Interactive Simulated Robot Construction and Controller Evolution

[Extended Abstract]

Paul Beliveau
University of Vermont
Burlington, VT 05405
paul.beliveau@uvm.edu

Greg Hornby
University of California
San Diego, CA 92093
gregory.s.hornby@nasa.gov

Josh Bongard
University of Vermont
Burlington, VT 05405
josh.bongard@uvm.edu

ABSTRACT

A robot's morphology affects not only its capabilities, but also its evolvability. Here we introduce an evolutionary robotics platform that will enable us to investigate the ability of non-roboticists to collectively explore the design space of evolvable robot body plans. Users create robots in a simulator using an interactive interface, then let evolution find a controller that enables efficient locomotion. The user can change the morphology of the robot and test the new design iteratively, altering the design based on the assessment of the robot's performance after a period of evolution of the controller. We investigate whether there is a correlation between the methods users choose to build robots and the evolvability of the robots.

Categories and Subject Descriptors

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

General Terms

Design, Experimentation

Keywords

Evolutionary robotics, User-guided, Crowd-sourcing

1. INTRODUCTION

It is not clear what aspects of a robot's morphology make it evolvable. Biological organisms exhibit a wide variety of successfully evolved morphologies that we observe to build general intuitions about the relationship between morphology and behavior. Here we introduce an evolutionary robotics platform that will enable us to assess the ability of members of the general public to collectively explore the space of evolvable robot body plans.

User-controlled simulations such as breve[3] and framsticks[4] are available to the general public. The Foldit[1] online game has been successful in allowing non-expert users to contribute to the prediction of possible protein configurations. Our goal is to use related methods to investigate evolvability.

GECCO'12 Companion, July 7–11, 2012, Philadelphia, PA, USA.
ACM 978-1-4503-1178-6/12/07.

2. METHODS

The design environment uses the Open Dynamics Engine (ODE) physics engine with a graphics window and keyboard-driven input to allow users to build a robot from an initial set of disconnected blocks, cylinders, and hinge joints. The environment includes a light source as a target, and the blocks and cylinders incorporate light sensors that the evolutionary algorithm can use to measure fitness.

We ran three trials of the experiment, using undergraduate students enrolled in computer science classes at the University of Vermont.

In the first two trials, the subjects were students who were enrolled in an evolutionary robotics class. The students had used ODE for their class work and were familiar with robotics in simulation. These early trials exposed shortcomings in the user interface; no results from them are presented here.

In the third trial, the subjects were students who were enrolled in a Human-Computer Interface class. With two exceptions, these students had no formal instruction in robotics. The exceptions were students who had previously taken the evolutionary robotics class.

The trial consisted of two 50-minute sessions: first a self-guided tutorial on how to use the simulator interface, then a session in which the subjects were asked to create a robot that had the best potential for evolving efficient locomotion. Two restrictions were imposed: the robot had to be "legged", and the size of the robot was limited to disallow a huge robot falling over in order to get close to the target.

Users were free to watch their robot moving and to change its morphology as often as they liked during the allotted time. The simulator captured the time-stamped command keystrokes from each user's session as well as the simulation environment that contained each user's robot.

3. RESULTS

About half of the students (nine total) produced a robot that could be simulated, though not all could be analyzed as intended (see below).

We re-created the student robots, standardized the starting distance for each robot from the target in the environment, and let the robots evolve for 24 hours (wall clock time). For this preliminary trial, we performed thirty independent runs of each student's robot, using the ALPS algorithm[2]. We took the best robot from each run and calculated the mean from the thirty runs to produce an average fitness value for each robot.

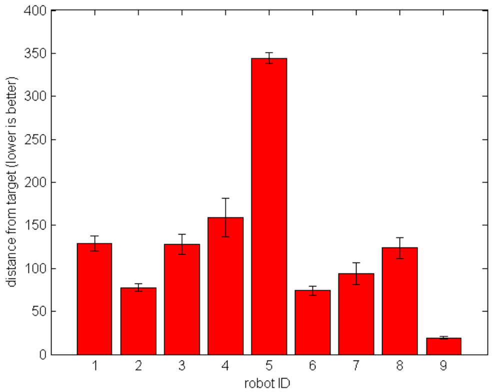

Figure 1: Robot fitness for nine sample robots. Error bars show the standard deviation in fitness for the thirty runs.

The bar graph in Figure 1 shows the average fitness for the robots in this preliminary trial, where fitness is defined as the distance from robot to target (smaller is better) at the end of the simulation time. Of the nine robots shown, the rightmost three were excluded from further analysis.

The best-performing robot, robot 9, moves by rolling, which disqualifies it. (The student found a way to make it roll using a hinge joint.) The other two unusable robots were created with an earlier version of the application, which did not include keystroke capture. For this reason, we cannot analyze the user inputs for these robots.

We are thus left with six robots and the command sequences that document their construction. We cannot draw conclusions from such a small number of datapoints, but scatterplots showing fitness and statistics from command keystrokes are shown in Figure 2 to illustrate our intentions for future research. Statistics include the count of subjects' use of commands, and time metrics.

4. DISCUSSION

The first two trials using the simulation environment showed that the simulator was not robust enough for computer-literate college students, much less for the general public. Subsequent simulator interface improvements helped, so that the third trial produced some data, but too small a percentage of users produced a working robot.

The plots of fitness versus user input statistics shown in Figure 2 represent a first step in analysis that might reveal a correlation between fitness and user behavior. For example, the mean time between commands might be interpreted as an indicator of the level of interest of the subject.

Planned future work includes improvements to the system interface in order to lighten the burden on the user and with the ultimate goal of making the system game-like enough to attract many more users, and to test whether subjects' intuitions about evolvability can be validated.

We hypothesize that users do not have to be robotics experts to collectively explore the space of robots amenable to optimization by an evolutionary algorithm; they may be able to use their intuition and practical knowledge of mechanical systems as well as visual feedback from the simulator to guide robot development.

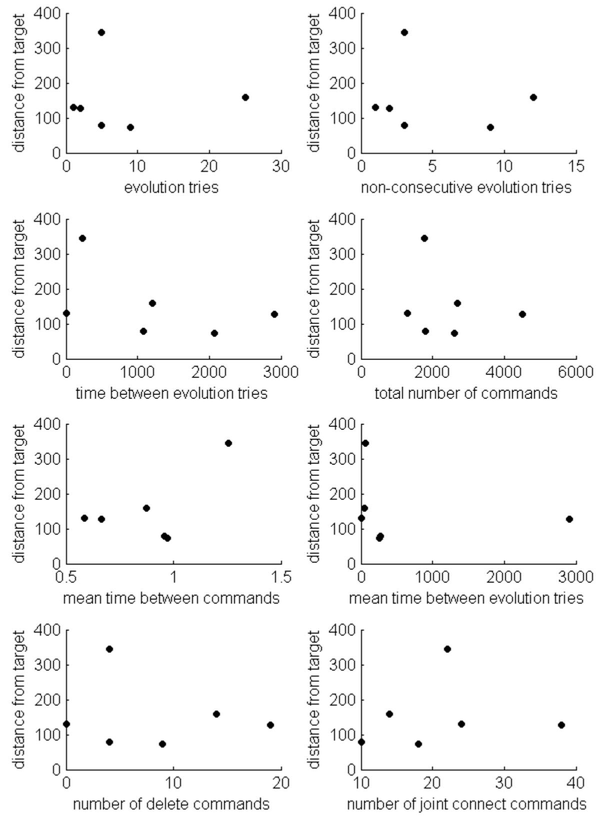

Figure 2: Selected user inputs vs. robot fitness (lower distance from target is better).

5. ACKNOWLEDGMENTS

This work was supported by National Science Foundation Grant PECASE-0953837 and DARPA M3 grant W911NF-1-11-0076. The authors would also like to thank the Vermont Advanced Computing Center.

6. REFERENCES

[1] S. Cooper. Predicting protein structures with a multiplayer online game. *Nature*, 466:756–760, 2010.

[2] G. S. Hornby. Alps: the age-layered population structure for reducing the problem of premature convergence. *Proceedings of the 8th annual conference on genetic and evolutionary computation*, pages 815–822, 2006.

[3] J. Klein. Breve: a 3d environment for the simulation of decentralized systems and artificial life. *www.spiderland.org/breve.pdf*, 2008.

[4] M. Komosinski. Framsticks. *www.framsticks.com/biblio*, 1998 - 2011.