Abstract of: Fast Damage Recovery in Robotics with the T-Resilience Algorithm

Sylvain Koos\(^1,2\), Antoine Cully\(^1,2\) and Jean-Baptiste Mouret\(^1,2\)
Sorbonne Universités, UPMC Univ Paris 06, UMR 722, ISIR, F-75005, Paris, France
CNRS, UMR 7222, ISIR, F-75005, Paris, France
mouret@isir.upmc.fr

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 and diagnosis procedures.

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 (Togelius et al., 2009), 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 (e.g. 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. (Hornby et al., 2005; Yosinski et al., 2011)). 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 a self-model for the robot, then evolving a controller with this simulation, Bongard et al. (Bongard et al., 2006) 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 undirected: the algorithm can “waste” a lot of time to improve parts of the self-model that are irrelevant for the task. Second, the diagnosis may be wrong, which leads to a useless contingency plan. Third, there is often a “reality gap” between a behavior learned in simulation and the same behavior on the target robot (Jakobi et al., 1995), but nothing is included in Bongard’s algorithm to prevent such gap to happen: the controller learned in the simulation may not work well on the real robot, even if the self-model is accurate.

Our algorithm is inspired by the “transferability approach” (Koos et al., 2013b), whose original purpose is to cross the “reality gap” that separates behaviors optimized in simulation to those observed on the target robot. 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 problem is transformed into a multi-objective optimization in which both performance in simulation and transferability are maximized. This optimization is performed with a multi-objective evolutionary algorithm (NSGA-II, Deb et al. (2002)).

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 Transferability-based 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 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

---

This paper is an extended abstract of Koos et al. (2013a).
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 transfers most of the adaptation time from real experiments to simulations of a self-model.

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 (Hoos and Stützle, 2005), policy gradient (Kohl and Stone, 2004) and Bongard’s algorithm (Bongard et al., 2006). 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 (Endres et al., 2012). 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).

Acknowledgements

This work has been funded by the ANR Creadapt project, (ANR-12-JS03-0009) and a DGA scholarship to A. Cully.

References

Bongard, J., Zykov, V., and Lipson, H. (2006). Resilient machines through continuous self-modeling. Science, 314(5802):1118–1121.

Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans. on Evol. Comp., 6(2):182–197.

Endres, F., Hess, J., Engelhard, N., Sturm, J., Cremer, D., and Burgard, W. (2012). An evaluation of the RGB-D SLAM system. In Proc. of ICRA.

Hoos, H. H. and Stützle, T. (2005). Stochastic local search: Foundations and applications. Morgan Kaufmann.

Hornby, G., Takamura, S., Yamamoto, T., and Fujita, M. (2005). Autonomous evolution of dynamic gaits with two quadruped robots. IEEE Trans. on Robotics, 21(3):402–410.

Jakobi, N., Husbands, P., and Harvey, I. (1995). Noise and the reality gap: The use of simulation in evolutionary robotics. Proc. of ECAL, pages 704–720.

Kohl, N. and Stone, P. (2004). Policy gradient reinforcement learning for fast quadrupedal locomotion. In Proc. of ICRA, volume 3, pages 2619–2624. IEEE.

Koos, S., Cully, A., and Mouret, J.-B. (2013a). Fast damage recovery in robotics with the t-resilience algorithm. The International Journal of Robotics Research, 32(14):1700–1723.

Koos, S., Mouret, J.-B., and Doncieux, S. (2013b). The transferability approach: Crossing the reality gap in evolutionary robotics. IEEE Trans. on Evol. Comp., 17:122–145.

Togelius, J., Schaul, T., Wierstra, D., Igel, C., and Schmidhuber, J. (2009). Ontogenetic and phylogenetic reinforcement learning. Künstliche Intelligenz, 23:30–33.

Yosinski, J., Clune, J., Hidalgo, D., Nguyen, S., Zagal, J., and Lipson, H. (2011). Evolving Robot Gaits in Hardware: the HyperNEAT Generative Encoding Vs. Parameter Optimization. Proc. of ECAL, pages 1–8.