Neurocomputing 86 (2012) 86–96

Contents lists available at SciVerse ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Biped locomotion control with evolved adaptive center-crossing continuous time recurrent neural networks

José Santos *, Ángel Campo

Department of Computer Science, University of A Coruña, Campus de Elviña s/n, 15071A Coruña, Spain

ARTICLE INFO

Article history:
Received 27 June 2011
Received in revised form
5 January 2012
Accepted 9 January 2012
Communicated by J. Torres
Available online 24 February 2012

Keywords:
Continuous time recurrent neural networks
Genetic algorithms
Evolutionary robotics

ABSTRACT

We used center-crossing continuous time recurrent neural networks as central pattern generator controllers in biped robots, together with an adaptive methodology to improve the ability of the recurrent neural networks to produce rhythmic activation behaviors. The parameters of the recurrent networks are adapted or modified in run-time to reach the center-crossing condition, so the nodes get close to the most sensitive region to their input. This facilitates the evolution of the networks that act as central pattern generators to control biped structures. The robustness of the adaptive networks to produce rhythmic activation patterns was checked as well as the improvements and possibilities this adaptation may add.

1. Introduction

Central pattern generators (CPGs) are pulsating collections of neurons in the spine that can produce rhythmic patterns of neural activity without receiving rhythmic inputs. They can be building blocks for the animals' locomotion neural circuits [11], as well as for other rhythmic activities (such as breathing or chewing). Büschges states that it is well established that "locomotor patterns result from the interaction between central pattern generating networks in the nervous system, local feedback from sensory neurons about movements and forces generated in the locomotor organs, and coordinating signals from neighboring segments or appendages" [4].

The work of Ijspeert [11] reviews different alternatives to define CPGs. Here we focus on the control of biped robots. Bipedal walking is a difficult task due to its highly unstable dynamic behavior [7]. Legged locomotion is characterized by cyclic activity of the limbs, and the defining feature of CPGs is a high degree of recurrence, which greatly biases the dynamics of the system toward cyclic activation patterns [22].

As McHale and Husbands [19] pointed out, although the characteristic equations associated with a specific network are a compact description of the network, we are as yet unable to predict from these equations the dynamic characteristics of the network when it is embodied in an environmental agent. Evolutionary robotics provides an alternative to the hand design of robot controllers, especially for autonomous robots acting in uncertain and noisy domains. Artificial evolution is used to automate the design procedure of these controllers, changing the focus from deciding how adaptive behaviors are to be generated to deciding what behaviors are generated [9,21]. Within this methodology, we used an adapted genetic algorithm to automatically obtain the neural controllers that will act as CPGs in the control of biped structures and for locomotion behaviors, where such controllers will use run-time adaptation of their parameters.

As indicated by Aoi and Tsuchiya [1], steady walking of a biped robot implies a stable limit cycle in the state space of the robot so, in the design of a locomotion control system, there are three problems associated with achieving a stable limit cycle: the design of the motion of each limb, interlimb coordination, and posture control. In addition to these problems, when environmental conditions change or disturbances are added to the robot, there is the added problem of obtaining robust walking. The aim of the adaptive methodology presented here is to solve these problems.

This paper is organized as follows. In Section 2 we summarize previous works on the control of biped robots, focusing on those using continuous time recurrent neural networks as controllers. In Section 3 we describe the center-crossing condition in such type of recurrent networks, which facilitates the design of neural oscillators. In Section 4 we explain the adaptive methodology developed, which starts from such center-crossing condition and which enables to easily obtain rhythmic oscillators for the biped locomotion behavior. Sections 5 and 6 describe the methods used, that is, the software simulation environment, the robot designs, and the evolutionary algorithm used. Section 7 exposes the results, describing with examples the potentials of our adaptive controllers. Finally, Section 8 indicates the conclusions and intended future work.

* Corresponding author.
E-mail address: santos@udc.es (J. Santos).

0925-2312/$ - see front matter © 2012 Elsevier B.V. All rights reserved.
doi:10.1016/j.neucom.2012.01.009

2. Previous work

Several models have been used to implement CPGs, including vector maps, systems of coupled oscillators and connectionist models [11]. As Ijspeert [11] indicates, with the neural models, the focus was on how rhythmic activity is generated by the network properties. Along this line, Beer [2] introduced the model of continuous time recurrent neural networks (CTRNN), one of the most used to represent CPGs. As Beer indicates, “CTRNNs is a class of neural models that is simple but dynamically universal” [3]. In [3] the author performed a study of the global parameter space structure of CTRNNs, estimating the probability of encountering different kinds of dynamic behaviors in CTRNN parameter space regions.

The work of McHale and Husbands [19] presents a comparative study of four types of neural networks to synthesize bipedal control systems: the conventional CTRNN [2], the center-crossing CTRNN [16] (explained in the next section), the plastic neural network (PNN) [8] and the GasNet developed by the authors. The PNNs incorporated run-time learning through Hebbian rules and GasNets were inspired by the action of nitric oxide as a neuromodulator. The authors’ interest was to evolve networks capable of achieving locomotion with a simulated biped. Of the 14 distinct networks tested (variants of those types), CTRNNs were shown to have advantages in most of the cases. Continuous time recurrent neural networks were able to attain higher average fitness, although GasNets obtained the highest fitness peak with cyclic locomotion.

However, as exposed in [16] with a statistical analysis, if we use an evolutionary method to obtain networks (CTRNNs) capable of producing rhythmic behavior, the probability that a random network population of a moderate size (100 individuals) contains one or more CTRNNs exhibiting pulse behavior is rather small. Furthermore, Reil and Husbands [22] evolved CTRNNs for a simulated biped. They showed that there is no need for proprioceptive information to control stable straight-line bipedal walking. They also reported that the fraction of evolutionary runs leading to stable walkers was only 10% (even allowing backward walking controllers). Moreover, the authors, in a second stage, provided the robot with two simulated ears and evolved controllers to approach a sound sensor so the robots could achieve directional walking. They experimented with incremental evolution, as the previously evolved weights were clamped and only the weights from the sensory unit were evolved.

Mathayomchan and Beer [16] additionally experimented with the inclusion of the so-called center-crossing networks. These networks can produce oscillatory behaviors in an easier way, since their nodes’ parameters are tuned so that the neurons operate in the most sensitive region. When they generated 10,000 random center-crossing CPGs and 10,000 completely random CPGs, they found that 26.6% of the center-crossing circuits produced oscillations, while for random circuits only 1.2%. When they used an evolutionary algorithm to search for control oscillators of a simple biped robot, they demonstrated that relative to a random initial population, seeding an initial population of an evolutionary search with center-crossing networks improved both the frequency of pulse-circuits occurring in a population and the speed with which high fitness pulse-circuits evolved.

Vaughan et al. [24] used two symmetrical feed-forward continuous time neural networks to control the movements of the legs in a biped structure. The networks received as input, among others, accelerations in the three axes and different joint angles of the robot, together with a time signal provided by a CPG that regulated the walking gait. The goal of the authors with these networks was the control of passive walkers for efficiency: by using passive dynamics and compliant tendons the biped conserved energy while walking on a flat surface. Izquierdo and Buhrmann [13] synthesized single circuits that performed two different behaviors: orientation to sensory stimuli (chemotaxis) with a simulated Khepera and legged locomotion in a simulated one-legged insect walking agent. The authors’ main interest was to demonstrate that small fully interconnected networks (CTRNNs) can solve the two tasks with the ability to switch between the behaviors from the interactions of the circuit’s neural dynamics, its body and environment. The networks were used as reflexive pattern generators, where all the neurons received sensory perturbations: from the leg angle during walking and from the proximity to the food during chemotaxis.

Meyer et al. [20] used the leaky-integrator model of the CTRNNs for calculating the value of the nodes of their neural controllers [14]. Their work was more focused on the developmental process of the phenotype from the genotype and on the incremental evolution of behavior modules. For example, the authors presented examples of obstacle-avoidance and locomotion for a hexapod robot, in which a first module generated a straight-line locomotion behavior, and an additional controller, obtained in a second stage, modulated the leg movements secured by the first controller making it possible for the robot to turn in the presence of an obstacle and avoid it.

With other alternatives different from CTRNNs, the neural oscillator proposed by Matsuoka [18] was widely used to model the firing rate of two mutually inhibiting neurons, described by a set of differential equations. This model was used in robotic applications to achieve designated tasks involving rhythmic motion which requires interactions between the system and the environment. For example, Matsubara et al. [17] proposed a learning framework for a CPG-based biped locomotion controller using a policy gradient reinforcement learning method for a 5-link biped robot. Their CPG-based controller was composed of a neural oscillator model (Matsuoka’s [18] model) and a sensory feedback controller which mapped the states of the robot to the input to the neural oscillator model. Matsuoka’s oscillator model was also used by Endo et al. [6], applying two oscillators to control the leg movement of a 3D robot model. One oscillator controlled the position of both legs in vertical direction, while the other controlled the position of both legs along the forward direction. They also implemented it on the QRIO robot. However, it is difficult to determine the CPG parameter values in this oscillator model for various robots and environments, since there is no general design principle to determine the parameter values.

Hein et al. [10] used the discrete time dynamics of a two neuron network oscillator to generate a core oscillation for the control of the joints of a biped. An additional layer of weights connects this basic oscillator to the motor neurons which control the joints. They focused their work on the transfer of the evolved controllers obtained in their simulation to real robots. von Twickel and Pasemann [25] evolved neuromodules for single morphologically distinct legs of a simulated hexapod walking machine. They demonstrated how small reflex-oscillators, which rely on the sensorimotor loop and hysteresis effects, generate effective locomotion. An analysis of the controllers showed that sensory inputs and dynamic effects, like hysteresis, play a major role in the generation of walking patterns and for robust behavior under changing environmental conditions, where the controllers worked without a CPG. Previously, Wischmann and Pasemann [26] used a recurrent neural controller to enable a simulated walking device to walk on a flat surface with minimal energy consumption. The applied evolutionary algorithm fixed neither the size nor the structure of the recurrent network. Such controller network used sensory information, as a foot contact sensor provided information to the network.

Ishiguro et al. [12] focused their work on the neuromodulation of a neural circuit that controlled a biped robot. The modulation, through the diffusion of neuromodulators, allowed the dynamic change of the circuit properties according to the current situation in real time, so the robot was able to cope with environmental perturbations. In the work of Manoonpong and Wörgötter [15] the focus was on experimental studies for the application of the efference copy (motor commands copied within the central nervous system) for improving locomotion control and determining terrain condition changes using a real biped robot.

The main problem or drawback in these works is the difficulty to determine the CPG parameter values to obtain oscillator networks. The center-crossing condition, explained next, was introduced in CTRNNs to facilitate that these networks can oscillate, although the condition does not guarantee such desirable behavior. So, our objective is focused on the improvement of the conditions to obtain networks with dynamic and oscillatory behavior. For that aim, we began with the center-crossing definition and we added an adaptation methodology, for that purpose, in the center-crossing CTRNNs. To do so, we defined adaptive center-crossing CTRNNs that adapt the bias parameters of the nodes to reach, in run-time, such center-crossing condition.

3. Center-crossing continuous time recurrent neural networks

In the conventional CTRNN [2] the state of a single neuron i is computed by the following equation:

$$\tau_i \dot{y}_i = -y_i + \left[\sum_{j=1}^{N} w_{ji} \sigma(y_j + \theta_j) \right] + I_i \quad (1)$$

where y_i is the state of neuron i, τ_i is a time constant, w_{ji} is the weight of the incoming connection from neuron j, σ is the sigmoid activation function, θ_i is the bias term or firing threshold of the node, and I_i is an external input.

As defined in [16], in a center-crossing CTRNN the null surfaces of all neurons intersect at their exact centers of symmetry. The null surface of a neuron is where the sum of the neuron bias and all the synaptic inputs is 0. This ensures that each neuron's activation function is centered over the range of net inputs that it receives.

Using a sigmoid activation function in the CTRNNs, a neuron has a firing frequency of 0.5 at its null surface ($\sigma(0)=0.5$), that is, center-crossing networks have neurons that on average have firing frequencies around this value. Hence, the center-crossing condition occurs when the neuron biases of all neurons are set to the negative of the sum of the input weights divided by 2:

$$\theta i = -\frac{\sum_{j=1}^{N} w_{ji}}{2} \quad (2)$$

This means that the bias exactly counteracts the sum of all the synaptic inputs when the connected neurons have a firing frequency of 0.5. In other words, the nodes of such type of networks should have an average firing value of 0.5, which implies that the neurons are in states of maximum sensitivity most of the time.

Small changes in synaptic inputs around the null surface can lead to a very different neuron firing frequency. Outside this range, the change of a net input has only sparse effect on the firing frequency. The richest dynamics should be found in the neighborhood of the center-crossing networks in the search space of parameters and, as Mathayomchan and Beer [16] indicate, one would expect that an evolutionary algorithm would benefit from focusing its search there.

4. Adaptive center-crossing CTRNNs

The center-crossing networks can be viewed as networks of neurons of maximum sensitivity. Such networks are expected to exhibit a wider range of dynamic behaviors than do random networks. Nevertheless, the use of the center-crossing condition does not guarantee rhythmic behavior in the corresponding network. A simple example: If we have a six-neuron CTRNN, where the incoming weights of a particular node are all equal to 1, then the θ threshold that corresponds with the center-crossing definition will be -3. When half of the incoming nodes are not firing (values near 0) and the others are at maximum values, the node with such bias will be in the most sensitive region, so the node can easily change its activation value, and at the same time can induce changes in the other nodes that are also in the sensitive regions of their activation functions. However, even with that bias, if four or more of the input neurons are activated near the maximum values, the node will never be in the sensitive region and it will never present a change in its value. Even if the initial incoming values of the node are initialized in such a way that half of the incoming nodes are at the maximum value (or near the maximum) and half at the minimum possible value, in most cases the dynamic behavior of the center-crossing network ends up with fixed time values in the nodes. In Mathayomchan and Beer's [16] words, referring to center-crossing CTRNNs: "unless a neuron's bias is properly tuned to the range of inputs it receives, that neuron will simply saturate on or off and drop out of the dynamics".

The upper part of Fig. 1 has three examples which correspond to a center-crossing network. The figures show the phase plot of the time evolution of two nodes. As shown in Fig. 1a–c, depending on the initial values of the network nodes, the same network can present three different dynamic behaviors. In addition, most of the initializations provide fixed point attractors.

So, the values of the inputs to the nodes must be taken into account if we want to force rhythmic activation in the nodes. As Ijspeert [11] indicates "the purpose of CPG models is to exhibit limit cycle behavior, i.e., to produce stable rhythmic patterns. When this is the case, the system rapidly returns to its normal rhythmic behavior after transient perturbations of the state variables". If we want to maintain the nodes in the most sensitive regions, the bias of a node i can be adapted in run-time to get it closer to the negative value of the input it receives (sum of all the synaptic inputs), that is,

$$\theta_i(t+1) = \theta_i(t) - coef\ \ y_i(t+1) \quad (3)$$

where $coef$ is a learning or adaptation coefficient [5]. With the integration of Eq. (1), the bias is changed according to the following formula (ignoring external inputs):

$$\theta_i(t+1) = \theta i(t) - coef[\Delta_t \tau_i^{-1}(-y_i + \textstyle\sum_{j=1}^{N} w_{ji}\sigma(y_j + \theta_j)) + y_i] \quad (4)$$

where Δ_t is the time step used in the integration of Eq. (1). Hence, the bias of each node is adjusted, in each iteration of the recurrent network, to the value that defines the center-crossing condition, as it is changed towards the negative value of the incoming activation at time t. Therefore, all the nodes will be near the most sensitive regions to induce activation changes. The magnitude of the learning coefficient determines how fast that situation is obtained. The last three examples of Fig. 1 correspond to a center-crossing network when the biases are adjusted in run-time (Fig. 1d–f). In this case, independently of the initial conditions, the trajectories fall in the same temporal limit cycle and, more importantly, they all exhibit a limit cycle behavior with stable rhythmic patterns that can be useful for our purpose of coordinating the rhythmic movements of the biped limbs.

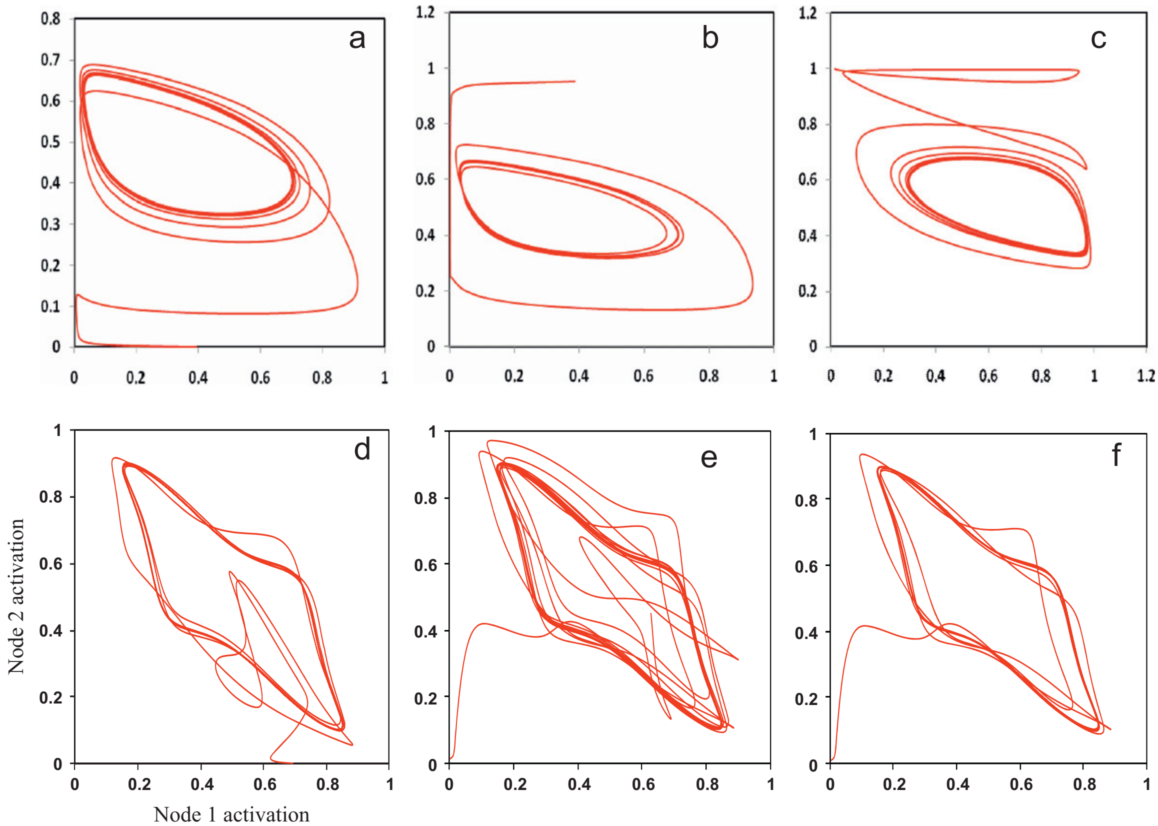

Fig. 1. Robustness against initial conditions. (a)–(c) show the phase plot of the temporal activation of two nodes of a center-crossing network when it was initialized with different activation values in the nodes. (d)–(f) show the same plot in an adaptive center-crossing CTRNN, where the biases are adjusted in run-time.

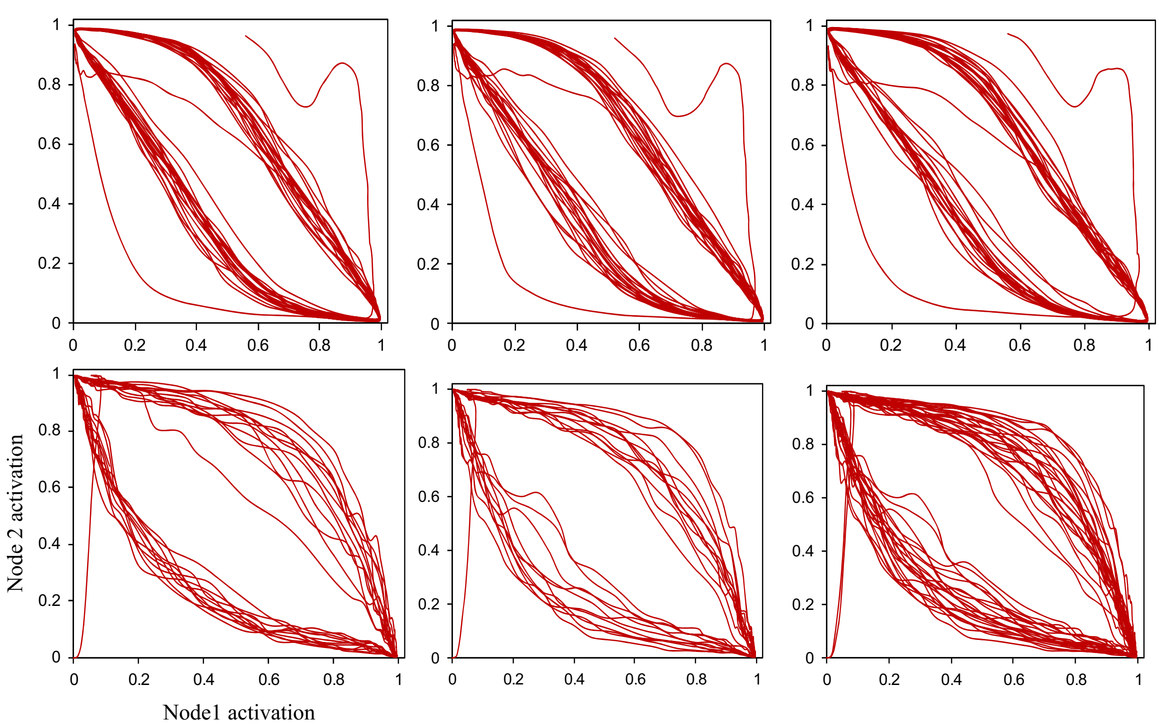

Fig. 2. Robustness against Gaussian noise. The upper figures correspond to different runs of an adaptive center-crossing CTRNN where Gaussian noise was added to the node values (σ^2=0.05). In the bottom figures Gaussian noise was added with σ^2=0.1.

With the same methodology, we can test the behavior of the adaptive CTRNNs when such networks operate under noisy conditions. Fig. 2 shows the robustness of the adaptive CTRNNs when Gaussian noise was added to the node values. In the upper illustrations a noise distortion with a variance σ^2=0.05 was added to the values of the nodes in each time iteration, while in the lower illustrations higher Gaussian noise with $\sigma^2 = 0.1$ was added. Even with such high values of σ^2, the adaptation of the biases tends to maintain the limit cycles of the network when it is iterated through time. This shows the capability of the methodology to deal with noisy environments, where the recurrent networks can behave as controllers in such situations in which the information reaching an input node is perturbed by the noise present.

5. Biped models

We used simulated bipeds in the experiments, employing the physics simulator open dynamics engine (ODE) [23] together with the *drawstuff* graphics library. The articulated structures consisted of rigid bodies connected through joints. We used two biped structures: a simple one without the upper body part and a complete body structure. Fig. 3 shows a snapshot of both structures and Fig. 4 indicates how the joints of the bipeds were controlled by the values of the neural controller. The simplest biped had two joints, each linking a hip with a leg, containing a knee joint and an ankle joint. The complete biped structure incorporated the arms and the head, with their corresponding joints, to the trunk. All the actuated joints had a degree of freedom and were simulated as torsional springs. The feet were composed of 6 consecutive arcs joined with fixed hinges, providing a concave structure in each foot.

Table 1 summarizes the main parameters for the joints in both structures. The outputs of the neural network nodes were scaled to provide a velocity that can reach the angle limits. The mass of each body part was proportional to its volume, the gravity was fixed to -9.81 m/s^2 and a time step of 0.01 s was used in the ODE simulation for each iteration in the environment. Maximum ground friction was used in the ODE simulator to avoid any possibility of the feet sleeping. The analysis of the capabilities of control using the CTRNNs, with or without the adaptive methodology, was similar with both structures, so the results will be presented in most cases with the simple biped structure.

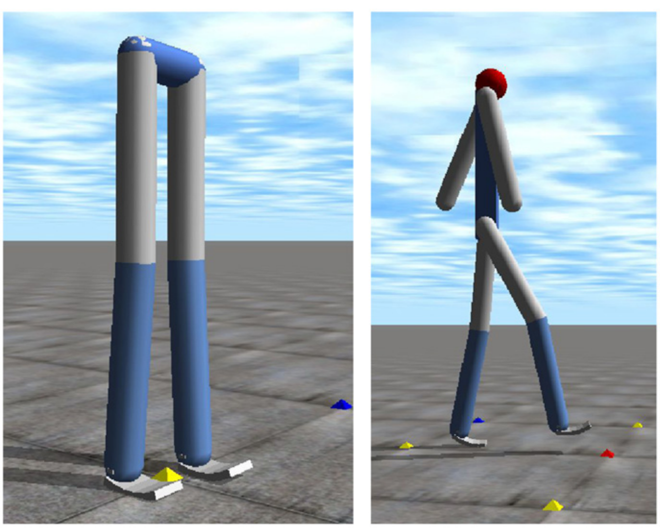

Fig. 3. Biped robot structures defined in the ODE environment.

6. Evolutionary algorithm

We used a standard genetic algorithm (GA) for the evolution of the CPGs (CTRNNs) for the locomotion behavior. Each individual of the genetic population encoded the defining parameters of the control network. In the case of simple CTRNNs, each network was codified by a vector that included the connection weights, the biases and the time constants associated to each neuron. In the case of center-crossing CTRNNs, the genotypes encoded the weights and time constants, whereas the biases were set according to Eq. (2) taking into account the encoded values of the weights. In the case of adaptive center-crossing CTRNNs, the initial biases were set as random values and modified in run-time according to Eq. (4), and again the weights and time constants were encoded in the genotypes.

Table 1
ODE parameters of the biped structures.

Head	Complete structure	Sphere, radius: 0.08 m
Trunk		Box, length: 0.1 m., width: 0.2 m., height: 0.7 m
Arms		Cylinder, height: 0.5 m., radius: 0.2 m
Angular shoulder displacement		$[-\pi/6, 0.75\pi]$
Hip dimensions	Simple structure	Cylinder, length: 0.2 m., radius: 0.05 m
Thigh and leg dimensions	Both structures	Cylinder, length: 0.5 m., radius: 0.05 m
Feet arcs		Box, length: 0.005 m., width: 0.1 m., height: 0.01 m
Angular hip displacement		$[-\pi/6, 0.75\pi]$
Angular knee displacement		$[-0.75\pi, 0]$
Angular ankle displacement		$[-0.25\pi, 0]$
g		-9.81 m/s^2
Time step in ODE simulation		0.01 s

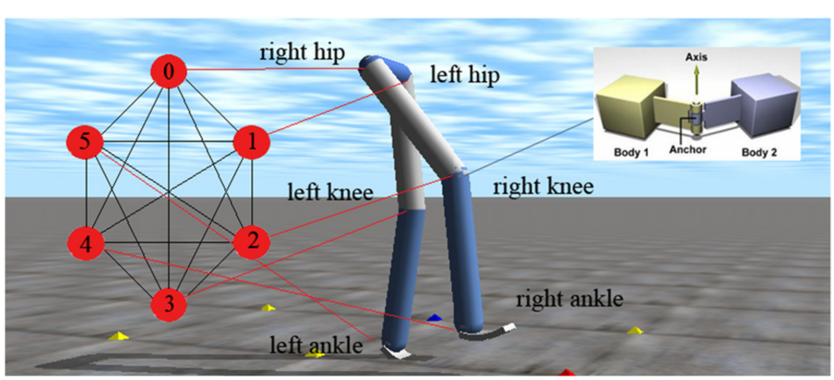

Fig. 4. CTRNNs applied to the control of the biped robots. The values of the NN nodes control the different joints of the simulated biped in ODE. The inset shows the hinge joint used at each of the contacts.

When using the adaptive methodology, we combined the global search of the GA with the run-time adaptation of the biases in each individual (network) of the genetic population. Note that this is not a Lamarckian combination, as the final values of the biases are not reverted in the genetic material of the individual, and the GA only searches for the optimal weights and time constants of the recurrent networks.

A population of 100 individuals was used in the different evolutionary runs. We used a rank-based method as selection operator: 25% of the best individuals of the population were replicated to generate the next population. As in [16], these individuals were mutated with a given probability in their genes (parameters), adding a random displacement whose magnitude was a Gaussian random variable with 0 mean and variance σ^2. We included the elitist selection, as the best individual was copied to the new population without any change. We did not use crossover operators because, as Reil and Husbands pointed out, there were "no identifiable functional units in the genotype and phenotype structure", given the epistasis present in these distributed connectionist structures, as well as the experimental evidence on the lack of efficiency of crossover in this problem domain [22].

Finally, as in most of the works mentioned, we used as fitness the distance traveled in a straight line by the biped in a given time (8 s in our experiments). Additionally, to avoid grotesque movements, a penalization was introduced when the center of gravity fell below a certain threshold (90% of the robot height) or, on the contrary, when it was above (110% of the robot height). The latter was to eliminate robots that jump to move quickly. In both cases, the fitness is set as 0, so the controller only summarizes fitness (the distance traveled) until one of such conditions occurs. This allows these inefficient controllers to progressively reach (through the genetic operators) larger distances with stable postures and movements. On the contrary to Reil and Husbands [22], we did not allow backward walking controllers.

We used fully interconnected CTRNNs as CPGs (with the possibilities commented), where all the neurons were motor neurons to control each of the six joints in the simplest biped structure, as Fig. 4 shows. This implies 48 parameters to evolve (36 connection weights, 6 biases and 6 time constants). If the biases were defined according to the central-crossing condition (or adjusted in run-time), the parameters were 42. With the complete biped structure, with 9 joints, there were 99 parameters to evolve (81 connection weights, 9 biases and 9 time constants) and 90 when center-crossing CTRNNs (with or without the adaptive methodology) were used.

In both cases, the connection weights and biases were constrained to lie in the range ± 16 (as in [16] and [22]), and the time constants were constrained in the range [0.5, 5], as in [22]. The time step for the integration of Eq. (1) was 0.1 s. Finally, to avoid the transient perturbations at the beginning of the temporal evolution of the network, each network controller was iterated a given number of steps (100) before taking the control of the biped joints. Figs. 1 and 2 show how, using the adaptive center-crossing CTRNNs, the perturbations progressively disappear until the network reaches its limit cycle. The chosen value of 100 steps allows a sufficient number of steps to eliminate the perturbations with the *coef* values used.

7. Results

7.1. Comparison of CTRNNs with different strategies

We tested the difficulty of obtaining CTRNNs, which act as CPGs, to provide rhythmic behavior, as well as the fitness obtained, with three strategies: (1) random genetic populations with random defining parameters of the CTRNNs (encoded in the genotype), in which the biases do not have any restriction (except for the range ± 16), (2) initial populations where the encoded weights and time constants were random, and the biases were defined according to the center-crossing condition (Eqs. (2) and (3)) and finally random genetic populations with random defining parameters (encoded weights and time constants) but the random initial biases were adjusted in run-time according to Eq. (4) (adaptive center-crossing CTRNNs). Fig. 5 summarizes the results of the evolutions with the GA using the simple biped structure. The quality evolutions were the average result of 50 different runs of the GA with different initial populations. The number of individuals was 100 for all tests (as in [16]). Table 2 summarizes the results at the end of the evolutionary processes (120 generations).

Mathayomchan and Beer [16] tested the seeding of the initial population with center-crossing CTRNNs on a walking task in a simple legged body. The authors demonstrated that center-crossing seeded searches evolved more reliably high-fitness circuits. They obtained the greatest difference, with regard to an initial random population, with a mutation variance σ^2 of 0.05, while the difference decreased with increasing values of σ^2.

The same σ^2 value of 0.05 was used for the random displacement of the mutation operator in the evolutions of Fig. 5. As expected, with the seeded initial populations with center-crossing NNs, the evolutionary algorithm obtained better CTRNNs that

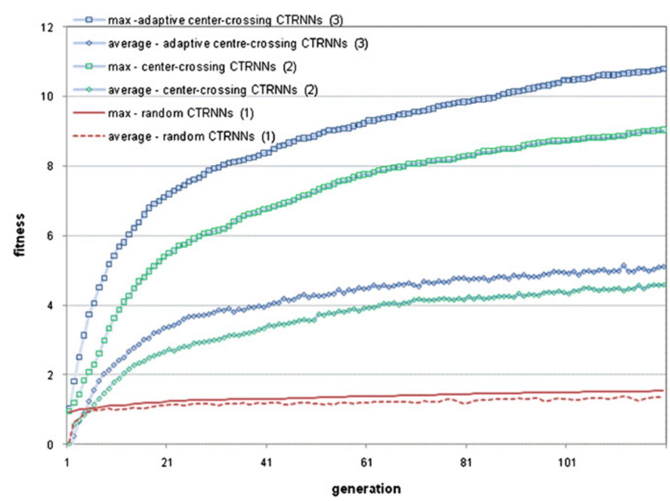

Fig. 5. Evolution of the quality of the best individual and the average quality of the population for three different conditions: random CTRNNs (1), initial center-crossing CTRNNs (2) and run-time adaptive center-crossing CTRNNs (3). Fitness is defined as the distance traveled in a straight line by the biped in a given time (8 s), as detailed in Section 6. The quality curves are an average of 50 different runs of the GA.

Table 2
Values obtained in the evolutions with the three strategies: (1) CTRNNs with random initial parameters, (2) initial center-crossing CTRNNs, (3) adaptive center-crossing CTRNNs.

	Strategy	Average value of the 50 runs	Standard deviation
Fitness of best individual	1	1.538	0.498
	2	9.072	2.795
	3	10.795	2.207
Average fitness of population	1	1.282	0.436
	2	4.558	1.758
	3	5.070	1.282

provided the required cyclic behavior. Average quality indicates a larger number of possible networks with such behavior in comparison to a random population of CTRNNs. Once oscillations are discovered with center-crossing CTRNNs, the evolutionary algorithm can fine-tune them into highly fit CPGs by matching the amplitude, period and phase of the oscillation to the characteristics of the body model.

In the third strategy (adaptive CTRNNs), with the use of the run-time adaptive biases, all the individuals of the genetic population presented rhythmic behavior. Most of such cyclic behaviors were usually not adequate for locomotion control, as the average fitness at the initial generation indicates, but the genetic algorithm had more NNs to fine-tune, so it obtained higher fitness controllers and in fewer generations.

Figs. 6 and 7 show, in the upper part, several steps of the swing and stand phases of the simple robot legs of two evolutionary obtained CTRNNs which used run-time bias adaptation. In Fig. 6 the robot was on a flat surface whereas the swing and stand phases in Fig. 7 were obtained with the biped on a slope of 8.6° ($coef$=0.01 in both cases). The lower parts of the figures show one

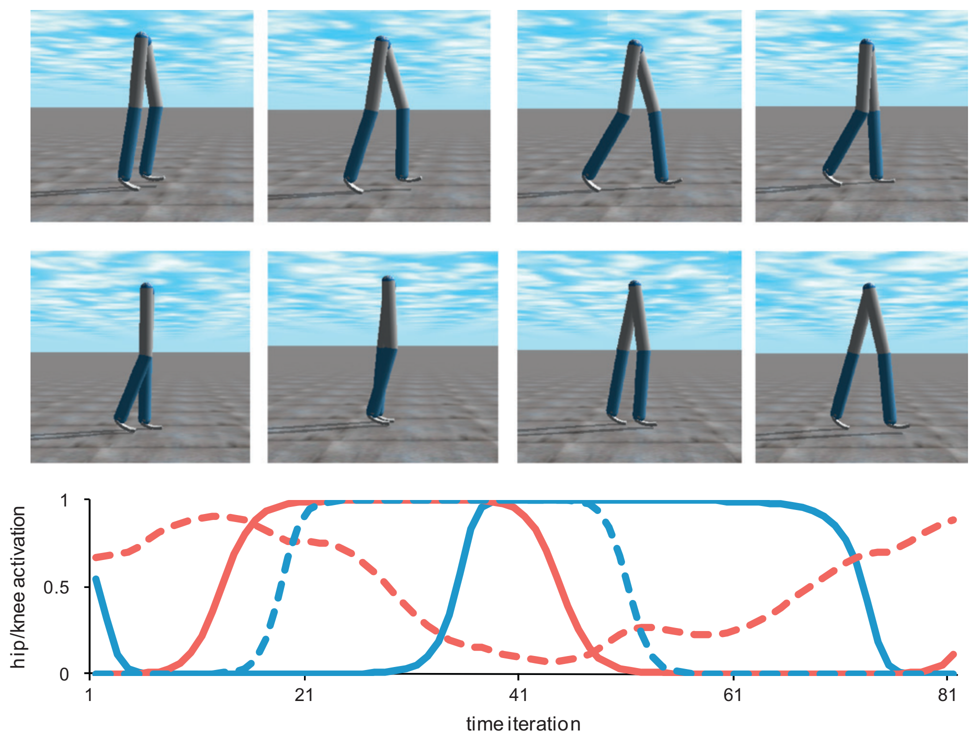

Fig. 6. Several steps in the swing and stand phases of the locomotion behavior on a flat surface with the simple biped structure. The bottom graph shows the rhythmic activation of the nodes that directly control the hip joints (continuous lines) and knee joints (dashed lines) during the necessary time iterations for the swing of both legs. See video vFig_6 corresponding to these graphs, and video vFig_6_massX5 when the mass of the biped components was multiplied by 5, supplementary material).

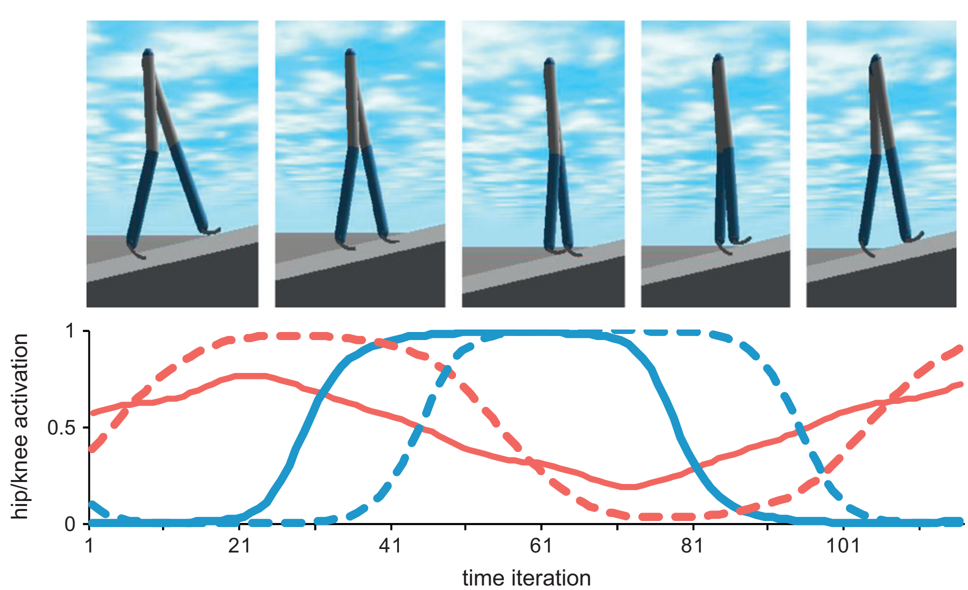

Fig. 7. Several steps in the swing and stand phases of the locomotion behavior on a slope (8.6°). The bottom graph shows the activation of the nodes that control the hip joints (continuous lines) and knee joints (dashed lines) (see video vFig_7, supplementary material).

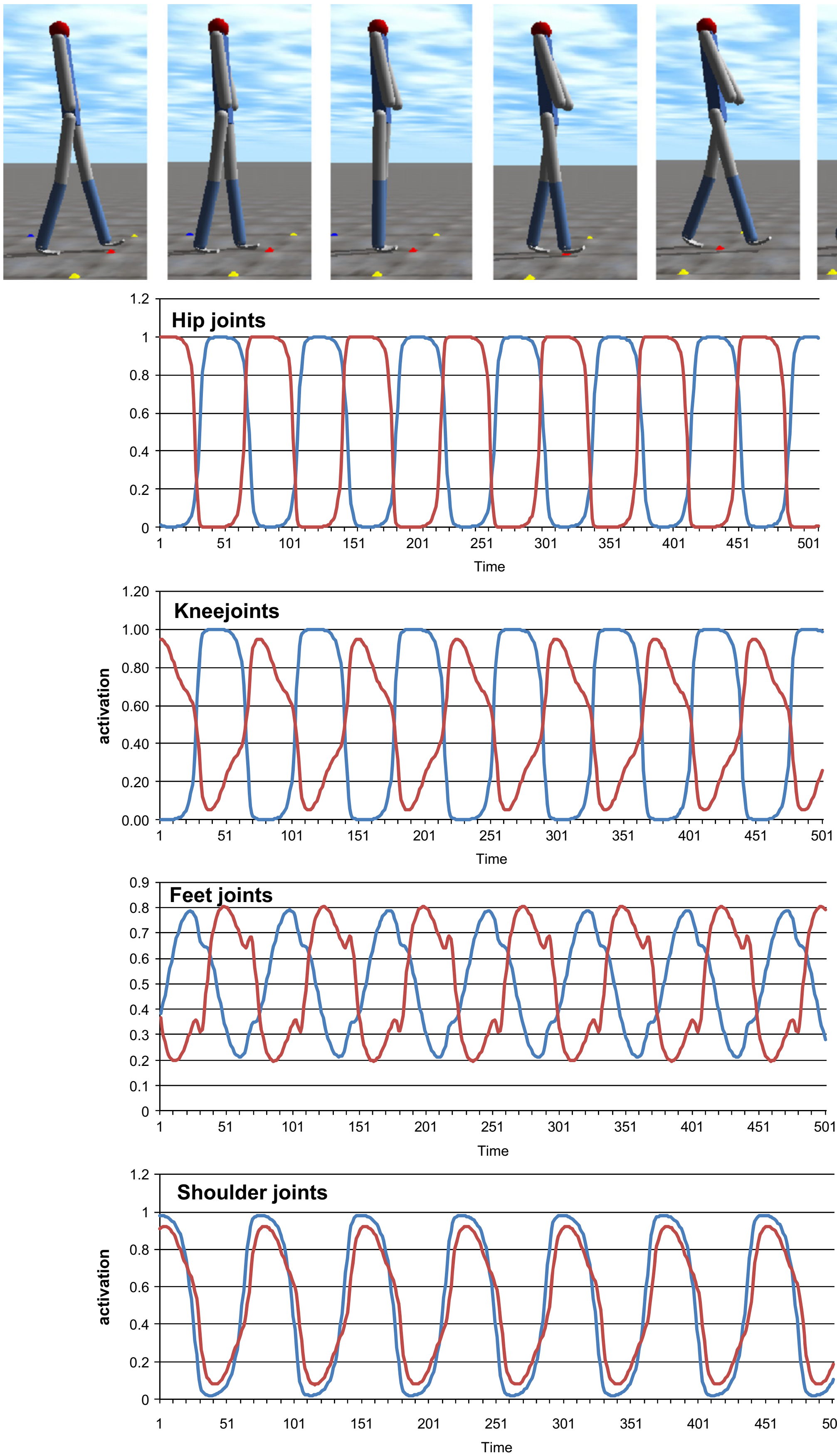

Fig. 8. Several steps in the swing and stand phases with the complete biped structure. The bottom graphs include the activation of the nodes that control the joints of the hips, knees, feet and shoulders (see video vFig_8 corresponding to these graphs, and video vFig_8_massX2 when the mass of the biped components was multiplied by 2, supplementary material).

cycle of the repetitive activation of the nodes that control the hip joints (continuous lines) and the knee joints (dashed lines), defining the swing and stand phases (the activation of the ankles are not shown for clarity). The cycle includes the time iterations that are needed for the swing of both legs. In both cases the evolutionary algorithm plays with the free parameters to obtain rhythmic activations with different time phases between the hip and knee joints of both sides and to obtain the best posture on different surfaces. For example, for slope, in the GA population, there are individuals that, when decoded in the corresponding neural controller, try to climb the slope with a posture which is not perpendicular to the slope surface or to the level surface, logically being more unstable. As there are adaptive center-crossing CTRNNs that are able to maintain the bipeds in more stable postures, the GA discovers the necessary parameters of the controllers for more stable locomotion behaviors.

The leg phases in Fig. 8 were obtained with the complete robot on a flat surface (*coef*=0.005). The robot needs 76 time iterations to complete the swing and stand phases in both legs. The upper part of the figure only shows snapshots to appreciate the phases in one leg, whereas the lower part shows the activation across several iterations to show the rhythmic activation provided by the network.

Finally, as the mass of the biped components was proportional to their volumes, we tested the capability of the adaptive center-crossing CTRNNs when the weight of the biped was increased. In the case of the simple biped, the adaptive controllers were able to achieve stable walkers even with high increases in the mass of the different components, until an increase factor of 10. In the case of the complete biped, the increase in the mass of the upper parts had a greater effect on the stable locomotion, because the evolved controllers were not able to obtain stable walkers when the mass of all the components was multiplied by values greater than 2. With this increase, it is difficult to obtain stable locomotion behavior. In the online version the videos of Figs. 6–8 are included (videos vFig_6, vFig_7 and vFig_8), together with videos when the mass was increased: the simple biped behavior when the mass was multiplied by a factor of 5 (vFig_6_massX5, *coef*=0.01) and a complete biped locomotion behavior when the mass of all the components was multiplied by a factor of 2 (vFig_8_massX2, *coef*=0.005).

Supplementary material related to this article can be found online at doi:10.1016/j.neucom.2012.01.009.

7.2. Influence of the adaptation coefficient on the locomotion behavior

The coefficient for bias adaptation of Eq. (4) determines how fast each node is set to the center-crossing condition. So, higher coefficient values can force cyclic behaviors with shorter periods, and vice versa. Hence, the rhythm of the locomotion behavior can be adjusted dynamically, providing a form of external control. We checked this possibility by obtaining the locomotion controllers using different values of the parameter *coef*.

Fig. 9 shows the activation values of the nodes which control the hip and knee joints of the simple biped robot, for three values of the parameter: 0.005, 0.01 and 0.1. The figure shows how the rhythmic activation acquires a higher frequency of activation with greater values of the parameter *coef*, as expected, given a higher value implies that the nodes are closer to the center-crossing condition in each iteration of the NN. With a low adaptation coefficient, the values of the biases of the nodes are changed more slowly in the direction of the center-crossing condition, so such condition is obtained with more NN temporal iterations, resulting in activation patterns that, on average, present higher periods of rhythmic activation. It is important to notice that, although this is the tendency with the *coef* parameter, the adaptive CTRNNs obtained by the genetic algorithm can present different frequencies of activation because the evolutionary algorithm can play with all the parameters, especially with

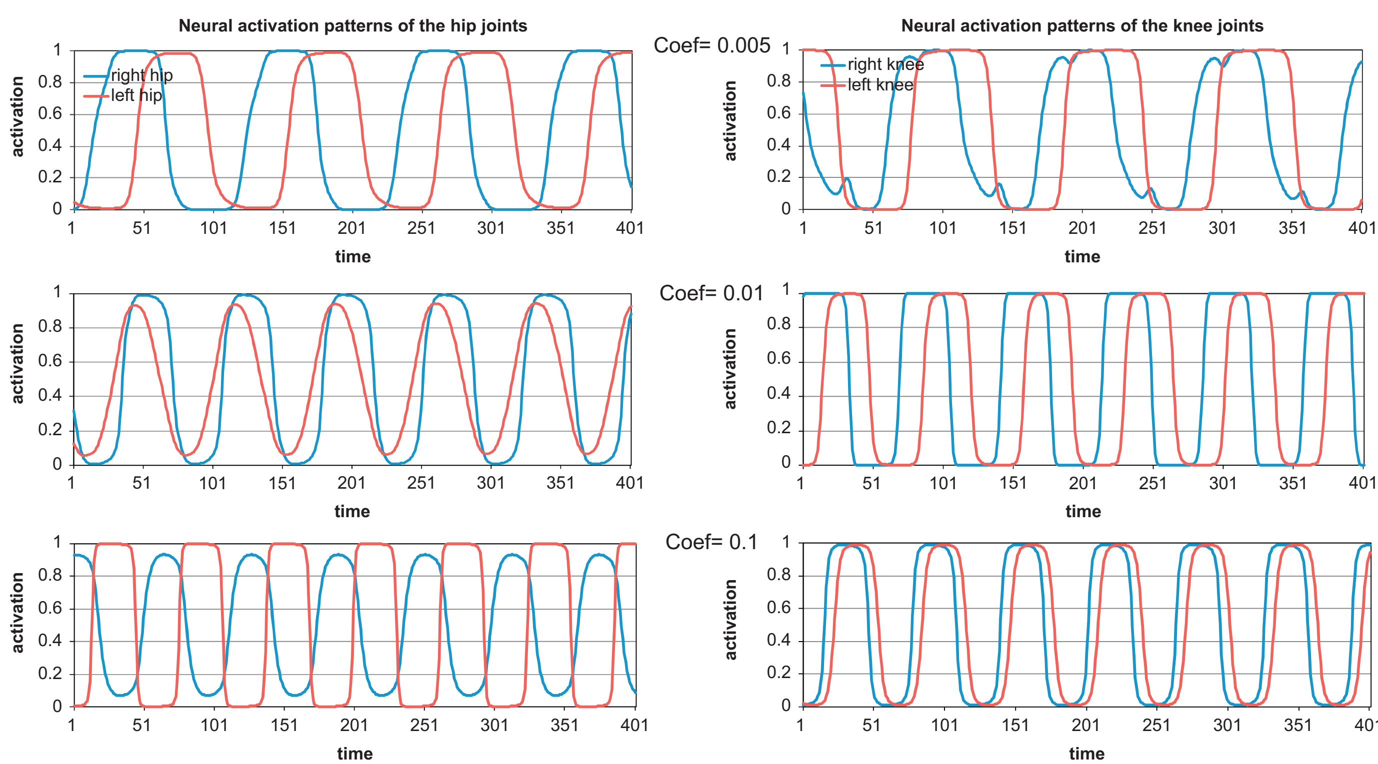

Fig. 9. Activation patterns in the nodes that directly control the hip and knee joints, using three values for the bias adaptation coefficient: 0.005 (upper figures), 0.01 (middle figures) and 0.1 (bottom figures) (see videos vFig_9_coef_0.005, vFig_9_coef_0.01 and vFig_9_coef_0.1, supplementary material).

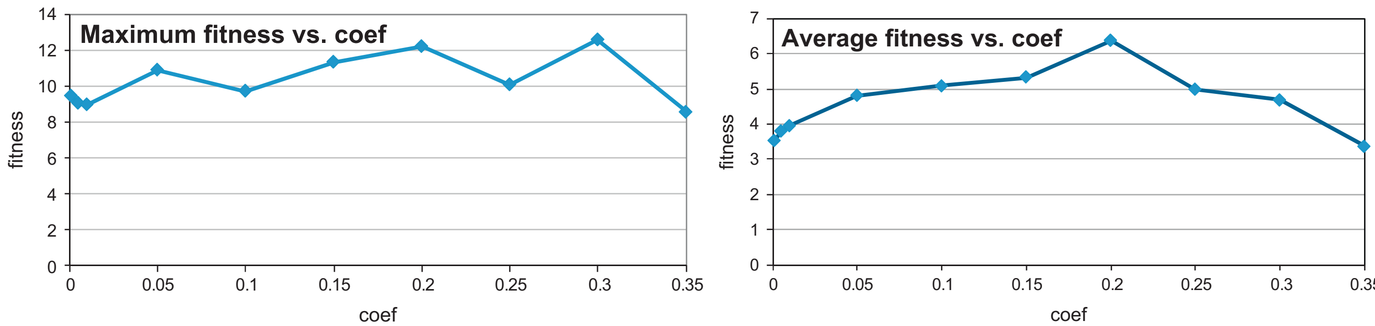

Fig. 10. Fitness of best individual and average fitness of the population obtained with different values in the parameter *coef* for the bias adaptation. The values are the average of 10 runs of the evolutionary algorithm with different initial populations.

the time constants, to obtain controllers with different rhythmic patterns.

Supplementary material related to this article can be found online at doi:10.1016/j.neucom.2012.01.009.

This is summarized in Fig. 10, which shows the fitness (distance traveled) of the locomotion controllers with different values of the parameter *coef* in the interval commented. The fitness values are an average of 10 different runs of the evolutionary algorithm using the corresponding value of *coef* in the neural controllers. Each run used a population of 100 individuals across 120 generations.

Fig. 10 indicates that, for values lower than 0.05, the adaptation to obtain the center-crossing condition has a low influence on the NN behavior, so it is more difficult to obtain a weight configuration providing rhythmic behavior, as seen in the lower value of the average fitness with *coef* values lower than 0.05. With *coef* values higher than 0.2, the NNs quickly obtain the center-crossing condition, with high frequency activation patterns, so most of the robots present locomotion behaviors that are not stable. Despite this, by the GA, a robot which presents a locomotion behavior with high-frequency activations can be discovered. These activations imply high-paced short steps, so a robot with high fitness can appear, as is the case with $coef=0.3$. Nevertheless, the average quality clearly indicates that the tendency is to obtain worse robots with *coef* values higher than 0.2, as the average quality evolution indicates.

8. Conclusions

We used adaptive CTRNNs as controllers in simulated bipeds. We used simulated evolution by means of a genetic algorithm to obtain the optimized networks for the required locomotion behavior. The adaptive methodology started from the center-crossing definition. Such center-crossing condition improves the frequency with which oscillatory circuits evolve, but, as we reasoned in the article it is not a sufficient condition to obtain oscillatory recurrent neural networks, as the condition takes into account only the static configuration of the connection weights. As Mathayomchan and Beer indicate "oscillations are essential to high performance locomotion" [16], so our work was focused on the improvement of the conditions to obtain such networks with dynamic and oscillatory behavior. In our adaptive methodology the recurrent neural controllers change the bias parameters of the nodes to obtain the center-crossing condition taking into account a dynamic view of the network, using the input information to a node in each temporal iteration instead of the static weight configuration.

The use of the center-crossing condition with the incorporation of adaptive biases allowed a faster evolution of the networks with the necessary rhythmic activation patterns. In addition, the controllers presented higher fitness with respect to non-adaptive center-crossing networks. Moreover, the value of the adaptation coefficient can control the period of the required activation behavior, so it has a modulation capability of the locomotion behavior obtained. Hence, the next step in our work is the modulation of network rhythmic activation through the bias adaptation methodology, so the network will be able, for example, to automatically adjust its cyclic behavior to the current surface it detects. Moreover, we will use cooperative co-evolution between the adaptive networks and several aspects of the morphology, such as the dimensions of the legs and feet components, to obtain the controllers in difficult surfaces such as stairs.

Acknowledgments

This work was funded by the Ministry of Science and Innovation of Spain through project TIN2011-27294.

References

[1] S. Aoi, K. Tsuchiya, Locomotion control of a biped robot using nonlinear oscillators, Auton. Robot. 19 (2005) 219–232.
[2] R. Beer, On the dynamics of small continuous time recurrent neural networks, Adapt. Behav. 3 (4) (1995) 469–509.
[3] R. Beer, Parameter space structure of continuous recurrent neural networks, Neural Comput. 18 (2006) 3009–3051.
[4] A. Büschges, Sensory control and organization of neural networks mediating coordination of multisegmental organs for locomotion, J. Neurophysiol. 93 (2005) 1127–1135.
[5] A. Campo, J. Santos, Evolution of adaptive center-crossing continuous time recurrent neural networks for biped robot control, in: Proceedings European Symposium on Artificial Neural Networks (ESANN 2010), Bruges (Belgium), 2010, pp. 535–540.
[6] G. Endo, J. Nakanishi, J. Morimoto, G. Cheng, Experimental study of a neural oscillator for biped locomotion using QRIO, in: IEEE International Conference on Robotics and Automation, Barcelona (Spain). 2005, pp. 598–604.
[7] D. Floreano, C. Mattiussi, Bio-Inspired Artificial Intelligence, MIT Press, 2008.
[8] D. Floreano, F. Mondada, Evolutionary neurocontrollers for autonomous mobile robots, Neural Networks 11 (1998) 1461–1478.
[9] I. Harvey, E.A. Di Paolo, E. Tuci, R. Wood, M. Quinn, Evolutionary robotics: a new scientific tool for studying cognition, Artif. Life 11 (2005) 79–98.
[10] D. Hein, M. Hild, R. Berger, Evolution of biped walking using neural oscillators and physical simulation, in: U. Visser, et al.et al.(Eds.), RoboCup 2007, vol. 5001, LNAI, 2008, pp. 433–440.
[11] A.J. Ijspeert, Central pattern generators for locomotion control in animals and robots: a review, Neural Networks 21 (4) (2008) 642–653.
[12] A. Ishiguro, A. Fujii, P. Eggenberger, Neuromodulated control of bipedal locomotion using a polymorphic CPG circuit, Adapt. Behav. 11 (1) (2003) 7–17.
[13] E. Izquierdo, T. Buhrmann, Analysis of dynamical recurrent neural networks evolved for two qualitatively different tasks: walking and chemotaxis, in: S. Bullock, J. Noble, R. Watson, M.A. Bedau (Eds.), Artificial Life XI: Proceedings of the Eleventh International Conference on the Simulation and Synthesis of Living Systems, MIT Press, Cambridge, MA, 2008, pp. 257–264.
[14] J. Kodjabachian, J.-A. Meyer, Evolution and development of neural controllers for locomotion, gradient-following, and obstacle-avoidance in artificial insects, IEEE Trans. Neural Networks 9 (5) (1998) 796–812.

[15] P. Manoonpong, F. Wörgötter, Efference copies in neural control of dynamic biped walking, Robot. Auton. Syst. 57 (2009) 1140–1153.
[16] B. Mathayomchan, R. Beer, Center-crossing recurrent neural networks for the evolution of rhythmic behavior, Neural Comput. 14 (2002) 2043–2051.
[17] T. Matsubara, J. Morimoto, J. Nakanish, M. Sato, K. Doya, Learning feedback pathway of CPG-based controller using policy gradient, Robot. Auton. Syst. 54 (11) (2006) 911–920.
[18] K. Matsuoka, Sustained oscillations generated by mutually inhibiting neurons with adaptation, Biol. Cybern. 52 (1985) 367–376.
[19] G. McHale, P. Husbands, GasNets and other evolvable neural networks applied to bipedal locomotion, in: From Animals to Animats 8—Proceedings of the Eighth International Conference on the Simulation of Adaptive Behavior, 2004, pp. 163–172.
[20] J.-A. Meyer, S. Doncieux, D. Filliat, A. Guillot, Evolutionary approaches to neural control of rolling, walking, swimming and flying animats or robots, in: R.J. Duro, J. Santos, M. Graña (Eds.), Biologically Inspired Robot Behavior Engineering, vol. 109, Physica-Verlag, 2002, pp. 1–43.
[21] S. Nolfi, D. Floreano, Evolutionary Robotics, MIT Press, 2000.
[22] T. Reil, P. Husbands, Evolution of central pattern generators for bipedal walking in a real-time physics environment, IEEE Trans. Evol. Comput. 6 (2) (2002) 159–168.
[23] R. Smith, The Open Dynamics Engine User Guide, 2003, 〈http://opende.sourceforge.net〉.
[24] E. Vaughan, E. Di Paulo, I. Harvey, The evolution of control and adaptation in a 3D powered passive dynamic walker, in: Proceedings of the Ninth International Conference on the Simulation and Synthesis of Living Systems, ALIFE'9, 2004, pp. 139–145.
[25] A. von Twickel, F. Pasemann, Reflex-oscillations in evolved single leg neurocontrollers for walking machines, Nat. Comput. 6 (2007) 311–337.
[26] S. Wischmann, F. Pasemann, From passive to active dynamic 3D bipedal walking—an evolutionary approach, in: M. Armada, P. Gonzalez de Santos (Eds.), Climbing and Walking Robots: Proceedings of the 7th International Conference (CLAWAR 2004), Springer-Verlag, 2004, pp. 737–744.

José Santos Reyes obtained an MS degree in Physics from the University of Santiago de Compostela, Spain, in 1989, and a Ph.D. from the same University in 1996. He is currently a senior lecturer in the Department of Computer Science at the University of A Coruña. His research interests include artificial life, neural computing, evolutionary computing and autonomous robotics.

Ángel Campo Penín obtained an MS degree in Computer Science from the University of A Coruña, Spain, in 2010. He is currently a Ph.D. student in the Department of Computer Science at the University of A Coruña. His research interests include neural computing, evolutionary computing and autonomous robotics.