Distributed Online Learning of Central Pattern Generators in Modular Robots

David Johan Christensen[1], Alexander Spröwitz[2], and Auke Jan Ijspeert[2]

[1] The Maersk Mc-Kinney Moller Institute, University of Southern Denmark
david@mmmi.sdu.dk
[2] Biorobotics Laboratory, Ecole Polytechnique Fédérale de Lausanne, Switzerland
{alexander.sproewitz,auke.ijspeert}@epfl.ch

Abstract. In this paper we study distributed online learning of locomotion gaits for modular robots. The learning is based on a stochastic approximation method, SPSA, which optimizes the parameters of coupled oscillators used to generate periodic actuation patterns. The strategy is implemented in a distributed fashion, based on a globally shared reward signal, but otherwise utilizing local communication only. In a physics-based simulation of modular Roombots robots we experiment with online learning of gaits and study the effects of: module failures, different robot morphologies, and rough terrains. The experiments demonstrate fast online learning, typically 5-30 min. for convergence to high performing gaits ($\approx$ 30 cm/sec), despite high numbers of open parameters (45-54). We conclude that the proposed approach is efficient, effective and a promising candidate for online learning on many other robotic platforms.

1 Introduction

Modular robots are made up from a number of interconnected robotic modules. Each module can communicate with neighbor modules, sense its local environment, and control its own actuators. By combining the modules in different configurations robots with different capabilities can be constructed. Since a robot's mobility is highly dependent on the details of its morphology, the flexibility of modular robot's morphology makes them an interesting platform for studying locomotion. However, control and adaptation of locomotion must be implemented in the context of the modular robot's distributed morphology. Further, since modular robots are polymorphic we desire a strategy which is not designed for a specific morphology. The strategy should rather optimize a variable number of control parameters, for a class of morphologies, while the robot is moving in its environment. In this paper we take a distributed control approach to tackle the problem: All modules have individual, identical, and autonomous controllers. Any module optimizes its own set of control parameters based on a global reward signal. The robot's locomotion pattern then emerges from the collective adaptations and behaviors of its modules. We hypothesize that such a distributed

strategy may be more robust and flexible since it may be independent to the specific robot's morphology and can adapt online to module failures or morphology changes. Ultimately, we anticipate that by studying such distributed strategies we may gain insights into how adaptive sensory-motor coordination can emerge and self-organize from billions of individual cells in biological organisms.

In this paper we study a distributed learning strategy for online optimization of locomotion gaits. We experiment with two quadruped robots constructed from Roombots modules. Each Roombots module has three actuators which we control using periodic actuation patterns generated by three local oscillators. Neighbor-to-neighbor communication between modules is used to synchronize the module's oscillators. These local connections make the oscillators form a central pattern generator (CPG) network covering the whole robot thereby enabling global synchronization. To enable life-long learning based on noisy fitness measurements we apply the model-less Simultaneous Perturbation Stochastic Approximation (SPSA) method. Each module optimizes its own local CPG parameter set based on a globally shared reward signal. Therefore, both the control and the learning are distributed without any centralized control necessary.

The rest of this paper is organized as follows: In Section 2 we summarize related work. In Section 3 we describe the methods which comprise the online learning strategy. The experimental platform and setup is described in Section 4. A number of experiments with simulated Roombots robots are presented in Section 5. The experiments demonstrate that the proposed strategy finds fitter gaits than random search optimization, works for different morphologies, can adapt to module failures, but converges to suboptimal gaits in rough terrains. We conclude in Section 6 that the proposed approach is efficient, effective and a good candidate for online learning of locomotion on many robotic platforms.

2 Related Work

Here, we review related work on evolutionary adaptation and online learning of modular robots for the task of locomotion. Karl Sims pioneered the field in the early 90's by co-evolving the morphology and control of simulated modular robots [10]. Later work succeeded in transferring similar co-evolved robots from simulation to hardware [6, 8]. An example of adaptation by evolution in modular robots was conducted by Kamimura et al., who evolved the coupling parameters of central pattern generators for straight line locomotion of modular M-TRAN robots [5]. By incorporating sensory entrainment in the optimization the authors were able to bridge the reality gap. Although appealing, one challenge with evolutionary approaches is that once transferred the robot is typically no longer able to adapt to major changes in the morphology or environment. To overcome this limitation optimization of locomotion gaits can be performed online. This was studied by Marbach and Ijspeert on the YaMoR modular robotic system [9]. Their strategy was based on Powell's method, which performed a localized search in the space of selected parameters of coupled oscillators. Parameters were manually extracted from the modular robot by exploiting symmetries. Follow-up

work by Spröwitz et al. demonstrated online optimization of 6 parameters on a physical robot in roughly 25-40 minutes [14]. We also try to realize simple, robust, fast, model-free, life-long learning on a modular robot. The main difference is that we seek to automate the controller design further in the sense that no parameters have to be extracted from symmetric properties of the robot. Further, our approach utilizes a form of distributed optimization. A similar approach was taken by Maes and Brooks who performed distributed learning of locomotion on a 6-legged robot [7]. The learning was distributed to the legs themselves. Our strategy is not dependent on the robot's specific morphology. Similarly, Bongard et al. demonstrated learning of locomotion and adaptation to changes in the configuration of a modular robot [1]. They took a self-modeling approach, where the robot developed a model of its own configuration by performing basic motor actions. In a physical simulator a model of the robot configuration was evolved to match the sampled sensor data (from accelerometers). By co-evolving the model with a locomotion gait, the robot could then learn to move with different morphologies. Our work presented here is similar in purpose but different in approach: The strategy is simple, model-less and computationally cheap to allow implementation on small embedded devices, such as modular robots. In previous work we studied distributed, morphology independent, online learning for ATRON and M-TRAN robots [2, 3]. This work was based on the same principles but the methods applied were different: instead of SPSA optimization we applied a simple reinforcement learning strategy and instead of coupled oscillators we applied discrete actions and gait-tables.

3 Methods

This section describes the methods for generating periodic actuation patterns for gait implementation and for online optimization of gait parameters. The methods are selected and combined into an online learning strategy with the following design goals in mind:

Morphology Independence: Since a modular robot can take on many different morphologies, the strategy should not be designed for any particular morphology but rather function on a class of different morphologies.

Life-long Learning: The morphology of a modular robot can change over time, either due to module failures, adding or removing of modules, or due to voluntary morphosis, therefore, the strategy must be able to continuously adapt while performing its function.

Noise Tolerance: The gaits must eventually be optimized directly on the physical robot. The interactions between the robot and its environment will be complex and in practice impossible to predict, therefore, the optimization strategy must be tolerant to noisy fitness measurements.

Simple Implementation: Modular robots are embedded devices with limited communication and computation abilities, thus, the strategy must require a minimal amount of resources and ideally be simple to implement on the distributed morphology that modular robots are.

3.1 Central Pattern Generators

Biological CPGs are special neural circuits found in vertebrates, able to produce a rhythmic signal without any external sensory input, where they for example control muscles during locomotion. We apply a CPG model for gait control because of their ability to generate periodic actuation patterns, ability to self-synchronize in a distributed system, open parameters which are appropriate for optimization, and finally since CPGs are biologically plausible. A review of CPGs and their use in robot control can be found in [4]. The specific CPG model we utilize is a Hopf oscillator in Cartesian space with diffusive coupling [15]. The advantages of this model include its simplicity, stable limit-cycle behavior, and explicit parameters for setting phase, amplitude and frequency. For an oscillator i the coupled differential equations are:

$$\dot{x_i} = \gamma(\mu - r_i^2)x_i - \bar{\omega}y_i \tag{1}$$

$$\dot{y_i} = \gamma(\mu - r_i^2)y_i + \bar{\omega}x_i \tag{2}$$

Where $r_i = \sqrt{x_i^2 + y_i^2}$ and the state variables are x and y. γ is a parameter that affects the speed of convergence towards the oscillators amplitude μ^2. $\bar{\omega}$ is the oscillator's frequency which is a function of a frequency parameter, ω, and is also affected by the sum of couplings to other oscillators. A coupling from oscillator i to oscillator j has a weight parameter, w_{ij}, and a desired phase difference ϕ_{ij}. Then the oscillator may be coupled to other oscillators using:

$$\bar{\omega} = \omega + \sum_{j=1}^{N} \frac{w_{ij}}{r_i} \left[(x_i y_j - x_j y_i)\cos\phi_{ij} - (x_i x_j + y_i y_j)\sin\phi_{ij}\right] \tag{3}$$

We use one oscillator to control the position of an actuator by using x_i as the control set-point for the actuator.

3.2 CPG Network Architecture

To enable a scalable, distributed, and morphology independent control strategy we design the network of CPG couplings so that it is equivalent on each module (homogeneous control). Each Roombots module is programmed with four oscillators: three which are used as set-points for its actuators (C_{m1}, C_{m2} and C_{m3}) and one which acts as a clock (C_c). The architecture is illustrated in Fig. 1. The robot is equipped with a distributed global clock, implemented as a network of in-phase clock oscillators, one per module. Each clock oscillator is coupled with the clock oscillators on its neighbor modules using local communication channels. This architecture is scalable since oscillators are only coupled neighbor-to-neighbor, so the computation/communication load is independent on the number of modules in the robot. Further, the architecture is distributed since the module controls itself based on its local state and local interactions. Finally, the architecture is morphology independent since when adding a new module to the robot new couplings can automatically be established using local communication. Therefore, the individual modules are not aware of the global module configuration.

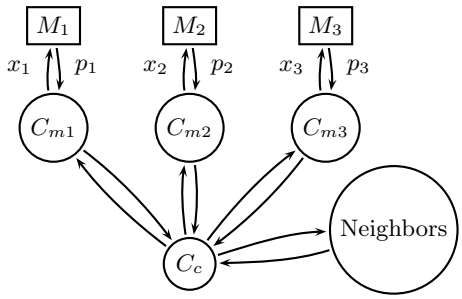


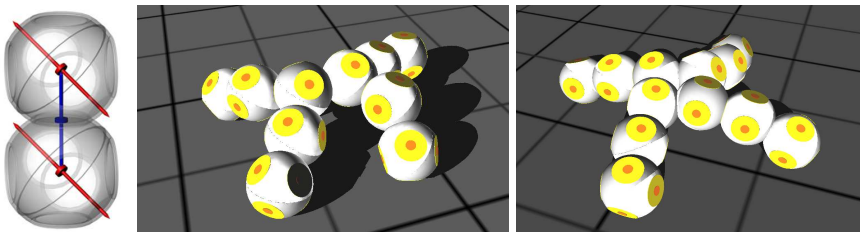
Fig. 1. The CPG network architecture of coupled oscillators within each Roombots. Three oscillators provide the set-points for the three servos (M_1 -M_3). A forth oscillator acts as a clock which is coupled in phase with clock oscillators on neighbor modules. The servo position, p_i, can be used as feedback in the oscillators although we do not use it here.

3.3 Learning Algorithm

For online optimization of CPG parameters we select the Simultaneous Perturbation Stochastic Approximation (SPSA) method by Spall [12]. This algorithm requires no explicit gradient and therefore no model of the robot. It is designed to build an approximation of the gradient from direct (generally noisy) measurements of the objective function. Further, SPSA only requires two measurements of the objective function per iteration (i.e. two robot trials with different controllers) independent on the number of adjustable parameters. Also, these measurements are made based on small perturbations of the same parameter set. Hence the robot's behavior only alters slightly while it is learning, unlike optimization based on population-based methods such as evolutionary algorithms. Finally, SPSA is simple to implement in a distributed fashion since each module may independently optimize its own parameters without knowledge of the other modules parameters or the need for any other coordination than simple synchronization of when the parameters are updated.

The SPSA method optimizes the parameter set $\hat{\theta}$ defined by the experimenter. In an iteration, k, it estimates the gradient, $g(\hat{\theta})$, based on two noisy measurements of the objective function $y(\hat{\theta})$:

$$\hat{g}_k(\hat{\theta}_k) = \frac{y(\hat{\theta}_k + c_k\Delta_k) - y(\hat{\theta}_k - c_k\Delta_k)}{2c_k} \begin{bmatrix} \Delta_{k1}^{-1} \\ \Delta_{k2}^{-1} \\ \vdots \\ \Delta_{kp}^{-1} \end{bmatrix} \tag{4}$$



(a) (b) Quadruped Type 1 (c) Quadruped Type 2

Fig. 2. (a) An illustration of the Roombots' three degrees of freedom. (b) A robot comprised of five Roombots modules. (c) A robot comprised of six Roombots modules.

Where c_k is a learning parameter and Δ_k is an vector of randomized ± 1. SPSA then updates $\hat{\theta}$ based on $\hat{g}_k(\hat{\theta}_k)$.

$$\Delta\hat{\theta}_k = -a_k \cdot \hat{g}_k(\hat{\theta}_k) \tag{5}$$

$$\hat{\theta}_{k+1} = \hat{\theta}_k + sign(\Delta\hat{\theta}_k) \cdot min(|\Delta\hat{\theta}_k|, \epsilon) \tag{6}$$

a_k is a learning parameter, we also added a max step-size, ϵ, to reduce the risk of instability.

4 Experimental Setup

The Roombots is a self-reconfigurable modular robot which is being developed at EPFL [13]. A Roombots module consists of two spherical parts, made up by four hemispheres in total, see Fig. 2(a). The hemispheres can actively be rotated relative to each other, thereby giving a Roombots module two "outer" and one "inner" actuated degree of freedom. The outer and inner hemispheres contain up to three active connectors respectively, which enable a module to connect to other modules. In this paper we experiment with the two different quadrupedal Roombots robots shown in Fig. 2. The Roombots modules are simulated in the commercial Webots robot simulator by Cyberbotics Ltd [16] which relies on Open Dynamic Engine (ODE) for simulation of collisions and rigid body dynamics [11]. The details of the Roombots model used are based on the current prototype of the Roombots as well as expected characteristics of the final design. The characteristics are kept fairly conservative but since module details will vary slightly compared to the final Roombots design, we cannot expect a perfect transfer to the physical modules once ready. The most important module parameters are: mass = 0.975 kg, actuation torque = 5 Nm, and a maximum rotational velocity = 2.62 rad/sec. Other environmental parameters include coefficients of friction and restitution, which are 1.0 and 0.5 respectively.

In the following experiments each module runs identical learning controllers and optimizes their behavior based on a single shared reward signal. For simplicity the reward is velocity computed as the distance traveled by the robot's center

(a) Coupled Oscillators

Symbol	Description	Value
ω	Frequency	0.8 Hz
γ	Amp. Contraction	1.0
θ_{cc}	Phase Difference	0.0
θ_{mc}	Phase Difference	2π - θ_{cm}
w_{cc}, w_{cm}	Coupling strength	5.0
w_{mc}	Coupling strength	1.5

(b) SPSA-based Learning

Symbol	Description	Value
c_k	Gain parameter	0.025 or 0.05
a_k	Gain parameter	0.00015
ϵ	Max Stepsize	5% of range
T	Time Step	4 sec.

Table 1. Fixed parameters.

Symbol	Description	Init. Val.	Range
μ^2	Amplitude	0.35π	$[0; 0.7\pi]$
θ_{cm}	Phase Difference	0.0	$[-\pi; \pi]$
x_{offset}	Offset	0.0	$[-0.2\pi; 0.2\pi]$

Table 2. Open parameters in the coupled oscillators.

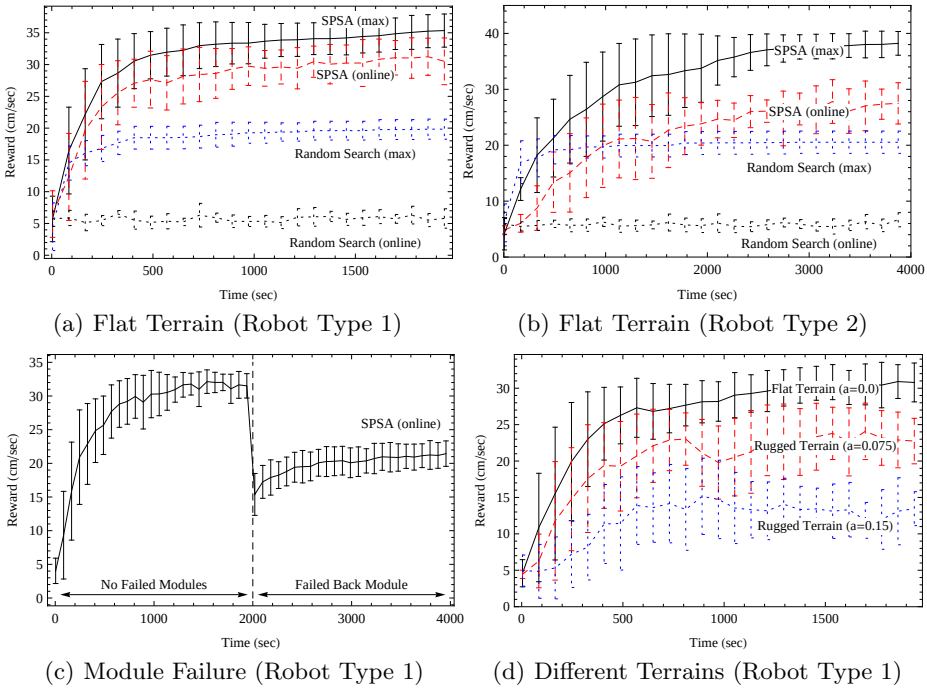
of mass in the xy-plane in a fixed length time duration: $y(\hat{\theta}) = \sqrt{\Delta x^2 + \Delta y^2}/T$. Each T seconds a single reward signal is sent to all the modules which corresponds to a measurement of either $y(\hat{\theta}_k + c_k\Delta_k)$ or $y(\hat{\theta}_k - c_k\Delta_k)$. After both measurements are performed the new $\hat{\theta}_{k+1}$ is computed. Fixed parameters for the SPSA-based learning and the CPG architecture are set as indicated in Table 1. The only parameter which is not the same for the two robots is the learning parameter c_k. For Type 1 $c_k = 0.025$ is appropriate, while it causes divergence in the learning for Type 2. Instead we set $c_k = 0.05$ for Type 2 at the cost of more gait variance during learning. The open parameters which must be optimized by the learning algorithm are shown in Table 2. For SPSA-based learning the open parameters, $\hat{\theta}$, are scaled between 0 and 1 and initialized to 0.5 (midpoint of the valid range). We found that random initialization often produces initial gaits too far from near optimal gaits, which causes the learning system to sometimes get caught in local optima. We plan to experiment with using random search optimization to find a good initial parameter set.

5 Experiments

In this section we present experiments with the proposed SPSA based strategy on simulated Roombots robots.

5.1 Morphology Independent Learning

To study the effects of different morphologies we performed experiments with SPSA-based learning and random search optimization on the Type 1 and Type 2 Quadruped robots. The average result of 10 trials with Type 1 is shown in Fig. 3(a). We observe that both the SPSA-based strategy and random search converge after approximately 10 minutes of trial and error behavior by the robot. This fast convergence gives strength to the claim that the learning could realistically be



(a) Flat Terrain (Robot Type 1)

(b) Flat Terrain (Robot Type 2)

(c) Module Failure (Robot Type 1)

(d) Different Terrains (Robot Type 1)

Fig. 3. The graphs show the average velocity of the robots as it improves over time. (a) and (b) SPSA-based learning compared to random search optimization. "Max" graphs indicate the average of the highest velocity measured so far in the trial. (c) Adaptation after back module failure. (d) Adaptation in three different terrains. All graphs are the average of 10 independent trials, bars indicate standard deviation.

utilized for life-long adaptation on the physical robot. Further, we observe that the online average velocity (30.7 cm/sec), measured from iteration 400 to 500, using the SPSA-based strategy is significantly higher than the maximum gait velocities (20.4 cm/sec) found by random search optimization ($P = 4.88 * 10^{-7}$). This result indicates that the parameter space is too large for random search to find the same solutions within the time given, further it also indicates the existence of gradients in the objective function that the SPSA-based strategy can exploit. For comparison we performed the equivalent experiment, using the same controller with the Type 2 robot. The results are shown in Fig. 3(b). Compared to learning with Type 1 robot we observe that for Type 2 the SPSA-based strategy converges slower (approx. 30 minutes) but still manages to finds high performing gaits. Also here we find that the SPSA-based strategy converges to significantly better gaits (27.6 cm/sec) than those found with random search (20.5 cm/sec) ($P=1.64 * 10^{-6}$).

By inspecting the solutions found by SPSA for the two robots we observe that all found gaits are similar to a trot, where the legs move together in diagonal

pairs. Although the solutions found with random search optimization showed greater variability all the gaits were still some variation of a fundamental trot. These observations indicate that a trot gait is a strong attractor point in the objective function for these particular combinations of robots, environment, and parameterized CPG controller.

In summary the SPSA-based control strategy allows us to perform online learning with two different robot morphologies without changing any part of the strategy. Except that we found it necessary to change the learning parameter, c_k. In future work we will try to remove this limitation by using adaptive learning rates to increase the strategy's morphology independence.

5.2 Adaptation after Morphosis and in Rugged Terrain

To study the effects of involuntary morphosis, i.e. morphological change, we performed experiments with module failures using the Type 1 robot. Initially the five modules comprising the robot are fully functional. After 500 iterations (2000 seconds), a module fails by locking its three actuators in their initial position defined by the starting pose of the robot (the CPG couplings stays intact). We then observe if the robot is able to adapt to this change in morphology by letting the robot learn for additional 500 iterations without resetting the learning parameters or any other part of the control system. For comparison we also performed the equivalent experiments with the learning disabled after module failure. In this case the robot does no longer adapt but keeps performing the same gait as just before the module failure. We performed two experiments: (1) In the first experiment a leg module fails. The module failure event is followed by a minor drop in velocity (from 31.4 cm/sec to 27.6 cm/sec), with no clear later improvement. The results of the equivalent experiment with no adaptation after module failure yields an average velocity of 27.1 cm/sec after module failure. Consistently we find that there is no significant difference in the average velocity between adaptation/no-adaptation after module failure ($P = 0.33$). So adaptation after morphosis seems not important in the case of a failing leg module for this robot structure. (2) In the second experiment the back module fails. In this case the event is followed by a major drop in average velocity, which seems to gradually improve after the event, see Fig. 3(c). The average velocity in the time interval 3000 sec to 4000 sec is 20.9 cm/sec with adaptation and 13.2 cm/sec without adaptation. Statistical analysis confirms that there is a significant difference between adaptation/no-adaptation in this case ($P = 0.00063$). So unlike the case of a failed leg module, in the case of failed back module life-long adaptation is important.

To study the effects of environment parameters on the learning we perform experiments with SPSA-based learning using a Type 1 robot in simulated rugged terrains. In the xy-plane the height of the terrain is defined by: $height(x, y) = a \cdot \cos(b \cdot y) \cdot \sin(b \cdot x)$. We set the parameters to: $b = 2.5$ meters and a to 0, 0.075 and 0.15 meters to create a terrain which vary from completely flat to a hilly terrain with relatively steep slopes and deep valleys. Fig. 3(d) shows the result of learning in these three different terrains. By visually inspecting the found gaits

we observe that in all cases the learning finds gaits which are able to move in the given terrain. As for flat terrain, in rough terrain the found gaits were also trot-like, however, the stride length was generally shorter. In addition, we found two effects on learning in increasingly rough terrain: (1) The first effect is a decrease in the final average velocity: 32.8 cm/sec ($\sigma = 2.59$), 23.2 cm/sec ($\sigma = 3.67$), 15.1 cm/sec ($\sigma = 2.31$) for $a = 0.0$, $a = 0.075$ and $a = 0.150$ respectively (measured by reevaluating the final gaits without adaptation). This effect is not surprising since we expect the robot to move slower in rougher terrain. (2) The second effect is a decrease in the ability to learn near optimal gaits. We observe this by reevaluating the gaits found in flat terrain ($a = 0.0$) in the two non-flat terrains ($a = 0.150$ and $a = 0.075$). Because the gaits are optimized in a different environment we would expect to see a decrease in performance compared to those optimized for the environment. However, instead we observe a drastic relative increase in performance: 30.6 cm/sec ($\sigma = 2.51$) and 24.7 cm/sec ($\sigma = 4.36$) for $a = 0.075$ and $a = 0.150$ respectively. This result indicates that the SPSA-based learning do not find near optimal gaits in the two terrains. The reason for the second effect is likely due to increased noise in the objective measurement (beyond the limits of SPSA's noise tolerance). We have observed a drastic drop in the average signal to noise ratio for the found gaits from $\overline{SNR} = \overline{y(\hat{\theta})}/\sigma = 23.1$, $\overline{SNR} = 10.5$ to $\overline{SNR} = 4.00$ for the three terrains respectively.

6 Conclusion

This paper reported on experiments using a distributed strategy based on the SPSA method for online optimization of a CPG network controlling the locomotion of modular robots. This online learning and control strategy was designed to be independent to the particular robot morphology, simple to implement in a distributed system, and to enable life-long adaptation based on a noisy reward signal. The strategy was evaluated in simulations of different quadrupedal Roombots robots. First, we found that the proposed strategy was appropriate for life-long learning since it could maintain a high performance during learning. Seconds, we found that the strategy could reliably optimize gaits with a considerably higher velocity than those found by random search. A near optimal gait (≈ 30 cm/sec) was typically found in 5-30 minutes. Third, we also found that the strategy enabled the robot to readapt its gait after involuntary morphosis (failed back module). Finally, we found that rough terrains decrease the strategy's effectiveness considerably since it drastically increased the amount of noise in the measured objective function. In conclusion the proposed strategy is efficient and effective on the Roombots robots and is a promising candidate for life-long online learning on many other robotic platforms. However, further work is required to integrate the strategy with appropriate sensor feedback to modulate or change between gaits while learning. In addition, we plan to study the strategy's ability to online co-optimize gait and morphological parameters for a broader class of robot morphologies.

7 Acknowledgments

This work was performed as part of the "Locomorph" project funded by the EU's Seventh Framework Programme (Future Emerging Technologies, Embodied Intelligence) and as part of the "Assemble and Animate" project funded by the Danish Council for Independent Research (Technology and Production Sciences).

References

1. J. Bongard, V. Zykov, and H. Lipson. Resilient machines through continuous self-modeling. *Science*, 314(5802):1118–1121, 2006.
2. D. J. Christensen, M. Bordignon, U. P. Schultz, D. Shaikh, and K. Stoy. Morphology independent learning in modular robots. In *Proceedings of International Symposium on Distributed Autonomous Robotic Systems 8 (DARS 2008)*, pages 379–391, 2008.
3. D. J. Christensen, U. P. Schultz, and K. Stoy. A distributed strategy for gait adaptation in modular robots. In *Proceedings of the IEEE Int. Conference on Robotics and Automation (ICRA)*, 2010.
4. A. J. Ijspeert. Central pattern generators for locomotion control in animals and robots: a review. *Neural Networks*, 21(4):642–653, 2008.
5. A. Kamimura, H. Kurokawa, E. Yoshida, S. Murata, K. Tomita, and S. Kokaji. Automatic locomotion design and experiments for a modular robotic system. *IEEE/ASME Transactions on Mechatronics*, 10(3):314–325, June 2005.
6. H. Lipson and J. B. Pollack. Automatic design and manufacture of robotic lifeforms. *Nature*, 406:974–978, 2000.
7. P. Maes and R. A. Brooks. Learning to coordinate behaviors. In *National Conference on Artificial Intelligence*, pages 796–802, 1990.
8. D. Marbach and A. J. Ijspeert. Co-evolution of configuration and control for homogenous modular robots. In *Proc., 8th Int. Conf. on Intelligent Autonomous Systems*, pages 712–719, Amsterdam, Holland, 2004.
9. D. Marbach and A. J. Ijspeert. Online Optimization of Modular Robot Locomotion. In *Proceedings of the IEEE Int. Conference on Mechatronics and Automation (ICMA 2005)*, pages 248–253, 2005.
10. K. Sims. Evolving 3d morphology and behavior by competition. In R. Brooks and P. Maes, editors, *Proc., Artificial Life IV*, pages 28–39. MIT Press, 1994.
11. R. Smith. Open dynamics engine. www.ode.org, 2005.
12. J. C. Spall. Multivariate stochastic approximation using a simultaneous perturbation gradient approximation. *IEEE Transactions on Automatic Control*, 37(3):332–341, Mar 1992.
13. A. Sproewitz, A. Billard, P. Dillenbourg, and A. J. Ijspeert. Roombots-mechanical design of self-reconfiguring modular robots for adaptive furniture. In *International Conference on Robotics and Automation (ICRA 2009)*, Kobe, Japan, May 2009.
14. A. Sproewitz, R. Moeckel, J. Maye, and A. J. Ijspeert. Learning to move in modular robots using central pattern generators and online optimization. *Int. J. Rob. Res.*, 27(3-4):423–443, 2008.
15. J. van den Kieboom. Biped locomotion and stability a practical approach. Master's thesis, University of Groningen, The Netherlands, 2009.
16. Webots. http://www.cyberbotics.com. Commercial Mobile Robot Simulation Software.