J. S. Gero (ed.), *Artificial Intelligence in Design'00,* 269-285.
© 2000 Kluwer Academic Publishers, Dordrecht.

EVOLUTION OF PHYSICAL MACHINES

Towards escape velocity

HOD LIPSON AND JORDAN B. POLLACK
Computer Science Department
Brandeis University

Abstract. Can evolutionary principles be used to automatically design and manufacture physical machines and discover new concepts not foreseen by human engineers? *Fully automated design and manufacture* has long been the Holy Grail of the CAD/CAM field. In this research we explore some directions of this prospect. We evolve machines for specified tasks and replicate them into reality, thereby establishing for the first time a complete *physical* evolutionary design cycle into reality. To create a minimally biased design space, only two low-level building blocks are used: linear actuation elements and sigmoidal control neurons. These elements can assemble together to give rise to a universe of possible forms of controlled machines with rigid and flexible components. Quasi-static motion is then used to simulate low momentum behavior, including flexion, material failure, collision and friction. Selected machines are then automatically replicated into reality using rapid prototyping techniques. The automatically designed and manufactured machines then perform in reality. The paper discusses this approach and provides examples of both virtual and real machines evolved for the task of locomotion.

1. Introduction

It can be argued that the act of *design* is perhaps the most prominent distinguishing hallmarks of human intelligence. It is therefore natural to expect issues regarding the understanding and emulation of the design process to drive multidisciplinary research across both science and engineering. But the quest for *both fully automated design and fully automated manufacture* has long been at the crux of technology as a whole: Can machines design newer machines and automatically transfer them into reality, eventually surpassing human capabilities and reaching escape velocity?

1.1 FULLY AUTOMATED DESIGN

By *fully automated design* we imply a machine that given a required functionality, autonomously produces one or several concept of machines that when built will achieve that functionality. The concepts must be produced without any human intervention – without inductive bias, where a human guides the solution process either directly or indirectly via a program or pre-coded knowledge. We focus on open-ended design, and specifically refrain from automation ubiquitous in today's CAD/CAM systems, in which custom tailored programs are used to solve specific engineering problems (as hard as they may be), but it is a human designer that has parameterized a solution space and the computer only calculates or optimizes these parameters.

Fully automated design is required for several reasons. First, at a practical level, it opens the possibility for assisting human designers; it would be convenient to have a button in a CAD system that when pressed, produces alternative design to the one currently displayed. Second, on a longer-term scale, automated procedures might eventually discover new ideas and concepts overseen by human engineers. Third – perhaps on a philosophical scale – some argue that there are problems that are so complex (like the human body and brain), that human designers will never reach using manual techniques.

1.2 FULLY AUTOMATED MANUFACTURE

By *fully automated manufacture*, we imply a machine that given a design, is able to autonomously embody it in physical reality. Again, the process must proceed without human intervention. We focus on open-ended manufacture, and refrain from pre-coded automation like CNC programs, geared to solve specific manufacturing tasks.

Fully automated manufacture is required for several reasons. First, on a practical level, we wish to verify the design stage. Second, on a longer-term scale, having fully automated design and manufacture opens the possibility for design and fabrication of task-specific one-of-a-kind solutions, which are not economically viable to generate in any other way. Finally – perhaps on a philosophical level – merging automatic design and fabrication might eventually *close the loop*, so that machines can design and fabricate newer machines, and these in turn can design and fabricate newer machines – towards *escape velocity* (Moravec, 1999).

1.3 STRUCTURE OF THIS PAPER

In this paper we show how using evolutionary techniques, machines are automatically designed for a given specification and then replicated into reality where they faithfully reproduce their virtual ancestors, with almost no human intervention. Our experiments so far involve fully automated design and manufacture of machines for the task of locomotion. The machines are composed of thermoplastic, motors and artificial neurons. Although the machines we produced are still far from achieving any truly useful practical task, and are still simpler that what any human team of engineers could design, we believe they demonstrate one realistic path towards the vision.

This paper is organized as follows. First, in the background section, we provide an overview of the evolutionary design approach, as well as a brief survey and critique of related work. The next section describes our approach and details its implementation. We then show both virtual and real machines automatically designed and fabricated for the task of locomotion. We conclude with some remarks on future hurdles that need to be addressed.

2. Background

Our design approach is based on evolutionary computation. The field of evolutionary-computation is essentially the attempt to perform functional optimization using techniques inspired by biological evolution and adaptation. Since its invocation in the seventies (Holland, 1975), genetic programming and evolutionary computation techniques have been applied to numerous engineering problems. In the majority of cases, however, evolutionary strategies have been used to seek the optimal values for a finite set of parameters, whose physical meanings and permissible values have been prescribed by a human designer. It is also the human designer that provides a fitness evaluation function that typically encodes many aspects of the preferred solution. In this respect the evolutionary aspect of the problem-solving process has been reduced to mere parametric-optimization, whereas it is in fact the human designer that performed the so-called *creative* conceptual thinking.

2.1 EVOLUTIONARY DESIGN

It is only recently that we have seen some efforts in attempt to bring evolutionary computation out of *parametric* optimization into a more open-ended design space (Bently, 1999). For example, Bently (1996) describes systems for evolutionary design of optical systems in which prisms can be

manipulated and added without limit in search of a solution to satisfying some optical specification. Similarly, he describes a system for open-ended design of tables. Funes and Pollack (1997) describe a system for design of static weight-carrying and distance-spanning Lego™ structures. These designs are also open-ended as they can accumulate more and more bricks of various sizes, effectively giving rise to an infinite universe of possible structures. Nevertheless, both these systems can only add building blocks of a given, finite repertoire, and so they are in essence only *semi-parametric*. A truly *non-parametric* system, in which the building blocks themselves are limitless, is yet to be demonstrated.

2.2 EVOLUTION IN ROBOTICS

A particular field that has attracted more than its share of evolutionary design effort is *robotics*. Both at the morphological and at the control levels we see attempts to identify reconfigurable building blocks, as well as self-assembling structures and self-designing architectures.

Reconfigurable robotics

The field of reconfigurable robotics addresses the ability of machines to change their physical form and structure – in a sense perform automated manufacture of a functioning system. Chirikjian (1994) demonstrated a metamorphic robotic system that is a collection of self-assembling two dimensional hexagonal and square units which are independently controlled mechatronic modules, each of which has the ability to connect, disconnect and climb over adjacent modules. Chen (1994) studied task optimal configurations, the kinematics and dynamics of reconfigurable robots, and evolutionary approaches to determine task optimal modular robot configurations. Similarly, Hara (1998) is developing two dimensional polymorphic linear cluster chain robots, which can be reconfigured to a variety forms. Yim (1994) developed 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. Kawauchi (1994) is developing the cellular robotic system (CEBOT) for cooperating autonomous self-organizing cells.

Evolutionary robotics

Most research effort in automatic design in robotics has been focused on the evolutionary design of controllers (Husbands and Meyer, 1998) for predominantly fixed morphology. 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. 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.

Yet evolving a controller for a fixed robotic body may not be a promising approach. Looking into nature we see animal brains of very high complexity controlling bodies that have been selected by evolution precisely because they were controllable by those brains. In nature, the body and brain of a horse are tightly coupled, the fruit of a long series of small mutual adaptations – like chicken and egg, neither one was designed first. There is never a situation in which the hardware has no software, or where a growth or mutation – beyond the adaptive ability of the brain – survives. We claim it is thus necessary to evolve both the brain and the body, simultaneously and continuously, from a simple controllable mechanism to one of sufficient complexity for a particular specialized task. In other words we hypothesize that for fully automated design it is necessary to abandon traditional disciplinary segregation, and evolve the entire machine concurrently.

Body-brain co-evolution

While robotic design is typically segregated into disciplines of software and mechanical hardware, there have been a few cases where evolution of both morphology and control were combined into so-called body-brain co-evolution.

Sims (1994) demonstrated a methodology for jointly evolving morphologies and controllers for three dimensional creatures composed of blocks and joints. Both the architecture of the body and the sequence of motion was evolved to create various types of locomotion such as walking, swimming, and jumping, as well as competing over a target or seeking light.

More recently, Komosinski and Ulatowski (1999) demonstrated their system *Framstics* in which tree-like creatures co-evolved and competed in a simulated world. Both Sims's and Komosinski's work yielded spectacular behaviors, in many respects reminiscent of natural insect life. Yet products of both these works remain virtual, and no attempt has been made to address their feasibility in reality, let alone being buildable without human intervention. There have been, however several attempts to evolve feasible hardware configurations; Lund *et al* (1997) 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 and turret radius, etc. 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) describe 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. However, the simple serial construction precludes spontaneous emergence of any innovative "interesting" or unforeseen solutions. Leger (1999) presented a system for evolutionary design of physical robots. His work, however, incorporated very high-level modules and knowledge such as entire wheeled platforms and C subroutines as single components.

3. Implementation

This section reports an experiment in which we evolved machines for the specific task of locomotion, and then replicated them automatically into reality. In doing so, we have established for the first time a complete synthetic *physical* evolution cycle, involving fully automatic design and manufacture.

In carrying out this experiment, we emphasized the following principles: (a) To reduce inductive bias, only the simplest elementary components and basic genetic operators were used to span the design space, (b) the entire machine, both body and brain, was evolved together as a single unit, and (c) selected machines were transferred into reality.

3.1 THE DESIGN SPACE

To enable an effective evolution process, it is necessary to define a substrate of elements that can be assembled together in various ways to give rise to an infinite universe of machines. To minimize inductive bias the substrate was comprised of only low-level building blocks: simplest linear actuation elements (for morphology) and sigmoidal neurons (for control). The linear actuators can change their lengths arbitrarily as function of a neuron connected to them, or remain with a fixed length if not connected to any neuron. The neurons respond to the sum of signals received from other neurons multiplied by synaptic connection weights. These two elements types can connect to each other and to themselves to create any arbitrary architecture: neurons can connect to other neurons and to themselves to create architectures such as feedforward or recurrent nets and arbitrary state machines. Actuators, both in motion and static can connect to each other using ball-joints, to create arbitrary truss architectures, and neurons can connect to actuators to produce controlled motion. It is important to emphasize that these elements were used since they are the simplest elements of motion and control available, and many higher-level machine components can essentially be reproduced out of these: rotational, translation and combined complex motion can be produced using only linear actuation (consider, for example, a Stewart platform). Ball-joint trusses can be assembled to reproduce any kind of rigid and articulated structure, containing fixed, free, rotational, planar and linear joint types. Moreover, the arbitrary constraint-free arrangement of these components can provide for various hierarchies of subcomponents, as well as groups of detached machines controlled by a single centralized brain or by multiple individual detached brains, etc. This is in contrast to other works in body-brain evolution (Sims, 1994; Komosinski and Ulatowski, 1999) whose morphological operators permit only tree-like architectures (and so, for example, could never evolve a simple four-bar mechanism). Sim's creatures also use various high-level control units that are capable of producing relatively complex operations such as oscillations – operations that require a recurrent net assembly in our case. Indeed, in some cases over 70,000 evaluations take place before even the first simplest "oscillator net and actuator" emerge – a single building block in Sims work. It should be noted, however, that in using only two elements our design space is only semi-parametric, as are earlier works in evolutionary design.

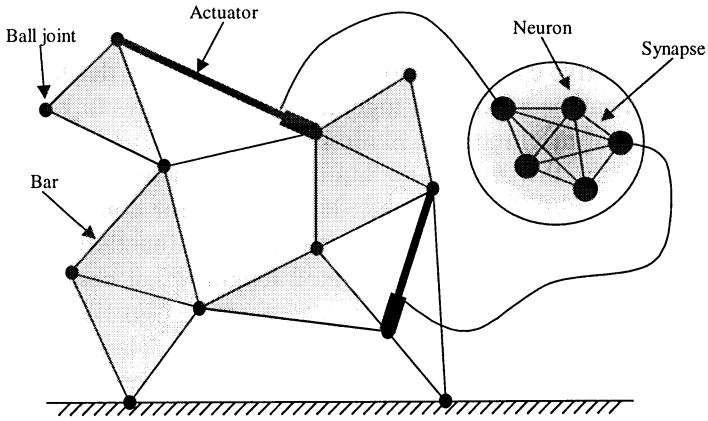


Figure 1. Schematic two-dimensional illustration of an evolvable robot, containing only linear actuators and control neurons. Some actuators are static (bars), and when connected with other actuators might form effectively rigid subcomponents (like the shaded triangles). Actual robots are three dimensional.

3.2 THE SIMULATOR

The simulator used in this research is based on quasi-static motion. The basic principle is that motion is broken down into a series of statically-stable frames solved independently. While quasi-static motion cannot describe high-momentum behavior such as jumping, it can accurately and rapidly simulate low-momentum motion. This kind of motion is sufficiently rich for the purpose of the experiment and moreover, it is simple to induce in reality since all real-time control issues are eliminated. In this respect we permit only those effects that we are confident will transfer into reality, but at the same time introduce a severe constraint on the design universe.

Simulator physics

Static solution of each frame is achieved by defining a global system energy term H and then slightly modifying each of the systems degrees of freedom so as to lower H according to its partial derivatives. This process continues until relaxation is reached or instability is determined. The terms included in H define the richness of the simulation.

For example, a basic model including bar flexion and gravitational energy would be:

$$H = \sum_{1}^{n} k\delta^2 + \sum_{1}^{n} mgh$$

where n is the number of bars, and for each bar i, m represents the mass, h represents the average height, and the term k represents the stiffness,

$$k_i = \frac{E_i A_i}{l_i}$$

where E is the material's module of elasticity, A is the cross section, and l is the length of the bar, and δ represents the difference between the bar's current length and its original length, given by

$$\delta_i = \sqrt{(v_x - w_x)^2 + (v_y - w_y)^2 + (v_z - w_z)^2} - l_i$$

and v,w represents the two current endpoints of the bar. Differentiating the total energy H with respect to each of the degrees of freedom (coordinates of the endpoints, in this case), produces direction of adjustment (second derivatives would produce a more accurate adjustment, etc.).

$$\Delta w_x = \frac{\partial H}{\partial w_x} \cdot \Delta s$$

where Δs (the relaxation factor) represents the adjustment magnitude for each iteration. Each degree of freedom is then updated by its adjustment value, and H is recalculated. Small Δs produce a stable but slower convergence, whereas large Δs can solve static frames faster but may run into stability problems. This process is repeated until adjustments go below a certain threshold.

Although this solution method is slower that simultaneous solutions, (e.g. finite elements), it is capable of solving correct highly-nonlinear cases (e.g. contact collision and snap-through buckling) and, more importantly, under-constrained states: for example, a truss in mid fall which is not yet fully supported (by six rigid-body equations), as it falls into position. Indeed as machines move and function within the simulator they often pass through such stages.

As additional real-world energy terms are added to H, it becomes more accurate. We have included terms for friction, collision between bars, external forces and noise.

3.3 THE EVOLUTION PROCESS

Starting with a population of 200-1000 machines that were initially comprised of zero bars and neurons, we conducted evolution in simulation. The fitness of a machine was determined by its locomotion ability: the net distance its center of mass moved on an infinite plane in a fixed duration of

12-24 cycles of the controlling recurrent net. The process iteratively selected fitter machines, created offspring by adding, modifying and removing building blocks, and replaced them into the population (Figure 2). Here again, we use only elementary operators of mutation that introduce least external knowledge. This process continued for several hundred generations. Both body (morphology) and brain (control) were co-evolved simultaneously.

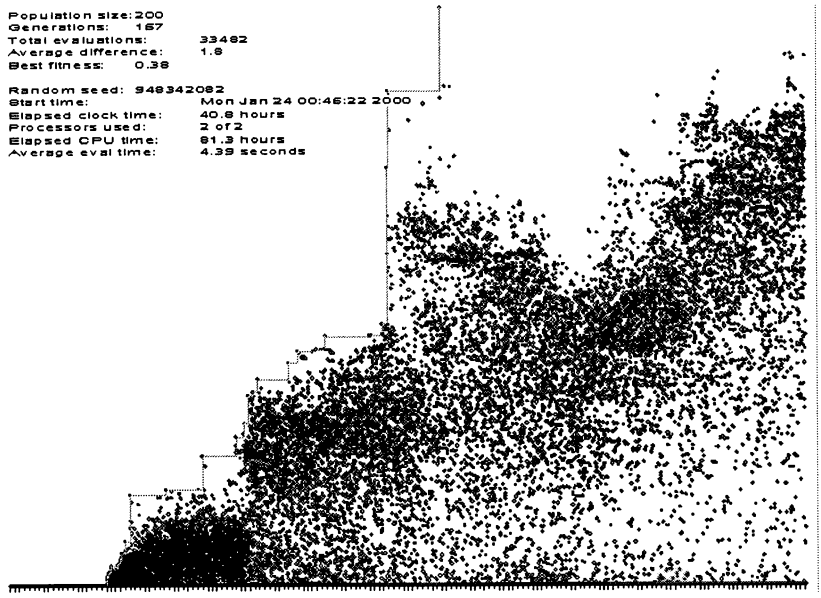


Figure 2. A typical plot of solution fitness (0-0.38) as function of generation (167). Note step increments as new mechanical and control "principles" are discovered.

The choice of genetic operators is again an entry point for inductive bias. In this work we attempted to use only minimal mutation operators that introduce very small changes to both the morphology and the control. The following operators were used (with probabilities):

- Add/remove small bar / disconnected neuron (1%)
- Mutate bar length / neuron weight (10%)
- Connect/disconnect neuron to bar (3%)
- Split vertex into two close vertices connected by short bar (3%)
- Split bar into two bars with new vertex between endpoints (3%)

The dice is rolled until at least one mutation is applied. After mutation, a new fitness is assigned to the individual by means of a simulation of the mechanics and the control (see details below). The offspring is inserted into the population by replacing an existing individual either chosen at random, chosen in inverse-proportion to its fitness, or using similarity proportionate

elitism (deterministic crowding, Mahfoud 1995). Various permutation of selection-replacement methods are possible; we typically used fitness-proportionate or rank selection with random replacement, or random selection with similarity proportionate replacement where similarity was approximated by the distance in the ancestral tree. The process continued for 500-5000 generations (approx 10^5 to 10^6 evaluations overall). The process was carried both serially and in parallel (on a 16-processor computer). On parallel computers we noticed an inherent bias towards simplicity: Simpler machines could complete their evaluation sooner and consequently reproduce more quickly than complex machines.

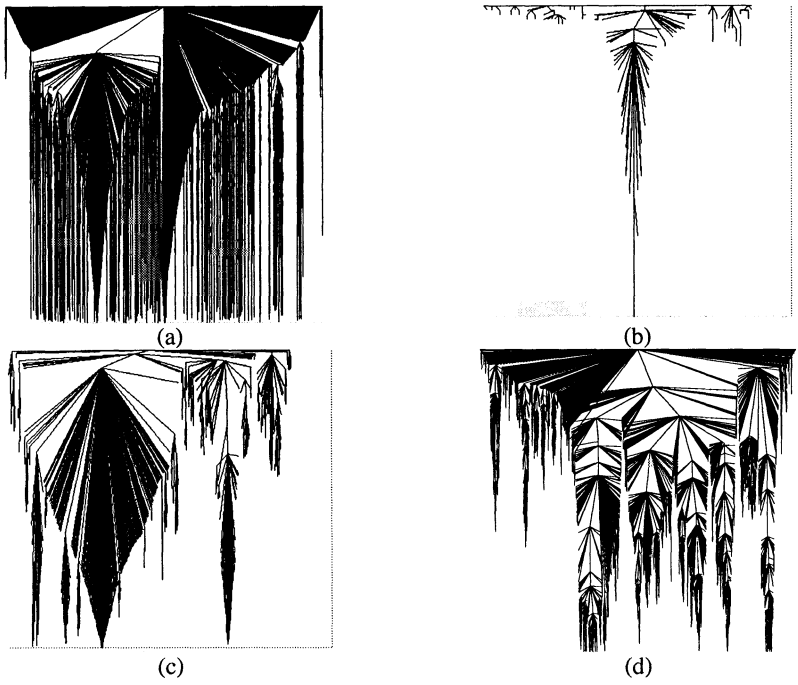


Figure 3. Phylogenic trees of four different evolutionary runs with different parameter settings, exhibiting various degrees of divergence and speciation: (a) extreme divergence, (b) extreme convergence, and (c,d) intermediate levels. Vertical axis represents generation and horizontal axis is arbitrary. The thinned trees depict several hundred generation each.

Sometimes, several tens of generations can pass before any active network is established and connected to an actuator, and any motion is generated at all. The complex stochastic nature of the evolutionary process is both its advantage and its drawback: while at sometimes the process can generate good designs, it can also run into premature convergence and stagnation; many of these dynamic properties of evolutionary computation

have been studied in the literature. Figure 2 shows phylogenic trees of four different evolutionary runs exhibiting various amounts of divergence.

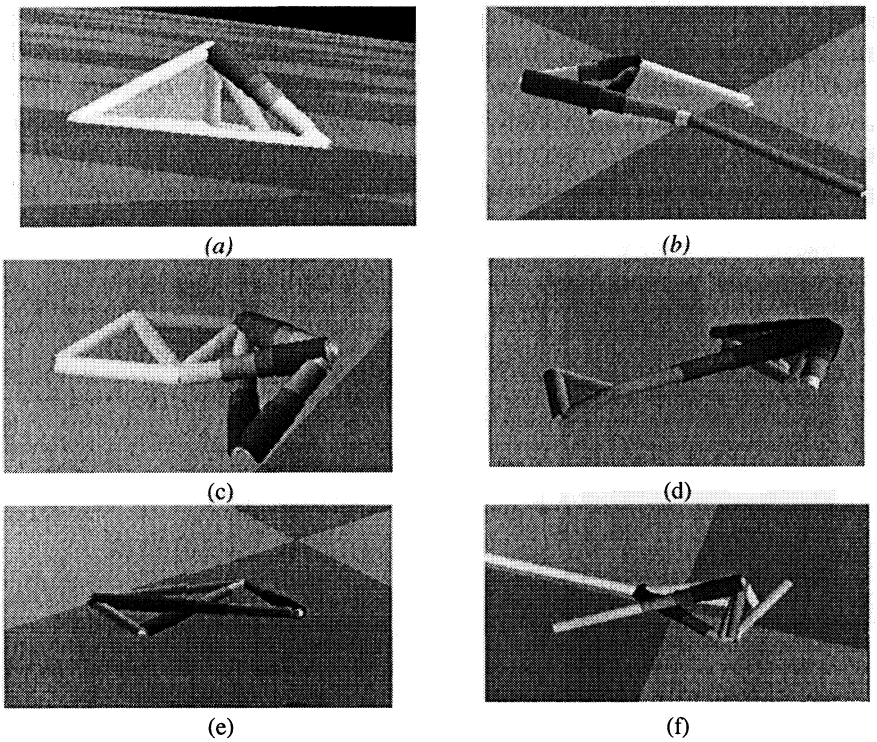


Figure 4. (a) A tetrahedral mechanism that produces hinge-like motion and advances by pushing the central bar against the floor. (b) A mechanism that advances using a sort of bi-pedalism: the left and right limbs are advanced alternating thrusts.(c) This four-actuator mechanism moves its two articulated components to produce crab-like sideways motion. (d) This surprisingly symmetrical machine uses three highly coordinated actuators to advance: while the upper tow limbs push, the central body is retracted, and vice versa. (e) This simple mechanism uses the top bar to delicately shift balance from side to side, shifting the friction point to either side as it creates oscillatory motion and advances. (f) This mechanism created a three dimensional body, from which it pushes an actuator directly onto the floor to create ratcheting motion. It has a few redundant bars dragged on the floor. Motion videos of these robots and others can be viewed at http://www.demo.cs.brandeis.edu./golem

Several evolution runs were carried out. The evolved robots exhibited various methods of locomotion, including crawling, ratcheting and some forms of pedalism. Some are shown in Figure 4, and a sample instance of an entire generation, thinned down to show only different individuals, is shown in Figure 5. Activation signals of a typical neural net are shown in Figure 6.

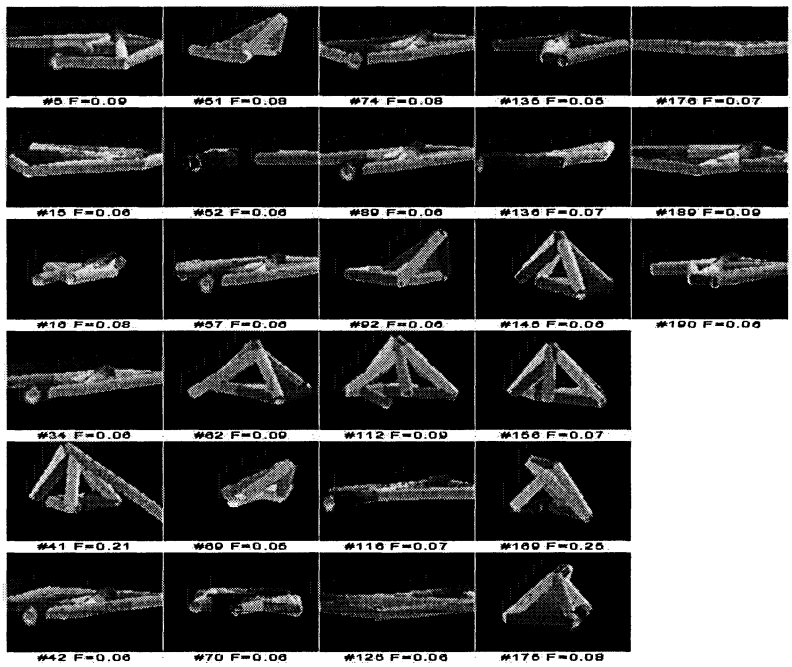


Figure 5. A sample instance of an entire generation, thinned down to show only different individuals. Note the two prevailing species.

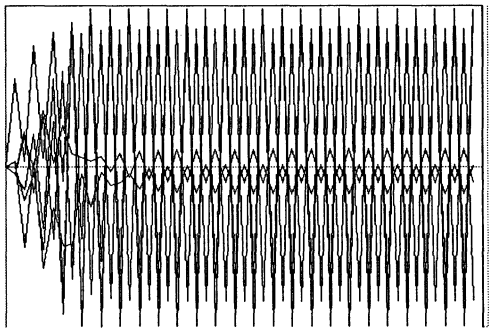
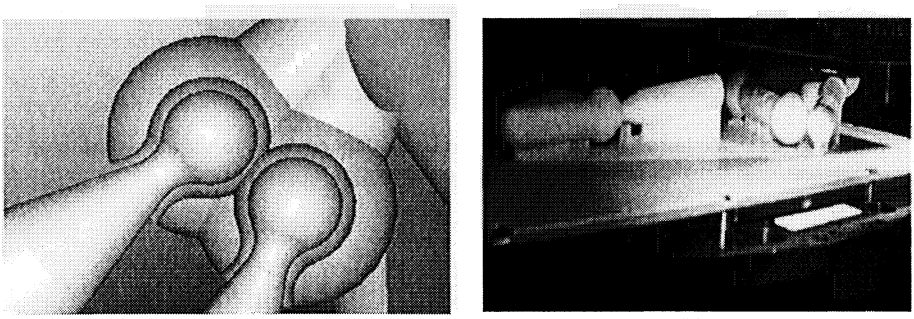


Figure 6. A typical activation pattern of a evolved neural network. Since no sensors are used, the evolved networks are essentially merely pattern generators.

3.4 THE FABRICATION PROCESS

Selected robots out of those with winning performance were then *automatically* replicated into reality: their bodies, which exist only as points and lines, are first converted into a solid model with ball-joints (Figure 7a) and accommodations for linear motors according to the evolved design. The

solidifying stage was automatic but used a hand-coded procedure describing a generic bar, joint, and actuator. The virtual solid bodies were then materialized using commercial rapid prototyping technology. This machine uses a temperature-controlled head to extrude thermoplastic material layer by layer, so that the arbitrarily evolved morphology emerges as a solid three-dimensional structure without tooling or human intervention (Figure 7b). The entire pre-assembled machine is printed as a single unit, with fine plastic supports connecting between moving parts; these supports break away at first motion. The resulting structures contained complex joints that would be difficult to design or manufacture using traditional methods (see Figure 7a). Standard motors are then snapped in, and the evolved neural network is executed to activate the motors. The physical machines (3 to date, Figure 8) then faithfully reproduced their virtual ancestors' behavior in reality.



(a) (b)

Figure 7. (a) Automatically fleshed joints, (b) Replication process

4. Conclusions

The results presented here demonstrate for the first time a complete cycle of fully automated design and manufacture. While both the machines and task we describe in this work are fairly simple from the perspective of what human teams of engineers can produce, and what biological evolution has produced, we have demonstrated for the first time a robotic bootstrap, where robotically designed electromechanical systems can be robotically manufactured. We have carefully minimized human intervention both in the design and in the fabrication stages. Besides snapping in the motors, the only human work was in informing the simulation about the universe it could expect to be manufacturable. We believe these results indicate one path towards fully automated design and manufacture.

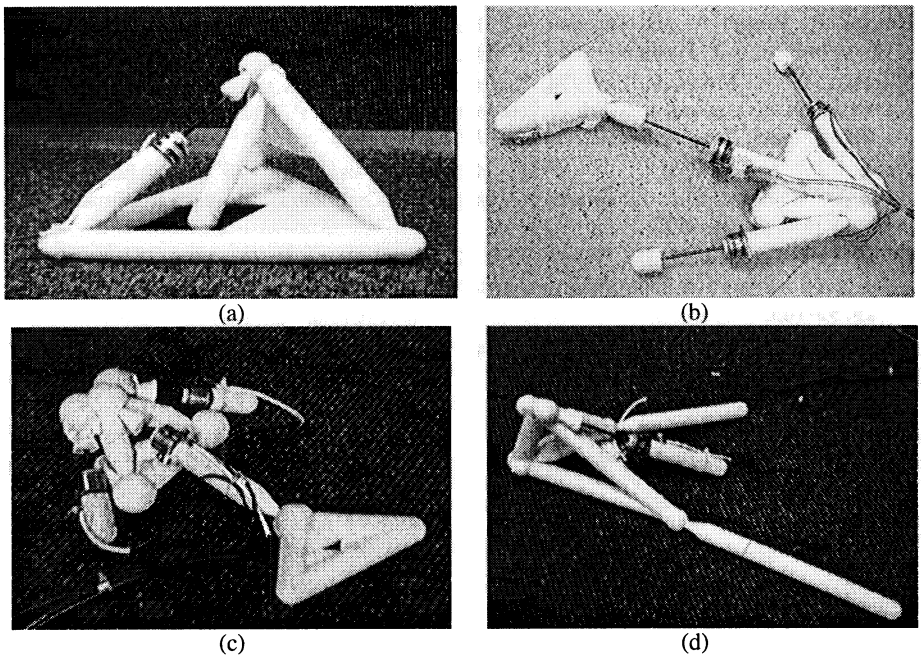


Figure 8. Printed, functioning machines as solutions to the locomotion task: (a)A tetrahedron corresponding to Figure 4a, (b,c) The arrow shaped machine corresponding to Figure 4d, and (d) a mechanism corresponding Figure 4f. Motion videos of these robots and others can be viewed at http://www.demo.cs.brandeis.edu/golem.

Acknowledgements

This research was supported in part by the Defense Advanced Research Projects Administration (DARPA) under Grant DASG60-99-1-0004. Hod Lipson acknowledges the generous support of the Fischbach Postdoctoral Fellowship.

References

Bentley, P. (ed) : 1999, *Evolutionary Design by Computers,* Morgan Kaufmann, San Francisco. pp. 387-403.

Chen, I.-M.: 1994, *Theory and Applications of Modular Reconfigurable Robotic Systems*, Ph.D. thesis, Cal Inst. of Technology.

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

Chocron, O. and Bidaud, P.: 1997, Genetic design of 3D modular manipulators, *Proceedings of the '97 IEEE International Conference on Robotics and Automation,* pp. 223-228.

Dittrich P., Burgel A. and Banzhaf W.: 1998, Learning to move a robot with random morphology, *in* P. Husbands and J. A Meyer. (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* D. Cliff, P. Husbands, J. Meyer and S. Wilson (eds), *From Animals to Animats III*, MIT Press.

Funes, P. and Pollack, J.: 1999, Computer evolution of buildable objects, *in* P. Bentley (ed.), *Evolutionary Design by Computers*. Morgan Kaufmann, San Francisco, pp. 387-403.

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

Galt S., Luk B. L. and 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. and Schultz A.: 1994, An evolutionary approach to learning in robots, *Proceedings of the Machine Learning Workshop on Robot Learning*, New Brunswick, NJ.

Gruau F. and 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. and Cliff D.: 1994, Seeing the light: Artificial evolution, real evolution, *in* Cliff, Husbands, Meyer and Wilson (eds) *Proceedings of 3rd International Conference on Simulation and Adaptive Behavior, From Animals to Animats 3,* MIT Press.

Holland, J.: 1975, *Adaptation in Natural and Artificial Systems*, The University of Michigan Press.

Husbands P. and Meyer J. A.: 1998, *Evolutionary Robotics*, Springer.

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

Kawauchi Y., Inaba, M. and Fukuda, T.: 1995, Genetic evolution and self-organization of cellular robotic system, *Dynamics, Control, Robotics, Design and Manufacturing*, **38** (3), 501-509.

Komosinski, M. and Ulatowski, S.: 1999, Framstics: Towards a simulation of a nature-like world, creatures and evolution, *ECAL '99*, pp. 261-265.

Leger, C.: 1999, *Automated Synthesis and Optimization of Robot Configurations: An Evolutionary Approach, Ph.D. Thesis*, Carnegie Mellon University.

Lund, H., Hallam, J. and Lee, W.: 1997. Evolving robot morphology, *Proceedings of IEEE Fourth International Conference on Evolution*. IEEE Press.

Mahfoud, S. W.: 1995, *Niching Methods for Genetic Algorithms*, Ph.D. Thesis, University of Illinois.

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

Mondada, F., Franzi, E. and Ienne, P.: 1993, Mobile robot miniaturization: A tool for investigation in control algorithms, *Proceedings of the 3rd International Symposium on Experimental Robotics*, Kyoto, Japan.

Moravec, H. P.: 1998, *Robot: Mere Machine to Transcendent Mind*, Oxford University Press.

Naito, T., Odagiri, R., Matsunaga, Y., Tanifuji, M. and 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.*

Salomon, R.: 1996, Increasing adaptivity through evolution strategies, *in* P. Maes, M. Mataric, J. Meyer, J. Pollack and S. Wilson, (eds), *From Animals to Animats: Proceedings of 4th International Conference on Simulation of Adaptive Behavior*, MIT Press.

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

Thompson, A.: 1997, Artificial evolution in the physical world, *in* Gomi (ed.), *Evolutionary Robotics: From Intelligent Robotics to Artificial Life (ER'97)*, AAI Books.

Yim, M.: 1993, A reconfigurable modular robot with many modes of locomotion, *Proceedings of the JSME International Conference on Advanced Mechatronics*, pp. 283-288.