chamfered holes and 4 hermaphroditic electrical connector sets.

Each connection plate has 2 photo-diodes and 4 LED's that are sequenced to allow the determination of the relative 6 DOF position and orientation of a mating plate. This facilitates closed loop docking of two modules and their connection plates.

3.1.2 Sensing, Computation and Communication

Each module contains a Motorola PowerPC 555 embedded processor with 1 megabyte of external RAM. This is a relatively powerful processor to have on every module and its full processing power has not yet been utilized. The final goal of full autonomy may require the use of these processors and memory.

In G2, the sensing is limited to hall-effect sensors built into the brushless DC motors serving both for commutation as well as joint position with a resolution of 0.45 degrees. In G3, sensing will include the BLDC hall effect sensors as well as a joint angle potentiometer, tactile whiskers, tension sensors on the interface pins and accelerometers for orientation and potentially bump.

Each module communicates over a semi-global bus using the (controller area network) CANbus standard. Two CANbuses on each module allows the chaining of multiple module groups to communicate without running into bus address space limitations.

3.1.3 Structure and Actuation

The segment structure consists of two frame elements which rotate relative to one another and carry the connection plate components, the actuator and the electronics. The can be rotated up to +90 or – 90 degrees. A brushless DC motor with gear reduction sits in the middle of the segment on the axis of rotation and actuates this single DOF.

Hobby servos typically used for radio controlled cars, boats and plane have been quite successfully used in all of the G1 versions. The standard size servos used deliver maximum torques of 0.7Nm with torque densities up to 11Nm/kg. While these hobby servos come in a variety of sizes and are easy to interface with both electrically and mechanically they are somewhat underpowered and fragile for this application (dozens of them have been broken over the last three years. More torque and robustness were desired for G2. An off the shelf MicroMo gear motor was selected which could deliver 5.6Nm of torque. This gear motor has a torque density of 19Nm/kg, and was satisfactory in many respects but weighs 300g and is about 110mm long. It was desired that the G3 modules conform to a 50mm x 50mm x 50mm volume limit. No standard BLDC motor with multi-stage planetary

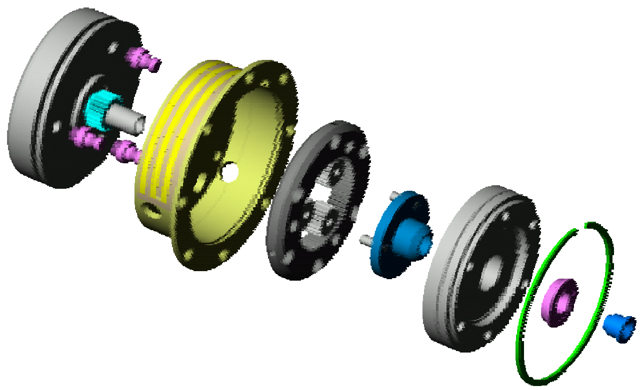

Figure 4: The PolyBot G3 drive train showing motor, planetary first stage and harmonic second stage.

gear set could satisfy the volume and form constraints of G3. A custom drive train (Figure 4) using a modified Maxon pancake motor, a custom planetary first stage and a Harmonic Drive Systems Inc. harmonic drive second stage was developed for G3. This drive can deliver 1.5Nm, weighs 72g and has a torque density of 21Nm/kg.

3.1.4 Node

The node is a rigid cube made of 6 connection plates (one for each face). It serves two purposes; one is to allow for non-serial chains/parallel structures, the other is to house higher power computation and power supplies. Portable power is very difficult to incorporate into modular systems, so PolyBot currently runs tethered to a power supply.

3.2 Locomotion and Manipulation Versatility

The PolyBot systems have demonstrated that n-modular systems can be very versatile by showing multiple modes of locomotion with a variety of characteristics. In addition, they have demonstrated some manipulation as well.

Some of the gaits that have been implemented resemble: a rolling track as in Figure 5, earthworm locomotion, turning and straight sinusoid snake-like locomotion as in Figure 6, three-legged caterpillar-like locomotion, a 3 x 4 array of cilia-like locomotion/manipulation, a 6 legged locomotion (using a tripod gait), a 3-segment slinky-like tumbling locomotion, and a 4 legged lizard-like locomotion.

Videos of most of these gaits are available for viewing on the Internet [16] and the video proceedings [17]. The earthworm, caterpillar, and 3-segment slinky were presented in previous publications [14].

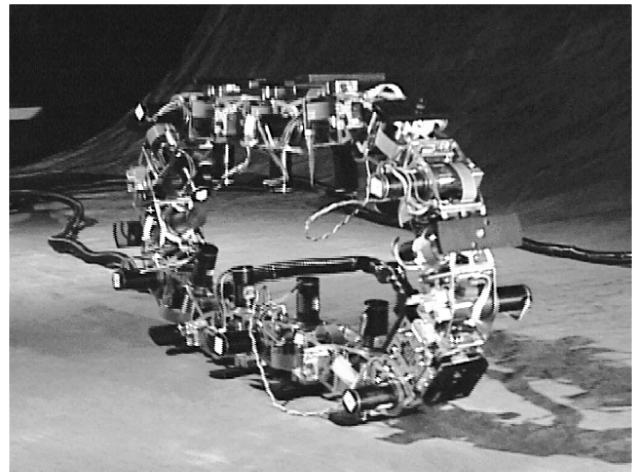

Figure 5: A loop of 23 G2 modules using a rolling track locomotion.

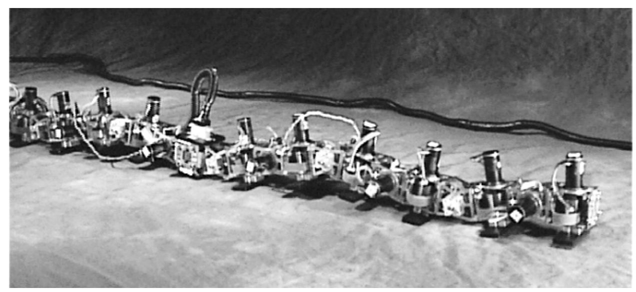

Figure 6: A snake-like sinusoid gait. The travelling wave causes forward locomotion.

In addition to the physically implemented gaits, several further gaits have been simulated: a 4 armed cartwheel-locomotion, exotic gait: carrying an object while rolling, a rolling loop with many feet on the outside rolling/walking, slinky locomotion moving on an x-y grid. Videos of these simulations may also be viewed on the Internet [16].

Since locomotion is essentially a dual of manipulation, many of the legged gaits were demonstrated to show manipulation of objects. In addition, open loop multi-arm manipulation was demonstrated as illustrated in Figure 8.

The sinusoid snake-like locomotion was demonstrated to work over a variety of obstacles including crawling in 4” diameter aluminum ducting pipes, up ramps (up to 30 degrees), over chicken wire, climbing 1.75” steps, over loose debris and wooden pallets.

In crossing obstacles with a single chain of modules like the sinusoid locomotion, two properties were determined to be essential. One is characteristic torque, a unitless quantity indicating the number of modules that can be raised to a cantilevered condition. In order to cross large

Figure 8: Four arms with three G1 modules each. A small ball is balanced on the end of one arm and passed to the top of an adjacent arm.

obstacles, like climbing stairs, the actuators need to supply large torques. For stairs, torque enough to lift about 0.3 meters worth of modules would be useful. The other property is compliance, compliance within the modules is useful for the system to conform to the terrain and gain maximal foot contact. For highly geared systems these two properties often conflict. The G1 modules with a characteristic torque of less than 5 do not have enough torque to demonstrate some gaits, but with their proportionally controlled, back drivable servos naturally conform to terrain and duct work. The G2 modules have a characteristic torque of 8 and PID control giving them little compliance. G3 will have a characteristic torque closer to 6. The additional sensors on G3 should facilitate some form of active compliance to terrain while an actuated ratchet mechanism will provide large static torque on demand.

3.3 Programming strategies

Programming the motion of n-modular systems with large numbers of modules can be difficult. Planning the self-collision-free motions can be difficult as the size of this space is exponential with the number of modules, n (proportional to the number of DOF) [6]. The inverse kinematics of serial chains with large n is also non-trivial as is the forward kinematics for parallel chains [3]. Adding the additional constraints of torque limits, joint limits and stability under gravity, the problem becomes impractical to solve optimally for the general case and even non-optimally in real-time.

Precomputed gait control tables have been an effective way to control large numbers of modules [15]. In fact, gait control tables controlled all of the implementations listed in Section 3.2. The details of the gait control table have been published previously [14]. In one demonstration, PolyBot G2 was tested over an obstacle course while under semi-teleoperated control, one module contains a set of gait control tables which were downloaded dynamically to the modules to perform such actions as turning, reversing direction, altering the speed and amplitude of the sinusoid gait and changing from loop gait to snake gait.

This method can be extended for general reconfiguration. Decomposing a structure into well known “sub-structures” which have precomputed motions for reconfiguration is one approach [1].

For many applications, a fixed set of configurations is sufficient. In this case, reconfigurations can be pre-planned off-line between every member of the set and stored in a table. In fact, configurations in the fixed set may be chosen specifically for ease of reconfiguration. In the reconfiguration example of Figure 1, the reconfiguration was hand-coded, though it only required seven attach and detach actions total. However, to fully exploit the versatility of the system, both for self-repair and task adaptation, generic reconfigurability will be required.

4. Conclusions and Future Work

Several issues need to be resolved before the three promises of modular reconfigurable systems; (versatility, robustness and low cost) can be realized. As the number of modules scales up, it is not clear that these properties apply. PolyBot is being constructed to explore these issues. The first two generations G1 and G2 address the versatility promise. Currently the maximum number of modules utilized in one connected PolyBot system is 32. PolyBot G1 and G2 generations have shown versatility in a variety of locomotion and manipulation tasks. G2 demonstrated reconfiguration including docking.

The next generation, G3, will need to address the robustness and self-repair issues, as there will be up to 200 modules (almost an order of magnitude more than any other implementation). In 2002 the project will demonstrate these 200 modules under teleoperated control. The goal for PolyBot G3 is to show 200 modules using robust autonomous locomotion, manipulation, and reconfiguration.

Acknowledgements:

This work is funded in part by the Defense Advanced Research Project Agency (DARPA) contract # MDA972-98-C-0009. The authors would like to thank Ying Zhang for programming the high level control for demonstrations on the G2 modules, Craig Eldershaw, Sam Homans, Arancha Casal, John Suh, Kevin Wooley, and An Thai Nguyen for help in final demonstration of the G2 modules and Lea Kissner for help in demonstrating the G1 modules. Thanks to Laura Valdivieso for helping to edit this paper.

References

[1] A. Casal, M. Yim, "Self-Reconfiguration Planning for a Class of Modular Robots," *SPIE Intl. Symposium on Intelligent Sys. and Advanced Manufacturing, Proceeding Vol. 3839*, pp. 246-257, Sept. 1999.

[2] I. Chen, J. Burdick, "Enumerating Non-Isomorphic Assembly Configurations of Modular Robotics Systems", *IEEE/RSJ Int. Workshop on Intelligent Robots and Systems* (IROS), pp. 1985-1992, 1993.

[3] J.J. Craig, *Introduction to Robotics*, Addison-Wesley, Reading, MA, 1989.

[4] T.Fukuda,S.Nakagawa,"Dynamically Reconfigurable Robotic System," *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pp. 1581-1586, 1988.

[5] K.Kotay, D.Rus, M.Vona, C.McGray, "The Self-reconfiguring Robotic Molecule," *Proc. of the IEEE International Conf. on Robotics and Automation*, pp424-431, May 1998.

[6] J.-C. Latombe, *Robot Motion Planning*, Dordrecht, Netherland, Kluwer, 1991.

[7] S.Murata, H.Kurokawa, S.Kokaji, "Self-Assembling Machine," *Proc. of the IEEE International Conf. on Robotics and Automation*, pp441-448, May 1994.

[8] S. Murata, H. Kurokawa, E. Yoshida, K. Tomita, S. Kokaji, "A 3D Self-Reconfigurable Structure," *Proc. of the IEEE International Conf. on Robotics and Automation*, pp432-439, May 1998.

[9] A. Pamecha, C. Chiang, D. Stein, G.S. Chirikjian, "Design and Implementation of Metamorphic Robots," *Proc. of the 1996 ASME Design Engineering Technical Conf. and Computers in Engineering Conf.*, Irvine, California, August 1996.

[10] C.J.J. Paredis, H.B.Brown, P.K. Khosla, "A Rapidly Deployable Manipulator System," *Robotics and Autonomous Systems* Vol. 21, pp. 289-304, 1997.

[11] D. Rus, M. Vona, "Self-reconfiguration Planning with Compressible Unit Modules," *Proc. of the IEEE International Conf. on Robotics and Automation*, pp2513-2520, May 1999.

[12] P. Will, A. Castano, W-M Shen, "Robot modularity for self-reconfiguration," *SPIE Intl. Symposium on Intelligent Sys. and Advanced Manufacturing, Proceeding Vol. 3839*, pp. 236-245, Sept. 1999.

[13] M. Yim, "A Reconfigurable Modular Robot with Many Modes of Locomotion," *Proc. of JSME International Conf. on Advanced Mechatronics*, pp.283-288, 1993

[14] M. Yim, "New Locomotion Gaits," *Proc. of the IEEE International Conf. on Robotics and Automation*, pp. 2508-2514, May 1994.

[15] M. Yim, *Locomotion with a Unit Modular Reconfigurable Robot*, Stanford University Mechanical Engineering Dept. thesis, 1994.

[16] http://www.parc.xerox.com/modrobots , online since 1997

[17] M.Yim, D.Duff, K.Roufas, L.Kissner, "PolyBot: demonstrations of a modular reconfigurable robot," Video Proc. of the IEEE Intl. Conf. on Robotics and Automation, 2000.

[18] http://robotics.stanford.edu/users/mark/multimode.html, online since 1994

[19] M.Yim, D.Duff, K.Roufas, "Modular Reconfigurable Robots, An Approach To Urban Search and Rescue," Proc. of 1st Intl. Wkshop on Human-friendly Welfare Robotic Systems (HWRS2000) Taejon, Korea, pp. 69-76, Jan. 2000

[20] Y. Zhang, M. Yim, J. Lamping, E. Mao, "Distributed Control for 3D Shape Metamorphosis," submitted *Autonomous Robots, special issue on self-reconfigurable robots*, 1999.