Coevolutionary Robotics

Jordan Pollack, Hod Lipson, Pablo Funes, Sevan Ficici, Greg Hornby
Dynamical and Evolutionary Machine Organization
Department of Computer Science
Brandeis University
Waltham Massachusetts 02454 USA
{pollack, lipson, funes, sevan, hornby}@cs.brandeis.edu
www.demo.cs.brandeis.edu

Abstract

We address the fundamental issue of fully automated design (FAD) and construction of inexpensive robots and their controllers. Rather than seek an intelligent general purpose robot — the humanoid robot, ubiquitous in today's research as the long term goal — we are developing the information technology that can design and fabricate special-purpose mechanisms and controllers to achieve specific short-term objectives. These robots will be constructed from reusable sensors, effectors, and computers held together with materials custom "printed" by rapid prototyping (RP) equipment. By releasing the goal of designing software controllers for EXISTING machines in favor of the automated co-design of software and hardware together, we will be replicating the principles used by biology in the creation of complex groups of animals adapted to specific environments.

Programming control software has become so difficult as more degrees of freedom and task goals are added to robots, that the most advanced ones do not get past the stage of teleoperation or choreographed behavior. In other words, they are puppets, not robots. Our primary hypothesis is that the reason current approaches to robotics often fail is because of an underestimation of the complexity of the software design problem. Traditionally, engineers will build a complex robot, complete with powerful motors and sensors, and leave for the control programmers to write a program to make it run. But if we look into nature, we see animal brains of very high complexity, at least as complex as the bodies they inhabit, which have been precisely selected to be controllable. New sensor and effector technology — for example, the micromotor, the optical position sensor, memory wire, FPGA's, biomimetic materials, biologically inspired retinas, and lately, MEMS, despite radical claims, cannot produce the desired breakthroughs. True robot success is task specific, not general purpose, and would be recognizable even if built of old electromechanical components.

In nature, the body and brain of a horse are tightly coupled, the fruit of a long series of small mutual adaptations — neither one was first. Today's horse brain was lifted, 99.9% complete, from the animal that preceded it. There is never a situation in which the hardware has no software, or where a growth or mutation — beyond the adaptive ability of a brain — survives. This chicken-egg problem of body-brain development is best understood as a form of co-evolution — agents learning in environments that respond to the agents by creating more challenging and diverse tasks.

By using a combination of commercial off-the-shelf (COTS) CAD/CAM simulation software and our own physical simulators constrained to correspond to real physical devices, we have been developing the technology for the coevolution of body and brains: adaptive learning in body simulations, and the migration of "brains" from simpler to more complex simulated bodies until the virtual robot steps into reality using extensions of today's rapid prototyping technology. Finally, the robot's brains must be robust enough to learn how to bridge the transition from virtual to actual reality.

1 Introduction

The field of Robotics today faces a practical problem: flexible machines with minds cost much more than manual machines, human operators included. Few would spend $2k on an automatic vacuum cleaner when a manual one is $200, or $500k on a driverless car when a regular car is $20k. The high costs associated with designing, manufacturing and controlling robots has led to the current stasis, where robots are only applied to simple and highly repetitive industrial tasks.

The central issue we begin to address is how to get a higher level of complex physicality under control with less human design cost. We seek more controlled and moving mechanical parts, more sensors, more nonlinear interacting degrees of freedom — without entailing both the huge fixed costs of human design and programming and the variable costs in manufacture and operation. We suggest that this can be achieved only when robot design and construction are fully automatic such that the results are inexpensive enough to be disposable.

The focus of our research is how to automate the integrated design of bodies and brains using a coevolutionary learning approach. The key is to evolve *both* the brain and the body, *simultaneously and continuously*, from a simple controllable mechanism to one of sufficient complexity for a task. Within a decade we see three technologies which are maturing past threshold to make this possible. One is the increasing fidelity of "silicon foundries," advanced mechanical design simulation, stimulated by profits from successful software competition. The second is rapid, one-off prototyping and manufacture, which is proceeding from 3d plastic layering to stronger composite and metal (sintering) technology. The third is our understanding of coevolutionary machine learning in design and intelligent control of complex systems.

2 Coevolution

Coevolutionary Learning is about capturing the open-world generative nature of biological evolution in software, to create systems of great complexity and flexibility without human design and engineering. It is different from ordinary genetic algorithms in that the "fitness function" is non-stationary, and these changing goals are created by the learning system itself, rather than being fully specified. There are many claims in the literature about the discovery of "arms races" and "coevolutionary feedforward loops," but in our opinion, there are only a few successful pieces of work to date on open-ended strategic discovery systems. Thomas Ray's TIERRA eco-system of artificial assembly language programs made the first strong claims, but are difficult to evaluate, while Hillis' work on coevolving sorting networks and difficult sequences pointed out several interesting heuristics. There is a line of robotic coevolution work using predator/prey differential games e.g., at Sussex University. However the best exemplars of the power of coevolution are Tesauro's work on TD-Gammon, which is one of the best backgammon players in the world, and Karl Sims' virtual Robots.

Karl Sims' work is particularly relevant. He developed a computer graphics simulator of the physics of robots composed of rectangular solids and several controlled joints, then simultaneously evolved the morphology of the robots and patterns of control using high-level neurally-inspired control constructs. As a form of "genetic art," some of his work was to evolve walking or swimming animats for movies. But by matching pairs of robots in a competition to take possession of a single target, he was able to observe a sequence of coevolutionary attack/defend stages in the evolved designs of his simulated robots.

In TD-Gammon, Tesauro used temporal difference learning in a neural network architecture as the basis for an evaluation function for backgammon (Tesauro, 1992), which under further development became one of the best players in the world (Tesauro, 1995). Although TD-Gammon may be seen as a success of Neural Networks or Reinforcement Learning, we suspected it was really the biggest success of a co-evolution strategy where a learner is embedded in an appropriately changing environment to enable continuous improvement. Many people have tried the idea of a computer learning-by-playing-itself before, beginning with Samuel's checker player, but without such notable and surprising success. Following a hunch, we basically replicated the effect of Tesauro's work using the much simpler learning method of *hill-climbing* (Pollack and Blair, 1998). In this work, we used the same feedforward network with 4000 weights as Tesauro, but trained with a very naive method. Given the current champion, we create a challenger by adding Gaussian noise and playing a small tournament between the current champion and challenger, and changing the weights of the champion if the challenger won. Analysis of why a naive method like hill-climbing could work for self-learning of backgammon strategy led to a deep insight about mediocrity in training and educational systems.

In games, in particular, the "setup" enables players in a population to compete against each other, and the fitness of a player is defined relative to the rest of the population. In theory, improvements in some learners' abilities trigger further improvements in others. In practice, this turns out to be a difficult goal to achieve. Players, especially in deterministic situations, often figure out how to narrow the scope of play, and how to draw each other, and thus stop the learning process, resulting in strategies which are not robust. These collusive "Mediocre Stable States" (MSS) are prevalent in co-evolution; Backgammon's instability in final outcomes — its reversability — helped prevent MSS's, and thus was a key feature which lead to the success in learning.

We have been evaluating ways of making other problems more like backgammon, and in heuristics for preventing mediocre stability and keeping co-evolutionary arms-races going. We have been able to scale up to harder combinatorial problems, like the design of sorting networks and functional cellular automata rules (Juille and Pollack, 1998).

Co-evolution, when successful, dynamically creates a series of learning environments each slightly more complex than the last, and a series of learners which are tuned to adapt in those environments. Sims' work demonstrated that the neural controllers and simulated bodies could be co-evolved. Unfortunately, his simulator has not been released, his robots are not constrained to be buildable, and no one has been able to replicate or extend the work. The goal of our research in coevolutionary robotics is to replicate and extend results from virtual simulations like these to the reality of computer designed and constructed special-purpose machines that can adapt to real environments.

We are working on coevolutionary algorithms to develop control programs operating realistic physical device simulators, both COTS and our own custom simulators, where we finish the evolution inside real embodied robots. We are ultimately interested in mechanical structures which have complex physicality of more degrees of freedom than anything that has ever been controlled by human designed algorithms, with lower engineering costs than currently possible because of minimal human design involvement in the product.

It is not feasible that controllers for complete structures could be evolved (in simulation or otherwise) without first evolving controllers for simpler constructions. Compared to the traditional form of evolutionary robotics, which serially downloads controllers into a piece of hardware, it is relatively easy to explore the space of body constructions in simulation. Realistic simulation is also crucial for providing a rich and nonlinear universe. However, while simulation creates the ability to explore the space of constructions far faster than real-world building and evaluation could, transfer to real constructions is often problematic. Because of the complex emergent interactions between a machine and its environment, final learning must occur in "embodied" form.

3 Research Thrusts

We thus have three major thrusts in achieving fully automated design of high-parts-count autonomous robots. The first is **evolution inside simulation**, but in simulations more and more realistic so the results are not simply visually believable, as in Sims work, but also tie into manufacturing processes. Indeed, interfacing evolutionary computation systems to COTS CAD/CAM systems through developer interfaces to commercial off-the-shelf mechanical simulation programs seems as restrictive as developing programming languages for 8K memory microcomputers in the middle 1970's. However, even though the current mechanical simulation packages are "advisory" rather than blue-print generating, and are less efficient than research code, as computer power grows and computer-integrated-manufacturing expands, these highly capitalized software products will absorb and surpass research code, and moreover will stay current with the emerging interfaces to future digital factories. The second thrust is to **evolve buildable machines**, using custom simulation programs. Here, we are willing to reduce the universe of mechanisms we are working with in order to increase the fidelity and efficiency of the simulators and reduce the cost of building resulting machines. The third is to perform **evolution directly inside real hardware**, which escapes the known limitations of simulation and defines a technology supporting the final learning in embodied form. This is perhaps the hardest task because of the power, communication and reality constraints.

We have preliminary and promising results in each of these three areas, which will be sketched out below.

3.1 Evolution in Simulation

We have been doing evolution of neural-network controllers inside realistic CAD simulations as a prelude to doing body deformation and coevolution. Our Lab has acquired a short term license to a state of the art CAD/CAM software package, which comprises a feature based solid-modeling system. Widely used in industry, it includes a mechanical simulation component that can simulate the function of real-world mechanisms, including gears, latches, cams and stops. This program has a fully articulated development interface to the C programming language, which we have mastered in order to interface its models to our evolutionary recurrent neural network software.

To date, we have used this system with evolved recurrent neural controllers for one and two segment inverted pendulums and for Luxo (an animated lamp creature, Figure 1). Many researchers have evolved such controllers in simulation, but no one has continuously deformed the simulation and brought the evolved controllers along, and no one else has achieved neural control inside COTS simulations. We believe this should lead to easy replication, extension, and transfer of our work.

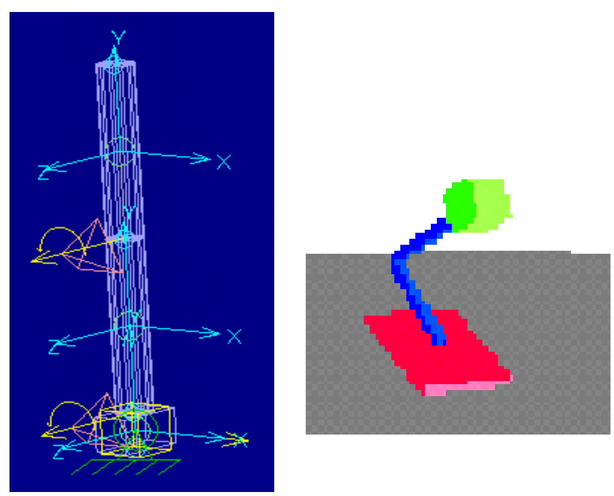


FIGURE 1. COTS CAD models for which we evolved RNN controllers; two segment inverted pendulum and Luxo.

Some of the ways to achieve continuous body deformation are:

- New links can be introduced with “no-op” control elements.
- The mass of new links can initially be very small and then incremented.
- The range of a joint can be small and then given greater freedom.
- A spring can be simulated at a joint and the spring constant relaxed.
- Gravity and other external load forces can be simulated lightly and then increased.

We have successful initial experiments consisting of evolving recurrent neural network controllers for the double-pole balancing problem, where we slowly “morphed” the body simulator by simulating a stiff spring at the joint connecting the two poles and relaxing its stiffness.

3.2 Buildable Simulation

These COTS CAD models are in fact not constrained enough to be buildable, because they assume a human provides numerous reality constraints. In order to evolve both the morphology and behavior of autonomous mechanical devices that can be built, one must have a simulator that operates under many constraints, and a resultant controller that is adaptive enough to cover the gap between the simulated and real world. Features of a simulator for evolving morphology are:

- Universal — the simulator should cover an infinite general space of mechanisms.
- Conservative — because simulation is never perfect, it should preserve a margin of safety.
- Efficient — it should be quicker to test in simulation than through physical production and test.
- Buildable — results should be convertible from a simulation to a real object.

One approach is to custom-build a simulator for modular robotic components, and then evolve either centralized or distributed controllers for them. In advance of a modular simulator with dynamics, we recently built a simulator for (static) lego bricks, and used very simple evolutionary algorithms to create complex lego structures, which were then manually constructed (Funes & Pollack, 1999)

Our model considers the union between two bricks as a rigid joint between the centers of mass of each one, located at the center of the actual area of contact between them. This joint has a measurable torque capacity. That is, more than a certain amount of force applied at a certain distance from the joint will break the two bricks apart. The fundamental assumption of our model is this idealization of the union of two Lego bricks together.

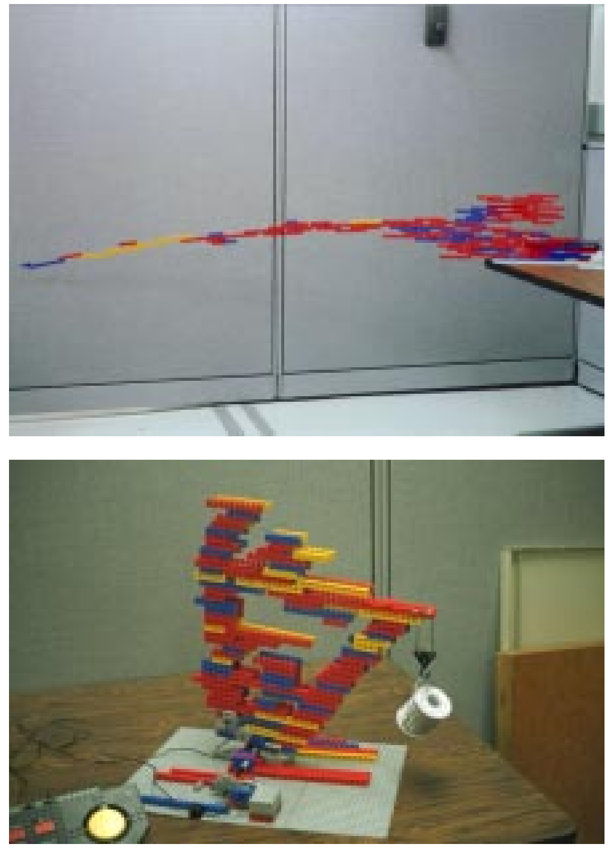


FIGURE 2. Photographs of the FAD Lego Bridge (Cantilever) and Crane (Triangle) Photographs copyright Pablo Funes & Jordan Pollack, used by permission.

The genetic algorithm reliably builds structures which meet simple fitness goals, exploiting physical properties implicit in the simulation. Building the results of the evolutionary simulation (by hand) demonstrated the power and possibility of fully automated design The long bridge of Figure 2 shows that our simple system discovered the cantilever, while the weight-carrying crane shows it discovered the basic triangular support.

The next step is to add dynamics to modular buildable physical components. Lego bricks are also not optimized for automatic assembly, but for young human hands. We are currently developing simulation and modeling software for coevolution in a universe of 3-d “living truss” structures of 2-d shapes controlled by linear motors, as seen in Figure 3.

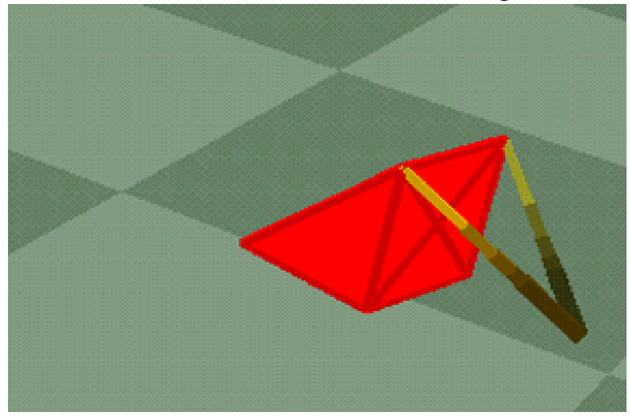
The simulated universe is based on quasi-static motion, where dynamics are approximated as a series of frames, each in full static equilibrium. We have focused on this kind of motion as it is simple and fast to simulate, yet still

FIGURE 3. Prototype "living truss" robot and detail of linear motor assembly

provides an environment sufficiently rich for enabling tasks such as locomotion and other dynamic behaviors. Moreover, it is easier to induce physically since real-time control issues are eliminated. The simulator handles arbitrary compositions of bars, connectors, actuators and controlling neurons, giving rise to arbitrary structures with natural hierarchy as bars aggregate into larger rigid components. The simulation involves internal forces, elasticity and displacements, as well as external effects such as collision, gravity, floor contact, friction, material failure, and energy consumption. Some examples are shown in Figure 4.

3.3 Embodied Evolution

Once a robot is built, learning must proceed in the real world. Anticipating robots composed of many smaller and simpler robots, our work on evolution in real robotic has focused technologically on two of the main problems — reprogramming and long-term power (Watson, Ficici, and Pollack, 1999). Many robots' batteries last only for a few hours, and in order to change programs, they have to be attached to a PC and the new program has to be downloaded. In order to do large group robot learning experiments, we have designed a continuous power floor system, and the ability to transfer programs between robots via IR communications. We are thus able to run a population of learning robots battery-free and wire-free for days at a time (Figure 5). Evolution is not run by a central controller that installs new programs to try out, but is distributed into the behavior of all the robots. The robots exchange data and program specifications with each other and this "culture" is used to 'reproduce' the more successful behavior and achievement of local goals.



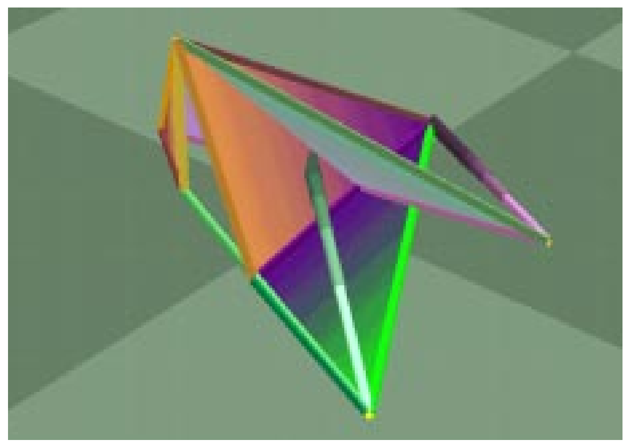


FIGURE 4. Simulated "living truss" robots: (a) hand designed, (b) random structure

The control architecture is a simple neural network and the specifications for it are evolved on-line. That is, each robot tries parameters for the network and evaluates its own success. The more successful a robot is at the task, the more frequently it will broadcast its network specifications via the local IR communications channel. If another robot happens to be in range of the broadcast, it will adopt the broadcast value with a probability inversely related to its own success rate. Thus, successful robots attempt to influence others, and resist the influence of others, more frequently than less successful robots.

We have shown this paradigm to be robust in both simulation and in real robots, allowing for parallel asynchronous evolution of large populations of robots with automatically developed controllers. These controllers compare favorably to human designs, and often surpass them when human designs fail to take all important environmental factors into account. The graph

FIGURE 5. Our 4” diameter robot picks up power from its environment and learns while on-line.

below (Figure 6) shows averaged runs of the robots in a “light gathering” task, comparing a random controller to a human designed controller, to the robots learning themselves.

Our research goals in this area involve group interactive tasks based on multi-agent systems, such as group pursuer evader models. We are planning to build another generation of throwaway powered robots which can hold larger programs. Embodied evolution is a necessary skill to enable the final step in our plan for fully automatic design, adapting the rapidly manufactured body to its real environment.

4 Related Research

The automatic design idea is perhaps the most challenging, as it entails imitation of one of humanity’s most prominent acts of intelligence: *creativity*. Current engineering practice advocates that design is primarily experience related, and various prescriptive design methodologies have been developed and taught (for examples, see Pahl & Beitz, 1996; French, 1994). These methodologies try to cover general purpose complex design tasks; however, at the base of these approaches is the human engineer who makes the critical decisions and spans the base of solution variety. Indeed, more recent approaches seek a more computational basis for engineering design, thereby relieving some of its dependency on experience, and relating it to foundations of information theory (Suh, 1990) and set theory (Yoshikawa, 1985). At the core of these methods too, however, lies a human engineer or a human-generated knowledge base, and hence they can never be fully automated by definition.

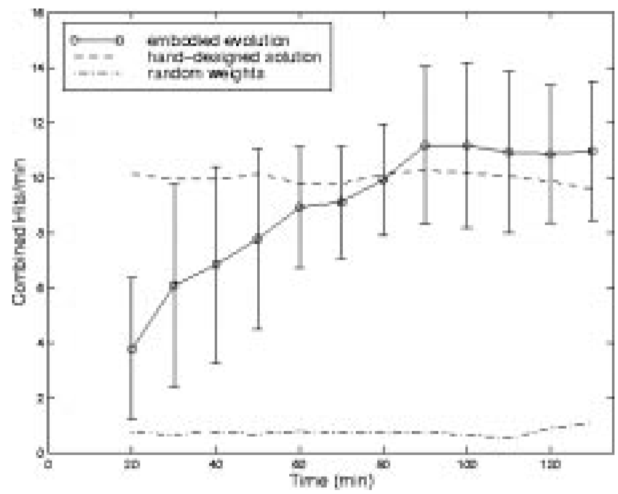


FIGURE 6. Averaged runs of the T1 robots in a “light gathering” task, with various controllers

While engineering design methodologies try to cover general-purpose practical design, a more limited arena of design research has emerged under the field of robotics. This field tries to develop controlled mechanisms that exhibit properties that, to a large extent, are inspired from biological creatures; properties such as locomotion, social behavior and autonomy are especially prevalent. This narrower focus has enabled robotic design to endeavor more closely to the goal of full automation, and will remain the focus of the following discussion.

In general, robotic design is a process that attempts to generate a set of physically embodied solutions. The set of solutions is required to meet a specification while residing within the scope of certain constraints. Both the specification and constraints can be thought of as assigning solutions a general attribute of merit, applied with a positive and negative stimulus, respectively. All three of these aspects — the specification, the constraints and the solution generating process — are crucial to the success of the design. Traditional robotic design has (wrongfully, in our opinion) addressed these *by discipline* in two separate efforts, that of designing the hardware components (the body), and that of designing the software controller (the brain).

Most research effort in automatic design in robotics has been focused on the evolutionary design of controllers (Husbands and Meyer, 1998). Several researchers have attempted to bypass the difficulties of hand coding the control architecture of mobile robots that have to perform given tasks in unknown and changing environments. Because of the impossibility of foreseeing the problems the robot will have to solve, because of the lack of basic design principles and because the scope of solution cannot (should not) be specified by a designer, a robot's controller is progressively adapted to a specific environment and task through an artificial selection process that eliminates ill-behaving individuals in a population while favoring the reproduction of better adapted individuals. Numerous aspects of evolutionary design of robotic controllers have been tested both in simulation and on real robots. A major candidate robot platform for evaluation of evolved controllers is the Khepera robot (Mondada et al, 1993), which is a circular mobile robot with 2 to 4 wheels and two dimensional motion. Various obstacle avoiding and light seeking behaviors were evolved, some in simulation only, some in simulation later embodied in the robot (Jakobi et al, 1997; Miglino et al, 1995, Nolfi et al, 1997; Salomon, 1996; Naito et al, 1997), and some directly in the robot (Floreano and Mondada. 1994). Other interesting attempts were carried out with different robot types, such as a visual-tracking gantry robot (Harvey et al, 1994), a NOMAD 200 mobile robot with 50 sensors (Grefensette and Schultz, 1994), six-legged robots (Gallagher et al, 1996) and eight-legged robot (Galt et al, 1997; Gomi et al 1997, and Gruau and Quatramaran, 1997). Another attempt to evolve controllers is Thompson's work (1997) to evolve hardware circuits as on-board controllers. In contrast with other work, Thompson tried to evolve the controller directly as an electronic circuit using Field Programmable Gate Arrays (FPGA's). Thompson's work, while basically a software controller, illustrates how evolutionary computation can take advantage of emergent physical effects.

However, evolution of controllers in robots can be very time consuming. Even evolving simple controllers for simple simulated robots takes hundreds and thousands of trials: (Gritz and Hahn, 1997) needed 7500 evaluations for Luxo; Moriarty & Miikulainen required 4000 evaluations for a robot arm; evolving controllers for more complex robots will take even longer. In serialized evolution embodied in real robots, the time problem is more acute. Mondada and Floreano spent 100 generations at 39 minutes a generation to evolve a controller to get a Khepera to grasp a ball. This took about 65 hours. In many cases the simulation requirements conflict; for example, efficiency contradicts feasibility. Several approaches to addressing this conflict have been proposed, such as the minimal simulation (Jakobi, 1998). However, this may not be enough. As Mataric and Cliff note (1996), the cost and errors in simulation may have grave implication to the prospects of traditional evolutionary robotics.

At the other end of the spectrum, there have been several attempts to generate robots whose actual physical body plan is variable. Here too we distinguish between pure simulations and physical attempts, as well as between simply reconfigurable robots and those that are continuously evolvable. Starting with physically reconfigurable robots, Chirikjian at John Hopkins University employs a Metamorphic Robotic System (Chirikjian, 1994), which is a collection of self-assembling two dimensional hexagonal and square units that are independently controlled mechatronic modules, each of which has the ability to connect, disconnect and climb over adjacent modules. At the California Institute of Technology, Chen (1994) studies task optimal configurations, the kinematics and dynamics of reconfigurable robots, and evolutionary approaches to determine task optimal modular robot configurations. Yim (1994) developed at Stanford a reconfigurable robot composed of multiple components of two types. This robot has been shown to be able to attain eight different forms in three dimensions, corresponding to different locomotion gates. Fukuda at Nagoya University is developing the cellular robotic system (CEBOT) for cooperating autonomous self-organizing cells (Fukuda, 1991). The above works, however, are directly programmed and do not involve an evolutionary or other general optimization process to derive the actual physical configuration and its corresponding controller.

Research on *evolving* the physical body plan *in conjunction* with a corresponding controller is rarer. Of particular relevance are Sims' simulations (1994) discussed earlier. There have been other attempts to evolve feasible hardware configurations; Lund, Hallam and Lee have evolved in simulation both a robot control program and some parameters of its physical body such as number of sensors and their positions, body size, turret radius, etc. (Lund et al., 1997b, Lund et al., 1997a). However, this evolution is parametric in the sense that it is limited only to parameters foreseen by the designer, and hence is not open-ended, and will not be able to adapt to unforeseen situations or provide new 'creative' designs. A recent work by Dittrich et al (1998) describes a Random Morphology Robot, which is an arbitrary two-dimensional structure composed of links and motors; the controller of this robot is evolved, and then manual changes are applied to the robot to test its behavior with an impaired or mutated body. However, an evolutionary design of the robot's configuration was not attempted. Recent work by Chocron and Bidaud (1997) describes an attempt to evolve both the morphology and the inverse kinematics of a modular manipulator composed of prismatic and revolute joints. A genetic algorithm searches for suitable configuration for a task given as a set of required effector configurations. We consider this attempt as being in the right direction. Again, however, the simple serial construction precludes spontaneous emergence of any innovative 'interesting' or unforeseen solutions.

5 Conclusion

Our work has both a theoretical and practical potential; we aim to understand and to innovate in software as well as in hardware. Our long-term vision is that both the morphology and control programs for self-assembling robots arise directly through hardware and software co-evolution: primitive active structures that crawl over each other, attach and detach, and accept temporary employment as supportive elements in "corporate" beings can accomplish a variety of tasks, if enough design intelligence is captured to allow true self-configuration rather than human redeployment and reprogramming. When tasks cannot be solved with current parts, new elements are created through fully automatic design and rapid prototype manufacturing. Once FAD and RP descend into the MEMS world, it is possible to contemplate a new "bootstrap," similar to the achievement of precision in machine tools, where artificial life gains control of its own means of production and assembly and is able to grow both in power, complexity, and precision.

This vision is easy to imagine, as it indeed was by both NASA scientists and by SF novelists of the 1960's(e.g., Dick, 1960), but quite difficult to work out in practice. There are many problems that need interactive solutions where the primary problem is the relationship between software and physical devices: this vision cannot be achieved either fully in simulation or fully in hardware. It is not a problem for engineers to solve once, but a problem of having machines learn how to automatically engineer physical systems along with their controllers. It is not a situation where a gee-whiz new sensor or effector (with "software-to-follow") can help.

We see several exciting research problems that are addressed by our recent work in this area: one problem is that global configurations of elements are dependent on local interaction, and simple processors inside each element will not suffice to calculate and control the overall configuration. That is why we first focus and develop the conventional algorithms for conservatively simulating structures, and then parallelize into agents, rather than hoping some simple pre-programmed behavior primitives will scale. A second problem is that computer aided design and manufacturing systems, where human designers work in teams to design mass manufactured products, is too expensive a system for a robot to call upon whenever it needs help. That is why we have to make state-of-the-art CAD/CAM subservient to our coevolutionary body-brain simulations rather than to their own human interface. A third problem is power distribution under changing configuration. Plugging and unplugging wires will not suffice. That is why we focus on the problems of power distribution for reconfiguring embodied evolutionary systems.

Our current research moves towards the overall goal down multiple interacting paths, where what we learn in one thrust aids the others. We envision the improvement of our hardware-based evolution structures, expanding focus from static buildable structures and unconnected groups to reconfigurable active systems governed by a central controller, and then the subsequent parallelization of the control concepts. We see a path from evolution inside CAD/CAM and buildable simulation, to rapid automatic construction of novel controlled mechanisms, from control in simulation to control in real systems, and finally from embodied evolution of individuals to the evolution of heterogenous groups that learn by working together symbiotically. We believe such a broad program is the best way to ultimately construct complex autonomous robots who are self-organizing and self-configuring corporate assemblages of simpler automatically manufactured parts.

6 References

Chen, I.-M. (1994). *Theory and applications of modular reconfigurable robotic systems*. Ph.D. thesis, California Institute of Technology.

Chirikjian, G. S. (1994). "Kinematics of a metamorphic robotic system," in *IEEE International Conference on Robotics and Automation*.

Chocron O., Bidaud P. (1997). "Genetic design of 3D modular manipulators," *Proceedings of the '97 IEEE Int. Conf. on Robotics and Automation*, Vol. 1., pp. 223-228.

Cliff, D. and Miller, G. (1995). "Tracking the red queen: Measurements of adaptive progress in co-evolutionary simulations," in *Third European Confrence on Artificial Life*, pp. 200–218.

Cliff, D., Harvey, I., and Husbands, P. (1996). "Evolution of visual control systems for robot," In Srinivisan, M. and Venkatesh, S., editors, *From Living Eyes to Seeing Machines*. Oxford.

Dick, P. (1960). *Vulcan's Hammer*. New York: Ace.

Dittrich P., Burgel A., Banzhaf W. (1998). "Learning to move a robot with random morphology," in Husbands P., Meyer J. A., (Eds.), *Evolutionary Robotics*, Springer.

Floreano, D. and Mondada, F. (1994). "Automatic creation of an autonomous agent: Genetic evolution of a neural network driven robot," in Cliff, D., Husbands, P., Meyer, J., and Wilson, S., editors, *from Animals to Animats III*. MIT Press.

French M. J. (1994). *Invention and Evolution*, 2nd Edition, Cambridge University Press.

Fukuda, T., Kawauchi, Y., Hara, F. (1991). "Dynamic distributed knowledge system in self-organizing robotic system," *CEBOT, Proc IEEE Int Conf. Rob Autom* v 3, pp. 1908-1913.

Funes, P., and Pollack, J. (1999). “Evolutionary Body Building,” *Artificial Life*, 4 (4).

Gallagher J. C., Beer R. D., Espenschield K. S., Quinn R. D. (1996). “Application of evolved locomotion controllers to a hexapod robot,” *Robotics and Autonomous Systems*, 19, pp. 95-103.

Galt S., Luk B. L., Collie A. A. (1997). “Evolution of smooth and efficient walking motions for an 8-legged robot,” *Proceedings of the 6th European Workshop on Learning Robots*, Brighton, UK.

Gomi T., and Ide K. (1997). “Emergence of gaits of legged robot by collaboration through evolution,” *Proceedings of the IEEE 3rd Int. Conf. on Evolutionary Computation*, IEEE Society Press.

Grefenstette J., Schultz A. (1994). “An evolutionary approach to learning in robots,” *Proceedings of the machine learning workshop on robot learning*, New Brunswick, NJ.

Gritz, L. and Hahn, J. K. (1997). “Genetic programming evolution of controllers for 3d character animation,” In *Genetic Programming 97*.

Gruau F., Quatramaran K. (1997). “Cellular Encoding for interactive evolutionary robotics,” in Husbands and Harvey (Eds.), *4th European Conference on Artificial Life*, MIT Press.

Harvey I., Husbands P., Cliff D. (1994). “Seeing the light: Artificial evolution, real evolution,” in Cliff Husbands, Meyer and Wilson (Eds.) *Proc. 3rd int. conf. on simulation and adaptive behavior, From animals to animats 3*, MIT Press.

Hillis, D. (1992). “Co-evolving parasites improves simulated evolution as an optimization procedure,” in C. Langton, C. Taylor, J. F. and Rasmussen, S., editors, *Artificial Life II*. Addison-Wesley, Reading, MA.

Husbands P., Meyer J. A. (1998). *Evolutionary Robotics*, Springer.

Jakobi N. (1998) *Minimal Simulations for Evolutionary Robotics*, Ph.D. Thesis. University of Sussex.

Juille, H. & Pollack, J. (1998). “Coevolving the ‘Ideal Trainer’: Discovery of Cellular automata rules,” *Proc. 3rd Genetic Programming conference*.

Kawauchi Y., Inaba M., Fukuda T. (1995). “Genetic evolution and self-organization of cellular robotic system,” *JSME Int. J. Series C. (Dynamics, Control, Robotics, Design & Manufacturing)*, 38 (3), pp. 501-509.

Lund, H., Hallam, J., and Lee, W. (1997a). “Evolving robot morphology,” in *Proceedings of IEEE Fourth International Conference on Evolution*. IEEE Press.

Lund, H., Hallam, J., and Lee, W. (1997b). “A hybrid gp/ga approach for co-evolving controllers and robot bodies to achieve fitness-specified tasks,” in *Proceedings of IEEE 3rd. International Conference on Evolutionary Computation*. IEEE Press.

Mataric, M. and Cliff, D. (1996). “Challenges in evolving controllers for physical robots,” *Robotics and Autonomous Systems*, 19(1), pp. 67-83.

Miglino O., Lund H. H. and Nolfi S. (1995) “Evolving mobile robots in simulated and real environments,” *Artificial Life*, 2, pp. 101-116.

Mondada F. Franzi E., Ienne P. (1993). “Mobile robot miniaturization: A tool for investigation in control algorithms,” *Proceedings of the 3rd Int. Symp. on Experimental Robotics*, Kyoto, Japan.

Naito T., Odagiri R., Matsunaga Y., Tanifuji M., Murase K., (1997). “Genetic evolution of a logic circuit which controls an autonomous mobile robot,” in Higuichi, Iwata and Liu (Eds.) *Evolvable systems: From biology to hardware*: Springer.

Nolfi S. (1997). “Evolving non-trivial behaviors on real-robots: a garbage collecting robot,” *Robotics and Autonomous Systems, 22, pp. 187-198.*

Pahl, G., Beitz W. (1994). *Engineering Design*, Springer Verlag.

Pollack, J. and Blair, A. (1998). “Co-evolution in the successful learning of backgammon strategy,” *Machine Learning*, to appear.

Ray, T. (1992). “An approach to the synthesis of life,” in C. Langton, C. Taylor, J. F. and Rasmussen, S., editors, *Artificial Life II*. Addison-Wesley, Reading, MA.

Salomon R. (1996). “Increasing adaptivity through evolution strategies,” In Maes, P., Mataric, M., Meyer, J., Pollack, J., and Wilson, S., editors, *From Animals to Animats: Proceedings of 4th Int’l conference on simulation of adaptive behavior*. MIT Press.

Samuel, A. L. (1959). “Some studies of machine learning using the game of checkers,” *IBM Journal of Research and Development*.

Sims, K. (1994). “Evolving 3d morphology and behavior by competition,” in Brooks, R. and Maes, P., editors, *Proceedings 4th Artificial Life Conference*. MIT Press.

Suh, N. P. (1990). *The principles of design*, Oxford University Press, New York.

Tesauro, G. (1992). “Practical issues in temporal difference learning,” *Machine Learning*, 8, pp. 257-277.

Thompson A. (1997). “Artificial evolution in the physical world,” in Gomi (Ed.) *Evolutionary Robotics*, AAI Books.

Watson, R., Ficici, S., and Pollack, J. (1999). “Embodied evolution: Embodying an evolutionary algorithm in a population of robots,” in Angeline, P., Michalewicz, Z., Schoenauer, M.,Yao, X., and Zalzala, A., eds., *1999 Congress on Evolutionary Computation*.

Yim, M. (1993). “A reconfigurable modular robot with many modes of locomotion,” in *Proc. of the JSME Int. Conf. on Advanced Mechatronics*, pp. 283–288.

Yoshikawa, H., Tomiyama, T., Kumazawa, M. (1985). “General design theory,” J Fac Eng Univ Tokyo Ser A: n 23, pp. 24-25.