Automatic design and manufacture of robotic lifeforms

Hod Lipson & Jordan B. Pollack

*Computer Science Department, Volen Center for Complex Systems,
Brandeis University, Waltham, Massachusetts 02454, USA*

Biological life is in control of its own means of reproduction, which generally involves complex, autocatalysing chemical reactions. But this autonomy of design and manufacture has not yet been realized artificially[1]. Robots are still laboriously designed and constructed by teams of human engineers, usually at considerable expense. Few robots are available because these costs must be absorbed through mass production, which is justified only for toys, weapons and industrial systems such as automatic teller machines. Here we report the results of a combined computational and experimental approach in which simple electromechanical systems are evolved through simulations from basic building blocks (bars, actuators and artificial neurons); the 'fittest' machines (defined by their locomotive ability) are then fabricated robotically using rapid manufacturing technology. We thus achieve autonomy of design and construction using evolution in a 'limited universe' physical simulation[2,3] coupled to automatic fabrication.

In the field of artificial life, 'life as it could be' is examined on the basis of understanding the principles, and simulating the mechanisms, of real biological forms[4]. Just as aeroplanes use the same principles as birds, but have fixed wings, artificial lifeforms may share the same principles, but not the same implementation in chemistry. Stored energy, autonomous movement, and even animal communication are replicated in toys using batteries, motors and computer chips.

Our central claim is that to realize artificial life, full autonomy must be attained not only at the level of power and behaviour (the goal of robotics, today[5]), but also at the levels of design and fabrication. Only then can we expect synthetic creatures to sustain their own evolution. We thus seek automatically designed and constructed physical artefacts that are functional in the real world, diverse in architecture (possibly each slightly different), and automatically producible with short turnaround time, at low cost and in large quantities. So far these requirements have not been met.

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[6]. However, studies in the field of evolutionary robotics reported to date involve either entirely virtual worlds[2,3], or, when applied in reality, adaptation of only the control level of manually designed and constructed robots[7–9]. These robots have a predominantly fixed architecture, although Lund[10] evolved partial aspects of the morphology, Thompson[11] evolved physical electric circuits for control only, and we evolved static Lego structures, but had to manually construct the resultant designs[12]. Other works involving real robots make use of high-level building blocks comprising significant pre-programmed knowledge[13]. Similarly, additive fabrication technology has been developing in terms of materials and mechanical fidelity[14] but has not been placed under the control of an evolutionary process.

Our approach is based on the 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 the latter to be more systematic and versatile. As a theoretical extreme, if we could use only atoms as building blocks, laws of physics as constraints and nanomanipulation for fabrication, the versatility of the manufacturable design space would be maximized. Earlier reported work that used higher-level components and limited architectures (such as only tree structures[2,3]) resulted in expedited convergence to acceptable solutions, but at the expense of truncating the design space. Furthermore, these design spaces did not consider manufacturability.

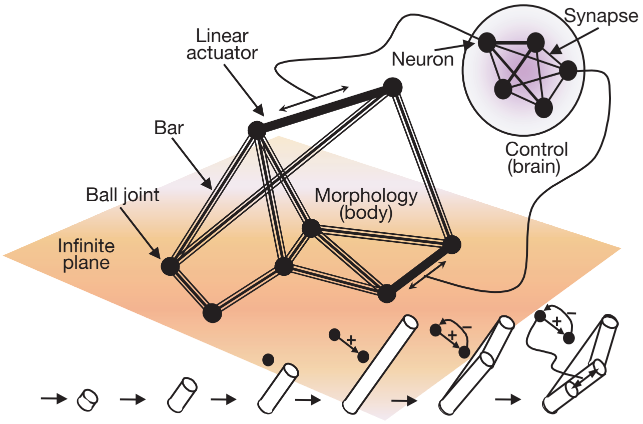

Figure 1 Schematic illustration of an evolvable robot. Bars connect to each other to form arbitrary trusses; by changing the number of bars and the way they connect, the structural behaviour of the truss is modified—some substructures may become rigid, while others may become articulated. Neurons connect to each other via synapses to form arbitrary recurrent neural networks. By changing the synapse weights and the activation threshold of the neuron, the behaviour of the neuron is modified. By changing the number of neurons and their connectivity, the behaviour 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. Because no sensors were used, these robots can only generate patterns and actions, but cannot directly react to their environment.

The design space that we used consisted 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 can represent arbitrary rigid, flexible and articulated structures—as well as multiple detached structures, and emulate revolving, linear and planar joints at various levels of hierarchy. Similarly, artificial 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 for the automatic generation of arbitrarily complex physical structures and the rapid construction of physically different bodies, including any that are composed of our building blocks. A schematic illustration of a possible architecture is shown in Fig. 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, 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 composed 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 that its centre 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). This process typically continued for 300 to 600 generations. Both body (morphology) and brain (control) were thus co-evolved simultaneously.

The simulator that we used for evaluating fitness (see Methods) 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 the sequence in Fig. 1 for an example). Various patterns of evolutionary dynamics emerged, some of which are reminiscent of natural phylogenetic 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 Fig. 3.

Selected (virtual) robots out of those with winning performance were then automatically converted into physical objects: their bodies, represented only as points and lines, were first expanded into solid models with ball joints and accommodations for linear motors according to the evolved designs (Fig. 4a). This 'solidifying' stage was performed by an automatic program that 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

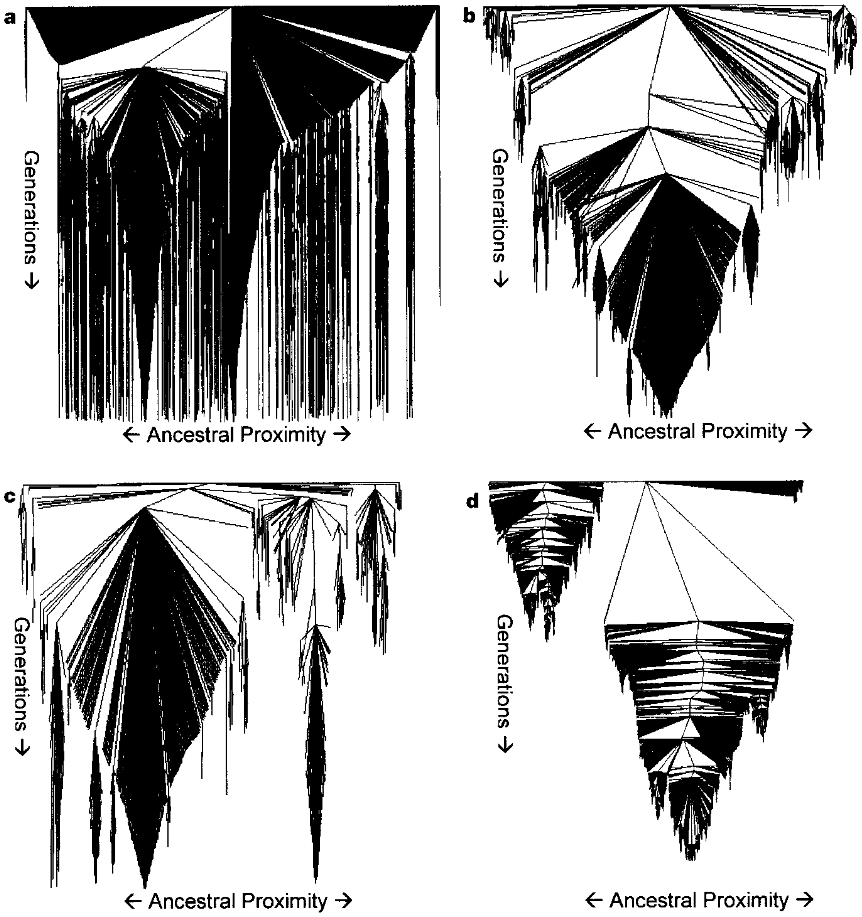

Figure 2 Phylogenetic trees of several different evolutionary runs. Each node in the tree represents an individual and links represent parent–child relationships. The vertical axis represents generations, and the horizontal axis represents ancestral proximity in terms of the hops along the tree necessary 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[22]; **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 are thinned, and depict several hundred generations each.

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 fabricated as a single unit, with fine 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 (Figs 4d and 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 (three to date) then faithfully reproduced their virtual ancestors' behaviour in reality (see Table 1).

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[15]). Not less surprising was the fact that some (for example, 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

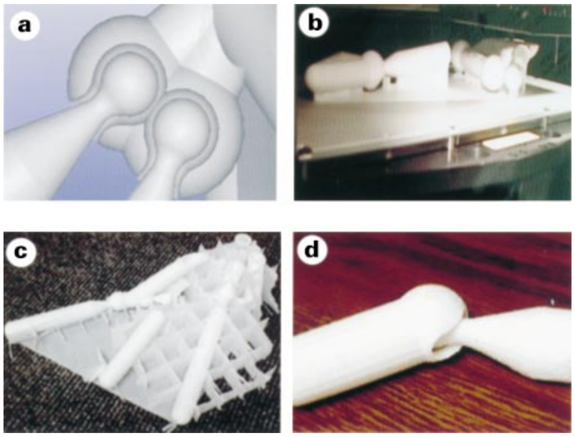

Figure 4 Physical embodiment process. **a**, Automatically 'fleshed' joints in virtual space; **b**, a physical replication process in a rapid prototyping machine that builds the three-dimensional morphology layer after layer; **c**, pre-assembled body in mid print with discardable support structure; **d**, a close-up image of a joint printed as a single unit. The ball is printed inside the socket.

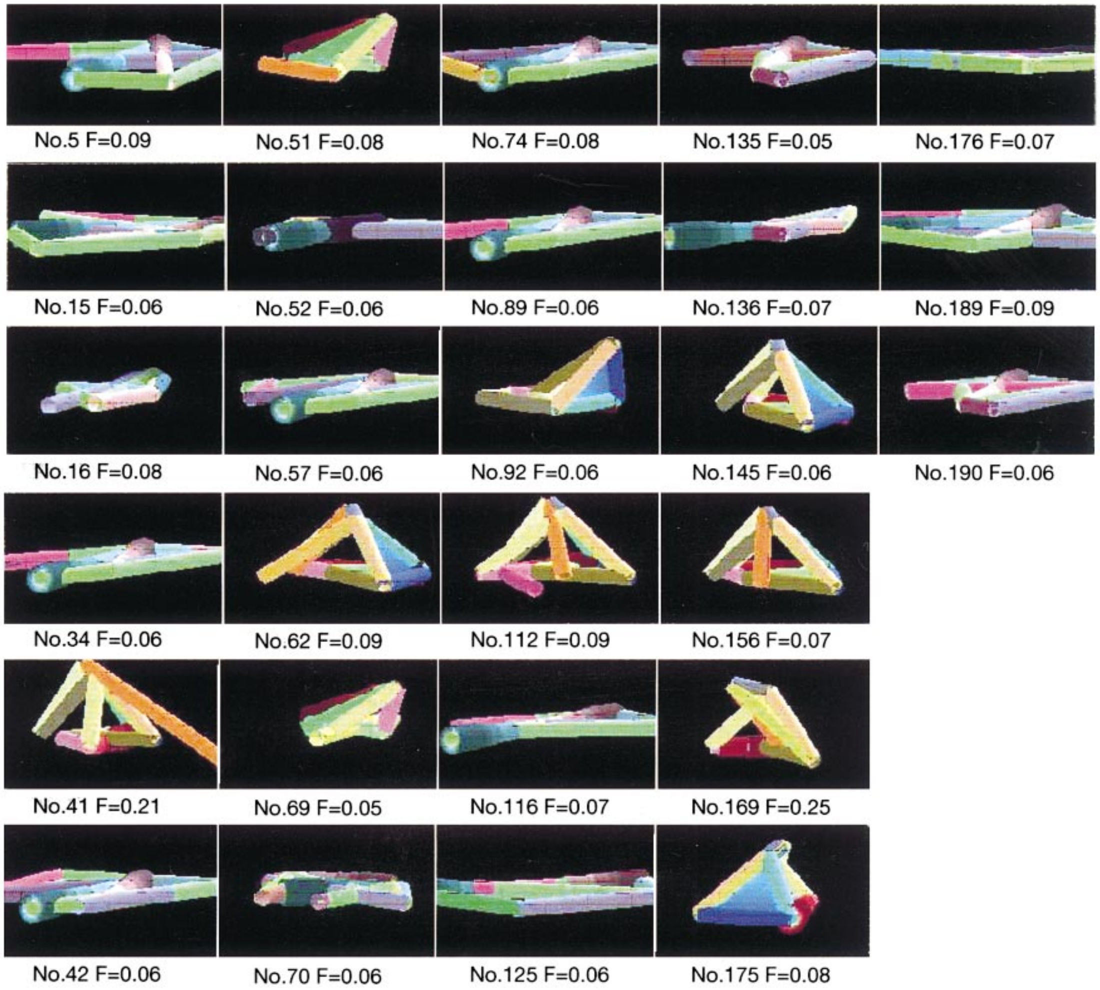

Figure 3 A generation of robots. An arbitrarily sampled instance of an entire generation, thinned down to show only significantly different individuals. The caption under each image provides an arbitrary index number (used for reference) and the fitness of that individual. Two subpopulations of robots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

would not significantly hamper their mobility. Three samples are shown and described in detail in Fig. 5, exploiting principles of ratcheting (Fig. 5a), anti-phase synchronization (Fig. 5b) and dragging (Fig. 5c). Others (not shown) used a sort of a crawling bipedalism, where a body resting on the floor is advanced using alternating thrusts of left and right 'limbs'. Some mechanisms used sliding articulated components to produce crab-like sideways motion. Other machines used a balancing mechanism to shift a friction point from side to side and advance by oscillatory motion. Table 1 compares the performances of three physical machines to their virtual ancestors. We note that although overall distance travelled in the second and third cases does not match, in all cases the physical motion was achieved using corresponding mechanical and control principles. The difference in distance results from the limbs slipping on the surface, implying that the friction model used in the simulation was not realistic.

Although both the machines and the task that we describe here are fairly simple compared with the products of human teams of engineers (and with the products of biological evolution), we have demonstrated a robotic 'bootstrap' process, in which automatically designed electromechanical systems have been manufactured robotically. We have carefully minimized human intervention in both the design and the fabrication stages. Apart from 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 that 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, to our knowledge, the first time any artificial evolution system has been connected to an automatic physical construction system. Taken together, our evolutionary design system, 'solidification' process, and rapid prototyping machine form a primitive replicating robot. Although there are many, many further steps before this technology could become dangerous[16], 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, to evolve[17], and to affect the physical world directly[18]. Eventually, the evolutionary process must accept feedback from the live performance of its products.

Future work is needed primarily 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 micro-electromechanic systems (MEMS), nanofabrication, and multi-material rapid prototyping that can embed circuits[19] and actuators[20] in bulk material, together with higher-fidelity physical simulation and an increased understanding of evolutionary computational processes, may pave the way for the self-sustaining progress that Moravec has termed "escape velocity"[21]. □

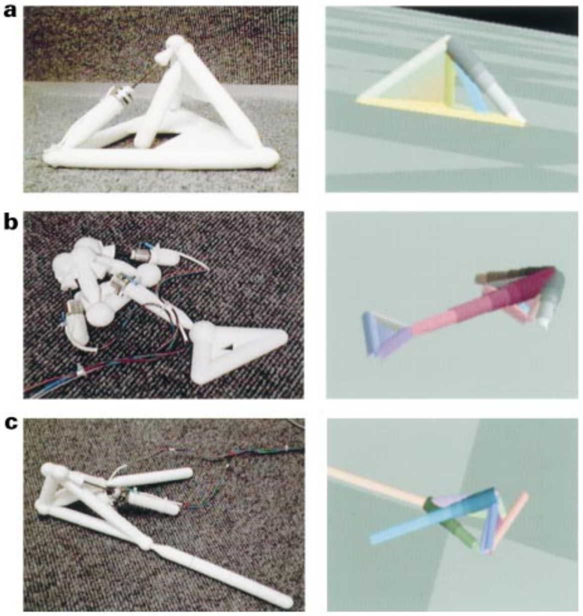

Figure 5 Three resulting robots. Real robots (left); simulated robots (right). **a**, A tetrahedral mechanism that produces hinge-like motion and advances by pushing the central bar against the floor. **b**, This surprisingly symmetric machine uses a seven-neuron network to drive the centre actuator 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 in the same way that they perform in simulation. Motion videos of these robots and others are available: see Supplementary Information.

Table 1 Results

	Distance travelled (cm)	
Machine	Virtual	Physical
Tetrahedron (Fig. 5a)	38.5	38.4 (35)
Arrow (Fig. 5b)	59.6	22.5 (18)
Pusher (Fig. 5c)	85.1	23.4 (15)

Comparison of the performance of physical machines versus their virtual origin. Values are the net distance that the centre of mass of each machine travelled over 12 cycles of neural network. Distances given in the column headed 'physical' are compensated for scale reduction (actual distance is shown in parentheses). The mismatch in the last two rows is primarily due to the slipping of limbs on the surface.

Methods

Robot representation

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

$\text{robot} := \langle\text{vertices}\rangle\langle\text{bars}\rangle\langle\text{neurons}\rangle\langle\text{actuators}\rangle$

$\text{vertex} := \langle x, y, z\rangle$

$\text{bar} := \langle\text{vertex 1 index, vertex 2 index, relaxed length, stiffness}\rangle$

$\text{neuron} := \langle\text{threshold, synapse coefficients of connections to all neurons}\rangle$

$\text{actuator} := \langle\text{bar index, neuron index, bar range}\rangle$

Evolution process

Experiments were performed using version 1.2 of GOLEM (Genetically Organized Lifelike Electro Mechanics), which is available at <http://www.demo.cs.brandeis.edu/golem>. We performed a simulated evolutionary process: the fitness function was defined as the net euclidean distance that the centre-of-mass of an individual moves 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 independently applied with the following probabilities: a small mutation in length of bar or neuron synaptic weight (0.1), the removal or 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 or detach neuron to bar (0.03). At least one mutation was applied. The mutations took place on the symbolic representation of the phenotype. After mutation, a new fitness was 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 their fitness, or using similarity-proportionate criteria (deterministic crowding[22]). 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 yielded equivalent results. The process continued for 300 to 600 generations (approximately 10^5 evaluations overall).

The process was performed 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 strategies[23] and evolutionary programming[24], because it directly manipulated continuous valued representations and used only elementary operators of mutation. Alternatively, we could have used genetic algorithms[25] and genetic programming[26] that introduce crossover operators that are 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[27], 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 set-up 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[28–30] 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 modelled. The use of relaxation permitted handling singularities (for example, 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. The material properties modelled correspond to the properties of the rapid prototyping material (modulus of elasticity, $E = 0.896$ GPa; specific density, $\rho = 1{,}000\,\mathrm{kg\,m^{-3}}$; yield stress, $\sigma_{\mathrm{yield}} = 19$ MPa). The neural network was simulated in discrete synchronized cycles. In each cycle, actuator lengths were modified in small increments not larger than 1 cm.

Received 6 March; accepted 27 June 2000.

1. Langton, C. *Artificial Life* (Addison-Wesley, Redwood City, California, 1989).
2. Sims, K. in *Proc. 4th Artificial Life Conf.* (eds Brooks, R. & Maes, P.) 28–35 (MIT Press, Cambridge, MA, 1994).
3. Komosinski, M. & Ulatowski, S. in *5th Eur. Conf. on Artificial Life ECAL '99* (eds Floreano, D. *et al.*) 261–265 (Springer, Berlin, 1999).
4. Smith, J. M. Byte-sized evolution. *Nature* **355,** 772–773 (1992).
5. Swinson, M. *Mobile Autonomous Robot Software* (Report BAA-99-09, DARPA, Arlington, Virginia, 1998).
6. Bentley, P. (ed.) *Evolutionary Design by Computers* (Morgan Kaufmann, San Francisco, 1999).
7. Floreano, D. & Mondada, F. in *From Animals to Animats III* (eds Cliff, D., Husbands, P., Meyer, J. & Wilson, S.) 421–430 (MIT Press, Cambridge, MA, 1994).
8. Husbands, P. & Meyer, J. A. *Evolutionary Robotics* (Springer, Berlin, 1998).
9. Nolfi, S. Evolving non-trivial behaviors on real-robots: a garbage collecting robot. *Robotics and Autonomous Systems*, **22,** 187–198 (1992).
10. Lund, H., Hallam, J. & Lee, W. in *Proc. IEEE 3rd Int. Conf. on Evolutionary Computation* (eds Fukuda, T. *et al.*) 384–389 (IEEE Press, Piscataway, NJ, 1996).
11. Thompson, A. in *Evolutionary Robotics: From Intelligent Robotics to Artificial Life* (ER'97) (ed. Gomi, T.) 101–125 (AAI Books, Ontario, 1997).
12. Funes, P. & Pollack, J. Evolutionary body building: adaptive physical designs for robots. *Artif. Life* **4,** 337–357 (1998).
13. Leger, C. *Automated Synthesis and Optimization of Robot Configurations: An Evolutionary Approach.* Thesis, Carnegie Mellon Univ. (1999).
14. Kochan, A. Rapid prototyping trends. *Rapid Prototyp. J.* **3,** 150–152 (1997).
15. Lenski, R. E., Ofria, C., Collier, T. & Adami, C. Genome complexity, robustness and genetic interactions in digital organisms. *Nature* **400,** 661–664 (1999).
16. Joy, B. Why the future doesn't need us. *WIRED Magazine* **8(4)**, 238–264 (2000).
17. Watson, R. A., Ficici, S. G. & Pollack, J. B. in *'99 Congress on Evolutionary Computation* (eds Angeline, P. *et al.*) 335–342 (IEEE Press, New York, 1999).
18. Beer, R. D. *Intelligence as Adaptive Behavior* (Academic, Boston, 1990).
19. Ziemelis, K. Putting it on plastic. *Nature* **393,** 619–620 (1998).
20. Baughman, R. H. *et al.* Carbon nanotube actuators. *Science* **284,** 1340–1344 (1999).
21. Moravec, H. *Robot—From Mere Machine to Transcendent Mind* (Oxford Univ. Press, 1999).
22. Mahfoud, S. W. *Niching Methods for Genetic Algorithms.* Thesis, Univ. Illinois at Urbana-Champaign (1995).
23. Rechenberg, I. *Evolutionsstrategie: Optimierung Technischer Systeme nach Prinzipien der Biologischen Evolution* (Frommann-Holzboog, Stuttgart, 1973).
24. Fogel, L. J., Owens, A. J. & Walsh, M. J. *Artificial Intelligence through Simulated Evolution* (Wiley, New York, 1966).
25. Holland, J. *Adaptation in Natural and Artificial Systems* (Univ. Michigan Press, Ann Arbor, 1975).
26. Koza, J. *Genetic Programming* (MIT Press, Cambridge, MA, 1992).
27. Gruau, F. & Quatramaran, K. in *Proc. 4th Eur. Conf. on Artificial Life* (eds Husbands, P. and Harvey, I.) (MIT Press, Cambridge, MA, 1997).
28. Chellapilla, K. & Fogel, D. Evolution, neural networks, games and intelligence. *Proc. IEEE* **87,** 1471–1496 (1999).
29. Hillis, D. in *Artificial Life II* (eds Langton, C., Taylor, J. F. & Rasmussen, S.) 313–322 (Addison-Wesley, Reading, Massachusetts, 1992).
30. Pollack, J. B. & Blair, A. D. Co-evolution in the successful learning of backgammon strategy. *Machine Learning* **32,** 225–240 (1998).

Supplementary information is available on *Nature*'s World-Wide Web site (http://www.nature.com) or from the London editorial office of *Nature*.

Acknowledgements

We thank the DEMO Lab members for useful discussions: P. Funes, R. Watson, O. Melnik, S. Ficici, G. Hornby, E. Sklar & S. Levy. We also thank K. Quigley and G. Widberg for technical assistance. This research was partially supported by the Defense Advanced Research Projects Agency and by the Fischbach Postdoctoral Fellowship.

Correspondence and requests for materials should be addressed to the authors (e-mail: lipson@cs.bradeis.edu or pollack@cs.brandeis.edu).