Automatic design and Manufacture of Robotic Lifeforms

Hod Lipson and Jordan B. Pollack

Computer Science Dept., Volen Center for Complex Systems

Brandeis University, Waltham MA 02454, USA

Complex biological forms reproduce by taking advantage of an arbitrarily complex set of auto-catalyzing chemical reactions. Biological life is in control of its own means of reproduction, and this *autonomy of design and manufacture* is a key element which has not yet been understood or reproduced artificially. To this date, robots – a form of artificial life[1]– are still designed laboriously and constructed by teams of human engineers at great cost. Few robots are available because these costs must be absorbed through mass production that is justified only for toys, weapons, and industrial systems like automatic teller machines.

Here we report a set of experiments in which simple electro-mechanical systems evolve from scratch to yield physical locomoting machines. Like biological lifeforms whose structure and function exploit the behaviors afforded by their own chemical and mechanical medium, our evolved creatures take advantage of the nature of their own medium - thermoplastic, motors, and artificial neurons[2]. We thus achieve autonomy of design and construction using evolution in a limited universe physical simulation[3,4] coupled to off-the-shelf rapid manufacturing technology[5]. This is the first time robots have been robotically designed and robotically fabricated.

The field of Artificial Life examines “life as it could be” based on understanding the principles and simulating the mechanisms of real biological forms[6]. Just as airplanes use the same principles as birds, but have fixed wings, artificial lifeforms may share the same principles, but not the same implementation in chemistry. Every feature of living systems seems wondrous until it is understood: Stored energy, autonomous movement, and even animal communication are no longer miracles, as they are replicated in toys using batteries, motors, and computer chips.

Our key claim is that to realize artificial life, *full autonomy* must be attained not only at the level of power and behavior (the goal of robotics, today[7]), but also at the levels of design and fabrication. Only then can we expect synthetic creatures to bootstrap and sustain their own evolution. We thus seek automatically designed and constructed physical artifacts that are (a) functional in the real world, (b) diverse in architecture (possibly each slightly different), and (c) producible in short turn-around time, low cost and large quantities. So far these requirements have not been met[8].

The experiments described here use evolutionary computation for design, and additive fabrication for reproduction. The evolutionary process operates on a population of candidate robots, each composed of some repertoire of building blocks. The evolutionary process iteratively selects fitter machines, creates offspring by adding, modifying and removing building blocks using a set of operators, and replaces them into the population (see *methods* section). Evolutionary computation has been applied to many engineering problems[9,10]. However, studies in the field of evolutionary robotics reported to date involve either entirely virtual worlds[3,4], or, when applied in reality, adaptation of only the control level of manually designed and constructed robots[11,12,13]. These robots have a predominantly fixed architecture, although Lund[14] evolved partial aspects of the morphology, Thompson[15] evolved physical electric circuits for control only, and we evolved static Lego structures, but had to manually construct the resultant designs[16]. Other works involving real robots make use of high-level building blocks comprising significant pre-programmed knowledge[17]. Similarly, additive fabrication technology has been developing in terms of materials and mechanical fidelity[18] but has not been placed under the control of an evolutionary process.

Our approach is based on use of *only elementary building blocks and operators* in both the design and fabrication process. As building blocks are more elementary, any inductive bias associated with them is minimized, and at the same time architectural flexibility is maximized. Similarly, use of elementary building blocks in the fabrication process allows it to be more systematic and versatile. As a theoretic extreme, if we could use only atoms as building blocks, laws of physics as constraints and nano-manipulation for fabrication, the versatility of the design space would be maximized. Earlier reported work used higher-level components and limited architectures (like only tree structures[3,4]) resulted in expedited convergence to acceptable solutions, but at the expense of truncating the design space. Furthermore, these design spaces did not consider manufacturability.

The design space we used was comprised of bars and actuators as building blocks of structure and artificial neurons as building blocks of control. Bars connected with free joints can potentially form trusses that represent arbitrary rigid, flexible and articulated structures as well as multiple detached structures, and emulate revolute, linear and planar joints at various levels of hierarchy. Similarly, sigmoidal neurons can connect to create arbitrary control architectures such as feed-forward and recurrent nets, state machines and multiple independent controllers (like multiple ganglia). Additive fabrication, where structure is generated layer by layer, allows automatic generation of arbitrarily complex physical structures and series of physically different bodies, including any composed of our building blocks. A schematic illustration of a possible architecture is shown in 1. The bars connect to each other through ball-and-socket joints, neurons can connect to other neurons through synaptic connections, and neurons can connect to bars. In the latter case,

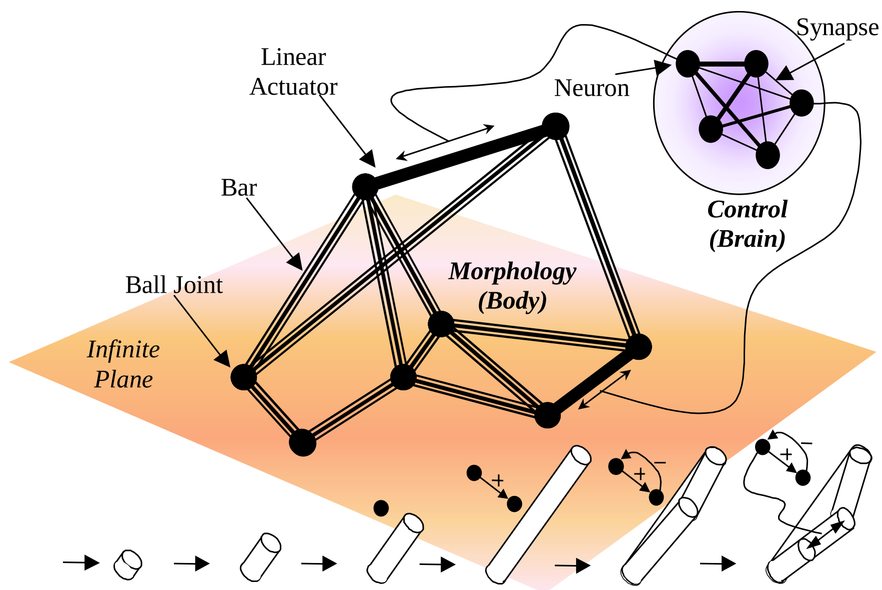


Figure 1. **Schematic illustration of an evolvable robot** Bars connect to each other to form arbitrary trusses; by changing the number of bars and the way they connect, the structural behavior of the truss is modified: some substructures may become rigid, while others may become articulated. Neurons connect to each other via synapses to form arbitrary recurrent neural networks. By changing the synapse weights and activation threshold of the neuron, the behavior of the neuron is modified. By changing the number of neurons and their connectivity, the behavior of the network is modified. Also, we allow neurons to connect to bars: in the same way that a real neuron governs the contraction of muscle tissue, the artificial neuron signal will control the length of the bar by means of a linear actuator. All these changes can be brought about by mutational operators. A sequence of operators will construct a robot and its controller from scratch by adding, modifying and removing building blocks. The sequence at the bottom of the image illustrates an arbitrary progression of operators that create a small bar, elongate it, and split it. Simultaneously, other operators create a neuron, add another neuron, connect them in a loop, and eventually connect one of the neurons to one of the bars. The bar is now an actuator. Since no sensors were used, these robots can only generate patterns and actions, but cannot directly react to their environment.

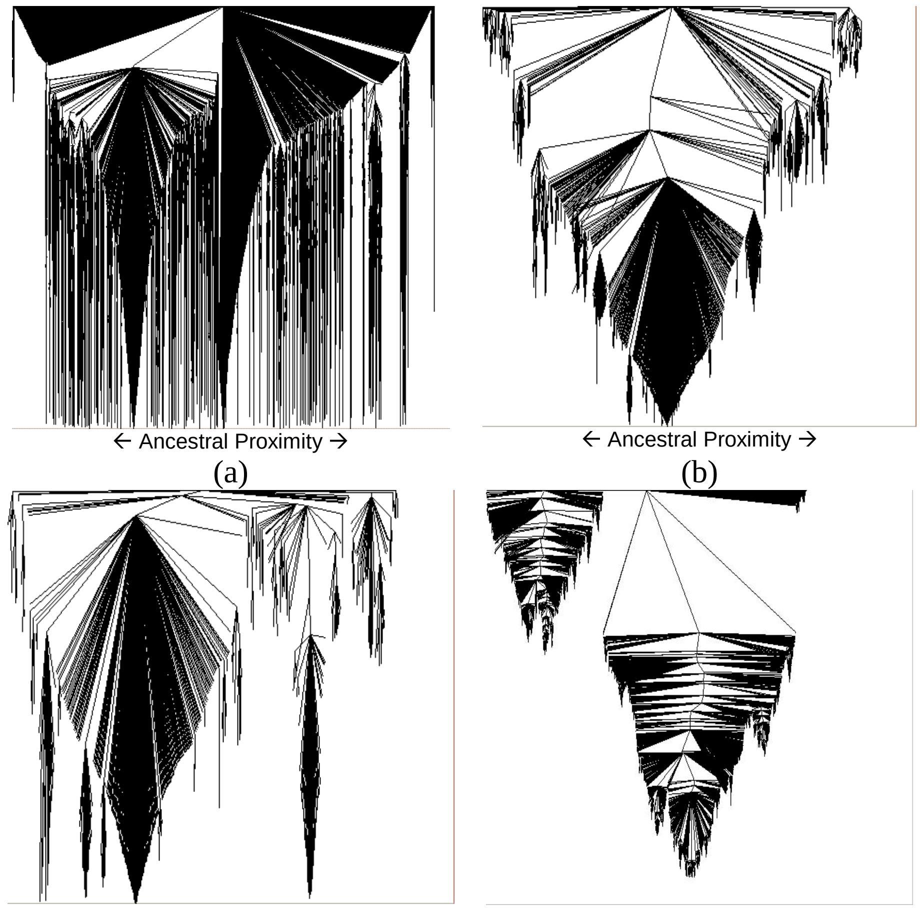


Figure 2. Phylogenetic trees of several different evolutionary runs. Each node in the tree represents an individual and links represent parent-child relationship. Vertical axis represents generations and horizontal axis represents ancestral proximity in terms of hops along the tree necessary to get from one individual to another. All trees originate at a common root denoting an empty robot with zero bars and actuators. Trees exhibit various degrees of divergence and speciation: (a) extreme divergence, resulting from niching methods[37] (b) extreme convergence, resulting from fitness-proportionate selection (c) intermediate level of divergence, typical of earlier stages of fitness-proportionate selection and (d) massive extinction under fitness proportionate selection. The trees were thinned and depict several hundred generations each.

the length of the bar is governed by the output of the neuron, by means of a linear actuator. No sensors were used.

Starting with a population of 200 machines that were comprised initially of zero bars and zero 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. The process iteratively selected fitter machines, created offspring by adding, modifying and removing building blocks, and replaced them into the population (see *methods* section). This process typically continued for 300 to 600

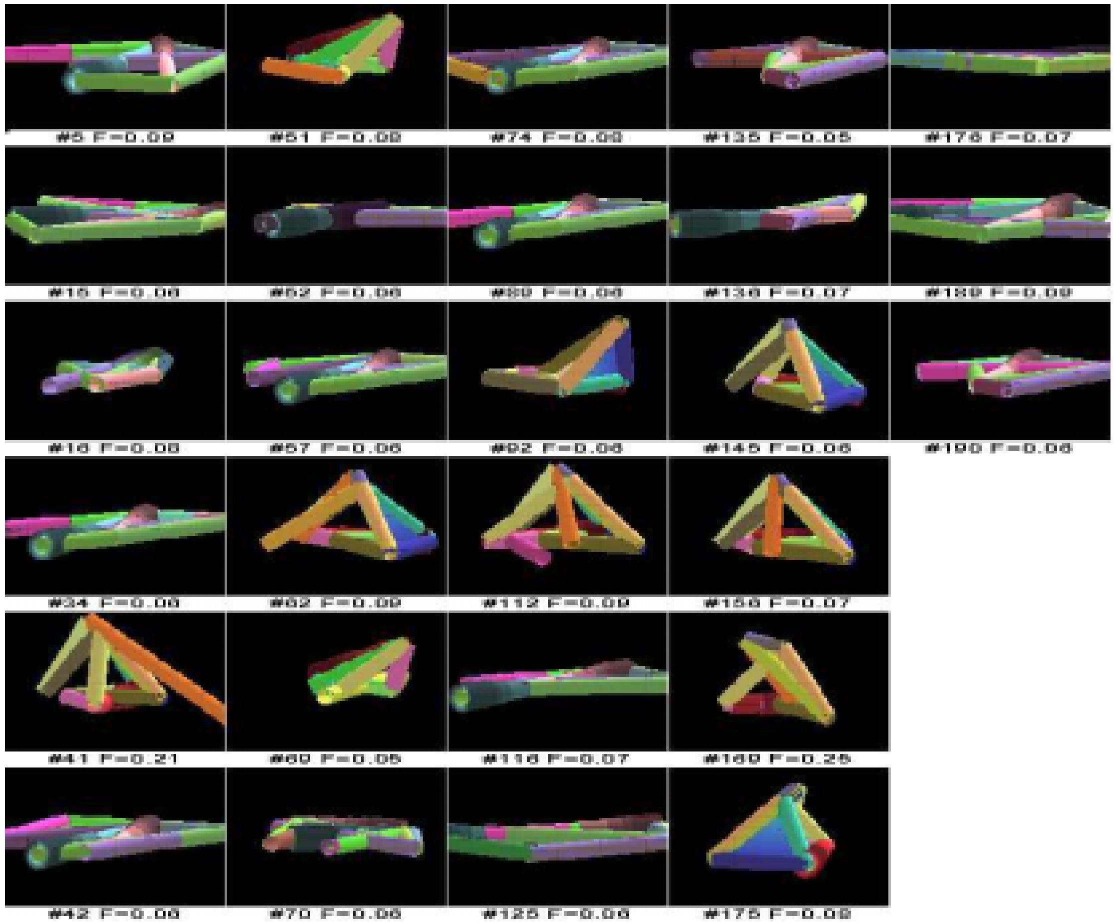


Figure 3. A generation. An arbitrarily sampled instance of an entire generation, thinned down to show only significantly different individuals. The caption under each image provides an arbitrary index (used for reference) and the fitness of that individual. Two subpopulations of robots are observable, each with its own variations: one flat on the ground, and the other containing some elevated structure.

generations. Both body (morphology) and brain (control) were thus co-evolved simultaneously.

The simulator we used for evaluating fitness (see *methods* section) supported quasi-static motion in which each frame is statically stable. This kind of motion is simpler to transfer reliably into reality, yet is rich enough to support low-momentum locomotion. Typically, several tens of generations passed before the first movement occurred. For example, at a minimum, a neural network generating varying output must assemble and connect to an actuator for any motion at all (see sequence in Fig 1 for an example). Various patterns of evolutionary dynamics emerged, some of which are reminiscent of natural phylogenic trees. Figure 2 presents examples of extreme cases of convergence, speciation, and massive extinction. A sample instance of an entire generation, thinned down to unique individuals is shown in Figure 3.

Selected robots out of those with winning performance were then *automatically* replicated into reality: their bodies, which exist only as points and lines, were first converted into a solid model with ball-joints and accommodations for linear motors according to the evolved design (Fig 4a). This solidifying stage was performed by a automatic program which combined pre-designed components describing a generic bar, ball joint, and actuator. The virtual solid bodies were then materialized using commercial rapid prototyping technology (Fig 4b). This machine used a temperature-controlled head to extrude thermoplastic material layer by layer, so that the arbitrarily evolved morphology emerged as a solid three-dimensional structure without tooling or human intervention. The entire pre-assembled machine was printed as a single unit, with fine

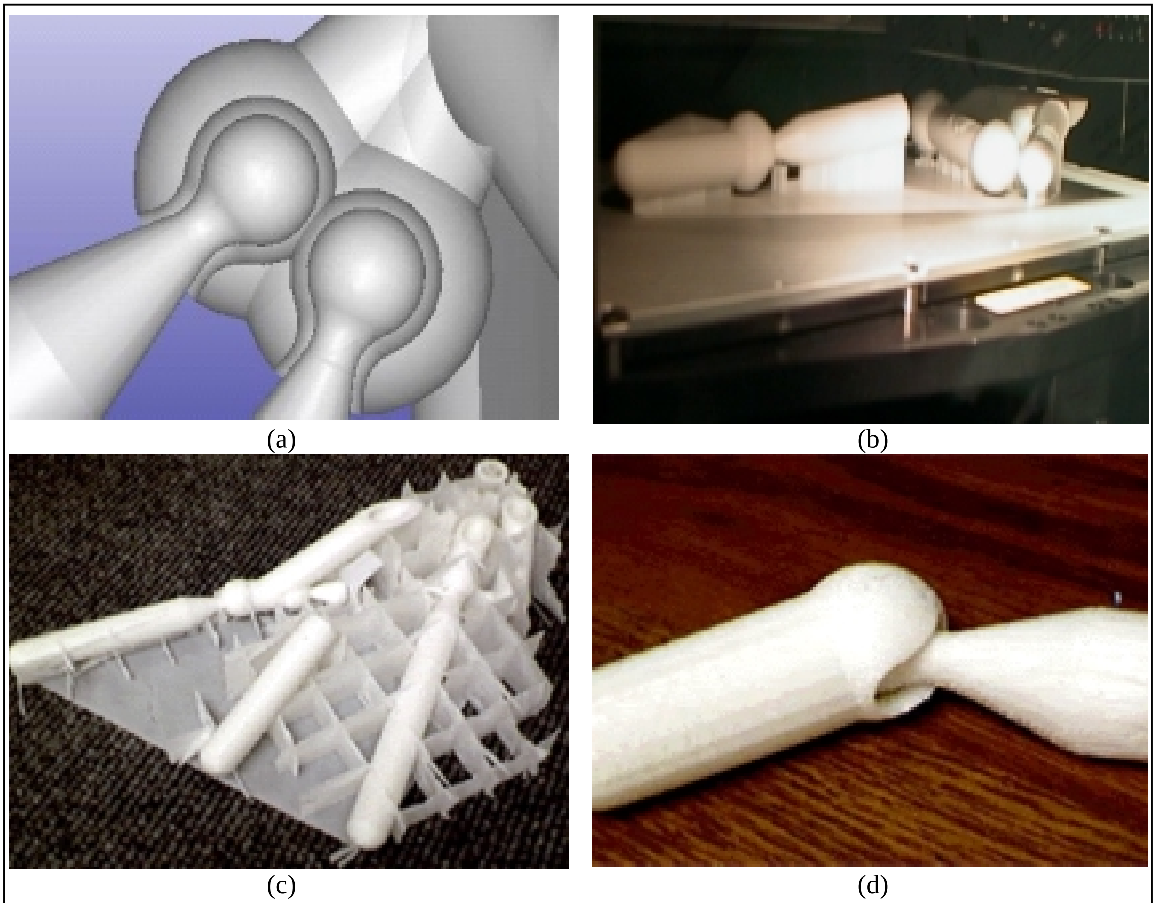


Figure 4. Physical embodiment process: (a) Automatically fleshed joints in virtual space, (b) physical replication process in a rapid prototyping machine that build the three dimensional morphology layer after layer, (c) pre-assembled body in mid print, (d) a close-up image of a joint printed as a single unit. Note ball is printed inside the socket.

Table I: Results

Distance traveled [cm]	Virtual	Physical
Tetrahedron (Figure 5a)	38.5	38.4 (35)
Arrow (Figure 5b)	59.6	22.5 (18)
Pusher (Figure 5c)	85.1	23.4 (15)

Comparison of performance of physical creatures versus their virtual origin. Values are net distance [cm] center of mass traveled over 12 cycles of neural network. Distances in physical column are compensated for scale reduction (actual distance in parentheses). Mismatch in last two rows is primarily due to slipping of limbs on surface.

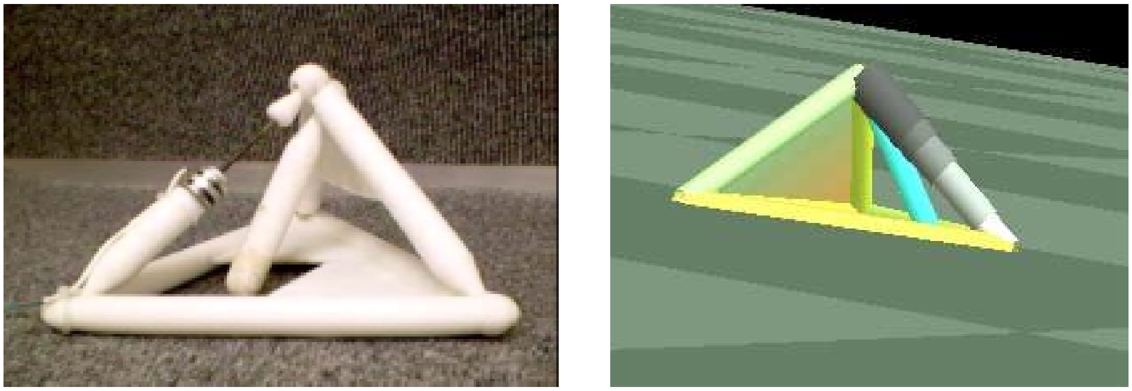
plastic supports connecting between moving parts (Fig 4c); these supports broke away at first motion. The resulting structures contained complex joints that would be difficult to design or manufacture using traditional methods (Fig 4d and Fig 5). Standard stepper motors were then snapped in, and the evolved neural network was executed on a microcontroller to activate the motors. The physical machines (3 to date) then faithfully reproduced their virtual ancestors' behavior in reality (see Table I).

In spite of the relatively simple task and environment (locomotion over an infinite horizontal plane), surprisingly different and elaborate solutions were evolved. Machines typically contained around 20 building blocks, sometimes with significant redundancy (perhaps to make mutation less likely to be catastrophic[19]). Not less surprising was the fact that some (e.g. Fig 5b) exhibited symmetry, which was neither specified nor rewarded for anywhere in the code; a possible explanation is that symmetric machines are more likely to move in a straight line, consequently covering a greater net distance and acquiring more fitness. Similarly, successful designs appear to be robust in the sense that changes to bar lengths would not significantly hamper their mobility. Three samples are shown and described in detail in Figure 5, exploiting principles of ratcheting (5a), anti-phase synchronization (5b) and dragging (5c). Others (not shown here) used a sort of a crawling bi-pedalism, where a body resting on the floor is advanced using alternating thrusts of left and right "limbs". Some mechanisms use sliding articulated components to produce crab-like sideways motion. Other machines used a balancing mechanism to shift friction point from side to side and advance by oscillatory motion. Table I compares the performances of three physical creatures to their virtual ancestors. Note that although overall distance traveled in the 2nd and 3rd cases does not match, in all cases the physical motion was achieved using corresponding mechanical and control implementations. The difference in distance results from slipping of the limbs on the surface, implying that the friction model used in the simulation was not realistic.

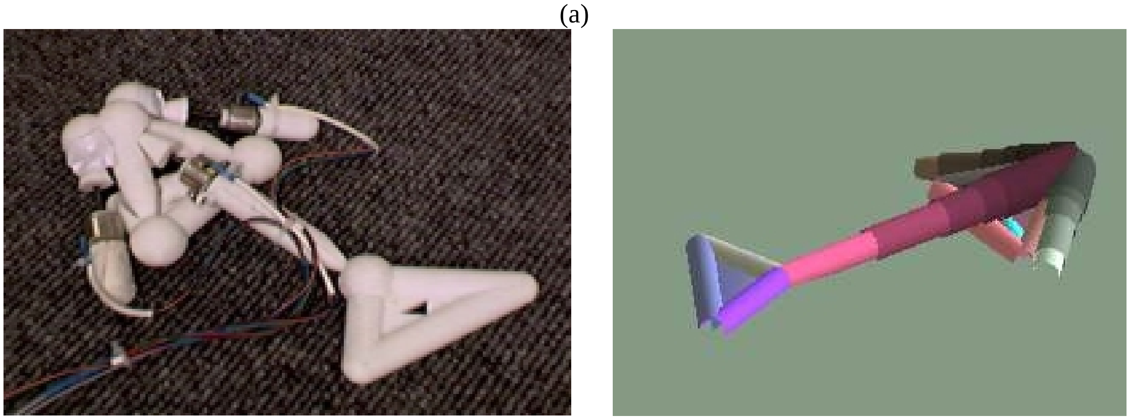
In summary, while both the machines and task we describe in this work are fairly simple from the perspective of what human teams of engineers can produce, and what biological evolution has produced, we have demonstrated for the first time a *robotic bootstrap*, where automatically designed electromechanical systems have been manufactured robotically. 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 that could be manufactured.

Without reference to specific organic chemistry, life is an autonomous design process that is in control of a complex set of chemical factories allowing the generation and testing of physical entities which exploit the properties of the medium of their own construction. Using a different medium, namely off-the-shelf rapid manufacturing, and evolutionary design in simulation, we have made progress towards replicating this autonomy of design and manufacture. This is the first time any artificial evolution system has been connected to an automatic physical construction system. All together, our evolutionary design system, solidification process, and rapid prototyping machine form a primitive "replicating" robot. While there are many, many further steps before this technology is dangerous[20], we believe that if indeed artificial systems are to ultimately interact and integrate with reality, they cannot remain virtual; it is crucial that they cross the simulation-reality gap to learn, evolve[21] and affect the physical world directly[22]. Eventually, the evolutionary process must accept feedback from the live performance of its products.

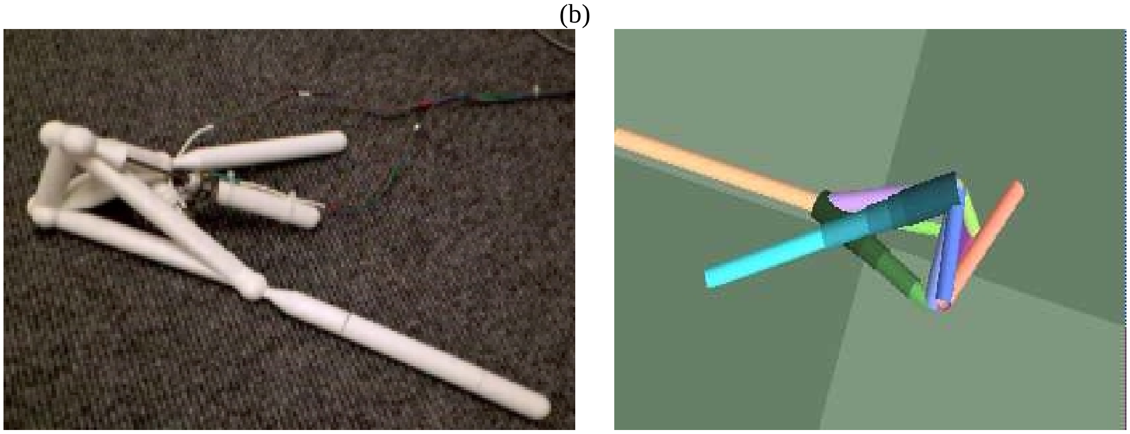
Future work is primarily needed in understanding how more complex modular structures might self-organize, and how these complex structures may transfer into reality under control of the evolutionary process. Technological advances in MEMS, nano-



(a)



(b)



(c)

Fig. 5. Three resulting robots: (a) A tetrahedral mechanism that produces hinge-like motion and advances by pushing the central bar against the floor. (b) This surprisingly symmetric machine uses a 7-neuron network to drive the center actuator in perfect anti-phase with the two synchronized side limb actuators. While the upper two limbs push, the central body is retracted, and vice versa. (c) This mechanism has an elevated body, from which it pushes an actuator down directly onto the floor to create ratcheting motion. It has a few redundant bars dragged on the floor, which might be contributing to its stability. Print times are 22, 12 and 18 hours, respectively. These machines perform in reality is the same way they perform in simulation. Motion videos of these robots and others can be viewed at http://www.demo.cs.brandeis.edu.

fabrication and multi-material rapid prototyping that can embed circuits[23] and actuators[24] in bulk structure, higher fidelity physical simulation, and increased understanding of evolutionary computational processes may pave the way for what Moravec has termed "Escape Velocity"[25].

Methods

Robot representation: A robot is represented by a string of integers and floating point numbers that describe bars, neurons, and their connectivity, as follows:

```
robot := <vertices><bars><neurons><actuators>
vertex := <x,y,z>
bar := <vertex1 index, vertex2 index, relaxed length, stiffness>
neuron := <threshold, synapse coefficients of connections to all neurons>
actuator := <bar index, neuron index, bar range>
```

Evolution process: Experiments were performed using version 1.2 of GOLEM (Genetically Organized Lifelike Electro Mechanics), which can be obtained from http://www.demo.cs.brandeis.edu/golem. We carried out a simulated evolutionary process: The fitness function was defined as the net Euclidean distance that the center-of-mass of an individual has moved over a fixed number (12) of cycles of its neural control. We started with a population of 200 null (empty) individuals. Each experiment used a different random seed. Individuals were then selected, mutated, and replaced into the population in steady-state as follows: The selection functions we tried were random, fitness proportionate or rank proportionate. The mutation operators used to generate an offspring were the following (with probability): Small mutation in length of bar or neuron synaptic weight (0.1), removal/addition of a small dangling bar or unconnected neuron (0.01), split vertex into two and add a small bar or split bar into two and add vertex (0.03), attach/detach neuron to bar (0.03). The dice were rolled until at least one mutation was applied. The mutations were applied directly to the symbolic representation of the phenotype. 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 was inserted into the population by replacing an existing individual. The replacement functions we tried chose individuals to replace either randomly, in inverse-proportion to its fitness, or using similarity proportionate criteria (deterministic crowding[26]). Various permutations of selection-replacement methods are possible; The results we report here were obtained using fitness proportionate selection and random replacement. However, using rank selection instead of fitness proportionate selection, or using random selection with fitness proportionate replacement yields equivalent results. . The process continued for 500-5000 generations (approx 10^5 to 10^6 evaluations overall). The process was carried out 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 (this could be avoided with a generational implementation).

Our evolutionary simulation was based on Evolutionary Srategies[27] and Evolutionary Programming[28], since it directly manipulated continuous valued representations and used only elementary operators of mutation. Alternatively, we could have used Genetic Algorithms[29] and Genetic Programming[30] that introduce cross-over operators sensitive to the structure of the machines, which might change the rate of evolution and lead to replicated structures. We did not form a morphological grammar from which the body is developed[31], but evolved directly on the symbolic representation of the phenotype. And, instead of separating body (morphology) and brain (control) into separate populations, or providing for a "neonatal" stage that might allow us to select for brains that are able to learn to control their bodies, we simply applied selection to bodies and brains as integrated units. This simplified experimental setup followed our focus on completing the simulation and reality loop, but we anticipate that the many techniques that have been developed in evolutionary and co-evolutionary learning[32,33,34] will enrich our results.

Simulation: Both the mechanics and the neural control of a machine were simulated concurrently. The mechanics were simulated using quasi-static motion, where each frame of the motion was assumed to be statically stable. This kind of motion is simple to simulate and easy to induce in reality, yet is rich enough to support various kinds of low-momentum motion like crawling and walking (but not jumping). The model consisted of ball-joined cylindrical bars with true diameters. Each frame was solved by relaxation: An energy term was defined, taking into account elasticity of the bars, potential gravitational energy, and penetration energy of collision and contact. The degrees of freedom of the model (vertex coordinates) were then adjusted iteratively according to their derivatives to minimize the energy term, and the energy was recalculated. Static friction was also modeled. The use of relaxation permitted handling singularities (e.g. snap-through buckling) and under-constrained cases (like a dangling bar). Noise was added to ensure the system does not converge to unstable equilibrium points, and to cover the simulation-reality gap[35]. The material properties modeled correspond to the properties of the rapid prototyping material (E=0.896GPa, ρ=1000Kg/m^3 σ_{yield}=19MPa). The neural network was simulated in discrete cycles. In each cycle, actuator lengths were modified in small increments not larger than 1 cm.

---

[1] Langton C., *Artificial Life,* Addison-Welsley, Redwood City, California, 1989

[2] See, for example, Haykin S., Neural Networks : A Comprehensive Foundation, 2nd Ed., *Prentice-Hall,* 1999

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

4 Komosinski M., Ulatowski S., “Framstics: Towrds a simulation of a nature-like world, creatures and evolution”, *ECAL ’99*, pp. 261-265, 1999

5 Dimos. D, Danforth S.C., Cima M.J., *Solid Freeform and Additive Fabrication*, MRS Symposium, Boston Massachusetts, 1998

6 Smith J. M., “Byte-sized evolution”, *Nature* 355, pp. 772-773, 1992

[7] Swinson,M “Mobile autonomous robot software”, Arlington VA: DARPA, BAA-99-09, 1998

8 Moravec H., “Rise of the Robots”, *Scientific American*, Vol. 281 No. 12, 1999

[9] Bentley P. (Ed.) *Evolutionary Design by Computers*. Morgan Kaufmann, 1999

[10] Embrechts, Mark J. Kewley, Robert Jr. Breneman, Curt *Computationally intelligent data mining for the automated design and discovery of novel pharmaceuticals*, Intell Eng Syst Artif Neural Networks **v 8** , p 397-403 , 1998

11 Floreano, D. and Mondada, F. 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, 1994

12 Husbands P., Meyer J. A., *Evolutionary Robotics*, Springer Verlag, 1998

13 Nolfi S., "Evolving non-trivial behaviors on real-robots: a garbage collecting robot", *Robotics and Autonomous Systems*, 1997

14 Lund, H., Hallam, J., and Lee, W. “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, 1997

15 Thompson A., "Artificial evolution in the physical world", in Gomi (Ed.) *Evolutionary Robotics: From intelligent robotics to artificial life* (ER'97), AAI Books, 1997

16 Funes, P. and Pollack, J. Evolutionary Body Building: Adaptive physical designs for robots.

Artificial Life 4: 337-357, 1998

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

[18] Kochan, A. *Rapid prototyping trends*, Rapid Prototyping J **v 3** : n 4 , p 150-152, 1997

19 Lenski R. E., Charles O., Collier T., Adami C., “Genome Complexity, robustness and genetic interactions in digital organisms”, *Nature* 400, pp. 661-664, 1999

[20] Joy, B., Why the future doesn’t need us. WIRED 8.04, 2000

[21] Watson, Richard A., Ficici, Sevan G. and Pollack, Jordan B. Embodied Evolution: Embodying an Evolutionary Algorithm in a Population of Robots. In ‘99 Congress on Evolutionary Computation. Angeline, Michalewicz, Schoenauer, Yao, Zalzala, eds. IEEE Press, 335-342, 1999

22 Beer R. D., *Intelligence as Adaptive Behavior*, Academic Press, 1990

23 Ziemells K., “Putting it on plastic”, *Nature* 393, pp. 619-620, 1998

[24] Baughman R. H., Cui C., Zakhidov A. A., Iqbal A., Barisci J.N., Spinks G.M., Wallace G.G., Mazzoldi A., Rossi D., Rinzler A.G., Jaschinski O., Roth S., Kertesz M., “Carbon Nanotube Actuators”, *Science*: Vol. 284 (#5418) 1340-1344, 1999

[25] Moravec H., Robot – From mere machine to transcendent mind, Oxford Univ Press, 1999

26 Mahfoud S. W., “Niching methods for genetic algorithms”, *Ph.D. Thesis*, University of Illinois at Urbana-Champaign, 1995

[27] Rechenberg I., Evolutionsstrategie: Optimierung Technischer Systeme nach Prinzipien der Biologischen Evolution, Frommann-Holzboog, Stuttgart, 1973

[28] Fogel L. J., Owensm A. J., Walsh M. J., Artificial Intelligence through Simulated Evolution, John Wiley, NY, 1966

29 Holland J., *Adaptation in natural and artificial systems*, University of Michigan Press, 1975

30 Koza J., *Genetic Programming*, MIT Press, 1992

31 Gruau F., Quatramaran K., "Cellular Encoding for interactive evolutionary robotics", in Husbands and Harvey (Eds.), *Proceedings of 4th Euro. Conf. on Artificial Life*, MIT Press, 1997

32 Fogel, D. Evolving Artificial Intelligence. *Ph.D. thesis*, University of California, San Diego. 1992

33 Hillis, D. “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., 1992

34 Juillé, H. and Pollack, J. B. “Coevolving the "Ideal" Trainer: Application to the Discovery of Cellular Automata Rules”. Proceedings of the *Third Annual Genetic Programming Conference* , Madison, Wisconsin, July 22 - 25, 1998

[35] N. Jakobi, P. Husbands and I. Harvey: *Noise and the Reality Gap: The use of Simulation in Evolutionary Robotics* In *Advances in Artificial Life: Proc. 3rd European Conference on Artificial Life*, Moran, F., Moreno, A., Merelo, J., Chacon, P. (eds.) Springer-Verlag, Lecture Notes in Artificial Intelligence 929 pp. 704-720, 1995