Evolution of Efficient Gait with Humanoids Using Visual Feedback

Krister Wolff and Peter Nordin

Department of Physical Resource Theory, Complex Systems Group
Chalmers University of Technology and Göteborg University
SE-412 96 Göteborg, Sweden
{wolff, nordin}@fy.chalmers.se

Abstract

In this paper we present the autonomous, walking humanoids Priscilla, ELVIS and ELVINA and an experiment using evolutionary adaptive systems. We also present the anthropomorphic principles behind our humanoid project and the multistage development methodology. The adaptive evolutionary system used is a steady state evolutionary strategy running on the robot's onboard computer. Individuals are evaluated and fitness scores are automatically determined using the robots onboard digital cameras and near-infrared range sensor. The experiments are performed in order to optimize a by hand developed locomotion controller. By using this system, we evolved gait patterns that locomote the robot in a straighter path and in a more robust way, than the previously manually developed gait did.

Keywords

Evolutionary Robotics, Genetic Programming.

1. Introduction

The applications of robots with human-like dimensions and motion capabilities, humanoid robots, are plentiful. Humanoid robots constitute both one of the largest potentials and one of the largest challenges in the fields of autonomous agents and intelligent robotic control. In a world where man is the standard for almost all interactions, humanoid robots have a very large potential acting in environments created for human beings [1].

In traditional robot control programming, an internal model of the system is derived and the inverse kinematics can thus be calculated. The trajectory for movement between given points in the working area of the robot is then calculated from the inverse kinematics. Even though this still is a very common approach, we propose for several reasons the concept of genetic programming for control programming of so-called bio-inspired robots [2] as e.g. a humanoid. The traditional geometric approach to robot control, based on modeling 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 [3]. Conventional industrial robots 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 too much time for reactive tasks [2]. For a robot that is conceived to operate in an actual human living environment, it is impossible for the programmer to consider all eventualities in advance. The robot is therefore required to have an adaptation mechanism that is able to cope with unexpected situations.

The anthropomorphic principle behind humanoids might be a stronger motivation factor then conventionally assumed. Consider for example the phenomenon of human left-handedness. Left-handed persons have been shown to have a shorter expected life length than right-handed persons. The standard explanation for the higher mortality rate is a higher accident frequency and the assumed explanation for this deviation is due to the fact that the world is built for right-handed people [11]. If such a minute deviation in behavior could cause accident frequencies measurable in as statistically significant mortality biases, we could expect considerable difficulties for a robot working in a human environment. The differences between human and robot will always be bigger there, than the difference between a left-handed and right-handed person. We aim at exploring and evaluating the consequences of a strong anthropomorphic principle where humanoids are built with very close correspondence with humans in terms of size, weight, geometry and motion capabilities. We have therefore devised a full-sized autonomous humanoid robot that is built around an accurate model of a human skeleton –the Priscilla robot.

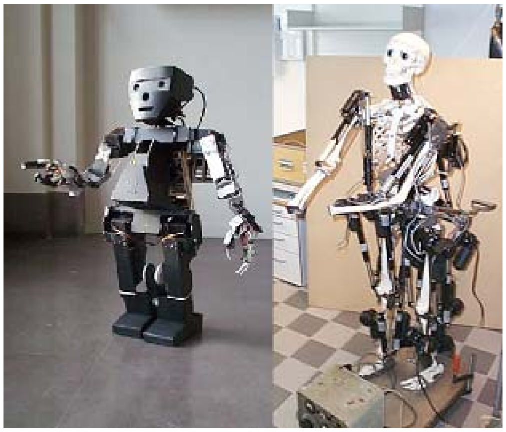

Figure 1. ELVIS (left) and Priscilla (right).

The skeleton design guarantees anthropomorphic geometry and enables close correspondence in movement capabilities.

From a methodological and developmental standpoint are the project guided by more than the anthropomorphic principle. Even though we have simulators for the Priscilla robot we strongly believe in the embodiment principle and we try to make most of our experiments on the full-size autonomous Priscilla robot. However, the over-all efficiency of our humanoid project has turned out to increase when using several smaller size prototypes. The mid-size prototype is called ELVIS and it is about 60 cm tall. The smallest size humanoid is the ELVINA type, about 25 cm tall. Two instances of ELVINA have been built to enable experiments with cooperating humanoids. In this paper we briefly introduce three humanoid robot prototypes.

A third guiding principle is the need for adaptivity when dealing with such a complex object as a humanoid in such a complex environment as everyday human life. We are furthermore using evolutionary algorithms and more specifically genetic programming as the adaptation method. Genetic programming is an efficient method for breeding symbolic structures such as computer programs and behavior definitions [4].

We present work in this paper evolving a gait pattern, using genetic programming and especially evolutionary strategies [4]. 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. Anytime learning can make use of evolutionary computation in a learning module for the robot to adapt to changes in the robot's capabilities without the use of internal sensors [5]. 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 [6]. This was also compared with a controller that was evolved with a real octopod robot [7]. 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 [1] quadruped robot dog [8]. The collisions between the robot and ball had different results in the real world than in the simulated world, however it did not affect ball-chasing performance. When a high degree of accuracy 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 do it many times faster.

The first attempt in using a real, physical robot to evolve gait patterns was made at the University of Southern California [9]. 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 [10] and [13]. These results show that evolutionary algorithms can be used on complex, physical robots to evolve non-trivial behaviors on those robots. In previous evolution with physical robots has a humanoid, biped robot not been used.

Our test problem is that of developing locomotion controllers for static gaits for our biped 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 [3]. 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

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

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 polygon. For a biped robot, the area of the support polygon is relatively small, compared to the altitude of where its center of mass is located. The corresponding measure for a robot that got four or more legs is relatively larger. 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.

2. Robot Platforms

All the three humanoids [2] carry their main computer power onboard. ELVIS and Priscilla have a small PC laptop, while ELVINA has the EyeBot MK3 controller onboard, carrying it as a backpack. The EyeBot MK3[3] consists of a 32-bit micro-controller board with a graphics display and four push buttons for user input.

2.1 Priscilla

The Priscilla robot consists of a plastic skeleton with titanium reinforcements and linear electric actuators.

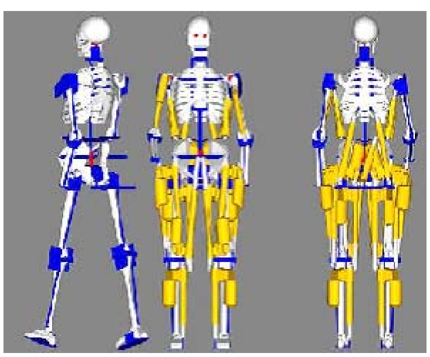

Figure 2. CAD-drawings of Priscilla showing skeleton, titanium reinforcements and actuators.

Linear electric actuators are more accurate and therefore easier to control than pneumatic actuators, which was also considered as an option when designing the robot. Even though pneumatic actuators are more powerful in an autonomous setting, linear electric actuators were chosen for the Priscilla humanoid.

One goal was to make the robot strong and fast enough to be able to walk with normal human walking speed. There are also a number of movements that the robot should be able to carry out when speed is not critical. That could be lifting the arm when holding an object in its hand, or rising from a chair. Such movements dictate other constraints on the actuators.

Once the requirements were defined, suitable actuators were chosen. For the legs Linak La30 actuators with different strokes is used and for the arms Warner Electric La1 actuators. For head movement we use standard off-the-shelf R/C servomotors as actuators, because of their convenience in connection to computer for control.

Figure 3. Picture of the actuators and servo. From top to bottom La30, La1 with stroke 0.115m and 0.06m respectively. To the right the R/C-servo.

The La 30 actuator is equipped with a passive brake that is activated automatically when the motor stops. This prevents the robot from consuming energy when not moving. The smaller La1 actuator is not available with brake. Both the La30 and La1 actuator have built-in potentiometers that provides us with the possibility to get accurate readings within the controller software of the state of each actuator at a certain time. The actuators weights, depending on the stroke, from about 1.0kg to 2.5kg. The weight is the main reason why not even more powerful actuators are used for the Priscilla humanoid.

2.2 Elvis

ELVIS is a scale model of a full-size humanoid with a height of about 60cm, built with 42 servos giving a high degree of freedom in legs, arms and

[2] http://humanoid.chalmers.se
[3] http://www.ee.uwa.edu.au/~braunl/eyebot/

hands. Microphones, cameras and touch sensors guide the robot. The imminent goals are to walk upright and to navigate through vision serving a prototype for Priscilla. Seven on board micro-controllers control the servos and sensors. ELVIS is autonomous, with onboard power supply and main processing unit, but many experiments are mainly performed with connection to a host computer. ELVIS can for instance walk fully autonomously.

2.3 Elvina

ELVINA is a simplified, scaled model of a full-size humanoid with body dimensions that mirrors the dimensions of a human [15]. The ELVINA humanoid is a fully autonomous robot with onboard power supply and computer, but many experiments are performed with external power supply. It is 28cm tall and it weights about 1490g 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.

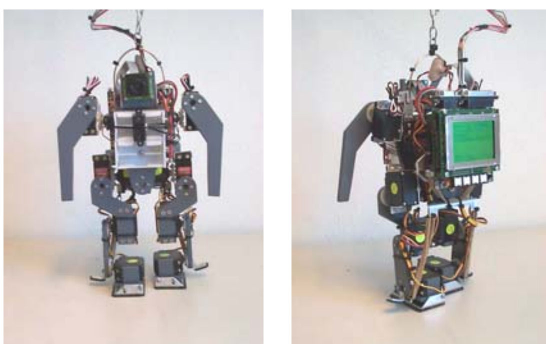

Figure 4. Pictures of the ELVINA Humanoid. Camera and PSD sensor visible (left) and controller board (right).

Vision is the most important sensor of this robot. Therefore, it is equipped with full color 24 bit digital camera, which is based on CMOS technology. The camera is directly connected to the controller board and mounted in the robot's head. The body also houses a near-infrared PSD (position sensitive detector) which is used to determine distances to nearby objects. In its present status, the robot is capable of static walking.

A single camera cannot be used to accurately measure the distance to a nearby object. This is instead achieved with the 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, which means that the sensor is relatively insensitive to the texture and color of the object at which it is pointed [14].

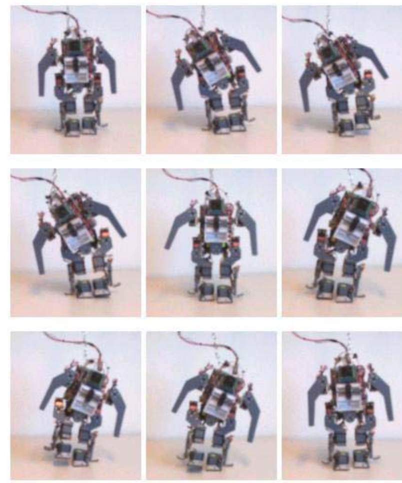

Figure 5. Series of pictures showing a complete gait cycle, from top left to bottom right.

In order to control the movements of a limb, the partial movements of all involved joints must be coordinated and synchronized to get the desired motion. For this reason, a servo locomotion module has been developed. The idea behind it is that twelve integer-valued vectors specify a complete gait cycle, each of the vectors specifying given positions of the robot's limbs. Each of the vectors consists in fact of a set of control parameters, a position value for each actuator and two time constants. The first of the vectors in the set correspond to the robot's initial position and the second vector corresponds to the second position of the robot and so on. By interpolation from the values of one vector to the values of another vector the robot's limbs is caused to smoothly move from one position to another position. The first time constant specifies how fast the limbs should move between consecutive positions and the other constant specifies time delay before the position of the limbs is updated. To obtain a complete gait cycle the set of vectors specifying it is interpolated once and the robot is made to continuously walk by iteration. All robot control programs that we have developed are implemented in C language.

3. Architecture

The philosophy behind all our robots is that the software architecture should mainly build on evolutionary algorithms and specifically genetic programming. Evolution is thus used to induce programs, functions and symbolic rules for all levels of control. Three hierarchical layers are used for control. Those are the reactive, the model building and the reasoning layer.

3.1 Reactive Layer

The first layer is a reactive layer based on on-line evolution of machine code. This method assumes that all fitness feedback is obtained directly from the actual robot. The disadvantage is that the GP individuals spend most of their time waiting for feedback from the physical environment. This results in moderate learning speed, and the constant movement shortens the life span of the hardware. The benefit of the method is its simplicity, and that the only constraints needed for the models being learned are that they should fulfil their task as a black box. This layer is used for reactive behaviors such as balancing.

3.2 Model Building Layer

To achieve higher learning speeds and more generic behaviour there is a second control layer that works with memories of past events. In this genetic reinforcement-learning framework, the system tries to evolve a model of the underlying hardware system and problem. The model maps sensor inputs and actions to a predicted goodness or fitness value. The currently best model is then used to decide what action results in optimal predicted fitness given current sensor inputs. This layer allows the genetic programming system to run at full speed without having to wait for feedback from the environment; instead it fits the programs to memories of past events. The machine code genetic programming approach used is called Automatic Induction of Machine Code GP, AIMGP [12]. AIMGP is about 40 times faster than conventional GP systems due to the absence of any interpreting steps. In addition, the system is compact, which is beneficial when working on board a real robot. The model-building layer is also used for basic control tasks.

3.3 Reasoning Layer

The third layer is a symbolic processing layer for higher "brain functions" requiring reasoning. The objective of this layer is to handle high level tasks such as navigation, safety and energy supply. This layer is built on "genetic reasoning", a method where evolution is used as an inference engine, requiring less heuristics to guide the inference procedure [12].

Each of these layers consists of modules for various tasks such as balancing, walking and image processing. Some system functions are represented as several modules spanning different layers.

4. Experiments

Several experiments has been performed on this architecture:

- Balancing, evolution of functions for balance
- Walking, evolution of efficient walking, described further in this paper
- Vision, evolution of 3-D vision
- Navigation, evolution of plans
- Audio orientation, evolution of stereo hearing
- Manipulation, evolution of eye hand co-ordination

In this section we describe an experiment with evolution of efficient walking.

4.1 Evolutionary Algorithm

The evolutionary algorithm used is a steady state evolutionary strategy [4], running on the robot's onboard computer. A population that stems from a manually developed individual is created with a uniform distribution over a given search range. Then four individuals are randomly selected from this initial population. These individuals are evaluated and their fitness is measured. The two individuals with the better fitness values are considered as parents and the two individuals with the lower fitness are replaced by the offspring of the parent individuals. The selection, evaluation and reproduction phases of the evolutionary strategy is then repeated until the maximum number of trials is reached.

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. The search ranges are set to suitable values in order to produce a sufficient amount of individuals in the population that are capable of good performance in the evaluation.

A tournament selection is used 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 generations a certain individual can be selected to be in the tournament 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

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

where c_{ki} is the ith gene of the kth child individual, p_{ki} is the ith gene of the kth parent individual, p_{1i} and p_{2i} are the ith 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

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

where $c_{ki,mutate}$ is the mutated ith gene of the kth 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.

4.2 Experimental method

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

The robot is placed on top of a table with a surface of relatively low friction during the evolution. A target wall of 50cm 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 (65cm above the table surface) there is a horizontal beam, used as a carrier for the power supply cables.

Each individual evaluates under as equal conditions as possible. The robot's starting position is at a distance of about 40cm from the wall and facing it. The experimenter centers the robot according to the black line by using its onboard camera. Once centered, the robot measures its distance with the PSD infrared range sensor and starts to locomote 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 line on the wall. Using these measurements and the time required for the locomotion trial, it calculate 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.

The primary task for the onboard camera is to provide a precise tool for determination of direction. Initially, the camera is set to continuos image mode, whereas the frames are put to the EyeBot's LCD screen and thus made visible to the experimenter. The robot is considered as centered when the image of the black line appears near the center of the LCD display. A single snapshot is then analyzed by the software's image processing routines [16] to precisely determine the robot's position relative the black stripe. This measure is then stored for later use.

After an individual has performed a trial, the camera is again used to determine how straight the robot moved during the trial. As the robot body remains fixed the camera pans from left to right.

When the black line appears on the LCD display, the camera stops moving and the difference between this value and the earlier obtained measure is considered as the angular deviation θ from the desired (straight) path of locomotion.

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.

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

$$score = v\left(d_0, d_f, t\right) \times s\left(\theta, d_f\right)$$

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 the 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

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

Here, f (θ) is a normalization function to convert θ into a 0 – 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 strip - 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

$$v\left(d_0, d_f, t\right) = \frac{d_0 - d_f}{t}$$

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 zero fitness score.

4.3 Results

For this evolutionary strategies experiment we used an initial population of 30 individuals and ran for nine generations. 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. 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. The qualities of different individuals were also tested in other ways than direct fitness measuring. To evaluate one generation, consisting of four individuals, took approximately 30 minutes in this experiment.

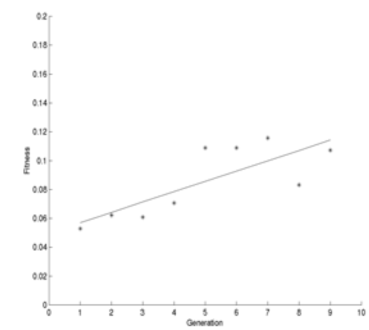

Figure 6. Plot of the average fitness scores

The above figure shows the average fitness scores for each generation 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.

4.4 Discussion

Evolving efficient gaits with real physical hardware is a challenging task. That is because the mechanical structure of the robot is non-rigid. When moving a limb (e.g. a leg), the trajectory, thus the limb's final position, is affected by from which position the movement started. How much the torso leans also affects the resulting position of the robot. The most vulnerable parts of the robot were proved to be the knee servos. Both these servos were replaced tree times. The torso and both the ankle actuators were exchanged once as well as the two hip servos.

5. Conclusions

The work presented in this paper constitutes of two main parts, the construction of a series of humanoid walking robots and a genetic programming experiment performed on the humanoids.

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 a genetic programming 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.

References

[1] P. Nordin and M. G. Nordahl (1999). An evolutionary architecture for a humanoid robot. Proceedings of the Fourth International Symposium on Artificial Life and Robotics (AROB 4th 99). Oita, Japan.

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

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

[4] W. Banzhaf, P. Nordin, R. E. Keller and F. D. Francone (1998). Genetic Programming~ An Introduction: On the Automatic Evolution of Computer Programs and Its Applications. San Francisco: Morgan Kaufmann Publishers, Inc. Heidelberg: dpunkt verlag.

[5] G. B. Parker and J. W. Mills (1999). Adaptive hexapod gait control using anytime learning with fitness biasing. Proceedings of the Genetic and Evolutionary Computation Conference (GECCO '99) (pp. 519-524). San Francisco: Morgan Kaufmann Publishers, Inc.

[6] N. Jakobi (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 (pp. 39—58). Berlin: Springer-Verlag.

[7] T. Gomi and K. Ide (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.

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

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

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

[11] M. S. Gazzaniga (1999). The New Cognitive Neurosciences. MIT Press, ISBN: 0262071959

[12] J. P. Nordin (1997), Evolutionary Program Induction of Binary Machine Code and its Application. Krehl Verlag, Muenster, Germany

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

[14] J. L. Jones, A. M. Flynn and B. A. Sieger (1999). Mobile Robots: Inspiration to Implementation. Massachusetts: AK Peters.

[15] J. Ziegler, K. Wolff, P. Nordin and W. Banzhaf (2001). Constructing a small humanoid walking robot as a platform for the genetic evolution of walking. Proceedings of the 1st International Conference on Autonomous Minirobots for Research and Edutainment, AMiRE 2001. Paderborn, Germany: The Heinz Nixdorf Institute..

[16] P. J. McKerrow (1991). Introduction to Robotics. Wollongong: Addison-Wesley.