Rapid Evolution of Robot Gaits

Joshua E. Auerbach, Grégoire Heitz, Przemyslaw M. Kornatowski, and Dario Floreano
Laboratory of Intelligent Systems
Ecole Polytéchnique Fédérale de Lausanne
Lausanne, Switzerland
{first name}.{last name}@epfl.ch

Categories and Subject Descriptors

I.2.9 [**Computing Methodologies**]: Artificial Intelligence—*Robotics*

Keywords

Evolution; Robotics; Gait Learning; Education

1. EXTENDED ABSTRACT

The promise of Evolutionary Robotics (ER) [7] to completely automatize the design of robot controllers and/or morphologies is an idea with great appeal not only to researchers, but also to students. Recently [1], we introduced the RoboGen$^{\mathrm{TM}}$ open-source software and hardware platform for Evolutionary Robotics[1], and described its success as an educational tool in a masters level course at EPFL. There it was shown that RoboGen could provide students with valuable hands on experience with Evolutionary Robotics, neural networks, physical simulation, 3D printing, mechanical assembly, and embedded processing.

However, when attempting to grab and hold student interest, the large requirements of time and computational resources required to achieve good results in ER systems may be discouraging. In fact, after two years of using RoboGen for class projects, the biggest student complaints all concerned the slow speed of evolutionary progress. In order to overcome these limitations, we present here a simple and effective technique for rapidly evolving robot gaits in a manner of seconds or minutes rather than hours or days. Moreover, through a previously executed, iterative procedure of tuning the simulator and the embedded processing of RoboGen robots, many of these evolved gaits can transfer to real robots with reasonable fidelity without requiring hardware-in-the-loop fitness evaluations [4, 12] or online simulator updates [2].

While there are several existing techniques for evolving natural looking and successful robot gaits [3, 6, 10], most existing approaches require upwards of hundreds of thousands of fitness evaluations to achieve good results. Even assuming a modest 1s of compute time per fitness evaluation and a reasonable degree of parallelism achievable on a modern personal computer (e.g. 8 concurrent evaluations) most methods may still require many hours to see successful gaits emerge.

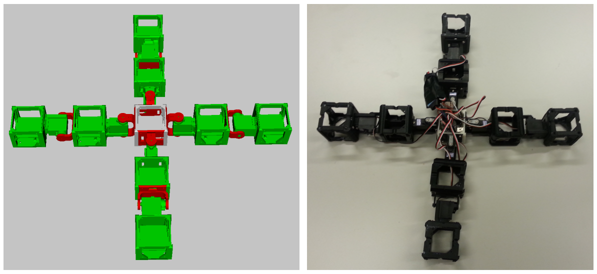

Figure 1: One of the RoboGen robots experimented with in this work. ***Left:*** **In simulation.** ***Right:*** **In reality.**

Here, we forgo the use of the aforementioned algorithms and the use of complex neural network architectures, and instead rely on two basic techniques: Compositional Pattern Producing Network (CPPN) encodings [9] and simple parameterized oscillator neurons. More specifically, each motor of a given robot is driven by an oscillator neuron parametrized by three values: period, amplitude, and phase offset from a central clock. The genome of each controller is a CPPN, which is queried with the physical position of each motor relative to the "core component" of the robot[2] and produces the three parameters of each oscillator (each on a dedicated CPPN output unit). The current version of RoboGen enforces robots to be in an initially planar configuration, and therefore only a two dimensional vector is required to represent the position of each motor. Therefore the CPPNs have three input units (two for the motor position and one for the bias) and three output units.

By endowing robots with oscillator neurons they are able to produce rhythmic gait patterns for free. By encoding the parameters of the oscillator neurons with CPPNs, evolution is able to capitalize on the geometric regularities present in most robot morphologies in order to produce successful gaits. Here we evolve CPPNs for a simple maximum

[1] http://www.robogen.org

[2] The "core component" in RoboGen robots is the component that houses the onboard microcontroller and battery. It is the one required element of a RoboGen robot.

GECCO'15 Companion, July 11–15, 2015, Madrid, Spain.

ACM ISBN 978-1-4503-3488-4/15/07.
DOI: http://dx.doi.org/10.1145/2739482.2764894

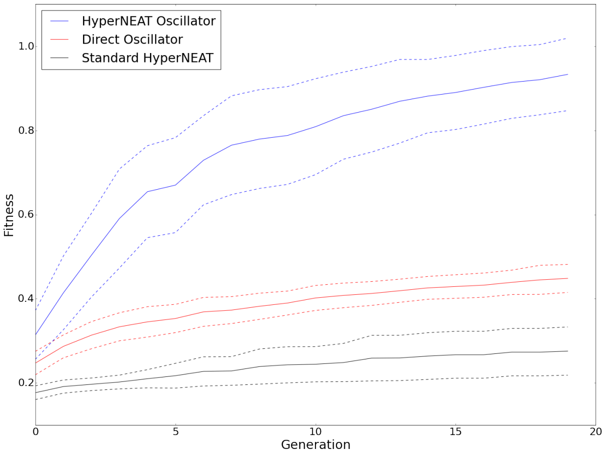

Figure 2: Comparison between evolving with standard HyperNEAT, HyperNEAT for oscillator neurons, and directly encoded oscillator neurons.

displacement task using the RoboGen simulator and HyperNEAT [8][3]. The controllers evolved using a population size of 100 for 20 generations yielding 2000 fitness evaluations, which can run in a few minutes time. As compared to using either a simple direct encoding of oscillator parameters or evolving standard neural networks with HyperNEAT, the ultra rapid approach used here is able to find significantly more fit solutions (see Fig. 2) for a variety of robots (such as the one depicted in Fig. 1). Additionally several of the gaits evolved using this technique were transfered onto a real robot (see 1 bottom) and showed a strong correspondence between simulated and evolved behaviors. Videos of this robot behavior both in simulation and reality can be seen at http://youtu.be/cIUlCMEtly0 and http://youtu.be/gaibm3RazXA, respectively.

While the robot behaviors evolved here clearly lack the complexity of some state-of-the-art ER results, and will not be directly usable in more complex tasks requiring reactive control or memory, the ability to rapidly evolve successful robot gaits will be instrumental in attracting students, hobbyists and others to Evolutionary Robotics. This ability will also be crucial if one wants to bring Evolutionary Robotics to the web [5] to attract a wider and more diverse audience to the field, and to incorporate real-time user feedback in the optimization process [11]. Work towards building a web based platform for RoboGen is currently underway.

Additionally, it is possible to extend the neural system to allow for more complex control by allowing oscillator neurons to be modulated by sensors or other neurons. In doing so, one can first rapidly evolve an open loop controller as is done here, so that users see real-time progress, and then gradually increase the controller sophistication for more complex tasks. Future work will extend the current system in this direction.

2. ACKNOWLEDGEMENTS

This project has received funding from the European Union's Seventh Framework Programme for research, technological development and demonstration under grant agreement no 308943.

3. REFERENCES

[1] J. Auerbach, D. Aydin, A. Maesani, P. Kornatowski, T. Cieslewski, G. Heitz, P. Fernando, I. Loshchilov, L. Daler, and D. Floreano. RoboGen: Robot Generation through Artificial Evolution. In *Artificial Life 14: Proceedings of the Fourteenth International Conference on the Synthesis and Simulation of Living Systems*, pages 136–137. The MIT Press, 2014.

[2] J. Bongard, V. Zykov, and H. Lipson. Resilient machines through continuous self-modeling. *Science*, 314:1118–1121, 2006.

[3] J. Clune, B. Beckmann, C. Ofria, and R. Pennock. Evolving coordinated quadruped gaits with the HyperNEAT generative encoding. In *Proceedings of the IEEE Congress on Evolutionary Computing*, pages 2764–2771, 2009.

[4] S. Koos, J.-B. Mouret, and S. Doncieux. The transferability approach: Crossing the reality gap in evolutionary robotics. *Evolutionary Computation, IEEE Transactions on*, 17(1):122–145, 2013.

[5] J. M. Moore, A. J. Clark, and P. K. McKinley. Evolutionary robotics on the web with webgl and javascript. *CoRR*, abs/1406.3337, 2014.

[6] G. Morse, S. Risi, C. R. Snyder, and K. O. Stanley. Single-unit pattern generators for quadruped locomotion. In *Proceedings of the 15th annual conference on Genetic and evolutionary computation*, pages 719–726. ACM, 2013.

[7] S. Nolfi and D. Floreano. *Evolutionary Robotics: The Biology,Intelligence,and Technology*. MIT Press, Cambridge, MA, USA, 2000.

[8] K. Stanley, D. D'Ambrosio, and J. Gauci. A hypercube-based encoding for evolving large-scale neural networks. *Artificial Life*, 15(2):185–212, 2009.

[9] K. O. Stanley. Compositional pattern producing networks: A novel abstraction of development. *Genetic Programming and Evolvable Machines*, 8(2):131–162, 2007.

[10] R. A. Téllez, C. Angulo, and D. E. Pardo. Evolving the walking behaviour of a 12 dof quadruped using a distributed neural architecture. In *Biologically Inspired Approaches to Advanced Information Technology*, pages 5–19. Springer, 2006.

[11] M. Wagy and J. Bognard. Collective design of robot locomotion. In *ALIFE 14: The Fourteenth Conference on the Synthesis and Simulation of Living Systems*, volume 14, pages 138–145.

[12] J. C. Zagal, J. Ruiz-del Solar, and P. Vallejos. Back to reality: Crossing the reality gap in evolutionary robotics. In *IAV 2004 the 5th IFAC Symposium on Intelligent Autonomous Vehicles, Lisbon, Portugal*, 2004.

[3] Using a modified version of Peter Chervenski's open-source MultiNEAT implementation: http://multineat.com/.