Gait Evolution for Humanoid Robot in a Physically Simulated Environment

Nesrine Ouannes[1], NourEddine Djedi[1], Yves Duthen[2], and Hervé Luga[2]

[1] LESIA Laboratory/ Med Khider Biskra University, Departement of computer science, BP 145 RP Biskra, Algeria
{nesrine.ouannes, noureddine.djedi}@lesialab.net
[2] Institut de Recherche en Informatique de Toulouse, Université de Toulouse - CNRS - UMR 5505 Toulouse, France
{Yves.Duthen,Herve.Luga}@univ-tlse1.fr

Abstract. This article describes a bio-inspired system and the associated series of experiments, for the evolution of walking behavior in a simulated humanoid robot. A previous study has demonstrated the potential of this approach for evolving controllers based on simulated humanoid robots with a restricted range of movements. The development of anthropomorphic bipedal locomotion is addressed by means of artificial evolution using a genetic algorithm. The proposed task is investigated using full rigid-body dynamics simulation of a bipedal robot with 15 degrees of freedom. Stable bipedal gait with a velocity of 0.94 m/s is realized. Locomotion controllers are evolved from scratch, for example neither does the evolved controller have any a priori knowledge on how to walk, nor does it have any information about the kinematics structure of the robot. Instead, locomotion control is achieved based on intensive use of sensory information. In this work, the emergence of non-trivial walking behaviors is entirely due to evolution.

1 Introduction and Motivation

Simulation of human motion has applications in several different fields like, entertainment industry, education, science visualization, architecture and medicine. Different research areas, with different aims, are involved in the study of the human motion to understand the fundamental dynamics and its control mechanisms. The human body comprises 206 bones, over 600 muscles and is controlled by a complex nervous system. The human motion is the result of 92 degrees of freedom. Researchers in biomechanics, robotics, and computer science work to understand human natural motion in order to reproduce it artificially.

The aim of humanoid robotic researchers is to obtain robots that can imitate the human behavior to collaborate, in the best way, with humans. Building a complete humanoid robot is a very complex task and researchers usually prefer solving simpler problems as the study of the biped robots, artificial hands, vision, questions about high level control etc.

D. Plemenos and G. Miaoulis (Eds.): Intelligent Comp. Graphics 2011, SCI 374, pp. 157–173.
springerlink.com

There are numerous application areas for robots with anthropomorphic shape and motion capabilities. In a world where man is the standard for almost all interactions, such robots have a very large potential acting in environments created for people. They can function in certain areas which are not accessible for wheeled robots, such as stairways or uneven terrain. Furthermore, robots capable of bipedal locomotion have the ability to interact with the environment using the whole body, and climb over large obstacles in their path as opposed to wheeled vehicles. The major drawback of legged robots is the challenge of creating controllers for them. The problem is complicated because of the number of degrees of freedom in each leg and because of changes in the body center of mass and momentum.

In the research literature, and on topics related to bipedal walking, the terms stability, equilibrium, and balance are often used inter-changeably. Throughout this paper, we will use the following notions in order to avoid confusion; the term stability refers to a system which could be analytically treated as stationary (that is the static case), whereas for a non-stationary system (the dynamic case), the terms balance and equilibrium are used.

An obvious problem confronting humanoid robotics is the generation of stable and efficient gaits. Whereas wheeled robots normally are statically balanced and remain upright regardless of the torques applied to the wheels, a bipedal robot must be actively balanced, particularly if it is to execute a human-like, dynamic gait. The success of gait generation methods based on classical control theory, such as the zero-moment point (ZMP) method [23], relies on the calculation of reference trajectories for the robot to follow.

In order to address this problem, alternative, biologically inspired control methods have been proposed, which do not require the specification of reference trajectories. The aim of this paper is to describe one such method, based on recurrent neural networks (RNNs), for control of bipedal robot.

2 Related Work

Various methods were proposed to generate bipedal robot locomotion using evolutionary techniques. Peterson [18] has reported on the development of a method for generating walking behaviors for bipedal robots. An adaptation of evolutionary programming (EP) to the case of nite state machines (FSMs) is used to operate on both the structure and the parameters of the robotic brain. The method has been demonstrated on a simplied ve link robot, constrained to move in the sagittal plane. Two test cases were used; energy optimization and robust balancing. Typically, robotics researchers employ bio-inspired control strategies based on articial neural networks (ANNs) [14, 26] or central pattern generators (CPGs) [5]. Often some kind of evolutionary algorithm (EA) is utilized for the design of the controller [18, 19, 2, 27], and [28].

Of course, the proposed technique would have a greater impact if demonstrated in a 3-dimensional rigid-body simulation instead. There exist, however, some examples in the research literature of synthesizing locomotion for full rigid-body simulated bipeds, by incorporating EAs.

Evolutionary computation, frequently involving the evolution of neural network controllers, has been successfully used to the automation of the process of gait creation [12, 11, 8, 20, 25, 6, 24, 15].

In recent years Jeff Clune [4] demonstrates that HyperNEAT, a new and promising generative encoding for evolving neural networks, can evolve quadruped gaits.

Recently [1] Amin Azarbadegan describes the design of an approach to evolve Simss creatures with morphology and behavior similar to biped animals, his hypothesis is that biases in morphology that encourage limb specialisation, combined with rewards for successful locomotion and carrying at the same time and realistic, physics-based penalties for falling in fitness function, would lead to creatures capable of bipedal locomotion.

He evolved bipedalism by incorporating physical damage and incentives for upright locomotion. The reward for carrying is reflected in the components of the fitness function involving keeping the head up, limiting the number of limbs and making two limbs exempt from damage.

In this paper we propose an approach that uses evolutionary techniques with neural network to evolve a controller for an humanoid robot that is physically simulated.

3 Models and Methods

3.1 Humanoid Robot Model

he humanoid robot model used here was created from the body and joint primitives available in ODE[1] . simulation package. To build the kinematical model, the geometrical data of the robot (link lengths, type and position of joints etc.) are needed.

A physically-based model of legged locomotion describes the nonlinear relationships between the forces and the moments acting at each joint and the feet etc., and the position, velocity and acceleration of each joint angle. In addition to the geometrical data, a dynamics model requires kinematical data as mass, centre of mass and inertia matrix for each link and joint, max/min motor torques and joint velocities which are difficult to obtain and are often an overlooked source of simulation inaccuracy. To simulate interaction with the environment detection and handling of collisions as well as suitable models of foot-ground contacts are required. In the context of simulation of autonomous robots for research purposes often uses the open source Open Dynamics Engine (ODE). ODE can handle collision detection for several geometric primitives.

The biped model used here is a fully tree-dimensional bipedal robot with 15 degrees of freedom as shown in the right most panel of fig.1. The robot model consists of 12 rigid-body parts, and 11 ODE joints. There are 1 fixed joint, 8 hinge joints and 2 Ball/socket joints used to connect the rigid-body parts into an articulated rigid-body structure (hinge, fixed and ball/socket are the internal names of specific joint types in ODE). The rigid-body primitives used are 8 capped cylinders, 3 rectangular boxes and one sphere. The robots structure is defined using multiple chains,

[1] Open Dynamic Engine.

Table 1 Body parameters of the humanoid robot

Body part	Geometry	Dimension (m)
Head	Sphere	Radius:0.188
Arm	Caped cylinder	0.14× 0.25× 0.44
Torso	Rectangular box	0.90× 0.25× 1.00
Thigh	Caped cylinder	0.20× 0.25× 0.70
Shank	Caped cylinder	0.20× 0.25× 0.70
Foot	Rectangular box	0.40× 0.50× 0.10

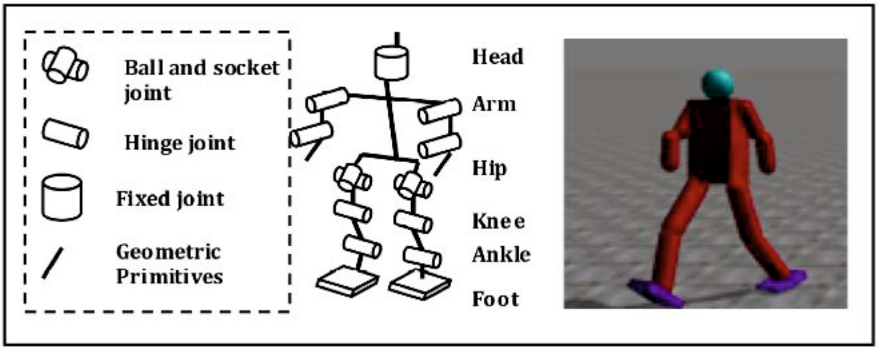

Fig. 1 I3-D biped model used in our simulation

starting from its feet with each link described in terms of the previous links. This composition results in a 15 degree of freedoms (DOFs) bipedal model.

3.2 Locomotion Control Using Recurrent Neural Network

A practical biped needs to be more like a human:

- Capable of learning new gaits when presented with unknown terrain. In this sense, it seems essential to combine force control techniques with more advanced algorithms such as adaptive and learning strategies. Conventional control algorithms for humanoid robots can run into some problems related to mathematical tractability, optimization, limited extendibility and limited bio-logical plausibility. The presented intelligent control techniques has the potential to overcome the aforementioned limitations;
- Furthermore, some researchers have begun considering the use of neural networks for control of humanoid walking [5, 17, 16, 14, 26]. This approach makes possible the learning of new gaits that are not weighted combinations of predened bipedal gaits. Various types of neural networks are used to generate walking behaviors and control design of humanoid robots such as multilayer perceptron, CMAC (Cerebellar Model Arithmetic Controller) networks, recurrent

neural network, RBF (Radial Basis Functions) networks or Hopeld networks, which are trained either by supervised or unsupervised (reinforced) learning methods. The majority of the proposed control algorithms have been veried by simulation, while there were few experimental verications on real biped and humanoid robots. Neural networks have been used, as efficient tools for the synthesis and off-line and on-line adaptation of biped walk. Another important role of connectionist systems in controlling of humanoid robots has been their ability to solve static and dynamic balance during the process of walking and running on terrain with different environmental characteristics.

3.2.1 Recurrent Neural Network (RNN)

An RNN is an artificial neural network [9] consisting of a number of neurons (nodes) with arbitrary connections (including self-coupling of individual neurons) (see figure 2). This network can operate either in discrete time as common in feed-forward networks (i.e. ANNs without feedback connections), or in continuous time. In the later case, using a simple neuron model, the dynamical behavior of the ith node in the network is governed by the equation:

$$\tau_i + \gamma_i = \sigma(\beta_i + \omega_{i\varphi}\gamma_i + \omega^I_{i\varphi} I\varphi)._i = 1, 2, ..., v \tag{1}$$

Where v is the number of neurons in the network, τ_i are time constants, γ_i is the output (activity) of node φ, $\omega_{i\varphi}$ is the (synaptic) weight connecting node v to $node_i$, $\omega^I_{i\varphi}$ is the weight connecting in-put node v to $node_i$, $I\varphi$ is the $v^t h$ external input to $node_i$, and β_i is the bias term, which determines the output of the neuron in the absence of inputs. $\sigma()$ is a sigmoid function whose main purpose is to restrict the activity of the neurons to a given range. The hidden, context, and output layers of the RNN all use the same bipolar sigmoid activation function (see equation 2 and fig. 3).

$$\sigma(c) = 2/(1 + e^{-\alpha c}) - 1. \tag{2}$$

The obvious advantage an RNN has over the traditional feed for-ward network is ”memory.” The use of feedback connections allows the RNN to have a ”memory” of past events. Thus, pattern presentation to the RNN will take into consideration what moment in time the pattern occurs. Biological neural networks process information in a similar fashion to the RNN.

The humanoid robot presented in this paper uses Elman (1990) [7] Recurrent Neural Network for its biological plausibility and powerful memory capabilities. Furthermore, biological neural net-works do not make use of back propagation for learning. Instead we use evolutionary algorithms to evolve locomotive behaviors.

In difficult real-world learning tasks, such as controlling robots, it is intractable to specify correct actions for each situation. In a complex control system, such as that used by the humanoid robot, specifying the correct outputs for each possible input combination and state is practically impossible. In these situations, optimal behavior must be learned by the exploration. In that case the reinforcement of good

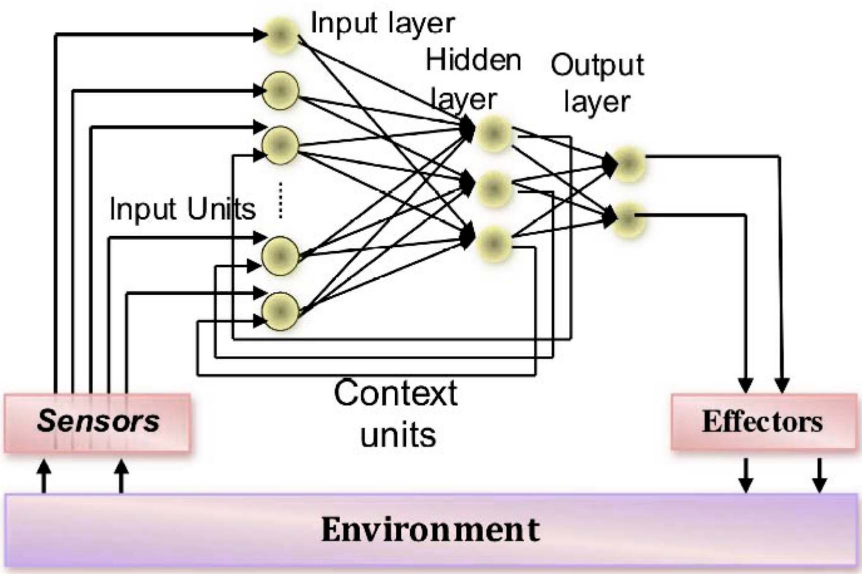

Fig. 2 The sigmoid bipolar function

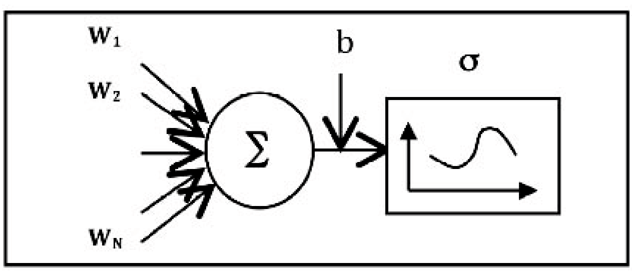

Fig. 3 The sigmoid bipolar function

decisions relies on some feedback from the system itself and the exploitation of learned knowledge from the environment. Genetic algorithms can be used as an optimization process to evolve neural networks that prove to be robust solutions to difficult real-world learning tasks without the need to supply additional information or for an external agent to direct the process.

3.3 The Model of RNN

The humanoid robot uses a set of sensors to collect data from the environment and feeds it to the RNN. Sensors monitor the internal state of the robot, such as joint angles are referred to as proprioceptive sensors. In this setting, the current joint angles of the previous time step of the simulation are used by the evolved controller to compute the next set of motor signals for the robot.

Simulating a biped robot in a realistic environment most likely re-quires feedback loops between the robot's control system and the robot's body, as well as between the control system and the environment.

The set of external sensors constitute the robot's ”window” upon the environment. Those sensors can measure values such as the robot acceleration or inclination relative to a fixed coordinate frame, light intensity, external forces applied to the robots body, etc. The arrangement of the sensors is shown in figure 3. In this figure, the humanoid robot has 15 sensors located throughout its body. The contact sensors indicate when the feet make contact with the ground plane, and the angle sensors measure the angle of each hinge joint of the robot body. The Ball and socket joints also contain angular velocity sensors that feed the rate of angular change back to the RNN. And finally, a direction sensor provides the ANN with a virtual compass.

3.4 Evolutionary Optimization of RNNs

In the standard genetic algorithm (GA) [10], which is one example of an EA, the variables of the problem are encoded in a xed-length string. By contrast, the EA used here acts directly on the RNNs.

The topology of our network is made up of 4 layers: an input layer, a hidden layer, an output layer and a context layer. The number of neurons contained in the input and output layer depends on the robots morphology (i.e. the number of sensors and effectors). Each in-terneuron connection within the RNN is assigned a weight.

Real-number encoding is used, i.e., all genes take oating point values in the open interval [0,1], which are then rescaled to the appropriate range.

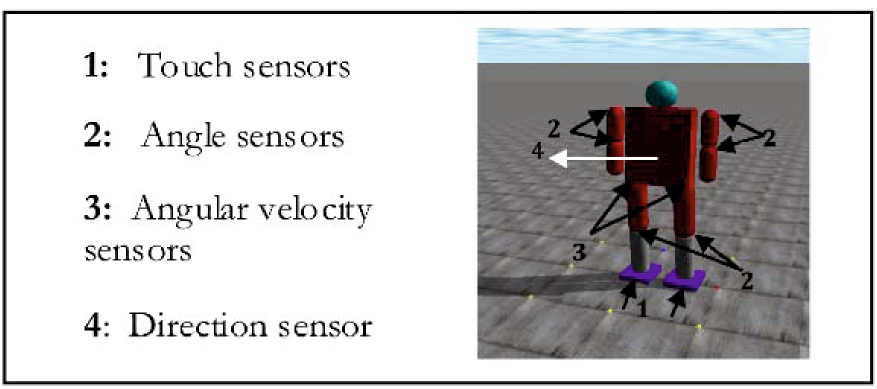

Fig. 4 Sensor arrangements

3.4.1 Genetic Algorithm

Genetic algorithms are a class of search and optimization heuristics inspired by Darwins theory of evolution. They operate on a population of individuals (potential solutions to a problem), updating the population in parallel, over many generations. With its mechanism, the genetic algorithm has proven to be a useful tool with large search spaces (such as, for example, to find the best parameters for locomotion). In genetic algorithms, the Fitness Function is of extreme importance: it is the operator that, during the evolution, evaluates all the individuals (their phenotypes) of the current generation. The concept of ”the best individual” depends on the problem. So, in the case of this work, a good phenotype is a set of parameters that causes a character to perform the desired movement. In a GA, the best individuals will be more likely to pass their genes to the individuals in the next generation. The fitness function drives the evolution and its definition is crucial for the algorithms performance: with a poorly designed fitness function the GA might miss good solutions. In this work, the purpose of the genetic algorithm is to optimize the weights of the neural network which controls the humanoid robot. A synergistic relationship exists between the GA and the RNN. The GA optimizes the RNN, and the RNN produces robot behavior that is then scored. This feedback will then drive the GAs population to converge to an optimum. At start-up, the population's chromosomes are initialized at random. The chromosome length is 1130 genes, with 1 gene per RNN weight. The number of connections represents the number of genes in the chromosome; a floating-point number represents each gene. To avoid the problem of premature convergence, a linear fitness function is used in this paper. The GA used here makes use of the standard single-point crossover operator. After the crossover operation, the gene has a probability of being mutated. The mutation operator uses a Gaussian perturbation rather than a random mutation. By perturbing the weights rather than randomly selecting values for the mutated weights, we favored gradual change. Table II presents the static parameters used for the GA.

3.4.2 The Fitness Function

The fitness function determines how an individual is rated in terms of genetic fitness, and indirectly influences the agent behaviors. The fitness function was carefully chosen such that it would tend to award efficient locomotive behaviors and penalize wasted effort. It is also based on the distance travelled by the robot within a certain period of time. A higher fitness score is awarded to humanoid robots that are able to travel larger distances in a given amount of time (10 seconds).

The reward is the Euclidean distance between the robots center of mass at its initial position and when it eventually fell.

4 Experiments

The environment in which evolution occurs is extremely important to the final results. The simulated environment imposes similar constraints on the virtual humanoid robot as the natural environment would on a real humanoid robot replica. It

Table 2 Genetic Algorithm Parameters

Parameters	Value
Population size	100
Elitism	20 %
Crossover Rate	70 %
Genomic Mutation Rate	1%
Selection type	Roulette wheel selection
Chromosome length	1130 (45 neurons)
The number of generations	Up to 200

is likely that any real humanoid robot counterpart would encounter "obstacles" in the natural environment. Thus, in order to allow the humanoid robot to learn to cope with obstacles, they should be modelled within the virtual environment. A complex environment with obstacles that the humanoid robot will need to know in order to avoid (or make use of) may provide for robust evolved behaviors.

4.1 The Software

The software is coded in C++ without parallelization and the experiments were carried out on a desktop workstation. The creatures 3D environment and physics are simulated by Russel Smiths Open Dynamics Engine version 0.11.2[2] .

4.2 Robot Morphology

For our simulation, we used a robot humanoid constrained to move on a flat surface. The robot has 15 degrees of freedom: torques can be applied at both knee joints and at both hip joints. In addition, an additional actuator controls the posture of the upper body.

4.3 Physics Parameters

Different character parameters have been used. To simulate a structure in a physical way, there are parameters like bodies mass, bodies dimension, bodies centre of mass offset, frictions, joints limits and maximum forces applicable by joints that are very important to obtain realistic movements. Some parameters are related because, for example, the bodies mass, dimension and COM (Centre of mass) offset affect the maximum forces applicable by joints. The physics engine simulates a continuous world in an approximate and discrete way. Setting the max force applicable by joints, it is important to re-member that the physics engine is not always accurate. For example greater forces and torques, cause greater errors for accumulate

[2] Available at: `http://ode.org/`

during the physics simulation. This can happen because the physics engine manipulates quantities that span an increasing number of orders of magnitude with fixed-precision numbers. The physics engine errors can implicate the violation of the movements limits of a joint, and thus incurring, unrealistic motions.

Static and dynamic frictions are other important parameters to generate a realistic motion. The friction values are related to the other parameters because, naturally, forces applicable and masses affect the bodys reaction during the motion. Using little values implicates more difficulty to achieve a goal because the character cannot obtain the right push to walk. The tests about different friction values have been useful to understand better how several ground specifications can affect the human motion. However, you can choose what material the robot and the ground are made of, and look up the corresponding static/dynamic friction parameters.

5 Results

We here show the ability of our system to evolve a controller of a humanoid robot in several situations. Two test cases were used: locomotion control on flat terrain and obstacle avoidance. The two experiments have been recorded on videos[3] .

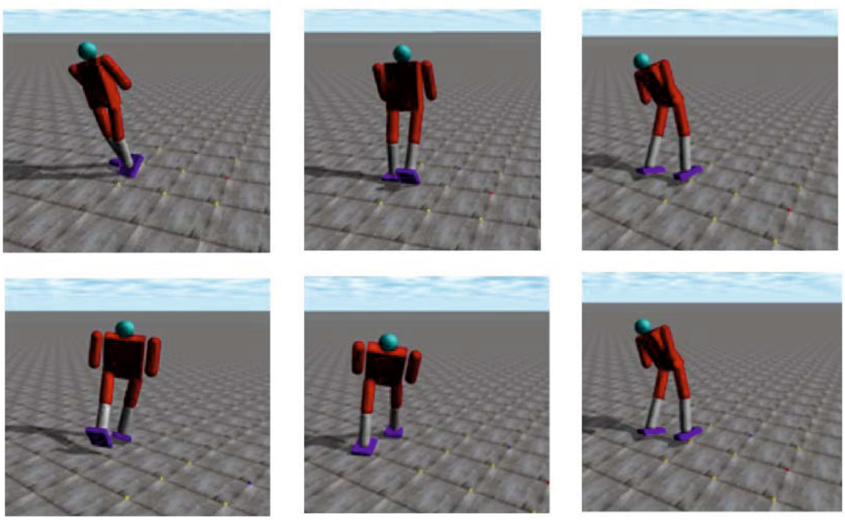

Fig. 5 A sequence of humanoid robot that is tryng to walk

[3] A video of our results can be viewed at: `http://siva.univbiskra.net/Demos/humanoid_robot_project.wmv`

5.1 Results of the Locomotion on Flat Terrain

Figure.5 shows some attempts to keep the robots balance, these images are from early generations so the movement is not displacement on this stage, because the robot is only moving on its place.

As fig. 6 shows, the robot can successfully walk on the flat terrain by periodically and alternatively stepping forward. However, it is observed that this evolved controller is unable to cope with slops. Note that at this stage, only the contact sensors were used in the feet.

Fig.7. Shows the oscillations of the hip-joint angle, knee-joint angle, ankle-joint angle and arm-joint angle of the left and right legs of the humanoid robot during locomotion relative a simulation period of 10 seconds.

The plots of each leg joint are them from the top individual in the best generation (that is generation 229).

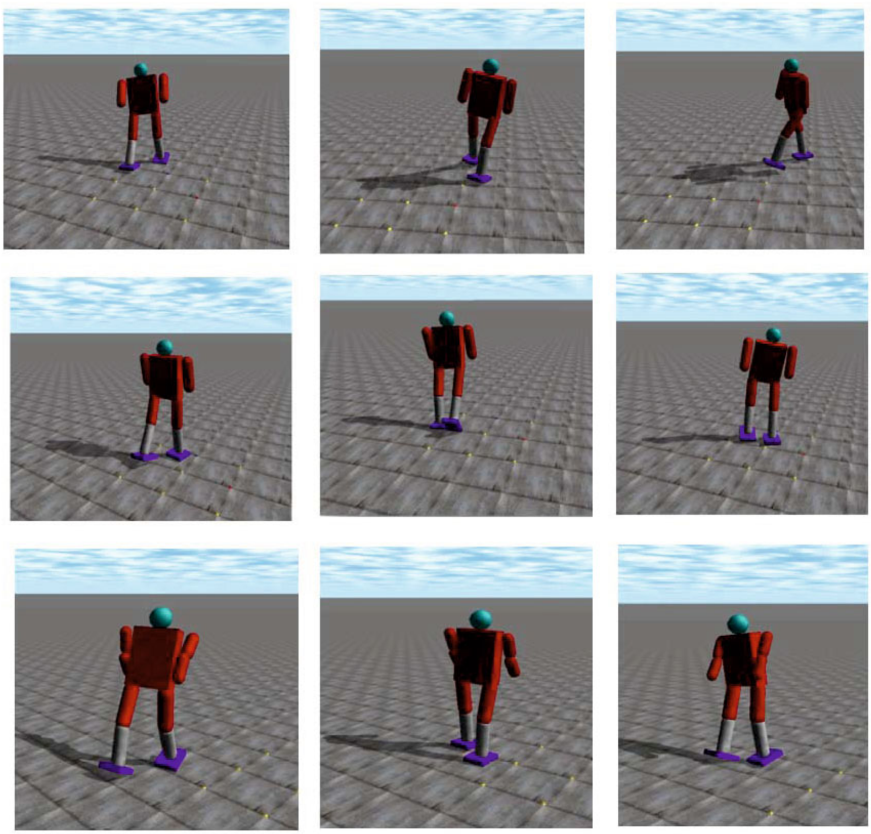

Fig. 6 The best evolved gait of humanoid robot that moves from right to left. This series of moves would be repeated over and over in a stable, natural gait.

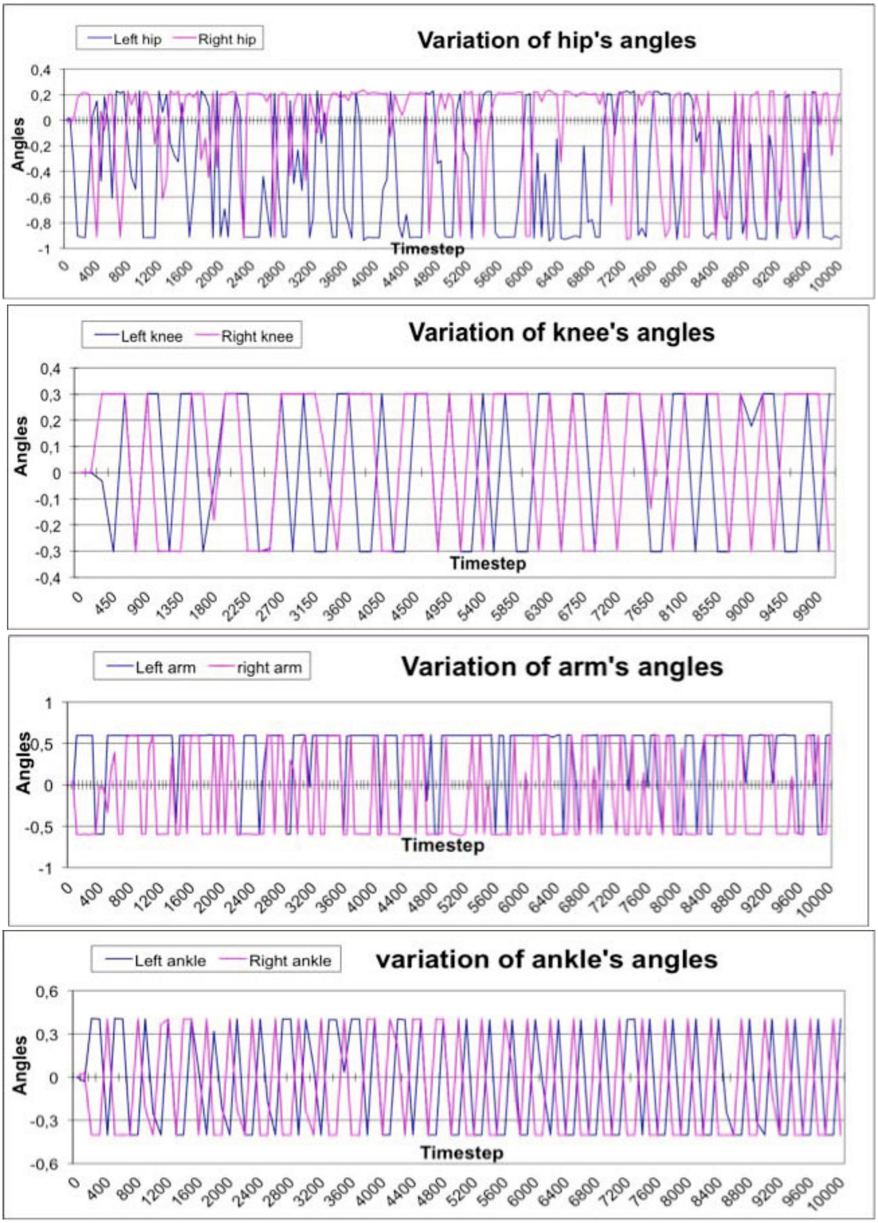

Fig. 7 Motor neuron activation levels of top individual in generation 229

By contrast, the other graphs represent symmetrical oscillations of the angles of the right and left feet. In general, we notice in the four graphs the oscillations from the right foot are almost opposite of those from the left foot

These graphs show that the robot succeeded in moving while advancing step by step in a non-periodic way (i.e. not simultaneously). This is due to the fact that in certain levels, the right foot is positioned according to its maximum angle but the left foots angle is on zero (null) value or the minimal value. This can be seen especially from the first graph of the variations of hips angles where the robot does not generate completely symmetrical movements. However, starting from the seventh second of the simulation, the movements of the hips become symmetrical.

The best evolved gait of humanoid robot that moves from right to left. This series of moves would be repeated over and over in a stable, natural gait.

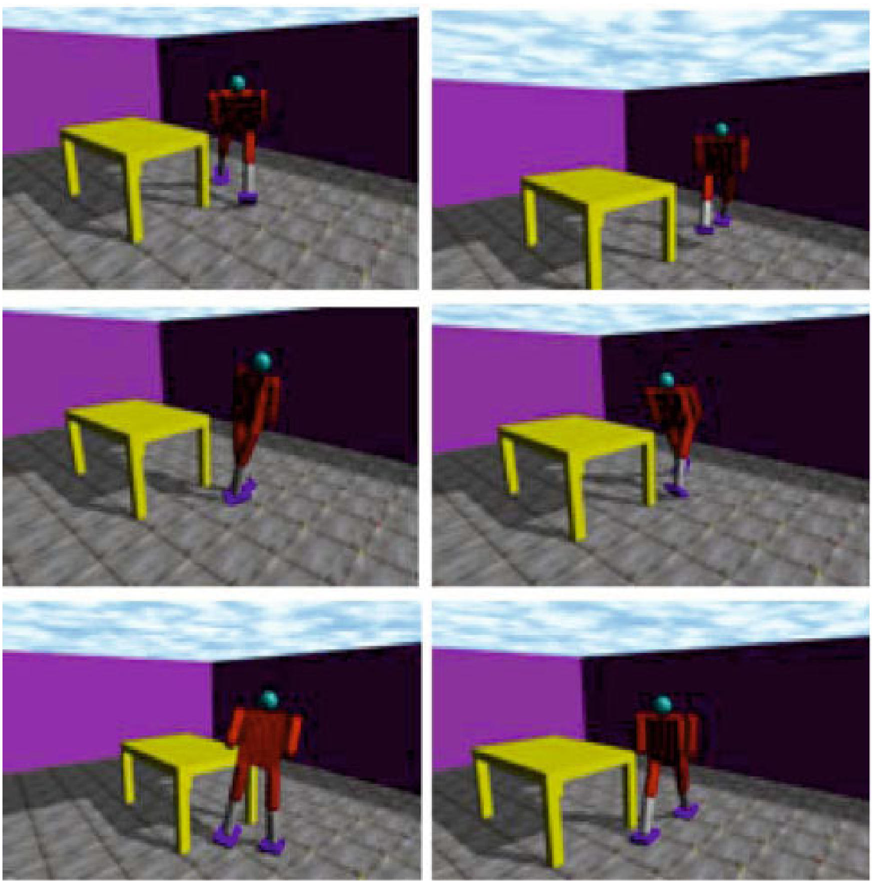

Fig. 8 Avoidance obstacle of the humanoid robot during locomotion

The robot can continue walking without falling over by modulating the torques applied at its feet.

5.2 *Results of the Locomotion on Terrain with Obstacles*

The robot can continue walking without falling over by modulating the torques applied at its feet. In this case study, two angles sensors are used in each foot to have more information about the obstacles.

When the humanoid robot detects obstacles during its course, it changes direction by swiveling its body to orientate itself forward the new direction. The best behavior obtained for obstacles avoidance is presented in fig. 8.

6 Discussion

The evolutionary process in this work was able to successfully produce a stable bipedal walking gait that allowed the humanoid robot to move forward throughout the complete duration of its evaluation period.

Fig.9 shows representative runs of the humanoid robot. Different distances are achieved with evaluated times (10 sec). The robot walking speed is accordingly higher.

Fig.9 shows also, average and maximum fitness values of the robot over 250 generations. Evolutionary runs took approximately two weeks of simulated time on a medium workstation. We noticed that for about 40 generations, no efficient gait emerges. Then, there is a nice steady increase in fitness up to about generation 140, followed by a jagged pattern with a steady upward trend. Any fitness above about

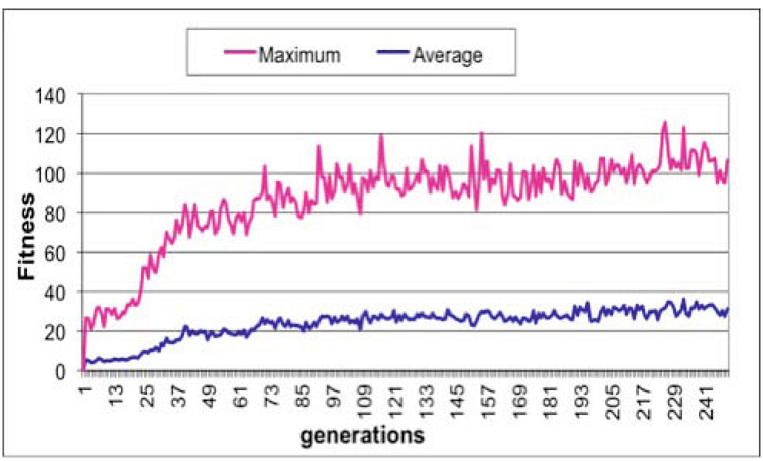

Fig. 9 Fitness graph of representative stable controller evolution. Top fitness and average fitness are shown.

the generation 200 will generally correspond to a reasonable walk in the forward direction, at generations 230 or above corresponding to quite good, stable walks. The maximum fitness generated was at generation 229, with a value of 93.76 corresponding to a fine forward walk with a slight limping gait.

Other patterns of locomotion included skating, with the robot keeping one foot constantly in contact with the ground and pushing along with the other.

7 Conclusion and Future Work

We have demonstrated the suitability of an evolutionary robotics approach to the problem of generating stable three-dimensional bi-pedal walking behaviors. The current implementation is able to evolve agents that are able to walk in a straight line on a planar surface without proprioceptive input. However, the use of proprioceptive sensors will become necessary to stabilize the biped on uneven terrain or in response to directional changes. Incorporation of such additional input is easy with the neural controller employed in this research.

The quality of the results is expected to be further improved by a refined fitness function, as well as a shift toward coupled neural oscillators instead of a single network. Furthermore, it is desirable to incorporate biomechanical knowledge about human walking in order to make maximum use of the passive dynamics of the bodies. These aspects are currently being implemented.

In theory, the results obtained here are directly transferable to em-bodied robots. In practice, however, there are likely to be complications due to a possible lack of accuracy of the physics engine. It re-mains to be seen whether this reality gap can be crossed with appropriate techniques such as noise envelopes [13].

Evolutionary algorithms typically use direct encodings, where each element of a phenotype is independently specified in its genotype. However, these direct encodings are limited in their ability to evolve complex, modular, and symmetric phenotypes because individual mutations cannot produce coordinated changes to multiple elements of a phenotype [11]. Such coordinated mutational effects can occur with indirect encodings, also called developmental or generative encodings, wherein a single element in a genotype can influence many parts of the phenotype [11]; [21]. Indirect encodings have been shown to produce highly regular solutions to problems [11]; [4]; [3] and [12], but their bias toward regularity makes it difficult for them to properly handle irregularities in problems [3].

Analyses suggest that HyperNEAT is successful because it em-ploys a generative encoding that can more easily reuse phenotypic modules. It is also one of the first neuroevolutionary algorithms that exploits a problem's geometric symmetries, which may aid its performance. [4]

For this reason it is important to do a comparative study of our system with the HyperNeat approach.

References

1. Azarbadegan, A., Broz, F., Nehaniv, C.L.: Evolving Simss Creatures for Bipedal Gait. In: IEEE Symposium on Artificial Life, Paris, France, April 11-15. Symposium series on computational intelligence, pp. 218–224 (2011)
2. Cheng, M.Y., Lin, C.S.: Genetic algorithm for control design of biped locomotion. Journal of Robotic Systems 14(5), 365–373 (1997)
3. Clune, J., Ofria, C., Pennock, R.T.: How a generative encoding fares as problem regularity decreases. In: Rudolph, G., Jansen, T., Lucas, S., Poloni, C., Beume, N. (eds.) PPSN 2008. LNCS, vol. 5199, pp. 358–367. Springer, Heidelberg (2008)
4. Clune, J., Beckmann, B.E., Ofria, C., Pennock, R.T.: Evolving coordinated quadruped gaits with the hyperneat generative encoding. In: IEEE Congress on Evolutionary Computing (CEC), Trondheim, Norway, pp. 2674–2771 (2009)
5. Doerschuk, P.I., Simon, W.E., Nguyen, V., Li, A.: A Modular Approach to Intelligent Control of a Simulated Jointed Leg. IEEE Robotics and Automation Magazine 5(2), 12–21 (1998)
6. Gallagher, J.C., Beer, R.D., Espenschied, K.S., Quinn, R.D.: Application of evolved locomotion controllers to a hexapod robot. Robotics and Autonomous Systems 19(1), 95–103 (1996)
7. Elman, L.J.: Finding Structure in Time. Cognitive Science 14, 179–211 (1990)
8. Gruau, F.: Automatic definition of modular neural networks. Adaptive Behavior 3(2), 151–183 (1995)
9. Haykin, S.: Neural Networks: A comprehensive foundation, 2nd edn. Prentice Hall, Upper Saddle River (1999)
10. Holland, J.: Adaptation in Natural and Artificial Systems. MIT Press, Cambridge (1992)
11. Hornby, G.S., Lipson, H., Pollack, J.B.: Generative representations for the automated design of modular physical robots. IEEE Transactions on Robotics and Automation 19, 703–719 (2003)
12. Hornby, G., Takamura, S., Tamamoto, T., Fujita, M.: Autonomous evolution of dynamic gaits with two quadruped robots. IEEE Transactions on Robotics 21(3), 402–410 (2005)
13. Jakobi, N., Husbands, P., Harvey, I.: Noise and the reality gap: The use of simulation in evolutionary robotics. In: Morán, F., Merelo, J.J., Moreno, A., Chacon, P. (eds.) ECAL 1995. LNCS, vol. 929, pp. 704–720. Springer, Heidelberg (1995)
14. Kun, A.L., Miller, W.T.: Control of variable speed gaits for a biped robot. IEEE Robotics and Automation Magazine 6(3), 19–29 (1999)
15. Liu, H., Iba, H.: A hierarchical approach for adaptive humanoid robot control. In: Proceedings of the 2004 IEEE Congress on Evolutionary Computation, Portland, Oregon, pp. 1546–1553, 20–23. IEEE Press, Los Alamitos (2004)
16. Miller III, W.T., Glanz, F.H., Kraft, L.G.: Application of a General Learning Algorithm to the Control of Robotic Manipulators. International Journal of Robotics Research 6(2), 84–98 (1987)
17. Miller, W.T.: Real-Time Neural Network Control of a Biped Walking Robot. IEEE Control Systems Magazine, 41–48 (1994)
18. Pettersson, J., Sandholt, H., Wahde, M.: A flexible evolutionary method for the generation and implementation of behaviors for humanoid robots. In: Proceedings of the IEEE-RAS International Conference on Humanoid Robotic, Japan, November 22-24, pp. 279–286 (2001)
19. Shan, J., Junshi, C., Jiapin, C.: Design of central pattern generator for humanoid robot walking based on multi-objective GA. In: Proc. International Conference on Intelligent Robots and Systems (IROS 2000), vol. 3, pp. 1930–1935. IEEE-RSJ, Takamatsu (2000)

20. Sims, K.: Evolving 3D morphology and behavior by competition. Artificial Life 1(4), 353–372 (1994)
21. Stanley, K.O., Miikkulainen, R.: A taxonomy for artificial embryogeny. Artificial Life 9(2), 93–130 (2003)
22. Taga, G., Yamaguchi, Y., Shimizu, H.: Self-organized control of bipedal locomotion by neural oscillators in unpredictable environment. Biological Cybernetics 65, 147–159 (1991)
23. Takanishi, A., Ishid, M., Yamazaki, Y., Kato, I.: The realization of dynamic walking by the biped walking robot WL-10RD. In: Proceedings of the International Conference on Advanced Robotics (ICAR 1985), pp. 459–466 (1985)
24. Téllez, R.A., Angulo, C., Pardo, D.E.: Evolving the walking behaviour of a 12 DOF quadruped using a distributed neural architecture. In: Ijspeert, A.J., Masuzawa, T., Kusumoto, S. (eds.) BioADIT 2006. LNCS, vol. 3853, pp. 5–19. Springer, Heidelberg (2006)
25. Valsalam, V.K., Miikkulainen, R.: Modular neuroevolution for multi-legged locomotion. In: GECCO 2008: Proceedings of the 10th annual conference on Genetic and Evolutionary Computation, pp. 265–272. ACM, New York (2008)
26. Wang, H., Lee, T.T., Gruver, W.A.: A neuromorphic controller for a three-link biped robot. IEEE Transactions on Systems, Man and Cybernetics 22(1), 164–169 (1992)
27. Wolff, K., Nordin, P.: Learning biped locomotion from first principles on a simulated humanoid robot using linear genetic programming. In: Cantú-Paz, E., Foster, J.A., Deb, K., Davis, L., Roy, R., O'Reilly, U.-M., Beyer, H.-G., Kendall, G., Wilson, S.W., Harman, M., Wegener, J., Dasgupta, D., Potter, M.A., Schultz, A., Dowsland, K.A., Jonoska, N., Miller, J., Standish, R.K. (eds.) GECCO 2003. LNCS, vol. 2723, pp. 495–506. Springer, Heidelberg (2003)
28. Ziegler, J., Barnholt, J., Busch, J., Banzhaf, W.: Automatic evolution of control programs for a small humanoid walking robot. In: Bidaud, P. (ed.) Proc. 5th International Conference on Climbing and Walking Robots (CLAWAR 2002), pp. 109–116. Professional Engineering Publishing (2002)