Evolution of adaptive center-crossing continuous time recurrent neural networks for biped robot control

Ángel Campo and José Santos
Computing Science Department- University of A Coruña (Spain), santos@udc.es

Abstract. We used simulated evolution to obtain continuous time recurrent neural networks to control the locomotion of simulated bipeds. We also used the definition of center-crossing networks, so that the recurrent networks nodes can reach their areas of maximum sensitivity of their activation functions. Moreover, we incorporated a run-time adaptation of the nodes' biases to obtain such condition. We tested the improvements and possibilities this adaptation adds, focusing in the use for biped robot control.

1. Introduction and previous work

Bipedal walking is a difficult task due to its highly unstable dynamic behavior [5]. 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.

The work of Ijspeert [7] reviews different alternatives to define CPGs. Here we concentrate on the control of biped robots. Legged locomotion is characterized by cyclic activity of the limbs. The defining feature of the CPGs is a high degree of recurrence, which greatly biases the dynamics of the system toward cyclic activation patterns [9].

Several neural models have been used to implement CPGs. Nevertheless, as McHale and Husbands comment [4], although the characteristic equations associated with a specific network are a compact description of it, 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 [8] provides an alternative to the handmade design of robot controllers, especially for autonomous robots acting in uncertain and noisy domains, as artificial evolution is used to automate the design procedure of the controllers.

Beer [1] introduced the model of Continuous Time Recurrent Neural Network (CTRNN), one of the most used as CPG. As Beer indicates "CTRNNs is a class of neural models that is simple but dynamically universal" [2]. The work of McHale and Husbands [4] presents a comparative study of three types of neural networks: the conventional Continuous Time Recurrent Neural network (CTRNN) [1], the Center-crossing CTRNN [3] (explained in the next section), the Plastic Neural Network (PNN) [6] 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), continuous time recurrent networks were shown to have advantages in most of the cases. CTRNNs were able to attain a higher level of average fitness, although GasNets obtained the highest fitness peak with cyclic locomotion.

However, as exposed in [3] with a statistical analysis, if we use an evolutionary method to obtain networks with rhythmic behavior, the probability that a random network population of a moderate size (100 individuals) contains one or more CTRNNs exhibiting oscillatory behavior is rather small. Also, Reil and Husbands [9] evolved CTRNNs for a *simulated* biped. They showed that there is no need of proprioceptive information to the control of stable straight-line bipedal walking. Moreover, when they conducted 100 evolutionary runs, the fraction of such runs leading to stable walkers was only 10% (even allowing backward walking).

Mathayomchan and Beer [3] 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 act in the activation region of maximum sensitivity. 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 only 1.2% of the random circuits did so. 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.

Given these results, our aim is to increase the probability to obtain oscillatory networks in an evolutionary search. We began with the center-crossing definition and we added an adaptation methodology for that purpose.

2. Center-crossing continuous time recurrent neural networks

In the conventional CTRNN [1] the state of a single neuron *i* is computed by the following equation:

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

where *y* is the state of each neuron, *τ* is a time constant, *w* is the weight of an incoming connection, *σ* is the sigmoid activation function, *θ* is the bias term or firing threshold of the node and *I* is an external input.

As defined in [3], 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 neuron bias and all the synaptic inputs sum to 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 (σ(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: $\theta_i = -\sum_{j=1}^{N} w_{ji} \Big/ 2$ (Eq. 2)

This means that the bias exactly counteracts the sum of all the synaptic inputs when the connected neurons have a 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 neurons are in states of maximum sensitivity most of the time.

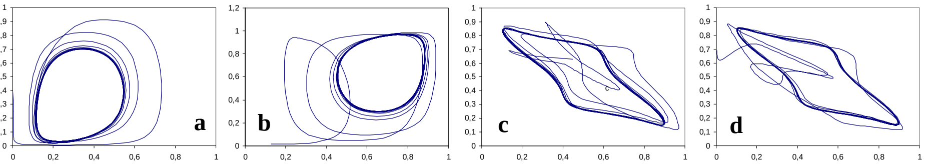

Fig 1: Robustness against initial conditions. Figures *a* and *b* show the phase plot of the temporal activation of two nodes of a center-crossing network when it was initialized with different activation values in the nodes. Figures *c* and *d* show the same plot in a network with run-time adjusted biases.

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 parameter space and, as Mathayomchan and Beer indicate [3], one would expect that an evolutionary algorithm would benefit from focusing its search there.

3. Bias adaptation

The center-crossing networks can be viewed as networks of neurons of maximum sensitivity. It is to be expected that such networks exhibit a wider range of dynamic behaviors than random networks would do. Nevertheless, the use of center-crossing networks does not guarantee that the corresponding network generates a rhythmic behavior. 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. 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 of its value. Even if the initial incoming values of the node are initialized in such a way that half of them 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 with fixed temporal values in the nodes.

Figure 1 has two examples which correspond to a center-crossing network. The figures show the phase plot of the time evolution of two nodes. Fig1.a and Fig1.b present two different dynamic behaviors depending on the initial values of the network nodes. 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 a rhythmic activation in the nodes. If we want to maintain the nodes in the regions of maximum sensitivity, 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). Thereby, it is changed according to the following formula (ignoring external inputs):

$$\theta_i(t+1) = \theta_i(t) - coef \cdot \left(\Delta_t \cdot \tau_i^{-1} \cdot \left(-y_i + \sum_{j=1}^{N} w_{ji} \cdot \sigma(y_j + \theta_j)\right) + y_i\right) \text{ (Eq. 3)}$$

where *coef* is a bias learning or adapting coefficient and Δ_t is the time step used in the integration of Equation 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*. Thereby, all nodes will be near the regions of maximum sensitivity to induce activation changes. The magnitude of the learning coefficient determines how fast that situation is obtained. The last two examples of Figure 1 correspond to a center-crossing network when the biases are adjusted in run-time. Independently of the initial conditions the trajectories fall in the same temporal limit cycle and, what is more important, 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.

4. Methods: Biped model and genetic algorithm

We used a simulated biped in the experiments employing the Open Dynamics Engine (ODE) physics simulator [10]. The articulated structure was created when rigid bodies were connected together through joints: two joints linking a hip and the legs, two knee joints and two ankle joints. These six actuated joints have a degree of freedom and were simulated as torsional springs. The angular hip displacement has a range of [-π/6,0.75·π], whereas the range is [-0.75·π,0] for the knees joints and [-0.25·π,0] for the ankles. 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 is proportional to its volume, the gravity was fixed to -9.81 m/s^2 and it was used a time step of 0.01s in the ODE simulation for each iteration in the environment.

We used a standard genetic algorithm (GA) for the evolution of the CPGs (CTRNNs) for the walking behavior. Each CTRNN is codified by a vector that includes the connection weights and the bias and time constants associated to each neuron. A population of 100 individuals was used in the different evolutionary runs. We used a rank-based method as selection operator: the 25% best individuals of the population are replicated to generate the next population. As in [3], these individuals were mutated with a given probability in their genes (parameters), adding them 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 is copied to the new population without any change. We did not use crossover operators because, as Reil and Husbands point out, there are “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 [9].

Finally, as in most of the previously mentioned works, we used as fitness the distance traveled in a straight line by the biped in a give time (8 seconds in our experiments). Additionally, in order to avoid grotesque movements, a penalization was introduced when the center of gravity felt below a certain height. We did not allow backward walking controllers as in [9].

We used fully interconnected CTRNNs as CPGs, where all the neurons were motor neurons to control each of the six joints. This implies 48 parameters to evolve (36 connection weights, 6 biases and 6 time constants). If the biases are defined according the central-crossing condition (or adjusted in run-time), the parameters are 42. The connection weights and biases were constrained to lie in the range ±16 (as in [3] and [9]), and time constants were constrained in the range [0.5, 5], as in [9]. The time step for the integration of Equation 1 was 0.1s. 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.

5. Results

We tested the difficulty to obtain CTRNNs which act as CPGs to provide a rhythmic behavior, as well as the fitness obtained, with three strategies: (1) random genetic populations with random defining parameters of the CTRNNs, (2) initial populations where the weights and temporal constants were random, and the biases were defined according to the center-crossing condition, (3) and finally random genetic populations with random defining parameters but the biases were adjusted in run-time according to Equation 3. Figure 2 summarizes the results of the evolutions with the GA. The quality evolutions are the average result of 50 different runs of the GA with different initial populations. The number of individuals was 100 in all those tests.

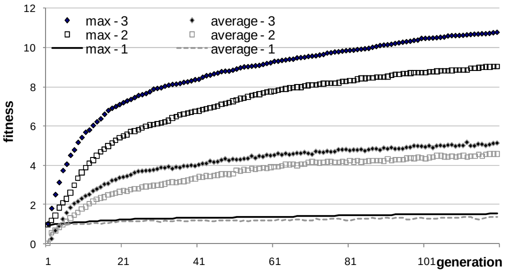

Fig 2: Evolution of the quality of the best individual and the average quality of the population for three different conditions: random CTRNNs (1), initial center-crossing CTRNNs (2) and run-time adapted center-crossing CTRNNs (3). The quality curves are an average of 50 different runs of the GA.

Mathayomchan and Beer [3] 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, regarding to an initial random population, with a mutation variance σ^2 of 0.05, while the difference decreases with increasing values of σ^2.

The same σ^2 of 0.05 was used in the evolutions of Figure 2. As it was expected, with the seeded initial populations with center-crossing NNs, the evolutionary algorithm obtained better CTRNNs that provided the required cyclic behavior. The average quality indicates the larger number of possible networks with such behavior with respect to a random population of CTRNNs. Once oscillations are discovered, 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.

With the use of the run-time adaptive biases all the individuals (adaptive CTRNNs) of the genetic population present a rhythmic behavior. It is likely that most of such cyclic behaviors are not adequate to the locomotion control, as the average fitness at the initial generation indicates. Nevertheless, the genetic algorithm has more NNs to fine-tune, so it obtains higher fitness controllers and in fewer generations.

Figure 3 shows several steps of the swing and stand phases of the legs of two evolutionary obtained CTRNNs which use run-time bias adaptation. The first one was obtained with the robot on a flat surface (coef=0.005) and the second one with it on a slope (coef=0.01). The figure shows one cycle of the repetitive activation of the nodes that control the hip joints (continuous lines) and the knees (dashed lines), defining the swing and stand phases (the activation of the ankles are not shown for clarity). In addition, the coefficient for bias adaptation of Eq. 3 determines how fast each node is set to the center-crossing condition. So, higher values in the coefficient 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. The videos of the behaviors of Fig. 3 and several videos with such control by the coefficient can be downloaded from [11].

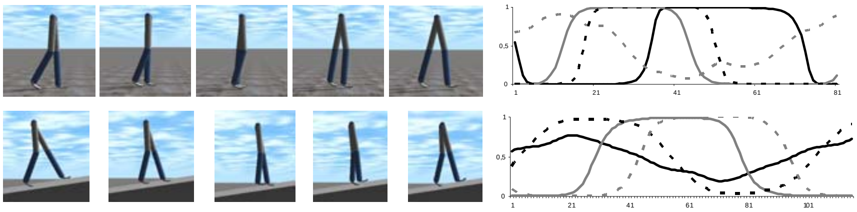

Fig 3: Several steps in the swing and stand phases of the locomotion behavior on a flat surface and on a slope. The right figures show the rhythmic activation of the network nodes that directly control the hip joints (continuous lines) and knees joints (dashed lines).

Conclusions

We used CTRNNs as controllers in simulated bipeds. A GA obtains the optimized networks for the required locomotion behavior. 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 a higher fitness regarding to non-adaptive center-crossing networks. The next step in our work is the modulation of network rhythmic activation through the bias adaptation methodology, so the network will be able to automatically adjust its cyclic behavior to the current surface it detects.

Acknowledgements: this paper has been funded by the Ministry of Science and Innovation of Spain through project TIN2007-64330.

References

[1] Beer, R., “On the Dynamics of Small Continuous Time Recurrent Neural Networks”, *Adaptive Behaviour* 3(4): 469-509, 1995.

[2] Beer, R., “Parameter Space Structure of Continuous Recurrent Neural Networks”, *Neural Computation* 18: 3009-3051, 2006.

[3] Mathayomchan, B., and Beer, R., “Center-Crossing Recurrent Neural Networks for the Evolution of Rhythmic Behavior”, *Neural Computation* 14: 2043-2051, 2002.

[4] McHale, G. and Husbands, P., “GasNets and other Evolvable Neural Networks applied to Bipedal Locomotion”, *From animals to animats 8 - Proceedings of the Eighth International Conference on the Simulation of Adaptive Behavior*, 163-172, 2004.

[5] Floreano, D. and Mattiussi, C., *Bio-Inspired Artificial Intelligence*, MIT Press, 2008.

[6] Floreano, D. and Mondada, F., “Evolutionary Neurocontrollers for Autonomous Mobile Robots”, *Neural Networks*, Vol. 11, pp. 1461-1478, 1998.

[7] Ijspeert, A.J., “Central Pattern Generators for Locomotion Control in Animals and Robots: a Review”, *Neural Networks* 21(4): 642-653, 2008.

[8] Nolfi, S. and Floreano, D., *Evolutionary Robotics*, MIT Press, 2000.

[9] Reil, T., and Husbands, P., “Evolution of Central Pattern Generators for Bipedal Walking in a Real-Time Physics Environment”, *IEEE Transactions on Evolutionary Computation* 6(2):159-168, 2002.

[10] Smith, R., “The Open Dynamics Engine User Guide”, 2003, *http://opende.sourceforge.net*

[11] *http://www.dc.fi.udc.es/ai/people/santos/esann2010/esann2010.html*