Modular Reconfigurable Robots in Space Applications

Mark Yim, Kimon Roufas, David Duff, Ying Zhang, Craig Eldershaw, Sam Homans

Palo Alto Research Center (PARC)
3333 Coyote Hill Rd
Palo Alto CA, 94304, USA
yim@parc.com

Keywords: modular, self-reconfigurable, robot, space, snake robot, locomotion

Abstract

Robots used for tasks in space have strict requirements. Modular reconfigurable robots have a variety of attributes that are well suited to for these conditions, including: the ability to serve as many different tools at once (saving weight), packing into compressed forms (saving space) and having high levels of redundancy (increasing robustness). In addition, self-reconfigurable systems can self-repair and adapt to changing or unanticipated conditions. This paper will introduce such a self-reconfigurable modular robot: PolyBot. PolyBot has significant potential in the space manipulation and surface mobility class of applications for space.

1 Introduction

Exploration of the planets, moons and other near bodies in space is a clear goal for NASA and the international space science community. A robotic approach to exploring these bodies has the benefit of being able to achieve many of the things a human could do but at lower cost without endangering human life. To be effective, such robotic systems must be versatile and robust with cost reduction becoming increasingly important. Modular reconfigurable (MR) robots have the potential to deliver all three of these characteristics[10]. MR robots are those systems that are made up of modules that can be rearranged. We are particularly interested in those systems that have many modules but only a few distinct types. If the systems are capable of actually reconfiguring themselves, the benefits are easier to realize. Self-reconfigurable modular reconfigurable systems include [1-6].

1.1 Space requirements

There are a variety of tasks that articulated robots can do in space including: space manipulation (servicing equipment in space) surface mobility (planetary exploration) and robotic colonies (outposts that are either self-sustaining, or preparatory for human colonies)[7]. In addition, articulated robots may perform scientific experiments that include sample and return of planetary atmosphere or terrain, manipulating the environment (moving rocks, drilling, etc) testing the composition of the atmosphere and other tests using arbitrary scientific equipment.

There are three characteristics of equipment that are advantageous to many space missions. 1) **Compactness and Lightness**: the cost of sending equipment into space is directly correlated to its size and weight. 2) **Robustness**: missions often have only one attempt to succeed (usually at great cost). 3) **Versatility and adaptability**: in exploration where the environments are inherently unknown, adaptability increases the chance of success.

MR systems, made up of repeated, regularly shaped modules, can be more easily packed into a given space. Since the systems can be arranged into different forms, the same robotic system can be used to perform a wide variety of tasks. Rather than sending many specialized tools, one MR system may suffice for many [8].

MR systems have large redundancy and so may be more robust. Modules can be used as replacements for failing modules. As the number of modules increases, the redundancy increases, but the probability of one module failing also increases. The system must have control strategies that exploit a graceful degradation of the system, rather than catastrophic failure from individual component failure.

MR systems have been shown to be versatile by the large variety and number of basic locomotion modes [9,10] and manipulation abilities[11]. However, the automatic adaptability of these systems is still an open problem. This requires the ability to self-reconfigure as well as to recognize that a situation warrants a new configuration and what new configuration would suffice. This might be due to changing environmental conditions or for self-repair.

1.2 Surface Mobility

Because of the high cost and one time nature of planetary exploration missions, high surface mobility is extremely important. To have a rover get hung up on an obstacle, tipped over or trapped would be considered a mission ending disaster. Future missions will require high mobility on rough terrain [20, 21].

Previous researchers have worked on improving mobile robot stability[22, 23]. The rocker-bogie mechanism is well documented and proven in a Mars rover [24] used to increase stability of the vehicle over rough terrain. Other vehicle designs include using large inflatable wheels [25] and legged vehicles.

The rest of this paper describes PolyBot, an MR system built at PARC. The next section describes PolyBot hardware, the third section, capabilities of PolyBot for application in space, and the forth section the software architecture and methods used to implement locomotion.

2 PolyBot Hardware

MR systems typically have a standardized method of attaching so that any two modules may be attached together. We call the point of connection a connection port. Connection ports serve three purposes: 1) physical connection, and 2) energy transfer and 3) communications. Connection ports may be sexed – two types of ports, one male and one female, like a typical power cord and household electrical outlet; or they may be hermaphroditic, containing both male and female components so all ports are identical. Other possible components within a module consist of actuation, sensing and/or some computational ability.

PolyBot is a modular reconfigurable robot system that uses hermaphroditic connection ports. The system is composed of two types of modules about 5 cm on a side, one called a *segment* and one called a *node*. Most of the functionality is found in the segment module; it has one degree of freedom (DOF) and two connection ports, a DC motor and a computer. Different versions also have varying amounts of sensing on board. Since segments have only two connection ports, they can only be attached end-to-end and thus form single chains. The node module is rigid with no internal DOF, six connection ports and a computer. Its primary purpose is to allow near arbitrary topologies (more than single chains). With enough segments and nodes, it is easy to approximate arbitrary structures. So far, experiments with these systems have concentrated on addressing the versatility issue. Future generations will address scalability and the promises of robustness and low cost.

There are two generations of PolyBot implemented and a third one being constructed at the time of this writing. G1 refers to Generation 1, G2 and G3 to Generations 2 and 3 respectively. The primary differentiator between these is that G1 is manually configurable while G2 and G3 have the ability to automatically reconfigure.

2.1 Generation One (G1)

The G1 module structure is laser cut plastic and is essentially cube shaped. It has one DOF involving rotating two opposing plates of the cube using commercial off the shelf hobby servos actuating through a +/- 90 degree range. 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 to, both electrically and mechanically, they are somewhat underpowered and fragile for this application (dozens of them have been broken over the last three years.)

There are four versions of G1. The first three versions (G1v1, G1v2, and G1v3 roughly shown in Figure 1) are quick prototypes with modules bolted together. The bolt holes are arranged in a square so that two modules maybe attached in one of four alignments at 90 degrees intervals. Control in the form of a 50Hz PWM signal to the servos is generated off board and brought to the modules though an external wiring harness. The control can be generated by third party servo controller boards which receive RS232 serial commands from a PC, or, as with G1v1, by a custom Motorola 68HC11 controller board. Power is also supplied to the modules in the form of 6V, approximately 0.5A per module. These versions are approximately 7x7x7 cm. Gary Haith at NASA Ames used G1v2 as the basis for their Snakebot experiments, a snake-like robot for planetary exploration[12]. Other researchers will be able to use the G1v2 design as it is available for research use by the research community.

In 1998 the G1 modules were used to show what is believed to be the first ever instance of a robot self-reconfiguring using two topologically different gaits. Specifically, this was reconfiguring from a rolling loop into a snake form. While this transformation was relatively easy since it only needed to detach (break) at one point, more complex cases have been performed and are discussed later.

The fourth version, G1v4, carries its own power (NiMH batteries) and computational resources, and is approximately 5x5x4 cm. Figure 2 shows G1v4. It needs no nodes as each module has four connection ports, three on one frame and one on the other. Manually pressing together two ports then twisting, locks the modules together. A short cable installed between each module establishes an RS485 communications bus connecting a PIC 16F877 (a small 8-bit microcontroller) on each module.

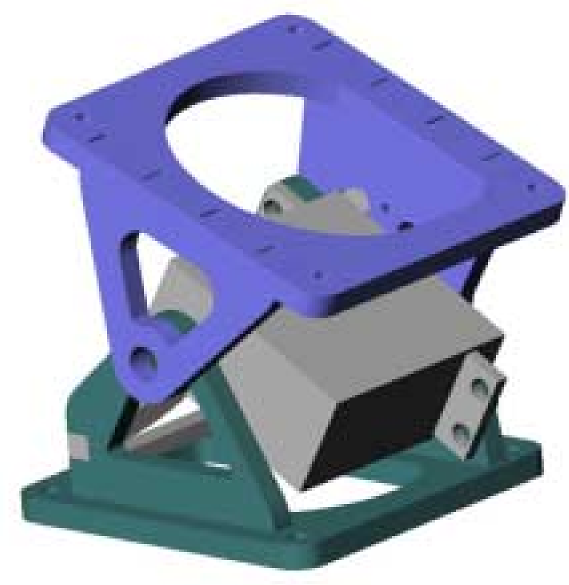

Figure 1: A CAD model of PolyBot G1v3. Modules consist of essentially three parts: two frames and a hobby servo to move them.

This version was developed as a test bed for experimenting with different gait modalities and sensors. With this goal in mind, much effort was expended to achieve ease of use, ease of manual reconfigurability, ease of programming and maintainability. Many of the gaits developed and presented in this paper were developed with the G1v4 modules. The goal is to ultimately have these gaits chosen autonomously to match environmental requirements by the next generation self-reconfigurable module, since G1v4 is not self-reconfigurable,

Figure 2: A G1v4 module with a communications cable attached. Two of four connection plates are visible. Two AAA batteries are mounted on the backside.

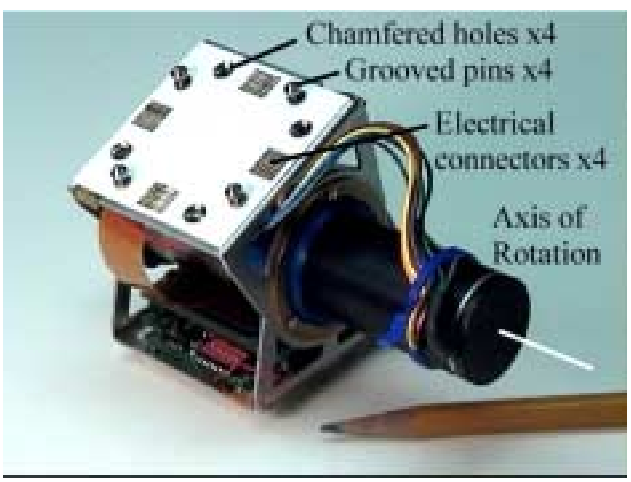

Figure 3 One G2 module showing the connection plate with 4 pins, 4 mating chamfered holes and 4 hermaphroditic electrical connector sets.

2.2 Generation Two (G2)

The second generation of PolyBot (G2) also had segments and nodes. Figure 3 shows a G2 segment. It can be divided into three subsystems: 1) structure and actuation, 2) sensing, computation and communication, and 3) connection plate.

2.2.1 Structure and Actuation

The structure is similar to G1 but is made of laser-cut stainless steel sheet. A brushless DC motor with a 134:1 gear reduction sits in the middle of the segment on the axis of rotation and can generate 5.6 Nm of peak torque. The size of the gearing causes the motor to protrude outside the cube shape. The G2 module weighs 416 grams.

2.2.2 Computation, Sensing 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 yet to be utilized. However, the ideal goal of full autonomy will likely require this power.

Hall-effect sensors built into the brushless DC motors serve both for commutation as well as joint position to a resolution of 0.45 degrees. Infrared emitters and detectors, mounted on each connection plate, serve primarily to aid docking but can also be used as proximity sensors.

Each module communicates over a global bus using the Controller Area Network (CAN) standard. Two CANbuses on each module allow multiple module groups to communicate without running into bus address space.

2.2.3 Connection Plate

Like G1, the PolyBot G2 modules allow two connection plates to mate in 90 degree increments allowing two modules to act together in-plane (motors aligned) or out-of-plane (axis of motors perpendicular). This 4-way attachment requires the electrical connectors to be both hermaphroditic as well as four times redundant.

The connectors were custom made as no commercial hermaphroditic connectors could be found with large enough current capacity and small enough size (1mm pitch). The connection plate consists of four grooved pins and four chamfered holes as shown in Figure 3. A shape memory alloy (SMA) actuator rotates a latching plate that catches the 4 grooves in the pins of the opposing connection port.

Each connection plate has two infra-red photo-diodes and four light emitting diodes that are sequenced to allow the determination of the relative 6 DOF position and orientation of a mating plate. This aids in the closed loop docking of two modules via their connection plates for reconfiguration[13].

2.2.4 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 possibly local power. G2 has high power requirements and was designed to run tethered to a power supply.

2.3 Generation Three (G3)

The third generation module has been prototyped shown in Figure 4 and is in production with over 100 modules expected. It is similar in functionality to the G2 modules with the following exceptions

Figure 4: A G3 PolyBot module prototype.

Size: the bounding box of the modules has been reduced to 5cm x 5cm x 4.5cm and the weight has been correspondingly reduced to less than 200 grams.

Power: it is expected that the power consumption will be reduced to at least half. In addition, while G2 had two power buses, one for electronics (8V) and another for the drive motor (12-24V); G3 has a single 35V bus which is used to directly power the motor and through a compact DC to DC converter, the electronics.

Connection plate: Modifications to the connection plates now allow passive attachment – i.e. snap fit latches. The IR docking system has been improved to be an order of magnitude more accurate. The electrical connectors can carry more current and are more mechanically robust.

Drive: A custom drive train (Figure 5) using a modified Maxon pancake motor, a custom planetary first stage and a harmonic drive second stage was developed for G3. This drive can deliver 1.5Nm, weighs 72g giving a torque density of 21Nm/kg. Its compactness allows the entire module to be combined within a cube for a more convenient form than that of G2.

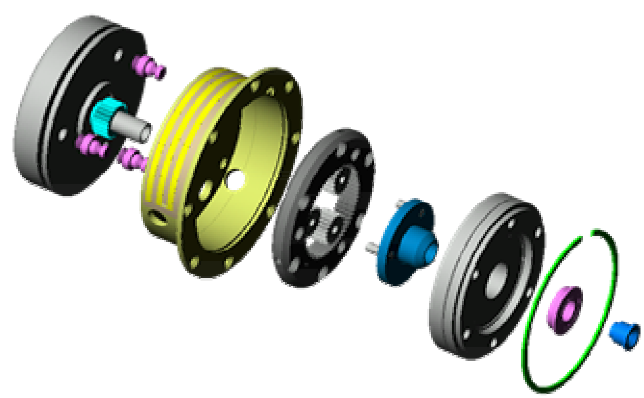

Figure 5: The PolyBot G3 drive train showing: motor, planetary first stage and harmonic second stage.

3 PolyBot Space Applications

Since PolyBot is very general in its construction, it could essentially be used in any application where an articulated robot could be used. However, we will focus on two potential applications: space manipulation and surface mobility.

3.1 Space manipulation

PolyBot is well suited to such tasks as satellite or space station inspection and maintenance since space is relatively “clean” and gravity free. The system need not be resilient to dirt, dust or mud which can be especially difficult for reconfiguring systems as it can interfere with connection mechanisms. For a single open chain snake-like robot in a planetary environment, the robot’s own weight is one of the major limitations in what it can do. Gravity-free environments greatly increase the torque-limited range of motion for these configurations.

The general versatility of the system should lead to cost savings, as mentioned earlier, as well as increased capability. Since many of the operations in space are exploratory in nature, unexpected needs may arise. An MR robot can reconfigure itself to suit the need. For example, if a longer reach on a robot is needed for a space station maintenance operation, more modules may be appended in a long chain (from a reserve set of modules). If more torque or force is needed to manipulate a satellite in space but not more reach, the robot could reconfigure itself to use parallel linkages increasing its ability to apply forces. In Figure 6, an array of modules are mounted on a plate to experiment with distributed manipulation. Using the same form of gaits used for locomotion, motion patterns for each module produce effective manipulation of large flat objects that are laid onto the platform. Experiments with different numbers of arms and different numbers of degrees of freedom per arm were performed and are presented in [11].

Free floating objects in space can often be difficult to manipulate autonomously since the objects may to rotate unexpectedly unless grasped rigidly. Since MR systems can form long chains, these chains may conform to or even envelope portions of objects as a means of grasping them.

Figure 6: using g1v3 modules in a 2 dimensional array.

Aspects of manipulation with MR systems in space that need to be addressed include inverse kinematics algorithms, applying large forces and torques where needed and for large space structures, traversing over the e structures.

Inverse Kinematics: MR systems can form hyper-redundant arm. Inverse kinematics for these arms has received some attention. Many solutions to this problem involve fitting the robot to a “back-bone” curve [17], an imaginary curve through space from the base of the robot to the desired end point.

A useful inverse kinematic solution would include avoidance of joint limits and torque limits. In addition it would be advantageous for MR systems if the algorithm finding the solution was fast and parallelizable. Several methods have been implemented on PolyBot. A brute force constrained optimization technique that incorporates both joint constraints as well as torque constraints [18] has been tested, as has a method based on dextrous workspaces formed by sub-chains [19]. The latter solution is relatively fast and maybe easily run in a distributed fashion.

Large forces and torques: In some instances, kinematically redundant mechanisms can be arranged so that the mechanical advantage of some modules can be used to generate large torques. By using closed chain configurations of PolyBot, the system can be brought into configurations in which the Jacobian of the robot's motion relative to its joint space becomes singular. In these positions the system has very large mechanical advantage over a small range. By adding a locking mechanism on each module, this range of large mechanical advantage can be increased using a ratcheting action[16]. The internal forces and the modules own weight under gravity are some of the main limitations of using the high mechanical advantage method. Here weightlessness in space is clearly advantageous.

Using the high mechanical advantage method, Polybot can be used for digging or moving rocks. Digging or uncovering layers of planetary or cometary surfaces is of interest to geologists and cosmologists. If this functionality is scaled up (either through larger modules, larger number of modules, or with longer term operation) it may also be useful in the preparation of terrain for the establishment of bases.[7]

Traversing space structures: Robot systems have been proposed to help construct or maintain large space structures [26]. To enable this they traverse the structure by docking and undocking with ports that are situated regularly over the structure. The robots would use these ports somewhat like a rock climber uses hand-holds. Since docking is one of the innate abilities for modular reconfigurable systems like PolyBot, it should be straightforward to unify the docking ports so that both the robot and the structures use the same physical and possibly electrical configurations. Since power and communications to the structure may be obtained through these ports, the robot need not carry its own power source, though local power may be useful for work on incomplete or damaged (and so unpowered) sections.

3.2 Surface Mobility

Another promising application is surface mobility for planetary exploration. The versatility of the MR systems allows it to be able to traverse a very wide range of terrain and overcome a large variety of obstacles.

3.2.1 Snake configuration

Figure 7: A snake-like sinusoid gait. The travelling wave causes forward locomotion.

One of the first configurations attempted was the "snake" or linear concatenation of modules as shown in Figure 7. This configuration is easily extended to an arbitrary number of modules without complicating the control.

Essentially, forward motion is achieved by propagating a waveform traveling down the length of the chain. Almost any waveform will result in some locomotion, though experiments suggest that the propagation of semi-circular arcs to be the most efficient [27]. Figure 7 shows G2 with a joint-space sinusoid waveform.

The snake form is particularly well suited for locomotion through highly constrained environments. In very rocky terrain such as found at the bottom of a rockslide, locomotion may be difficult. These areas may also provide particularly interesting areas for geologists. The G1 PolyBot prototypes were shown to be able to maneuver through a pile of wooden pallets and even through a 10cm diameter aluminum tube (just 1.4 times its body width).

3.2.2 Loop configuration

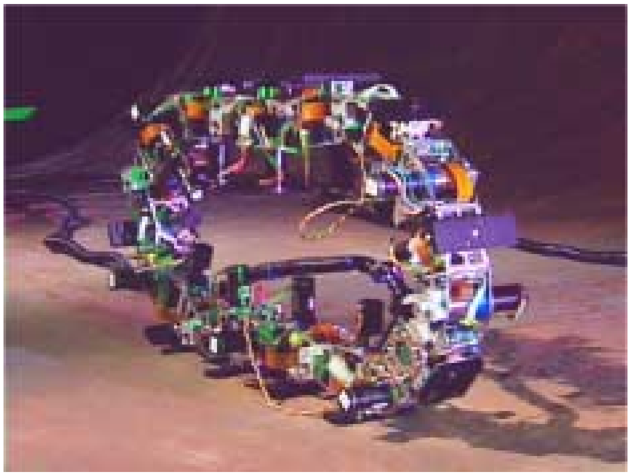

Figure 8: A loop of 23 G2 modules using rolling track locomotion.

Figure 8 shows the G2 modules in a loop configuration that rolls like a tread. Just as a wheeled style of locomotion tends to be more energy efficient than legged ones, this gait is the most power efficient gait found to date. Some initial tests with ten G1v4 modules powered by on-board AAA batteries led, to about 0.5 kilometer travel (about 2000 body lengths) on one charge. It is expected that further optimizing the motion as well as using more sophisticated battery technology would drastically increase the range.

This gait is well suited to moving on straight, flat terrain and even climbing some structures, however it is susceptible to tipping over if moving laterally across a slope.

3.2.3 Climbing over obstacles

One type of obstacle is a step. The size of the step relative to the size of the robot is one way to measure the difficulty of the obstacle. For example, normal human stairs are roughly 20 cm high, and a human maybe 200 cm tall, this translates to an obstacle one tenth of a body height.

Figure 9: A loop configuration conforming to terrain as it climbs stairs,

Moving down stairs in an uncontrolled fashion is relatively easy; the snake-like configurations achieved this with some control by having some compliance within the system to somewhat adapt to the shape of the terrain as it traverses it. Climbing up stairs is more difficult given the actuator limits of serial chains. Figure 9 shows the loop configuration climbing stairs. In this case, the robot conforms closely to the shape of the terrain (each step) as it climbs. Having a closed chain allows a parallel effort relaxing some of the actuation requirements compared to the snake-like gaits. The steps used in this demonstration are proportionately large at roughly half body height (at its tallest).

Being able to both climb over proportionally large obstacles as well as climb through small holes is unique to systems that can perform radical changes in configuration like PolyBot.

3.2.4 Climbing porous surfaces

By adding short spikes to the bottom of some modules, the G1v4 modules were able to climb near vertical porous surfaces as shown in Figure 10. The spikes grab onto porous material (like hard dirt, porous rock, or a ceiling tile as in the figure). Modules at the bottom of the robot release and form a “bump” which travels up the robot in a manner similar to the way a caterpillar climbs up a tree. This same style of climbing has also been demonstrated in climbing chain link fences but with hooks rather than spikes.

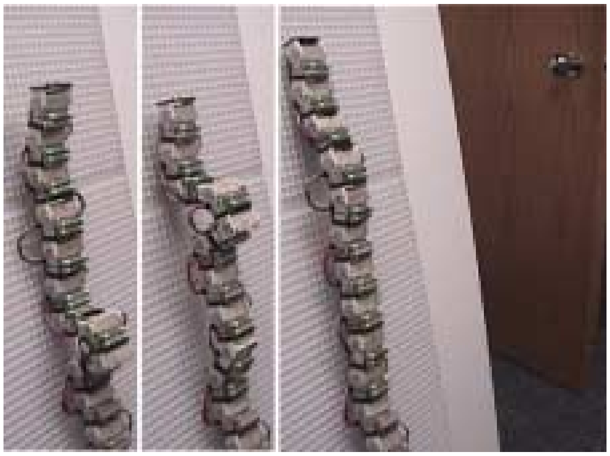

Figure 10: Caterpillar locomotion climbing a near vertical porous material (ceiling tile)

3.2.5 Other considerations

One of the issues that MR systems must overcome is dealing with environmental hardening. If the system is to reconfigure, the connection ports must be robust to dust and dirt and other contaminants. Self-wiping connectors and proper sheathing of the modules are some steps toward this.

While it is clear that the rolling track or loop gait is very much more power efficient than others, it still may not be efficient enough. Some NASA studies indicate travel up to 10 kilometers for some tasks and 100's kilometers for others[7] is needed. Portable and renewable power continues to be a major development area.

3.3 Self-Reconfiguration

Figure 11: A four-legged spider-like configuration with G2 modules.

For many space applications, having the ability to self-reconfigure to change tasks or adapt to the environment could be advantageous. PolyBot has demonstrated self-reconfiguration going from a loop form to a snake form to a spider form shown in Figure 11. Transforming from a loop to a snake is relatively easy as the robot simply detaches at one point (like falling apart). The snake to the spider transition is more difficult as first the two ends of the snake dock with a point at the center of the robot forming a figure-8. The two loops of the figure-8 then break apart to produce four legs. The automatic docking process uses IR emitters and detectors to guide the docking process. The docking process has so far only been demonstrated for planar motions, though sensing schemes have been developed to allow docking in 3-space[13].

For complicated configurations, the reconfiguration sequence can become long. The space of possible sequences of attaches and detaches typically grows exponentially with the number of modules. We have developed algorithms to automatically generate the reconfiguration sequence. These methods rely on representing the confguration as a graph and manipulating the graph with a set of specified primitive motions[28][29].

4 Gait Representation and Generation

One of the promises of modular robots like PolyBot is flexibility in locomotion. In addition to being able to adopt various configurations, each configuration of PolyBot can move using gaits. Up until now, most of the locomotion gaits for modular robots use precomputed gait control tables. This is essentially a simplified finite state machine for each module, with a prescribed sequence of behaviors for each module. It has been shown that gait control tables are an effective way to control large numbers of modules [14]. However, the problem with precomputed tables is that they are a purely open loop strategy. The resultant locomotion may not be suitable for unstructured environments such as found during planetary exploration.

The software architecture used for PolyBot G2 is based on a master/slave architecture. The master computes the gait table and downloads this table to all the segment modules on the fly. The segment modules act as slaves that simply execute the gait table once received. PolyBot G2 has been tested over obstacle courses which require such actions as turning, reversing, varying the speed and amplitude of the sinusoid gait and changing from a loop to a snake.

One problem with a single master and multiple slaves is that it does not scale well. As the number of modules increases, the single master rapidly becomes a bottleneck. Furthermore, downloading gait tables is not cost-effective in terms of communication. This can be addressed by just sending compact parameters that describe gait tables to the segments so that the actual table construction is local.

The current software architecture for G2 and the future G3 design is built on the CANbus communication protocol. We developed a new network layer protocol, Massively Distributed Control Network (MDCN) on CAN that can address up to 100,000 of nodes in CAN's extended identification format. MDCN allows arbitrary sized messages to be sent through a socket based protocol. Above this we developed one higher communication layer, the Attributes and Services Model (ASM). Its role is to coordinate and synchronize data over multiple processes and processors [15].

ASM is a general model of distributed computation, with which several software design patterns [30] have been developed. One of the design patterns is an "event-trigger" pattern, where events are a special type of attribute associated with triggers such as clock ticks, interrupts or thresholds. Services can then be event-driven, where the function executed will depend on the type of trigger. For some of PolyBot's locomotive capabilities shown in this paper, a phase automata design pattern is an effective way of representing and generating the desired gaits.

4.1 Phase Automata Pattern

Phase automata are generally event-driven discrete state machines with periodic behaviors. The phase of each module indicates that module's particular starting point in the automata in a continuous time domain. Phase automata are efficient representations of hybrid systems with both high level discrete event-driven and lower level continuous characteristics. A phase automaton extends classical event-driven service with: a state, a direction, an initialization routine, and an event handler as a service function. The event handler in general consists of two parts: an Action() function and a NextState() function. Figure 12 shows the class diagram of the phase automaton.

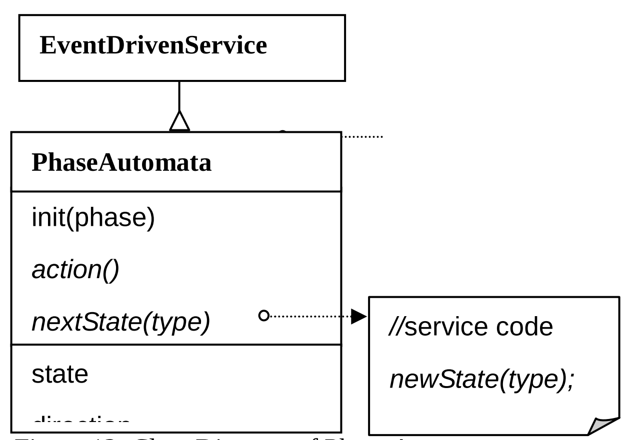

Figure 12: Class Diagram of Phase Automata

The initialization routine will be executed at the beginning of start() which in turn is inherited from EventDrivenService. The initialization routine's parameter indicates the initial delay phase for this particular automaton. This routine is also responsible for setting the initial state and action based on the phase delay. All automata have persistent state, but phase automata also have direction variables. For this particular application, these are mostly used for gait control.

Phase automata provide a more general framework for gait control than gait tables [14]; they can represent both time driven and sensor driven gaits, periodic and non-periodic gaits, and local and global gaits. In the next section, a phase automaton is used to describe a conforming loop gait (discussed previously). This uses both time driven and touch sensor driven events.

4.2 Example: Loop Gait

Loop gaits are a method of locomotion that resembles a rolling tractor tread. They can be easily implemented with phase automata. The hardware of the loop consists of a set of PolyBot modules arranged in a loop, each module having a microprocessor and one actuated degree of freedom. A basic loop is formed from sets of modules forming two semicircles attached by two other sets of modules forming straight lines (Figure 13).

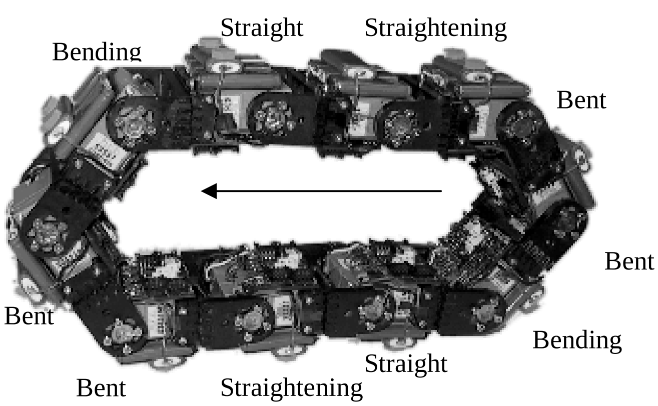

Figure 13: PolyBot Loop

As the loop rolls, each module goes twice through a four-step cycle of (1) bending to 60 degrees, (2) freezing, (3) straightening, and (4) freezing, before returning to its original position in the loop. Each module runs a two-level control; the phase automaton sets the desired angle attribute, the control mode attribute and the timer,; and a low level linear controller tracks the desired behavior.

4.2.1 Simple time-based loop

For a simple time-based loop gait the phase automata requires only four states (Figure 14):

Straight: the joint angle is straight and not moving;

Bending: the joint angle is moving from straight to bent;

Bent: the joint angle is bent and not moving;

Straightening: the joint angle is moving form bent to straight.

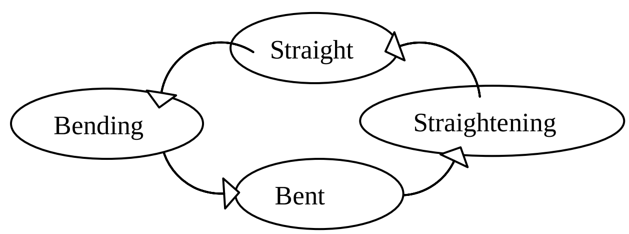

Figure 14: State Diagram of a Simple Time-based Loop

The trigger for each state is a timer: the duration for Straightening and Bending is a constant T; the duration for Straight is (n-1)T, where n is the number of modules in a straight segment between two semicircles; and the duration for Bent is (m-1)T, where m is the number of modules in a semicircle. With ten modules, as in Figure 14, there are three modules in the semicircular portion bending at 60 degrees (m=3), and two modules in the straight portions (n=2). The constant T determines how fast the loop is to travel; the period for this phase automaton is (n+m)T, or (N/2)T, where N is the number of modules in the loop.

Modules in the loop have different initial phases, i.e. they start at staggered locations in the state diagram. This is implemented with the phase delay parameter such that each module is delayed by time T relative to its neighbor. This translates to starting with a different phase (the horizontal axis) of Figure 15.

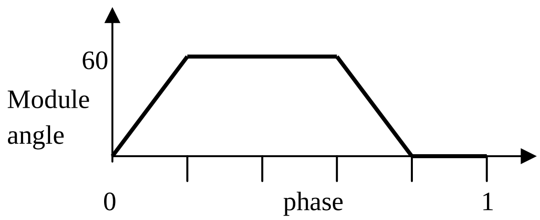

Figure 15: Angle of one cycle of a simple loop.

4.2.2 Conforming Loop

The time-based loop gait is not robust when traveling through rough terrain or with broken modules. By utilizing sensors that can detect contact with the terrain, a similar gait can be generated that conforms to the terrain. A sensor based phase automata can be used to implement this by adding conforming and relaxing states.

The loop can be partitioned into four parts: the *head* is the leading three modules of the loop; the *tail* is trailing four modules of the loop; the *conformed* section is a set of modules connecting the head to the tail (nominally the bottom part of the loop in contact with the terrain), the *relaxed* section is a set of modules that connects the tail to the head (nominally the top part of the loop).

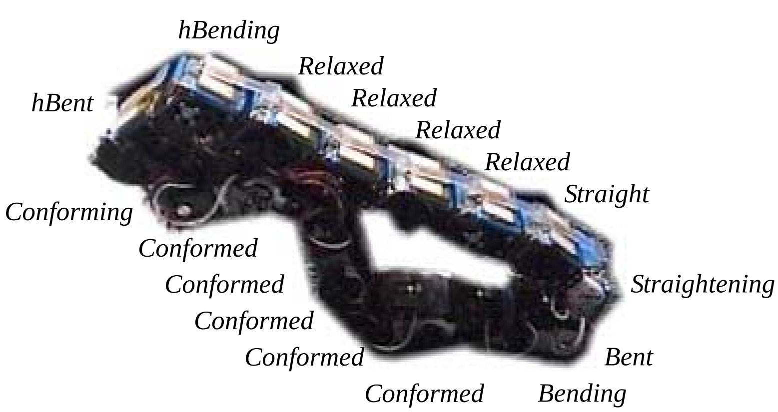

Figure 16: Conforming loop showing the state of each module.

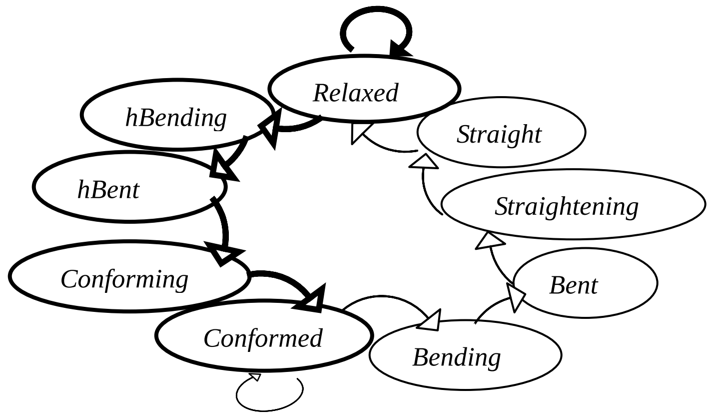

Figure 17: State Diagram of a Conforming Loop

There are nine states used for this gait as shown in Figures 16 and 17, the tail uses the same four states as the simple timed-based loop gait, the head, relaxed and conformed sections however use a sensor-based transition indicated with thick lines in Figure 17. In fact all of these transitions occur with the same event. When a module in the *Conforming* state senses contact with the environment the modules in states *Conforming*, *hBent*, *hBending*, and the last module in the *Relaxed* state transition to the *Conformed*, *Conforming*, *hBent* and *hBending* state respectively. Note that here events are triggered by events not local to a module but remotely. Notification of this event is sent through the network from the conforming module. The other states are time-driven as in the simple time-based loop gait.

The states are as follows:

Conforming: moving from bent inwards to bent outwards stopping if it makes contact with the environment;

Conformed: the joint stops;

Relaxed: no control effort, moves in a compliant fashion subject to external forces;

hBending: same as *Bending* except transitioned by touch sensor of the active conforming module;

hBent: same as *Bent* except transitioned by touch sensor;

At any given time, there is one module in each of the states except the *Relaxing* and *Conforming* states which varies since the head and tail sections trigger independently. There is a potential problem if one of the sensed triggers for the head occur too rapidly or too slowly causing the head and tail to collide (zero *relaxed* or zero *conformed* states). However in practice this rarely occurs if the modules are sensing the environment correctly.

Figure 18: Conforming Loop

Figure 18 shows a conforming loop with a total of 16 modules. Two modules in the semi-circular portion at each end, each bending 90 degrees.

Since the robot conforms to whatever terrain it makes contact with, it is inherently very stable and has the ability to climb over large obstacles. This gait has been demonstrated to climb over objects as high as 12cm, this is 1.2 times the nominal height of the loop.

5 Summary and Conclusions

PolyBot is a modular reconfigurable robot that can self-reconfigure. This paper describes several versions of the hardware that have been developed and the experiments done with them. The experiments have shown that PolyBot has a number of characteristics that make it well suited to perform important tasks related to space exploration, in particular manipulation in space and surface mobility. PolyBot can use energy efficient modes of locomotion in open environments or switch to other modes to overcome large obstacles or squeeze through tight places.

Acknowledgment

This work is funded in part by the Defense Advanced Research Project Agency (DARPA) contract # MDA972-98-C-0009.

References

[1] P. Will, A. Castano, W-M Shen, “Robot modularity for self-reconfiguration,” *SPIE Int. Simp. on Intelligent Sys. and Advanced Manufacturing, Proc. Vol. 3839*, Sept. 1999, pp. 236-245,.

[2] Pamecha, A., Chiang, C., Stein, D., Chirikjian, G., "Design and Implementation of metamorphic Robots", *ASME Design Engineering Technical Conference-Computers and Engineering*, Irvine, CA, 1996.

[3] S. Murata, H. Kurokawa, S. Kokaji, “Self-Assembling Machine,” *Proc. IEEE Int. Conf. on Robotics and Automation*, May 1994, pp441-448,.

[4] D. Rus, M. Vona, “Self-reconfiguration Planning with Compressible Unit Modules,” *Proc. of the IEEE Int. Conf. on Robotics and Automation*, May 1999, pp. 2513-2520

[5] Unsal, C., Killiccote, H., Khosla, P., "A Modular Self-Reconfigurable Bipartite Robotic System: Implementation and Motion Planning", *Autonomous Robots*, pp. 23-40, Vol. 10, No. 1, 2001

[6] Yim, M., Zhang, Y., Lamping, J., Mao, E., "Distributed Control for 3-D Metamorphosis", *Autonomous Robots*, pp. 41-56, Vol. 10, No. 1, 2001

[7] NASA, Human Exploration of Mars: The Reference Mission (Version 3.0 with June, 1998 Addendum) of the *NASA Mars Exploration Study Team, Exploration Office, Advanced Development Office, Lyndon B. Johnson Space Cente*r, Houston, TX 77058, June, 1998

[8] Farritor, S., Dubowsky, S., "On Modular Design of Field Robotic Systems", *Autonomous Robots*, pp. 57-66, Vol. 10, No. 1, 2001.

[9] P. S. Schenker, et. al. "Reconfigurable robots for all terrain exploration", in *SPIE, Sensor Fusion and Decentralized control in Robotic Systems III Vol 4196*, November 2000

[10] M. Yim, D. Duff, K. Roufas, “PolyBot: a Modular Reconfigurable Robot” *Proc. of the IEEE Int. Conf. on Robotics and Automation*, April 2000.

[11] M. Yim, J. Reich, A.A. Berlin, “Two approaches to distributed manipulation.” In *Distributed Manipulation*. H. Choset, K. Borhinger eds., Norwell, MA: Kluwer Academic Publishing; 2000.

[12] E. Smalley, “NASA gets snake robot off the Ground,” *Technology Research News* [online news magazine] http://www.trnmag.com/Stories/101100/Snake_Robot_101100.htm, October 11, 2000

[13] K. Roufas, Y. Zhang, D. Duff and M. Yim, “Six Degree of Freedom Sensing for Docking Using IR LED Emitters and Receivers,” *Int. Symp. for Experimental Robotics 2000* (ISER) Honolulu Hawaii.

[14] M. Yim, “A reconfigurable modular robot with many modes of locomotion,” in *Proc. JSME Int. Conf. on Advanced Mechatronics*, 1993 pp 283-288

[15] Y.Zhang, K.D.Roufas, M.Yim, “Software Architecture for Modular Self-Reconfigurable Robots” *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems* (IROS), Oct. 2001

[16] M.Yim, D.Duff, Y.Zhang, “Closed chain motion with large mechanical advantage,” *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems* (IROS), Oct. 2001

[17] Chirikjian, G. S. and Burdick, J. W., ``Geometric approach to hyper-redundant manipulator obstacle avoidance ", *J. of Mechanical Design - Transactions of the ASME*, Vol. 114, no. 4, 1992, pp. 580-585

[18] M. P. J. Fromherz, M. Hoeberechts, and W. B. Jackson, “Towards Constraint-based Actuation Allocation for Hyper-redundant Manipulators.” In: *CP’99 Workshop on Constraints in Control (CC'99)*, Alexandria, VA, Oct. 1999.

[19] S.K. Agrawal, L.Kissner, M.Yim, “Joint Solutions of many degrees-of-freedom systems using dextrous workspaces,” in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, 2001

[20] Hayati, S., et al., "Microrover Research for Exploration of Mars," *Proc. of the AIAA Forum on Advanced Developments in space Robotics*, University of Wisconsin, Madison, August 1-2, 1996.

[21] *A scientific rationale for mobility in planetary environments*, Space Studies Board, National Research Council, National Academy Press, 1999.

[22] Iagnemma, K., and Dubowsky, S., "Mobile robot rough-terrain control (RTC) For planetary exploration", *Proc. ASME Design Engineering Technical Conferences*, Baltimore, Md, Sept. 2000..

[23] Sreenivasan, S., and Wilcox, B, “Stability and Tractino Control of an Actively Actuated Micro-Rover,” *Journal of Robotic Systems* **11** (6):487-502, 1994

[24] Bickler, B., "A New Family of JPL Planetary Surface Vehicles," *Proc. Missions, Technologies, and Design of Planetary Mobile Vehicle*, Toulouse, France, 1992

[25] Jones, J.A. “Inflatable Rover Demonstration,” *video proc of IEEE Conf. on Robotics and Automation*, (ICRA) San Francisco CA, May 2000,

[26] Yangsheng Xu and Ben Brown, "A space station robot walker and its shared-control software", *Proc of the AIAA/NASA Conference on Intelligent Robots*, Vol. 1, pp.123-130, Houston, TX, 1994.

[27] Yim, M., Homans, S. and Roufas, K., ``Climbing with Snake-like Robots" *Proc. of the IFAC Workshop on Mobile Robot Technology*, Jejudo, Korea, May 21-22, 2001

[28] Yim, M., Goldberg, D. and Casal, A., ``Connectivity Planning for Closed-Chain Reconfiguration," *SPIE, Sensor Fusion and Decentralized control in Robotic Systems III* Volume 4196, Nov. 2000

[29] Casal A, “Reconfiguration Planning for Modular Self-Reconfigurable Robots ", PhD Thesis, Dept. of Aero/Astro. Stanford University, 2002

[30] E. Gamma, R. Helm, R. Johnson, J. Vlissides, *Design Patterns: Elements of Reusable Object-Oriented Software*, Addison-Wesley Professional Computing Series, 1995.

Mark Yim

Mark Yim is a senior member of the research staff, manager of the Smart Electro-Mechanical Systems Area at the Palo Alto Research Center (formerly Xerox PARC) and leads a project on modular, reconfigurable robot systems. He received his PhD in mechanical engineering from Stanford University in 1994 and has published in the areas of planning, distributed robotics, robots for search and rescue, optimal control, robotics in education, virtual reality and haptics. Mark has authored over 40 patents and in 1999 was chosen for the TR100, the top 100 young innovators by Technology Review Magazine.

Kimon Roufas

Kimon Roufas was born in Chicago, IL. His family later moved to Greece. There he built his first robot out of a modular hamster cage and a Commodore 64 at the age of thirteen. His family then moved to Belgium, where he graduated from an international high school in Waterloo. In 1996 he graduated from Rensselaer Polytechnic Institute with a B.S. in mechanical engineering and minors in electrical engineering and French. After receiving his M.S. in mechanical engineering from Stanford University in 1998, he began working at Xerox PARC on Mark Yim's robotics project. In 2001 he received his M.S. in electrical engineering from Stanford. He continues to work at the Palo Alto Research Center and his professional interests are in embedded networks, wireless communications, and sensors.

David Duff

David Duff has been at PARC working on Modular Robotics since the fall of 1998 where he's primarily responsible for the mechanical design of the PolyBot robots. Dave grew up in Palo Alto, CA, and has a Master's in Engineering from Stanford. Prior to joining PARC he spent nearly a decade working for Lockheed in Sunnyvale, CA on Strategic Defense Initiative projects (a.k.a. Star Wars) as a ballistic missile defense system analyst and architect and three years at WET Design in Los Angeles as a mechanical design engineer for the fountain equipment at the Bellagio Hotel (opened in 1998).

Dave lives with his wife and two young children in the coastal range Redwoods just west of Palo Alto.

Ying Zhang

Ying Zhang received her Ph.D. in Computer Science from University of British Columbia in 1994. Since then, she has been a member of research staff at Xerox Research Centers: Wilson Center for Research and Technology, Xerox Architecture Center and Palo Alto Research Center. Ying has been working on various projects within Xerox and has a wide interest in Computer Science. Her graduate work focused on modeling, specification and verification of hybrid dynamic systems, constraint-based systems, object-oriented parallel and distributed systems and control synthesis. While at Xerox, she has been working on advanced modeling environments, simulation and control of xerographic processes, Xerox DataGlyph toolkits and software architecture for Modular Robotics. Her current work is on Java-based real-time embedded control and communication architecture for massively distributed reconfigurable modular robots.

Craig Eldershaw

Craig Eldershaw is a member of the research staff at the Palo Alto Research Center (PARC). Having studied at four different universities in three different countries, Craig earned his PhD in computer science at The University of Oxford in 2001. While his background ranges from data mining to parallel computing and optimisation, the topic of his doctoral thesis was heuristic motion planning. He has published in the areas of robotics, parallel computing and numerical mathematics. His more recent focus has been in embedded computation.

Sam Homans

Sam Homans lives in Oakland with his wife and two children. Originally from the Boston area, he studied Sculpture and Drawing at Harvard. After 10 years in the wilderness making art, commercial sculpture, designing and building wooden furniture, and after a few years in the special effects business, he attended the Product Design Program at Stanford, where among other things he worked with a team at the Rapid Prototyping Lab on a micro-air vehicle called the "Mesicopter." He designed the mechanical aspects and interface for the G1V4 modules and a new modular tactile interface called Digital Clay. Digital Clay began as a Master's Project exploring more direct (hands-on) CAD interfaces entitled "Material as Interface."