Evolutionary Techniques in Physical Robotics

Jordan B. Pollack, Hod Lipson, Sevan Ficici, Pablo Funes, Greg Hornby, and Richard A. Watson

Computer Science Dept., Brandeis University, Waltham, MA 02454, USA,
pollack@cs.brandeis.edu,
http://www.demo.cs.brandeis.edu

Abstract. Evolutionary and coevolutionary techniques have become a popular area of research for those interested in automated design. One of the cutting edge issues in this field is the ability to apply these techniques to real physical systems with all the complexities and affordances that such systems present. Here we present a selection of our work each of which advances the richness of the evolutionary substrate in one or more dimensions. We overview research in four areas: a) High part-count static structures that are buildable, b) The use of commercial CAD/CAM systems as a simulated substrate, c) Dynamic electromechanical systems with complex morphology that can be built automatically, and d) Evolutionary techniques distributed in a physical population of robots.

1 Introduction

The field of Robotics today faces a practical problem: most problems in the physical world are too difficult for the current state of the art. The difficulties associated with designing, building and controlling robots have led to a stasis [1] and robots in industry are only applied to simple and highly repetitive manufacturing tasks.

Even though sophisticated teleoperated machines with sensors and actuators have found important applications (exploration of inaccessible environments for example), they leave very little decision, if at all, to the on-board software [2]. Hopes for autonomous robotics were raised high several times in the past — the early days of AI brought us Shakey the robot [3], and the eighties the "subsumption architecture" [4]. Yet today, fully autonomous robots with minds of their own are not a reality.

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

The focus of our research is thus how to automate the integrated design of bodies and brains using a coevolutionary learning approach. Brain/body coevo-

J. Miller et al. (Eds.): ICES 2000, LNCS 1801, pp. 175–186, 2000.
© Springer-Verlag Berlin Heidelberg 2000

lution is a popular idea, but evolution of robot bodies is usually restricted to adjusting a few morphological parameters in an otherwise fixed, human-engineered automaton [5–7]. We propose that the key is to evolve both the brain and the body, simultaneously and continuously, from a simple controllable mechanism to one of sufficient complexity for a task. We then require a replication process, that brings an exact copy of the evolved machine into reality. Finally, the transfer between a simulated environment and reality — the "reality gap" — needs further embodied adaptation.

We see three technologies that are maturing past threshold to make this possible. One is the increasing fidelity of advanced mechanical design simulation, stimulated by profits from successful software competition [8]. The second is rapid, one-off prototyping and manufacture, which is proceeding from 3D plastic layering to stronger composite and metal (sintering) technology [9]. The third is our understanding of coevolutionary machine learning in design and intelligent control of complex systems [10–12].

2 Coevolution

Coevolution, when successful, dynamically creates a series of learning environments each slightly more complex than the last, and a series of learners which are tuned to adapt in those environments. Sims' work [13] on body-brain coevolution and the more recent Framsticks simulator [14] demonstrated that the neural controllers and simulated bodies could be coevolved. The goal of our research in coevolutionary robotics is to replicate and extend results from virtual simulations like these to the reality of computer designed and constructed special-purpose machines that can adapt to real environments.

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

It is not feasible that controllers for complete structures could be evolved (in simulation or otherwise) without first evolving controllers for simpler constructions. Compared to the traditional form of evolutionary robotics [15–19] which serially downloads controllers into a piece of hardware, it is relatively easy to explore the space of body constructions in simulation. Realistic simulation is also crucial for providing a rich and nonlinear universe. However, while simulation creates the ability to explore the space of constructions far faster than real-world building and evaluation could, transfer to real constructions is problematic. Because of the complex emergent interactions between a machine and its environment, learning and readaptation must occur in "embodied" form as well [20–22].

3 Research Thrusts

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

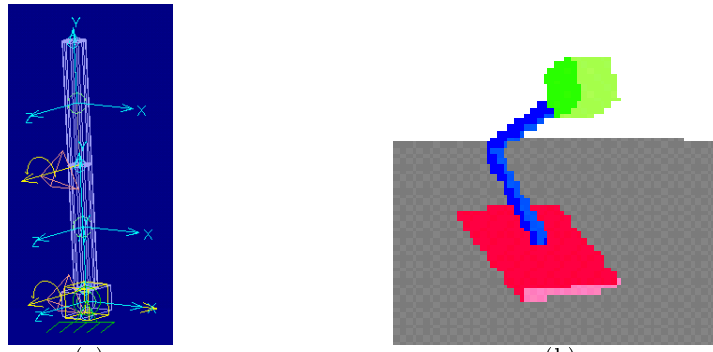
3.1 Evolution in Simulation

We have been doing evolution of neural-network controllers inside realistic CAD simulations as a prelude to doing body reconfiguration and coevolution. Our lab is using CAD/CAM software package that comprises a feature based solid-modeling system [1]. Widely used in industry, it includes a mechanical simulation component that can simulate the function of real-world mechanisms, including gears, latches, cams and stops. This program has a fully articulated development interface to the C programming language, which we have used in order to interface its models to our evolutionary recurrent neural network software.

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

[1] Parametric Technology Corporation's *Pro/Engineer*.

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



(a) (b)

Fig. 1. Commercial CAD models for which we evolved recurrent neural net controllers: (a) a two-segment inverted pendulum (b) A Luxo lamp.

Some of the ways to achieve continuous body deformation are:

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

3.2 Buildable Simulation

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

- Representation —- should cover a universal space of mechanisms.
- Conservative — because simulation is never perfect, it should preserve a margin of safety.
- Efficient — it should be quicker to test in simulation than through physical production and test.

- Buildable — results should be convertible from a simulation to a real object.

One approach is to custom-build a simulator for modular robotic components, and then evolve either centralized or distributed controllers for them. In advance of a modular simulator with dynamics, we recently built a simulator for (static) Lego bricks, and used very simple evolutionary algorithms to create complex Lego structures, which were then manually constructed [23–25].

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

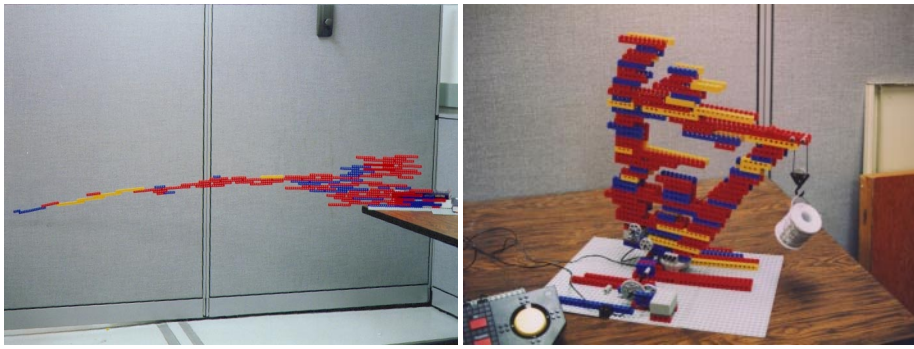


Fig. 2. Photographs of the FAD Lego Bridge (Cantilever) and Crane (Triangle). Photographs copyright Pablo Funes & Jordan Pollack, used by permission.

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

3.3 Evolution and construction of electromechanical systems

The next step is to add dynamics to modular buildable physical components. We are experimenting with a new process in which both robot morphology and control evolve in simulation and then replicate automatically into reality. The robots are comprised of only linear actuators and sigmoidal control neurons embodied in an arbitrary thermoplastic body. The entire configuration is evolved for a particular task and selected individuals are printed pre-assembled (except

motors) using 3D solid printing (rapid prototyping) technology, later to be recycled into different forms. In doing so, we establish for the first time a complete physical evolution cycle. In this project, the evolutionary design approach assumes two main principles: (a) to minimize inductive bias, we must strive to use the lowest level building blocks possible, and (b) we coevolve the body and the control, so that that they stimulate and constrain each other.

We use arbitrary networks of linear actuators and bars for the morphology, and arbitrary networks of sigmoidal neurons for the control. Evolution is simulated starting with a soup of disconnected elements and continues over hundreds of generations of hundreds of machines, until creatures that are sufficiently proficient at the given task emerge. 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 [26,27].

Several evolution runs were carried out for the task of locomotion. Fitness was awarded to machines according to the absolute average distance traveled over a specified period of neural activation. The evolved robots exhibited various methods of locomotion, including crawling, ratcheting and some forms of pedalism.

Selected robots are then replicated into reality: their bodies are first fleshed to accommodate motors and joints, and then copied into material using rapid prototyping technology. Temperature-controlled print head extrudes thermoplastic material layer by layer, so that the arbitrarily evolved morphology emerges pre-assembled as a solid three- dimensional structure without tooling or human intervention. Motors are then snapped in, and the evolved neural network is activated (figure 4). The robots then perform in reality as they did in simulation.

3.4 Embodied Evolution

Once a robot is built, it may well be necessary to "fine-tune" adaptation in the real world. Our approach is to perform adaptation via a decentralized evolutionary algorithm that is distributed and embodied within a population of robots. The distributed and asynchronous operation of this evolutionary method allows the potential for being scaled to very large populations of robots, on the order of hundreds or thousands, thus enabling speedup that is critical when using evolution in real robots.

Technologically, this introduces two main problems: long-term power and reprogramming [28,29]. Many robots' batteries last only for a few hours, and robots typically have to be attached to a PC for new programs to be uploaded. In order to do large group robot learning experiments, we have designed a continuous-power floor system, and utilized infra-red (IR) communications to transfer programs between robots. We are thus able to run a population of learning robots battery-free and wire-free for days at a time (figure 5). Evolution is not directed

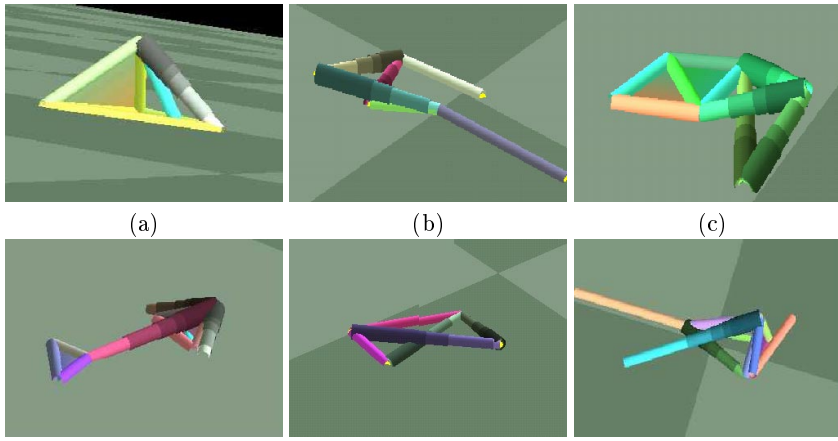


Fig. 3. (a) A tetrahedral mechanism that produces hinge-like motion and advances by pushing the central bar against the floor. (b) Bi-pedalism: the left and right limbs are advanced in alternating thrusts. (c) Moves its two articulated components to produce crab-like sideways motion. (d) While the upper two 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 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.

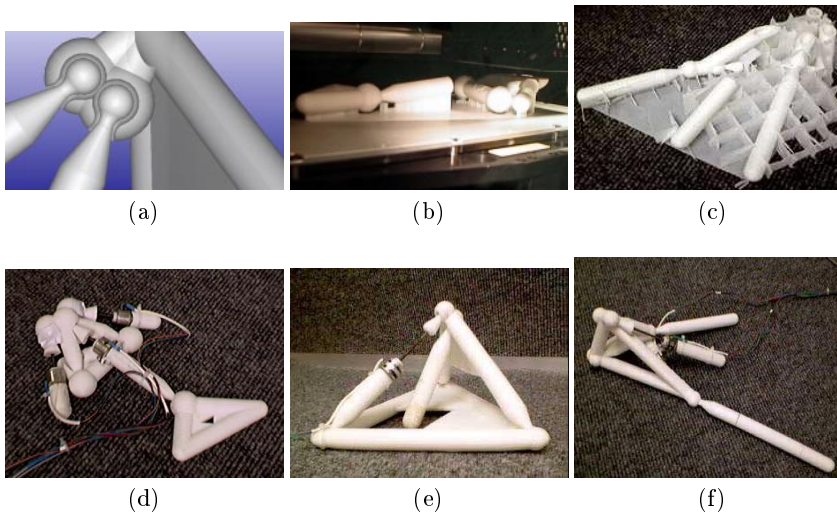
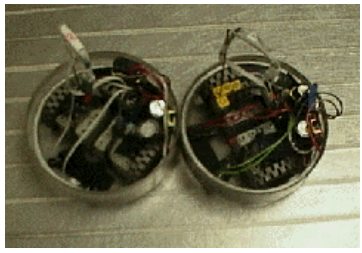
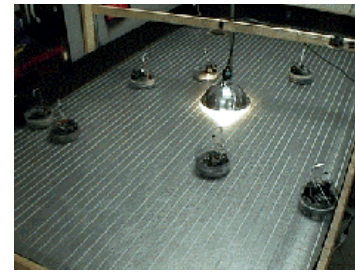


Fig. 4. (a) Fleshed joints, (b) replication progress, (c) pre-assembled robot (figure 3f), (d,e,f) final robots with assembled motors

by a central off-board computer that installs new programs to try out, but rather is distributed into the behavior of all the robots. The robots exchange program specifications with each other and this "culture" is used to 'reproduce' the more successful behaviors and achievement of local goals.



(a)



(b)

Fig. 5. 4" diameter robot picks up power from its environment and learns while on-line.

The control architecture is a simple neural network and the specifications for it are evolved online. Each robot tries parameters for the network and evaluates its own success. The more successful a robot is at the task, the more frequently it will broadcast its network specifications via its local-range IR communications channel. If another robot happens to be in range of the broadcast, it will adopt the broadcast value with a probability inversely related to its own success rate. Thus, successful robots attempt to influence others, and resist the influence of others, more frequently than less successful robots. We have shown this paradigm to be robust both in simulation and in real robots, allowing for the parallel, asynchronous evolution of large populations of robots with automatically developed controllers. These controllers compare favorably to human designs, and often surpass them when human designs fail to take all important environmental factors into account. Figure 6 shows averaged runs of the robots in a light seeking task, comparing evolved controllers to random and human-designed ones.

Our research goals in this area involve group interactive tasks based on multi-agent systems, such as group pursuit and evasion, box pushing, and team games. These domains have been out of reach of traditional evolutionary methods and typically are approached with hand-built (non-learning) controller architectures [30–33]. Work that does involve learning typically occurs in simulation [34–37], or in relatively simple physical domains [38–42].

4 Conclusion

Can evolutionary and coevolutionary techniques be applied to real physical systems? In this paper we have presented a selection of our work each of which addresses physical evolutionary substrate in one or more dimensions. We have

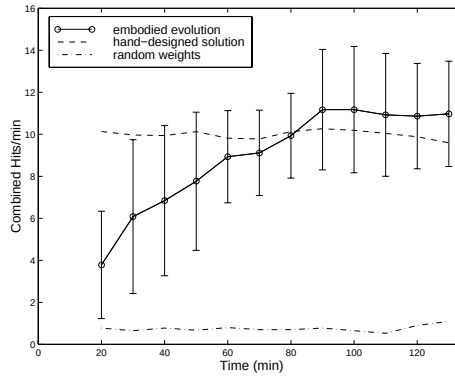


Fig. 6. Averaged runs of robots in a "light gathering" task, with various controllers.

overviewed research in handling high part-count static structures that are buildable, use of commercial CAD/CAM systems as a realistic simulated substrate, dynamic electromechanical systems with complex morphology that can be built automatically, and evolutionary techniques distributed in a physical population of robots.

Our long-term vision is that both the morphology and control programs for self-assembling robots arise directly through hardware and software coevolution: primitive active structures that crawl over each other, attach and detach, and accept temporary employment as supportive elements in "corporate" beings can accomplish a variety of tasks, if enough design intelligence is captured to allow true self-configuration rather than human re-deployment and reprogramming. When tasks cannot be solved with current parts, new elements are created through fully automatic design and rapid prototype manufacturing.

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

Acknowledgements

This research was supported in part by the National Science Foundation (NSF), the office of Naval Research (ONR), and the Defense Advanced Research Projects Agency (DARPA). Thanks to the MIT Media Lab for supplying the microcontrollers used in our embodied evolution experiments.

References

[1] Moravec, H.P.: Rise of the robots. Scientific American (1999) 124–135
[2] Morrison, J., Nguyen, T.: On-board software for the mars pathfinder microrover. In: Proceedings of the Second IAA International Conference on Low-Cost Planetary Missions, John Hopkins University Applied Physics Laboratory (1996)
[3] Nilsson, N.J.: A mobile automaton: An application of artificial intelligence techniques. In: Proceedings of the International Joint Conference on Artificial Intelligence. (1969) 509–520
[4] Brooks, R.: Intelligence without representation. Artificial Intelligence **47** (1991) 139–160
[5] Cliff, D., Noble, J.: Knowledge-based vision and simple visual machines. Philosophical Transac tions of the Royal Society of London: Series B **352** (1997) 1165–1175
[6] Lee, W., Hallam, J., Lund, H.: 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 (1996) 384–389
[7] Lund, H., Hallam, J., Lee, W.: Evolving robot morphology. In: Proceedings of IEEE Fourth International Conference on Evolutionary Computation, IEEE Press (1997) 197–202
[8] Sincell, M.: Physics meets the hideous bog beast. Science **286** (1999) 398–399
[9] Dimos, D., Danforth, S., Cima, M.: Solid freeform and additive fabrication. In Floro, J.A., ed.: Growth instabilities and decomposition during heteroepitaxy, Elsevier (1999)
[10] Pollack, J.B., Blair, A.D.: Coevolution in the successful learning of backgammon strategy. Machine Learning **32** (1998) 225–240
[11] Juillé, H., Pollack, J.B.: Dynamics of co-evolutionary learning. In: Proceedings of the Fourth International Conference on Simulation of Adaptive Behavior, MIT Press (1996) 526–534
[12] Angeline, P.J., Saunders, G.M., Pollack, J.B.: An evolutionary algorithm that constructs recurrent networks. IEEE Transactions on Neural Networks **5** (1994) 54–65
[13] Sims, K.: Evolving 3d morphology and behavior by competition. In Brooks, R., Maes, P., eds.: Proceedings 4th Artificial Life Conference. MIT Press (1994)
[14] Komosinski, M., Ulatowski, S.: Framsticks: Towards a simulation of a nature-like world, creatures and evolution. In Dario Floreano, J.D.N., Mondada, F., eds.: Proceedings of 5th European Conference on Artificial Life (ECAL99). Volume 1674 of Lecture Notes in Artificial Intelligence., Springer-Verlag (1999) 261–265
[15] Floreano, D., Mondada, F.: Evolution of homing navigation in a real mobile robot. IEEE Transactions on Systems, Man, and Cybernetics (1996)

[16] Cliff, D., Harvey, I., Husbands, P.: Evolution of visual control systems for robot. In Srinivisan, M., Venkatesh, S., eds.: From Living Eyes to Seeing Machines. Oxford (1996)
[17] Lund, H.: Evolving robot control systems. In Alexander, ed.: Proceedings of 1NWGA, University of Vaasa (1995)
[18] Gallagher, J.C., Beer, R.D., Espenschield, K.S., Quinn, R.D.: Application of evolved locomotion controllers to a hexapod robot. Robotics and Autonomous Systems **19** (1996) 95–103
[19] Kawauchi, Y., Inaba, M., Fukuda, T.: Genetic evolution and self-organization of cellular robotic system. JSME Int. J. Series C. (Dynamics, Control, Robotics, Design & Manufacturing) **38** (1995) 501–509
[20] Jakobi, N.: Evolutionary robotics and the radical envelope of noise hypothesis. Adaptive Behavior **6** (1997) 131–174
[21] Mataric, M.J., Cliff, D.: Challenges in evolving controllers for physical robots. Robotics and Autonomous Systems **19** (1996) 67–83
[22] Floreano, D.: Evolutionary robotics in artificial life and behavior engineering. In Gomi, T., ed.: Evolutionary Robotics. AAI Books (1998)
[23] Funes, P., Pollack, J.B.: Computer evolution of buildable objects. In Husbands, P., Harvey, I., eds.: Fourth European Conference on Artificial Life, Cambridge, MIT Press (1997) 358–367
[24] Funes, P., Pollack, J.B.: Evolutionary body building: Adaptive physical designs for robots. Artificial Life **4** (1998) 337–357
[25] Funes, P., Pollack, J.B.: Computer evolution of buildable objects. In Bentley, P., ed.: Evolutionary Design by Computers. Morgan-Kaufmann, San Francisco (1999) 387 – 403
[26] Lipson, H., Pollack, J.B.: Evolution of machines. In: Proceedings of 6th International Conference on Artificial Intelligence in Design, Worcester MA, USA. (2000) (to appear).
[27] Lipson, H., Pollack, J.B.: Towards continuously reconfigurable robotics. In: Proceedings of IEEE International Conference on Robotics and Automation, San Francisco CA, USA. (2000) (to appear).
[28] Watson, R., Ficici, S., Pollack, J.: Embodied evolution: Embodying an evolutionary algorithm in a population of robots. In Angeline, P., Michalewicz, Z., Schoenauer, M., Yao, X., Zalzala, A., eds.: 1999 Congress on Evolutionary Computation. (1999)
[29] Ficici, S.G., Watson, R.A., Pollack, J.B.: Embodied evolution: A response to challenges in evolutionary robotics. In Wyatt, J.L., Demiris, J., eds.: Eighth European Workshop on Learning Robots. (1999) 14–22
[30] Beckers, R., Holland, O., Deneubourg, J.: From local actions to global tasks: Stigmergy and collective robotics. In Brooks, R., Maes, P., eds.: Artificial Life IV, MIT Press (1994) 181–189
[31] Balch, T., Arkin, R.: Motor schema-based formation control for multiagent robot teams. In: Proceedings of the First International Conference on Multi-Agent Systems ICMAS-95, AAAI Press (1995) 10–16
[32] Rus, D., Donald, B., Jennings, J.: Moving furniture with teams of autonomous robots. In: Proceedings of IEEE/RSJ IROS'95. (1995) 235–242
[33] Donald, B., Jennings, J., Rus, D.: Minimalism + distribution = supermodularity. Journal on Experimental and Theoretical Artificial Intelligence **9** (1997) 293–321
[34] Tan, M.: Multi-agent reinforcement learning: independent vs. cooperative agents. In: Proceedings of the Tenth International Machine Learning Conference. (1993) 330–337

[35] Littman, M.: Markov games as a framework for multi-agent reinforcement learning. In: Proceedings of the International Machine Learning Conference. (1994) 157–163
[36] Saunders, G., Pollack, J.: The evolution of communication schemes of continuous channels. In Maes, P., Mataric, M., Meyer, J.A., Pollack, J., Wilson, S., eds.: From Animals to Animats IV, MIT Press (1996) 580–589
[37] Balch, T.: Learning roles: Behavioral diversity in robot teams. In: 1997 AAAI Workshop on Multiagent Learning, AAAI (1997)
[38] Mahadevan, S., Connell, H.: Automatic programming of behavior-based robots using reinforcement learning. In: Proceedings of the Ninth National Conference on Artificial Intelligence (AAAI '91). (1991) 8–14
[39] Mataric, M.: Learning to behave socially. In Cliff, D., Husbands, P., Meyer, J.A., Wilson, S., eds.: From Animals to Animats 3, MIT Press (1994) 453–462
[40] Mataric, M.: Reward functions for accelerated learning. In Cohen, W., Hirsh, H., eds.: Proceedings of the Eleventh International Conference on Machine Learning, Morgan Kaufman (1994) 181–189
[41] Parker, L.: Task-oriented multi-robot learning in behavior-based systems. Advanced Robotics, Special Issue on Selected Papers from IROS '96 **11** (1997) 305–322
[42] Uchibe, E., Asada, M., Hosoda, K.: Cooperative behavior acquisition in multi mobile robots environment by reinforcement learning based on state vector estimation. In: Proceedings of International Conference on Robotics and Automation. (1998) 1558–1563