Robotic Self-Repair in a Semi-Structured Environment

Wooram Park, David Albright, Charles Eddleston, Wai K. Won, Kiju Lee, G.S. Chirikjian*

Department of Mechanical Engineering
Johns Hopkins University
Baltimore, Maryland, USA

**Email: gregc@jhu.edu*

***Abstract* – The ability of natural organisms to self-assemble, self-repair and reproduce in an environment with sufficient nutrients is one of the defining features of life. In this paper, we build on both our own previous work and that of others to demonstrate the feasibility of robotic systems that can assemble functional copies of themselves from either basic sets of parts or from incomplete replicas that are not functional. Robots capable of self-repair (particularly using in situ resources) will have a profound impact on the way lunar and planetary surfaces are transformed for human use during exploration and colonization. We demonstrate concepts in robotic self-repair and self-assembly using parts from LEGO Mindstorm kits together with a patterned planar environment consisting of five stations. Each station has a location code and contains one of five robot subsystems. Four subsystem may be located at any of the four stations and the base unit is located at the center of the main track (and it can be placed in any of four distinct orientations, but is positioned at the center of the track), indicating $4 \times 4!$ possible layouts. The design we demonstrate is robust to these permutations, i.e., the functional robot can assemble (or ``repair'') a copy of itself by visiting each station and executing a sequence of behaviors that accounts for all possible permutations. This demonstrates a step in the direction of robotic self-repair and self-replication in unstructured environments, and represents a departure from previous efforts at JHU that were concerned with autonomous self-replication in completely structured environments.**

Index Terms – robotic self-repair, self-assembly, self-replication

I. INTRODUCTION

This paper presents an autonomous self-repairing robot using a LEGO RCX controller. The design is based on a line tracking algorithm, thus the robot simply follows a black line/track and recognizes the replica's subsystems placed on the track. To identify each subsystem, we implement a barcode system. The barcode is placed on the track, so that the robot can read it during tracking the black line. The robot consists of five different subsystems: base, right wheel, left wheel, fork and tail. The base can be considered as a ``broken'' version of the original robot, which is to be ``repaired'' by attaching the four other subsystems. All parts are designed to be small and light so that the vehicle can carry the parts easily. In addition, the parts each have their own functions and characteristics, which make our model very different than self-reconfigurable modular robots studied in the literature. In the remainder of this section, we review some previous efforts related to mechanical self-replicating systems. In Sections II and III, we present the software and hardware design of our prototype robot. Our conclusions are presented in Section IV.

Related Works

The first theoretical study of self-replicating machines was performed by John Von Neumann [6]. He postulated that it must be possible to design a machine that would be able to build another machine just like itself. Although his ideas about self-replicating systems were never implemented as a macro-scale kinematic machine, they have been applied in several research areas such as: cellular automata, macromolecular chemistry, nanotechnology, and computer viruses [4].

Penrose, in [5], presented the first implementation of a passive self-replicating mechanical system. His idea was to design and to construct simple units or bricks with such properties that a self-reproducing machine could be built out of them. He demonstrated the assembly of passive elements under external vibration. This however, did not have any actuation or computational intelligence, and cannot be considered a robot.

After a few decades, NASA became interested in self-replicating robots as a potential means for space development [7]. Interest in this concept has been revived in recently with the long-term goal of self-replicating factories on the moon ([8], [9]). In contrast, our current work is focused on the nearer-term goals of robotic self-repair using in-situ resources.

Since the 1990's, interest in self-reconfigurable modular systems has been increasing ([2], [11], [12], [13], [25]). Algorithms for self-assembly using modular robots have been developed by Murata, Rus, and Butler [19], [20]. Using self-reconfigurable modular robots, Tomita ([16]) and Yim ([17], [23]) discussed the repair capabilities of a self-reconfigurable system. In a different context, Russel and Kime described the self-diagnosis and self-repairable system in [21] and [22].

Moreover, Bererton and Khosla presented a repairable team of robots in [14] and [15]. Their work is fundamentally different than Tomita's and Yim's in that the robot can be repaired by other robots in a team, not repaired by itself. A substantial body of literature is also developing in the area of self-assembling circuits and devices at the nano and micro scales. D. Gracias and K. Bohringer have been concerned with the micro-scale self-assembly and micro-structures ([26], [27]). E. Klavins has developed graph grammars which can be used to replicate randomly labeled strands of particles [28]. J. Lohn has investigated evolvable systems in which computer algorithms can design and optimize the systems automatically for space missions [29]. Whereas the goal in these works is similar in that a functional copy of a device is constructed without human intervention, the physical implementations are more reminiscent of Penrose's work (i.e., relying on forces of nature and passive components) rather than anything that would be considered an active autonomous self-replicating or self-repairing robot.

Recently, our lab has built several prototypes of self-replicating systems in order to develop the concept. As the first step, we constructed remote-controlled [1] and semi-autonomous self-replicating systems [18]. These works, the robot consists of several prefabricated subsystems and the controller. We extended this work to consider a fully autonomous system ([3]), and it demonstrated the feasibility of self-replicating mechanical systems. In addition, a self-replicating, electromechanical circuit was presented in [10]. The circuit uses an electromechanical device as a substrate in order to construct functional copies of itself. Through these works, we have proved that robotic self-replication is no longer science fiction, and have given form to abstract ideas circulating in the literature for fifty years. As current research, our lab is working toward autonomous self-replicating systems in unstructured environments. That means the robot is completely autonomous and independent from human control (including the step of structuring the environment). Robotic self-repair in a partially structured environment is one step in the evolution of this concept.

II. MECHANICAL DESIGN FOR SELF-REPLICATING AND SELF-REPAIRING SYSTEMS

A. Barcode scheme

Our design involves implementing a barcode system to identify each part of the system. We tried several different approaches to attach the barcode to each subsystem. The final design is concerned with a barcode on the track instead of placing it on the body of each subsystem. The base (RCX controller unit) is placed at the center of the main track. At each side of the base where the additional parts are supposed to go, a line is extended out to the main track. On the main track, the same barcode scheme is used to differentiate between the four parts that need to be attached to the base. Fig. 1 shows a base unit and a part of the track on which the barcode is placed.

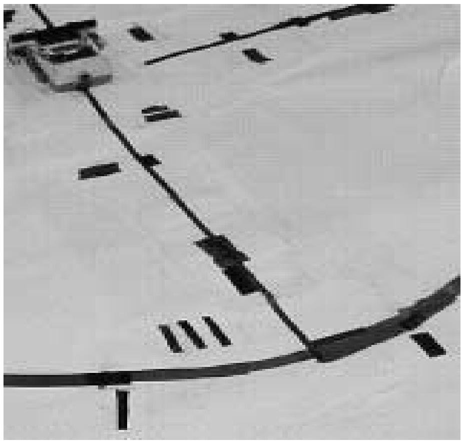

Fig. 1. Barcode scheme used inside the main track

In addition to this barcode scheme, a new bar code scheme is also implemented since the paths leading to the base need to be distinguished. This is due to the fact that the functional (original) robot can basically start anywhere on the main track. In order to solve this problem, a new bar code scheme was implemented. We constructed each sub-track to place the subsystems and used a new barcode scheme. The new barcode scheme is basically the reverse of the barcode scheme used to pick up the parts. Instead of starting the barcode counter when the left sensor sees a black line, this scheme starts when the right sensor sees a black line. Then, it increments the counter by one each time the left sensor picks up a black line. It finally ends counting when the left sensor picks up another black line.

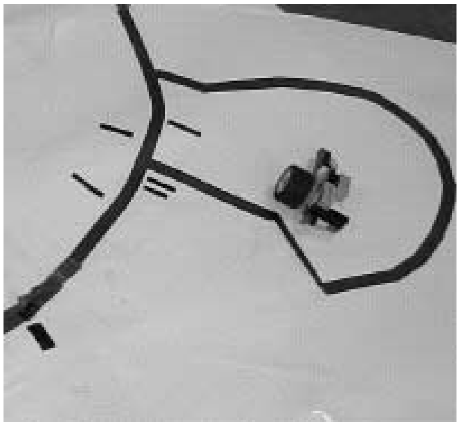

Fig. 2. Barcode scheme on the sub-track

These two schemes are fundamentally the same, but it is able to turn and place the parts at the right angles, resulting in consistent part placement. When we put all of these components together, a new, working robot is created that falls within all of the restrictions set up. The car can start out anywhere on the main track. When it has finished delivering all of the parts, the robot will simply follow the track until it runs out of batteries or is turned off.

B. Software design

The programming is accomplished by using the BricX programming environment that allows much more flexibility in the design. It means that we are able to modify the software easily when the system encounters any problem, rather than changing the hardware design, which is sometimes very hard to modify. Since the Lego Mindstorm programming environment is not very reliable, a C compiler specifically designed for Lego Mindstorms set was used. This C compiler is called NQC, Not Quite C, and was developed by David Baum. In addition, a GUI was also used in order to make the programming of the RCX controller much easier.

The basic line-tracking algorithm is rather simple. Among three sensors used in the system, it uses the second sensor, which is placed at the center of the robot, to track the line. If the second sensor detects black, the robot starts to turn right until second sensor sees white, and if it sees white, the robot will start to turn left until it sees black. This algorithm is placed within an infinite loop so that it will continuously follow the inner lane of the track. When the second sensor tries to determine whether it is white or dark, it uses a threshold constant to distinguish between the two. The sensor is constantly taking readings. It classifies brightness on a scale of 0 to 100 in which 100 would be the brightest. The threshold constant basically sets up the cut off point for being bright. Anything above the threshold will be considered white, and anything below it will be considered black. When turning left, the robot simply turns on the right motor and moves it in the forward direction for 0.05 seconds. After that, it turns off the motors and takes another light reading to determine what to do next.

The robot reads the barcodes using two other sensors (left and right sensors). The main algorithm makes heavy use of line tracking, because it detects the black from the white board and identifies the routine from the barcode. Since we used the light sensors from the LEGO kit and they are not very sensitive to recognize any black line, we had to perform several trials to make the adjust width of the barcodes. After this, we made barcodes that are readable for the robot.

C. *Hardware Design*

Our vehicle consists of five parts; base, right wheel, left wheel, fork and tail. All parts are designed to be small and light so that the vehicle can carry the parts easily. In addition, the parts have their own functions and characteristics (Fig. 3).

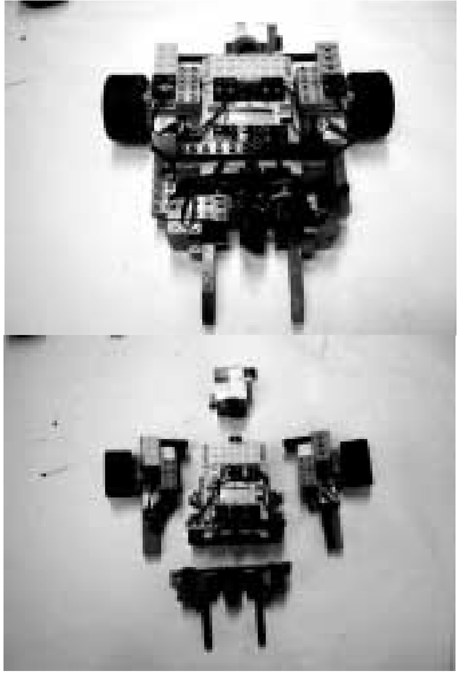

Fig. 3. The vehicle assembled (top) and the vehicle disassembled (bottom).

The base is the RCX controller. The base includes the magnets for mechanical connection and the terminals for electrical connections. We put four ball bearings underneath the base. They enable the vehicle to move with low friction. Wheels include a motor, the magnets for mechanical connection and the terminals for electrical connections. The tail adds weight to the back of the car, so that when the car is carrying the other pieces it does not tip up at all. The added weight also increases friction between the ground and the wheel. Thus, it also helps the vehicle go straight and turn without the wheels slipping. In addition, the fork is designed as a subsystem to allow the robot to push and carry parts with it. We put three sensors on the fork, and the second sensor is for tracking the line and the others are for counting the barcodes.

As connections between subsystems, we consider two methods; electrical connections and mechanical connections. One part of the electrical connection consists of a wire, a steel plate, a metal spring, copper sheet and aluminium wire, and the other part consists of a wire and a steel plate (Fig. 4). Each element is soldered to the adjacent elements. Here the copper spring occupies the gap between two steel plates. Since a spring has flexibility, even if the gap gets wider or narrower a little bit, the copper spring guarantees the electrical connection. Furthermore we winded the aluminium wire at the terminal and it eliminates the accidental connection failure. Each wheel has two electrical connections to the base. The fork has 6 electrical connections to the base.

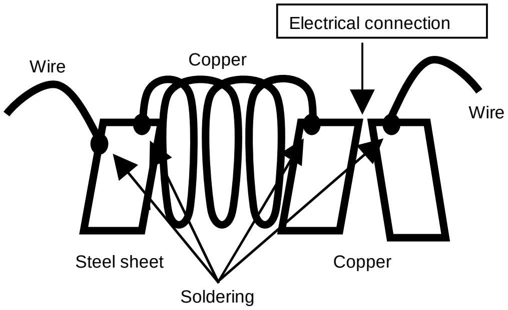

Fig. 4. A diagram explaining the electrical connections

We used two pairs of magnets for one mechanical connection. If we use one pair, the part can turn freely and the structure is not rigid. In using two pairs of magnets, we put two magnets in a row on one side so that two front magnetic poles of the magnets are opposite. It prevents sliding and improves the rigidity of the structure. As seen in Fig.5, the special configuration of the magnets causes the arrangement along the tangential direction of the faces of the parts as well as the normal directional attraction.

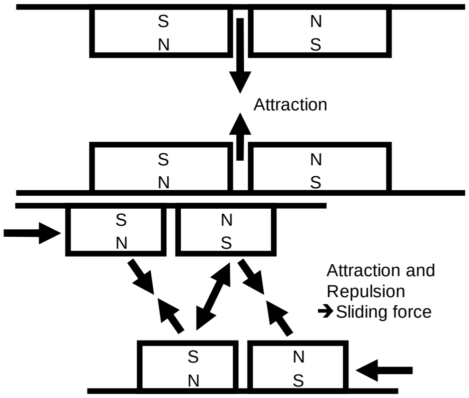

Fig. 5. Schematic explanation of mechanical connection

III. DISCUSSION AND FUTURE WORK

The robot presented here is able to construct a functional copy of itself with no external control. The subsystems are still prefabricated and the function of the robot is mostly oriented to the RCX unit. The accuracy and stability are also problems to be solved. Since we are focusing on development of systems to demonstrate concepts leading toward the eventual application of lunar development, we have tried to build the system as simply as possible. In the real application, this philosophy will have the effect of minimizing the mass of materials that have to be sent from the earth.

Our next step is to make subsystems having similar complexity not only mechanically, but also in electrically. We plan to use the electric circuits with relays and transistors instead of the RCX controller, so we can make separate circuits for each subsystem. In addition, we will consider a new method to identify each subsystem.

IV. CONCLUSION

We presented an autonomous self-repairing robot using a LEGO RCX controller. The design was based on a line tracking algorithm. Therefore the robot simply follows a black line and recognizes the replica's subsystems placed on the track. To identify each subsystem (which is necessary due to the fact that subsystem locations can be permutated without this knowledge being stored in the robot's memory), we implemented a barcode system placed on the track. The robot consists of five different subsystems: base, right wheel, left wheel, gripper and tail. Through this work and other previous efforts, we have demonstrated the feasibility of robotic systems capable of self-replication and self-repair in a rather structured laboratory setting using `toy' models. After developing these concepts further, our longer term goal is to design and construct `real' robotic systems that demonstrate these capabilities, as well as processing of simulated lunar regolith for use in construction of components used in self-repair.

ACKNOWLEDGMENTS

This work was performed under a NASA NIAC Phase I award during the Spring of 2004. The authors were students in Chirikjian's Mechatronics course in which this work was performed as a class project.

REFERENCES

[1] G.S. Chirikjian, and J. Suthakorn, " Toward Self-replicating robots," *Proceedings of the Eight International Symposium on Experimeental Robotics(ISER)*, Italy, July 2002.

[2] G.S. Chirikjian, A. Pamecha, and I. Ebert-Uphoff, "Evaluating efficiency of self-reconfiguration in a class of modular robots," *Journal of Robotic Systems*, vol. 13, no. 5, 1996, pp.317-338.

[3] J. Suthakorn, A.B. Cushing, and G.S. Chirikjian, "An autonomous self-replicating robotic system," Proc.

[4] Sipper, M., "Fifty years of research on self-replication: An overview," *Artificial Life*, 4(3), 1998, pp.237-257.

[5] Penrose, L.S., "Self-reproducing machines," *Scientific American*, Vol. 200, No. 6, 1959, pp 105-114.

[6] J.V. Neumann, A.W. Burks, "Theory of self-reproducing automata," *University of Illinois Press*, 1962

[7] R.A. Freitas, Jr., and W.P. Gilbreath, (Eds.), "Advanced automation for space missions," Proceedings of the 1980 NASA/ASEE summer study, Chapter 5: Replicating systems concepts: Self-replicating lunar factory and demonstration, NASA, Scientific and Technical Information Branch (Conference Publication 2255), Washington, DC: US government printing office, 1982

[8] V.M. Chadeev, "Allowance for robot self-replication in industrial automation," *Automation and Remote Control*, vol. 61, 2000, pp. 1752-1757.

[9] G. Friedman, "Self-replication technology for the space solar power mission," workshop presentation for the Joint NASA/NSF Workshop on Autonomous Construction and Manufacturing for Space Electrical Power Systems, April 2002.

[10] W.A. Hastings, M. Labarre, A. Viswanathan, S. Lee, D. Sparks, T. Tran, J. Nolin, R. Curry, M. David, S. Huang, J. Shuthakorn, Y. Zhou, G.S. Chirikjian, "A minimalist parts manipulation system for a self-replicating electromechanical circuit," IMG'04, Genoa Italy, July 1-2, 2004.

[11] T. Fukuda, S. Nakagawa, F. Hara, "Dynamic distributed knowledge system in self-organizing robotic systems: CEBOT," *Proc. IEEE Intl. Conf. On Robotics and Automations*, Sacramento, 1991, pp. 1908-1913.

[12] T. Fukuda, Y. Kawauchi, "Cellular robotic system(CEBOT) as one of the realization of self-organizing intelligent universal manipulator," *Proc. IEEE Intl. Conf. On Robotics and Automations*, Cincinnati, May 1990, pp. 662-667.

[13] K. Hosokawa, T. Fujii, H. Kaetsu, H. Asama, Y. Kuroda, I. Endo, "Self-organizing collective robots with morphogenesis in a vertical plane," *JSME International Journal Series C-Mechanical Systems Machine Elements and Manufacturing*, vol. 42, no. 1, 1999, pp. 195-202.

[14] C. Berenton, P.K. Khosla, "Toward a team of robots with repair capabilities: A visual docking system," ISER, 2000

[15] C. Berenton, P.K. Khosla, "An analysis of cooperative repair capabilities in a team of robots," ICRA, 2002

[16] K. Tomita, S. Murata, H. Kurakawa, E. Yoshida, and S. Kokaji, " Self-assembly and self-repair method for a distributed mechanical system", *IEEE Transaction on Robotics and Automation*, 15:1035-1045, 1999.

[17] M. Yim, D. Duff, and K. Rufas, "PolyBot: A modular reconfigurable robot", *IEEE Intl. Conf. Robotics and Automation*, vol. 1, 2000, pp. 514-520.

[18] J. Suthakorn, Y. Kwan, and G.S. Chirikjian, "A semi-autonomous replicating robotic system," *Proc. 2003 IEEE International Conference on Intelligent Robots and Applications (CIRA)*, Kobe, Japan, July 2003, pp. 776-781.

[19] S. Murata, H. Kurokawa, and S. Kokaji, "Self-assembling machine," *Proceedings of the IEEE International conference on Robotics and Automation*, SanDiego, CA, 1994, pp. 441-448.

[20] Z. Butler, S. Murata, and D. Rus, "Distributed replication algorithm for self-reconfiguring modular robots", *Proc. 6th Intl. Symposium on Distributed Autonomous Robotic Systems (DARS '02)*, Fukuda, Japan, June 2002, pp. 25-27.

[21] J.D. Russel, C.R. Kime, "System fault diagnosis: closure and diagnosability with repair", *IEEE Trans. on Computers*, vol. 24, no. 11, 1975, pp. 1078-1089.

[22] J.D. Russel, C.R. Kime, "System fault diagnosis: masking, exposure, and diagnosability without repair", *IEEE Trans. on Computers*, vol. 24, no. 12, 1975, pp. 1155-1161.

[23] M. Yim, Y. Zhang, J. Lamping, E. Mao, "Distributed control for 3D metamorphosis", Autonomous Robots, vol. 10, 2001, pp. 41-56.

[24] K. Kotay, Daniela Rus, M. Vona, C. McGray, "The self-reconfiguring robotic molecule", *Proceedings of the IEEE International conference on Robotics and Automation*, vol. 1, 1998, pp. 424-431.

[25] Castano, A., A. Behar, P. Will, The CONRO modules for self-reconfigurable robots, IEEE Transactions on Mechatronics, 7(4), 403-409, Dec. 2002.

[26] D.H. Gracias, I. Choi, M. Weck, G.M. Whitesides, "Mesoscale self-assembly", *Algorithmic and Computational Robotics*: *New Direction*, eds, B. Donald, K. Lynch and D. Rus, A.K. peters, Ltd (2001)

[27] K.F. Bohringer, "Surface modification and modulation in microstructures: controlling protein adsorption, monolayer desorption, and micro-self-assembly", *IOP Journal of Micromechanics and Microengineering (JMM)*, 13(4): S1-S10, July 2003.

[28] E. Klavins, "Universal self-replication using graph grammars," *International Conference on MEMs, NANO and Smart Systems*, Banff, Canada, 2004.

[29] J. Lohn, J. Crawford, A. Globus, G. Hornby, W. Kraus, G. Larchev, A. Pryor, D. Sriviastava, "Evolvable systems for space applications", *International Conference on Space Mission Challenges for Information Technology (SMC-IT)*, 2003.