Presented at the 6^{th} International Symposium on Artificial
Intelligence, Robotics and Automation in Space
Montreal, Canada, 18–22 June 2001

A New Breed of Explorer: Development of a Network of Mobile Robots for Space Exploration

T D Barfoot, E J P Earon, G M T D'Eleuterio

University of Toronto Institute for Aerospace Studies
4925 Dufferin Street, Toronto, Ontario, Canada, M3H 5T6
tim.barfoot@utoronto.ca, e.earon@utoronto.ca, gde@utias.utoronto.ca

Keywords collective robotics, planetary space exploration, network science, decentralized control, multiagent systems, autonomous agents

Abstract

A concept network of autonomous mobile robots intended to carry out tasks related to planetary space exploration is described. Many aspects of the system have been fashioned to make the robots function as independently as possible yet still accomplish cooperative tasks. Representative space exploration tasks are outlined followed by brief descriptions of the hardware and control. One specific task, deploying an array of sensors for network science, is used as an example. Technical challenges and preliminary experimental results are discussed.

1 Introduction

Autonomy has perhaps no more serious a connotation than when considered in the context of space exploration. With the telecommunications era blossoming on Earth, it is becoming difficult to truly be alone on our planet. However, with no foreseeable method of circumventing the laws of physics, autonomy of the space explorer is not merely desirable but necessary. Most missions of the past have accomplished this through either the presence or telepresence of man. To travel beyond our solar system will invite prohibitive communication latencies to Earth. However, even within our solar system (e.g, Mars), teleoperations can have a delay of tens of minutes. This can be a serious limitation when equipment is designed for a finite lifespan (e.g., solar panel degradation due to wind and dust).

Despite this, an abundance of science needs to be done on and below the surface of Mars and the other planets and planetoids in our solar system. One of the most popular concepts in planetary science today is *networking*. Network science commonly refers to that requiring a distribution of (possibly simultaneous) measurements or a distribution of platforms. Consider, for example, seismology studies of an alien body that will require sending a signal from one point on the surface to be read at several other points in order to analyze the material characteristics of the body. Or, consider the deployment of a very-low frequency array (VLFA) on the Moon to allow for hitherto unattainable astrophysical observations using radio astronomy. Such an observatory will require a number of dipole units deployed over a region of a few hundred square kilometres. This concept was in fact studied by the 1993 session of the International Space University [6]. These and other examples of network science could be facilitated by a network of small mobile robots, similar to a colony of ants. Certainly the effect of communication latencies would be magnified if several robots were to be controlled through teleoperations, escalating the need for autonomy.

Figure 1: From right to left: Deimos, Phobos, Eris, Enyo, Thanatos, Moros. Six mobile robots making up the RISE network.

Network science missions have been in the works since the mid-1980s [2] including NASA's (National

Aeronautic and Space Administration) Mars Environmental Survey (MESUR) which had plans for as many as 16 stationary landers, ESA's (European Space Agency) Marsnet which also intended to have a small network of surface stations, and most recently France's NetLander which plans for 4 landers and is proposed to be launched in 2007. Scientific fields that lend themselves to network science include, but are not limited to, seismology, meteorology, geology, magnetism, geophysics (subsurface heat flow, water detection), astrobiology, astronomy, mineralogy and cartography. Many planetary scientists believe that is only through distributions of measurements, as provided by network science, that we will be able to fully understand the inner workings of a planet.

Having outlined the potentially extensive role that network science can play in planetary exploration, it is only proper that we address the technical objectives that need to be achieved to realize network science. What is required is, in essence, a method of deploying a distributed system of sensors and actuators. One option in the large spectrum of possibilities, is *network robotics*, that is, a network of mobile robots or rovers. To investigate the possibilities of such a network, we have constructed a testbed facility, the RISE (Robotics In Space Exploration) Network. This facility consists of six mobile robots which communicate with each other and with a desktop computer through radio communications. Figure 1 shows the six mobile robots of the RISE Network. The focus of RISE is on the autonomous control of such a network. Each robot has its own local computing facilities yet the group must work together to accomplish the types of task that would be necessary for network science. A centralized controller is not required but can be used to issue high-level commands (e.g., start, stop, pause) or upload/download information from the robots. The desktop computer serves in this regard and as an observer (e.g., plotting telemetry). One goal of actually implementing this group of autonomous agents in hardware is to better ascertain the needs of an actual space flight system of mobile robots.

This paper is organized as follows. A brief review of other work in network robotics and robotics for planetary exploration is provided. The task requirements of network science are distilled into a few representative tasks which are to be accomplished by the RISE Network. Our choices for sensors and actuators are described in enough detail to highlight the key attributes of each, including a description of a dynamic radio communication network protocol. A description of our behaviour-based control architecture is given followed by initial results on one of the representative tasks, deploying a sensor array.

2 Related Work

Our work is attempting to fuse space robotics research with that of network robotics. There is not a great deal of related work that incorporates both these elements and as such the review will focus more on the network robotics side.

Brooks [3] put down the foundations for multiagent mobile robotics in his popularization of behaviour-based robotics in the late 1980s. His *subsumption architecture* works in stark contrast to traditional logic based artificial intelligence. It is a robust decentralized control architecture for any type of robot. Lower priority behaviours yield control of the robot to higher priority ones when they became active. Some very interesting examples of robot behaviour such as wall following and obstacle avoidance were demonstrated in real-time, a goal that had eluded logic based approaches for decades. Behaviour-based control provides a substrate for mobile robotics and naturally accommodates decentralized control which is distributed over multiple robots. Communication may be easily incorporated and is used by iRobot [7] which has been looking at behaviour-based control of groups of small mobile robots for military and other applications. They demonstrate similar types of tasks to the present work. Researchers at iRobot continue to use a descendent of Subsumption Architecture known as *Behaviour Language* for household and research robots.

Sojourner, the rover on NASA's 1998 Pathfinder mission to Mars, is the most successful robotic space explorer to date. This rover was a descendent of JPL's Rocky program which had been motivated by Brooks and others at MIT. Gat et al. [5] is a notable reference which looks at the direct application of behaviour-based control to mobile robots for planetary space exploration. Although Sojourner performed a large part of the mission through teleoperations, it was allowed to run autonomously using a behaviour-based controller towards the end of its operational lifetime.

Matarić [10] was among the first to focus on multiagent control of real robots. Her early work consisted mostly of developing elaborate control structures for groups of mobile robots by hand. Flocking, herding, following and other behaviours were demonstrated. This work outlined systematic approaches to programming single and groups of robots to work together. Later work has focused on learning group behaviours for about 4-10 robots. Other work deals with learning tightly coupled tasks such as to push a box towards a light [11] using two robots. Here communication is used in a sensory sharing capacity to make the task easier to solve.

Kube and Zhang [9] also work primarily in hardware. Their collective box pushing experiments are significant as the robots are able to accomplish a difficult task in a matter of seconds rather than hours (as some experiments require).

Robotic soccer [8] is an up and coming focal point in collective robotics research. RoboCup is the principal organization in this area. Stone [12] has dominated the simulator league in this competition. There are also several leagues involving real robots. Balch [1] also describes another soccer simulator, *Javabots*. Robotic soccer provides a common framework for collective robotics researchers from all over the world to directly compare their results.

Certainly collective robotics is being investigated from a number of very different perspectives. We have detailed only a small portion of the current work in this field but hope these examples are representative in their complexity and depth. This is very much an active research field whose popularity seems to be growing very quickly.

3 Task Requirements

Our focus has been on the development of control for a network of robots performing tasks representative of space exploration. Through a previous survey [2], we have arrived at a number of such tasks based on the needs of planetary scientists. The most relevant task is the deployment of an array of sensors in some predefined configuration to take simultaneous measurements at many locations. For example, a number of dipole units could be deployed over a region of a few hundred square kilometres to form a very-low frequency array on the Moon in order to perform radio astronomy. The number of units could be on the order of hundreds. These would be dispersed in a grid or cloud. An array of sensors enables scientists to collect distributions of data rather than a single point measurement. According to many planetary scientists, this will be the key to understanding the atmospheric and geophysical attributes of other planets. This paper will demonstrate the ability of our network of mobile robots to autonomously deploy themselves into a predefined configuration in the presence of obstacles and without the use of a centralized controller.

Another task of great interest is mapping a future landing site for human visits to other planets. Although this does not necessarily require a network of robots, the task can be greatly sped up by using

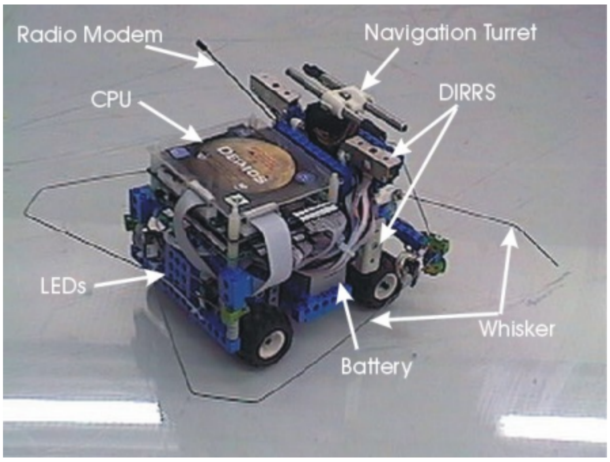

Figure 2: Detail of a single robot with features labelled.

multiple units. We are looking at methods of combining the maps built up by multiple robots but this will not be presented here. Also in this category is the collection of samples from a large area. A third task is deploying large pieces of equipment that are too heavy for a single robot to move (e.g., large solar panels). To this end we are investigating multirobot boxpushing [9, 11] in which robots must cooperate in order to move the box to some desired location. Again, this will not be presented here.

4 The RISE System

4.1 Sensors

Sensors were not chosen to facilitate a particular task but rather as a general suite appropriate to a planetary rover. Figure 2 shows locations of sensors on one of the robots. Four DIRRSs (Digital InfraRed Range Sensors) point forward and return a distance to objects up to 1 metre ahead of each robot. These were selected over sonar due to the possibility that such robots might be required to operate in a low or no atmosphere situation. We attempted to have our control methods not rely on any sensors that will not be available in a real space exploration mission. Furthermore, six simple whisker-style touch sensors (on/off) are located on the front, back and sides of each robot to sense collisions with obstacles.

4.2 Navigation

Onboard navigation is perhaps the most challenging aspect of mobile robot sensing. On Earth this problem has been circumvented to some degree with GPS

(Global Positioning Systems) but this is not available on other planets just yet. However, to use only inertial or odometry navigation invites massive errors in position. Some form of absolute positioning in a global coordinate system is required for the types of tasks mentioned above (assuming we care where, for example, an array of sensors is deployed). We use a landmark navigation system based on triangulation of position by observation of three lights. Each robot is equipped with a navigation turret which is able to detect the angles to each of these lights. Based on these angles a position is derived by numerically solving the trigonometric problem.

We feel that this system is more realistic than GPS (for space exploration) because each rover is computing its own position with its onboard resources rather than simply having its position provided from an external source. It fits with the autonomous philosophy which we have tried to maintain throughout this project. However, it does rely on knowing the coordinates of these lights a priori. The lights are meant to represent either human-placed beacons or natural landmarks whose positions are already known (e.g., stars or landscape features). One alternative to this system (which is not implemented) is for each robot to use its teammates as landmarks. This would provide relative positioning of the rovers but not an absolute offset of the group.

It takes between 1 and 2 seconds for a robot to obtain its global position using our navigation system. It must stop moving, sweep the navigation turret, and perform calculations. It is undesirable to do this with a high frequency and as such a second navigation system based on odometry (there is an encoder on the rear axle) is used to update the position in between uses of the global system. This local system provides fairly good data for a few metres of travel whereupon the global system is employed once more. The performance of our integrated navigation system is ± 0.15 metres or about one rover length. This means that our system is not capable of performing tasks requiring navigation better that this accuracy.

4.3 Communication

Each robot is equipped with a radio modem called a Radio Packet Controller (RPC) from Radiometrix. This allows the robots to communicate with each other and a base station desktop computer with finite bandwidth ($\approx$ 10 kbps). The range of these modems is approximately 30 metres. All broadcasts are heard by all modems (each robot must decide autonomously whether or not to make use of the incoming information). The robots communicate with the base station for two reasons. First, this enables us to issue high-level commands (such as start, stop, pause) which affect the entire group. This should not be viewed as centralized control as it does not necessarily mean that the robots will immediately obey the issued command (e.g., they might be in a self-preservation behaviour that has a higher priority than following orders). Second, it allows us to gather any data that is measured by the robots (both from their sensors and any potential payload instruments). In the event that a robot is out of range of the base station, our network protocols allow messages to be relayed via another robot (or a long chain of robots) to the base station. Thus, the communication range of the entire system may be extended greatly beyond the limits of a single robot. This provides new possibilities for applications in and of itself; we could automatically establish a communication link between two points that would otherwise be out of contact (either due to distance or obstacles) by appropriately positioning the robots.

The robots communicate with each other to share information about their environment and thus make it possible (or easier) to solve cooperative tasks. For example, if three robots are to form a triangle they need to know the positions of the other robots. This can be accomplished through sophisticated local sensors (e.g., vision) or much more simply through communication [11]. Our robots broadcast a 'ping' at a rate of approximately 1 Hz. This ping is heard by both the base station and the other robots. It typically consists of an identification number, coordination data, and position data as well as other sensor data although the exact details vary from task to task. Any robot which listens may make use of the data in the ping; robots do not broadcast to a specific receiver. More lengthy communications also occur but these are typically between the base station and a particular robot (e.g., to download a map that the robot has been building).

It should be pointed out that the base station does not provide any help in terms of solving the task. In fact, the network of robots does not require it to run at all. As with the control, the communication has been designed in a completely autonomous framework. Robots can join or leave the group and the system will continue to function as the ping broadcasts do not require acknowledgement from the other robots.

4.4 Actuators

Many of the electro-mechanical components have been constructed using LEGO Technic^{TM} pieces.

LEGO^{TM} was used to enable rapid prototyping. Each robot has two DC motors, one to drive the back axle through an open differential and one to drive the front axle to enable four-wheel drive. There is a passive suspension system that enables all four wheels to remain in contact with the ground on slightly curved surfaces. The steering and navigation turret are driven by Airtronics model airplane servomotors. The steering linkage has some 'play' in it which makes odometry unreliable past a few metres of travel.

The computational facilities of each robot consist of a microcontroller, Infineon $C164$, with 1 MB of RAM and 1 MB of flash-ROM (for program storage). The processor runs at 20 MHz. In additional, an Amtel $89C52$ handles all low level hardware routines. The two processors communicate via a dual-port RAM (10 Kb). Interface electronics were designed and assembled in-house. The total power draw of the electronics is under 400 mA. Power is a major factor to consider for space systems which partly guided the choice to use behaviour-based control as it does not require huge computational resources.

The battery of each robot is a custom package of 8 NiMH (Nickel Metal Hydride) cells, size AA. The capacity of our batteries is about 1600 mAh. Starting with fresh batteries a robot can run from 2 to 3 hours.

5 Control

To allow each robot to act independently with limited resources virtually dictates using some form of behaviour-based control. Traditional control methods as well as traditional artificial intelligence methods are either too computationally intensive, unable to function in real time, or rely on complex models of the robot's environment which must be known a priori. Here we are in the business of exploration by a robot which is far away with limited power and mass. A variant of behaviour-based control was used.

5.1 Architecture

The structure of the behaviour language used is very similar to Subsumption Architecture [3] in that the system has prioritized *layers*. The system may be built up one layer at a time without affecting earlier work. There is a root (highest priority) with many layers below. In our implementation all layers are running on the same processor but they could be split over multiple processors with minimal communication. Figure 3 shows the qualitative structure of this architecture. Each layer has a flag which is set high when some condition has been met. The highest priority layer with a flag set high is the *active layer*. The actuator outputs from the active layer are the ones that affect the robot. Within each layer we use a linear program flow. There are alternate paths (subbehaviours) and counters (for time and distance). When this short program is completed, the behaviour is reset.

5.2 Example

Here we describe our instance of the above behaviour-based architecture used to deploy an array of sensors. The layers from highest priority may roughly be described as follows:

root Serves as entry point to control algorithm

get data Causes robot to read data in communications buffer when it is present

send data Causes robot to send a 'ping' after a certain amount of time

search all Causes robot to seek other robots when completely out of communication (not implemented)

search base Causes robot to look for the base station when unable to communicate with it (not implemented)

override Allows base station to act as a centralized control in an emergency only

seek power Causes robot to seek energy when battery is low (not implemented but does send a message back to the base station instead)

get position Causes robot to perform a global navigation sweep after a certain distance has been travelled or time expended

complete Causes robot to notify the base station when task has been completed

unstick Monitors the odometer progress of the robot over time and does a random manoeuvre if no progress has been made for a certain time

avoid obstacle Detects obstacles through DIRRSs/whiskers and performs an appropriate manoeuvre to get obstacle out of path of travel

seek goal Causes robot to drive towards the center of a large circle when far away from that circle (circle specified a priori)

avoid rover Causes robot to move away from neighbouring robots when they are too close together (requires communication of positions between rovers)

stop Stops the robot

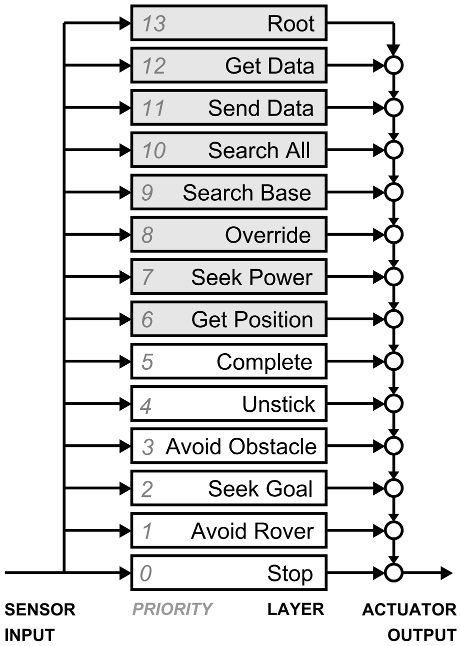

Figure 3: Instance of behaviour-based architecture used to *deploy an array of sensors*. Shaded behaviours are task-independent while unshaded behaviours are specific to the sensor array task.

5.3 Embodiment

It could be argued that some the algorithms used in this work could have been tested in simulations rather than hardware. In fact, the entire control structure was originally developed in a simulator before being fine tuned on the robots. As part of the RISE project, a simulator was developed to facilitate rapid development of network robotics control structures [4]. The simulator was extremely helpful in the development of most of the layers in the above example. The exact same control code can be run in the simulator and on the real robots. The key differences (which made it necessary to actually try everything in hardware) was the communication (e.g., finite bandwidth, lost messages), hard to model disturbances in the navigation (e.g., robots blocking each other from viewing the lights), cone-shaped view of DIRRSs, and blind spots in whisker sensors. In the end, most of the nonidealities of the hardware did not require major modifications to the control code which attests to the robustness of the controllers. There were several iterations of fine tuning to be done on the hardware but there would have been many more without the simulator. We believe highly in validating controllers through *embodiment*.

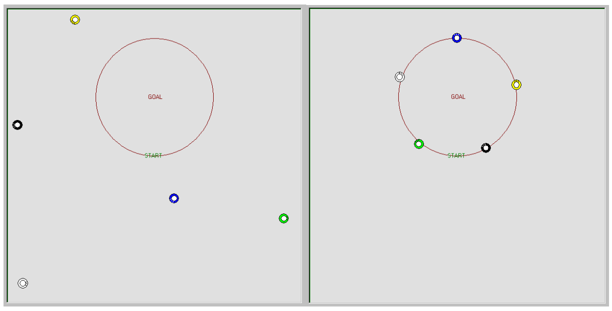

Figure 4: Task representing deploying an array of sensors. (left) Random initial condition. (right) Desired final condition. Taken from RISE simulator.

6 Experimental Results

We describe our preliminary findings on the task of deploying an array of sensors. Our representation of this task is to require that from random locations in a large space, the robots should equally distribute themselves along the circumference of a circle (another shape could be substituted) in the presence of obstacles. Figure 4 depicts the sensor array task. The robots are told where this circle should be in a global reference frame. They are not able to sense the circle of tape on the floor (i.e., it exists to allow performance assessment only by the experimenter). The lights used for navigation are along one side of the workspace. The obstacles are large flat rocks (the robots can 'see' the lights above the rocks). If taller rocks are used the navigation is affected adversely (as not all three lights are always visible) and the total performance of the system worsens.

Figures 5 and 6 show, through a sequence of photographs, 5 robots performing the sensor array task both with and without obstacles. This was tried with groups of 2 to 5 robots[1] (with no changes in the code at all) and in each case the robots did manage to spread out as desired. In the case of 5 robots, we have made about 50 runs and in most cases the robots were highly successful (i.e., sometimes the pentagon was slightly irregular). In a few cases the pentagon was very irregular. Most cases took about 1 or 2 minutes to arrive at the final configuration.

7 Discussion

We have tried to build a realistic concept network of autonomous planetary rovers. It is difficult to represent a real space navigation system in a lab situation. Beacon lights in a real system would be difficult to

[1] Although we have 6 robots in total, the sixth robot, Thanatos, was not functioning at the time of writing.

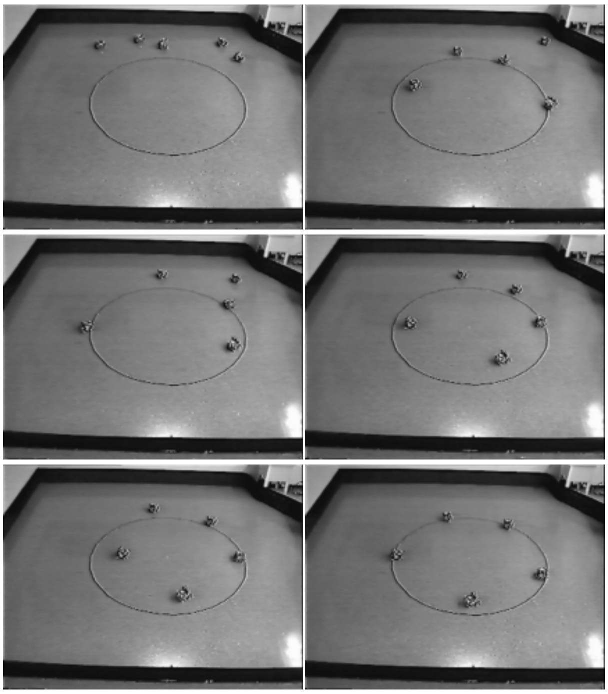

Figure 5: Sequence of photographs showing 5 robots forming a pentagonal array (with no obstacles).

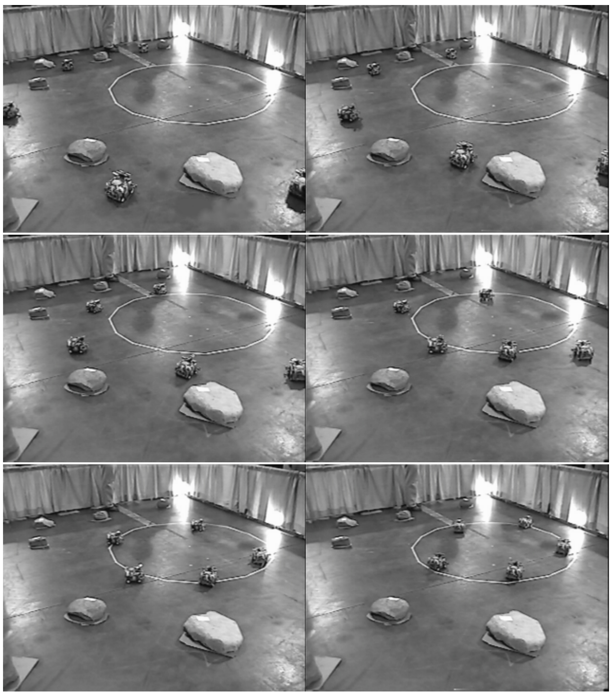

Figure 6: Sequence of photographs showing 5 robots forming a pentagonal array (with obstacles).

deploy and in the case of spreading the robots out over a very large area (as would likely be required for a real network space science task) it would be impractical. However, the control structure we have developed is independent of how a robot's coordinates are obtained. As discussed above, we wanted to put as much of the burden as possible on the robots rather than on external mechanisms (such as GPS).

The sophistication of the robots themselves is fairly low. Each robot cost less than $2000 (Cdn) in parts. They have relatively few sensors yet complex behaviours have been built up through behaviour-based control. A real space-qualified system would have better and great numbers of local sensors and probably higher bandwidth communications. We feel this would only make the system better although the development of the control might be more involved to make use of the added information.

Scaling is always an important issue in multiagent research. How will the system function as more units are added? Because we have made each robot autonomous, the addition of robots is easy (we showed the very same code to work with 2, 3, 4, or 5 robots). The limiting factor in our system will certainly be the communications bandwidth. If too many robots are trying to broadcast at once, many messages will be lost and although this will not in itself stall the system (since broadcasts do not require acknowledgement), performance will be affected. A truly scalable system would function under sparse (local) communications between robots [7, 13]. In the present network, this only occurs if the robots are quite far apart (about 30 metres). However, with 5 robots we are not facing difficulties yet.

All network robotics researchers would probably agree that the sheer amount of maintenance required by such an experiment can be overwhelming. It is often difficult to have all robots functioning simultaneously. One of the major contributing factors to this phenomena is the power system. Batteries discharge at different levels and thus must be replaced. Although we are able to monitor all power levels remotely from the base station, it is still necessary to physically swap the old batteries with the new. This greatly reduces the autonomy of the experiment. A major improvement would be to allow the rovers to charge themselves either through solar panels (which might take a long time) or a charging station. We are currently investigating these possibilities to make the experiment conform more to the 'robotics in a glass box' philosophy.

Two other changes with this same philosophy in mind are the ability to download new programs over the already present radio network and to be able to control the system over the internet. Right now we must download new code through a hard serial link, one robot at a time. Together these changes would ideally allow us to reprogram and command the entire network from any internet browser. Different experiments could be run without any on site maintenance.

8 Conclusion

We have described our vision and development of a new breed of space explorer. With relatively simple hardware (e.g., LEGO^{TM}, model airplace servos, radio modems, 20 MHz computer) we are able to accomplish a cooperative task involving 5 robots. This task is representative of the deployment of an array of sensors. Such arrays are required to accomplish network science, a future trend in planetary space science.

Space exploration necessitates autonomy. Our design of behaviour-based controllers for networks of robots have been shown to be successful, robust, and portable. We have attempted to adhere to the 'robotics in a glass box' philosophy but found improvements could be made in terms of downloading new programs and replenishing power. Future generations of our robots will attempt to address these issues. Only when mobile robotics systems are able to function without maintenance or supervision on a long term basis will they truly be autonomous.

Acknowledgements

We would like to acknowledge support of the RISE project by CRESTech (Centre for Earth and Space Technology), NSERC (Natural Science and Engineering Research Council), CSA (Canadian Space Agency), MD Robotics (MacDonald Dettwiler Space and Advanced Robotics Ltd.), Dynacon Enterprises Ltd. We would also like to thank Thierry Cherpillod for the design and construction of the embedded electronics and Lawrence Lok, Klementyna Niewiadomska, Stephen Chan, and Jordan Shapiro for help in the construction of the robots.

References

[1] T. Balch. Learning roles: Behavioural diversity in robot teams. In *AAAI Workshop on Multiagent Learning*, 1997.

[2] T. D. Barfoot, T. Cherpillod, E. J. P. Earon, and G. M. T. D'Eleuterio. Network science and network robotics: A brief survey of possibilities for planetary science. Utias technical report, University of Toronto Institute for Aerospace Studies, 1999. Prepared for the Canadian Space Agency.

[3] R. A. Brooks. A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Autonomous Systems*, RA-2:14–23, 1986.

[4] E. J. P. Earon, T. D. Barfoot, and G. M. T. D'Eleuterio. Development of a multiagent robotic system with application to space exploration. Como, Italy, 2001. Advanced Intelligent Mechatronics (AIM01). To appear.

[5] E. Gat, R. Desai, R. Ivlev, J. Loch, and D. Miller. Behaviour control for robotic exploration of planetary surfaces. *IEEE Transactions on Robotics and Automation*, 10(4):490–503, August 1994.

[6] InternationalSpaceUniversity. International lunar farside observatory and science station. WWW, 1993. http://exploration.jsc.nasa.gov/explore/Data/Lib/DOCS/EIC039.HTML.

[7] iRobot. Swarm: Distributed programming of autonomous robots. WWW, 2000. http://www.irobot.com/rd/research_swarm.htm.

[8] H. Kitano, M. Asada, Y. Kuniyoshi, I. Noda, and E. Osawa. Robocup: The robot world cup initiative. In *Proceedings of the First International Conference on Autonomous Agents*, Marina del Ray, 1997. ACM Press.

[9] R. Kube and H. Zhang. Collective robotic intelligence. In J. A. Meyer, H. Roitblat, and S. W. Wilson, editors, *Simulation of Adaptive Behaviour: from Animals to Animats 2*. MIT Press, 1992.

[10] M. J. Matarić. Designing emergent behaviours: From local interactions to collective intelligence. In J. A. Meyer, H. Roitblat, and S. W. Wilson, editors, *Simulation of Adaptive Behaviour: from Animals to Animats 2*. MIT Press, 1992.

[11] M. J. Matarić. Using communication to reduce locality in distributed multi-agent learning. *Journal of Experimental and Theoretical AI, special issue on Learning in DAI Systems*, 10(3):357–369, 1998.

[12] P. Stone. *Layered Learning in Multiagent Systems: A Winning Approach to Robotic Soccer*. MIT Press, Cambridge, Massachusetts, 2000.

[13] R. A. Watson, S. G. Ficici, and J. B. Pollack. Embodied evolution: Embodying an evolutionary algorithm in a population of robots. In *Congress on Evolutionary Computation*, Washington D C, July 6-9 1999.