Online Discovery of Locomotion Modes for Wheel-Legged Hybrid Robots: a Transferability-based Approach

S. KOOS* and J.-B. MOURET

ISIR, Université Pierre et Marie Curie-Paris 6, CNRS UMR 7222.
Paris, 75005, France
**E-mail: koos@isir.upmc.fr*
http://people.isir.upmc.fr/koos/

Wheel-legged hybrid robots promise to combine the efficiency of wheeled robots with the versatility of legged robots: they are able to roll on simple terrains, to dynamically adapt their posture and even to walk on uneven grounds. Although different locomotion modes of such robots have been studied, a pivotal question remains: how to automatically adapt the locomotion mode when the environment changes? We here propose that the robot autonomously discovers its locomotion mode using optimization-based learning. To that aim, we introduce a new algorithm that relies on a forward model and a stochastic multi-objective optimization. Three objectives are optimized: (1) the average displacement speed, (2) the expended energy and (3) the transferability score, which reflects how well the behavior of the robot is in agreement with the predictions of the forward model. This transferability function is approximated by conducting 20 experiments of one second on the real robot during the optimization. In the three investigated situations (flat ground, grass-like terrain, tunnel-like environment), our method found efficient controllers for forward locomotion in 1 to 2 minutes: the robot used its wheels on the flat ground, it walked on the grass-like terrain and moved with a lowered body in the tunnel-like environment.

Keywords: Hybrid robot; Learning; Multi-objective Evolutionary Algorithm; Transferability

1. Introduction

Wheel-legged hybrid robots aim at combining the efficiency of wheeled robots with the versatility of legged robots:[1–3] by adding wheels at the end of legs, they can act like wheeled robots on simple terrains and adapt their posture to the shape of an uneven ground; they can also stop their wheels and be equivalent to a classic legged robot. Several papers deal with the control of such wheel-legged robots[1–3] but most of them describe sin-

gle controllers that can take advantage of the legs to adapt the posture of the robot.[2,3] A few papers investigate different locomotion modes, such as rolling with passive wheels and walking.[1] Nevertheless, none of them tackles one of the most important questions: *how should the robot select its locomotion mode?* And, since there are an infinity of possible situations and an infinity of hybrid locomotion modes, *how to discover the best controller in an unforeseen situation?* The present paper introduces a new algorithm to answer these two questions. In the typical scenario (see Fig. 1), the robot first moves on a flat terrain, in which using the wheels is *a priori* efficient; the robot then encounters a tall grass field in which its wheels are not working anymore: it has to find a new locomotion mode; finally, it leaves the grass field and enters a tunnel with a low ceiling that should never be hit: a new adaptation is required. This scenario illustrates three situations, but the goal of the present paper is to introduce a general adaptation algorithm that could be used in any situation and for any robot.

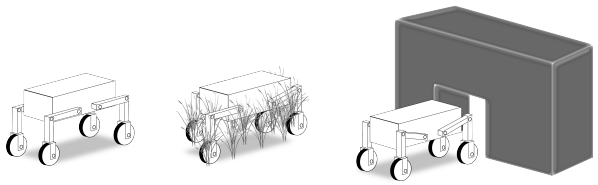

Fig. 1. Typical scenario made of three situations : (left) uniform flat ground, (center) grass-like terrain, (right) tunnel-like terrain.

2. Proposed Approach

2.1. *Overview*

Online adaptation of a mobile robot to its environment can be split in three different phases, as pictured on Fig. 2, left: (1) the robot detects that the environment has changed or that its current behavior is not efficient anymore; (2) the robot enters an adaptation phase, during which new efficient locomotion modes are looked for; (3) an efficient behavior is next selected and exploited, while no new situation is encountered. We here focus on the adaptation phase (phase 2).

To discover locomotion modes, the robot can *learn* a new strategy by itself. Such a situation is a typical use case of reinforcement learning[4] or optimization-based learning,[5] but these algorithms require a large number

of trials on the real robot, making the learning phase long[a] and potentially dangerous. The Estimation-Exploration Algorithm[6] proposes a faster alternative which relies on an automatically learned, internal dynamic model of the robot: using it, a star-shaped quadruped robot only needed to perform 15 simple actions to learn a new walking behavior after the loss of a leg. However, each time a disagreement is detected between the forward model and the reality, the EEA requires to learn a new forward model from scratch; we think that this complex step is inefficient if the disagreement stems from a change of the environment (most parts of the forward model should still be reliable) and not from a change in the morphology of the robot. Additionally, actions performed by the robot are not goal-directed: the robot can spend a long time to model a part of its morphology which is useless for its goal.

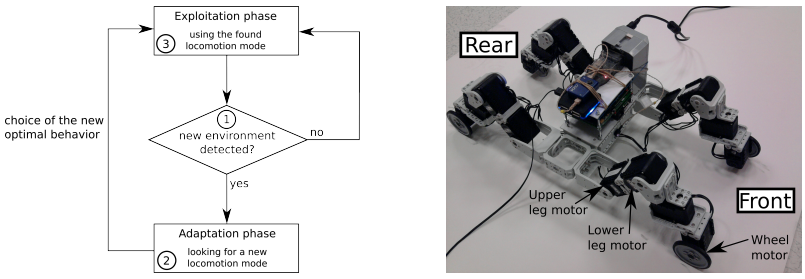

Fig. 2. Left, the three different phases of our adaptation algorithm. Right, the 12-DOF quadrupedal wheel-legged robot used for the experiment.

To overcome these limitations, we propose that the adaptation phase relies on a similar dynamic forward model, except that it will be provided by the robot's designer. In our approach, the learning algorithm does not modify this model but it discovers what potentially interesting behaviors are not properly working in reality in order to avoid them. We take inspiration from the transferability approach,[7] by looking for good compromises between performance objectives and the transferability measure, an approximate function that reflects how well the reality matches the prediction of the forward model for a given controller. This last function is learned by transfering a few well-chosen controllers on the real robot during the optimization and then by comparing the behaviors in simulation and in reality.

[a] about 3 hours to learn a quadruped locomotion pattern for the Aibo robot[4] and about 20 minutes for a snake-like robot.[5]

2.2. *Algorithm*

The several locomotion modes brought into play by hybrid wheel-legged robots can be interpreted as different compromises between performance and expended energy. For instance, walking behaviors are more versatile than rolling behaviors, but leads to higher energy consumption. We consequently formulate the learning problem as a multi-objective optimization process with four objectives to be optimized:

(1) average displacement speed of the robot;
(2) expended energy to perform the behavior[b];
(3) approximated transferability measure $\hat{T}$;
(4) behavioral diversity objective.

This last objective allows to maintain behavioral diversity among the population, which efficiently enhances exploration of the controller state space.[8,9] To quantify the diversity of a controller from the already transferred ones, we define a behavioral diversity value as follows. Let $\mathfrak{C}_T$ be the set of the already transferred controllers and b_{dist} a behavioral distance defined in simulation, the behavioral diversity value $diversity(c)$ for a given controller c is:

$$diversity(c) = \min_{c_i \in \mathfrak{C}_T} b_{dist}(c,\ c_i)$$

The behavioral distance function b_{dist} is based on a set of behavioral features defined by the user and that describe the behaviors of controllers in simulation. The distance between two controllers is then computed as the Euclidean distance between their vectors of behavioral features.

The relation that links the behavioral features in simulation and the exact transferability measure is interpolated Inverse Distance Weighting (IDW) by interpolating. An outline of the learning algorithm is pictured on Fig. 3:

A. For each individual x of the population, the four objective values are computed in simulation (average speed, expended energy, approximated transferability and behavioral diversity).
B. The controllers are optimized via a multi-objective evolutionary algorithm (MOEA) during N generations.

[b]Walking uses more energy than rolling

C. Every N generations, the controller of the current population with the highest behavioral diversity is transferred onto the real robot:
   C1. the behavior of the robot is logged;
   C2. the corresponding exact transferability value is computed based on the simulated and real behaviors;
   C3. the approximated transferability measure is updated with the new data by IDW interpolation; back to step A.

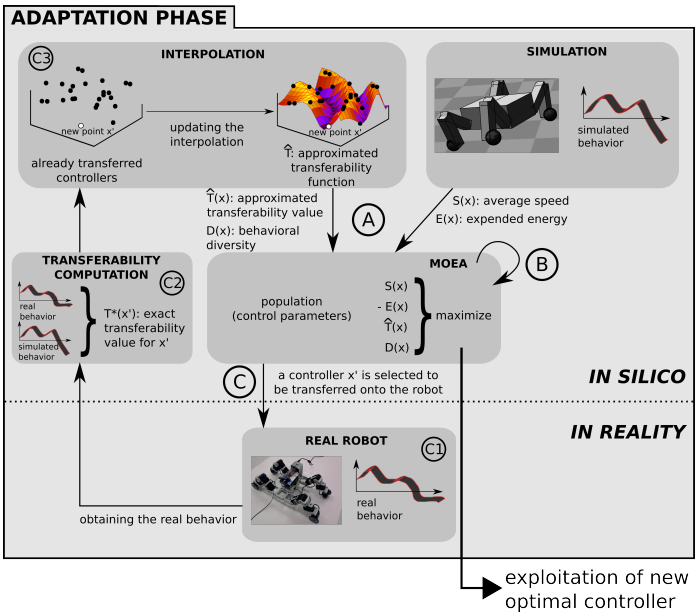

Fig. 3. Outline of the adaptation phase. The four objectives are computed for the controllers (step A), which are optimized during N generations with a multi-objective evolutionary algorithm (MOEA, step B). The controller with the highest behavioral diversity is then transferred onto the real robot and the generated data allows to update the approximated transferability function (steps C1 to C3).

3. Application

We tested this new approach on a 12-DOF quadrupedal wheel-legged hybrid robot (Fig. 2, right) inspired by the Hylos robot.[2] Each leg of the robot includes 4 Dynamixel AX-12+ Robot Actuators from Bioloid Kit. In the experiment, the two upper motors and the wheel motor of each leg are controlled. The control signals of all the motors depend on three optimized real

parameters $(p_1,\ p_2,\ p_3) \in [-1,1]^3$: p_1 is linked to the posture of the robot, p_2 to the amplitude of legs' movements and p_3 to the speed of wheels.The behavioral features b^i are directly derived from these parameters and their values are indicated for each case in the following equations. Concerning the leg motors, the desired angular position α^d of a motor i at time t is obtained by:

$$\alpha^d(i,t) = \frac{5\pi}{48} \cdot p_1 - \begin{cases} \frac{5\pi}{24} \cdot p_2 \cdot sin(2\pi t - \phi(i)) & \text{, if } p_2 > 0 \quad (b^1 = p_1,\ b^2 = p_2) \\ 0 & \text{, otherwise} \quad (b^1 = p_1,\ b^2 = 0) \end{cases}$$

The phase angle $\phi(i)$ is 0 for the upper leg motors of each leg and $\pi/2$ for the lower leg motors of each leg (for orientation, see Fig. 2, left). Both motors of one leg consequently have the same control signal with different phases. New angle positions are sent every 0.05 seconds.

The four wheel motors are controlled with the same speed value v defined as follows depending on the p_3 parameter ($v_{max} = 6$ rad/s):

$$v = \begin{cases} p_3 \cdot v_{max} & \text{, if } p_3 > 0 \quad (b^3 = p_3) \\ 0 & \text{, otherwise} \quad (b^3 = 0) \end{cases}$$

The average speed and the expended energy are computed in a simulation model based on Open Dynamics Engine (ODE)[c]. Whatever the real environment looks like, it always simulates the displacement of the robot on a flat ground during one second. Energy is crudely approximated by summing the angular movement of each degree of freedom. The displacement speed of the robot in reality is measured during one second with a motion tracking system but further work will rely on on-board visual odometry.[10] The exact transferability measure T^* is the opposite of the absolute variation between the average speeds measured with the forward model and on the robot.

The four objectives are simultaneously optimized with NSGA-II, a state-of-the art multi-objective evolutionary algorithm[11d]. This stochastic algorithm finds an approximate set of all Pareto-optimal trade-offs. The size of population is set to 40 and the optimization process stops after 200 generations. The approximate transferability measure is updated every ten generations (N = 10), by transferring the controller of the current population with the highest behavioral diversity, i.e. we perform 20 experiments

[c] `http://www.ode.org`

[d] This work has been implemented within the Sferes$_{v2}$ framework.[12] The source code is available at: `http://www.isir.fr/evorob_db`

by run[e]. At the end of the optimization, we extract from the final non-dominated set the solutions whose transferability values are higher than -1 cm/s. We then select the controller, which minimizes the distance to an ideal controller with an average speed of 18 cm/s and an energy value of 25.[7] This ideal controller corresponds to the most efficient rolling behavior on flat ground.

4. Results

We investigated a scenario made of three situations to which the robot has to adapt: (1) a flat ground without any constraint, (2) a grass-like environment in which wheels are blocked and, (3) a tunnel-like environment in which robot's movements are blocked if $p_1 < 0.7$. To obtain statistical results, we performed 5 experiments in each situation. Results (figure 4) show that: (1) the robot autonomously chose to use its wheels when it was put on a flat ground, (2) it learned to walk when its wheels were unavailable and, (3) it lowered its body when it encountered the tunnel. Each learning phase required 1 to 2 minutes with a recent multi-core computer[f] (including the tests on the real robot). Table 1 reports quantitative results: the rolling behavior found on flat ground cannot be used on grass (average speed is null) and is often useless in the tunnel (average speed is low: many controllers did not work), but it is the most energy-efficient controller; the walking mode requires more energy but it appears more versatile; the controllers optimized for the tunnel use slightly more energy than those optimized for flat ground only because the robot has to lower its body. *These measures show that adapting the locomotion mode was always useful and often mandatory for the robot to move forward.*

Table 1. Average speed for each controller and in each situation (5 trials), standard deviations are in brackets; energy is in arbitrary unit.

	flat ground (cm/s)	grass (cm/s)	tunnel (cm/s)	energy (E)
learned on flat ground	17.1 (0.3)	0.0 (0)	8.6 (9.9)	-26 (3)
learned in grass	13.1 (1.9)	13.1 (1.9)	8.3 (7.6)	-33 (1)
learned in tunnel	15.9 (1.7)	0.0 (0)	15.9 (1.7)	-27 (2)

[e]At generation 0, a random controller is transferred.
[f]Intel Xeon E5520 dual quad-core at 2.3 GHz

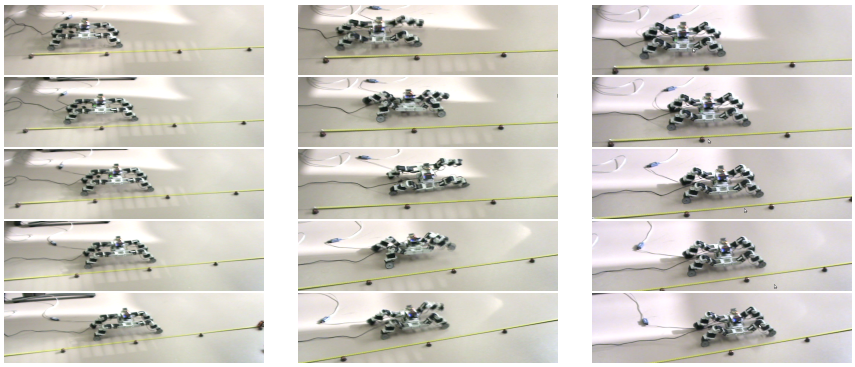

Fig. 4. Typical selected behaviors: (left) flat ground: rolling behavior; (center) grass-like: walking behavior; (right) tunnel: rolling behavior with a lowered body.

References

1. G. Endo and S. Hirose, Study on roller-walker (multi-mode steering control and self-contained locomotion), in *Proc. of IEEE-ICRA'00*, 2000.
2. C. Grand, F. Benamar, F. Plumet and P. Bidaud, *The International Journal of Robotics Research* **23** (2004).
3. P. Jarrault, C. Grand and P. Bidaud, Large Obstacle Clearance Using Kinematic Reconfigurability for a Rover with an Active Suspension, in *Proc. of CLAWAR'10*, 2010.
4. N. Kohl and P. Stone, Policy gradient reinforcement learning for fast quadrupedal locomotion, in *Proc. of IEEE-ICRA'2004*, 2004.
5. A. Sproewitz, R. Moeckel, J. Maye and A. J. Ijspeert, *The International Journal of Robotics Research* **27**, 423 (2008).
6. J. Bongard, V. Zykov and H. Lipson, *Science* **314**, 1118 (2006).
7. S. Koos, J.-B. Mouret and S. Doncieux, Crossing the Reality Gap in Evolutionary Robotics by Promoting Transferable Controllers, in *Proc. of GECCO'2010*, 2010.
8. S. Doncieux and J.-B. Mouret, Behavioral diversity measures for evolutionary robotics, in *Proc. of IEEE-CEC'10*, 2010.
9. J.-B. Mouret and S. Doncieux, *Evolutionary Computation* (2011, to appear).
10. D. Nistér, O. Naroditsky and J. Bergen, *Journal of Field Robotics* **23**, 3 (2006).
11. K. Deb, S. Agrawal, A. Pratab and T. Meyarivan, A Fast Elitist Non-Dominated Sorting Genetic Algorithm for Multi-Objective Optimization: NSGA-II, in *Proc. of PPSN VI*, 2000.
12. J.-B. Mouret and S. Doncieux, Sferes_v2: Evolvin' in the multicore world, in *Proc. of IEEE-CEC'10*, 2010.