# High Resilience in Robotics with a Multi-Objective Evolutionary Algorithm

Sylvain Koos syvlain.koos@gmail.com Antoine Cully cully@isir.upmc.fr Jean-Baptiste Mouret mouret@isir.upmc.fr

ISIR, Université Pierre et Marie Curie-Paris 6, CNRS UMR 7222 4 place Jussieu, F-75252, Paris Cedex 05, France

# **Categories and Subject Descriptors**

I.2.6 [Computing Methodologies]: Artificial Intelligence— Learning; I.2.9 [Computing Methodologies]: Artificial Intelligence—Robotics

#### **General Terms**

Algorithms

## Keywords

Evolutionary Robotics; Resilience; Legged robot; Recovery

#### 1. INTRODUCTION

Damage recovery is critical for autonomous robots that need to operate for a long time without assistance. Most current methods are complex and costly because they require anticipating each potential damage in order to have a contingency plan ready.

An alternative line of thought is to let the robot learn on its own the best behavior for the current situation. If the learning process is open enough, then the robot should be able to discover new compensatory behaviors in situations that have not been foreseen by its designers. Classic reinforcement learning algorithms are hard to apply to low-level robotic problems [11], but evolutionary algorithms (EAs) are good candidates to find original solutions because they can optimize in the continuous domain and work on the structure of controllers, for instance by evolving neural networks.

When evolving controllers for robots, EAs are reported to require many hundreds of trials on the robot and to last from two to tens of hours (e.g. [5, 12]). These EAs spend most of their running time in evaluating the quality of controllers by testing them on the target robot. Since, contrary to simulation, reality cannot be sped up, their running time can only be improved by finding strategies to evaluate fewer candidate solutions on the robot.

By first learning learning a self-model for the robot, then evolving a controller with this simulation, Bongard et al. [1] designed an algorithm for resilience that makes an important step in this direction. Nevertheless, this algorithm has a few important shortcomings. First, actions and models are



Figure 1: (A) The hexapod robot is not damaged. (B) The left middle leg is no longer powered. (C) The terminal part of the front right leg is shortened by half.

undirected: the algorithm can "waste" a lot of time to improve parts of the self-model that are irrelevant for the task. Second, it is computationally expensive because it includes a full learning algorithm (the second stage, in simulation) and an expensive process to select each action that is tested on the robot. Third, there is often a "reality gap" between a behavior learned in simulation and the same behavior on the target robot [6], but nothing is included in Bongard's algorithm to prevent such gap to happen: the controller learned in the simulation stage may not work well on the real robot, even if the self-model is accurate. Last, one can challenge the relevance of calling into question the full self-model each time an adaptation is required, for instance if an adaptation is only temporarily useful.

Our algorithm is inspired by the "transferability approach" [9], whose original purpose is to cross the "reality gap" that separates behaviors optimized in simulation to those observed on the target robot[6]. The main proposition of this approach is to make the optimization algorithm aware of the limits of the simulation. To this end, a few controllers are transferred during the optimization and a regression algorithm (here a SVM) is used to approximate the function that maps behaviors in simulation to the difference of performance between simulation and reality. To use this approximated *transferability function*, the single-objective optimization in which both performance in simulation and transferability are maximized. This optimization is performed with a a multi-objective evolutionary algorithm (NSGA-II, [2]).

The same concepts can be applied to design a fast adaptation algorithm for resilient robotics, leading to a new algorithm that we called "T-Resilience" (for Transferabilitybased resilience). If a damaged robot embeds a simulation of itself, then behaviors that rely on damaged parts will not be transferable: they will perform very differently in the

This paper is an extended abstract of [8].

Copyright is held by the author/owner(s).

*GECCO'13 Companion*, July 6–10, 2013, Amsterdam, The Netherlands. ACM 978-1-4503-1964-5/13/07.



Figure 2: Performances (distance covered in 3 seconds) obtained in case B (top) and C (bottom). On each box, the central mark is the median, the edges of the box are the 25th percentile and the 75th percentile. The whiskers extend to the most extreme data point which is no more than 1.5 times the length of the box away from the box. Each algorithm has been run 5 times and distances are measured using the external motion capture system. Except for the T-Resilience, the performance of the controllers found after about 25 transfers (tests) and after about 20 minutes (time) are depicted (all T-Resilience experiments last about 20 minutes and use 25 transfers). The horizontal lines denote the performances of the reference gait, according to the CODA scanner (dashed line) and according to the SLAM algorithm (solid line).

self-model and in reality. During the adaptation process, the robot will thus create an approximated transferability function that classifies behaviors as "working as expected" and "not working as expected". Hence the robot will possess an "intuition" of the damages but it will not explicitly represent or identify them. By optimizing both the transferability and the performance, the algorithm will look for the most efficient behaviors among those that only use the reliable parts of the robots. The robot will thus be able to sustain a functioning behavior when damage occurs by learning to avoid behaviors that it is unable to achieve in the real world. Besides this damage recovery scenario, the T-Resilience algorithm opens a new class of adaptation algorithms that benefit from Moore's law by transferring most of the adaptation time from real experiments to simulations of a self-model.

#### 2. EXPERIMENTS

We evaluate the T-Resilience algorithm on an 18-DOFs hexapod robot that needs to adapt to motor failures and

broken legs (figure 1); we compare it to stochastic local search [4], policy gradient [7] and Bongard's algorithm [1]. The algorithms are implemented in the Sferes<sub>v2</sub> framework [10]. The behavior on the real robot is assessed on-board thanks to a RGB-D sensor coupled with a state-of-the-art SLAM algorithm [3]. For each experiment, a population of 100 controllers is optimized for 1000 generations. Every 40 generations, a controller is randomly selected in the population and transferred on the robot.

Using only 25 tests on the robot and an overall running time of less than one hour on a recent laptop, T-Resilience consistently leads to substantially better results than the other approaches (figure 2).

Videos: http://youtu.be/xbizdSB00Fs

http://pages.isir.upmc.fr/~mouret/t\_resilience/

### ACKNOWLEDGMENTS

Funded by the ANR Creadapt project, ANR-12-JS03-0009 and a UPMC/DGA scholarship to A. Cully.

#### **3. REFERENCES**

- J. Bongard, V. Zykov, and H. Lipson. Resilient machines through continuous self-modeling. *Science*, 314(5802):1118–1121, 2006.
- [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, 2002.
- [3] F. Endres, J. Hess, N. Engelhard, J. Sturm, D. Cremers, and W. Burgard. An evaluation of the RGB-D SLAM system. In *Proc. of ICRA*, 2012.
- [4] H. H. Hoos and T. Stützle. Stochastic local search: Foundations and applications. Morgan Kaufmann, 2005.
- [5] G. Hornby, S. Takamura, T. Yamamoto, and M. Fujita. Autonomous evolution of dynamic gaits with two quadruped robots. *IEEE Transactions on Robotics*, 21(3):402–410, 2005.
- [6] N. Jakobi, P. Husbands, and I. Harvey. Noise and the reality gap: The use of simulation in evolutionary robotics. *Proc. of ECAL*, pages 704–720, 1995.
- [7] N. Kohl and P. Stone. Policy gradient reinforcement learning for fast quadrupedal locomotion. In *Proc. of ICRA*, volume 3, pages 2619–2624. IEEE, 2004.
- [8] S. Koos, A. Cully, and J.-B. Mouret. Fast damage recovery in robotics with the t-resilience algorithm. *International Journal of Robotics Research*, 2013.
- [9] S. Koos, J.-B. Mouret, and S. Doncieux. The transferability approach: Crossing the reality gap in evolutionary robotics. *IEEE Transactions on Evolutionary Computation*, 17:122–145, 2013.
- [10] J.-B. Mouret and S. Doncieux. Sferes\_v2: Evolvin' in the Multi-Core World. In *Proc. of IEEE CEC*, pages 4079–4086, 2010.
- [11] J. Togelius, T. Schaul, D. Wierstra, C. Igel, and J. Schmidhuber. Ontogenetic and phylogenetic reinforcement learning. *Kuenstliche Intelligenz*, pages 30–33, 2009.
- [12] J. Yosinski, J. Clune, D. Hidalgo, S. Nguyen, J. Zagal, and H. Lipson. Evolving Robot Gaits in Hardware: the HyperNEAT Generative Encoding Vs. Parameter Optimization. *Proc. of ECAL*, pages 1–8, 2011.