The Seventeenth International Symposium on Artificial Life and Robotics 2012 (AROB 17th '12),
B-Con Plaza, Beppu, Oita, Japan, January 19-21, 2012

Evolution of locomotion in a simulated quadruped robot and transferral to reality

Kyrre Glette[1], Gordon Klaus[1], Juan Cristobal Zagal[2], and Jim Torresen[1]

[1]University of Oslo, Norway
[2]University of Chile, Chile
[1]{kyrrehg, gordonk, jimtoer}@ifi.uio.no, [2]jczagal@ing.uchile.cl

Abstract: In this paper, we study the suitability of using simulation in the evolution of locomotion in a quadruped robot. The goal of the evolution is to design a control system that produces fast gaits. We evolve gaits in simulation, and then the best controllers are transferred into the real custom built robot and compared with their simulated versions. The results show effective locomotion, with a 1.8 times improvement in speed over earlier results. Finally, we investigate some measures to reduce the difference between simulated and real locomotion.

Keywords: robotics, simulation, genetic algorithms

1 INTRODUCTION

It can be a challenging task to manually design and optimize a control system that enables a robot to walk, particularly an unconventional robot in a complex environment. Evolutionary design allows us to automate this otherwise time consuming and intellectually demanding process [1, 2]. Furthermore, it provides a general procedure that can be applied to various robot morphologies, and can even be used for adapting to changes in the environment or in the robot itself such as damage to or loss of limbs [3]. Importantly, evolution is unconstrained by engineering conventions, and is thus able to find solutions that a human is unlikely to find.

For time-consuming tasks like robotic locomotion, it becomes impractical to perform large-scale evolutions involving tens of thousands of evaluations on the target hardware, due both to time needed and to mechanical wear. For example, in [4], a total of 1217 evaluations were performed on the target hardware, spread across 7 different learning algorithms; and as those authors point out, this number is much smaller than desirable in terms of achieving both high quality results and statistical significance, and certainly took a great deal of time to carry out. (See [5] for another such example.) To overcome this issue, evaluations can instead be performed in a simulator. With sufficient CPU power, many individuals can be evaluated in simulation in the time it would take to evaluate one in reality, while at the same time eliminating mechanical wear and unreliable human intervention.

Due to unavoidable inaccuracies in the simulator, one encounters what is known as the "reality gap" [6], that is, a difference in performance between the simulation and the real system. Despite this problem, reasonable-quality solutions can still be found by evolution which at least qualitatively reflect the behavior predicted by simulation when they are transferred to reality. However, tuning the simulator to reduce the reality gap should do more than just make designs more transferable; we expect it will allow simulated agents to cope with and exploit otherwise inaccurately or completely unrepresented dynamics of the system, thus reshaping or opening up new territory for evolution to explore. Several interesting approaches are currently being explored for dealing with the issue of the reality gap [7, 8, 9, 10].

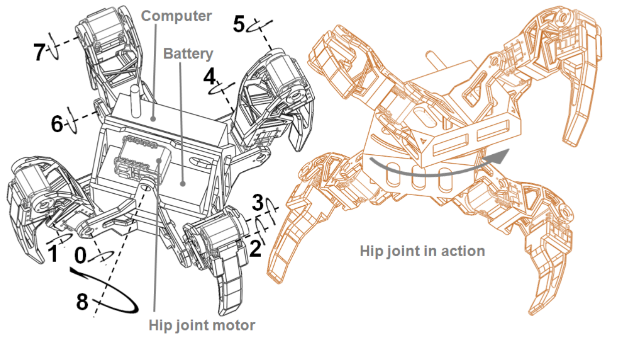

Fig. 1: Schematic view of the robot with the nine joints and associated servos labeled.

In this work, we investigate the suitability of using simulation in the evolution of locomotion in the QuadraTot quadruped robot. The robot has some unconventional characteristics, such as limited motion of the legs and a hip joint allowing the body to twist; see Fig. 1. Given this hardware, the goal of evolution is to design a control system that produces the fastest gait – i.e., the gait that results in the greatest distance covered in a fixed time period. The best controllers that come out of this process are then transferred into the real robot and compared with their simulated versions. Finally, we investigate some measures to reduce the difference between simulated and real locomotion.

2 IMPLEMENTATION

The QuadraTot robot, pictured in Figs. 1 and 2, has 9 joints: two in each leg and one at its center, each actuated by a Dynamixel AX-18A (AX-12A in the "knee" joints) servo. Fully extended, it measures 68 cm across; crouched, 34 cm. With the power supply separate from the body in order to

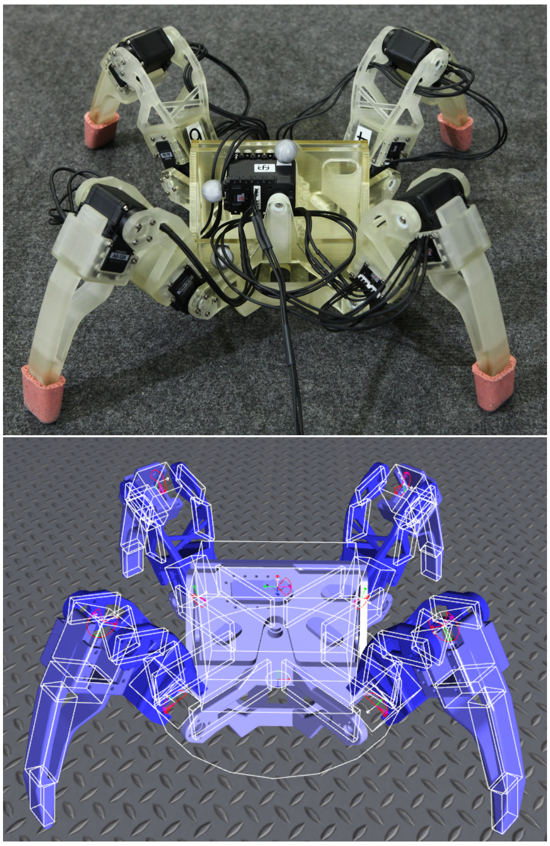

Fig. 2: The robot in reality and in simulation. Note the spherical markers for motion capture on the real robot. The white lines show the shapes for the simulation model.

minimize the strain on the relatively weak servos, the robot weighs 1.4 kg. The body parts[1] were printed on the Objet Connex 500 3D Printing System. One feature was added beyond the original design of the robot: silicone rubber "socks" were attached to its feet to improve traction.

To simulate the dynamics of the QuadraTot robot, we employed the NVIDIA PhysX physics simulation software library. PhysX provides accurate approximations of 3D rigid body motions, collision detection, and motorized joints, to name a few of the features relevant to our work. We built a model of the robot in PhysX to capture the salient aspects of its design such as the lengths of its body parts, positions and types of its joints, and the masses and rough shapes of its parts. The building blocks for the part shapes were boxes of various sizes and orientations. A detailed 3D mesh was presented using OpenGL to make visualization of the simulation more appealing and intuitive. The shapes and the overlying mesh can be seen in Fig. 2. The servos were modeled as motorized revolute joints, where the motor force was proportional to the difference between the actual and target angle; joint parameters were calibrated by observing the response of the real servos.

[1]Available online at `http://creativemachines.cornell.edu/evolved-quadruped-gaits`

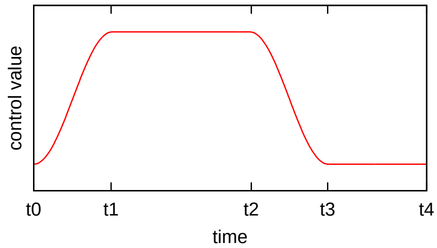

Fig. 3: The motor control function.

To control the motors in each of the robot's joints, we implemented a simple periodic control system consisting of a parameterized smoothed pulse, illustrated in Fig. 3. All controllers operate at the same frequency, but have different curve parameters, as well as individual phase shifts, ϕ. The *attack* parameter decides the time between t_0 and t_1, $pause_0$ the time between t_1 and t_2, *decay* the time between t_2 and t_3, and $pause_1$ the time between t_3 and t_4. The controller output is further parameterized by selecting the *center angle* for the pulse, as well as the *amplitude*.

Our reason for choosing such a simple system is threefold: First, the control system is not the focus of this work; second, in nature, locomotion on flat ground is observed to involve simple periodic motion; and third, the relatively small number of parameters reduces the size of search space for the genetic algorithm (GA), thus speeding up evolution.

For the evolutionary runs, we used the Simple GA method of the GALib software library. The fitness value for the experiments was calculated as the average speed of the robot during the evaluation, i.e., the total distance traveled divided by the simulated evaluation time. The joint parameters were encoded for each of the 9 joints in a binary genome with a total length of 314 bits. A population of 200 individuals was evolved for 300 generations. The bit-flip mutation probability was set to the inverse of the genome length. One point crossover was used with a probability of 0.2.

To observe the motion of the real robot, we employed an infrared camera-based NaturalPoint OptiTrack motion capture system consisting of 8 cameras. Four reflective markers were placed on the robot core to identify its motion. The position of one of these markers, sampled at 60Hz, was logged for the experimental results in the next section.

3 EXPERIMENTAL RESULTS

Where possible, we set simulator parameters to exactly known values, such as the force of gravity and the masses and dimensions of body parts. We set joint angle limits to constrain the motion to a range consistent with the physical robot: the inner joints could swing between -106°and 75°, the outer joints between -141°and 125°, and the hip joint between -53°and 53°. Joint motor forces were set to a very large value. We guessed, rather than measured, the coefficients of static and dynamic friction to both be 1.0. Finally, we chose values for the skin width (i.e., the depth to which

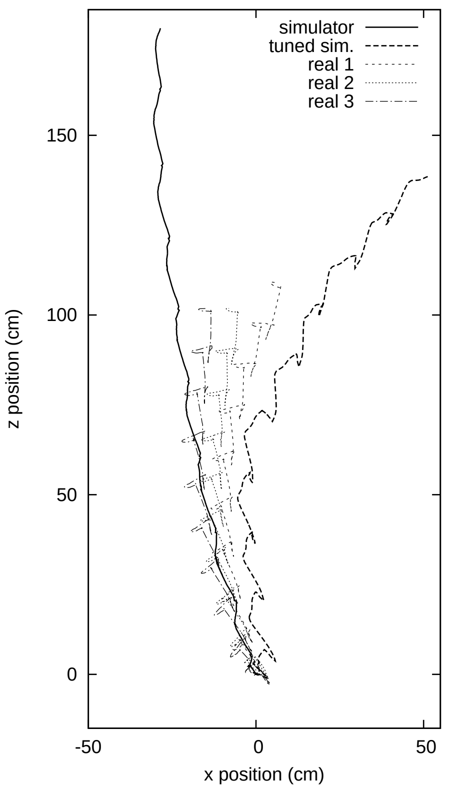

Fig. 4: Horizonal position plot, gait 1 over 10 seconds.

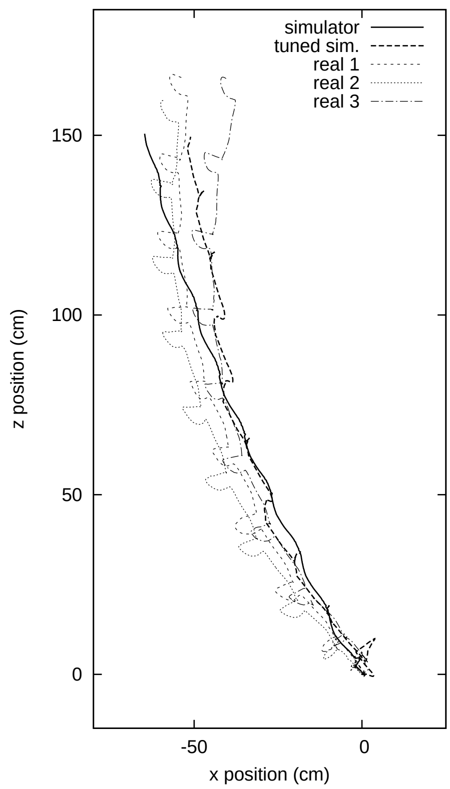

Fig. 5: Horizonal position plot, gait 2 over 10 seconds.

objects are allowed to interpenetrate before being considered to have collided), the joint constraint solver iteration count, and the simulator time step as recommended by the PhysX documentation and from prior experience to 0.05 cm, 200 iterations, and $\frac{1}{60}$ s, respectively.

With the simulator thus configured, we ran the GA to produce some gait patterns, and selected two of the fittest gaits from the final generation based on their qualitatively different strategies. The selected gaits were transferred into the real robot and their positions were recorded[2] for a period of 10 seconds. The resulting gait speeds are compared in Table 1. We then tuned the simulator as an initial attempt to reduce the reality gap for these two gaits. Both static and dynamic friction coefficients were reduced to 0.4. Also, to reproduce a perceived differential slipping tendency in the real robot's feet, anisotropic friction was added such that the feet would slip more easily from side to side (friction coefficients set to 0.1 in this direction) than front to back (friction coefficients set to 0.8 in this direction). Figs. 4 and 5 compare the real and simulated (both before and after tuning) motions for these gaits.

4 DISCUSSION

We observe from the experiments that the GA found promising gaits given the robot design and the imposed restrictions. The gaits shown in Figs. 4 and 5 employ two different strategies that could be similar to what a human designer might implement: the former is reminiscent of a quadruped animal; and the latter uses a worm-like locomotion strategy where two of the legs serve mostly as support. However, it appears that the solutions could be further tuned to improve performance. The parameter-based model imposed several constraints, including the amount of rotation possible in each joint, thus limiting the range of motion of the limbs. If a greater range of motion were allowed, this would result in both a larger search space and the possibility of exceeding physical limitations; however, the simulator-based approach would be more suitable for this than risking invalid combinations on the target hardware. We noticed that the core joint of the robot was not exploited noticeably in the evolved gaits. This may be due to the fact that its movement was restricted due to frequent motor overload on the real robot. A better simulation of the allowed forces could open up a path towards higher quality gaits.

The observed locomotion speeds were larger than those in found by previous online evolution experiments. Our best

Table 1: Evolved gait speeds

gait	simulated	real (avg 3)
gait 1	18.3 cm/s	10.5 cm/s
gait 2	16.4 cm/s	17.8 cm/s
best in [4]	NA	9.7 cm/s

[2]Gait videos: `http://folk.uio.no/kyrrehg/quad`

gait in reality achieved 17.8 cm/s, whereas highest speed found by [4] was only 9.7 cm/s. This difference could be explained by the different control systems and their parameterization or by the larger number of evaluations performed in the evolutionary search with the simulator. Additionally, it should be noted that there are some differences in the experimental setups, in particular with regard to the surface friction and the material used for coating the robot feet, as well as some differences in servo models.

As expected, we observed a difference between the simulated robot behavior and the behavior in the real robot, although it was not as large as anticipated. From the observations of the first gait, the qualitative impression of the behavior is the same, but the real robot turns faster. It seems that this is because of differences in friction between the simulator and reality, which might also account for the more jagged trajectory observed in the position plot. The reality gap observed in the second gait was smaller; although the trajectories were not identical, there was little impact on the final position. We suspect that friction played a smaller role in this gait due to the different contact motion between the legs and the surface, and the initial results from tuning the simulator support this claim. These results indicate that the simulator gives a fairly realistic model of the real robot and that additional tuning is likely to further improve the realism.

We observed during the experiments on the real robot that, after several runs, the motors seemed to lose energy, eventually shutting down with an overload error. We tried to mitigate this problem by disconnecting the power between runs. This should be addressed in future experiments by carefully tuning the model of the motor strengths in the simulator and possibly also instructing the evolution to discourage locomotion which stresses the motors too much.

5 CONCLUSION AND FUTURE WORK

We have developed a simulation environment for an unconventional robot design, where learning has previously only been performed on the real robot. This has allowed design of locomotion through a GA in which evaluations were performed in a simulator, saving time, resources, and wear associated with running EAs on real robots. The results using a simple parametric approach are better than an earlier approach using real world evolution for parametric gaits as well as HyperNEAT. A promising path of future work would be to investigate the use of more sophisticated control algorithms on the proposed evolutionary setup, for even more efficient locomotion. Some of the evolved gaits suffer from the reality gap when transferred to the real robot, whereas others perform closer to the simulated behavior. Tuning of some simulator parameters seems to be one way to improve transferral to reality. Future work should investigate various mechanisms for facilitating the transfer from simulated solutions to real ones, including more sophisticated motion capture of the robot limbs, better and more automated tuning of the simulator, as well as evolving for robustness, e.g. by introducing noise into the simulation.

ACKNOWLEDGEMENTS

The authors would like to thank Jason Yosinski for sharing experience from earlier experiments on the robot as well as providing helpful comments on the paper.

REFERENCES

[1] S. Doncieux, N. Bredeche, and J.-B. Mouret, "Exploring new horizons in evolutionary design of robots," in *Proceedings of the IEEE/RSJ international conference on Intelligent robots and systems (IROS)*, 2009.

[2] I. Harvey, P. Husbands, D. Cliff, A. Thompson, and N. Jakobi, "Evolutionary robotics: the sussex approach," *Robotics and Autonomous Systems*, vol. 20, pp. 205–224, 1997.

[3] H. Lipson, J. C. Bongard, V. Zykov, and E. Malone, "Evolutionary robotics for legged machines: From simulation to physical reality," in *Proceedings of the 9th Int. Conference on Intelligent Autonomous Systems*, pp. 11–18, 2006.

[4] J. Yosinski, J. Clune, D. Hidalgo, S. Nguyen, J. C. Zagal, and H. Lipson, "Evolving robot gaits in hardware: the hyperneat generative encoding vs. parameter optimization.," in *Proceedings of the 20th European Conference on Artificial Life*, pp. 890–897, 2011.

[5] V. Zykov, J. C. Bongard, and H. Lipson, "Evolving dynamic gaits on a physical robot," in *Proceedings of Genetic and Evolutionary Computation Conference, Late Breaking Paper, GECCO*, 2004.

[6] N. Jakobi, P. Husbands, and I. Harvey, "Noise and the reality gap: The use of simulation in evolutionary robotics," in *Advances in Artificial Life: Proc. 3rd European Conference on Artificial Life*, pp. 704–720, Springer-Verlag, 1995.

[7] S. Koos, J.-B. Mouret, and S. Doncieux, "Crossing the reality gap in evolutionary robotics by promoting transferable controllers," in *Proceedings of the 12th annual conference on Genetic and evolutionary computation*, GECCO '10, pp. 119–126, 2010.

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

[9] J. C. Bongard, V. Zykov, and H. Lipson, "Resilient Machines Through Continuous Self-Modeling," *Science*, vol. 314, no. 5802, pp. 1118–1121, 2006.

[10] J. Bongard, "Synthesizing physically-realistic environmental models from robot exploration," in *Proceedings of the 9th European conference on Advances in artificial life*, ECAL'07, pp. 806–815, 2007.