Complete Process for Target Form Self-Replication from Homogeneous Self-Reconfigurable Robot Modules

Jacob Everist, Wei-Min Shen
Information Sciences Institute
University of Southern California
4676 Admiralty Way, Suite 1001
Marina del Rey, CA 90292
everist@usc.edu

Abstract

We describe a complete system for the autonomous self-replication of multiple instances of an user-defined target robot form using self-reconfigurable robots. We describe the tools that are required, the steps of the assembly process, and the work required to realize a working system. We show how this system is practical for a real self-reconfigurable robot system. We give some experimental results that show the feasibility of the system and describe future work required to give a complete demonstration.

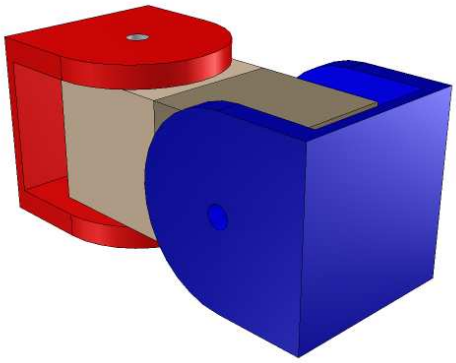

Figure 1: Superbot - Single Module

Introduction

We seek a way to build hundreds of instances of an arbitrary, user-specified robot morphology. One way to do this is with the use of self-reconfigurable robots which are a sort of building-block system for robots. These modules can be composed together to create arbitrary robot configurations suitable for solving different types of tasks. For instance, out of the same set of modules, one could build a snake, a 4-legged walker, or a digging machine. The property of being self-reconfigurable means that these robots are capable of shape-changing on their own. The system we use is the Superbot (Shen et al., 2006) and can be seen in Figure 1 and 2.

Currently, there is no systematic process for realizing the full potential of self-reconfigurable robot systems when working on the order of thousands of robot modules. The desire to create hundreds of copies of a target robot morphology force us to specify the details of a solution. We must seek a complete end-to-end process using the lessons learned from robot self-replication and self-assembly experiments. Seminal work on self-reproduction can be found in (von Neumann, 1966) and a large survey of self-replication can be found in (Freitas Jr. and Merkle, 2004).

This work is important because a full system that is capable of manufacturing hundreds of robots from the substrate of thousands of homogeneous self-reconfigurable robot modules will validate the usefulness of self-reconfigurable robots for very large autonomous operations. This would be the first large-scale application of the concepts of self-replication in a working system. We also demonstrate a viable substrate for macro-scale self-replication experiments outside of computer simulators.

The other important contribution of this work is that we show how the concepts of self-replication of stochastic systems in micro-scale and nano-scale environments can be used in a macro-scale environment. We augment the existing general theories from chemical-based self-replicating systems and scale them up to work in a larger, slower, less stochastic environments.

This is a hard problem because there is no pre-existing workable framework or plan for realizing large-scale self-replication in a macroscopic system. There currently exist only theories and models of how they would work in real-life. Our process is a blueprint for creating a real working system in a robot system that is currently being built. We derive part of our lessons from a previous plan for large-scale self-assembly in (Shen et al., 2003) that received partial implementation in (Everist et al., 2004).

Chemical or nano-scale self-replication environments also make large use of the inherent noisy random environment forces that dominate at that scale and make for quick operations over a short period of time. These concepts need to be transfered and modified up to a macro-scale environment that does not have the same scale and speed of random

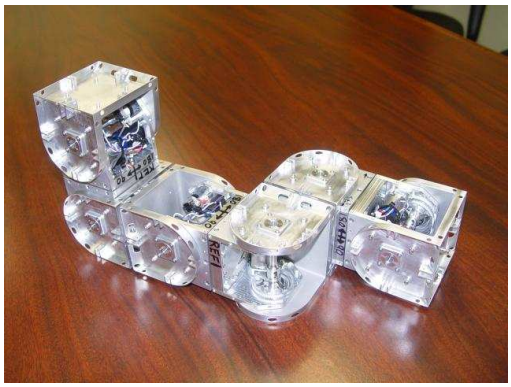

Figure 2: Superbot - 3-Module Configuration

motion.

Other robot systems that have accomplished robot self-reproduction on small scales are enzyme-based replication (Rubenstein et al., 2004a), and self-reproducing machines (Zykov et al., 2005). These two systems are both self-reconfigurable modular robots. Another is the pure self-replicating robot that assembles a copy of itself by (Suthakorn et al., 2003). Robot systems that have accomplished robot self-assembly are the graph grammar robot system using Brownian motion (Klavins et al., 2004) and the more deliberative beam assembly robots in a near frictionless environment (Everist et al., 2004).

This paper is organized into the following sections. Section 2 describes our full process for the production of a multiple target morphologies from the substrate of thousands of homogoneous modules and describes the experiments done and not yet done to validate the overall approach. Section 3 concludes the paper with lessons learned and future work.

Process Overview

There are nine steps to designing and running a working self-assembling and self-replicating robot system with homogeneous self-reconfigurable robots. We list them here and go into detail for each individual step.

Specify the Target Robot Form

We must first decide what kind of robot we wish to build out of the homogeneous self-reconfigurable robot modules. This involves specifying the topology of the robot and how the modules are connected together. This can be represented as a graph or an adjacency matrix. However, with modules with multiple and distinct docking faces, one might have to use a more nuanced description such as the graph method described in (Hou and Shen, 2006). We select the ”helper robot” in Figure 3 as our target form.

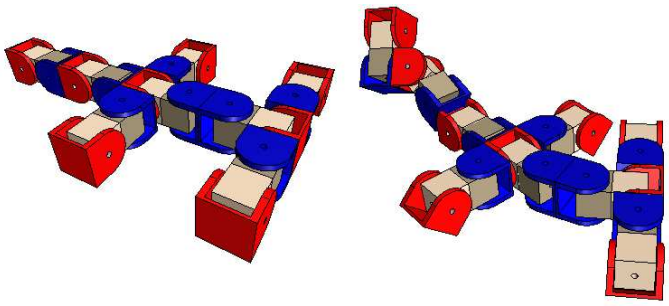

Figure 3: Helper Robot, Helper Robot in Action

Generate Process Grammar for Assembly to Target Form

Once we have a target robot topology, we must generate a process grammar that describes the assembly process for making the final robot form. This method is exactly like that described in (Klavins et al., 2004) where the rules describe which modules should connect together and which connections are unacceptable. In the paper, they used Brownian motion of the modules to make arbitrary random connections, and through the process grammar rules, filter out the illegal connections and maintain the acceptable ones. Given enough time, the target robot shape is created. We use this method in our process to guarantee that assembly is accomplished in a predictable way.

Design Assembly-Line Process into ”Piles” of Intermediate Robot Shapes

Given that we are expecting to have thousands of modules in our system, and that we have no intention of putting our robots into a frictionless, random motion type environment, we must separate the assembly steps into an assembly-line to make the approach feasible for large systems in a reasonable amount time.

From the process grammar, we must identify intermediate robot shapes that can be assembled at a distinct location by bringing the corresponding robot parts together and assembling them. However, our assembly-line need not be as rigidly precise and controlled as expected in industrial manufacturing. Given that we have previous successful experiments in inter-robot docking (Rubenstein et al., 2004b) and self-replication using a random walk in moderate collections of self-reconfigurable robots (Rubenstein et al., 2004a), we believe it is only necessary that the robot parts congregate into ”piles” whether they can assemble themselves into the intermediate robot shapes. Once the intermediate robot shapes are completed, they are then transported to the next appropriate ”pile” for further assembly. This concept is illustrated in Figure 4.

Utilize Some Modules to Create Helper Robots for Cross-Pile Transportation

Most, if not all, intermediate robot forms are going to have limited mobility. In order to finish the assembly process in a timely manner, we introduce "helper" robots into the process to facilitate the transportation of intermediate forms between assembly piles. These helpers are simply responsible for grabbing finished intermediate robot parts, navigating to the next appropriate pile, and dropping the part into the pile. These helper robots are analogous to worker bees or ants in insect colonies. The function of helper robots is similar to the role of FIMER robots in (Shen et al., 2003). The number of helper robots needed is dependent on how quickly the assembly process is required to be completed. When the assembly process is mostly completed, these helper robots can be disassembled and re-purposed to create more copies of the target robot form. The role of helper robots is also shown in Figure 3.

In this case, the helper robot is actually our target form, so this is a true instance of self-replication. The helper robot is assisting in construction of its own copies. As seen in the figure, four legs and an articulated spine provide much of the mobility while a tail with a docking connector at the end provides the ability for the robot to tow loose parts around from pile-to-pile. Figure 3 also shows the helper robot towing a single module around.

If our target form were not the helper robot, some helpers would still need to be constructed to assist in the timely assembly of the target form. At least one helper needs to be pre-assembled for the system to assist in mobility and adequately perform self-replication.

For Each Intermediate Form, Generate Appropriate Behavior to Satisfy the Assembly Process

For each stage in the assembly process, the intermediate robot forms will have a certain assembly function they need to achieve. To do this, they need some form of rudimentary mobility to move the part in space and align to neighboring parts for assembly. In addition, they require a means to know which modules they should connect to and in what orientation. Finally, they must also call the helper robots when they require transportation after finishing assembly in the current pile.

Model Process for Feasibility Using Stochastic Rate Equations

Once we have specified the assembly process and the behaviors the intermediate robots are going to have, we must do some basic modeling to determine the feasibility of the system. There is no central task manager that is keeping track of resources, so a stochastic model is useful to determine whether or not the current approach is balanced enough to produce the number of robots you desire in a timely manner.

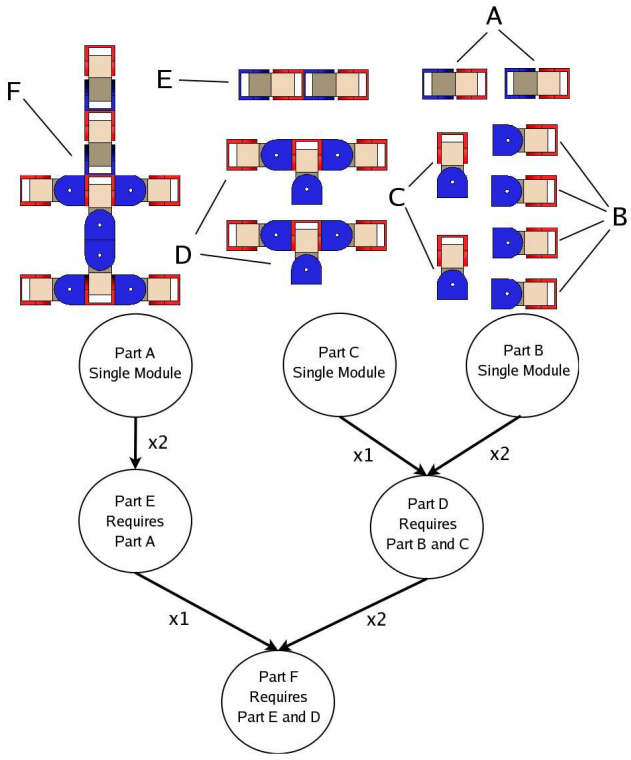

Figure 4: Assembly-Line Pile Process

If the assembly is not designed properly, the process could lead to starvation scenarios when there is low availabilty of a critical part but over-supply of some lesser part. We can use the stochastic modeling techniques as described in (Lerman and Galstyan, 2001). In this paper, they describe how to model the behaviors and populations of robots using coupled differential equations. This allows us to plot the population of all robots and their intermediate forms over time and get a good prediction of how the current assembly approach will work out in the real world.

Verify Each Docking Condition with Experiments

In each assembly pile, we must verify that the desired docking action can occur. This requires us to do a physical experiment for each docking scenario and adjust the assembly process to work around any difficulties. Fortunately, we have successful experiments in inter-robot docking in (Rubenstein et al., 2004b) as well as more detailed work on the general problem of sensor-based connector alignment for docking in (Payne et al., 2006). Using these resources, we can accomplish nearly any docking scenario if we isolate the assembly process into intermediate assembly steps.

Simulate to Confirm Modeling Results

A good simulation should be prudent to verify our expectations of how the overall assembly process is going to work before investing significant time and resources into a

real physical system. We can use such robot simulators as Player/Stage or Gazebo (Gerkey et al., 2006) to test our expectations. The results of these simulations should closely correspond to the results of our stochastic models. If they do not, there is either an unreasonable assumption in our model or an error in the simulator.

Run Whole-System Physical Experiment with Self-Reconfigurable Robots

Finally, we will build and run the whole assembly process on the physical system. The robot system should have the capacity for inter-module docking, sensors for navigation and neighbor-module detection, and enough communications to determine whether docking between neighbors is appropriate. Each part must have enough mobility to do a random-walk and a helper robot design must have enough strength and mobility to haul the intermediate parts from pile to pile.

Conclusion

We believe this is the first top-to-bottom plan for a viable system for self-replication of arbitrary robot forms. We require the availability of self-reconfigurable robot modules as the substrate for self-assembly and self-replication. This system also breaks out of the constraints of microscale and nanoscale stochastic methods commonly utlized in the literature and repurposes them for a scalable macroscopic robotic system with the concept of random walk "piles" and an assembly-line process with transporting helper robots.

Further work is required to verify each step of the process and demonstrate complete self-assembly and self-replication in a real physical robot system.

Acknowledgements

This research is supported in part by NASAs Cooperative Agreement NNA05CS38A, and in part by US Army Research Office under the grants W911NF-04-1-0317 and W911NF-05-1-0134. We also wish to thank Mike Rubenstein for his enthusiasm and helpful dicussions about practical self-replication.

References

Everist, J., Mogharei, K., Suri, H., Ranasinghe, N., Khoshnevis, B., Will, P., and Shen, W.-M. (2004). A system for in-space assembly. In *Proc. of the Int'l Conf. on Intelligent and Robotic Systems*.

Freitas Jr., R. and Merkle, R. (2004). *Kinematic Self-Replicating Machines*. Landes Bioscience.

Gerkey, B., Vaughan, R., Howard, A., and Koenig, N. (2006). Player/stage/gazebo. http://playerstage.sourceforge.net/.

Hou, F. and Shen, W.-M. (2006). Mathematical foundation for hormone-inspired control for self-reconfigurable robotic systems. In *Proc. of the Int'l Conf. on Robotics and Automation*.

Klavins, E., Ghrist, R., and Lipsky, D. (2004). Graph grammars for self assembling robotic systems. In *Proc. of Int'l Conf. on Robotics and Automation*.

Lerman, K. and Galstyan, A. (2001). A general methodology for mathematical analysis of multi-agent systems. USC Information Sciences Technical Report ISI-TR-529.

Payne, K., Everist, J., Hou, F., and Shen, W.-M. (2006). Single-sensor probabilistic localization on the seres self-reconfigurable robot. In *Proc. of Conf. on Intelligent Autonomous Systems*.

Rubenstein, M., Krivokon, M., and Shen, W.-M. (2004a). Robotic enzyme-based autonomous self-replication. In *Proc. of the Int'l Conf. on Intelligent and Robotic Systems*.

Rubenstein, M., Payne, K., Will, P., and Shen, W.-M. (2004b). Docking among independent and autonomous conro self-reconfigurable robots. In *Proc. of the Int'l Conf. on Robotics and Automation*.

Shen, W.-M., Krivokon, M., Chiu, H., Everist, J., Rubenstein, M., and Venkatesh, J. (2006). Multimode locomotion for reconfigurable robots. *Autonomous Robots*, In Press.

Shen, W.-M., Will, P., and Khoshnevis, B. (2003). Self-assembly in space via self-reconfigurable robots. In *Proc. of the Int'l Conf. on Robotics and Automation*.

Suthakorn, J., Cushing, A., and Chirikjian, G. (2003). An autonomous self-replicating robotic system. In *Proc. of the 2003 IEEE/ASME Int'l Conf. on Advanced Intelligent Mechatronics*.

von Neumann, J. (1966). *The Theory of Self-Reproducing Automata*. Univ. of Illinois Press.

Zykov, V., Mytilinaios, E., Adams, B., and Lipson, H. (2005). Self-reproducing machines. *Nature*, 435:163–164.