Constructing a Small Humanoid Walking Robot as a Platform for the Genetic Evolution of Walking

Jens Ziegler[1], Krister Wolff[2], Peter Nordin[2], and Wolfgang Banzhaf[1]

August 17, 2001

[1] University of Dortmund, Dept. of Computer Science
D-44227 Dortmund, Germany
E-mail:{ziegler,banzhaf}@ls11.cs.uni-dortmund.de
[2] Chalmers University of Technolgy, Dept. of Physics
S-41296 Göteborg, Sweden
E-mail:{wolff,nordin}@fy.chalmers.se

Abstract

Walking robots form the next challenge in the field of autonomous robots. This paper describes the construction of a fully autonomous humanoid walking robot as a platform for machine learning algorithms like e.g. Genetic Programming. Built from off-the-shelf components, the described humanoids are cheap, robust and easy to program, which make them an ideal test platform for several experimental approaches in machine learning, sensor fusion, or adaptive control. In addition to these research related topics, the walking robots are an ideal tool for educational purposes.

1 Introduction

The aim of this work is twofold: first, it presents a small humanoid and fully autonomous walking robot that is in turn used as, second, a test platform for the evolution of walking with Genetic Programming. Autonomous mobile robots are becoming more and more important, because they are expected to solve tasks that humans are not able to cope with or that humans ought not to cope with [3, 5, 6, 7]. This requires extensively autonomous robots, because with growing complexity of the problems it will be no longer possible for the programmer to take all eventualities into account from the outset. A special form of mobile robots are walking robots. This term includes all robots that locomote without wheels, caterpillars or similar devices on firm ground. Estimations say that almost 40% of the earth's land mass is not accessible for robots on wheels. Walking robots are successfull even in commercial applications. SONY just started with their second edition of the four-legged robot dog AIBO[1], and demonstrated a small two legged humanoid robot, a smaller

[1] http://www.aibo.com

version of the well known Honda P3[2], that is ought to assist in nursing the elderly in the future.

The evolution of robot control programs has been the topic of recent publications. A general introduction into the concept of Genetic Programming can be found in [1, 2, 12]. Several applications of Genetic Programming (or, more generally, Evolutionary Algorithms) to the task of controlling autonomous robots are given in, e.g., [10, 11, 15, 16]. The evolution of crawling or walking robots can be found e.g. in [13, 17, 9]. Especially the field of walking robots becomes more and more important, a fact that recently lead to the constitution of a german national research program "Autonomes Laufen", that includes different research groups, from engineering to computer science and biology. For biological inspiration, gait patterns of stick insects have been analyzed to gain more detailed information on natural gait coordination algorithms [8], that in turn has influences on the design of robust and fast walking gait patterns.

To evolve gait patterns or walking agents, one has to have a machine which does the walking: a robot. This leads to a general decision between two approaches: (1) using a real robot for the experiments, or (2) using a simulated robot. The pros and cons for each of the solutions cannot be listed here, since the literature would fill several book shelves. A short introduction into the discussion with additional references can be found in [4]. For our purposes, the real robot solution has been favored, for several reasons: (i) a real robot comes along with real world problems, not with artificial ones. (ii) The results of a machine learning technique like GP can impressively be demonstrated with a real walking robot, much more impressive than with e.g. plots of fitness values. (iii) the construction of a small humanoid robot under the constraints given in sect. 2 is a challenging task in itself, and thus worth of doing.

The following sections are devoted to the description of the hardware, firmware, and software components of the small humanoid walking robots ELVINA and ZORC.

2 Hardware

This section describes the hardware components of the robot, including actuators, skeletton, and sensors. The robot is about 35 cm high and weighs about 1.5 kg, if all necessary devices are mounted. This includes the camera, the controller, the batteries, etc.

2.1 Actuators

To stick with the idea of a robot as unexpensive as possible, all motors were selected from available standard servo motors. The motors are Hi-Tec[3] servos with the following specifications:

- HS-945 MG, with $\omega = 0.16 sec/60^o$ at 4.8 V and an output torque of $8.8 kg \cdot cm$ at 4.8 V. Weight = 56g
- HS-225 MG, with $\omega = 0.14 sec/60^o$ at 4.8 V and an output torque of $3.9 kg \cdot cm$ at 4.8 V. Weight = 32g

Two different types of motors were used because of different demands for different joints. The knee and ankle joint have to be very powerful and need the stronger motors. The

[2]http://www.honda.co.jp/english/technology/robot

[3]http://www.hitecrcd.com

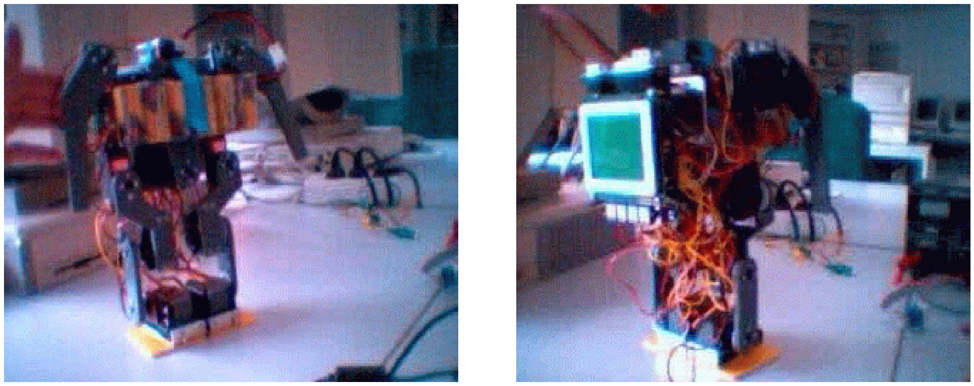

Figure 1: ZORC, standing autonomously on two legs.

arms and shoulder servos need not to be so powerful, so that these joints can be built from weaker and lighter servos.

The servos have a workspace according to their position on the robot. A servo that moves the arm forward and backward may have a 180^o degree workspace, while the torso servo will have only 90^o freedom. The exact dimensions of the workspace are defined in the **hardware description table (HDT)**, whose values are additionally influenced by the servo type (manufacturer, etc.). The HDT has to be cross-compiled before it is downloaded to the robot. Each servo can then be controlled by setting the nominal values for the angle. The nominal values lie in an intervall $[0, \ldots, 256]$, where 0 and 256 represent the minimum (maximum) angle, which ist given by the HDT. All intermediate angles are linearly interpolated in 256 steps. It is a useful feature of this type of motors that no effort (neither computationally nor with additional sensors) has to be spent to control the angle. All servos have an integrated controller (normally a P-controller) and autonomously adjust the necessary torques to keep the angle as close as possible to the desired value. This, on the other hand, leads to an almost unlimited need for electrical current, should the servo be prevented to reach the desired value, either by an awkward configuration of the limbs or by a hardware restriction (collision). The digital control circuits and the power supply circuits therefore have to be separated, to avoid uncontrolled impulses on control lines.

2.2 Skeleton

The skeleton of the robot is made of 5mm thick PVC material that is easy to cut and grind. It is, additionally, light and rugged. All plastic parts are connected via the servos, so that servos and plastic together form the skeletton of the robot. A picture of the robot can be seen in fig. 1. To keep with the human model, all dimensions of the parts were specially designed. The length of the upper leg equals almost the length of the lower leg, which is the natural ratio of a human leg. The length of the upper body is amost the same as the length of the legs and the center of gravity of the fully equipped robot lies slightly above the center of rotation of the torso. These ratios all mirror the dimensions of a human body, so that it really can be seen as a simplified scaled version. The form of the legs can be seen in detail in fig. 2.

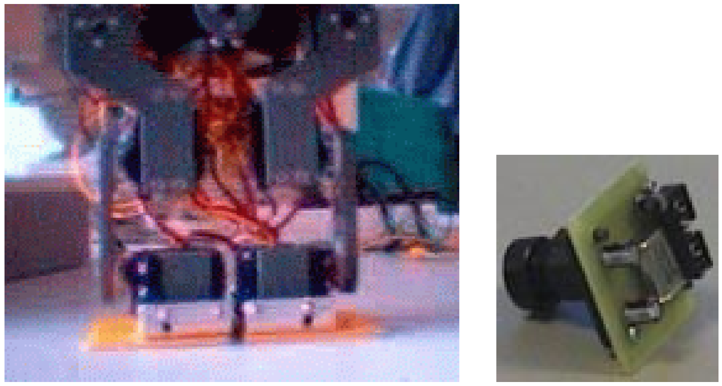

Figure 2: RIGHT: Detailed view of the construction of the legs. LEFT: The CMOS camara of the robots.

2.3 Sensors

The robot is equipped with a color CMOS camera (see fig. 2) that can be mounted on the head. Data from the camera can directly be sent to the controller. The controller (explained in detail in sect. 3) is able to process up to three frames per second, a framerate that drops if the image processing routines get more precise. It is therefore necessary to keep a good balance between necessary computational effort to evaluate sensor input and the more essential calculation of motions. It is possible to add more sensors to the robot (e.g. infrared distance sensors), but the robot does not have them in the actual version.

3 Controller

The humanoid has the EyeBot MK3[4] controller on-board, mounted as a backpack. Some of the main features of the controller may be given here:

- 25MHz 32bit Controller (Motorola 68332)
- large graphics LCD (128x64 pixels)
- 1 parallel port, 2 serial ports, 8 digital inputs, 8 digital outputs, 8 analog inputs
- battery level indication
- 4 input buttons

A picture of the back mounted controller can be seen in fig. 3. The controller can be connected to a PC/Workstation via the serial interface. This type of connection allows a speed of up to 128 kBit/s. It is possible to mount a wireless data link, so that the robot is not affected by cables. Due to the fact that the serial link is only necessary for downloading programms, ELVINA and ZORC don't have a wireless connection. to the host computer. The programmes can be developed offline on the host computer and, after a cross compillation, can be downloaded in a special format to the robot controller. The push buttons onboard the controller allow to have a simple user interface for the programs running onboard the controller.

[4]Available via http://www.joker-robotics.com. A website with all features of the EyeBot MK3 is http://www.ee.uwa.edu.au/ braunl/eyebot/extra/features.txt.

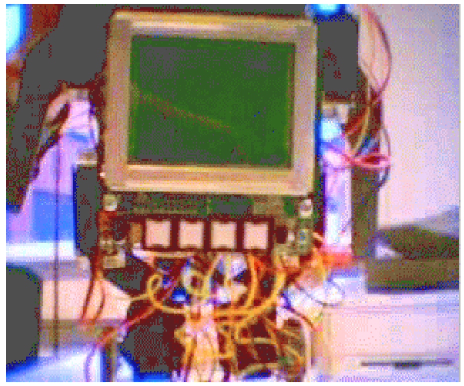

Figure 3: The controller, mounted on the back of ZORC.

4 Evolution of Motions

Programming and control of humanoid robots is difficult, because of the high dimensionality of the movements and the complex sensory and motor limitations, let alone the various uncertanties that arise during the operation. Our approach of machine learning does not imply that the evolved controllers depend on specific information on the robot morphology such as certain lengths, masses or distances. On the contrary, it is one of the main goals of this work to make the evolution of robot controllers as independent as possible from morphology related information. This is the opposite of a usual engineering approach, that crucially depends on exact information about masses, lengths and inertia tensors in order to construct trajectory based movements of robots, which is computationally expensive and slow. If a robot controller uses information of the kinematic model to solve its control task —e.g. to compute an inverse transformation in order to calculate the joint angles of a robot arm consisting of rotational joints from a given position of the TCP (*tool center point*) in world coordinates— the correctness of the calculation depends on unchanged parameters of the hardware, either real or simulated. It is in fact a hard task to reach a sufficient stiffness of the involved hardware (gears, joints, and bodies) to comply with the computed values. If, like it is the case with the used servo motors, the accuracy of the absolute positioning of the motors decreases with operating time or a joint looses the ability to reach certain positions, then the outcome of the inverse transformation which depends on the correct mathematical model is useless. Additionally, the algorithm for the inverse transformation is correct only for a single robot. So, in our case, morphology-related information, although available in principle, will not be used. Another new approach [14] uses imitation to control the movements of a humanoid robot, again in order to avoid the inflexible trajectory based control.

4.1 From single joint motions to walking

A first step toward control of movements of a legged robot is to move the single joints according to a desired movement of more complex parts of the robot, e.g. a limb. This partial movements of all involved joints obviously depend on the desired behavior of the whole (e.g. swing or stance phase of a single leg) and have to be coordinated and synchronized in order to get the desired motion. The movements of joints are caused by actuators that, basically, apply a torque to two or more connected rigid bodies. Thus, describing a movement of a leg of a robot requires to give the time dependent values

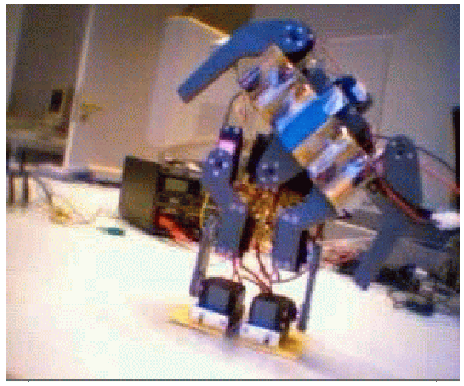

Figure 4: ZORC, doing a small step with his right leg.

of the acting torques for each involved joint during the motion. Most robot languages encapsulate the necessary torques and require only a nominal value for either angle or translation length of the motor.

Coordinating the movement results in the necessity to give a time series of motive forces or nominal angles for each joint of the leg, which sums up to 4 joints for a single leg and an overall of 13 joints for the small humanoid walker at each discrete timestep t during the motion.

4.2 The Genetic Programming System

Controlling the movement of a robot leg requires a sequence of instructions for each joint of the leg. This sequence has to be coordinated in time to achieve the desired movement in sufficient quality. The Genetic Programming system now has to fulfill certain requirements:

- The structure of the individual has to be interpreted as a robot control program.
- Therefore it is necessary to have operations in the global operator set that allow to control motors by (i) adressing them and (ii) assinging a nominal value. This value can either be a moment or an angle and depends of the implemented firmware (drivers).
- The quality of the executed individual has to be measured and fed back into the algorithm.

4.3 Fitness

The fitness of an individual can then be seen as the sum of squared differences between a certain point of the robot and an a priori defined ideal trajectory, that e.g. describes the desired movement of this special point in space and time. The quality of a movement can be measured as the sum of squared differences between sample points and the nominal value. The motion of the observed points are sampled with a given frequency ω, so that f is

$$f(x) = \sum_{i=0}^{t_{end} \cdot \omega} \big(x(i) - \overline{x}(i)\big)^2, \tag{1}$$

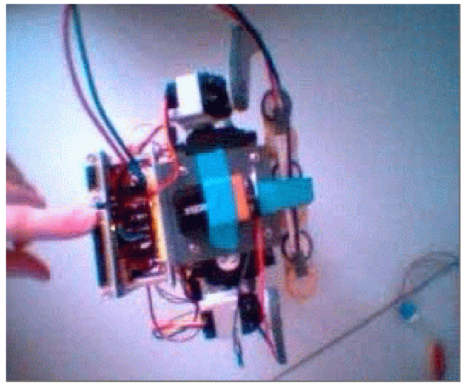

Figure 5: Top view of ZORC.

where $\overline{x}(i), x(i)$ are the values (positions) of the observed point (the nominal position) at time $t = i/\omega$. The motion of the robot can for example be measured by a top mounted camera (a view of the camera can be seen in fig. 5), or, to get the position and orientation of the robot in 3D coordinates, with two cameras. The camera(s) sends the frames to a special graphics routine that extracts the position of the robot in world coordinates. The difference between the robot and the nominal position can then be determined exactly. It is possible in principle to measure the movements of the robot with other devices (e.g. in [9], the robot tows a computer mouse to measure the motion on a 2D plane), but a non-interfering method is to favor here, because an additional device might have unpredictable influence on the stability of an upright standing legged robot. The fitness function (1) is a straightforward way of measuring the quality of a control program. There are many additional properties of the robot's behavior that can be used to calculate the fitness, like energy consumption, step height, body movement, etc.

It has proven in preliminary experiments that to speed up the evolution it is useful to abort the execution of a walking program in an early stage of the evaluation when its intermediate fitness values exceed a certain threshold. This prevents the GP algorithm to spend too much time in evaluating individuals that probably have poor quality.

5 Links

A collection of photos and movies of ZORC can be found on the web. A first official 100cm-run-contest between ZORC and ELVINA took place at the Harenberg Center, Dortmund in January 2001.

- http://ls11-www.informatik.uni-dortmund.de/people/ziegler/robotics.html
- http://ls11-www.informatik.uni-dortmund.de/people/ziegler/ZORC-Gehen01.m1v
- http://ls11-www.informatik.uni-dortmund.de/people/ziegler/ZORC-Stehen02.avi

6 Discussion and Outlook

This paper describes in detail the construction of a small humanoid walking robot. It is fully autonomous, equipped with servo motors and a color camera. The robots are designed to allow easy experiments with machine learnig algorithms like Genetic Programming to evolve control programs that will enable the robot to learn to walk. An experimental setup is presented that will be used in future work to examine the possibilities of the evolution of locomotive behaviors without any internal model or representation of the real machine.

Acknowledgements

This project is supported by the Deutsche Forschungsgemeinschaft (DFG), under grant Ba 1042/6-1, and is partly funded by the DAAD (Deutscher Akademischer Austauschdienst).

References

[1] P. J. Angeline. Genetic programming and emergent intelligence. In K. E. Kinnear, Jr., editor, *Advances in Genetic Programming*, chapter 4, pages 75–98. MIT Press, 1994.

[2] W. Banzhaf, P. Nordin, R. E. Keller, and F. D. Francone. *Genetic Programming – An Introduction; On the Automatic Evolution of Computer Programs and its Applications.* Morgan Kaufmann, dpunkt.verlag, 1998.

[3] V. Braitenberg. *Vehicles: experiments in synthetic psychology.* MIT Press, 1984.

[4] R. Brooks. Artificial life and real robots. In *European Conference on Artificial Life*, pages 3–10, 1992.

[5] R. A. Brooks. A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation*, 1:253–262, 1988.

[6] R. A. Brooks. New approaches to robotics. *Science*, 253:1227 – 1232, 1991.

[7] R. A. Brooks. Coherent behavior of many adaptive processes. In D. Cliff, P. Husbands, J.-A. Meyer, and S.W. Wilson, editors, *From Animals to Animats 3: Proceedings of The Third International Conference on Simulation of Adaptive Behavior*, pages 22–29. MIT Press/Bradford Books, 1994.

[8] H. Cruse. Coordination of leg movement in walking animals. In *From animals to animats. Intl. Conf. on Simulation of Adaptive Behavior*, pages 105–119, 1991.

[9] P. Dittrich, A. Bürgel, and W. Banzhaf. Learning to control a robot with random morphology. In *Proceedings Evo-Robot-98, P. Husbands and J.-A. Meyer, Eds.*, pages 165–178, 1998.

[10] J. J. Grefenstette and A. C. Schultz. An evolutionary approach to learning in robots. In *Machine Learning Workshop on Robot Learning*, New Brunswick, NJ, 1994.

[11] L. Gritz and J. K. Hahn. Genetic programming evolution of controllers for 3-D character animation. In J. R. Koza, K. Deb, M. Dorigo, D. B. Fogel, M. Garzon, H. Iba, and R. L. Riolo, editors, *Genetic Programming 1997: Proceedings of the Second Annual Conference*, pages 139–146, Stanford University, CA, USA, 13-16 July 1997. Morgan Kaufmann.

[12] J. R. Koza. *Genetic Programming.* MIT Press, Cambridge, MA, 1992.

[13] M. A. Lewis, A. H. Fagg, and A. Solidum. Genetic programming approach to the construction of a neural network control of a walking robot. In *Proceedings of the 1992 IEEE InternationalConference on Robotics and Automation*, pages 2618–2623, Nice, France, May 1992.

[14] M. Matarić. Getting humanoids to move and imitate. *IEEE Intelligent Systems*, pages 18–24, 2000.

[15] P. Nordin and W. Banzhaf. Genetic programming controlling a miniature robot. In E. V. Siegel and J. R. Koza, editors, *Working Notes for the AAAI Symposium on Genetic Programming*, pages 61–67, MIT, Cambridge, MA, 10–12 November 1995. AAAI, Menlo Park, CA.

[16] M. Olmer, W. Banzhaf, and P. Nordin. Evolving real-time behavior modules for a real robot with genetic programming. In *Proceedings of the international symposium on robotics and manufacturing*, Montpellier, France, May 1996.

[17] G. F. Spencer. Automatic generation of programs for crawling and walking. In Stephanie Forrest, editor, *Proceedings of the 5th International Conference on Genetic Algorithms, ICGA-93*, page 654, University of Illinois at Urbana-Champaign, 17-21 July 1993. Morgan Kaufmann.