Not All Physics Simulators Can Be Wrong in the Same Way

[Extended Abstract]

Shane Celis
Morphology, Evolution & Cognition Laboratory
Vermont Complex Systems Center
College of Engineering and Mathematical Sciences
University of Vermont
scelis@uvm.edu

Josh Bongard
Morphology, Evolution & Cognition Laboratory
Vermont Complex Systems Center
College of Engineering and Mathematical Sciences
University of Vermont
jbongard@uvm.edu

ABSTRACT

Transferring designs in evolutionary robotics from simulation to reality remains problematic. It has been addressed by using quasi-static physics simulators, adding noise to encourage robustness, and evolving primarily in simulation then evolving on actual hardware for fine-tuning. This paper experiments with this idea: All physics simulators have errors, but if the errors are distinct, one might profitably use multiple simulators to detect unrealistic physical behavior in simulation. Two physics simulators are used to evolve a controller for quadruped locomotion. Preliminary results validate some assumptions and further work is suggested.

Categories and Subject Descriptors

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

General Terms

Experimentation

1. INTRODUCTION

Transferring designs evolved *in silico* to real robots remains a difficult problem within Evolutionary Robotics (ER), known as the reality gap problem. Many methods have been introduced to address this problem. Noteably the GOLEM project by Pollack and Lipson used a quasi-static physics simulator[7]. This method limits the kinds of robots that can be evolved, e.g., robots that exploit dynamics like passive walkers can not use this method. Pollack et al. [6] evolved controllers in simulation then tranferred controllers to a real robot which was then evolved further. Hornby et al. [3] evolved robots directly on hardware without simulation. Forgoing simulation has drawbacks: it requires robust hardware, needs a controlled environment, has slower evaluation time, and is more difficult to fully automate. Jakobi [4] proposed minimal simulations with large variations in nonessential aspects to produce robust controllers. Koos et al. [5] compare measurements between simulation and a real robot to compute a simulation-to-reality (STR) disparity measurement.

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

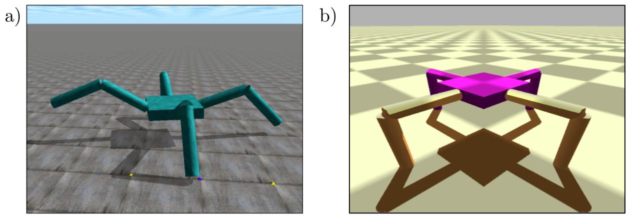

Figure 1: Quadruped in a) Open Dynamics Engine and b) Bullet Physics Library

ER is particularly susceptible to errors in physics simulators because evolution will exploit unrealistic physics that provide any advantage. These "cheating" robots may score a high fitness and displace other robots in the population that had a more realistic behavior, i.e., a behavior that was more transferable.

The idea this paper explores is informed by the following intuition. Physics simulators attempt to produce realistic physical behavior. A physics simulator will not succeed in all cases. Physics simulators have their own tolerances and sensitivities owing to the choice of integrator, collision detector, collision response, and so on. So one might reasonably expect each physics simulator to fail in different ways. These distinct failures might serve as a good proxy to measuring transferability.

2. METHOD

The robot used in this investigation is a quadruped walker. It consists of cuboid body with eight cylinders attached, which serve as its legs. The robot has eight degrees of freedom denoted $\{\theta_1, \theta_2, \ldots, \theta_8\}$. In Figure 1b the state of the robot's joints is $\theta_i = 0$ for all i. The joint range is constrained to $[-\pi/4, \pi/4]$. The hinge joints are position controlled. Four touch sensors $\{s_1, s_2, s_3, s_4\}$ are associated with the distal limbs. When limb j is in contact with the ground, $s_j = 1$ otherwise $s_j = 0$.

The controller is a feed-forward artificial Neural Network (NN) with sigmoidal activation function. It accepts four inputs from the touch sensors s_j. The NN has eight motor outputs that determine target angle of each joint. The bias coefficients of the NN are zero. The controller updates the desired angle θ_i at 10 Hz. The genome is represented by a

32 element vector that encodes the weights w_{ij} of the NN.

$$\theta_i = \frac{\pi}{4} \tanh(\sum_{j=1}^{4} w_{ij}\, s_j)$$

The robot is evaluated in two physics simulators: 1) the Open Dynamics Engine (ODE) and 2) the Bullet Physics Library. The physics simulators use a fixed time step of 0.01 seconds. The robots are evaluated for 10 simulated seconds.

The distance traveled in the x-y ground plane by ODE and Bullet is denoted by d_1 and d_2 respectively, measured in meters. Three experiments are conducted based on these measurements. Experiment A maximizes d_1 and minimizes $|\Delta d| = |d_1 - d_2|$. Experiment B maximizes d_1 and maximizes $|\Delta d|$. Experiment C, the control experiment, maximizes d_1 only. Twenty independent trials of each experiment are performed using the multi-objective optimization algorithm NSGA-II[2] with a population of 20 and 250 generations.

3. RESULTS AND DISCUSSION

Figure 2 shows the non-dominated solutions for twenty independent trials of experiment A, B, and C. In experiment A (fig. 2a), when d_1 is maximized and $|\Delta d|$ is minimized, the individuals achieve successful displacement in both physics engines.

In experiment B (fig. 2b), when d_1 and $|\Delta d|$ are maximized, the same controller can achieve a high displacement in one physics simulator and little displacement in another. These two classes of controllers may allow for some interesting analysis, discussed further in the next section.

In experiment C (fig. 2c), only d_1 is maximized. The distance d_2 is shown for the sake of comparison with experiments A and B. Experiment C shows that when optimizing on the basis of only one physics simulator, the controller performs worse in the other physics simulator, as one would expect. However, the spread of controllers is not strongly biased towards $d_2 = 0$.

4. FUTURE WORK

The two classes of controllers, divided by the dashed line, shown in fig. 2b could be analyzed to determine what makes the controller good or bad in one simulator versus another. This could reveal two things: 1) sensitivities of the simulator and 2) what means a controller is using to achieve its task.

To explore this idea more fully, ideally, one would like to use any n physics simulators that are available. However, writing and maintaining n robot evaluators—one for each physics simulator—is a burden. The Physics Abstraction Layer (PAL) [1] currently supports twelve different simulators. Using PAL might make this approach more practical.

Having multiple physics simulators available, one might profitably apply multiple model approaches such as those used in climate projection[8] to produce more robust controllers.

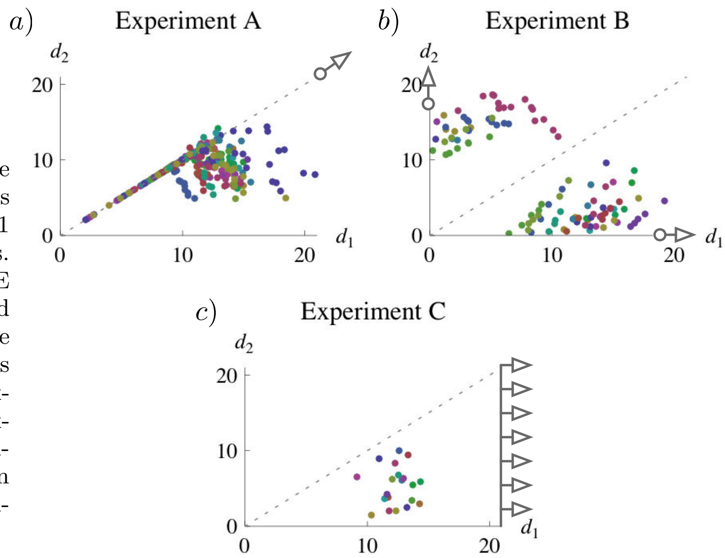

Figure 2: The points represent the distances achieved by non-dominated controllers in both physics simulators. The color of each point represents which of the twenty trials produced it: same color, same trial. The dashed line represents the case where the distances are the same, i.e. $\Delta d = 0$. The arrows represent the direction of optimization for each experiment. Experiment B shows that a controller can be found that performs well in one simulator while doing poorly in the other. Experiment C shows that $d_2 < d_1$ in all cases.

5. REFERENCES

[1] A. Boeing and T. Bräunl. Evaluation of real-time physics simulation systems. In *GRAPHITE '07: Proceedings of the 5th international conference on Computer graphics ...* ACM, Dec. 2007.

[2] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, Apr. 2002.

[3] G. Hornby, S. Takamura, J. Yokono, O. Hanagata, T. Yamamoto, and M. Fujita. Evolving robust gaits with AIBO. In *Robotics and Automation, 2000. Proceedings. ICRA '00. IEEE International Conference on*, pages 3040–3045, 2000.

[4] N. Jakobi. Minimal simulations for evolutionary robotics. *Doctoral dissertation, University of Sussex*, 1998.

[5] S. Koos, J. Mouret, and S. Doncieux. Crossing the reality gap in evolutionary robotics by promoting transferable controllers. *Proceedings of the 12th annual conference on Genetic and evolutionary computation*, pages 119–126, 2010.

[6] J. Pollack, H. Lipson, S. Ficici, P. Funes, G. Hornby, and R. Watson. Evolutionary techniques in physical robotics. *Evolvable Systems: From Biology to Hardware*, pages 175–186, 2000.

[7] J. B. Pollack and H. Lipson. The GOLEM project: evolving hardware bodies and brains. In *Evolvable Hardware, 2000. Proceedings. The Second NASA/DoD Workshop on*, pages 37–42, 2000.

[8] C. Tebaldi and R. Knutti. The use of the multi-model ensemble in probabilistic climate projections. *Philosophical Transactions of the Royal Society A*, 365(1857):2053–2075, Aug. 2007.