Filling the Reality Gap: Using Obstacles to Promote Robust Gaits in Evolutionary Robotics

Kyrre Glette, Andreas Leret Johnsen, and Eivind Samuelsen
Department of Informatics
University of Oslo
P.O. Box 1080 Blindern, 0316 Oslo, Norway
Email: {kyrrehg,andrlj,eivinsam}@ifi.uio.no

***Abstract*—In evolutionary robotics, which concerns automatic design of robotic systems using evolutionary algorithms, the well-known reality gap phenomenon occurs when transferring results from simulation to real world robots. Several approaches have been proposed to tackle this challenge, such as improving the simulator, avoiding poorly simulated solutions, or promoting robust controllers by introducing noise in the simulation. In this paper we investigate if the addition of a set of small obstacles in the simulated environment can help promote more robust gaits when transferred to a real world robot. In total 80 robot gaits are tested in the real world, evolved using flat and obstacle-seeded ground planes, and using two different scenario difficulties. The results show that in the baseline scenario the proposed obstacle method has little impact on the reality gap of the evolved gaits, whereas there is a significant reduction for the difficult scenario: The average real world performance ratio is 2.3 times higher than the result obtained with the flat plane, and there are no null-performing gaits.**

I. Introduction and Related Work

In hostile and remote environments it would be a great advantage to deploy robots which display properties such as robustness, adaptivity, and flexible locomotion strategies, in order to remove the need for human intervention as much as possible. In such environments, legged or more unconventional mechanical strategies for locomotion may be relevant. Evolutionary robotics attempts to automatically optimize robotic systems through the use of evolutionary algorithms, i.e., population-based search algorithms inspired by evolution in nature. ER has shown considerable success at optimizing locomotion control strategies for robot bodies, like four- or six-limbed walking robots [1], [2], random morphologies [3], or even flying robots [4].

While evolving uniquely on the targeted real-world robot would intuitively give good results, and there are indeed several successful reports of this [1], [5]–[7], the approach also has some shortcomings: Firstly, as the time required to evaluate a single gait is usually several seconds, the time required to perform a full evolutionary search with hundreds of individuals and generations can become prohibitively expensive. E.g. in [1] one evolutionary run was restricted to 500 evaluations, which took 25 hours. Secondly, the excessive real-world evaluation of candidate solutions, which are explorative and may sometimes contain invalid or extreme behavior, may quickly lead to wear and breakdown of the robot platform [6], [7]. Finally, the evaluation of a large number of solutions usually requires some degree of supervision as the robot quickly can fall over, get stuck, or otherwise perform invalid actions which hampers the continued evaluation of solutions.

Due to the abovementioned reasons there is a strong argument for introducing simulation in the evolutionary search, as it eliminates the need for supervision, there are no wear and tear issues, and a high amount of CPU power allows for a large increase in the number of practical evaluations. Such a higher budget for evaluation may also increase in improved results for more complex control strategies [8]. The introduction of simulation-based evolutionary robotics comes with the *reality gap* challenge, that is, controllers evolved in simulation often perform much worse on the real robot [9]. This is due to inaccuracies in simulation combined with evolution's tendency to produce overfit solutions. Thus, one may end up with solutions which exploit features nonexistent in the real world.

One approach seeking to minimize the effect of the reality gap is to make the simulation model more accurate. This could be done by manually tuning the simulator, or automatic tuning by improving an initial assumption through feedback from the real world [10]–[12]. Here, one strategy would be to only sample from the real world in areas where the good solutions are thought to be. On the other hand, the estimation-exploration approach automatically builds a complete simulation model through exploratory actions on the real robot [13]. However as pointed out in [14] it may be a challenge for the approach to produce a sufficiently accurate self-model from a limited number of exploratory actions. Another approach to the reality gap challenge is to acknowledge that some areas of the search space are poorly simulated and thus do not transfer to good solutions on the real robot. Instead, by sampling the performance of some solutions in the real world, one builds an estimate of the transferability of solutions, and avoid searching in the areas which are thought to transfer poorly [15].

A classic approach to minimizing the effect of the reality gap is to introduce noise in the simulations, in order to promote more robust solutions [9]. Noise can be applied in sensors and actuators, and should help the search avoid overfitting to untransferable solutions. It is however pointed out that the right amount of noise is required to produce satisfactory results [9].

Interestingly, some related challenges were encountered in [1] when evolving gaits purely on a real world quadruped robotic platform. In fact, when evolving gaits on a specific flat carpet, the resulting gaits were dragging the robot's feet, resulting in poor transferability from the carpet to differently textured surfaces. The experimenters introduced a set of small

obstacles on the ground during evolution, which in turn resulted in a gait where the robot lifted its feet and were more transferable to the other surfaces. This example illustrates well that the reality gap is present also when transferring between different environments in the real world, and as such advocates the need for robust solutions rather than simulators highly tuned for one specific environment. In most real world robotic applications the environment is likely to change to some degree, and thus we would like at least some degree of robustness in the evolved strategies.

In this paper, we draw inspiration from the small rods used in the real-world evolutionary experiments in [1], and investigate whether an approach of using small obstacles to roughen the surface can improve robustness in a simulation-oriented evolutionary gait design process. Specifically, we would like to investigate if this approach could lessen the reality gap commonly associated with evolutionary robotics. To this means, we evolve several robotic gaits in a simulated obstacle-enabled environment, and compare their performances, including their real-world performance, against robotic gaits evolved in a totally flat environment. Moreover, we introduce scenarios with two levels of difficulty, i.e., in one of the experiments the locomotion ability is limited, making the search for a gait considerably harder.

The remainder of the paper is structured as follows: Sec. II describes the employed robot, the simulation model, and how its control system can be optimized using an evolutionary search. Then, Sec. III documents the performed experiments and presents the obtained results, which are subsequently discussed in Sec. IV. Finally, conclusions and directions for future work are given in Sec. V.

II. Robot and Simulation

This section describes the robot used for our experiments, its simulation model and the encoding of the control system to be optimized by an evolutionary algorithm.

A. QuadraTot robot

For the evolutionary experiments we employ the QuadraTot quadrupedal robot platform, first introduced in [7]. A photo of the robot can be seen in Fig. 1. The robot has 9 joints, 2 for each of the 4 legs, and one in the center part, which gives an unconventional mechanical design. Movement of the legs is more restricted than usual, whereas the center joint gives an additional but unusual degree of freedom. The robot design is intentionally unconventional, suitable for evolutionary robotics experiments. Moreover, the robot design is open source, using off-the-shelf components and body parts which can be downloaded for printing on a 3D printer[1]. The joints are actuated by Robotis Dynamixel AX series servos, where the inner joints use the stronger AX-18A model and the outer ("knee") joints use AX-12A model. While the robot is initially designed for carrying a battery and a small computer in its body, these components have been removed and are replaced with an umbilical cable supplying power and control signals from an external computer. This makes the robot lighter and less prone to servo overload. In our copy of the robot we have printed the parts with an Object Connex 500 3D printer and attached silicone rubber "socks" on the legs for improved traction. The total weight of the robot is 1.4 kg, and in the crouched position it measures 34 cm across.

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

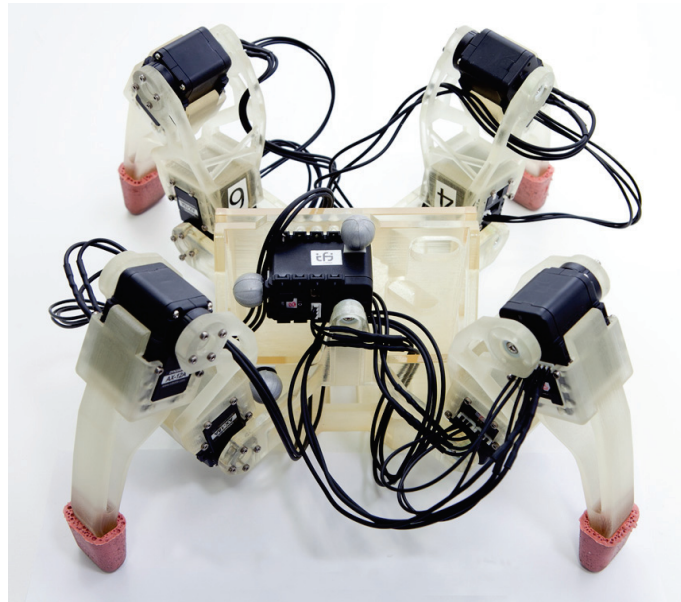

Fig. 1. The QuadraTot robot. Note the servos working as revolute joints in the legs and the central servo enabling the base to be twisted. Reflective markers are attached to enable motion capture.

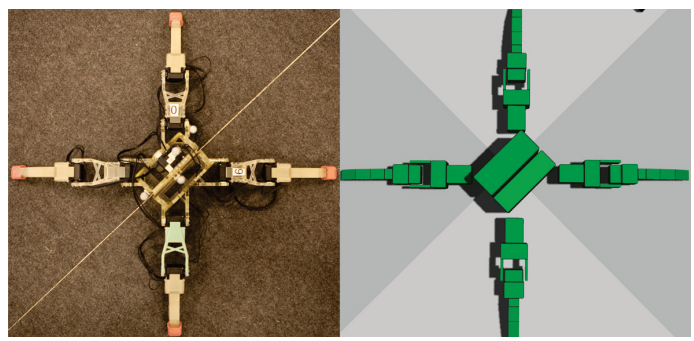

Fig. 2. Top view of the robot and the corresponding simulation model.

B. Simulation

The rigid body physics are simulated using the Nvidia PhysX engine (version 3.3 beta-2), and a model was constructed to capture the most important features of the QuadraTot robot. Boxes were used as approximations for the body parts, and a top view of the constructed model can be seen in Fig. 2. It should be noted that the simulation framework used in this paper is an updated version compared to the one presented in previous experiments [8], [16]. The simulation is run with a fixed timestep of 1/128 s. The rigid body parts were combined using the *articulation* system in PhysX, which is supposed to be a robust solution method for linked bodies, at the expense of being computationally more expensive than using standard joints. The robot parts were modeled with both static and dynamic friction coefficients of 0.7, and the ground plane and obstacles had static and dynamic friction coefficients of 0.2 and 0.15, respectively.

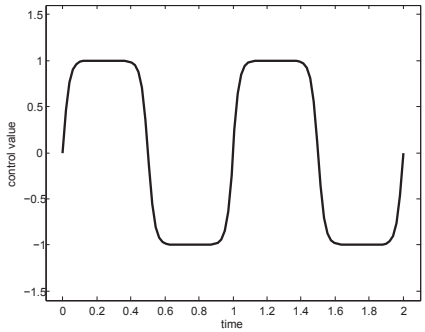

Fig. 3. Example plot of the control function.

TABLE I. JOINT PARAMETER RANGES, IN DEGREES

Parameter	Min.	Max.
Center Amplitude	0	22.5
Center Offset	-22.5	22.5
Center Phase	-180	180
Hip Amplitude	0	45
Hip Offset	-63	-17.2
Hip Phase	-180	180
Knee Amplitude	0	45
Knee Offset	61.4	119
Knee Phase	-180	180

C. Control function

The control function for each of the 9 joints is inspired by [14], and can be expressed as:

$$c_{\alpha,\phi,\beta}(t) = \alpha \tanh(4 \sin(2\pi(t + \phi))) + \beta$$

Where c is the periodic function computing the control angle of the servo as a function of time t, and α, ϕ, and β adjust the amplitude, the phase, and the angular offset, respectively. The $\tanh$ component is added to give a flattening effect on the extremities of the wave, which in turn may lead to longer ground contact times and a more stable gait. An example plot of the control function can be seen in Fig. 3.

D. Genome Representation and Mutation

The evolvable parts of the control function are the α, ϕ, and β parameters, and these are allowed to evolve differently for each of the 9 joints. This gives a genome consisting of 27 floating point values, which are allowed to evolve within the ranges shown in Tab I. The "hip" and "knee" joints share the same parameter ranges, but are otherwise evolved independently of each other.

Based on experimentation, only mutation is used as an evolutionary operator; Gaussian mutation with a standard deviation of 2.865 degrees is applied to every value in the genome. For the evolutionary runs, all genome values are initially set to zero and then mutated 20 times.

III. EXPERIMENTS AND RESULTS

This section describes the experimental details, the two main experiments, and presents the results.

A. Scenarios and fitness function

For the experiments in simulation we employed two main types of scenarios: a flat plane, and a plane with obstacles. In the obstacle scenarios five different random seeds were used to generate five different configurations of boxes of varying dimensions in front of the robot. The length and width of the boxes were in the range 1-21 cm, and the height between 1-3 cm. An example of an obstacle scenario can be seen in Fig. 4. In all experiments, the fitness measure is how far the robot is able to move in a predefined direction in the ground plane during 8 seconds. The direction is perpendicular to the axis of the center joint and parallel to the ground plane.

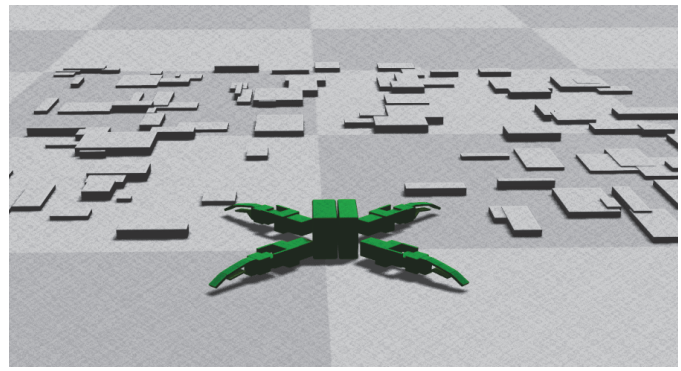

Fig. 4. One of the obstacle scenarios used in the experiments. The robot is in its starting position and will be rewarded fitness for walking straight ahead, away from the camera

B. Baseline experiment

For evolution we use a simple real-valued GA. Parent selection is done by binary tournaments, and survivor selection is done by truncation, keeping the best individuals. The population size is 256, and each run consists of 200 generations. First, 100 runs are conducted with the flat plane as the fitness scenario, and then a total 100 runs are conducted with obstacles, that is, 20 runs for each of the different obstacle scenarios.

For the 100 runs with the flat scenario, we choose the 20 best gaits, all from different runs, for real-world evaluation. For the obstacle runs, we choose the 4 best gaits from different runs from each of the obstacle scenarios, giving a total of 20 gaits. Real-world evaluation of the simulation-evolved gaits is performed by placing the robot on a flat carpet surface and measuring the distance moved using an OptiTrack motion capture system consisting of 8 infrared cameras. The distance is measured after 8 seconds, and each gait is evaluated 5 times from the same starting point.

C. Restricted experiment

A variation of the baseline experiment was performed, where the robot's control system was limited. This restriction was introduced by forcing all knee joints to move in the same phase, that is, the phase parameter was fixed. The rationale for this approach is to investigate how a more difficult task affects the resulting reality gaps. Apart from the fixed phases of the knee joints, all other experimental details are similar to the baseline experiment.

D. Results

The results from running all 80 selected gaits in a flat simulation environment and thereafter in the real world are

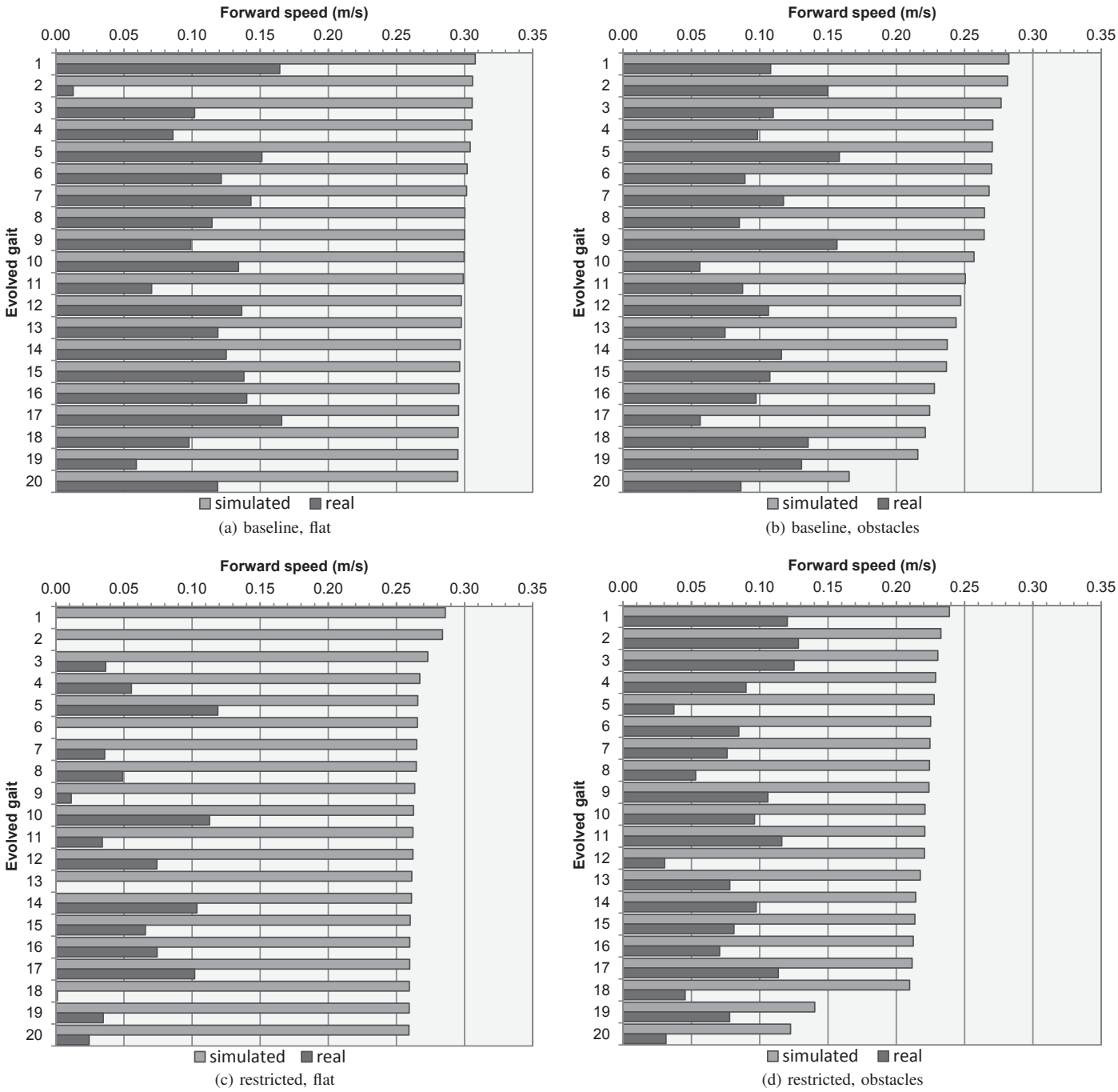

Fig. 5. Flat surface performance in simulator and real world of 20 selected gaits, evolved in different experiments and scenarios. Gaits are sorted by decreasing simulator performance. Note that gait 1 in (c) actually has a real world performance of -0.03 m/s, which is not displayed.

TABLE II. PERFORMANCE MEASURES FROM THE 20 SELECTED GAITS IN EACH OF THE 4 SCENARIOS.

	baseline		restricted	
	flat	obstacles	flat	obstacles
avg. sim (m/s)	0.30	0.25	0.27	0.21
avg. real (m/s)	0.11	0.11	0.04	0.08
min. real (m/s)	0.01	0.06	-0.03	0.03
avg. eval. std.dev.	0.01	0.01	0.01	0.01
avg. $\frac{\text{real}}{\text{simulated}}$	0.38	0.43	0.17	0.39
min. $\frac{\text{real}}{\text{simulated}}$	0.04	0.22	-0.12	0.14

displayed in Fig. 5. The average and minimum values for the 20 gaits from each scenario are shown in Tab. II. Each gait was evaluated five times on the real robot, and the standard deviations for these five measures are averaged and reported in the same table. Here we also display the ratio between the real world performance and the simulated performance as a measure of the reality gap. While average real world performance ratios of flat and obstacle evolution in the baseline experiment are different, the difference between the ratios is not statistically significant. In the restricted experiment,

the difference between the reality gap ratios is statistically significant (at $p < 0.01$, Mann-Whitney U test), and the average real world performance ratio is 2.3 times higher than the result obtained with the flat plane scenario.

IV. Discussion

From the experimental results, we can make the following observations:

- In the baseline experiment, the addition of obstacles in the evolutionary training does not significantly affect the reality gap. Although the slightly higher average reality to simulation ratio as well as higher minimum value for the obstacle-trained gaits as seen in Tab. II may indicate some advantage to the technique, more data is needed to make any conclusions.
- In the restricted experiment however, there is a clear and also statistically significant difference in the reality to simulation ratios when comparing the gaits evolved in the flat environment to those evolved with obstacles. Not only is the reality gap smaller when evolving with obstacles, but also a higher average absolute performance is achieved. In addition, some of the gaits evolved on the flat surface display very poor real-life performance, standing still or even move slightly backwards. It is also interesting to note that the two best performing gaits in simulation, as seen in Fig. 5(c), are among these poor-performing gaits. None of the gaits evolved in the obstacle scenario display such critically poor performance.
- We believe the major reason for the above observations is due to the difficulty of finding a good gait in the restricted scenario, which in turns steers the evolutionary search towards convergence and overfitting in some niches. Some of these solutions may then tend to exploit strategies which are not valid, or have little margin, in the real world. The obstacle scenario in turn then helps the search steer clear of such pitfalls, encouraging solutions where legs are lifted properly. On the other hand, in the baseline experiment the fitness landscape of the search space is probably smoother, and several areas lead to robust performing solutions. It would however be interesting to continue the experiments over a higher number of generations than 200, to see if increased convergence would affect the results.
- While the obstacles could be seen as a kind of noise on the ground plane, they are actually static during an evolutionary run as compared to changing the obstacles for each fitness evaluation. In fact, as the real-life only experiments from [1] suggest, these obstacles are crucial for promoting a robust gait, and play another role than the real-world noise found in actuators and surface friction within the same flat surface. It would still be interesting to compare the efficiency of this technique with noise techniques as employed in other evolutionary robotics experiments, e.g. [9]. It may also be possible that a hybrid noise and obstacle approach could prove efficient. While the restricted experiment may seem artificial in the case of our employed robot, evolutionary robotics is in many cases applied to finding locomotion strategies for complex, limited, or otherwise difficult to control robots. Therefore it is possible that the technique may prove useful also for such cases.
- Although the obstacles reduce the reality gap in the restricted scenario, the reality performance ratio is still below 50% in all cases, meaning there must be other factors playing important roles in the observed performance loss. As no single gait comes close to their simulated performance, we speculate the reason being an overestimate on the speed or strength of the servos. This issue needs to be investigated further if additional experiments are carried out.
- For all approaches, while the simulated performances of different gaits were quite similar, there is significant variation between the real-world performances. This is probably due to different gait strategies with equal simulation performance having different real-world transferability properties. The lack of clear correlations between simulated and real performances makes it difficult to predict the real-world performance of a gait. It is possible that picking gaits with a larger variety in simulated performances could have revealed some correlation. In any case, this topic should be investigated further, as such a predictability would be an essential feature for the usefulness of an evolutionary robotics technique.

V. Conclusion and future work

In this paper we have investigated the addition of small obstacles in simulated scenarios as a means to promote evolution of robust robotic gaits. Several evolved gaits have been tested in real life, and the technique led to a significantly reduced reality gap for a difficult task, while showing little impact for an easier task. The technique can thus be a possible approach to reduce overfitting and reality gap problems for difficult tasks. In the future it would be interesting to validate the technique further by applying it to different robots, and increasing the range of scenarios, such as e.g. investigating different obstacle properties. Further, the approach should be compared to noise based techniques, as well as other techniques for reducing the reality gap, such as the transferability approach [15].

References

[1] G. S. Hornby, S. Takamura, T. Yamamoto, and M. Fujita, "Autonomous evolution of dynamic gaits with two quadruped robots," *Robotics, IEEE Transactions on*, vol. 21, no. 3, pp. 402–410, 2005.

[2] J. Kodjabachian and J.-A. Meyer, "Evolution and development of neural controllers for locomotion, gradient-following, and obstacle-avoidance in artificial insects," *Neural Networks, IEEE Transactions on*, vol. 9, no. 5, pp. 796 –812, sep 1998.

[3] P. Dittrich, A. Burgel, and W. Banzhaf, "Learning to move a robot with random morphology," in *Evolutionary Robotics, First European Workshop, EvoRobot98*, ser. Lecture Notes in Computer Science, vol. 1468. Springer Berlin Heidelberg, 1998, pp. 165–178.

[4] P. Augustsson, K. Wolff, and P. Nordin, "Creation of a learning, flying robot by means of evolution," in *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2002)*, 2002, pp. 1279–1285.

[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] L. M. Garder and M. E. Hovin, "Robot gaits evolved by combining genetic algorithms and binary hill climbing," in *GECCO '06: Proceedings of the 8th annual conference on Genetic and evolutionary computation*. ACM, 2006, pp. 1165–1170.

[7] 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*, 2011, pp. 890–897.

[8] S. Lee, J. Yosinski, K. Glette, H. Lipson, and J. Clune, "Evolving gaits for physical robots with the hyperneat generative encoding: The benefits of simulation," in *Applications of Evolutionary Computation*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2013, vol. 7835, pp. 540–549.

[9] 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*. Springer-Verlag, 1995, pp. 704–720.

[10] J. C. Zagal and J. Ruiz-Del-Solar, "Combining simulation and reality in evolutionary robotics," *J. Intell. Robotics Syst.*, vol. 50, no. 1, pp. 19–39, Sep. 2007.

[11] G. Klaus, K. Glette, and J. Torresen, "A comparison of sampling strategies for parameter estimation of a robot simulator," in *Simulation, Modeling, and Programming for Autonomous Robots*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2012, vol. 7628, pp. 173–184.

[12] R. Moeckel, Y. N. Perov, A. T. Nguyen, M. Vespignani, S. Bonardi, S. Pouya, A. Sproewitz, J. van den Kieboom, F. Wilhelm, and A. J. Ijspeert, "Gait optimization for roombots modular robotsmatching simulation and reality," in *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*. IEEE, 2013, pp. 3265–3272.

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

[14] S. Koos, A. Cully, and J.-B. Mouret, "Fast damage recovery in robotics with the t-resilience algorithm," *International Journal of Robotics Research*, vol. 32, no. 14, pp. 1700–1723, 2013.

[15] S. Koos, J.-B. Mouret, and S. Doncieux, "The transferability approach: Crossing the reality gap in evolutionary robotics," *Evolutionary Computation, IEEE Transactions on*, vol. 17, no. 1, pp. 122–145, 2013.

[16] K. Glette, G. Klaus, J. C. Zagal, and J. Tørresen, "Evolution of locomotion in a simulated quadruped robot and transferral to reality," in *Proceedings of the 17th International Symposium on Artificial Life and Robotics*. ALife Robotics, 2012, pp. 1139–1142.