Evolving Systems manuscript No.
(will be inserted by the editor)

Fault-Tolerant Gait Learning and Morphology Optimization of a Polymorphic Walking Robot

David Johan Christensen · Jørgen Christian Larsen · Kasper Stoy

Received: 1 February 2013 / Accepted: 4 June 2013

Abstract This paper presents experiments with a morphology-independent, life-long strategy for online learning of locomotion gaits. The experimental platform is a quadruped robot assembled from the LocoKit modular robotic construction kit. The learning strategy applies a stochastic optimization algorithm to optimize eight open parameters of a central pattern generator based gait implementation. We observe that the strategy converges in roughly ten minutes to gaits of similar or higher velocity than a manually designed gait and that the strategy readapts in the event of failed actuators. We also optimize offline the reachable space of a foot based on a reference design but finds that the reality gap hardens the successfully transference to the physical robot. To address this limitation, in future work we plan to study co-learning of morphological and control parameters directly on physical robots.

Keywords Online Learning · Locomotion · Modular Robots · Reconfigurable Robots · Fault-Tolerance · Central Pattern Generators · Morphology Optimization

1 Introduction

Biologically inspired, embodied robots are attracting increasing attention for their potential properties such as adaptivity, robustness, versatility, and agility (Pfeifer et al 2007). The ability of biological organisms to adapt to dynamical changes in the environment (e.g. terrain) and morphology (e.g. growing or damaged body) inspired the work described in this paper. Our long-term vision is to design robots that like biological organisms are able to perform life-long self-adaptation of both their control and morphology to such changing conditions.

Modular self-reconfigurable robots are a type of system that in principle is able to self-adapt both control and morphology. Such robotic systems are polymorphic in the sense that by assembling the modules in different configurations, robots with different morphologies and thereby capabilities can be constructed. A robot's mobility is highly dependent on the details of its morphology. Therefore, the morphological flexibility of modular robots makes them an attractive platform for studying robot locomotion.

We must design locomotion control with careful attention to the interdependence with the robot's morphology and environment. However, since modular robots are polymorphic, we do not want to design a control strategy limited to a specific morphology. The strategy should rather be adaptive to enable optimization of a variable number of control parameters, for a class of morphologies. 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 should ideally be life-long to enable adaptation to changes in morphology or environment during the lifetime of the robot.

In this paper, we address the challenge of interdependence between environment, morphology, and locomotion control by proposing an online learning strategy. To fit the characteristics of modular robots, the

D. J. Christensen
Department of Electrical Engineering
Technical University of Denmark
Kgs. Lyngby, Denmark
E-mail: djchr@elektro.dtu.dk

J. C. Larsen · K. Stoy
The Maersk Mc-Kinney Moller Institute
University of Southern Denmark
Odense, Denmark
E-mail: {jcla, kaspers}@mmmi.sdu.dk

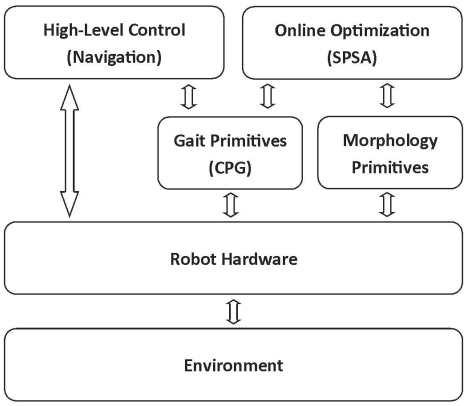


Fig. 1 Illustration of a robot control framework with online optimization of morphology and control. The focus of this paper is online optimization of gaits and offline optimization of morphology.

strategy is morphology-independent and life-long. The strategy controls each actuator based on periodic actuation patterns generated by an oscillator. A network of coupled oscillators that form an entrained central pattern generator (CPG) controls the robot. To enable life-long learning based on noisy fitness measurements we apply the model-less Simultaneous Perturbation Stochastic Approximation (SPSA) method by Spall (1992). We optimize the phase-shift of each oscillator based on the robot's measured velocity. Eventually we envision a control framework (see Fig. 1) that based on the same methods can optimize not only control parameters but also online optimize physical parameters of the robot. In other words, we envision a robot able to co-learn morphology and control.

1.1 Summary of Contributions

The main contributions of this paper are as follows:

- We propose a life-long online learning strategy that optimizes CPG parameters based on a stochastic optimization strategy (SPSA).
- We experimentally demonstrate the strategy on an 8-DOF quadruped LocoKit robot that converge on average within 10 min.
- We present proof-of-concept experiments where the strategy adapts to failed actuators.
- We demonstrate how SPSA can be used to offline optimize morphological parameters of the leg design.
- We transfer the offline-optimized legs to the real robot, but our attempts are unsuccessful which leads us to hypothesize that online optimization of morphology is a more viable approach.

This paper is an extended version of a paper that was presented at the 2012 IEEE Conference on Evolving and Adaptive Intelligent Systems (Christensen et al 2012) and includes previously unreported experiments on morphological optimization and learning (Sec. 6.3 and 6.4). This paper builds on previous work where we studied the proposed strategy on simulated Roombot robots, which is very different in terms of degrees of freedom and modularity (Christensen et al 2010b).

1.2 Outline

In Sec. 2, we provide an overview of related work with focus on adaptive locomotion of modular robots. In Sec. 3, we describe the LocoKit and the design of a quadruped utilized for experiments. The proposed control and learning strategy is described in Sec. 4. We describe the experimental setup with the robot on a boom and establishment of learning parameters in Sec. 5. Experiments on online learning, adaptation to failures, offline optimization of leg design, and online learning with such optimized designs are described in Sec. 6. We discuss the results and suggest future work in Sec. 7 and give conclusions in Sec. 8.

2 Related Work

2.1 Reconfigurable Modular Robots

Modular robots are comprised of robotic modules that a user can assemble in numerous configurations to construct various robot morphologies depending on the task scenario (Fukuda and Nakagawa 1988). Homogeneous reconfigurable modular robots are systems where all the modules have the same combination of mechanics and electronics. Alternatively, heterogeneous modular robots contain several types of modules with different functionality. The degree to which the modules are self-contained can vary from autonomous mobile modules (Groß et al 2006) to systems which combines passive and active components (such as LEGO Mindstorms). Heterogeneous systems (Duff et al 2001; Yim et al 2007b; Stoy et al 2007) includes the LocoKit (Larsen et al 2012) which we utilize in this paper. More details on the history and mechatronics of modular robots can be found in recent books and surveys (Yim et al 2007a; Stoy et al 2010; Murata and Kurokawa 2012).

2.2 Optimization of Locomotion Gaits

Evolutionary algorithms are a popular way to optimize locomotion gaits for modular robots. In the early 90's,

Sims (1994) pioneered the field by co-evolving the morphology and control of simulated modular robots. Later work succeeded in transferring similar co-evolved robots from simulation to hardware (Lipson and Pollack 2000; Marbach and Ijspeert 2004). An example of adaptation by evolution in modular robots was conducted by Kamimura et al (2005), who evolved the coupling parameters of central pattern generators for straight-line locomotion of modular M-TRAN robots. The authors were able to bridge the reality gap by incorporating sensory entrainment in the optimization.

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. Also, the reality gap between the simulated environment, where the robot evolved, and the real robot environment, where it should function, is often so far apart that the evolved robot cannot successfully be transferred to reality (Brooks 1992; Mataric and Cliff 1996). One approach to overcome such limitations is to use a minimal simulation as argued by Jakobi (1998). Minimal simulators only model the relevant robot-environment interactions and use randomness to model all other interactions. In summary, offline optimization based on evolutionary algorithms introduces a set of difficult challenges that require significant ingenuity to solve.

Alternatively, we can optimize locomotion gaits online. This was studied on the YaMoR modular robotic system Marbach and Ijspeert (2005). Their strategy utilized Powell's method, which performed a localized search in the space of selected parameters of coupled oscillators. To reduce the search space the authors manually extracted parameters from the modular robot by exploiting symmetries. Follow-up work by Sproewitz et al (2008) demonstrated online optimization of six parameters on a physical robot in roughly 25-40 minutes. We also try to realize simple, robust, fast, model-less, life-long learning on a modular robot. The main difference is that we seek to automate the controller design further in the sense that we do not have to reduce the search space, e.g. by extracting parameters from the bilateral symmetry of the robot.

2.3 Distributed and Morphology Independent Control

Most related work performs control and optimization in a centralized fashion. However, our approach utilizes a control and optimization strategy that is also appropriate for a distributed implementation. Maes and Brooks (1990) took a similar distributed learning approach to locomotion on a 6-legged robot. The legs themselves were learning distributed. The potential advantages of a distributed strategy include inherent morphology independence and fault tolerance.

Our strategy is not dependent on the robot's specific morphology. Similarly, Bongard et al (2006) demonstrated learning of locomotion and adaptation to changes in the configuration of a modular robot. 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, they evolved a model of the robot configuration 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 typical modular robots.

2.4 Fault-tolerant Control

This paper includes experiment on adaptation after failures of the robots actuators. It is an attractive possibility to realize fault-tolerance and self-repair by taking advantage of modular robot's redundancy and ability to adapt and self-reconfigure. Previous work has demonstrated fault-tolerance and self-repair on modular robots engaged in locomotion and self-reconfiguration (Yoshida et al 1999; Fitch et al 2000; Stoy and Nagpal 2004; Bongard et al 2006; Christensen et al 2008a). For example, in a paper by Mahdavi and Bentley (2003) the control of a snake like robot was optimized online using a genetic algorithm. The genetic algorithm recovered from failures in the SMAs actuating the robot.

2.5 Locomotion Control

This paper utilizes a model of central pattern generators (CPGs) to generate actuation patterns for locomotion. Related work has often applied CPGs to control locomotion of snakes and legged robots, e.g. (Taga et al 1991; Kimura et al 1999; Crespi and Ijspeert 2006; Righetti and Ijspeert 2006). The advantages of CPGs include: stable limit cycle behavior, appropriate for distributed implementation, few control parameters, suited to integrate sensory feedback signals, and offers a good substrate for learning and optimization algorithms (Ijspeert 2008). For example, Pouya et al (2010) used particle swarm optimization to optimize CPG gaits for Roombot robots that both contained rotating and oscillating actuators. Genetic regulatory network is an alternative biologically inspired method that previously has

been used to control modular robot locomotion (Bongard 2002; Zahadat et al 2010, 2012).

2.6 Optimization Strategy

The choice of learning strategy critically affects the performance of the system. A paper by Kohl and Stone (2004) presented an experimental comparison of different algorithms for online optimization of locomotion gaits for the AIBO robot. They compared four machine learning algorithms and found that simpler algorithms (hill climbing and policy gradient) performed better on the problem than the more complex algorithms (amoeba and genetic algorithm). A recent survey by Silva and Machado (2012) provides additional details about optimization of mechanical structure and locomotion gaits for legged locomotion. In this paper we utilize a stochastic optimization algorithm (SPSA by Spall (1992)) to optimize the parameters of the central pattern generators and the kinematics of the robot's leg design. In previous work we used distributed reinforcement learning (DRL) for morphology independent learning of discrete actions and gait-tables to control locomotion of modular robots (Christensen et al 2008a, 2010a). The main advantage of SPSA over DRL is that it allows optimization in a continuous space, which is appropriate for central pattern generators.

3 LocoKit - A Robotic Building Kit

The robot, used for experiments in this paper, is built from the polymorphic modular robotic building kit called "LocoKit" (Larsen et al 2010, 2012). The objective of LocoKit is to realize a flexible user reconfigurable modular robotic kit, which is lightweight, affordable and enables quickly realize energy efficient and robust legged robots. The LocoKit aims at realizing these design goals through a layered heterogeneous structure, with the following layers:

- Skeleton layer - low weight mechanical components
- Actuation layer - currently electrical servos
- Electronics layer - sensing, power, and computation

The LocoKit separates the mechanics, actuation, and electronics in simple reconfigurable modular components to increase flexibility and reduce the weight and complexity of individual modules.

The LocoKit components include glass fiber-reinforced plastic rods that connect everything in the system, and connection components made of 3D printed acryl. For actuation, the system uses the Dynamixel RX-10 servo. Note that the prototype of the LocoKit used in this paper is a relative early prototype and the details of the kit has changed significantly in the newest version (Larsen et al 2012).

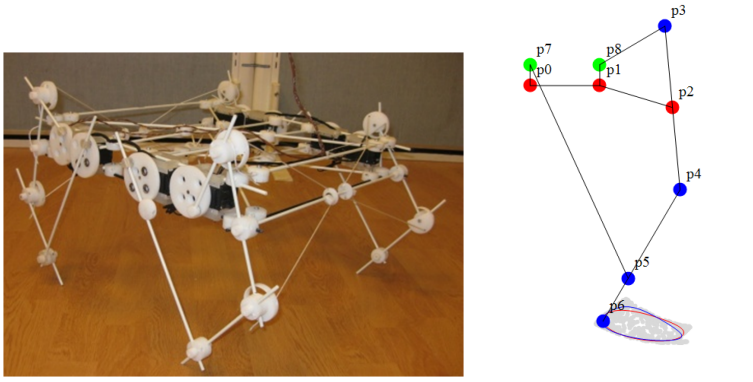
Based on the LocoKit components we constructed a quadruped robot as our experimental platform, see Fig. 2(a). We based the leg design on a 4-bar linkage, which we actuate using two actuators that are able to rotate infinite. Fig. 2(b) illustrate the reachable space of the foot. This design is inspired by Theo Jansen's StrandBeest (Jansen 2007). It is our long-term design goal with the LocoKit to enable the design of energy efficient robots. This affects several of our design choices: i) we place the actuators in the body of the robot to keep the weight (and thereby momentum) of the legs to a minimum. ii) A typical gait will be generated by continuous phase-shifted rotations of the robots eight actuators - by not using oscillatory actuation we minimize the accelerations (and thereby energy consumption) of the actuators. In this paper, for simplicity and reliability we control the servos from a central PC and with external power. However, the system also includes onboard electronics boards for power, processing and communication, and motor control.

4 A Strategy for Life-long Online Learning

This section describes an adaptive locomotion strategy based on CPGs for generating periodic actuation patterns for gait implementation and SPSA for online optimization of gait parameters.

4.1 Central Pattern Generator and Network Architecture

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. More details about CPGs and their use in robot control can be found by consulting the survey by Ijspeert (2008). The specific CPG model we utilize is a Hopf oscillator in Cartesian space with diffusive coupling by van den Kieboom (2009). 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



(a) (b) (c)

Fig. 2 (a) A quadruped robot with eight degrees of freedom constructed from the polymorphic robotic LocoKit. (b) The reachable space (shaded area) and two example foot trajectories are shown for the 2 DOF leg. In the kinematic model p0, p1 and p2 are kept stationary, p7 and p8 (disks) rotate around p0 and p1 (actuators) respectively in order to move p3 - p5 (joints) and p6 (foot). (c) The CPG network architecture of coupled oscillators that controls the LocoKit robot. The network consists of eight motors controlled by oscillators coupled to a single central oscillator.

differential equations are:

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

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

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 couple to other oscillators using:

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

The angular position of an actuator can be controlled from software and Θ is the target angular position (setpoint). To make an actuator oscillate with a given frequency, phase-shift and amplitude we set its setpoint to $\Theta = x_i$. For the LocoKit robot, to make the actuator rotate continuously with a given frequency and phase-shift we set it to $\Theta = \arctan(x_i/y_i)$ and select the appropriate quadrant.

The LocoKit robot is programmed with nine coupled oscillators: eight which are used as setpoints for its actuators (C_{m1}, C_{m2} ... C_{m8}) and one which acts as a central clock (C_c). Fig. 2(c) illustrate the architecture. The centralized architecture can easily be made distributed without significantly affecting the system performance, as we did in previous work (Christensen et al 2010b).

4.2 Learning Algorithm

For online optimization of CPG parameters we apply the Simultaneous Perturbation Stochastic Approximation (SPSA) method by Spall (1992).

No single optimization strategy will be universally good for all problems (Wolpert and Macready 1997). Our motivation for selecting the SPSA algorithm is that it has several attractive characteristics for online learning on modular robots: (i) Morphology-Independence - The SPSA algorithm requires no explicit gradient and therefore no model of the robot. SPSA build an approximation of the gradient from direct, generally noisy, measurements of the objective function. (ii) Scales to high-dimensional problems - SPSA only requires two measurements of the objective function per iteration (i.e. two robot trials with slightly different controllers) independent on the number of adjustable parameters. (iii) Local life-long learning - SPSA use small perturbations of the same parameter set to make the measurements of the objective function. Hence the robot's behavior only alters slightly while it is gradually learning (unlike global optimization methods such as GA and PSO (Kennedy and Eberhart 1995)). (iv) Distributed learning - SPSA is simple to implement in a distributed fashion since each module may independently optimize its own parameters (unlike GA and PSO). The modules only need a simple coordination of when to update the parameters based on shared measurements of the objective function.

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

$y(\hat{\theta})$:

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

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

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

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

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

5 Experimental Setup

5.1 Physical Setup

In the process of learning how to walk, the robot will need the freedom to try out numerous different gaits, while observing its locomotion speed. For our experiments, we mount the robot on a boom, which gives us the advantage of being able to run experiments for a longer time without human interaction. The boom provides threaded power to the robot and removes 665 grams of weight of the robot (i.e. 41 percent) by using counterweights, and thereby minimizing the risk of the robot breaking itself during the experiments. Clearly, this lift together with the momentum of the boom affects the dynamics of the robot. We do however accept this source of error, since the purpose of the experiments is to validate the learning strategy on a physical robot, not to find efficient gaits for the particular quadruped operating without a boom. The boom has a radius of 1.5 m and an encoder measures the position of the robot with resolution of 0.5 cm/degree in the end of the arm where the robot is mounted. To fasten the robot onto the boom, we place a universal joint between the robot and the boom. This enables the robot to move in the roll and pitch angle while still being fixed in the yaw angle. The robot can move in the up/down and forward/backward direction but not sideway.

5.2 Control Parameters

We must establish several parameters before we can perform experimental trials with the robot. The reward signal maximized by the learning system, $y(\hat{\theta})$, is a measurement of the velocity. We estimate the velocity as the distance moved by the robot in five steps:

$$v = |p_{t+T} - p_t|/T \quad (7)$$

This corresponds to a time of $T = 5 \cdot 1/\omega = 7.14$ sec, where the actuators are set to rotate with a frequency of $\omega = 0.7$ Hz. The oscillators are tightly coupled with weight parameters, w_{ij}, as described in previous work (Christensen et al 2010b). Both learning parameters, c_k and a_k, are set at fixed values to enable life-long learning. Generally, c_k can be set based on about how much the control parameters should be changed to cause an measurable effect on the objective function. For our purpose we set $c_k = 0.025$ which corresponds to a variation in the phase-shifts of ± 9.0 degrees while learning. Similarly, a_k can be set based on how much a control parameter should be changed given a typical measured velocity difference. We set $a_k = 0.0015$ which corresponds to a phase shift change of 10.8 degrees at a typical 1 cm/sec velocity difference. These learning parameters are set quite high to achieve fast convergence, potentially at the cost of convergence to a local optima or divergence.

The learning strategy optimizes the eight phase-shifts for the eight actuators from an initial gait that has all eight phase-shifts set to 0. For comparison we utilize an ideally symmetrical trot gait which has four phase-shifts set to π and four to $-\pi$. We previously manually designed this trot gait for high velocity for the purpose of a public demonstration.

5.3 Controller Implementation

The strategy is implemented as part of the "Assemble and Animate" (ASE) framework (Christensen et al 2011). The objective of the ASE framework is to develop a flexible and adaptive control framework, which enable rapid development and deployment of modular reconfigurable robots. We designed the framework to be multi-purpose by providing both novel and well-known distributed control strategies for modular robots. Further, the framework is multi-platform which enable us to cross-compile the same control programs (written in C) for both simulated Roombots (Webots), LocoKit (PC or embedded controller), ATRON (USSR or TinyOS), and several other platforms.

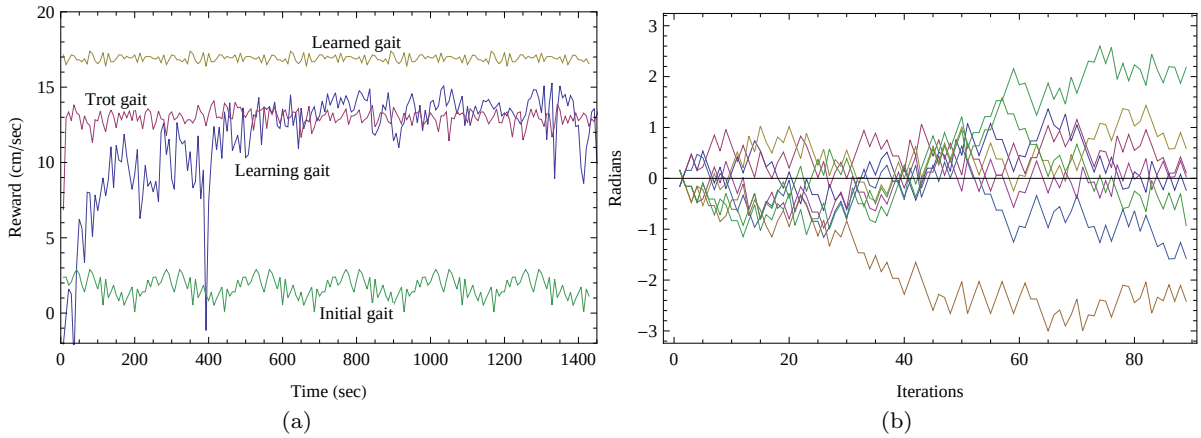


Fig. 3 Online learning with quadruped robot. (a) The reward (average robot velocity in five steps) is shown as a function of time in four cases: for a learning gait, for the initial start-gait of learning, for a manually designed trot gait, and for the learned gait (CPG control and no learning). (b) Online learning of eight control parameters (phase-shifts).

6 Experiments

6.1 Online Gait Learning

In this experiment, we validate the learning strategy on the quadruped robot. As explained above we mount the robot on a boom and send feedback wireless, using ZigBee, from the boom-encoder to a PC. The PC controls the robot's actuators based on the proposed SPSA and CPG based strategy. In each trial, we let the robot learn for a minimum of 20 minutes until the gait velocity has stabilized.

The reward graph of a typical learning example is shown in Fig. 3(a). For comparison the graph also shows the measured velocity of a manually designed trot gait which has an average velocity of 12.9 cm/sec (with a standard deviation of σ=0.49). We observe that the robot while learning improves its initial velocity from around 0 cm/sec until it stabilizes around 13.0 cm/sec after approximately eight minutes. Further, when running the final learned gait without online learning we find that the robot is moving with a velocity of 17.5 cm/sec, which is significantly faster than the manually designed gait. The learning strategy's perturbations of CPG parameters cause this decrease in average velocity while learning.

The adaptations of the corresponding eight open parameters are shown in Fig. 3(b). The parameters do not show any clear convergence towards specific values, but fluctuate over time. The reasons for this includes: (i) The problem does not have one unique solution, but instead a large set of equivalent solutions (e.g. equally phase-shifted parameters) and that many nearby solutions have a similar high performance level. (ii) The learning is life-long and the learning parameters (c_k and a_k) are not gradually decreased, why the strategy continues to explore the parameters (driven by the random vector in Equation 4). The fluctuation of the parameters ensures that the robot can adapt to environmental or morphological changes, but it also means that the system can temporarily deviate towards worse performing solutions. By externally observing the behavior of the robot, it is clear that the solution quickly converge to a trot-like gait. This observation falls well in line with the result from our previous work (Christensen et al 2010b) that different quadruped Roombots also converged to trot-like gaits.

Exp nr.	Learning	Playback
Trot	-	12.9 cm/sec
1	12.8 cm/sec	13.2 cm/sec
2	13.6 cm/sec	17.5 cm/sec
3	14.7 cm/sec	16.9 cm/sec
4	13.3 cm/sec	14.3 cm/sec
5	14.0 cm/sec	15.5 cm/sec

Table 1 Average velocity in five learning trials.

We have run five learning trials with similar results. An overview of the results is shown in Table 1. The table shows the average velocity at the end of five learning trials and the average velocity of "playing back" the learned controller, i.e. CPG control without learning enabled. For comparison, the table also shows the average velocity of the manually designed trot gait. Fig. 4 illustrate the average fitness graph for the five trials.

We observe that the different trials converge to gaits with similar velocity (from 12.8 to 14.7 cm/sec). On average this is slightly faster (13.7 cm/sec) than the manually designed trot gait (12.9 cm/sec). The average velocity of playing back the learned gaits is 15.5 cm/sec, which means that the parameter permutations

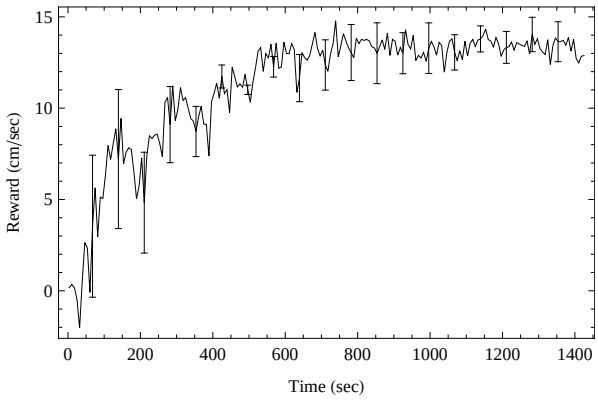


Fig. 4 Average of 5 trials of online learning with LocoKit quadruped. Error bars indicate one standard deviation.

while learning decrease the velocity with on average 11.6 %. In the five trials it takes 4-11 minutes (on average 8 minutes) before the learning robot is moving with an average velocity faster than 12 cm/sec. This fast convergence confirms our results with simulated Roombots which would optimize 45-54 open parameters in 5-30 min. to gaits with 35-50 % faster velocity than gaits found by blind random search (Christensen et al 2010b).

In all five trials the learning converge to similar trot-like gaits but the gaits are not identical or equivalent as we note by observing the variation of the playback velocity in Table 1. We observe that any actuation will bring the some robot into movement that with minor adjustments will cause the robot to move in a specific direction. The strategy gradually (as opposed to in discrete steps) improves the gait from a poor initial gait to a close to optimal gait (see Fig. 3(a)). Based on these observations, we will characterize the fitness landscape as relatively smooth with flat plateaus and local optima. Therefore, the strategy will be very likely to converge to a high velocity gait but it will generally not be a global optima.

In summary, the learning strategy is fast, reliable, and effective in converging to gaits comparable or slightly better than the manually designed gait.

6.2 Online Adaptation to Failures

In this experiment, we use the quadruped LocoKit robot to study the strategy's ability to adapt to failures of the actuators. In two different experiments, we let the robot continue to learn after several of the actuators has stopped in a predefined position to simulate a failure of the actuators.

In the first experiment we stop the two actuators controlling a hind-leg. The result is shown in Fig. 5(a). If we do not enable learning the robot's average velocity drops to 6.9 cm/sec. If we enable online learning, the robot regains an average velocity of 9.61 cm/sec while learning. Although we cannot make strong conclusions based on a single trial, we expect this to be a beneficial effect of adaptation. Simulation experiments on adaptation to failures shows a significant effect for the same strategy in previous work (Christensen et al 2010b).

In the second experiment four actuators controlling the "knee" of each leg is stopped. The result is shown in Fig. 5(b). This causes the velocity to drop to 12.2 cm/sec (no learning). If learning is enabled the robot achieves an average velocity of 13.1 cm/sec while learning and of 14.4 cm/sec if executing the learned gait. In this trial, the effect of adaptation is less clear. In fact it seems that the quick drop in velocity due to the failure, makes the gait diverge temporary to an less efficient gait, which the strategy then readapts to its previous performance within a couple of minutes.

In summary, the experiments indicate that the online learning strategy is able to adapt to morphological changes such as failed actuators, but the concrete effect depends on complex interactions between the robot, its environment, the control system, and the type of failure.

6.3 Optimizing the Leg Design

Manual robot design using the LocoKit is challenging because of the many open parameters such as stick lengths and stick-joint topology. Therefore, manual designs are likely to be suboptimal. In future work, to address this challenge, we plan to design a small module that can dynamically change the length of a stick on the robot. This will enable online adaptation of both morphology and control. However, as a first step to explore the potential we perform offline optimization of the manually designed leg based on a kinematic simulation.

We optimize six selected parameters (stick lengths) from the leg design that are simple to modify on the physical robot. We then use the same SPSA strategy as utilized for online learning to maximize the following fitness function:

$$fitness = \alpha \cdot \frac{A}{A_{max}} + \beta \cdot \frac{S}{S_{max}} + \gamma \cdot \frac{L_{min}}{L} \tag{8}$$

A is the area of the reachable space of the foot, which is important for locomotion in rough terrain. S is the maximum step length, which is essential for high velocity. L is the total stick lengths in the leg, which correlates with stability and physical practicality.

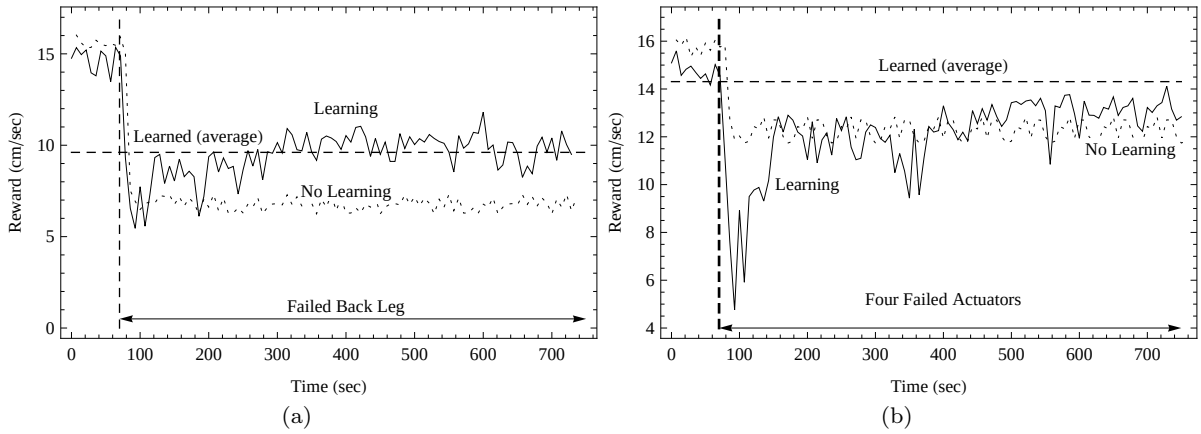


Fig. 5 Online adaptation to actuator failures in the LocoKit quadruped. (a) The effect of single back leg failure (two actuators stops). (b) The effect of failure on all knees (four actuators stop).

To maximize A and S and minimize L we scale each part according to their extreme values, which we find by individual separate optimizations to $A_{max} = 10253mm^2$, $S_{max} = 139mm$, and $L_{min} = 370mm$. Therefore, each part of the fitness function is scaled in the interval $[0, 1]$ and by setting $\alpha+\beta+\gamma = 1$ we can likewise scale the fitness function in the interval $[0, 1]$. To further guide the optimization clear of non-desired designs, we add hard constraints to the solutions. If a solution breaks a constraint it receives $fitness = 0$. Such constraints include continuous foot trajectories and limits on individual stick lengths.

We then combine the different fitness parts by setting: $\alpha = 0.6$, $\beta = 0.25$, and $\gamma = 0.15$ which we experimentally found would balance the leg design in accordance with our expectations. As the initial solution for the optimization strategy, we use the dimensions of the manually designed leg.

Figure 6 shows the fitness graph of a typical optimization run, where SPSA optimize the overall fitness from 0.29 to 0.82. The reachable space and two example foot trajectories are shows for the manual leg design in Figure 7(a) and the optimized leg design in Figure 7(b). We note that the leg possibly has an improved potential for foot movement because of its increased step length. However, since the simulation is limited to kinematics we must study the effects on walking on the physical robot to see if we can transfer the offline-optimized legs to reality. Our working hypothesis is that it is possible to use SPSA to optimize physical parameters online and these results give a first positive indication of feasibility.

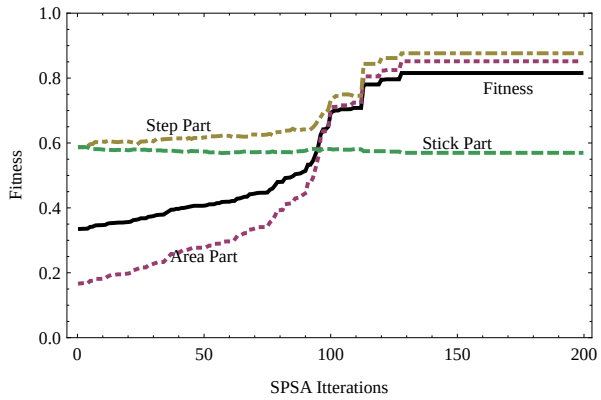


Fig. 6 The fitness graph shows how the overall fitness is optimized over the number of iterations. The development of the three components comprising the fitness function is also shown.

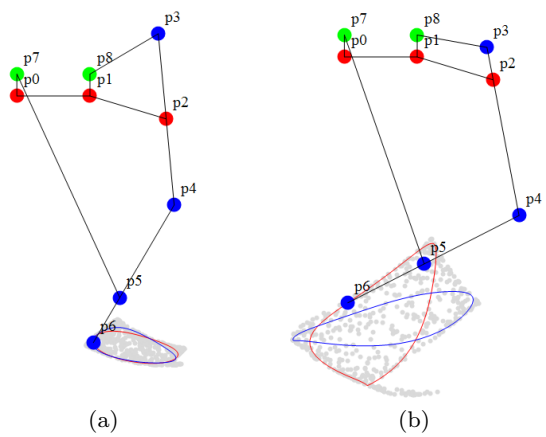


Fig. 7 Optimization of leg design based on kinematics simulation. (a) Manually designed leg, which we use as a reference design. (b) Optimized leg design based on the reference design.

6.4 Online Learning with Offline Optimized Legs

The objective of this experiment was to apply the online learning strategy to LocoKit robots with offline-

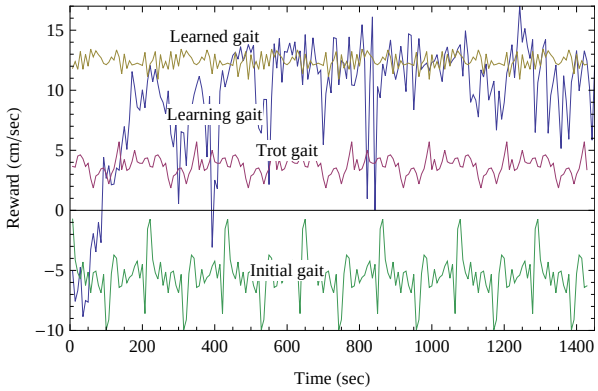


Fig. 8 Online learning on a LocoKit quadruped with automatically optimized leg design.

optimized legs. However, after physical trials with several robot designs we observed that in most cases the automatic optimization process was not able to produce robots able to walk even if we manually designed the gait. The main problem was that legs would move with high amplitude movements that would cause the robot to jump and fall in ways that would quickly break the rather fragile plastic components of the LocoKit. In future prototypes of the LocoKit we have produced the components in a stronger material. However, for these experiments, we introduced additional constraints in the kinematic simulation. We defined a rather narrow legal reachable space - outside this space, a leg design would receive zero fitness. The effect of this constraint was that the optimization process would find leg designs quite similar to the manually designed leg and therefore more likely to be transferable.

We tested one such optimized leg design with the online learning strategy. The trial is shown in Figure 8. The trial is successfully adapting the gait, but with an average velocity of 10.7 cm/sec and 12.4 cm/sec for learning and playback respectively we note that this optimized robot design is not significantly faster than the manually designed robot. The trot and initial gait are identical to the one applied to the manually designed robot. However, note that these gaits produce a very different optimized robot because of the changes in the kinematics design of the legs.

One fundamental problem with this approach is the use of an insufficient simulator. The simulation model only enables kinematic leg optimization and does not allow a holistic evaluation and optimization of the locomotion behavior. Therefore, several of the offline optimized leg designs failed to produce viable solutions, i.e. we failed to transfer the designs successfully from the simulator to the real robot.

To make the reality gap narrower, we could apply a physics-based simulator with rigid body dynamics and collision detection, that could more accurately simulate the robot, e.g. the Unified Simulator for Self-Reconfigurable Robots (Christensen et al 2008b). However, it is a fundamental challenge to transfer simulation results to physical robots due to the difficulty of simulating the complex interactions between the robot and its environment (Brooks 1992; Mataric and Cliff 1996). To address this issue, in future work, we plan to perform the morphological optimization online on the physical robots thereby eliminating the reality gap.

7 Discussion

In this paper, we considered the problem of locomotion and demonstrated a robot able to perform online learning and adaptation to failed actuators. In addition, we demonstrated how to optimize the morphology of the robot in simulation using the same strategy. However, transference to the physical robots was largely unsuccessful. Clearly, we need to resolve a number of challenges before realizing our vision of fully autonomous walking robots able to perform life-long self-adaptation of control and morphology. Some of the most critical challenges are the following:

Autonomous learning: Currently the robot is unable to sense autonomously its own velocity as needed by the learning strategy. Embedding sensors in the robot for detecting self-motion may enable the robot to learn without external sensing (i.e. the boom-encoder feedback). Several types of sensors may be applicable, including inertial measurement units (IMU), and indoor or global positioning systems (IPS/GPS). Specifically, optical flow is a biologically plausible way to sense self-motion (Warren et al 2001; Lappe and Frenz 2009). However, we need a careful comparison of the different alternatives.

Learning in a natural environment: In the presented experiments, we mount the robot on a boom that prevents it from falling over while learning. In order to learn in natural environments the robot must be able to detect such events, for example, by equipping the robot with a tilt-sensor (i.e. accelerometer). The robot could potentially learn a motor sequence to stand itself up based on appropriate sensor feedback, as demonstrated by Morimoto and Doya (2001). In addition, the robot could learn a self-model (Bongard et al 2006) which in some situations could be used to predict and avoid falling over.

Morphology independence: The learning strategy will generally not converge on robot morphologies that tend to be unstable (e.g. humanoids), where body

parts can self-collide, or where the gait velocity has a low signal-to-noise ratio. Currently, the strategy does not attempt to deal with these issues and it is up to the robot designer to construct a robot on which the learning can converge. The strategy does however converge on a large class of legged morphologies and different robot types as demonstrated in this paper and in our previous work (Christensen et al 2010b). We anticipate that by integrating more sensor feedback in the control system it is possible to increase the morphology independence of the strategy.

Learning morphology: The LocoKit is currently not able to adapt its own morphology as required if the robot should be able to online co-learn morphology and control parameters. Self-reconfigurable robots is a type of modular robots able to autonomously change the configuration of their modules (Stoy et al 2010). In future work we will investigate how to enable morphosis in the LocoKit system by designing components that enables the system to change its morphology online (e.g. controllable stick lengths). We anticipate the proposed learning strategy is applicable with minor adjustments to such as co-learning of morphology and control scenario.

Navigation: Currently the robot simply learns a forward moving gait. However, to learn autonomously in natural environments the robot must be able to navigate around obstacles, dangers, and potentially solve a specific task (such as tracking a moving target). We can use the proposed strategy to learn a selection of different gaits, such as turning. A higher-level navigation strategy can select an appropriate gait to steer the robot based on sensors (e.g. distance sensors). Depending on the environment and the task, the navigation strategy is not required to be complex, for example, a modified strategy based on Braitenberg (1986) could be utilized in many situations.

8 Conclusion

This paper described a control strategy for online lifelong learning of locomotion gaits. We implemented a gait as a central-pattern-generator (CPG) and optimized eight open parameters using a stochastic optimization algorithm (SPSA). We experimentally evaluated the strategy on a physical robot constructed from the polymorphic modular robotic LocoKit. We found that the strategy was able to find efficient locomotion gaits by online optimization on average within 10 minutes. We also performed experiments on continued adaptation after failures of several actuators and found that the system was able to readapt after such failures. Finally, we utilized SPSA to offline optimize the kinematics of a leg design. However, we found the reality gap was too wide to transfer successfully the designs to the physical robot.

In the future, we will work towards realizing online co-learning of morphology and control in a natural environment. Further, numerous improvements could be studied for the proposed strategy, most significantly, adaptive learning parameters in order to make the strategy even more generic.

9 Acknowledgements

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.

References

Bongard J (2002) Evolving modular genetic regulatory networks. In: Proceedings of the Congress on Evolutionary Computation (CEC), IEEE Computer Society, Washington, DC, USA, CEC ’02, pp 1872–1877

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

Braitenberg V (1986) Vehicles: Experiments in synthetic psychology. MIT press

Brooks RA (1992) Artificial life and real robots. In: Toward a Practice of Autonomous Systems: Proceedings of the First European Conference on Artificial Life, Cambridge, MA, USA, pp 3–10

Christensen DJ, Bordignon M, Schultz UP, Shaikh D, Stoy K (2008a) Morphology independent learning in modular robots. In: Proceedings of International Symposium on Distributed Autonomous Robotic Systems 8 (DARS 2008), pp 379–391

Christensen DJ, Schultz UP, Brandt D, Stoy K (2008b) A unified simulator for self-reconfigurable robots. In: Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems

Christensen DJ, Schultz UP, Stoy K (2010a) A distributed strategy for gait adaptation in modular robots. In: Proceedings of the IEEE Int. Conference on Robotics and Automation (ICRA), pp 2765–2770

Christensen DJ, Sproewitz A, Ijspeert AJ (2010b) Distributed online learning of central pattern generators in modular robots. In: Proceedings of the 11th International Conference on Simulation of Adaptive Behavior (SAB2010), Paris, France, pp 402–412

Christensen DJ, Schultz UP, Moghadam M (2011) The assemble and animate control framework for modular reconfigurable robots. In: Proceedings of the IROS Workshop on Reconfigurable Modular Robotics: Challenges of Mechatronic and Bio-Chemo-Hybrid Systems

Christensen DJ, Larsen JC, Stoy K (2012) Adaptive strategy for online gait learning evaluated on the polymorphic robotic locokit. In: Proceedings of the IEEE Conference on Evolving and Adaptive Intelligent Systems (EAIS)

Crespi A, Ijspeert AJ (2006) AmphiBot II: An Amphibious Snake Robot that Crawls and Swims using a Central Pattern Generator. In: Proceedings of the 9th International Conference on Climbing and Walking Robots (CLAWAR 2006), pp 19–27

Duff D, Yim M, Roufas K (2001) Evolution of polybot: A modular reconfigurable robot. In: Proceedings of Harmonic Drive International Symposium, Nagano, Japan

Fitch R, Rus D, Vona M (2000) A basis for self-repair using crystalline modules. In: Proceedings, Intelligent Autonomous Systems (IAS-6), Venice, Italy

Fukuda T, Nakagawa S (1988) Dynamically reconfigurable robotic system. In: Proceedings of the IEEE International Conference on Robotics & Automation (ICRA'88), pp 1581–1586

Groß R, Bonani M, Mondada F, Dorigo M (2006) Autonomous self-assembly in a swarm-bot. In: Proc. of the 3rd Int. Symp. on Autonomous Minirobots for Research and Edutainment (AMiRE 2005), Springer, Berlin, Germany, pp 314–322

Ijspeert AJ (2008) Central pattern generators for locomotion control in animals and robots: a review. Neural Networks 21(4):642–653

Jakobi N (1998) Running across the reality gap: Octopod locomotion evolved in a minimal simulation. In: Evolutionary Robotics, Springer, pp 39–58

Jansen T (2007) Theo Jansen: The Great Pretender. OIO Publishers

Kamimura A, Kurokawa H, Yoshida E, Murata S, Tomita K, Kokaji S (2005) Automatic locomotion design and experiments for a modular robotic system. IEEE/ASME Transactions on Mechatronics 10(3):314–325

Kennedy J, Eberhart R (1995) Particle swarm optimization. In: Neural Networks, 1995. Proceedings., IEEE International Conference on, IEEE, vol 4, pp 1942–1948

van den Kieboom J (2009) Biped locomotion and stability a practical approach. Master's thesis, University of Groningen, The Netherlands

Kimura H, Akiyama S, Sakurama K (1999) Realization of dynamic walking and running of the quadruped using neural oscillator. Autonomous Robots 7(3):247–258

Kohl N, Stone P (2004) Machine learning for fast quadrupedal locomotion. In: The Nineteenth National Conference on Artificial Intelligence, pp 611–616

Lappe M, Frenz H (2009) Visual estimation of travel distance during walking. Experimental brain research 199(3):369–375

Larsen J, Garcia R, Stoy K (2010) Increased versatility of modular robots through layered heterogeneity. In: Proceedings of the ICRA Workshop on Modular Robots, State of the Art, Anchorage, Alaska, pp 24–29

Larsen JC, Brandt D, Stoy K (2012) Locokit: A robot construction kit for studying and developing functional morphologies. In: Proceedings of 12th International Conference on Simulation of Adaptive Behavior (SAB 2012), Odense, Denmark, Lecture Notes in Computer Science, vol 7426, pp 12–22

Lipson H, Pollack JB (2000) Automatic design and manufacture of robotic lifeforms. Nature 406:974–978

Maes P, Brooks RA (1990) Learning to coordinate behaviors. In: National Conference on Artificial Intelligence, pp 796–802

Mahdavi SH, Bentley PJ (2003) An evolutionary approach to damage recovery of robot motion with muscles. In: Seventh European Conference on Artificial Life (ECAL03), Springer, pp 248–255

Marbach D, Ijspeert AJ (2004) Co-evolution of configuration and control for homogenous modular robots. In: Proc., 8th Int. Conf. on Intelligent Autonomous Systems, Amsterdam, Holland, pp 712–719

Marbach D, Ijspeert AJ (2005) Online Optimization of Modular Robot Locomotion. In: Proceedings of the IEEE Int. Conference on Mechatronics and Automation (ICMA 2005), pp 248–253

Mataric M, Cliff D (1996) Challenges in evolving controllers for physical robots. Robotics and Autonomous Systems 19(1):67 – 83

Morimoto J, Doya K (2001) Acquisition of stand-up behavior by a real robot using hierarchical reinforcement learning. Robotics and Autonomous Systems 36(1):37–51

Murata S, Kurokawa H (2012) Self-Organizing Robots. Springer

Pfeifer R, Lungarella M, Iida F (2007) Self-organization, embodiment, and biologically inspired robotics. Science 318(5853):1088–1093

Pouya S, van den Kieboom J, Spröwitz A, Ijspeert AJ (2010) Automatic gait generation in modular robots: "to oscillate or to rotate; that is the question". In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Taipei, Taiwan, pp 514–520

Righetti L, Ijspeert A (2006) Programmable central pattern generators: an application to biped locomotion control. In: Proceedings 2006 IEEE International Conference on Robotics and Automation (ICRA), pp 1585 –1590, DOI 10.1109/ROBOT.2006.1641933

Silva MF, Machado JT (2012) A literature review on the optimization of legged robots. Journal of Vibration and Control 18(12):1753–1767

Sims K (1994) Evolving 3d morphology and behavior by competition. In: Brooks R, Maes P (eds) Proc., Artificial Life IV, MIT Press, pp 28–39

Spall JC (1992) Multivariate stochastic approximation using a simultaneous perturbation gradient approximation. IEEE Transactions on Automatic Control 37(3):332–341

Sproewitz A, Moeckel R, Maye J, Ijspeert AJ (2008) Learning to move in modular robots using central pattern generators and online optimization. Int J Rob Res 27(3-4):423–443

Stoy K, Nagpal R (2004) Self-repair through scale independent self-reconfiguration. In: Proceedings of IEEE/RSJ International Conference on Robots and Systems (IROS), Sendai, Japan, pp 2062–2067

Stoy K, Lyder A, Garcia R, Christensen DJ (2007) Hierarchical robots. In: Proc. of the IROS Workshop on Self-Reconfigurable Modular Robot, San Diego, CA

Stoy K, Brandt D, Christensen DJ (2010) Self-Reconfigurable Robots: An Introduction. Intelligent Robotics and Autonomous Agents series, The MIT Press

Taga G, Yamaguchi Y, Shimizu H (1991) Self-organized control of bipedal locomotion by neural oscillators in unpredictable environment. Biological Cybernetics 65:147–159, 10.1007/BF00198086

Warren W, Kay B, Zosh W, Duchon A, Sahuc S (2001) Optic flow is used to control human walking. Nature neuroscience 4(2):213

Wolpert DH, Macready WG (1997) No free lunch theorems for optimization. IEEE Transactions on Evolutionary Computation 1(1):67–82

Yim M, Shen WM, Salemi B, Rus D, Moll M, Lipson H, Klavins E (2007a) Modular self-reconfigurable robot systems: Challenges and opportunities for the future. IEEE Robotics & Automation Magazine 14(1):43–52

Yim M, Shirmohammadi B, Sastra J, Park M, Dugan M, Taylor CJ (2007b) Towards robotic self-reassembly after explosion. In: Video Proceedings of the IEEE/RSJ Intl.Conf.on Intelligent Robots and Systems (IROS), San Diego CA

Yoshida E, Murata S, Tomita K, Kurokawa H, Kokaji S (1999) An experimental study on a self-repairing modular machine. Robotics and Autonomous Systems 29:79–89

Zahadat P, Christensen DJ, Schultz UP, Katebi SD, Stoy K (2010) Fractal gene regulatory networks for robust locomotion control of modular robots. In: Proceedings of the 11th International Conference on Simulation of Adaptive Behavior (SAB2010), Paris, France

Zahadat P, Schmickl T, Crailsheim K (2012) Evolving reactive controller for a modular robot: Benefits of the property of state-switching in fractal gene regulatory networks. From Animals to Animats 12 pp 209–218