A Baseline-Realistic Objective Open-Ended Kinematics Simulator for Evolutionary Robotics

Riley Konsella
Union College
807 Union St
Schenectady, New York 12308
konsellr@union.edu

Frank Chiarulli
Union College
807 Union St
Schenectady, New York 12308
chiarulf@union.edu

John Peterson
Union College
807 Union St
Schenectady, New York 12308
petersoj@union.edu

John Rieffel
Union College
807 Union St
Schenectady, New York 12308
rieffelj@union.edu

ABSTRACT

Most modern applications of Evolutionary Robotics (ER) rely upon computer-based physics simulations in order to model the behavior of the systems in question. One of the greatest challenges in the field of ER, therefore, is the development of robust, high-precision and accurate physics simulators that can model all necessary and relevant real-world interactions in an computationally efficient manner. Up until now, most popular ER simulators are nonetheless deficient in one or many of these properties. Here we introduce a new competitive simulator, the Baseline-Realistic Objective Open-Ended Kinematics Simulator (BROOKS) that outperforms other off-the-shelf simulators in most criteria. Our simulator is free, open-sourced, and easy to modify. It can model a wide range of robotic platforms, substrates and environments. Moreover, we claim solutions produced within the BROOKS simulator perform almost identically in the real-world, thereby helping to address one of the most challenging aspects of simulation in Evolutionary Robotics: the *Reality Gap.* Ultimately, we believe that BROOKS will establish a new baseline against which all other simulators should be compared.

KEYWORDS

Evolutionary Robotics, Simulation, Reality

ACM Reference format:
Riley Konsella, Frank Chiarulli, John Peterson, and John Rieffel. 2017. A Baseline-Realistic Objective Open-Ended Kinematics Simulator for Evolutionary Robotics . In *Proceedings of GECCO '17 Companion, Berlin, Germany, July 15-19, 2017,* 4 pages.
https://doi.org/http://dx.doi.org/10.1145/3067695.3082164

GECCO '17 Companion, July 15-19, 2017, Berlin, Germany

ACM ISBN 978-1-4503-4939-0/17/07…$15.00
https://doi.org/http://dx.doi.org/10.1145/3067695.3082164

1 INTRODUCTION

The field of Evolutionary Robotics has always been deeply tied to the use of realistic physics simulators. One of the earliest examples is certainly the work of Karl Sims, whose evolved embodied agents were evolved (and even co-evolved) within a compellingly realistic simulation that modeled both ground-based and fluidic interactions [16]. The impact of this paper effectively launched the field of Virtual Creatures. An exhaustive list of simulation-based Evolutionary Robotics papers would be incredibly long, however some notable landmarks in simulation-based Evolutionary Robotics include Hornby's generatively-encoded robots [11], Bongard's early GRN-based robots [2], and Hiller's VoxCAD-based soft robots, [10].

Throughout the history of Evolutionary Robotics, many researchers have developed their own physics engines (such as Sims, Hornby, and Hiller) in order to perform their research. The advantage of home-grown simulators lies in the ability to fine-tune characteristics to the problem at hand, and choosing to model only those factors deemed relevant. The downside, however, is that, even when open-sourced, these engines are rarely scrutinized by large developer communities, and are rarely easily extended by outside developers. Alternatively, more recent ER research has been based upon upon off-the shelf physics simulators, such as the Open Dynamics Engine (ODE) (used in work on evolved tensegrities by Rieffel [14, 15] among others), PhysX (Glette *et al.* [8]), and Bullet Physics (used as the basis for Sunspiral *et al*'s NTRT tensegrity robot research [4] , among others). The advantage of these simulators is that they are generally developed by large communities, and can have relatively fast development cycles and longer lists of features. However, this comes at the cost of incorporating features in the physics engine which a particular ER researcher may not require.

When used to answer abstract research questions about morphology or control, simulated solutions are often sufficient. However, if researchers are interested in transferring evolved results into the physical world they face an additional challenge. Typically, solutions evolved in simulation often struggle when transferred to the real world – Jakobi termed this disparity between simulated and evolved solutions the "Reality Gap" [12]. Several approaches exist to trying to "cross" the reality gap - beginning with Jakobi's own modeling of sensor noise [12], and including more sophisticated techniques such as Bongard's co-evolutionary self-modeling [1], Cully *et al.*'s Bayesian-Optimization approach [6], and Mouret's transferability approach [13].

Regardless of the choice of simulator, most approaches to modeling real-world physics have weaknesses and drawbacks. One of the most significant in the context of evolutionary robotics is the handling of collisions between complex geometries. Further, if a simulator's time-step is too large, then colliding bodies may lead penetrate, requiring a large restitution force to correct – leading to the common evolutionary exploitation of robots optimizing "movement" by shooting rapidly into space. Using a smaller time-step can reduce (but not eliminate) this type of problem, however, this in turn leads to increased computational effort and longer overall simulation times. Other challenges include numerical instabilities and unexpected oscillations.

Of course, others have decided to eschew simulation entirely and, inspired by the embodiment philosophy of Rodney Brooks (who quipped that "the world is its own best model" [3]), have evolved solutions directly in the real world, without resorting to simulation. This trend began with the "Sussex Approach" of Harvey *et al* [5, 9, 17], and encompasses the Embodied Evolution of Watson and Ficici [18], Floreano and Mondada [7] and Zykov *et al* [19].

This diversity of simulator choice in Evolutionary Robotics presents an obstacle to further progress in the field. There are compelling arguments towards finding a common simulation environment, or at the least finding a solid and reliable baseline evolutionary robotics simulator against which others can compared and contrasted. In this work we present one such candidate system.

2 THE BROOKS PHYSICS ENGINE

Here we introduce a new cutting-edge physics simulator for evolutionary robotics that, for the first time, has sufficient fidelity to the real world to resolve the "reality gap", thereby setting a new gold standard for evolutionary robotics simulators, and a firm baseline against which others can be measured. We have named this the Baseline Realistic Objective Open-Ended Kinematics Simulator, or BROOKS. By "baseline realistic" we mean it provides a complete set of necessary (although not necessarily minimal) features required of evolutionary robotics simulators.

3 FEATURES OF THE BROOKS SIMULATOR

The BROOKS environment has been under development for quite some time, and can be considered both widely accessible and open-sourced (although some might argue that the source code is somewhat inscrutable). At our web page we offer a simple cross-platform download that can be easily installed by novices within a matter of minutes. We will describe several of the features of BROOKS that distinguish it from competing robotics simulators in the section below. This is far from an exhaustive list, but should nonetheless be compelling.

A Diversity of Terrain models

BROOKS offers a wide range of ground plane and surface textures (with a variety of frictional coefficients), and can even model thermal environments such as snow and ice, as demonstrated by Figure 1. Most of these terrains can be easily installed by the end-user,

Figure 1: BROOKS can also simulate a variety of complex surface textures and frictional coefficients that are often challenging for competing simulators to model, including material such as snow and wood mulch.

although some models, such as persian carpet, may require additional costs. Similarly, certain regional and localization restrictions makes it difficult for all users to model environmental thermal features such as snow. Typical simulations involve temperatures between 10°C and 30°C, however a limited number of users have reported environments much more extreme than these.

ROS Compatable

BROOKS has been compatible with the popular ROS Robotic operating system since the launch of ROS. BROOKS plugins exist for all ROS-compatible robots, such as the TurtleBot - however many of these modules must be purchased separately from the robot manufacturer.

Sensor Models

BROOKS contains a wide range of sensors models, from LiDAR to ultrasound to simple acoustic microphones. Moreover, the noise models for these sensors are guaranteed to be accurate simulacra of their real-world counterparts. BROOKS even models more overlooked sensor failure modes, such as those caused by loose or damaged cabling, or unanticipated reflections. Again, specific sensor module plug-ins can be purchased at additional charge from the sensor retailer.

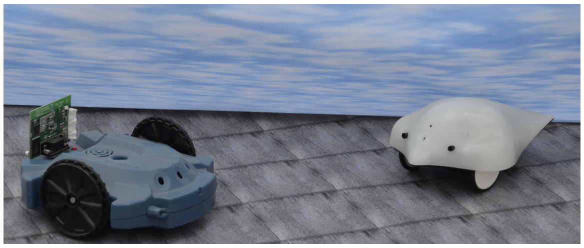

Figure 2: Our BROOKS simulator can simulate multiple complex robotic morphologies with a high degree of accuracy. Shown here are two wheeled educational robots on the default texture surface.

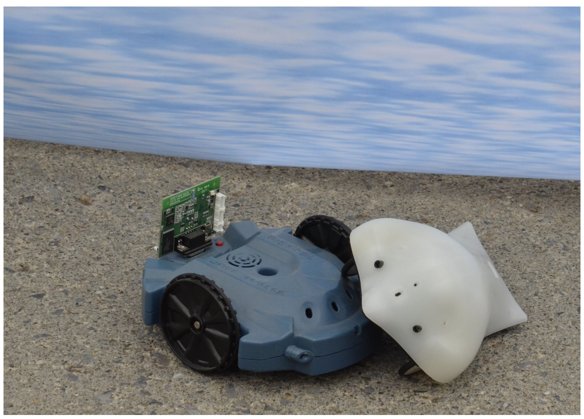

Figure 3: Dynamic collisions between multiple rigid bodies are handled smoothly, without any errors caused by restitution forces or ground-plane penetration.

Mutiple Materials

The BROOKS simulator can handle a variety of materials for modeled objects – not just conventional rigid body dynamics and kinematics, but also soft-body, fabric, fluid, and even plasma-based materials. This is particularly valuable for the emerging field of soft robotics, which often struggles to find suitable simulators. An example of a soft-bodied robot is shown in Figure 5

Complex Morphologies and Robust Collision Models

BROOKS can handle a variety of morpologies - from simple Platonic solids to more convoluted morphologies. Moreover, as shown by Figure 3, the simulator can handle collisions between objects quite well, with little risk of unrealistic penetrations.

Inherently Parallelized

Not only is BROOKS cross-platform and freely available, it is cheaply and inherently parallelized. Simulating a system of 1000 robots causes no slow-down to the system, beyond initial setup time. This opens the door to a variety of interesting multi-robot and multi-agent evolutionary robotics experiments. It is hypothesized that multiple parallel instantiations of BROOKS can exist simultaneously, however this has yet to be proven - and the transfer of data between instances is particularly difficult.

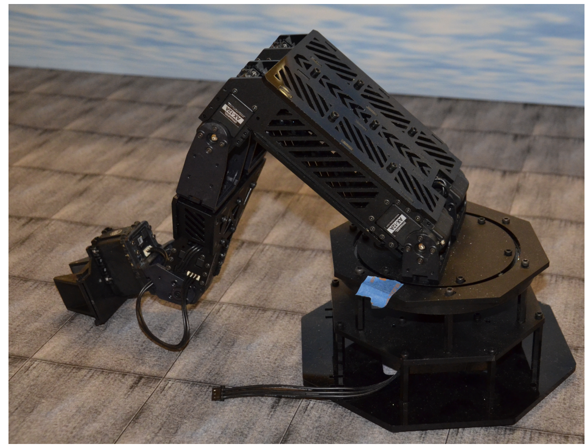

Figure 4: The BROOKS simulator can model a wide variety of robots, such as this dexterous manipulator arm. Realistic high-resolution graphics rendering, including shadows, adds to the veracity of the simulation and aids in debugging.

Integrated Graphics and Sound

The graphics library built into BROOKS is optimized for speed, and hyper-realistic. In fact, it has been hardware-optimized in such a way that using the graphics engine contributes no latency to the simulation in any way, so there is no incentive to disable the graphics functions to speed processing. The graphics library can closely replicate all components and joints within a structure, as well as realistic shadows, textures, and even lens flare – all of which are helpful in debugging. Similarly, BROOKS capably models the a wide range of realistic sounds.

4 CURRENT WEAKNESSES OF BROOKS

Unfortunately, as with all simulators, BROOKS has some notable trade-offs, shortcomings, and undesired features that are the cost for all of the features described above.

Fixed Timestep

The time-step of BROOKS is effectively hard-coded, and cannot be effectively changed – although a built-in "strobe" feature can have the effect of slowing down a variety of phenomena. There exist a few theoretical methods of altering the length of time passed in some locations within BROOKS relative to the operator of the simulation. However, these methods are unlikely to be practical in most situations. Advancements in the navigability of the space within a BROOKS simulation, which are required to make this change to the timestep possible, are not expected to become available in the near future.

Figure 5: The BROOKS simulator can model the dynamics of soft-bodied robots with a multiple degrees of freedom, including this model of a completely soft 3D-printed robot.

Region and Locale Specific

Some BROOKS modules may not work in all regions or locales and highly specific environments will not be available in most simulation settings.

Hard-Coded constants

Many of the fundamental constants, including gravity, are either hard-coded or vary regionally , and can be complex to overcome (although in some cases different gravity settings can be modeled in aquatic and high-altitude settings). Simulating environments on frictionless surfaces, for instance, is difficult. This is in part due to the fact that BROOKS is fine-tuned to correspond directly to real-world results. As real-world technology expands in a way that requires new options for environments and the gravitational constant, new settings will become available to some specific users. Rather than attempt to change these setting directly, users are encouraged to leave most default settings, like Earth gravity and the speed of light, untouched. Measurements within BROOKS work best with the standards set by the International System of Units.

Infrequent Updates

The BROOKS simulator does not appear to have been updated at any point since its inception, and it is unlikely to receive an update in the foreseeable future. In fact, any update to the physics engine would dramatically alter the accuracy of previous BROOKS simulations to the real world. For this reason, updates are discouraged.

5 CONCLUSIONS

Our claim is that the BROOKS simulator sets the standard for high-precision and realistic physics simulators, capable of high-fidelity modeling of a diverse range of robot morphologies and environments. Although other simulators may have the edge in some features, particularly in terms of being able to speed up evaluation at the expense of fidelity, we argue that our simulator should serve as a type of baseline, against which most other simulators should be compared.

ACKNOWLEDGMENTS

The authors would like to thank the undergraduate research assistants of the Union College Computer Science Department. The authors would also like to thank the workshop organizers for their open-minded inclusivity. Finally, and above all, we would like to thank the readers for their sense of humor and tolerance for satire (granted, satire with a serious agenda).

REFERENCES

[1] Josh Bongard, Victor Zykov, and Hod Lipson. 2006. Resilient machines through continuous self-modeling. *Science* 314, 5802 (2006), 1118–1121.

[2] Josh C Bongard and Rolf Pfeifer. 2003. Evolving complete agents using artificial ontogeny. In *Morpho-functional Machines: The new species*. Springer, 237–258.

[3] Rodney A Brooks. 1990. Elephants don't play chess. *Robotics and autonomous systems* 6, 1-2 (1990), 3–15.

[4] Ken Caluwaerts, Jérémie Despraz, Atıl Işçen, Andrew P Sabelhaus, Jonathan Bruce, Benjamin Schrauwen, and Vytas SunSpiral. 2014. Design and control of compliant tensegrity robots through simulation and hardware validation. *Journal of The Royal Society Interface* 11, 98 (2014), 20140520.

[5] Dave Cliff, Philip Husbands, and Inman Harvey. 1993. Evolving visually guided robots. In *From Animals to Animats 2. Proceedings of the Second International Conference on Simulation of Adaptive Behavior*, Jean-Arcady Meyer, Herbert L Roitblat, and Stewart W. Wilson (Eds.). MIT Press, Cambridge MA, 374–383.

[6] Antoine Cully, Jeff Clune, Danesh Tarapore, and Jean-Baptiste Mouret. 2015. Robots that can adapt like animals. *Nature* 521, 7553 (2015), 503–507.

[7] Dario Floreano and Francesco Mondada. 1994. Automatic creation of an autonomous agent: Genetic evolution of a neural network driven robot. In *From Animals to Animats 3: Proceedings of the Third International Conference on Simulation of Adaptive Behavior*, Dave Cliff, Philip Husbands, Jean-Arcady Meyer, and Stewart W. Wilson (Eds.). MIT Press, 421–430.

[8] Kyrre Glette and Mats Hovin. 2010. Evolution of artificial muscle-based robotic locomotion in PhysX. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*. IEEE, 1114–1119.

[9] Inman Harvey, Phil Husbands, and Dave Cliff. 1994. Seeing the light: artificial evolution, real vision. In *From Animals to Animats 3:Proceedings of the Third International Conference on Simulation of Adaptive Behavior*, Dave Cliff, Philip Husbands, Jean-Arcady Meyer, and Stewart W. Wilson (Eds.). MIT Press, Cambridge MA, 392–401.

[10] Jonathan Hiller and Hod Lipson. 2012. Automatic design and manufacture of soft robots. *IEEE Transactions on Robotics* 28, 2 (2012), 457–466.

[11] Gregory S Hornby and Jordan B Pollack. 2001. Body-brain co-evolution using L-systems as a generative encoding. In *Proceedings of the 3rd Annual Conference on Genetic and Evolutionary Computation*. Morgan Kaufmann Publishers Inc., 868–875.

[12] Nick Jakobi, Phil Husbands, and Inman Harvey. 1995. Noise and the reality gap: The use of simulation in evolutionary robotics. *Advances in artificial life* (1995), 704–720.

[13] Jean-Baptiste Mouret, Sylvain Koos, and Stéphane Doncieux. 2013. Crossing the reality gap: a short introduction to the transferability approach. *arXiv preprint arXiv:1307.1870* (2013).

[14] John Rieffel, Francisco Valero-Cuevas, and Hod Lipson. 2009. Automated discovery and optimization of large irregular tensegrity structures. *Computers & Structures* 87, 5 (2009), 368–379.

[15] John A Rieffel, Francisco J Valero-Cuevas, and Hod Lipson. 2010. Morphological communication: exploiting coupled dynamics in a complex mechanical structure to achieve locomotion. *Journal of the royal society interface* 7, 45 (2010), 613–621.

[16] Karl Sims. 1994. Evolving virtual creatures. In *Proceedings of the 21st annual conference on Computer graphics and interactive techniques*. ACM, 15–22.

[17] Adrian Thompson. 1996. An evolved circuit, intrinsic in silicon, entwined with physics. In *Proceedings of the First International Conference on Evolvable Systems: From Biology to Hardware*, Tetsuya Higuchi, Masaya Iwata, and Weixin Liu (Eds.). Springer, 390–405.

[18] Richard A Watson, Sevan G Ficici, and Jordan B Pollack. 2002. Embodied evolution: Distributing an evolutionary algorithm in a population of robots. *Robotics and Autonomous Systems* 39, 1 (2002), 1–18.

[19] Viktor Zykov, Josh Bongard, and Hod Lipson. 2004. Evolving dynamic gaits on a physical robot. In *Proceedings of Genetic and Evolutionary Computation Conference (GECCO), Late Breaking Paper, GECCO*, K. Deb, R. Poli, W. Banzhaf, H.-G. Beyer, E. Burke, P. Darwen, D. Dasgupta, D. Floreano, J. Foster, M. Harman, O. Holland, P.L. Lanzi, L. Spector, A.G.B. Tettamanzi, D. Thierens, and A. Tyrrell (Eds.), Vol. 4. Springer.