EVOLUTION OF EFFICIENT GAIT WITH AN AUTONOMOUS BIPED ROBOT USING VISUAL FEEDBACK

Krister Wolff and Peter Nordin

Chalmers University of Technology
Department of Physical Resource Theory, Complex Systems Group
S-412 96 Göteborg, Sweden
email: wolff@fy.chalmers.se, nordin@fy.chalmers.se

Abstract

We have developed an autonomous, walking humanoid robot 'elvina' and performed experiments in evolutionary programming with it, in order to optimize a by hand developed locomotion controller. A steady state evolutionary strategy is running on the robot's onboard computer. Individuals are evaluated and fitness scores are automatically determined using the robots onboard vision system and sensors. By using this system, we evolve gait patterns that locomote the robot in a straighter path and in a more robust way than the previously manually developed gait did.

1 Introduction

The applications of robots with human-like dimensions and motion capabilities are plentiful. In a world where man is the standard for almost all interactions, humanoid robots have a very large potential acting in environments created for humans. Both in industry and in academia walking humanoid robots attracts an accelerating interest (Nordin & Nordahl, 1999). In 1996, Honda Corporation presented the P2 humanoid robot, which is a biped robot that can walk like a human, even up and down stairs. A smaller and lighter robot, P3, was introduced in 1997 and recently, they presented the humanoid robot ASIMO, which is conceived to function in an actual human living environment in the near future (Honda, 2001). The Sony Corporation announced in November 2000 the development of a small biped walking robot, SDR-3X , which is a platform for exploration of new possibilities for entertainment robots (Sony, 2001).
In traditional robot control programming, an internal model of the system is derived and the inverse kinematics can be calculated. The trajectory for movement between given points in the working area of the robot is then calculated from the inverse kinematics. The traditional geometric approach to robot control, based on modelling of the robot and derivation of leg trajectories, is computationally expensive and requires fine-tuning of several parameters in the equations describing the inverse kinematics (Nolfi & Floreano, 2000).Conventional industrial robots work in a well defined and hard controlled environment and could hardly cope with the reality noise in a human living environment. They are designed in such a way that a model can be easily derived,

but for the development of bio-inspired robots, this is not a primary design principle. Thus, a model of the system is very hard to derive or to complex so that a model-based calculation of actuator commands requires to much CPU time to be useful in practice. However the traditional approach still is a very common, we propose for several reasons the concept of evolutionary adaptive systems for control programming of so-called bio-inspired robots as e.g. a humanoid (Dittrich et al, 1998), (Wolff & Nordin, 2001).

The primary goal for our work presented in this paper is to evolve a gait pattern, using evolutionary programming and especially evolutionary strategies (Banzhaf et al, 1998). To do this, one has to choose between two main alternatives: using a real robot for the evolution, or using a simulated robot. Several experiments with simulations, with different approaches, have been reported recently. A methodology for developing simulators for evolution of controllers in minimal simulations has been proposed and shown to be successful when transferred to a real, physical octopod robot (Jakobi, 1998). This was also compared with a controller that was evolved with a real octopod robot (Gomi & Ide, 1998). It was found that it matched better the physical constraints of the robot hardware. Using simulation, ball-chasing behavior has been evolved and successfully transferred to a real AIBO (Sony, 2001) quadruped robot dog (Hornby et al, 2000). When a high degree of fidelity is necessary, it is desirable to be able to evolve with a physical robot. We want to show that evolution of controllers with complex, physical robots can be carried out in reality, although evolving with a simulator would be many times faster.

The first attempt in using a real, physical robot to evolve gait patterns was made at the University of Southern California (Lewis et al, 1992). Neural networks were evolved as controllers to get a tripod gait for a hexapod robot with two degrees of freedom for each leg. Recently, a group of researchers at Sony Corporation presented the results of their work with evolving locomotion controllers for dynamic gait of their quadruped robot dog AIBO (Hornby et al, 1999) and (Hornby et al, 2000). These results show that evolutionary algorithms can be used on complex, physical robots to evolve non-trivial behaviors on those robots.

Evolution of static walking with a biped robot is much more difficult than it is with a robot that has a greater number of legs. A static gait requires that the projection of the center of mass of the robot on the ground lie within the support polygon formed by feet on the ground (Nolfi & Floreano, 2000). This is obviously easier to fulfill with a robot that got four, six, eight or more legs. However, dealing with biped locomotion leads us into a partly different problem domain. When a biped robot is walking (static), it is supported only by one foot at the ground during an appreciable period of time. Only this single foot then constitutes its support area. For a biped robot, the area of support is relatively small, compared to the altitude of its center of mass. The corresponding measure for a robot with four or more legs is more favourable. Therefore it is easier for a robot with many legs to maintain balance than it is for a biped robot, as the motion of walking dynamically changes the stability of the robot.

Our test problem is that of developing locomotion controllers for static gaits for our biped robot 'elvina'. In previous evolution of gaits with physical robots has a humanoid, biped robot not been used.

2 Robot Platform

The robot used in our experiments is 'elvina', which is a simplified, scaled model of a full-size humanoid with body dimensions that mirrors the dimensions of a human. The 'elvina' humanoid is a fully autonomous robot with onboard power supply and computer, however many experiments are performed with external power supply. It is 28 cm tall and it weights about 1.49 kg including batteries. Each of the two legs has 5 degrees of freedom, of which 4 dof is active and 1 dof is passive. The head, the torso and the arms has 1 dof each, giving a total of 14 dof. The robot is equipped with a digital CMOS color camera, mounted in its head. The computer is attached to the back of the robot's body. The body also houses a near-infrared PSD (position sensitive detector) that is used to determine distances to nearby objects. In its present status, the robot is capable of static walking.

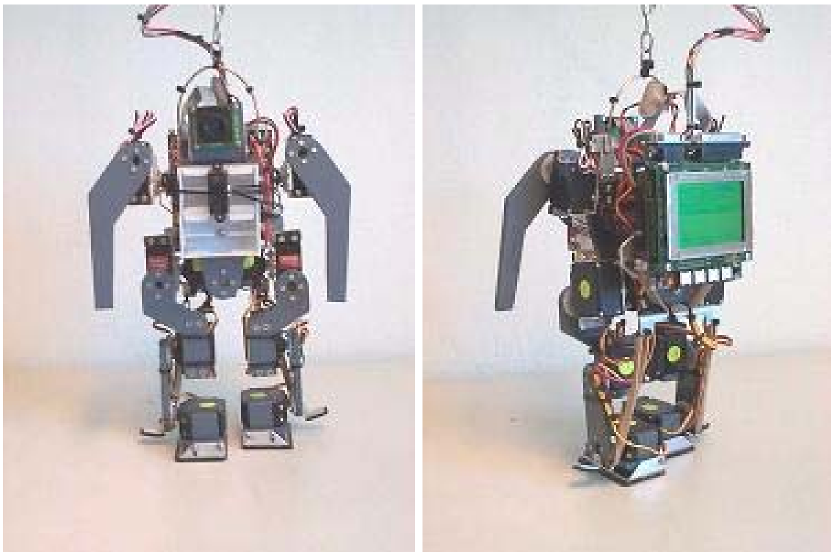

Figure 1: Pictures of the 'elvina' humanoid

2.1 Mechanical and electrical properties

The body structure of the robot is constructed with the actuators as the main elements. The actuators that constitute the different sections of the body are connected to each other with parts made of 6 mm thick PVC board so that they together form the robot body. This PVC material fulfills all necessary requirements since it is inexpensive and lightweight, yet strong and durable.

The robot is assembled with standard off-the-shelf R/C servomotors as actuators. This kind of servo has an integrated closed loop position control circuit, which detects the pulse-code modulated signal that emanates from the controller board for commanding the servo to a given position. In this implementation, each servo is assigned a target position by the robot control program by addressing it an integer value within the interval $\{0, 255\}$. Two different sizes of servomotors are used for 'elvina'. For the four ankle and hip joints we use the stronger ones with an output torque of 0.87 Nm and for

the other eight joints we use servomotors with an output torque of 0.38 Nm.
Since the actuators are very energy consuming, the power supply of the robot turns into a delicate problem. The power source must meet requirements such as high capacity, low weight and reasonable costs. Although the controller board can provide power at a constant voltage, a separate power circuit for the actuators is preferable. Noise and power glitches produced by the high currents of these components must not be allowed to interfere with the controller board circuits. The robot is equipped with four 1.2 volt, 1700 mAh NiMH cells as power source for the actuators and a single 9 volt alkaline battery for the controller board, altogether with a weight of 0.15 kg. This gives the robot an effective operating time of approximately twenty minutes.
The robot has the EyeBot MK3 (Bräunl, 2002) controller onboard, carrying it as a backpack. The EyeBot MK3 consists of a 32-bit micro-controller board with a graphics display and four push buttons for user input, and it also has a serial communications interface. The robot control programs are developed on a host computer, and after a cross-compilation the binary is downloaded to the EyeBot controller. The serial line is then only used for uploading experimental data to the host computer since all signal processing is carried out on the EyeBot controller itself.

2.2 Sensory System

Vision is the most important sensor for this robot. Therefore, it is equipped with a full color 24-bit digital camera, which is based on CMOS technology. The camera is directly connected to the controller board, and physically attached to an actuator on top of the robots torso. This arrangement gives the camera, relatively to the robot's body, one degree of freedom in the horizontal plane and a camera sweep angle of 85 degrees.
A single camera cannot be used to accurately measure the distance to a nearby object. This is instead achieved with a near-infrared PSD range sensor, which consists of an IR emitter and a position sensitive detector in a single package. The principle of this sensor is based on triangulation. The emitter, placed below the detector in the package, illuminates a small spot on an obstacle with modulated IR light. A lens forms an image of the spot on the active element at the back of the detector. The output of the detector element is a function of the position on which the image is falling (Jones et al, 1999). Within the range of about eight to 40 cm distance to the object, a value of sufficient accuracy (resolution less than 1 mm) is produced.

2.3 Firmware and Software

The EyeBot MK3 controller board is running an operating system that consists of two main parts, the Robot Basic I/O System, RoBIOS, and the Hardware Description Table, HDT. The same RoBIOS is shared by all hardware configurations of a robot controlled by an EyeBot, but the HDT differs to account for different sensors or actuators connected to the actual hardware. Each actuator has a unique workspace according to its position on the robot (Ziegler et al, 2001). The workspaces for all actuators are specified in the HDT file by setting suitable values.

3 Evolutionary Algorithm

The evolutionary algorithm used is a steady state evolutionary strategy (Banzhaf et al, 1998), running on the robot's onboard computer. The method of steady-state tourna-

ment selection is used to select individuals to breed. This implies that there are no well-defined generations, but a successive change of the population (Nordin, 1997).
The initial population is composed of 30 individuals with 126 genes randomly created with a uniform distribution over a given search range. The search range for each parameter type (e.g. speed, delay and servo position) is determined from experience in manually developing gaits. The search range is defined as the magnitude of the Euclidean distance between a certain gene in the manually developed individual and the corresponding gene in a randomly created individual.
We use a tournament selection to select individuals for parents and the individuals to be replaced by their offspring. Four different individuals are randomly picked from the population and then evaluated one at a time. The two individuals who get the higher fitness are considered as parents and their offspring, produced by recombination and mutation, replaces the two individuals with the lower fitness in the population. The number of tournaments a certain individual can be selected to be in is unrestricted.
For reproduction both mutation and recombination is used. Recombination takes the two individuals considered as parents, p_1 and p_2, and creates two child individuals, c_{1i} and c_{2i}. Each gene of the child c_{ki} then gets the value

$$c_{ki} = p_{ki} + \alpha_{ki}(p_{1i} - p_{2i}) \tag{1}$$

where c_{ki} is the *i*:th gene of the *k*:th child individual, p_{ki} is the *i*:th gene of the *k*:th parent individual, p_{1i} and p_{2i} are the *i*:th gene of the two parents p_1 and p_2. The α_{ki} is a number randomly chosen to be either -1 or +1.
In each of the child individuals produced, 20 % of the genes are mutated by a small amount. The genes in these two individuals are selected by random to undergo mutation and it is possible for a gene to be mutated several times. The gene to be mutated gets a value according to the equation

$$c_{ki,mutate} = c_{ki} + \delta_{ki} m_{ki} \tag{2}$$

where $c_{ki,mutate}$ is the mutated *i*:th gene of the *k*:th child individual, c_{ki} is the gene to be mutated. The δ_{ki} denotes a number randomly chosen to be either -1 or +1. The m_{ki} are a random number with uniform distribution that determines how much each gene should be mutated and it is set proportional to each parameter type's search range. That is, for the delay parameter, m_{ki} values are set to maximum 6% of its search range and for the speed and servo position parameters, m_{ki} values are, in a similar way, set to 33% maximum respectively.

4 Experimental Method

The short term aim in our experiments is to optimize a set of integer values, used as control parameters for a biped robot gait. They should make the robot move faster, straighter and in a more robust manner than the manually developed set of parameters.

4.1 Experimental Setup

The experimental environment is shown in the figure below. The robot is placed on top of a table with a surface of relatively low friction during the evolution. A target wall of 50 cm height and white color is placed at one end of the table and to mark the center of that end, there is a vertical black stripe on the wall. Right above the robot (65 cm above the table surface) there is a horizontal beam, used as a carrier for the power supply cables and for the security chain. In order to minimize the cable's influence on the robot during locomotion, there is a counterweight connected to the cables via a string that is extended around a pulley. The purpose of this arrangement is that the counterweight drags the cables and the security chain as the robot moves forward.

Figure 2: The experimental setup

4.2 Evaluation

Each individual is evaluated under as equal conditions as possible. The robot's starting position is at a distance of about 40 cm from the wall and facing it. The experimenter centers the robot according to the black stripe by using its onboard camera. Once centered, the robot measures its distance with the PSD infrared range sensor and starts to move towards the wall. After a fixed number of gait cycles it stops. Again it measures its distance from the wall and pans its head (camera) to search for the black stripe on the wall. Using these measurements and the time required for the locomotion trial, it calculates a fitness value for this actual individual. The robot is then manually reset to its starting position by the experimenter for the next individual to be evaluated.

4.3 Sensor feedback

After an individual has performed a trial, the camera is used to determine how straight the robot moved during the trial, with good precision. The difference from the desired (straight) path of locomotion and the actual path is considered as the angular deviation θ.

While the robot uses its onboard camera for determination of direction, distances are measured using a near-infrared PSD range sensor located at the robot's chest. Initially the robot is manually moved to a specific starting point, about 40 cm from the target wall. At this fixed position, the start distance is determined by averaging six consecutive PSD sensor readings, with 0.2 s interval between the readings. After an individual has performed a trial successfully (i. e. without falling) it stops and again uses the PSD sensor to determine its stopping distance from the wall.

4.4 Fitness

To determine an individual's fitness score both its average velocity during the trial and its ability to move in a straightforward path is taken into account. The fitness score function is defined as

$$score = v(d_0, d_f, t) \cdot s(\theta, d_f) \tag{3}$$

where $v(d_0, d_f, t)$ is the average velocity of the robot during the trial and $s(\theta, d_f)$ is the straightness function. The d_0 and d_f denote the initial and the final distances to the target wall respectively and t is the time passed during the trial. The straightness function is dependent of both the angular deviation θ and the robot's final distance to the target wall and it is thus defined as

$$s(\theta, d_f) = \frac{d_f(f(\theta) - 1) + 150 - 10f(\theta)}{140} \tag{4}$$

Here, $f(\theta)$ is a normalization function to convert θ into a 0 to 1 measure. The values 150 and 10 are used as constants for the straightness function because they are raw values corresponding to the maximum and minimum measurable distances for the PSD sensor. The straightness function accounts for the robot's final distance from the black target stripe, with the robot at a fixed orientation θ being larger when the robot stops closer to the target wall. Finally, the average velocity function is defined as

$$v(d_0, d_f, t) = \frac{d_0 - d_f}{t} \tag{5}$$

In the case when an individual does not maintain the robot's balance during a complete trial (e.g. the robot falls) it receives a score of zero. This has to be done manually by the experimenter, since the experimental system is not equipped with any device for detecting that kind of occurrence.

5 Results

For this experiment we used an initial population of 30 individuals. The best-evolved individual received a fitness score of 0.1707. The manually developed individual was also tested and received a fitness score, averaged over three trials, of 0.1051 ± 0.02. The best-evolved individual outperformed the manually developed individual both in its ability to maintain the robot in a straight course and in robustness, i.e. with a less tendency to fall over.

In the below figure the average fitness scores is shown as dots and the line is produced by statistical analysis, i.e. linear regression, of the dots. Since the slope of the line is positive, we observe a tendency towards better and better fitness values. There was no improvement observed in the robot's speed of locomotion. The best-evolved individual and the manually developed individual could both move the robot in a speed of approximately 10.5 cm/minute. They were tested over a 100 cm run, which took 9.5 minutes each.

Tested with the manually developed individual, the robot proved to have a sidewise deviation of about 33 cm on a 100 cm distance. When the best-evolved individual was tested, it moved the robot forward on both types of surfaces with a sidewise deviation of less than 5 cm on a 100 cm distance.

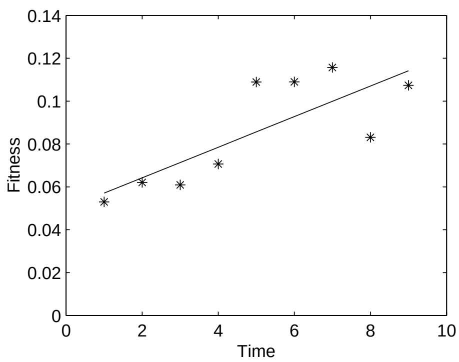

Figure 3: The Resulting Averaged Fitness Scores.

6 Discussion and conclusions

The work presented in this paper constitutes of two main parts, the construction of a small humanoid walking robot and experiments in evolutionary programming performed with it. By manually developing locomotion module parameters, the robot was made capable of autonomous static walking in a first stage. In the next stage we performed an evolutionary experiment on the robot in order to improve the manually developed gait. For this, we used a steady state evolutionary strategy that was run on the robot's onboard computer. This algorithm evolved an individual that outperformed the previously manually developed set of parameter values in a sense that it moved the robot in a straighter path and in a more robust way. To run an evolutionary experiment took about five hours. Between each tournament of four individuals evaluated, we paused the experiment for about 15 minutes in order to spare the hardware

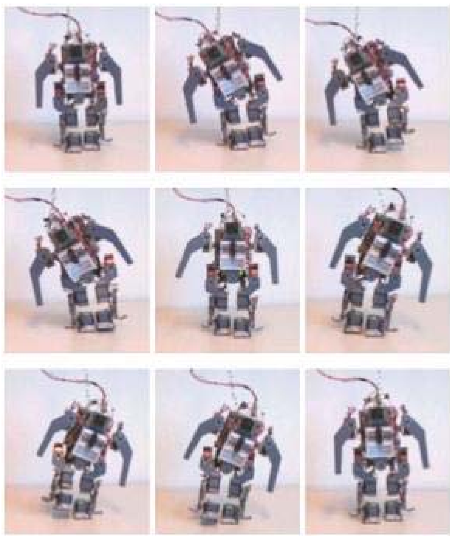

Figure 4: Series of pictures showing a complete gait cycle, from top left to bottom right.

and especially the actuators. The main reason for this is that the actuators accumulate heat when they are running continuously under heavy stress. They then run the risk of getting overheated and gradually destroyed. We also observed that the position control circuit of the servomotors is sensitive to temperature. When commanding a servomotor to a given position by addressing it a fixed integer value within the interval $\{0, 255\}$, the physical angle of the servo's output shaft dislocates over time as the temperature of the servo increases. Since the robot's feet are coupled to each other via nine actuators their relative positions are then affected so much by this drift that it could cause the robot to fall. One way to handle this problem is, as mentioned above, to run the robot intermittent so that the servos maintain an approximately constant temperature.
Evolving efficient gaits with real physical hardware is a challenging task. In the six months of manually developing gaits and testing the evolutionary algorithm, frequent maintenance of the robot was indeed necessary. The torso and both the ankle actuators were exchanged once as well as the two hip servos. The most vulnerable parts of the robot were proved to be the knee servos. Both these servos were replaced tree times.
We believe that the 'elvina' robot platform is capable of development. In future research we aim to improve the hardware and software of the 'elvina', resulting in a more robust and durable robot platform. Then it will be possible to do more experiments with evolution in order to have the robot to accomplish tasks that are more complicated. Such could be using vision to navigate, collaborate and interact with other robots of this kind and balancing and walking on an inclined plane.

References

Banzhaf W., Nordin J. P., Keller R. E. & F. D. Francone (1998), *Genetic Programming An Introduction: On the Automatic Evolution of Computer Programs and Its Applications.* San Francisco: Morgan Kaufmann Publishers Inc.

Bräunl T. (2002), *EyeBot - Online Dokumentation.* Last visited: 27 February 2002 <http://www.ee.uwa.edu.au/~braunl/eyebot/>

Dittrich P., Burgel A. & Banzhaf W. (1998), *Learning to move a robot with random morphology.* Phil Husbands and Jean Arcady Meyer, editors, First European Workshop on Evolutionary Robotics (165-178). Berlin: Springer-Verlag.

Gomi T. & Ide K. (1998), *Emergence of gait of a legged robot by collaboration through evolution.* P. K. Simpson, editor, IEEE World Congress on Computational Intelligence. New York: IEEE Press.

Honda Motor Co., Ltd (2001), *Honda Robot Top Page.* Last visited: 27 February 2002 <http://world.honda.com/robot/>

Hornby G. S., Fujita M., Takamura S., Yamamoto T. & Hanagata O. (1999), *Autonomous evolution of gaits with the Sony quadruped robot.* Proceedings of the Genetic and Evolutionary Computation Conference. San Francisco: Morgan Kaufmann Publishers Inc.

Hornby G. S., Takamura S., Hanagata O., Fujita M. & Pollack J. (2000), *Evolution of controllers from a high-level simulator to a high dof robot.* J. Miller, editor, Evolvable Systems: from biology to hardware; In Proceedings of the third international conference (ICES 2000) Lecture Notes in Computer Science; Vol. **1801** (80-89). Berlin: Springer-Verlag.

Hornby G. S., Takamura S., Yokono J., Hanagata O., Yamamoto T. & Fujita M. (2000), *Evolving robust gaits with AIBO.* IEEE International Conference on Robotics and Automation (3040-3045).

Jakobi N. (1998), *Running across the reality gap: octopod locomotion evolved in a minimal simulation.* Phil Husbands and Jean Arcady Meyer, editors, First European Workshop on Evolutionary Robotics (39-58). Berlin: Springer-Verlag.

Jones J. L., Flynn A. M., & Sieger B. A. (1999), *Mobile Robots: Inspiration to Implementation.* Massachusetts: AK Peters.

Lewis M. A., Fagg A. H. & Solidum A. (1992), *Genetic programming approach to the construction of a neural network for control of a walking robot.* In Proceedings of the IEEE International Conference on Robotics and Automation. New York: IEEE Press.

McKerrow P. J. (1991), *Introduction to Robotics.* Wollongong: Addison-Wesley.

Nolfi S. & Floreano D. (2000), *Evolutionary Robotics: The Biology, Intelligence, and Technology of Self-Organizing Machines.* Massachusetts: The MIT Press.

Nordin J. P. (1997), *Evolutionary Program Induction of Binary Machine Code and its Applications.* Muenster, Germany: Krehl Verlag.

Nordin, J. P. & Nordahl, M. G. (1999), *An evolutionary architecture for a humanoid robot.* Proceedings of the 4th International Symposium on Artificial Life and Robotics. Oita, Japan.

Sony Corporation (2001), *AIBO.* Last visited: 27 February 2002 <http://www.aibo.com/>

Sony Corporation (2001), *SONY: News: Press Release.* Last visited: 27 February 2002 <http://www.sony.co.jp/en/SonyInfo/News/Press/200011/00-057E2/>

Wolff K. & Nordin J. P. (2001), *Evolution of Efficient Gait with Humanoids using Visual Feedback.* In Proceedings of the 2nd IEEE-RAS International Conference on Humanoid Robots, Humanoids 2001 (99-106). Waseda University, Tokyo, Japan: Institute of Electrical and Electronics Engineers, Inc.

Ziegler J., Wolff K., Nordin J. P. & Banzhaf W.(2001), *Constructing a small humanoid walking robot as a platform for the genetic evolution of walking.* In U. Rückert, J. Sitte and U. Witkowski, editors, Proceedings of the 5th International Heinz Nixdorf Symposium: Autonomous Minirobots for Research and Edutainment, AMiRE 2001 (51-59). Paderborn, Germany: Heinz Nixdorf Institute, University of Paderborn.