Adaptation of a distributed controller depending on morphology

Nathan Labhart and Shuhei Miyashita
Artificial Intelligence Laboratory
University of Zurich
Zurich, Switzerland
{labhart,miya}@ifi.unizh.ch

Abstract

In this paper we investigate the influence of an agent's morphology on its neural controller. Our model consists of a number of identical modules, each of which comprises of two half-wheels for movement and a Central Pattern Generator (CPG) as its own neural control. Based on a series of simulation experiments, we conclude that one single type of CPG can adapt well to different types of morphologies, and that there seems to be a suitable or optimal morphology depending on the environmental givens.

Keywords: Morphology, Central Pattern Generator, evolutionary algorithm

1 Introduction

It is a widely accepted fact that the neural controller of an agent has to match the complexity of its body as well as the complexity of its task environment ("Principle of Ecological Balance", [1]). Normally, controller and morphology evolve in parallel, mutually influencing each another, and taking into account the interaction with the environment. In this paper, we investigate the influence of an agent's morphology on the evolution of its neural control – inspired by the centipede, where locomotion is achieved by "synchronizing" a number of two-legged body segments.

In the present research, a series of simulation experiments are carried out with modular robots, where each module consists of a body and a Central Pattern Generator (CPG).

J. C. Bongard and R. Pfeifer tested hypotheses about the behavioral effect of specific morphological features by keeping the neural controller constant across different body sizes, masses, and morphologies [2]. The essential issues of how to develop a cellular robotic system were described by Y. Kawauchi et al. Their project "CEBOT" included optimization methods for the structure of both hardware and software [3]. S. Murata et al designed a modular robotic system in hardware, which could metamorphose into desired configurations, and showed results of changes in morphology [4]. In the "Conro" project by A. Castano, A. Behar, and P. M. Will, each module was self-contained (it included its own processor, power supply, communication system, sensors, and actuators). These modules were designed to work in groups as part of a large configuration [5]. A similar project is presented by M. W. Jørgensen et al with "ATRON", which consists of several fully self-contained robot modules [6]. Using nonlinear coupled oscillators for the CPG was described by A. Ijspeert and J.-M. Cabelguen. Their experiments on salamander movement show how configurations could combine global couplings from the segment oscillators in order to achieve "realistic" locomotion data [7].

2 Model

2.1 Single Module

In this first set of experiments, the agents are simulated in the physics engine ODE. In its base form, a module consists only of a motor, rigid joints to stick several modules together, one set of half-wheels, and a neural control mechanism (Fig. 1).

The half-wheels of one module are connected by a fixed axis (i.e. they always turn in parallel) and are constrained in movement from $-\frac{\pi}{2}$ to $+\frac{\pi}{2}$. This restriction is introduced to exploit the interaction with the environment: by taking into account the static and dynamic friction on the ground, the back-and-forth movement of the half-wheels causes the whole configuration to move.

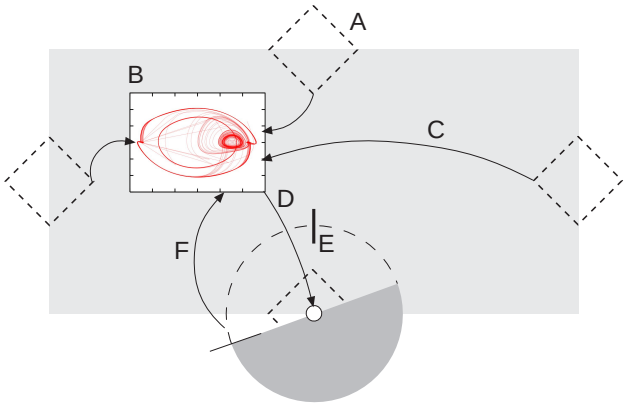


Figure 1: A single module. A: connector to neighbor module; B: CPG; C: input from neighbor module; D: forward/backward command; E: stopper (to restrict movement from $-\frac{\pi}{2}$ to $+\frac{\pi}{2}$); F: half-wheel angle input.

2.1.1 CPG

The CPGs are modeled as nonlinear oscillators in x, v space, taking the angles of the corresponding half-wheels as inputs and returning a "forward" or "backward" command depending on a threshold θ.

The controller inside every module is connected not only to its own half-wheels, but also to the immediately neighboring modules. Thus, every CPG has a limited knowledge about the state of its neighborhood, but not about the entire configuration:

$$\tau \dot{v}_i = -\frac{x_i^2 + v_i^2 - E_i}{E_i} v_i - x_i + \sum_j w_j s_j \tag{1}$$

$$\tau \dot{x}_i = v_i \tag{2}$$

where i is the number of the current module, j is the number of the neighbor module, w_j are the weights for the connection to the neighbors s, and τ and E_i are constants. This CPG is an oscillator with amplitude $\sqrt{E_i}$ and period $2\pi\tau$.

$$\begin{cases} x + v > \theta_{high} & \text{"backward"} \\ x + v < \theta_{low} & \text{"forward"} \end{cases} \tag{3}$$

2.2 Configurations

Like building blocks, these modules are attached to each another to form various morphologies. In order to understand the influence of morphology on the controlling mechanism, we implement the same type of neural control in two different configurations and compare the CPGs as well as the traveled distances.

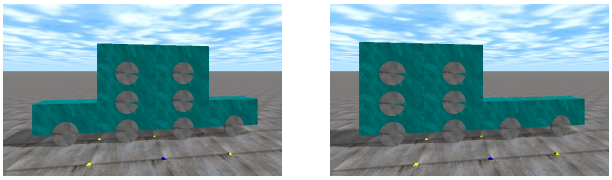
module mass	1.4kg	τ	0.2
module L×W×H	1×0.5×0.5m	θ_{low}	-0.3
half-wheel radius	0.2m	θ_{high}	0.3

Table 1: Simulation properties

This model is deliberately kept simple and easily extensible for future experiments. In addition, it allows a relatively straightforward way of implementing it in hardware in order to test the outcome of the simulation in the real world.

3 Simulation

We implement two configurations with 8 modules each, which differ only in the distribution of the four top modules, so that the center of gravity is shifted to one end of the robot (Fig. 2).



(a) "Hat-shaped" (b) "Slanted hat"

Figure 2: Two configurations simulated in ODE.

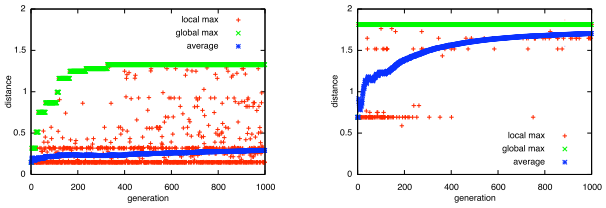
In order to understand the influence of morphology on the controlling mechanism, we let the two configurations evolve the E_i parameters of their CPGs over 1000 generations and then compare their performance. Minimal Generation Gap [8] evolution is applied to adapt this controller to the respective morphology (i.e. configuration of the modules). The fitness function is defined as the distance a configuration can cover within a limited time frame. In addition, we analyzed the trajectories of the CPG parameters from the first, the 500th, and the 1000th generation.

For each generation, the simulation runs for 2000 time steps. In every 50th step, the positions of the half-wheels are recorded, and the CPG values (x, v) are calculated for 1000 iterations (Δt steps), taking into account the half-wheel positions. The resulting CPG values define whether a "forward" or "backward"

command is issued to the corresponding half-wheel. Once the movement command has been executed, the configuration "slides" on the ground for a few (less than 50) timesteps, and the next CPG calculation cycle takes place.

4 Results and discussion

As can be seen from the distance plots (Fig. 3), the asymmetrical configuration clearly "makes use" of the shifted center of gravity with an average distance of 0.8387m, compared to 0.4482m in the symmetrical configuration. The different morphology is also reflected in the CPG trajectories (Fig. 4, 5).



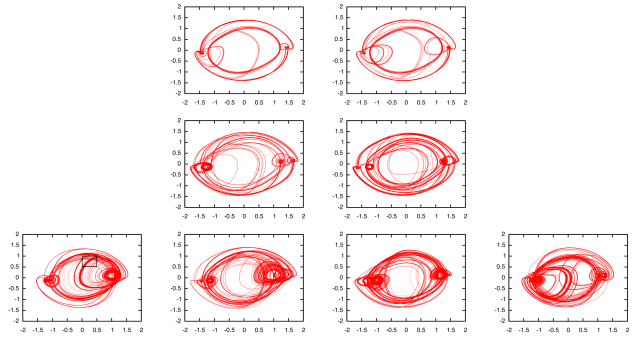
(a) "Hat-shaped" (b) "Slanted hat"

Figure 3: Performance (traveled distance) of both configurations.

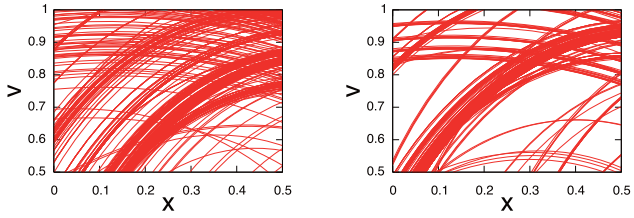
The experiments show that a slight change in morphology results in different performances. The average distance covered by the asymmetrical configuration is about 6 times as large as that of the symmetrical one; furthermore, the symmetrical morphology takes much longer to evolve its CPG parameters into a set of values that performs well.

By comparing the CPG's x, v trajectories of one morphology over time we find that the further the configuration moves, the more distinct the trajectories become (Fig. 4(b), 5(b)). In contrast to the symmetrical "hat-shaped" morphology, the CPG adapts to the asymmetrical morphology of the "slanted hat" configuration more quickly, and the CPG trajectories are more distinct.

A comparison of the two morphologies suggests that evolution apparently exploits the friction on the ground, taking into account the asymmetrical weight distribution of the "slanted hat" morphology where the weight on the left side is higher than on the right.

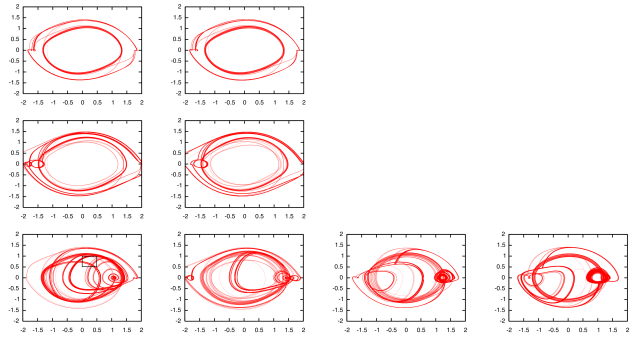


(a) "Hat-shaped" morphology

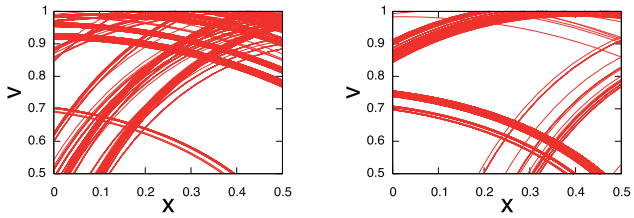


(b) Magnified trajectories from the bottom-left module of generation 1 (left) and 1000 (right)

Figure 4: CPG trajectories of the symmetrical configuration.



(a) "Slanted hat" morphology



(b) Magnified trajectories from the bottom-left module of generation 1 (left) and 1000 (right)

Figure 5: CPG trajectories of the asymmetrical configuration.

These results imply two things: On the one hand, one single CPG can adapt well to different types of

morphologies; on the other hand, there seem to be suitable or optimal morphologies depending on the environmental givens (in this case, friction).

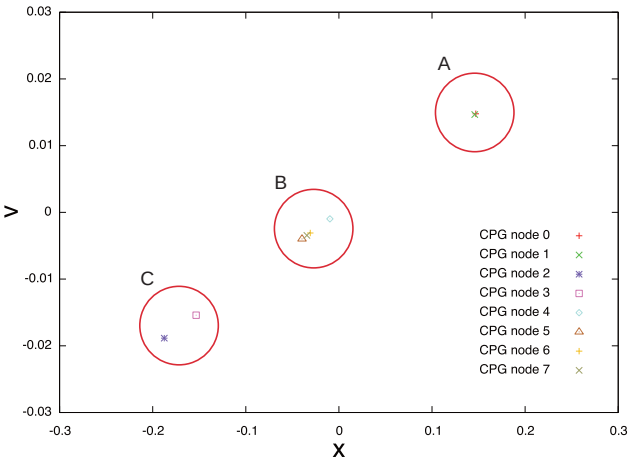


Figure 6: Mean values of all CPGs in the 1000th generation of the symmetrical configuration. The four modules which touch the ground are marked with A and C, the four upper modules are marked with B.

Furthermore, we observe in the symmetrical configuration that the four modules which touch the ground shift their CPG's center (mean value) away from the zero point in an asymmetric way (Fig. 6). This enables the "hat shaped" morphology to move at all, rather than staying at the same position (as could be expected from its symmetrical shape). The top four modules, where the half-wheels do not directly influence the movement of the configuration, keep their CPG centers very close to the zero point. The "slanted hat" morphology does not exhibit such a behavior, as it is already asymmetric by design.

Acknowledgments

This research is partially supported by the Swiss National Science Foundation, project #200021-105634/1 and project #200021-109864/1.

References

[1] Pfeifer R, Scheier C (1999), Understanding Intelligence. MIT Press, Cambridge, pp. 312–315

[2] Bongard JC, Pfeifer R (2002), A Method for Isolating Morphological Effects on Evolved Behaviour. In: Hallam B, Floreano D et al (ed), Proceedings of the Seventh International Conference on the Simulation of Adaptive Behaviour (SAB2002), MIT Press, pp. 305–311

[3] Kawauchi Y, Inaba M, Fukuda T (1994), A Study on Cellular Robotic System (A Realization of a Robotic System Capable of Adaptation, Self-organization, and Self-evolution). JRSJ 12(1):116–132

[4] Murata S, Tomita K, Yoshida E et al (1999), Self-Reconfigurable Robot – Module Design and Simulation. Proceedings of 6th International Conference on Intelligent Autonomous Systems (IAS-6), pp. 911–917

[5] Castano A, Behar A, Will PM (2002), The Conro Modules for Reconfigurable Robots. IEEE/ASME Transactions on Mechatronics, 7(4):403–409

[6] Jørgensen MW, Østergaard EH, Lund HH (2004), Modular ATRON: Modules for a self-reconfigurable robot. Proceedings of 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 2068–2073

[7] Ijspeert A, Cabelguen J-M (2003), Gait Transition from Swimming to Walking: Investigation of Salamander Locomotion Control Using Nonlinear Oscillators. In: Kimura H, Tsuchiya K, Ishiguro A et al (ed), Adaptive Motion of Animals and Machines. Springer, Tokyo, pp. 177–188

[8] Satoh H, Yamamura M, Kobayashi S (1996), Minimal Generation Gap Model for GAs Considering Both Exploration and Exploitation. Proceedings of IIZUKA '96, pp. 494–497