Neuro-Controllers for Walking Machines - An Evolutionary Approach to Robust Behavior

Joern Fischer, Frank Pasemann, and Poramate Manoonpong

Fraunhofer Gesellschaft, Department of Autonomous Intelligent Systems (AIS.INDY), Schloss Birlinghoven, D-53754 Sankt Augustin, Germany
joern.fischer@ais.fhg.de, frank.pasemann@ais.fhg.de, poramate.manoonpong@ais.fhg.de

On a first view complex machines with many degrees of freedom seem to need a complex control structure. Articles about the architecture of the controllers of walking machines, which probably belong to this category, often confirm this prejudice [8, 3]. Using evolutionary techniques to obtain small neural controllers seems much more promising [5, 4, 1, 2]. In the following article we present a method to evolve neural controllers keeping the complexity of the networks and the expense to evolve them small. We analyze the neural entities in order to understand them and to be able to prevent an unwanted behavior. As a last proof the resulting networks are tested on the real walking machine to find out the limitations of the physical simulation environment as well as of the neural controller. Our real platform is a six-legged walking machine called Morpheus. The task it performs is an obstacle avoidance behavior in an unknown environment.

1 Introduction

To evolve and to understand efficient neural controllers for physical walking machines is still a challenge in the area of robotics [5]. Many attempts to evolve such controllers end up in networks, which are too large to be understood in detail. Our aim is to evolve small, simple and robust networks, which can be coupled together in order to get a more complex behavior. This modular neurodynamics approach is used together with a physical simulation environment to evolve controllers for real world tasks. Here this procedure is demonstrated for a six-legged robot, called Morpheus. The following chapter describes the main technical specifications of the platform. The next one starts with the evolution of behavior in a physical simulation environment, which then leads to an implementation on the real platform. Finally, the results are discussed and an outlook is given.

2 The research platform: Morpheus

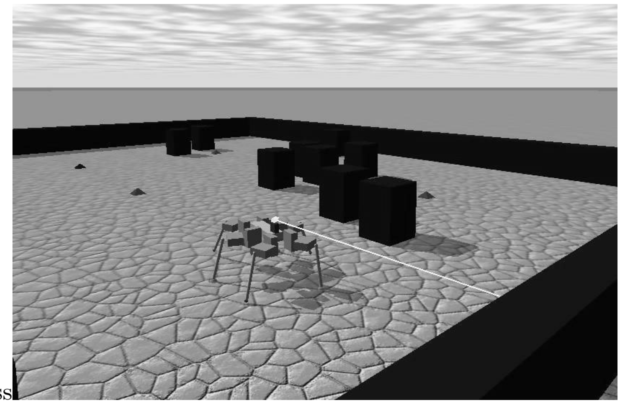
Morpheus is a six-legged robot, where each leg has two degrees of freedom. Each joint is controlled by a strong full metal servomotor. Like on real insects and scorpions the alignment of the feet is kept on an ellipsoid, which helps the robot, when it is turning around and enhances its stability. The levers are kept short and the body is narrow to ensure optimal force exploitation. The building blocks are kept modular to be able to change components very quickly and to enable the construction of different morphologies. Each of Morpheus joints has a potentiometer, which is used as a sensor for the actual leg position. In addition an infrared sensor is placed on a pivoting head on the front side of the robots body. All in all Morpheus has 13 active degrees of freedom and 13 sensors (the head motor does not return an angle value). The control of the robot is kept on a simple but powerful board, the "MBoard", which is able to control up to 32 motors, and which has 36 analog sensor inputs and a size of 130 mm x 42 mm.

3 Evolving neurocontrollers for walking

To use an evolutionary approach effectively it is necessary to have a physical simulation environment, which simulates the robot together with the environment as fast and as exact as possible. Our simulator is based on ODE (Open Dynamics Engine)[1] and enables an implementation, which is faster than real time and which is precise enough to mirrow corresponding behavior of a physical robot. This simulation environment is connected to our Integrated Structure Evolution Environment (ISEE). Structure evolution refers to the fact that the networks architecture is not fixed and that the number of neurons as well as of connections can be enlarged or reduced during the evolutionary process [7]. To preserve parts of a neural network which have already an appropriate performance, it is possible to fix these structures, and to continue the evolution by adding neurons or connections or removing parts of the network which are not fixed.

After having implemented Morpheus in the physical simulator one has to define a suitable fitness function, which in the simplest case may be given by the distance the robot walked during a given time interval. The desired neurocontroller for the pure walking task has no input neurons and 12 output neurons providing the motor signals. For simplicity connections between output neurons are suppressed in the first experiment. The simulation environment is a simple plane on which Morpheus should learn to walk along a straight line. The simplest solution for the hidden layer was found to be a quasi-periodic oscillator composed of two neurons [6]. By using symmetric output weights a typical tripod gate was obtained, which enabled an efficient forward movement.

[1] see also: http://opende.sourceforge.net/



ss

Fig. 1. The simulated robot surviving by performing an obstacle avoidance behavior.

3.1 Evolving neurocontrollers for obstacle avoidance

For the second task, obstacle avoidance, is endowed with an additional motor, turning an infrared sensor left and right in an oscillatory motion. Furthermore, one has to insert obstacles into the simulated environment. The difficulty with environments for obstacle avoidance is to define a suitable fitness function. The robot should move as far as it can, but to avoid obstacles may start to walk in circles. This should of course be prevented. The simplest way to define the fitness is to take the Euclidean distance from the start to the end point of the robots trajectory, and to let it run in an environment, called the "witches cauldron" (comp. Fig.2).

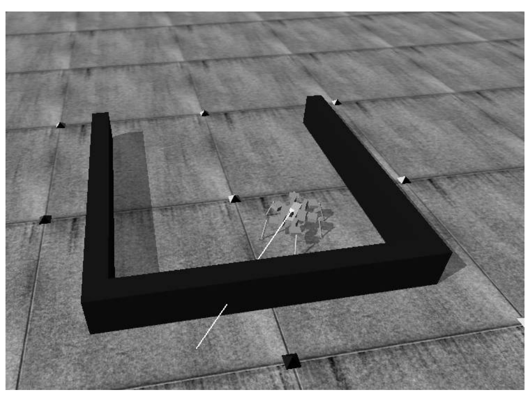


Fig. 2. For each episode only one of the 4 walls is left out at random. With this strategy only those individuals survive, which are able to avoid the walls. The fitness function is simplified to the Euclidean distance of the robots movement.

This environment consists of walls around the starting point, with one of the sides open. This opening is chosen at random at the beginning of each episode. Then each individual has to pass through the opening in order to maximize its fitness value. To reuse the controller evolved first for the tripod gait, its structure is fixed and used for a starting population of obstacle avoidance controllers. Various results were obtained by the evolution process, performing more or less good, but to describe the mechanisms will go beyond the scope of this article. The simplest network with a reasonable performance is shown in Fig.3. The infrared sensor directly inhibits the neurons, which are responsible for the movement of the upper leg motors of the right side. If an obstacle is detected, the amplitude of the movement of these motors is scaled down and the robot turns to the right side in order to avoid the obstacle.

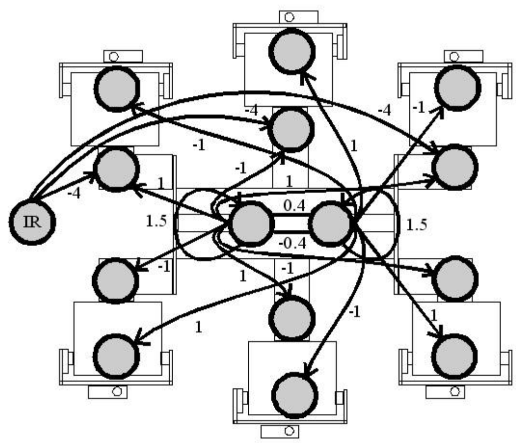


Fig. 3. This is the simplest neural network we evolved for an obstacle avoidance behavior. The input neuron and the 13 output neurons were predefined and should not have recurrent connections. With only two hidden neurons Morpheus walks forward. The infrared sensor is directly connected to the neurons for the upper motors on the right side of Morpheus. If an obstacle is detected these neurons are driven into saturation, the concerning motors move with lower amplitude and the robot turns to the left.

4 Implementation on the physical machine

The network shown in Fig.3 is first tested on Morpheus with a serial connection from a 1.8 Ghz personal computer and later it is implemented as an assembler program directly into the robots microcontroller. In both cases the network was updated with 50Hz. This limit was given by the servo controller boards which work with a PWM signal of a period of 20 ms. The behavior was the same in both cases, though the autonomous version calculated the output and the synaptic weights each in a range of only one byte. Both systems showed a good performance, even in more complex environments. Difficulties appeared only for legs of chairs or desks. Fig.4 shows a typical behavior of the six-legged robot avoiding an unknown obstacle.

ss

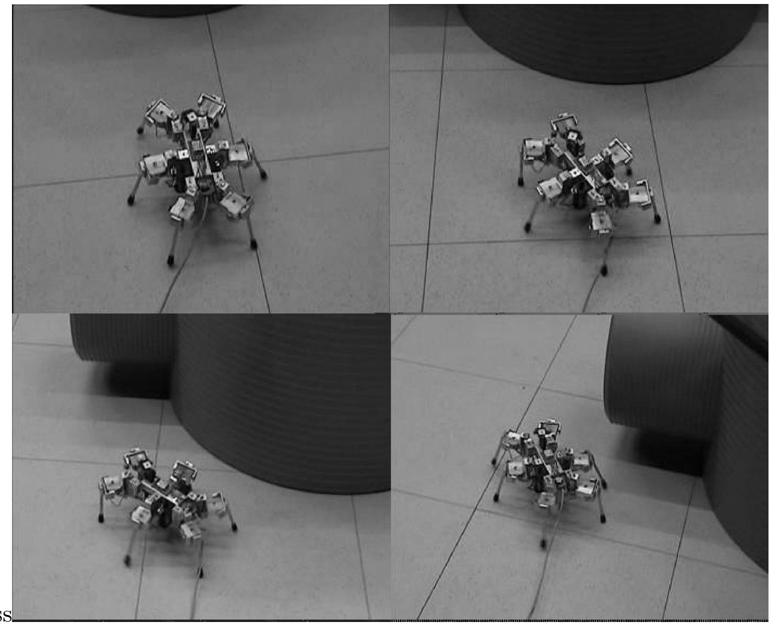


Fig. 4. The obstacle avoidance behavior of the six legged walking machine Morpheus.

5 Conclusion and outlook

With a physical simulation environment and our structure evolution algorithm we evolve networks for different tasks. We begin with a task of low complexity and analyze the resulting networks. We decide which weights should be fixed and after increasing the task difficulty we continue with the evolution by adding or removing neurons from the network and evaluating the new individuals. We obtain small and robust networks, which may be understood in their functionality as well as in their dynamical properties. This method is demonstrated on the six-legged robot Morpheus performing an obstacle avoidance task. In our future research we will work on the fusion of neural entities and of new methods to analyze time discrete neural networks.

References

1. Beer, R.D., Chiel, H.J. and Gallagher, J.C. (1999) Evolution and analysis of model CPGs for walking II. General principles and individual variability. J. Computational Neuroscience 7(2):119-147
2. Calvitti, A. and Beer, R.D. (2000) Analysis of a distributed model of Leg coordination. I. Individual coordination mechanisms. Biological Cybernetics 82:197-206
3. Cruse, H., Dean, J., Kindermann, T., Schmitz, J., Schumm, M. (1999) Walknet - a decentralized architecture for the control of walking behavior based on insect studies In: Hybrid Information Processing in Adaptive Autonomous Vehicles.(ed) G. Palm. Springer
4. Gruau, F. (1995) Automatic definition of modular neural networks, Ph. D. thesis, Ecole Normale Superieure de Lyon, Adaptive Behavior 3(2), 151–184.
5. Parker G.B., Li Z. (2003) Evolving Neural Networks for Hexapod Leg Controllers, Proceedings of the 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)(pp. 1376-1381).
6. Pasemann, F., Hild, M., Zahedi, K. , SO(2)-Networks as Neural Oscillators, Lecture Notes In Computer Science, Biological and Artificial computation: Methodologies, Neural Modeling and Bioengineereng Applications, IWANN 7'th 2003
7. Pasemann, F., Steinmetz, U. Hlse, M. and Lara (2001) B. "Robot Control and the Evolution of Modular Neurodynamics", Theory in Biosciences, 120, pp. 311-326
8. Schmitz, J., Dean, J., Kindermann, Th., Schumm, M., Cruse, H. (2001) A biologically inspired controller for hexapod walking: simple solutions by exploiting physical properties. Biol. Bull. 200, 195-200 (invited paper)