Teleoperated Modular Robots for Lunar Operations

Al Globus*
CSC, NASA Ames Research Center, California, 94035, USA

Greg Hornby*
and Greg Larchev*
QSS, NASA Ames Research Center, California, 94035, USA

Matt Hancher*
and Howard Cannon†
and Jason Lohn*
NASA Ames Research Center, California, 94035, USA

Solar system exploration is currently carried out by special purpose robots exquisitely designed for the anticipated tasks. However, all contingencies for *in situ* resource utilization (ISRU), human habitat preparation, and exploration will be difficult to anticipate. Furthermore, developing the necessary special purpose mechanisms for deployment and other capabilities is difficult and error prone. For example, the Galileo high gain antenna never opened, severely restricting the quantity of data returned by the spacecraft. Also, deployment hardware is used only once.

To address these problems, we are developing teleoperated modular robots for lunar missions, including operations in transit from Earth. Teleoperation of lunar systems from Earth involves a three second speed-of-light delay, but experiment suggests that interactive operations are feasible.[1] Modular robots typically consist of many identical modules[2] that pass power and data between them and can be reconfigured for different tasks providing great flexibility, inherent redundancy and graceful degradation as modules fail. Our design features a number of different hub, link, and joint modules to simplify the individual modules, lower structure cost, and provide specialized capabilities.

Modular robots are well suited for space applications because of their extreme flexibility, inherent redundancy, high-density packing, and opportunities for mass production. Simple structural modules can be manufactured from lunar regolith *in situ* using molds or directed solar sintering.

Software to direct and control modular robots is difficult to develop. We have used genetic algorithms to evolve both the morphology and control system for walking modular robots.[3] We are currently using evolvable system technology to evolve controllers for modular robots in the ISS glove box.

Development of lunar modular robots will require software and physical simulators, including regolith simulation, to enable design and test of robot software and hardware, particularly automation software. Ready access to these simulators could provide opportunities for contest-driven development ala RoboCup (http://www.robocup.org/). Licensing of module designs could provide opportunities in the toy market and for spin-off applications.

*Evolvable Systems, Neuro Engineering Group, Computional Sciences Division
†Model Based Diagnosis and Recovery Group, Autonomy and Robotics, Computional Sciences Division

Nomenclature

ABS	Acrylonitrile butadiene styrene,a thermoplastic used in many products
EVA	Extra Vehicular Activity (space walk)
FPGA	Field Programmable Gate Array
ISS	International Space Station
LEO	Low Earth Orbit
hub	a module with more than two connectors
joint	an articulateable link, e.g., a hinge; may be powered or unpowered
link	a module with exactly two connectors
module	an individual robot component that can be connected to any other module via standardized connectors
regolith	the fine-grained, loose surface layer on the Moon
SLA	stereo-lithography apparatus

I. Introduction

Exploration of the solar system, to say nothing of colonization, will require construction, operation and repair of extensive facilities in orbit, on the lunar and martian surface, and on the asteroids. While space suited astronauts can certainly perform these functions, space is a hostile environment full of hazards. Vacuum and temperature extremes can be handled by space suites, but operations above the van Allen belt are exposed to much higher radiation levels than in LEO making extensive EVA problematic if not impossible. Thin space suits actually increase biologically harmful radiation since cosmic heavy primaries (nuclei of the heavier elements) striking the suit generate showers of secondary particles that are not absorbed and cause a great deal of damage. Astronauts significantly beyond LEO must therefor spend most of their time in heavily shielded environments.

These hazards make human space operations extremely expensive. To affordably explore the solar system we need capable, flexible robots operating autonomously, under astronaut control, and teleoperated from the ground to build and operate space facilities to meet ambitious exploration goals. We propose using the lunar environment to develop modular robotics capable of building, operating, and repairing unmanned and manned facilities to enable sustained, affordable, and highly capable solar system exploration.

The lunar environment provides an excellent test bed for planetary surface operations because of its proximity to Earth. This means that resupply and reflight is a matter of days or weeks, rather than years as with Mars, and that teleoperation faces only a three seconds speed-of-light delay. Human operators can adapt to such short delays for many tasks. For example, we have shown that untrained operators readily adapted to a three second delay when driving simulated rovers through a maze.[1] With teleoperators available to work robots out of difficult situations, automation can be gradually introduced into an operational setting, insuring that the right problems are solved.

To date, the robots deployed in space have had a fixed morphology. That is to say, the hardware configuration can not be changed after launch. Furthermore, every robot launched to date has worked alone. Although Spirit and Opportunity rovers operated concurrently on Mars, they landed far apart and could not repair or even observe each other. Each of these meter-scale, solar powered, six wheel rovers has a robot arm for positioning sensors and actuators as well as mast mounted TV cameras and other sensors. While extremely well designed and operated, the fixed morphology and complete lack of close team work has put severe limits on the capabilities of these and other space robots to date.

We propose improving this situation by developing modular robots for lunar operations. In general, modular robots consist of multiple reconfigurable modules with standardized inter-module interfaces. Most of the modular robots to date have consisted of identical modules capable of articulation and, in some cases, reconfiguration (attaching to and detaching from each other). Modules pass power and data to each other. These robots are not particularly capable, being able to reconfigure and move a bit, and are limited to the

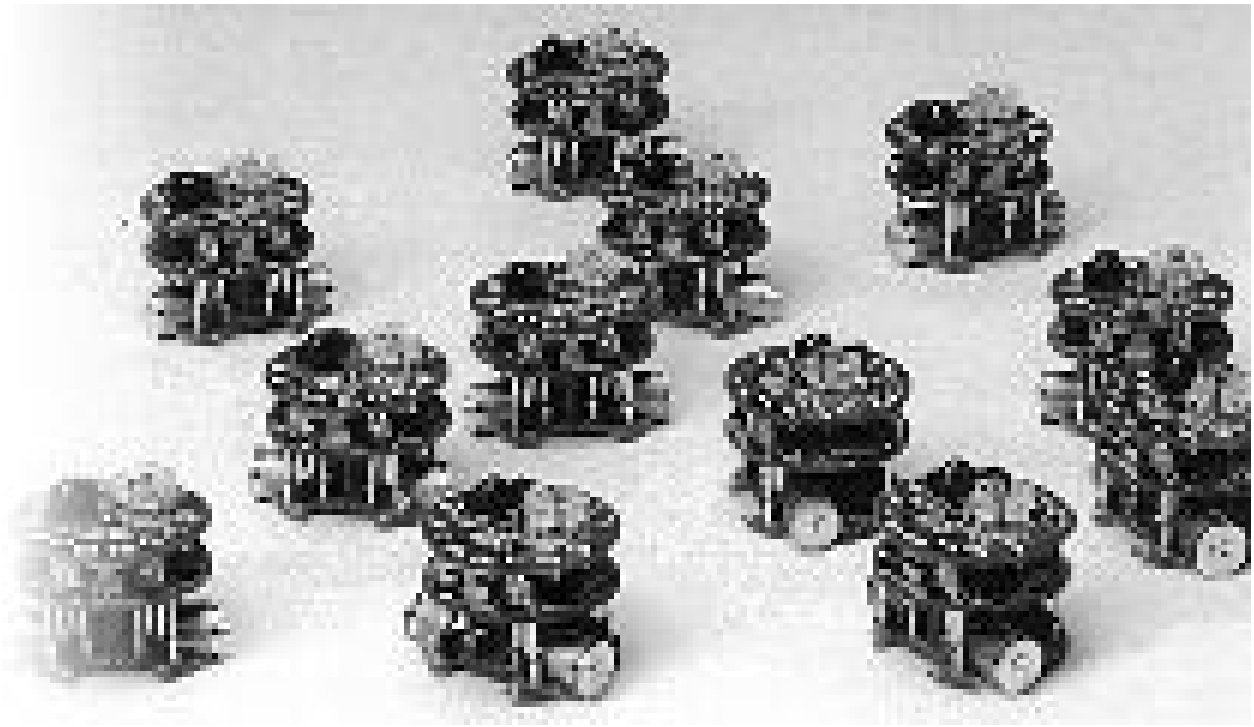

Figure 1. Fukuda and Kawauchi's micro autonomous robotic systems

laboratory. Work to date has included:

1. Fukuda and Kawauchi

   One of the first modular robotic systems was CEBOT, for cellular robotic system, developed by Fukuda and Kawauchi.[4] In this system all modules were the same but it was not able to self-reconfigure, rather reconfiguration was carried out by a manipulator arm.

2. I-Ming Chen

   Chen developed Assembly Incidence Matrices (AIM) to represent a modular robot assembly configuration. AIM was used to develop methods for enumerating assembly configurations[5] to assist in automatically generating the equations of motion of a particular modular robot configuration.[6] AIM has also been used as the representation to encode configurations for evolution with a genetic algorithm.[7] In addition, his group has developed a method for controlling the end effector of a modular robot with a joystick.[8]

3. Mark Yim

   Yim's initial modular robotics work was at Stanford University from 1992-1994. He developed the Polypod, a modular robot consisting of two module types.[9] One type is called a Segment and combines a prismatic joint with a revolute joint for two degrees of freedom. The second type is called a Node, which is a rigid cube that supplies power to Segment modules. Following this work he developed three generations of PolyBots. The first generation PolyBot was a prototype consisting of a single module type which has a one degree of freedom revolute joint. The second generation has Segment and Node modules.[10] The third generation PolyBot is essentially the same as the second generation but is a little

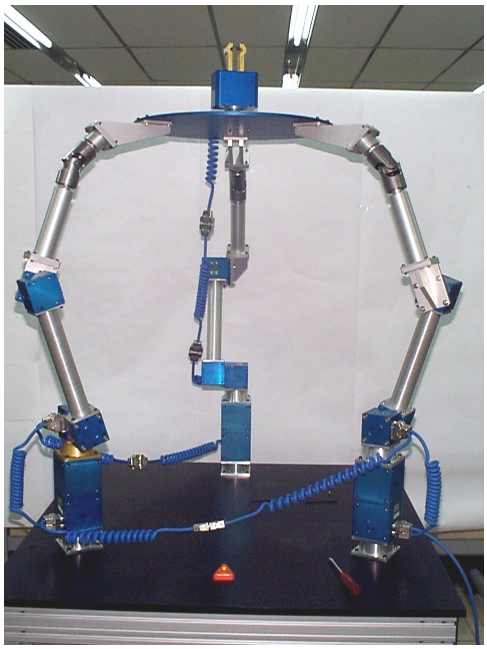

Figure 2. One of I-Ming Chen's robots

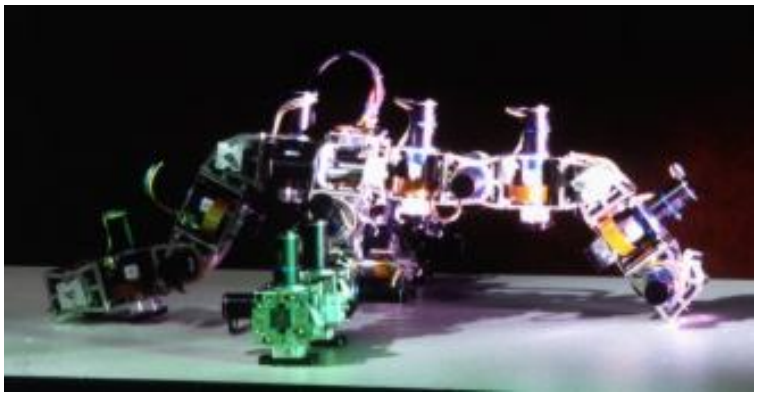

Figure 3. Mark Yim's Polybot configured for walking

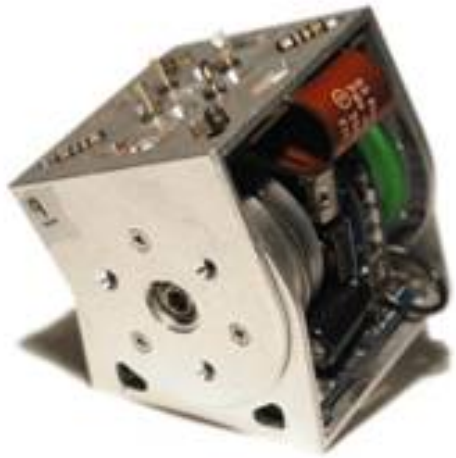

Figure 4. Mark Yim's third generation Polybot module

smaller and lighter, and has additional sensors.[11] Development of these modular robotic platforms has focused primarily on locomotion gaits and self-reconfiguration.

4. Daniella Rus

   Rus took a different route, developing self-reconfigurable modular robotic systems in which modules actuate by expanding and contracting. The first such robot, the Crystalline Robot, only actuated in two dimensions.[12] Another modular robotic system developed by Rus is the Molecule robot, in which each module consists of two atoms connected by a rigid connection.[13] Each atom has two degrees of freedom through two revolute joints. More recent work is on algorithms for generic modular robot locomotion and reconfiguration inspired by cellular automata.[14]

5. Gregory Hornby

   While most work in the field of modular robotics focuses on the design of the hardware modules and methods for reconfiguring them, Hornby has focused on methods for automatically designing configurations and controllers.[15,16] One of the challenges of configuration design is developing methods that scale to systems of thousands of modules. To address scalability, Hornby uses generative representations where parts of the encoding can be reused in the definition of system configurations. Configurations are created using an evolutionary algorithm acting on this representation to evolve modular robotic systems for locomotion. This system was able to evolve configurations of hundreds of modules.

Although modular robots are at an early stage of development, they offer a number of substantial advantages over fixed-morphology robots. Specifically, modular robots are inherently flexible, redundant, reliable and low-cost.

First, modular robots are flexible. They can be reconfigured to perform many different tasks. For example, a set of modules could be configured into scaffolding to hold a camera high above the lunar terrain to inspect initiation of mining activities, then be reconfigured into a number of walking robots to move to another location and set the camera up again to inspect habitation hardware being buried for radiation protection.

Modular robots are inherently redundant and reliable since many identical modules are launched together. As modules fail, capabilities degrade gracefully. The whole system only fails if all modules of a critical type fail. This has important implications for base resupply. Each resupply launch may contain a number of

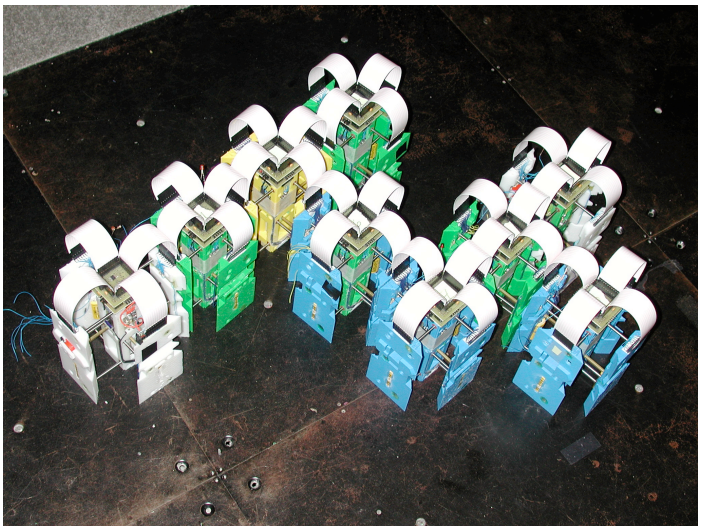

Figure 5. Daniela Rus' crystal robot

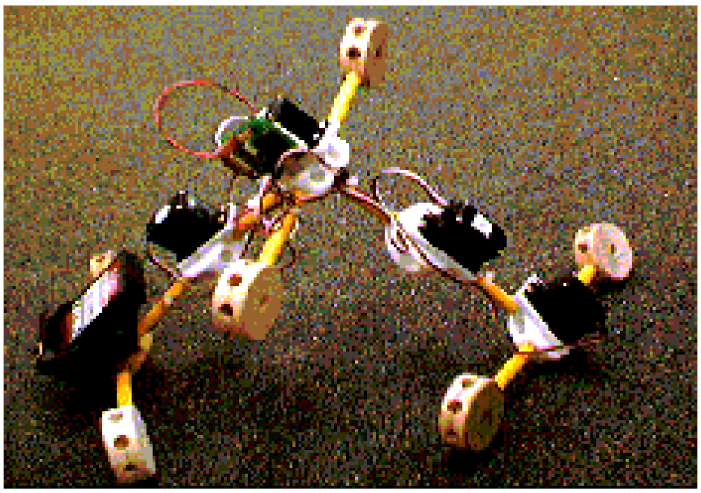

Figure 6. One of Greg Hornby's evolved walking robots implemented with Tinker Toys

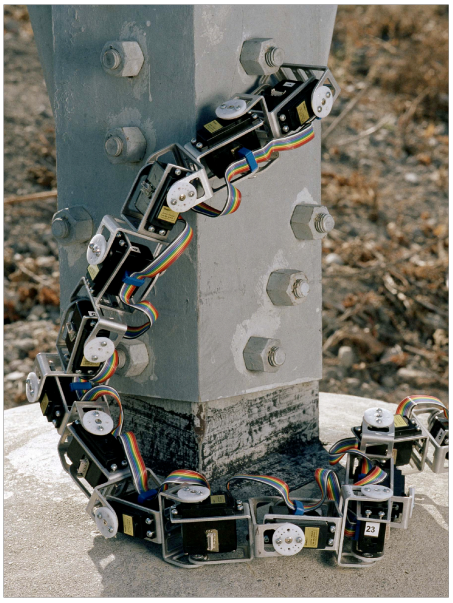

Figure 7. NASA Ames' modular snake robot[17]

modules, the types chosen at the last minute to replace failed modules as well as add new capability. Since the modules are reconfigurable, modules may be packed optimally for launch and redeployed on arrival. Modules may even be used en-route to maintain and inspect the spacecraft, and deploy articulateable hardware.

Modular robot hardware is inherently low cost. Cost are reduced by repetitively building and operating the same modules over and over. Repetition allows learning. Current technology permits cm-scale modules weighing a few tens of grams each, allowing thousands of modules to be delivered to the lunar surface in a single launch, permitting large-run assembly-line economies of scale.

Modular robots have three major drawbacks:

1. In a sense, modular robots may be considered a software material. Software causes the robots to become different entities at different times with substantially different properties and capabilities. This requires very sophisticated control and autonomy software. However, modular robotics software is extremely undeveloped. Only mobility, end effector placement and reconfiguration have been demonstrated.

2. Sophisticated modular robot control and autonomy software will require first rate flight systems software (operating systems, development environments, etc.) and compute hardware. Current flight hardware and systems software are far behind that available on the PCs used by many elementary school students. Flight hardware has inherent limitations due to the radiation environment, but there is no excuse for the second class operating systems and software development environments often used in space flight development. This software cries out for an Open Source solution supported by NASA, DOD, the major aerospace contractors and other heavy space users.

3. For any particular task, a special purpose robot can almost always be designed that is more effective than the comparable modular robotic solution. While the ability to reuse modular robot components for many fundamentally different tasks is a huge advantage, there will be times where some tasks are required so frequently that a special purpose robot is superior.

II. Initial Modular Robot Design

We are investigating centimeter-scale [a] hub/link/joint modular robots similar to tinker toys with additional specialized modules for space exploration tasks. Since each module performs only a single function, the hardware is substantially simplified. All modules types can pass data and power to each other, although purely structural hubs and links may be used as well to reduce cost when large scaffolds must be erected. It may be possible to manufacture passive hubs and links using lunar materials. All modules connect with each other using a single standardized interface. Reconfiguration may be accomplished either by making every module capable of connecting/disconnecting on its own or with specialized modules to make and break connections to simplify module hardware. Modules come in four types: links, joints, hubs, and end-effectors.

Table 1. Module Types

module type	description
link	straight sticks of different lengths with a connector at each end
joint	short link with an internal rotary or prizmatic joint
hub	a compact module with more than two connectors
end-effector	provides specialized functionality; often has only one connector

Link modules are physically passive. Each link connects two other modules. As with tinker toys, links (the links are the equivalent of the wooden sticks) of several standardized lengths can enable rapid construction of large or small robots. The lengths of links to be provided for a mission or set of missions must be defined by a series of trade studies.

Joint modules also connect to two modules, but unlike links provide actuation. Each joint module contains an internal, powered rotary or prizmatic joint. This design separates mechanical actuation from the inter-module interface, simplifying both. If more power is need than can be provided by a single joint module, they can be ganged in parallel just as five or six men can lift a piano that would brake the back of any one of them. The exact set of joints to provide, motor size, internal sensors, and so forth must be defined by a series of trade studies.

Hubs have multiple connectors to enable a variety of thee dimensional geometries. A brick geometry with six connectors enables construction of structures resembling cartesian grids. This has a disadvantage in that such structures lack shear strength. A tetrahedral hub geometry with four connectors allows a diamond lattice geometry which is strong in all directions, but most people are much less familiar with this geometry leading to greater configuration-design and training costs. Many other geometries are possible. Our first generation hub modules have five connectors: a triangle in one plane and two connectors 180 degrees apart normal to the triangle. This enables a variety of global geometries with a single hub type. The precise hub geometries to implement requires to a series of trade studies.

Specialized modules can provide major computation. For example, a iPod with connection hardware added could provide substantial computation in a small, robust package at low cost for ground and ISS demonstrations. When additional computation is needed, simply add more computation modules. Similarly, Bluetooth and other wireless communication systems can be adapted from inexpensive consumer products for initial development.

Specialized sensor modules can provide video cameras, range finders, and other sensors. These sensor modules, which use the same connectors as all other modules, can be deployed with any of several robots as needed for the task at hand. The precise capabilities for sensor modules requires trading cost, power, thermal requirements, and mass against capabilities.

Specialized modules can also provide power and thermal control during the lunar day. Battery modules may, of course, be used and taken to a recharging station as needed. Small amounts of power may be supplied

[a] Our first generation modules are approximately six cm across.

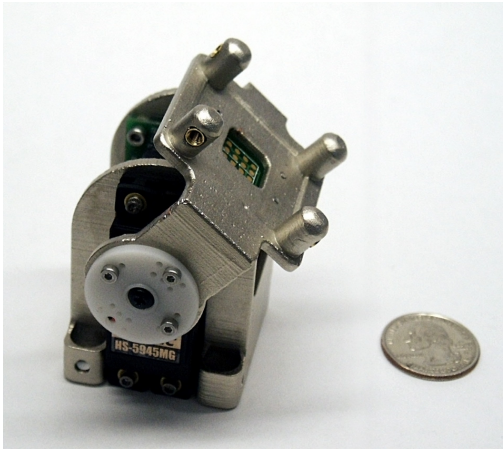

Figure 8. Ames hinge module.

by solar array modules with multiple connectors that are held normal to the sun. Larger amounts of wireless power may be obtained by micowave rectenna modules kept pointed at a large, fixed, solar array on the lunar surface with the power converted to microwaves and beamed. Thermal control on the lunar surface may be accomplished by thin mirror modules oriented to reflect sunlight onto modules that are too cold or, more likely, shading modules that are overheating. Power and reflector sizing will require a series of trade studies.

Fine physical control and manipulation can be accomplished by conventional hand-like robotics and/or grippers with standard intermodule interfaces. This brings up an interesting issue: should simple tools like shovels, screw drivers, hammers, wire cutters and so forth have standard modular robot interfaces or should they by held by a gripper? This requires additional trade studies.

Another major trade study is self-reconfiguration vs. specialized reconfiguration modules. In some of the existing modular robot work modules can attach to and detach from each other. However, this significantly complicates the interface hardware, at least in the modules designed to date. An alternative is to design modules to be mated/demated by external control and developing a module type specifically for this task. This alternative has the disadvantage that the reconfiguration modules will constitute a bottleneck and, if all of them fail, the robot will not longer be reconfigurable – at least until the next resupply flight brings new reconfiguration modules up.

Other specialized modules might include pads for feet so that connector interfaces are not directly driven into the lunar regolith. Wheel modules are also possible to improve mobility.

We favor cm-scale modules since that is the smallest that can be easily manufactured. In general, the smaller the modules the better. Smaller modules mean that more can be launched for the same mass, providing greater flexibility and redundancy. Smaller modules also have a greater surface area per unit mass providing more power from solar cells and thermal benefits as well. Smaller modules imply smaller, weaker motors per module, but more powered modules can be ganged together. Smaller modules can fit into tighter spaces, without sacrificing the ability to build large structures simply by using more modules.

A. First Generation Ames Modular Robots

We have designed and are currently manufacturing an initial set of low-cost hub and joint modules. The joints are hinge-type modules similar to those found in early versions of Yim's Polybot and the existing NASA Snakebot.[17] The hubs have a novel topology with five connection points arranged at both 60° and 90° angles. With this design it is possible to construct rectangular and hexagonal lattices for use in assembling larger

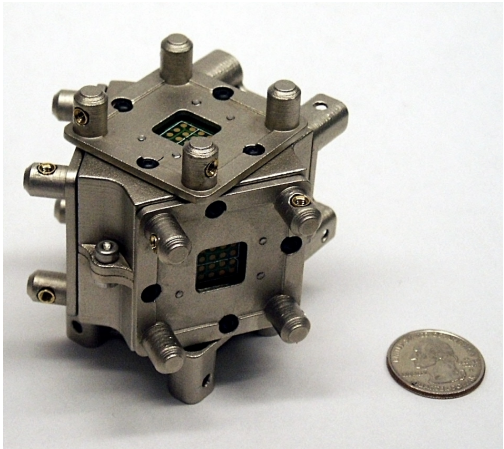

Figure 9. Ames hub module.

structural units. The connectors have four-way electrical and mechanical symmetry, and were designed for easy manual reconfigurability using thumbscrews. These modules, and other interoperable modules to follow, will thus provide an inexpensive research platform for studying the many open questions that do not directly involve automated reconfiguration.

In these designs we have chosen to take advantage of modern rapid-prototyping technology. The mechanical components are printed in an ABS-like plastic using a stereo-lithography apparatus (SLA) and then electroplated with thin layers of copper and nickel. This results in substantially more rugged parts than are typically produced by SLA while keeping the mass and small-quantity cost very low. Low module mass is important because the inexpensive servomotors in the joint modules have limited torque. The low manufacturing cost has far greater implications, however, because it opens the door for inexpensive production of a wide variety of modules. Once the initial two module types are fully operational we will begin to explore a range of other link, joint, hub, and end-effector module types in an attempt to learn more about what collection of modules is most useful for a variety of real-world applications. This exploration would be financially difficult using conventional manufacturing techniques with high per-part set-up costs.

Each module contains a microcontroller and FPGA to handle control and communications. The assembled modules form a simple ad-hoc network which will be used as a testbed for a variety of distributed control problems. As a simple example, we are exploring the distributed computation and control of end-effector trajectories given only the local knowledge of each module. Though this problem is trivial to solve using a centralized computer, such a solution does not scale to the limit of many small modules assembled to create large robots. In this limit distributed cooperative control of complex (and therefore overconstrained) robot topologies is expected to be particularly important, as the modules will have to closely coordinate their behavior at a low level in order to ensure proper motion and reasonable energy efficiency.

B. Software

Dividing robotic exploration functionality into separate modules, each with a single function, substantially reduces hardware complexity. It also eliminates the need to make individual modules redundant since many modules of each type are launched; reducing hardware complexity still further. However, there is a price to pay. The software to direct and control modular robots will be extremely difficult to develop. Fortunately, almost all of the software can be developed in simulation, and the rigid body simulators necessary are readily available, although regolith and fabric simulators needed for some tasks will be more difficult to develop.

The ability to study the capabilities of a set of module designs in simulation provides a unique opportunity

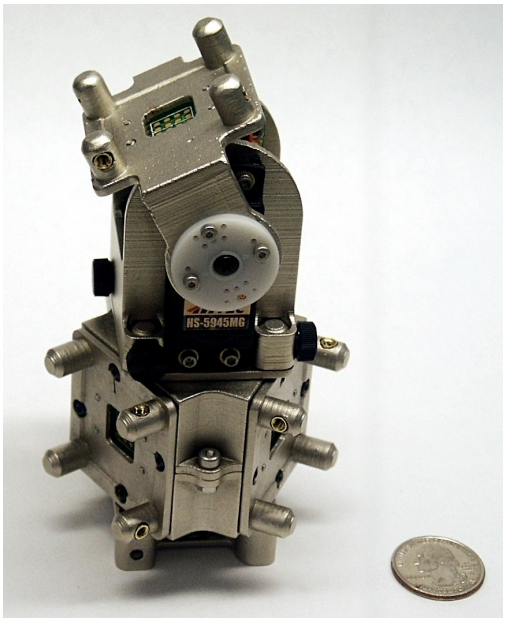

Figure 10. Connected Ames hinge (on top) and hub modules.

to perform the large number of trade studies necessary to determine module sizing, hub geometry, sensor capabilities, gripper choices, small tool integration and so forth. For a given choice of module designs, evolutionary algorithms can be used to design robot configurations and control strategies for a large selection of exploration tasks. These tasks can be executed in software simulation to evaluate the designs and provide direction to the evolution.

Evolutionary algorithms include the genetic algorithm, genetic programming, evolutionary strategies, stochastic hill-climbing, and even simulated annealing. These algorithms typically start with one or more randomly generated solutions to a problem, then iteratively evaluate the solution and probabilistically modify the better solutions to evolve an acceptable solution to the problem at hand. In pseudo-code:

```
generate N random solutions to a problem given some representation
evaluate each solution
loop
    parent(s) = randomly chosen solution(s), with bias towards better solutions
    child = random modification of one parent and/or a combination of parts of two parents
    evaluate child
    accept or delete child (details depend on the algorithm)
until satisfactory solution found or exhaustion sets in
```

To solve a problem using evolutionary systems requires a representation for solutions (called individuals), a way to modify and/or combine solutions (called variation operators – mutation and crossover), and some way to evaluate individuals (called the fitness function) [b]. The fitness function must be able to evaluate any individual, no matter how good or bad, and must be able to determine which of two individuals is best almost all the time. If there are a too many ties, evolution receives no direction and degenerates into random search.

[b] note that terminology varies a bit among the communities using these algorithms

There are often two representations for a solution, the genotype and the phenotype. The genotype is designed to make mutation and crossover easy and effective and to limit the size of the search space. The genotype is used as input to a program that generates the phenotype, which is then evaluated by the fitness function. For example, in our work the genotype is a computer program in a special purpose, generative language. Generative means that it evolves subroutines and other reuse constructs. The program is executed to generate robot morphology and a neural network for control.[3]

Many decisions must be made to chose the exact set of modules for solar system exploration: link lengths, motor strength, hub geometry, self-connect/disconnect vs. specialized modules, tool modules vs. grippers, etc. Furthermore, these decisions interact with each other. For example, the size of the joint motors must be consistent with the longest link to insure that at least one link can be moved without burning out the motor. One way to approach this problem is to hire hundreds of aerospace engineers to analyse as many cases as they can. This works, but is a tad bit expensive, U.S. aerospace engineers run from a quarter to a half a million dollars a year including overhead.

Another way to reduce, although not eliminate, the engineering time needed to make these choices is to design the module set using evolvable systems. The methods have recently begun to produce human-competitive results in real-world application domains. As an example, in NASAs Space Technology 5 (ST5) mission a requirements-compliant evolved antenna was produced that is scheduled to be flown in 2005[18].[19] The evolved antenna will be the first evolved hardware ever flown on a NASA mission. An initial design was as good as a human-designed model. A second design was evolved in less than 4 weeks when communications requirements were revised because the anticipated orbit changed.

To use evolvable systems for module design trade studies, one must evolve solutions to a large set of space exploration tasks using different module designs. The quality of all the solutions can be compared and the best module design chosen. It is also possible to evolve design parameters (e.g., link lengths). While this requires a great deal of computer time to test many tasks on millions of configurations and controllers, it is an embarassingly parallel problem so thousands of processors can be employed efficiently.

The ST5 antenna and walking modular robots were evolved on approximately 80 processors in a simple Linux cluster. This will clearly be inadequate for modular robotics trade studies. Fortunately, NASA Ames is currently configuring a 10,000 node system, called Columbia, with more powerful individual processors.

Evolving modular robot configuration and control systems has been demonstrated. Hornby[3] has evolved both the morphology and controller for walking modular robots. The best robot was then built and tested to demonstrate that the evolved robot worked as expected. This approach must be scaled up considerably to evaluate module designs for a wide variety of space exploration tasks.

To use evolutionary systems to perform the necessary trade studies we must determine a representative set of tasks for modular robots to perform. This can be accomplished by designing representative missions, and then determining the set of tasks necessary to perform those missions. Performance on the tasks can then be used as the fitness function for evolutionary trade studies.

III. Lunar Base

To identify a representative set of tasks, let us examine the development of a lunar base to exploit *in situ* resources . The purpose of the base is to mine lunar materials, specifically the loose, surface regolith, for on-site use and for delivery to orbit to enable extensive exploration and colonization of the solar system. Note that the lunar regolith is rich in silicon, oxygen, and metals – most of the mass for rocket fuel (O_2), all the bulk materials needed for solar power satellites, and perfectly adequate material for radiation protection. Furthermore, it may be possible to manufacture simple, passive modules from lunar regolith using modular robots, molds and parabolic mirror modules to focus sunlight.

The base is built up by a series of missions to the same location to enable mining operation without developing a gigantic Earth-to-orbit booster. Each mission is launched from Earth by conventional vehicles. The first mission has sufficient modules to prepare for the second, and a number of missions are expected to be necessary before sufficient modules and equipment are onsite to begin mining operations.

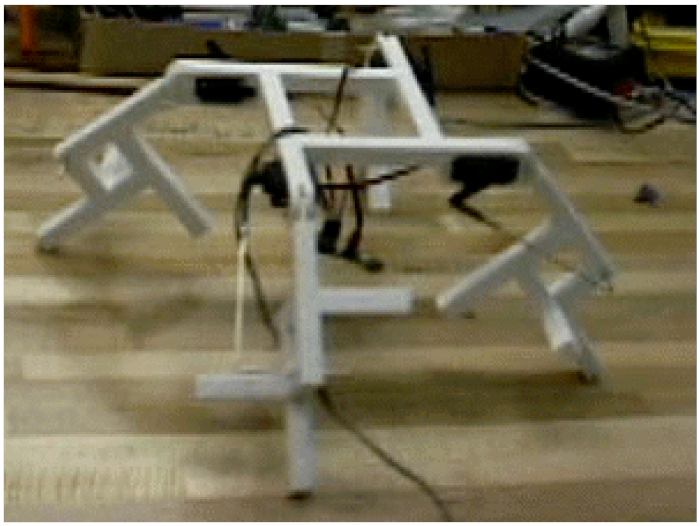

Figure 11. One of Greg Hornby's evolved walking robots implemented by manufacturing with a 3d printer

The LEO-Moon transfer vehicle uses fuel efficient, low-thrust solar electric propulsion, so considerable time (months) is needed to reach lunar orbit. Small chemical thrusters are required for lunar landing. Once on the Moon the spacecraft's solar arrays will be used to power the base, its communication antennas will be used to link to Earth, and processors to provide computation. Only the propulsion systems, attitude control, and star trackers will be useless on the Moon. These can be removed and returned to orbit for reuse.

The modules are tightly packed during launch. On reaching orbit, the modules reconfigure to unfold the solar arrays, thermal radiators, and communication antenna. The modules can also deploy cameras to inspect the craft and assist with checkout. Once the vehicle is properly configured, the modules can reconfigure for a long cruise phase as the vehicle spirals into higher Earth orbit, eventually transitioning to a Lunar orbit. During cruise phase, module utility is limited to inspection of the spacecraft.

As landing approaches, the modules reconfigure to stow the solar arrays, thermal radiators and communication antenna, then reconfigure again as landing legs. Small chemical thrusters slow the craft and the modular robots absorb the final shock of landing. The robots then lower the vehicle to the surface; and redeploy the solar arrays, thermal radiators, and communication antenna for surface operations.

The first task upon configuring the spacecraft for surface operations is to deploy a highly reflective canopy to protect modules during the lunar night. By limiting operations to the lunar day, solar power is available and cold-induced materials brittleness is avoided. The reflective surface is on the inside to keep the temperature under the canopy as high as possible.

Note that this is the first operation that cannot be modeled with a rigid body simulator. The dynamics of a thin, flexible canopy require a more sophisticated model to insure that the material does not bind, rip, or become entangled with the surface or other mission hardware. The canopy can be held up by strategically placed passive modules and conveniently positioned lunar rocks if available. The outer rim can be buried in lunar regolith to hold the canopy in place. Great strength is not needed since there is no wind or rain. However, sufficient strength and stability will be needed to support regolith dust thrown up by subsequent landings.

Manipulation of lunar regolith to bury the rim of the canopy also cannot be simulated by a rigid body simulator in reasonable time. If each grain of regolith is simulated by a body, the calculations necessary to determine the collisions would overwhelm any of today's computers, although other modeling techniques

are available. Fortunately, the unfolding and securing operation can be tested on Earth with sufficiently strong joint motors. The Lunar mission joint motors can be weaker since the objects on the lunar surface experience only 1/6g.

Once night-time shelter has been secured, the remaining modules (those not needed to support the canopy) may be reconfigured to prepare a landing site for the first resupply mission. First, modules are configured for surface roving and exploration; perhaps small walkers with video camera and wireless communication modules. A site must be found close enough for mutual support and far enough that materials ejected by the rocket exhaust during landing will not present insurmountable problems. A radio beacon is then placed at the site to guide the next mission and rocks removed as necessary. Then the hard part starts.

If a solid landing pad could be created then rocket exhaust necessary for landing would not kick up so much debris. Apollo lunar landers deposited dust kilometers from the launch site. Dust and rocks thrown up by a landing can obscure sensor windows, foul mechanism, and even bury small systems.

One way to reduce debris is to melt or sinter a landing pad. In this context, to sinter means to heat the regolith below the melting point but sufficiently to fuse the particles. If the surface impinged by the thrusters is sufficiently fused, much less debris will be distributed.

The simplest approach is to melt the surface. Lunar basalt has a melting temperature of approximately 1400-1600°K. If allowed to solidify quickly it forms into a very hard, exceptionally strong, brittle polymeric glassy material. If cooling is slower the material is crystalline and much less brittle. Since the landing site is used rarely, brittleness is not a problem. Even if it cracks on landing it can be repaired (or another site prepared) before the next mission. The surface may be melted by concentrated sunlight as these temperature are well within the capabilities of terrestrial solar furnaces. Sunlight may be concentrated by many small mirror modules deployed by modular robots, or by a larger mirror made of the same material as the canopy and shaped by appropriately placed modules. Sunlight can be directed to the working area by a smaller, module-controlled flat mirror.

Allen, Graf and McKay demonstrated sintering of full-scale bricks of lunar regolith simulant in two hours at 1100°C.[20] Small scale compaction and a thermally insulating mold were found to be necessary for strength. Modular robots can compact the regolith simply by walking (and perhaps jumping up and down) on it. This process requires slightly lower temperatures than melting and creates a much weaker material.

Once the second landing site is prepared, the module robots reconfigure for a traversal and proceed to the third landing site location to repeat the process. A second identical spacecraft is flown to the Moon, albeit with a different set of modules. Some of these modules will reinforce the landing site team if necessary. The rest will deploy equipment to begin production of simple, passive modules to replace and thus free the more capable modules supporting the canopies and other static tasks.

Once the lunar base can produce passive modules (those without power or data) growth is no longer dependent solely on resupply from Earth. This can be accomplished by sending molds for several link and hub modules on the second flight, along with a high temperature bucket. Lunar regolith is melted in the high-temperature bucket, then poured into the modules. For this to work, the mechanical interface between modules must be simple and robust to manufacturing errors. The modules produced can be used for simple structural tasks such as holding up the canopy, building towers for video observation, and maintaining the shape of solar furnace canopy material.

It may also be possible to manufacture passive modules from lunar regolith using a process similar to that employed by 3D printers today. First, the robots smooth the regolith surface. Then a parabolic mirror is used to sinter a thin layer of regolith with the proper shape. A layer of loose regolith is then deposited and another layer of sintering accomplished. Thus, parts are built up one layer at a time.

The third and later missions will bring components for a lunar mass driver to deliver regolith to orbit. A mass driver is an electromagnetic catapult using magnetic waves to propel a magnetically-levitated bucket along a track using the same principles as maglev trains and Disneyland's California Screamin' roller-coaster. 1800g acceleration, requiring only a tenth of a second to reach lunar escape velocity, has been demonstrated in the laboratory.[21] This allows launch from the Moon to escape velocity without propellant, only electrical energy. Buckets are filled with regolith by modular robots with shovel end effectors, accelerated to escape

velocity, their regolith payload is released, and then the bucket is deccellerated and brought around for reuse. The energy required to accelerate the bucket itself (although not the payload) can be reclaimed during the decceleration phase. Mass drivers for lunar-materials-to-orbit delivery were extensively studied in the 1970's and 80's.[22]

The setup and operation of the mass driver is a significant challenge for modular robotics. While the necessary mechanical actuators can probably be designed and manufactured, the ability to properly control the robots is well beyond current state of the art and will require substantial improved automony and teleoperation software to achieve affordably.

IV. Task Autonomy

There is one serious problem both special purpose and modular space robots share: to date both have had extremely limited autonomy. With the exception of a few limited experiments, every move of every space robot has been precisely scripted on the ground or, in the case of the Canadian robot arms, interactively controlled by astronauts in orbit. This leads to marching armies of ground controllers to generate command sequences and analyze sensor inputs to generate the next set of command sequences. For example, approximately 240 ground personnel were assigned to operate the Spirit and Opportunity Rovers on Mars during the prime mission.[23] This a little like trying to coach a football team by telling every player exactly how to move each limb on every play. It would be a lot easier to just tell the quarterback who to hand off to or have the receivers go long. The individual players can figure out the details, the robots should too.

Autonomous operation of modular robots for the lunar mine mission is well beyond the state of the art and is likely to remain so for quite some time. Not only are some of the tasks complex, but there are large number of error conditions the robots can get themselves into. Anticipating and developing escape strategies for all possibilities in advance is nearly impossible, and certainly expensive. For example, one of the vehicles in DARPA's recent off-road autonomous vehicle race traversed many kilometers of difficult terrain but eventually got stuck. After a long unsuccessful struggle to free itself, a human operator was brought in and was able to back the vehicle out in a few seconds.

The problems associated with completely autonomous robotics can be solved by teleoperation, where human operators on the ground guide the robots through the detailed execution of each task. This is possible in principle, but would require such a large ground team as to be unaffordable. Worse, not only is detailed guidance for each joint of each robot tedious, but the minimum three second speed-of-light delay to the Moon and back substantially reduces efficiency.[1] Thus, neither completely autonomous operation nor second-to-second teleoperation are practical alternatives.

Fortunately, there is a middle ground, sometimes called task automation. Mission operations can be divided into discrete tasks, each one of which is amenable to automation. Human teleoperators (or, later, automatic planners) then direct the robots by identifying the next task to accomplish, and the robot executes the task autonomously. For example, rather than have the robot decide that a particular rock must be removed from the landing site and where to put it, a human operator could identify the errant rock and where is should be taken. The robot then has the more tractable task of moving the rock from point A to point B. This still leaves the problem of error conditions.

In the early days of automobile development, engineers tried to anticipate every possible problem and build vehicles strong enough to survive all imaginable possibilities. This worked but resulted in massive cars because of over-design for situations that never happened. One of Henry Ford's major innovations in developing the Model T was to build the vehicle as light as possible, test under extreme conditions, and redesign the parts that failed.

Similarly, although task automation is tractable, it will be very difficult to anticipate all probable failure modes. Since radio communication with Earth only involves a three second speed-of-light delay and the LEO-Moon transfer vehicles can function as relays, when robots get into trouble they can enter a safe mode and call for help. Human teleoperators can then take joint-level control, like the DARPA autonomous vehicle contestant, and work the robot out of its situation, with help from other modular robots if appropriate. A

record of the rescue can be stored in a database and used to guide development of new automation software to detect and recover from the situation automatically in the future.

To implement this approach, the high level operations described above must be broken down into tractable individual tasks. These tasks, the robotic equivalent software subroutines, can then be developed first in software simulators, then, where feasible, on ground testbeds, and finally on the Moon. Furthermore, the individual tasks can be used in the fitness function for the evolutionary systems conducting modular robot component design trade studies. The automation solutions discovered by evolution can be cleaned up, analyzed, and used as the basis for baselined task automation software.

V. Spiral Development

We expect each task to go through three steps before integration into the operational control software:

1. Software simulation
2. Ground testbed operation
3. Testing on the Moon

For some tasks it may also make sense to also test in orbit, particularly if a large centrifuge capable of 1/6g becomes available. No such hardware is in development.

A. Software Simulation

Software only simulation has major advantages for early stages. A wide variety of scenarios can be set up, tests are quick and inexpensive, results can be visualized in complete detail, lunar gravity levels are easily modeled, and costs are much lower than hardware tests.

Most modular robot operations can be simulated with a rigid body simulator. Each module, or joint component, is modeled as a rigid body responding to Newton's laws of motion. We are currently using the Open Dynamics Engine (ODE - http://ode.org/) an excellent open source product. We are integrating ODE into a simulation environemnt that also simulates module computation and inter-module communication to examine the control and communications algorithms. This will allow us to test the scalability of various control schemes, and it will opens the door to better automated search of robot morphology and control spaces using evolutionary or other algorithms.

While the current version of ODE is perfectly adequate for trade studies, developing engineering level control software is expected to require improvements. Specifically, a more accurate integrator [c] and better friction models will be required. It will also be necessary to integrate ODE with lunar regolith models and fabric-type models for the canopy and solar mirror materials which are expected to be too thin to be modeled as rigid bodies. Since the ODE source code is readily available, these modifications can be made without rewriting existng functionality.

Although ODE can model the bulk of mobile robot operations, modeling the forceful interaction between the robot end effector and the terrain is a much more difficult prospect. Singh[24] divides this into two related, but potentially separable problems. The first is to determine the effect on the shape of the terrain when it is penetrated by a bucket or blade. The second problem is to compute the reaction forces on the end effector caused by this penetration. Both problems can be solved via finite element methods (FEM), however these approaches are computationally taxing, and cannot be computed in real time. Alternatively, first order approximations may be separately applied to these problems. For instance, to calculate the resulting terrain shape, the profile of the bucket or blade as it traverses through its trajectory could be intersected with a grid based volumetric model of the terrain. This essentially results in a trench in the volumetric model, which is subsequently allowed to "settle" into a configuration consistent with the natural angle of repose of the

[c] he integrator is the part of ODE that moves the bodies through time.

material. Likewise, approximate soil reaction forces can be computed using a first principles static analysis of the wedge of material just in front of the moving blade. Luengo[25] and Cannon[26] demonstrate the ability to learn the parameters of these equations, and model these forces with reasonable accuracy many times faster than real time.

Although these approximation techniques may be used to model forceful interactions, they rely on a simplifying assumption limits their use in this application. In particular, these techniques assume that the material is relatively homogeneous. Lunar regolith may contain inclusions of sufficient size relative to the size of the end effectors such that this assumption is probably not valid. So although the approximation techniques may be useful for first order analysis and testing, verification will ultimately be required using bench level experiments.

Using the simulators requires specific module designs, which, as we have seen, can be subjected to trade studies by a combination of evolutionary software and the models. The simulators can be used to develop task level automation software and teleoperation user interfaces. Teleoperation interfaces will require substantial human testing to optimize. Task level control should be relatively easy, but joint level control necessary for rescuing robots in trouble may require significant creativity to get good results.

Lunar gravity and terrain can be easily modeled, but there are some important weaknesses of software simulation that must be born in mind. All models are inaccurate. Models must be tuned to answer fairly specific question and cannot be used outside their accuracy domain. In our case, the models must be accurate enough to answer two questions:

1. Trade studies: how well do specific module designs perform, relative to other designs, for exploration missions?

2. Software design: how does control, direction, and planning software perform given a set of modules designs?

Inaccuracies in the models can impact the quality of the answers. Specifically,

1. Manufactured items are rarely proportioned or operate exactly as designed.

2. Friction models are notoriously difficult to get right and inaccuracies will affect low-level control software.

3. Fine control, particularly fit detail, can be inaccurate.

4. The modules are not true rigid bodies.

5. Dust is a major factor on the lunar surface and is very difficult to model. Apollo astronauts were covered with a clingy dust after even short periods on the Moon. When connecting and disconnecting modules, dust can play a major role. Mechanisms fouled by dust will exhibit substantially different properties. Dust can obscure cameras, reduce solar array power output, and reduce the solar intensity achievable by mirrors.

Most of these inaccuracies can be minimized by bench level physical experiments with the relevant systems to parameterize the models. For tasks involving lunar regolith there is limited regolith (that returned by Apollo astronauts) available for experiments. However, simulant can be produced from volcanic ash at a reasonable price (approximately $40,000 for 30 tons).

Model parameters (e.g., module length, motor power, friction parameters, etc.) can be varied slightly with random noise to insure that control and direction software is robust to small deviations in the actual modules. The fitness function can be run several times with different noise profiles and the worst score taken. This will favor robust over brittle solutions.

The existence of reasonably accurate software simulators substantially reduces the entry cost for developing modular robotic software. Since a moderate sized modular robotic team can easily be simulated on a

modern PC costing only a few hundred dollars, not only can aerospace and academic organizations develop modular robot software, but interested individuals such as students and retired programmers can do the work. Space development enjoys great popularity in large parts of the public, and there are some who will put significant volunteer efforts into NASA related work.

Free development can be significantly stimulated by contests, with or without prizes. The RoboCup annual soccer robot contest regularly draws over a thousand contestants, none of whom are paid by the contest organizers. RoboCup has both physical and software-only leagues. The recent DARPA autonomous off-road land vehicle contest drew 106 teams with a $1 million prize. Total development expenditures by the contestants dwarfed this figure. Finally, the Ansari X Prize, $10 million for the first vehicle to carry three people or the equivalent mass into space twice within three weeks, has drawn 20 entries, several of which have conducted flight tests. The leading contender alone has spent several times the purse in development. All this suggests that a well designed series of contests to develop software to configure, control and direct simulated modular robots operating in a lunar environment may supplement traditional software development activities.

B. Earthly Test Beds

Once an initial module-set design has been chosen and a reasonable number of modules manufactured it is possible to initiate ground testing. Both teleoperation interfaces and automation software can be tested on the ground in suitable facilities. NASA Ames has two such facilities: the MarsScape outdoor testing facility and a large vacuum chamber in building 242. Both will require addition of lunar regolith simulant. Commercial Wi-Fi wireless communication systems are perfectly adequate for controlling modular robots without tethers.

As with any model, ground-based testing facilities cannot exactly model the target environment. The main problem is that 1/6g is impossible, so joint motors sized for lunar operations will be unable to function properly. Also, dust electrostatics will be difficult or impossible to reproduce. FInally, the lunar thermal environment is very difficult to model. Temperature extremes can cause module disconnect/connect and global fit problems due to expansion and contraction of parts.

One opportunity opened up by actually producing modules is for toys and spinoff applications. Relatively inexpensive modules incapable of withstanding the rigors of space may find a fertile market in the toy business, and perhaps even for practical Earthbound applications. Contests using physical modules are also possible, but the cost of entry is much higher than for software-only contests. Thus, the government may need to provide contestants with modules, thus limiting the number of potential participants.

C. Lunar Testing

The ultimate test site is, of coarse, the lunar surface. Such testing is extremely expensive, but can become useful once enough modules have been amassed on the surface than not all are needed for ongoing operations.

VI. Conclusion

We are in the initial stages of developing modular robotics for space operations. Such systems have great promise for orbital, asteroidal, lunar and martian operations. In this paper we have focused on lunar operations. Modular robots are particularly well suited to space operations due to their inherent flexibility, redundancy, and low manufacturing cost.

A great number of design decisions must be made, which is difficult since modules are intended to be used for a wide variety of tasks, many of which will not be known for decades. We propose approaching these trade studies using evolutionary systems to automatically generate solutions to a large number of operational problems. These solutions can be compared to choose a reasonable set of module designs.

Control software is the Achilles Heel of modular robotics and a substantial effort will be necessary to develop it. We are examining task-level automation with human teleoperators to handle unexpected situations. Both human operations and automatic planners can be used to sequence tasks under non-failure situations. Most software can be developed in software-only simulations then migrated to ground and lunar testbeds.

Properly developed, automated modular robotics may substantially contribute to exploration and settlement of the high frontier.

Acknowledgments

This work was funded by NASA's Computing, Information, & Communications Technology Program, Advanced Information Systems Technology Program (contract AIST-0042), and by the Intelligent Systems Program. We would like to thank Peter Cheeseman and Silvano Colombano for many stimulating discussions and ideas.

References

[1]Globus, A., "Remote Teleoperation Earth to Moon - An Experiment," *Fifth Princeton/AIAA Conference on Space Manufacturing*, 1981.

[2]Yim, M., *Locomotion with a Unit-Modular Reconfigurable Robot*, Ph.D. thesis, Stanford University, 1994.

[3]Hornby, G. S. and Pollack, J. B., "Creating High-Level Components with a Generative Representation for Body-Brain Evolution," *Artificial Life*, Vol. 8, No. 3, 2002.

[4]Fukuda, T. and Kawaichi, Y., "Cellular robotic system (CEBOT) as one of the realization of self-organizing intelligent universal manipulator," *Proc. of the Intl. Conf. on Robotics and Automation*, 1990, pp. 662–667.

[5]Chen, I. and Burdick, J. W., "Enumerating the Non-Isomorphic Assembly Configurations of Modular Robotic Systems," *Proceedings of the IEEE/RSJ International Conference on Intelligent Robotic Systems*, 1993, pp. 1985–1992.

[6]Chen, I.-M. and Guilin, Y., "Automatic Model Generation for Modular Reconfigurable Robot Dynamics," *SME Journal of Dynamic Systems, Measurement, and Control*, Vol. 120, No. 3, 1998, pp. 346–352.

[7]Chen, I.-M. and Burdick, J. W., "Determining task optimal robot assembly configurations," *IEEE International Conference on Robotics and Automation.*, 1995, pp. 132–137.

[8]Chen, W., Yang, G., Ho, H. L., and Chen, I.-M., "Interactive-Motion Control for Modular Reconfigurable Manipulators," *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2003, pp. 1620–1625.

[9]Yim, M., "A Reconfigurable Modular Robot with Many Modes of Locomotion," *Proc. of the JSME Int. Conf. on Advanced Mechatronics*, 1993, pp. 283–288.

[10]Yim, M., "PolyBot: a Modular Reconfigurable Robot," *IEEE Intl. Conf. on Robotics and Automation*, 2000.

[11]Yim, M., Roufas, K., Duff, D., Zhang, Y., Eldershaw, C., and Homans, S., "Modular Reconfigurable Robots in Space Applications," *Autonomous Robots*, Vol. 14, 2003, pp. 225–237.

[12]Rus, D. and Vona, M., "Self-reconfiguration Planning with Compressible Unit Modules," *IEEE Intl. Conf. on Robotics and Automation*, 1999.

[13]Kotay, K. and Rus, D., "Algorithms for Self-reconfiguring Molecule Motion Planning," *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2000.

[14]Butler, Z., Kotay, K., Rus, D., and Tomita, K., "Generic Decentralized Control for a Class of Self-Reconfigurable Robots," *Proc. of the Intl. Conf. on Robotics and Automation*, 2002.

[15]Hornby, G. S., Lipson, H., and Pollack, J. B., "Evolution of Generative Design Systems for Modular Physical Robots," *Proc. of the Intl. Conf. on Robotics and Automation*, 2001, pp. 4146–4151.

[16]Hornby, G. S., Lipson, H., and Pollack, J. B., "Generative Representations for the Automatic Design of Modular Physical Robots," *IEEE Transactions on Robotics and Automation*, Vol. 19, No. 4, 2003, pp. 703–719.

[17]Heith, G., Colombano, S., and Neveu, C., "A Serpentine Locomtion System for Biomorphic Robotics," *Presentation at the Bio-inspired Engineering for Exploration Systems (BEES) Conference*, December 2000.

[18]Lohn, J. D., Linden, D. S., Hornby, G. D., Kraus, B. F., Rodriguez, A., and Seufert, S., "Evolutionary Design of an X-Band Antenna for NASA's Space Technology 5 Mission," *Proceedings of the 2004 IEEE Antenna and Propagation Society International Symposium and USNC/URSI National Radio Science Meeting*, Vol. 3, 2004, pp. 2313–2316.

[19]Lohn, J. D., Hornby, G. S., and Linden, D. S., "An Evolved Antenna for Deployment on NASA's Space Technology 5 Mission," *Genetic Programming Theory and Practice II*, edited by U.-M. O'Reilly, R. L. Riolo, T. Yu, and B. Worzel, Kluwer, in press.

[20]Allen, C. C., Graf, J. C., and McKay, D. S., "Sintering Bricks on the Moon," *Engineering, Construction, and Operations in Space IV*, American Society of Civil Engineers, 1994, pp. 1220–1229.

[21]O'Neill, G. K., *The High Frontier*, Space Studies Institute Press, Princeton, New Jersey, 1989.

[22]O'Neill, G. K., "Space Resources and Space Settlements," Tech. Rep. NASA SP-428m 1977, http://lifesci3.arc.nasa.gov/SpaceSettlement/spaceres, NASA, 1977.

[23]Bass, D. S., "Personal Communication," August 2004.

[24]Singh, S., "The State of the Art in Automation of Earthmoving," *Journal of Aerospace Engineering*, Vol. 10, No. 4, October 1997, pp. 179–188.

[25]Luengo, O., Singh, S., and Cannon, H., "Modeling and Identification of Soil-tool Interaction in Automated Excavation," *IEEE/RSJ International Conference on Intelligent Robotic Systems*, October 1998, pp. 1900–1906.

[26]Cannon, H. and Singh, S., "Models for Automated Earthmoving," *International Symposium on Experimental Robotics*, March 1999, pp. 139–148.