Evolution of Central Pattern Generators for Bipedal Walking in a Real-time Physics Environment

Torsten Reil, Phil Husbands

Abstract— **We describe an evolutionary approach to the control problem of bipedal walking. Using a full rigid-body simulation of a biped, it was possible to evolve recurrent neural networks that controlled stable straight-line walking on a planar surface. No proprioceptive information was necessary to achieve this task. Furthermore, simple sensory input to locate a sound source was integrated to achieve directional walking. To our knowledge, this is the first work that demonstrates the application of evolutionary optimization to three-dimensional, physically simulated biped locomotion.**

Keywords— **evolutionary robotics, bipedal walking, evolutionary algorithms, recurrent neural networks, physics**

I. Introduction

Bipedal walking is a difficult task due to its intrinsic instability, and developing successful controller architectures for this mode of locomotion has proved substantially more difficult than for other types of walking [1].

There is considerable interest in this matter, from disciplines as diverse as robotics, computer graphics, virtual reality, and biology. However, previous approaches have been based on conventional control strategies. As will be discussed shortly, this brings about considerable complications and limitations. In addition, past work has been constrained by only limited available means to simulate the physics of the body to be controlled, thus making it either necessary to build robots or resort to simplified models in simulation.

Given the inherent difficulties in designing stable controllers for natural looking bipedal walking, it was decided to investigate the use of evolutionary robotics techniques [2] in developing recurrent dynamical neural-network-based controllers for the task. This paper describes successful experiments in evolving controllers for a realistically simulated biped. The structure of the paper is as follows:

We first review previous work on controller architectures for bipedal walking. These are subsequently contrasted with the approach taken here: evolutionary robotics. Section II describes the implementation of both the biped and the neural controller, as well as the evolutionary algorithm used in this research. As shown in the subsequent results section, this combination succeeded in producing natural looking bipedal walking. Section IV addresses the integration of sensory input and describes a corresponding, successful experiment. The article closes with a discussion of the research presented.

Department of Zoology, University of Oxford, South Parks Road, Oxford, OX1 3PS, UK. Email: torsten.reil@zoo.ox.ac.uk

COGS, University of Sussex, Falmer, Brighton, BN1 9QH, UK

A. Related Work

The major thrust of research on bipedal walking has come from computer graphics and robotics. In the case of the former, animation techniques such as motion capture [3] have come to dominate the area. Motion capture essentially implies filming the desired human behaviour, and using the obtained data to animate a computer generated equivalent. The advantage of this approach is clearly the ability to immediately generate realistic bipedal motion dynamics. However, Laszlo *et al.* [4] make it clear that "motion capture does not provide us with sufficient understanding to create more general walking motions, especially when conditions are unpredictable, when new motions need to be generated, or when dealing with non-human characters."

These shortcomings can be overcome by a second approach, which is based on a semi-physical representation of the biped and a controller to create movement patterns. With techniques such as inverse kinematics and inverse dynamics, virtual limbs can be placed at the desired positions, and the required forces are computed accordingly.

Several workers [4][5] have followed this approach to create computer animations of humans. The equations of motion are either produced specifically for the model to be animated or are generated with available packages [5][6]. Typically, a finite state machine determines the control actions (with cyclic states such as heel contact, toe contact, unloading or flight [5][4][7]), and special forms of limit cycle control may be applied to achieve the necessary stability [4]. The animated end results of these efforts closely resemble natural motion patterns, but may nevertheless fail to convince the human eye in specifically designed "motion Turing tests" [5] (these tests confront human subjects with computer generated and real-life animations, and ask them to discern between the two). This lack of realism is a direct consequence of the controller architecture employed; a state machine does not readily produce the fluctuations typical of real locomotion. More significantly, it cannot easily be extended to integrate sensory input. Furthermore, the creation of state machines can be a cumbersome process, as states have to be identified, implemented and fine tuned by hand for each type of gait to be modelled [5].

Bipedal locomotion in robots is subject to the physical laws of the natural world, and hence short cuts like motion capture are not available. Bipedal robots with varying complexities have been produced and controlled by several researchers [8]-[16]. As with computer graphics models, the corresponding controller architectures are typically based on state machines with special algorithms added on top to provide the necessary stability. Most recently, Pratt *et al.* [14] have used Virtual Model Controllers for planar bipedal robots. Here, virtual mechanical components are attached to the robot and exert real actuator torques or forces. For example, a *virtual dog track bunny* is used to maintain a desired velocity in a planar biped robot. A state machine changes the virtual component connections or parameters at each state transition. Together with a set of simple rules for, e.g., height, pitch, and speed stabilisation, this allows a more intuitive development of stable controller architectures and somewhat eases the problem of mathematical tractability encountered in previous attempts [13]. In addition to testing and optimising control strategies on the real robot, Pratt and Pratt [17] have used a rigid-body simulation [6] to create a realistic model of a biped. This allowed efficient experimentation with the robot's natural dynamics (such as passively swinging legs).

In summary, with few exceptions, such as Miller [18], who utilizes reinforcement learning for training a neural net, previous approaches to bipedal walking have been based on engineering techniques like state machines and conventional control theory. As remarked on earlier, this causes a number of problems: a) mathematical tractability, b) manual optimisation, c) limited extendability, and d) limited biological plausibility. It is argued here that the evolutionary robotics approach presented below has the potential to overcome the first three constraints by improving on the last one, biological plausibility.

B. Evolutionary Robotics

Evolutionary robotics was introduced as an alternative to the hand design of robot controllers, especially for autonomous robots acting in uncertain and noisy domains [19][20]. Evolutionary algorithms are used to search spaces of controllers (and potentially body and sensor layouts too) described by a set of variables encoded on the artificial genotype. The fitness function is usually task-based, that is, high scores are achieved by controllers that enable the robot to perform the desired task well. These controllers are nearly always in the form of some kind of artificial neural network (ANN). The job of the evolutionary search algorithm ranges from optimizing the parameters of a fixed-architecture ANN [21][22][23] to exploring complex network spaces where the architecture and many properties of the nodes and connections are under evolutionary control [24][25].

There have been many successful applications of evolutionary robotics to date, ranging from simple reactive behaviours in wheeled robots with IR proximity sensors [22][26], through visually guided behaviours in simple wheeled robots [27][28], to fairly complex non-reactive behaviours in simple wheeled robots [29], and a variety of locomotion controllers for 6- and 8-legged robots [30]- [34]. For far more detailed reviews of the field see [37][2].

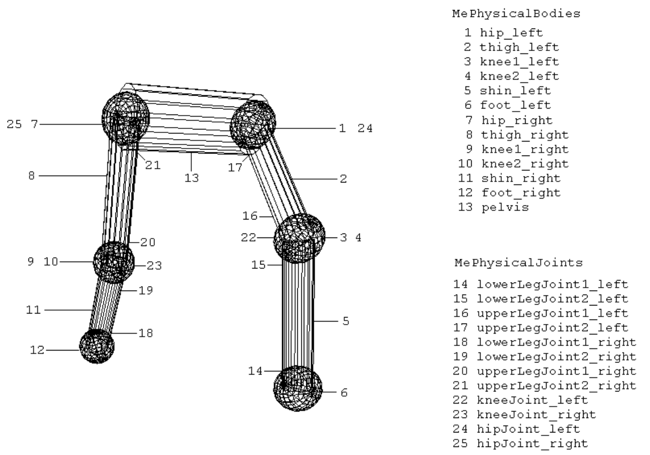

Fig. 1. MathEngine implementation of the biped. Note that although two bodies are used to implement the knee, only one body is necessary.

To date evolutionary robotics techniques have not been applied to a task as dynamically unstable as controlling bipedal locomotion.[1] It is this inherent instability (generally two-legged walkers will fall over without continuous active control) that provides severe challenges to the hand design of such controllers, especially if smooth natural walking is required. However, given the success of evolved locomotion controllers for relatively stable hexapod and octopod robots [30]-[34], it was deemed appropriate to investigate the use of such techniques for developing bipedal locomotion controllers. As will be seen later in this paper, evolutionary robotics methods were indeed successful in finding stable controllers for bipedal walking.

II. Implementation

A. The Biped

A.1 MathEngine Bodies and Joints

Unlike previous, embodied approaches, the agent to be controlled here is modelled using the rigid body dynamics simulation SDK of *MathEngine*$^{TM.}$ This allows the evaluations to be run significantly faster than real time [38], and thus greatly increases the efficiency of the evolutionary approach.

MathEngine's Fast Dynamics Toolkit was developed to overcome the two most pressing problems in the simulation of physics: complexity and speed. Programmed in C, it supports bodies, joints, contacts and forces, the attributes of which can be set by the user [39]. Once set up, a physical world is integrated by the engine over time in user-defined intervals. The SDK used here (1.0.5) is shipped with an OpenGL and Direct3D renderer to visualize the scene.

The implementation of the bodies for this research is characterized by the need to capture the fundamental fea-

[1] While other researchers such as Rodrigues [35] and de Garis [36] did use evolutionary optimization in the context of bipedal walking, to the authors' knowledge, no research has so far demonstrated the applicability of evolved recurrent neurocontrollers for a real-time and physically realistic biped simulation.

Composite Body	Length	Mass
Pelvis	0.5 m	1 kg
Upper Leg	0.5 m	0.6 kg
Lower Leg	0.5 m	0.6 kg

TABLE I
Walker dimensions and masses. ME bodies are combined to meta-bodies

Joint Angle	Radians	Degrees
knee_front_back	-1.4 to 0.0	-89.1 to 0.0
hip_lateral	-0.8 to 0.8	-50.9 to 50.9
hip_front_back	-1.3 to 1.6	-84.7 to 101.7

TABLE II
Angle limits of biped joints. Values were obtained heuristically

tures of a biped while limiting the body's complexity and degrees of freedom. Thus, the model used here consists of two articulated legs connected by a link. Thirteen MathEngine bodies and eleven joints are used to implement these structures, as illustrated in Figure 1 (Because each leg consist of two composite bodies with two spheres and one connecting link each, the present implementation uses two bodies for each knee).

The degrees of freedom (DOF) of the joints are: hip joint: 2 DOF (pitch/roll) - knee: 1 DOF (pitch), giving a total of 6 DOF.

Although important for real walking (for example in birds or humans), feet and ankle joints are not implemented. They impose additional DOF and would therefore considerably increase the controller's search space. In addition, the capabilities of MathEngine SDK 1.0.5 make realistic foot-floor contact a computationally expensive endeavour (due to the need for multiple contact points). Sphere-plane contacts (sphere radius: 8 cm) provide an uncomplicated and fast alternative and are used instead. As will become clear later this simplification does not come at a noticeable cost in terms of the overall body dynamics.

A.2 Actuators

Muscle action is modelled by proportional derivative (PD) controllers [40][5], which are essentially equivalent to damped torsional springs. Their *modus operandi* is characterized by the following equation:

$$T = k_s(\theta_d - \theta) - k_d\dot{\theta} \tag{1}$$

with T being the torque force, k_s the spring constant, k_d the damping constant, θ_d the desired angle, and θ the current angle.

Rather than directly defining the strength of actuator forces, the controller updates the natural orientation of the PD controller (the desired angle of the limb). Eq. 1 is then used to compute the force necessary to move the limb to that position. The spring constant and damping value determine the strength and the tendency to oscillate. Their values therefore significantly influence the realism of movements. By means of manual experimentation, values of $k_s = 5$ and $k_d = 4$ were found to be appropriate and are used for all actuators.

The PD approach largely eliminates the need to physically limit joint angles, since the same effect is achieved by constraining the range of θ_d. Table II shows the constraints used for the biped introduced earlier. In addition,

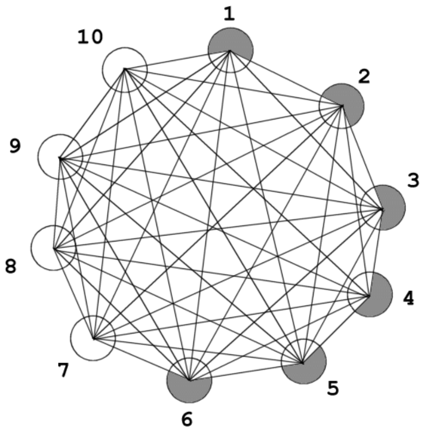

Fig. 2. The recurrent neural network used to control bipedal walking. Shaded nodes are motor neurones. Note that connections are bi-directional and asymmetric.

PDs provide control at the mechanical level as they automatically reduce any discrepancy between the current and desired angle; whether this discrepancy has come about by a controller-mediated updated value or by the dynamics of the physical world is irrelevant. As a consequence, the tendency of limbs, for example, to buckle under the mass of the body is counteracted directly by the PDs, a feat which would otherwise have to be accomplished by the neural controller.

B. Controller Architecture

Legged locomotion is characterized by cyclic activity of the limbs. In vertebrates and many invertebrates, the underlying rhythmic neural activation patterns are created by designated network structures called central pattern generators (CPGs) [41]. The defining feature of these is a high degree of recurrency, which greatly biases the dynamics of the system towards cyclic activation patterns.

In order to capitalize on comparable inherent dynamics, the controller architecture used in this research is based on a recurrent neural net, the structure of which is depicted in Figure 2. (Similar networks have been used successfully as CPGs by [42] and [32], in both cases for multilegged robots.)

Each network consists of ten fully interconnected neurons. Besides the weights, the behaviour of a node is governed by two other parameters, a time constant τ_j and a bias t_j. At each iteration (time step: 0.02s), the activity

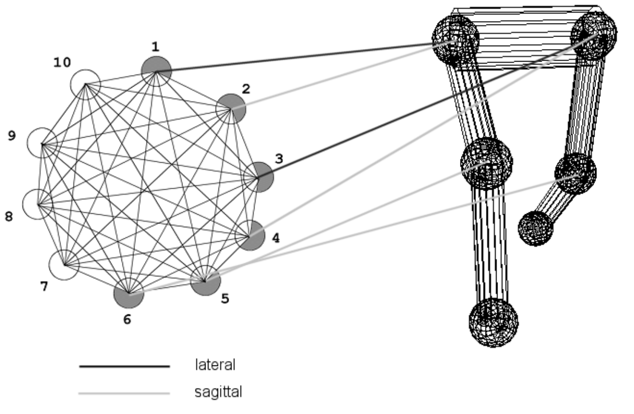

Fig. 3. Motor connections between controller and walker. The hips have 2 DOFs each (sagittal (i.e., front to back) and lateral), the knees have 1 DOF each (front to back).

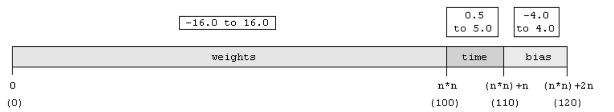

Fig. 4. Encoding scheme. Chromosome index is shown in general form (n: number of nodes) and for special case of n = 10 (in brackets). Parameters are encoded as real values. Ranges in boxes represent upper and lower bounds of the respective parameter types.

of the jth neuron is computed according to equation 2,

$$\tau_j \dot{A}_j = -A_j + \sum w_{ij} O_i \tag{2}$$

with τ_j being the time constant of the jth neuron, A_j its activity, O_i the output from the ith neuron, and w_{ij} the weight from the ith to the jth neuron.

The corresponding output is calculated as follows:

$$O_j = (1 + e^{(\alpha_j - A_j)})^{-1} \tag{3}$$

where α_j is the bias of the jth neuron.

Nodes 1 to 6 are special in that they control the biped's actuators and they can therefore be considered to be motor neurons. Their outputs (from 0.0 to 1.0) are scaled to map to the angle limits listed in Table II. Figure 3 schematically depicts the motor connections.

C. Evolutionary Algorithm

C.1 Encoding Scheme and Population Parameters

The parameters to be optimized are weights, time constants, and biases. The encoding scheme spatially separates the three types in the chromosome (Figure 4).

Parameter values are coded as real numbers, with different ranges for each data type. Following [42] and [32], these are [-16.0, 16.0] for the weights, [0.5, 5.0] for the time constants, and [-4.0, 4.0] in the case of the biases. The assignment of these ranges is simplified by the spatial separation of the types; similarly, different mutation rates and sizes can be applied. While the former remains constant throughout the chromosome, the differential implementation of the latter is necessary due to the varying parameter value ranges. This is achieved by using Gaussian distributions. Table III shows the mutation sizes in form of standard deviations from mean 0. Values exceeding the allowed range are clipped to the maximally permitted level. (This is known to create disproportionate accumulations around the clipping points [43], which were, however, found to be negligible in this work.)

Parameter	σ	μ_{I}	μ_{M}
weight	8	0	0
time constant	1.5	2.75	0
bias	3	0	0

TABLE III

Initialization (i) and mutation (m) distributions for different parameter types. Standard deviation and separate means of Gaussian distribution are shown

The mutation rate is calculated so as to cause on average one change per chromosome. Thus, for larger networks an accordingly lower rate per locus is applied. Together with typically small mutation sizes (see Table III), this ensures that the evolutionary search is local and gradual. Each population consists of 50 individuals, and its individual controllers are initialized with randomized values using the initialization distributions of Table III. Rank-based selection is used for reproduction with a fittest fraction of 0.5 (this essentially means culling the bottom half of the population and replacing it with a copy of the top half [44]). No crossover operations are applied both on theoretical (no identifiable functional units in the genotype and phenotype structure [45]) and empirical grounds (recent experimental evidence on lack of efficiency of crossover in this problem domain [25]).

C.2 Evaluation of Controllers

Despite the complexity of bipedal locomotion, it is possible to reduce the fitness function to the following two components:

a) Maximize distance travelled from origin

b) Do not lower centre of gravity below a certain height

The first objective implicitly includes the locomotion component while at the same time rewarding walking in a straight line rather than in circles (note that this would not be true for *Maximize overall distance travelled*). The second goal combines two further factors: it penalizes falling down as well as grotesque movements. (Much of the second point is already prohibited by constraining the joint angles in the physical model.) To improve efficiency evaluations are terminated early if they are unpromising, i.e. as soon as objective b) is not met. Hence, the fitness function for a biped on an x-z walking plane can be expressed as follows:

$$W = \sqrt{(x_t - x_0)^2 + (z_t - z_0)^2} \tag{4}$$

with W being the fitness, x and z the planar components of the walker position, and t the time of the evalua-

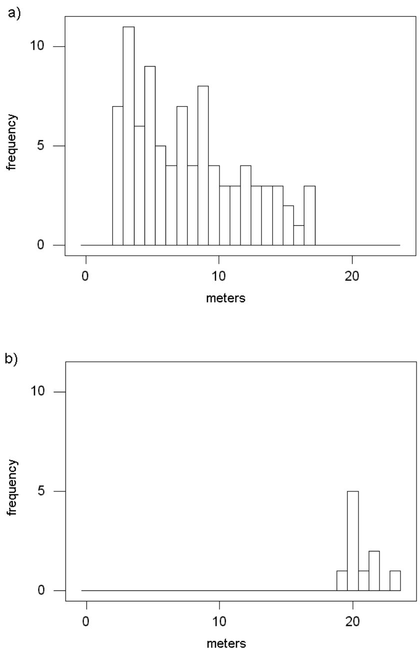

Fig. 5. Distribution of distances covered by unstable (a) and stable (b) individuals. The top individuals of a total of 100 different evolutionary runs are shown. Bin size: 1m.

tion termination. Evaluations are started with all neuron activations set to zero and with the biped set to an upright stance. One run is performed per evaluation, with a maximum possible length of 50 seconds each.

III. Results

Populations of randomly initialized individuals were evolved according to the fitness criteria outlined above. In order to yield meaningful statistics, 100 evolutionary runs were conducted, each consisting of 120 generations. Figure 5 shows the distribution of the distances covered by the top individual of each run in the last generation. For reasons of clarity, the distributions for unsuccessful (unstable) and succesful (stable) controllers are shown separately.

Evaluations of individuals in distribution (a) were terminated prematurely because their centre of gravity fell below the specified height (see II.C.2). Individuals of distribution (b), on the other hand, did not fall over in the given amount of time (50 seconds). Additional trials with such controllers showed them to be capable of walking for an indefinitely long period. They can therefore be regarded as stable.

The fraction of evolutionary runs leading to stable walkers was 10%, the average walking distance of which was 20.577m (σ=1.083). This compares to an average walking distance of 7.878m (σ=4.352) for the unstable walkers. Figure 6 shows the fitness graph of a run resulting in such a stable controller. The neural activation patterns of the top controller in generation 120 are depicted in Figure 7. Figure 8 contains a biped motion sequence of the evolved controller. All controllers evolved in the course of the ex-

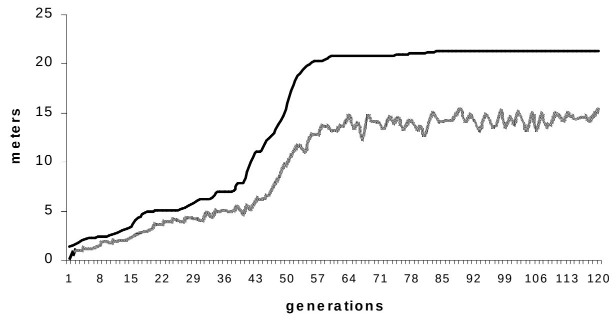

Fig. 6. Fitness graph of representative stable controller evolution. Top fitness (black) and average fitness (grey) are shown

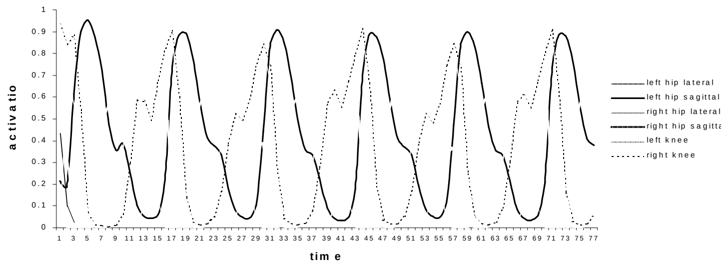

Fig. 7. Motor neurone activation levels of top individual of generation 120 (Figure 6).

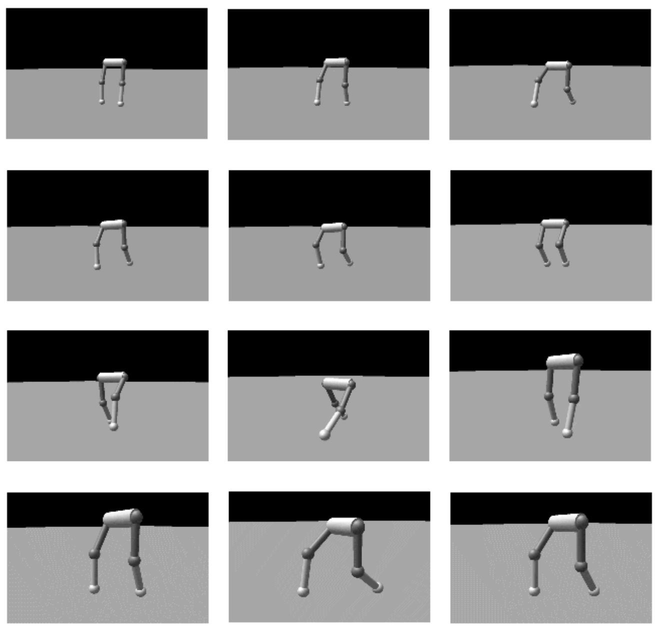

Fig. 8. Motion sequence of biped controlled by top individual of generation 120 (Figure 6). Frame order from left to right, top to bottom. See text for details

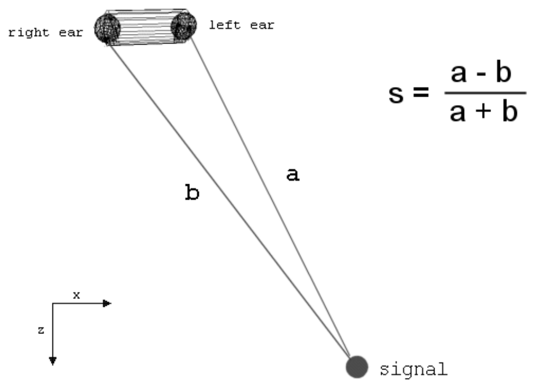

Fig. 9. Preprocessing of two signal values resulting in final single value s. The difference is divided by the sum of the two signal strengths to give a stronger directional signal close to the sound source. This also causes the signal to vanish with the biped facing the sound source.

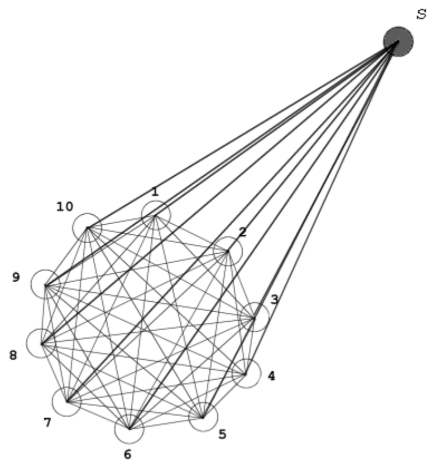

Fig. 10. Each node of the RNN receives as input the preprocessed signal value s.

periments walked in a straight line, a direct result of the fitness function (Eq. 4). Backwards walking controllers were also evolved, albeit at a lower frequency than their forward counterparts. The overall diversity of walkers was large; gaits differed markedly from each other both in terms of speed (as seen in Figure 5) and use of limbs. Knee movements in particular showed considerable variation ranging from fully swinging to constantly extended. Moreover, several gaits displayed asymmetry, both in stride length and limb use.

A. *Efficiency of Evolutionary Runs*

The fact that only 10% of evolutionary runs led to stable walking appeared to indicate room for improvement. For the majority of unsuccessful controllers, analysis showed that the little distance they did cover was controlled by the settling phase of the recurrent net. We therefore added an additional fitness criterion that actively rewarded cyclic activity. This markedly increased the proportion of successful runs (to ca. 80%), but was not reflected in a proportionate improvement of the overall time efficiciency (i.e. successful controllers per processor cycle). The reason for this lies in the fact that unsuccessful controller evaluations are aborted early (see II.C.2), thus taking up only limited computational resources.

IV. Integrating Sensory Input

The controllers described in section III are purely rhythm generating structures. Although sufficient to produce stable walking behaviour in a non-fluctuating environment, they are not capable of dealing with rough terrain or responding to external stimuli. In order to achieve this, sensory input must be integrated and the CPG activity modified accordingly. A simple set of experiments was carried out to explore the potential to integrate basic sensory input and will be described now.

- The biped is to walk towards the equivalent of a sound source. It is equipped with two 'ears', the inputs of which are preprocessed to give a single signal which becomes stronger with decreasing distance from the source.[2] In addition the signal vanishes when the biped is directly facing the source. (Figure 9). The signal is fed into the CPG as depicted in Figure 10. The population is initialized uniformly with clones of the top individual from the run depicted in Figure 6. The CPG weights are clamped, but the weights of the ten connections between the sensory node and the RNN nodes are under the control of the EA. At each evaluation, the biped starts from its default position and is presented successively with two sound source locations. Because the task is to approach the signal as closely as possible, the fitness function is the negative distance of the walker to the sound source at the time of termination (as caused by the conditions outlined in II.C.2), or the natural end of the evaluation (after 50 seconds).

A. *Results*

Figure 11 depicts an evolutionary run with the population seeded with individuals from the run depicted in Figure 6. The graph is characterized by an initially strong increase in fitness, but it fails to reach the maximum fitness value of 0.0.

As illustrated in Figure 12, the controller succeeded in walking towards the respective signal positions in the two runs.

However, visual analysis of the walkers made clear that the gait becomes unstable close to the respective signal sources. This is particularly true for the second, right run.

To further investigate the ability of the sensors to modulate the net's activity pattern as well as to examine the reasons for the eventual instability, the neuronal activation patterns of the two runs were recorded and are represented in Figure 13.

The activation graphs indicate that the turning behaviour is at least partly achieved by modulating the ampli-

[2]This approximataion does not hold true when the biped is very close to the sound source (i.e., when the distance to the source is comparable with the distance between the agent's ears).

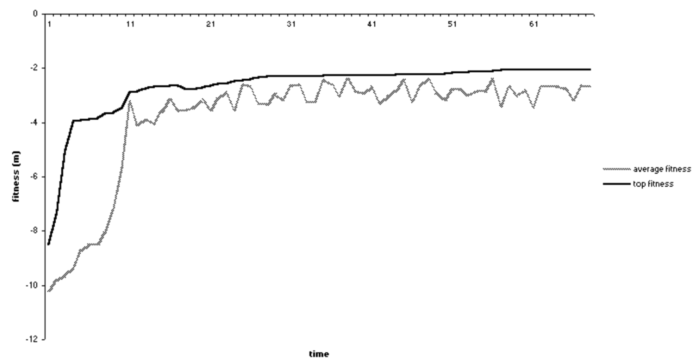

Fig. 11. Fitness graph of bipedal walking with sensory integraton. The population was seeded with individuals from the run depicted in Figure 7.

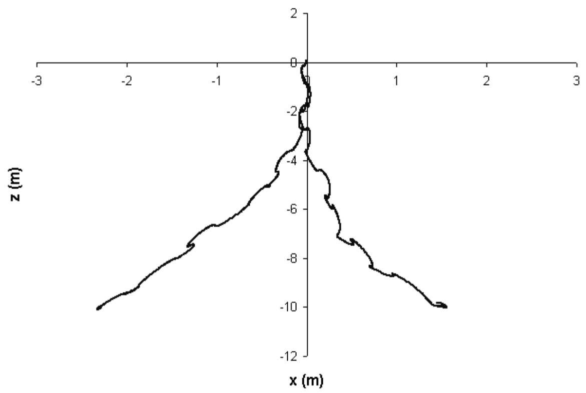

Fig. 12. Trajectories of biped with sensory integration (two runs are shown). Signals are located at [-3,-10] and [3,-10]. Run one: left. Run two: right.

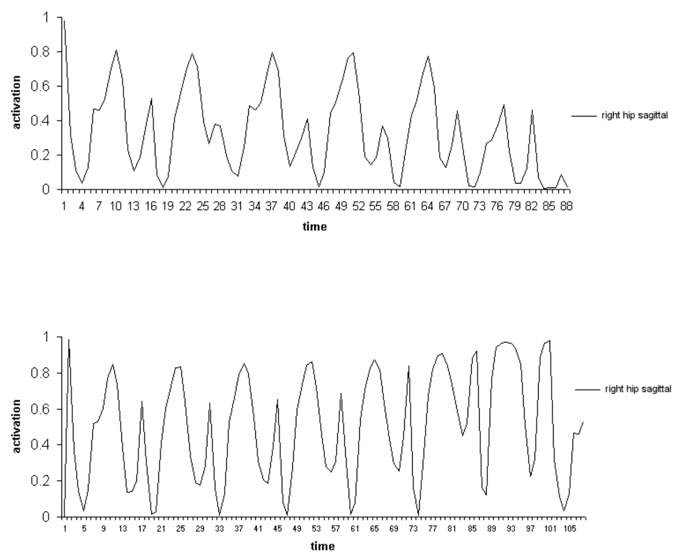

Fig. 13. Neural activation graph of biped walking towards right (top) and left (bottom) signal. Only activation of right hip sagittal (i.e. front to back) neuron is shown. (Other neural activations did not differ markedly from those of Figure 7.)

tude of the right hip sagittal motor neuron (which controls the front to back movement), with a decrease resulting in right (top graph) and an increase resulting in left (bottom graph) turning. Additional experiments were carried out, including evolution of the behaviour from scratch (i.e. unseeded populations), and evolution with seeded populations but with evolvable CPG weights. However, neither of these additional experimental series produced results superior to those documented above.

V. Discussion

Despite an extremely simple fitness function, the EA employed in this research was capable of producing stable straight-line walkers without the use of proprioceptive sensory input. Evolutionary algorithms rely on evolvable systems, and the gradual nature of the fitness graphs (e.g. Figure 6) indicates that the recurrent networks used here can indeed be optimized in a continuous, gradual manner. This notion has recently been corroborated by Rendel [47], who has shown that the fitness landscape underlying the current controller architectures is very smooth. For example, it was possible to gradually modify the amplitude and period of specific motor neuron cycles without affecting those of others.

A further characteristic of the current set-up is the ability to create a large diversity of gaits, both in terms of speed and the use of limbs. Several gaits showed considerable similarity to human walking, although this was not specifically selected for. A potential way to further increase the realism of the motion is to select for minimum energy expenditure (a simple measure for this would be the average actuator activity). It is expected that this fitness component will particularly reward the use of knees. In the current implementation, several controllers walked with extended legs, because this is concomitant with a large stride length. Humans, however, use the momentum of a forward-swinging lower leg, which is energetically more favourable [17].

The evolved controllers were further characterized by nonrepetitive activity cycles; instead, small fluctuations were observed (Figure 7). Similar fluctuations have elsewhere been found to contribute to the perceived realism of simulated locomotive behaviour [48]. Real bipedal walking contains fluctuations in successive cycles, and it is the lack of these that the human eye picks upon in other artificial walking bipeds.

The preliminary experiments on the integration of sensory input indicate that CPG activity can indeed be modified by external stimuli in a meaningful way (Figure 13). However, it is clear that the current sensory architecture is insufficient to modulate the biped's behaviour and retain stability. For example, the simple pre-processing function causes large, destabilising, fluctuations if the biped is close to the signal source. This problem is further intesified by the lack of active balancing mechanisms. The integration of proprioceptive (e.g., limb positions and velocities) and vestibular (balance) input is therefore a necessary next step to achieve more interactive and robust behaviour.

A question that was not systematically explored is in how far the network size affects the efficiency of the approach, both in terms of search space as well as internal dynamics of the net. With the current architecture, a linear increase in the number of nodes leads to an quadratic increase of the corresponding search space. A possible way to circumvent this problem is to employ identical subnetworks for each leg. Such a constellation seems to reflect the natural arrangement of coupled oscillators more accurately [49] [50] and has been successfully used elsewhere in the context of multilegged locomotion [42] [51].

We would like to reiterate that the stable walkers arrived at did not require proprioceptive input to achieve stable walking in a straight line. This corroborates results obtained elsewhere [52] that show that mechanical walkers can attain stable straight-line walking on a planar surface without active balance control. While those bipeds were mechanically fine tuned to exploit gravity as an energy source, the implementation presented here relies on evolutionary optimisation to fine tune active actuation.

We believe that the neuroevolutionary approach described here brings about several major benefits: a) it is fully automated, hence changes in morphology or actuator implementations can be easily accommodated by re-evolving the controllers, b) the diversity of locomotive behaviours is large because the system does not require *a priori* knowledge as to how to solve the control problem, and c) the evolved controllers are computationally very cheap (typically taking up 0.5% of the processing power required by graphics and physics).

VI. Conclusion

We have demonstrated the suitability of an evolutionary robotics approach to the problem of stable 3D bipedal walking in simulation. The current implementation is capable of walking in a straight line on a planar surface without the use of proprioceptive input. However, the use of the latter will become necessary to stabilize the biped on uneven terrain or in response to directional changes. The neural controller employed in this research lends itself to the incorporation of such additional input.

The quality of the results is expected to further improve by a refined fitness function, as well as a shift towards 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 embodied robots. In practice, however, there are likely to be complications due to a possible lack of accuracy of the physics engine. It remains to be seen whether this 'reality gap' can be crossed with appropriate techniques such as noise envelopes [26].

VII. Acknowledgements

The authors would like to thank Colm Massey and Will Wray (*MathEngine*) for support concerning the physics engine, and Neil Pattinson, Julian Worby and David Raubenheimer for helpful discussions. We are also grateful to the anonymous reviewers of a previous version of this paper for their helpful and constructive comments.

This research was supported by the EPSRC and through the Guy Newton Scholarship in Zoology with Wolfson College, Oxford.

References

[1] Beer, R. D., Chiel, H. J., Quinn, R. D., Ritzmann, R. E.(1998) Biorobotic Approaches to the Study of Motor Systems, *Current Opinion in Neurobiology*, **8**: 777-782.

[2] J.-A. Meyer, P. Husbands and I. Harvey (1998) Evolutionary Robotics: A survey of applications and problems. In P. Husbands and J.-A. Meyer (Eds), *Evolutionary Robotics*, 1-21, Springer Verlag LNCS vol. 1468.

[3] Bruderlin, A., Williams, L. (1995) Motion Signal Processing. *Computer Graphics Proceedings, Annual Conference Series (SIGGRAPH '95)*, pp. 97-104.

[4] Laszlo, J.F., van de Panne, M., Fiume, E. (1996) Limit Cycle Control and its Application to the Animation of Balancing and Walking. *Computer Graphics Proceedings, Annual Conference Series (SIGGRAPH '96)*, pp. 155-162.

[5] Hodgins, J.K. (1996) Three-Dimensional Human Running. *Proceedings of the IEEE Conference on Robotics and Automation.*.

[6] MIT Creatures, http://www.ai.mit.edu/projects/leglab/ simulations/simulations.html.

[7] van de Panne, M., Kim, R., Fiume, E. (1996) Virtual Wind-up Toys for Animation, *Graphics Interface '94*, pp. 208-215.

[8] Ogo, K., Ganse, A., Kato, L. (1980) Dynamic walking of biped walking machine aiming at completion of steady walking. *Third Symposium on Theory and Practice of Robots and Manipulators.* Elsevier Scientific Publishing.

[9] Kato, T., Takanishi, A., Jishikawa, H., Kato, I. (1983) The realization of the quasi-dynamic walking by the biped walking machine. *Fourth Symposium on Theory and Practice of Robots and Manipulators*, Polish Scientific Publishers.

[10] Takanishi, A., Tochizawa, M., Karaki, H., Kato, I. (1989) Dynamic Biped Walking Stabilized with Optimal Trunk and Waist Motion. *IEEE/RSJ International Workshop on Intelligent Robots and Systems '89*, pp 187-192.

[11] Miura, H., Shimoyama, I. (1984) Dynamic Walk of a Biped. *International Journal of Robotics Research* **3**: 30-74.

[12] Furusho, J., Sano, A. (1990) Sensor-Based Control of a Nine-Link Biped. *International Journal of Robotics Research*, **9**(2): 83-98.

[13] Pratt, J., Pratt, G. 1998. Intuitive Control of a Planar Bipedal Walking Robot. *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '98)*, Leuven, Belgium 1998.

[14] Pratt, J., Dilworth, P., Pratt, G. (1997) Virtual Model Control of a Bipedal Walking Robot. *Proceedings of the IEEE International Conference on Robotics and Automations (ICRA '97)*, Albuquerque, NM. 1997.

[15] T. Mita, T. Yamaguchi, T. Kashiwase and T. Kawase (1994) Realization of a high speed biped using modern control theory, *Int. J. CONTROL*, **40**: 1, 107-119.

[16] Honda P3 Humanoid Robot, http://www.honda.co.jp/english/ technology/robot/index.html.

[17] Pratt, J., Pratt, G. (1998) Exploiting Natural Dynamics in the Control of a Planar Bipedal Walking Robot. *Proceedings of the Thirty-Sixth Annual Allerton Conference on Communication, Control, and Computing*, Monticello, IL, September 1998.

[18] Miller, W. T. (1994) Real-time neural network control of a biped walking robot. *IEEE Control Systems*, **14**: 41-48.

[19] Brooks, R. (1992) Artificial Life and Real Robots. In: F. Varela and P. Bourgine (Eds) *Proc. First European Conf. on Artificial Life*, 3-10, MIT Press.

[20] Husbands, P., Harvey, I. (1992) Evolution versus design: Controlling autonomous robots. In *Proc. 3rd Annual Conf. on Artificial Intelligence, Simulation and Planning*, 139-146, IEEE Press.

[21] Beer, R., Gallagher, J. (1992) Evolving dynamical neural networks for adaptive behavior, *Adaptive Behavior* **1**(1): 91-122.

[22] Floreano, D., Mondada, F. (1994) Automatic creation of an autonomous agent: genetic evolution of a neural network driven

robot. In D. Cliff et al. (Eds) *From Animals to Animats 3*, MIT Press.

[23] Nolfi, S., Floreano, D., Miglino, O., Mondada, F. (1994) How to evolve autonomous robots: Different approaches in evolutionary robotics. In R. Brooks and P. Maes (Eds) *Artificial Life IV*, MIT Press.

[24] Floreano, D., Mondada, F. (1996) Evolution of plastic neurocontrollers for situated agents. In P. Maes *et al.* (Eds) *From Animals to Animats 4*, MIT Press.

[25] Husbands, P., Smith, T., Jakobi, N.,O'Shea, M. (1998) Better living through chemistry: evolving GasNets for robot control, *Connection Science*, **10** (3&4): 185-210.

[26] Jakobi, N., Husbands, P., Harvey, I. (1995) Noise and the reality gap: the use of simulation in evolutionary robotics. In J. Moran et al (Eds) *Proc. 3rd European Conf. on Artificial Life*, Springer Verlag.

[27] Harvey, I. Husbands, P. (1994) Seeing the light: artificial evolution, real vision. In D. Cliff *et al.* (Eds) *From Animals to Animats 3*, MIT Press.

[28] Floreano, D., Nolfi, S., Mondada, F. (1998) Competitive co-evolutionary robotics: From theory to practice, *From Animals to Animats 5*, 515-524, MIT Press..

[29] Nolfi, S. (1997) Evolving non-trivial behaviour in autonomous robots: Adaptation is more powerful than decomposition and integration. In T. Gomi (ed) *From Intelligent Robots to Artificial Life (ER'97)*, AAI Books.

[30] Gallagher, J., Beer, R., Espenschiel, K., Quinn, R. (1996) Application of evolved locomotion controllers to a hexapod robot. *Robotics and Autonomous Systems* **19**: 95-103.

[31] Gruau, F. (1997) Cellular encoding for interactive evolutionary robotics. In P. Husbands and I. Harvey (Eds), *Fourth European Conf. on Artificial Life*, MIT Press.

[32] Jakobi, N. (1998) Running Across the Reality Gap: Octopod Locomotion Evolved in a Minimal Simulation. In Husbands, O., Meyer, J. (eds), *Lecture Notes in Computer Science - Evolutionary Robotics.* Springer Verlag.

[33] Gomi, T and Ide, K (1997) Emergence of gaits of a legged robot by collaboration through evolution. *Proc. Int. Symposium on Artificial Life and Robotics*, Springer Verlag.

[34] Kodjabachian, J., Meyer, J.-A. (1998) Evolution and development of neural networks controlling locomotion, gradient following, and obstacle avoidance in artificial insects. *IEEE Transactions on Neural Networks.* MISSING

[35] Rodgrigues, L., Prado, M., Tavares, T., da Silva, K., Rosa, A. (1996) Simulation and Control of Biped Locomotion - GA Optimization, *Proceedings of IEEE International Conference on Evolutionary Computation*, pp.390-395, Nagoya, Japan.

[36] de Garis, H. (1990) Genetic Programming : Evolution of Time Dependent Neural Network Modules Which Teach a Pair of Stick Legs to Walk, *9th. European Conference on Artificial Intelligence*, August 1990, Stockholm, Sweden.

[37] Mataric, M, Cliff, D. (1996) Challenges in evolving controllers for physical robots. *Robotics and Autonomous Systems*, **19**: 67-83.

[38] www.mathengine.com

[39] MathEngine SDK 1.x Code Book, MathEngine PLC.

[40] van de Panne, M., Fiume, E. (1993) Sensor-Actuator Networks. *Computer Graphics Proceedings, Annual Conference Series, (SIGGRAPH '93)*, pp. 335-342.

[41] Calabrese, R. (1998) Cellular, synaptic, network, and modulatory mechanisms involved in rhythm generation, *Current Opinion in Neurobiology* **8**: 710-717.

[42] Beer, R.D., Gallagher, J.C. (1992) Evolving dynamic neural networks for adaptive behaviour. *Adaptive Behaviour* **1**: 91-122.

[43] Bullock, S. (1999) Are Artificial Mutation Biases Unnatural?, *Proceedings of the 5th European Conference on Artificial Life (ECAL '99)*, D. Floreano, J-D. Nicoud, F. Mondada (eds.), pp. 64-78, Springer Verlag.

[44] Hillis, W. D. (1992). Co-evolving parasites improve simulated evolution as an optimization procedure. In Langton, C. *et al.* (eds.) *Artificial Life II,* Addison-Wesley. pp. 313-324.

[45] Watson R., Hornby, G., Pollack, J. (1998) Modelling Building-Block Interdependency, *PPSN V*, Eiben, Back, Schoenauer, Schweffel (eds.), Springer Verlag.

[46] Getting, P. and Dekin, M. (1985) Tritonia Swimming. A model system for integration within rhythmic motor systems. In Selverston A.I. (ed.). *Model Neural Networks and Behaviour*, pp. 3-19.

[47] Rendel, M. (2001) The Simulated Evolution of Central Pattern Generators, *Undergraduate Project*, University of Oxford.

[48] Bodenheimer, B., Shlevman, A. V., Hodgins, J. K. (1999) The Effects of Noise on the Perception of Animated Human Running, In Magnenat-Thalmann, N., Thalmann, D. (eds.), Computer Animation and Simulation '99, Eurographics Animation Workshop, pp. 53-63, Springer Verlag, Wien.

[49] Skinner, F. K., Mulloney, B. (1998) Intersegmental coordination in invertebrates and vertebrates. *Current Opinion in Neurobiology* **9**:725-732.

[50] Grillner, S., Wallen, P., Viana di Prisco, G. (1990) Cellular network underlying locomotion as revealed in a lower vertebrate model: Transmitters, membrane properties, circuitry, and simulation. *Cold Spring Harbor Symp. Quant. Biol.* **55**: 779-789.

[51] Ijspeert, A., Hallam, J., Willshaw, D. (1998) From lampreys to salamanders: evolving neural controllers for swimming and walking, In R. Pfeifer *et al.* (Eds) *From Animals to Animats 5*, pp. 390-399, MIT Press.

[52] Adolfsson, J., Dankowicz, and A. Nordmark (1998) 3-d stable gait in passive bipedal mechanisms, *Proceedings of 357 Euromech, 1998.*