This article can be cited as L. F. Castillo, M. G. Bedia, G. Isaza, J. Velez and F. Seron, Development of Believable Bots in Videogames Capable of Learning During Runtime, International Journal of Artificial Intelligence, vol. 12, no. 1, pp. 117-128, 2014.

Development of Believable Bots in Videogames Capable of Learning During Runtime

Luis F. Castillo[2,3], Manuel G. Bedia[1], Gustavo Isaza[2] Jairo Velez[3], Francisco Seron[1]

[1]Departamento de Ciencias de la Computación,
Universidad de Zaragoza (Spain)
Email: {mgbedia, fseron}@unizar.es

[2] Departamento de Sistemas e Informática, Ci^{2}Dt^2-GITIR Research Center & Group,
Universidad de Caldas, Sede Prinicipal (Colombia);
Email: {gustavo.isaza, luis.castillo} @ucaldas.edu.co

[3]Grupo de Investigación Ingeniería del Software,
Universidad Autónoma de Manizales (Colombia);
Email: {jvelez, lfcastil} @autonoma.edu.co

ABSTRACT

*In this paper we show that, thanks to the features of Continuous-Time Recurrent Neural Network (CTRNNs), whose neurons act at different activation level of time, the controller of a bot is capable of learning during runtime in the UT2004 videogame environment without changing any of the network parameters. This behaviour has been described in simple forms of life (e.g. examples of the small nematode worm C. elegans, evidence for the formation of associations between temperatures and food has been known for quite some time) and, in this paper, we adapt these ideas applying them to a current commercial video game. The particular experimental conditions are as following: A bot is evolved in a task where it needs to search and discriminate its base camp and the enemy's camp and associate them with the altitude where the camp is, depending on its experience. The task requires either instrumental or classical conditioned response to be learned. In the last part of the paper is analytically analyzed the best-evolved agent's behaviour and, also, it is explained how some ideas and methodologies that come from models of biological theory, can be easily adapted to the control of virtual characters or "bots", i.e. synthetic agents with human behaviour***.**

Keywords: Concept learning, connectionism, neural nets, parameter learning

Computing Classification System: I.2.6

1. INTRODUCTION

The use of Continuous-Time Recurrent Neural Network (CTRNNs) (Beer, 1995) to design controllers for mobile agents has excelled in Evolutionary Robotics (Harvey et al, 2005). While it is true that physical agents were used in the projects undertaken initially, in the last few years the trend of implementing models adapted to the design of synthetic agents in virtual simulation has dominated (Jacobi, 1998) such as in video games. The field of video games in addition to being a powerful entertainment industry is also a framework for experimentation and research to test Artificial Intelligence techniques in complex environments. This article aims to adapt a learning technique to

the control of virtual characters or “bots” (synthetic agents with human behaviour) using CTRNNs for a current commercial video game. Associative learning requires responses, which are paired with a particular stimulus. Organisms at several levels of ‘complexity’ provide evidence for this, including many extraordinarily simple ones (e.g. examples of the small nematode worm C. elegans, evidence for the formation of associations between temperatures and food has been known for quite some time). Unreal Tournament 2004, also known as UT2004 or UT2K4, is a first-person action video game, mainly oriented to the multi-player experience whose aim varies according to the chosen game mode, but always includes the use of violence (virtual, of course) to achieve the goal. Unreal Tournament 2004 is part of the Unreal series and was co-developed by Epic Games and Digital Extremes and published by Atari. The goal is to get a bot controlled by a CTRNN with the ability to learn in real time without synaptic plasticity, i.e. without making changes to the network parameters.

In this paper we have shown that it is possible to build bots able to mix two types of learning in realtime. Although, after a process of learning, the controller of a bot has fixed their parameters, the agent still can generate adaptive behaviour without changing the value of the parameters, exploiting a model of associative and operative learning coupled, it is possible to successfully adapt a controller obtained in a experimental environment to the real one, by using an algorithm of differential evolution, and allowing the possibility of carrying out evolutionary process in a parallel way. Based on the previous results, agents that were programmed using this technique show an improved level of believing and a better humanness rate compared to standard non-player characters.

In the first section, the experiment and desired behaviour for the bot is described. Then the focus shifts to how to achieve the desired behaviour using the evolution of CTRNN. In the section that follows, the capacity of the CTRNN to learn without synaptic plasticity is analysed. Section 4 includes an analysis of the dynamics of said CTRNN with its environment and formally analyses bot behaviour in depth from its bifurcation diagrams. Lastly, the conclusions drawn from the experiment are shown.

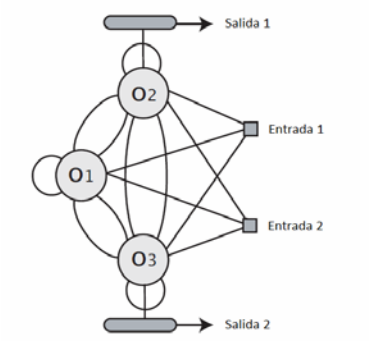
2. THEORICAL FRAMEWORK

2.1. Continuous-time Recurrent Neural Networks (CTRNN)

Among the community that studies autonomous agents, there is a growing interest in using dynamic neural networks to control the behaviour of agents (Collins and Jefferson, 1991) along with the evolution of such networks ((Werner and Dyer, 1991); (Beer and Gallagher, 1992); (Spiessens and Torreele, 1992); (Miller and Cliff, 1994); (Beer, 1995b)). More recent researches have evidenced algorithms used in optimization in relation with the learning (Gao and Huang, et al, 2012), (Precup, et al, 2012), (Ali, et al, 2013) and (Asawarungsaengkul, et al, 2013). These studies, among others, have shown that the combination of genetic algorithms and neural networks is an interesting technique to develop control structures in autonomous agents. However as a drawback, the required time for the evolution is the most important limit that seems to show the evolutionary focus. The main advantages are their noise resistance, interpolation ability, parallelism, its ability to model dynamic systems for dynamic and recurrent neural networks, and the response to the its evolution using Genetic Algorithms since the meaning of their actions “emerge” from both mechanisms. On the other hand,

This article can be cited as L. F. Castillo, M. G. Bedia, G. Isaza, J. Velez and F. Seron, Development of Believable Bots in Videogames Capable of Learning During Runtime, International Journal of Artificial Intelligence, vol. 12, no. 1, pp. 117-128, 2014.

updating neurons is normally synchronous (appropriate for when the neuron network task involves the recognition and classification of patterns), but neural networks that use time as one of the multiple information processing supports are used more frequently each time. This is because one of the characteristics that should be drawn consists of the temporary relationship of other characteristics. Its performance is completely asynchronous with some interneuron connections that produce delays in the input of neurons. For the tasks outlined in this paper, a strategy based on the use of Continuous Time Recurrent Neural Networks (CTRNN) has been chosen (Beer, 1995a). In such networks cycles may exist in their structure and are mathematically equivalent to Dynamic Systems. In contrast with feed forward neural networks, which only support reactive behaviours, dynamic neuron networks allow the agent to start an action regardless of the immediate situation and organise its behaviour in anticipation of future events (Funahashi and Nakamura, 1993). This type of neuron network also helps describe the agent as a Dynamic System coupled with the environment in which it is located, since it is shown that this type of neuron network is the simplest continuous dynamic neuron network nonlinear model. The neurobiological interpretation of the CTRNN has been shown and can be found in (Beer, 1995a)



#

Figure 1. Continuous-time Recurrent Neural Networks

The mathematical representation of the CTRNNs will be shown next. We begin with the behaviour of a neuron in its network. The activation is usually interpreted as the average activation frequency of the neuron a_i in the interval Δt of the update based on the value of the sum s_i of that neuron at that moment (the value of the weighted sum with the synaptic weights of the output of the neuron at instant *t-1*)

$$a_i^t = f(s_i^{t-1}) , \tag{1}$$

where

$$s_i^{t-1} = \sum_{j=1}^{N} w_{ij} a_j^{t-1} + \theta_i . \tag{2}$$

Several activation features exist in the specialised literature, although the sigmoid function is commonly used:

$$f(s) = \sigma(s) = \frac{1}{1+e^{-\alpha s}} . \tag{3}$$

where the value α indicates the velocity of sigmoid growth. The Continuous-time Recurrent Neural Networks are obtained when *Δt* tends to zero in the networks previously described. Its mathematical description is

$$\dot{y}_i = \frac{1}{\tau_i} * (-y_i + \sum_{j=1}^{N} w_{ji} * \sigma(y_j + \theta_j) + I_i) \qquad i = 1,2,....,N , \tag{4}$$

with

$$\sigma(\chi) = \frac{1}{1+e^{\chi}} , \tag{5}$$

where I_i represents an external input and τ makes each of the neurons time-dependent, since for different values the decrease of the level of activation of the neuron is faster or slower. The update rate of the neural network should be significantly greater (the interval between two updates will be smaller) than the value of *t*.

The result of this representation of the time in the CTRNN structure is crucial, because:

- On the one hand, it allows a form of short term memory.
- It can be shown that a CTRNN approximates any dynamic system under certain conditions (Beer, 1995a); (Funahashi and Nakamura, 1993)(Moreno et al , 2012)
- Due to the previous point, a view of the neural network is opened as a dynamic system partially coupled with the sensors and environment.

2.2. Application of CTRNNs to the bots design

To implement bots, the Unreal Tournament 2004 was used with the Gamebots 2004 expansion (which provides a way to run bots in the videogame) and the Pogamut 3 platform (which allows for the programming of the virtual agent using Java (using JDK 6) programming language and connecting it and receiving information from the videogame using a plugin for Netbeans. When choosing a certain behaviour for the bot, in which learning is required for its run time, it was decided to adapt the behaviour shown to the UT2004 environment by the Caenorhabditis elegans nematode (Izquierdo et al, 2008). Such behaviour consists of associating two stimuli (associative learning): temperature and food. This model was chosen because the C. elegans is a popular choice among researchers in the field of CTRNN evolution, since it shows behaviours simple enough to be modelled by small CTRNNs and complex enough to explore the memory capabilities of the CTRNNs [(Izquierdo et al, 2008)When adapting this model to the UT2004 environment using Pogamut, a 2D environment with an altitude gradient is used along one of their dimensions, which is shown in Figure 2A. In the model, there are two base types: “enemy” and “ally”. Each base is located only in regions with a particular range of altitudes: “high” between [9,10] and “low” between [-10,-9]. The region where each base is located depends on the type of environment: in the A-ent the “enemy” base is located in the “high” region and

This article can be cited as L. F. Castillo, M. G. Bedia, G. Isaza, J. Velez and F. Seron, Development of Believable Bots in Videogames Capable of Learning During Runtime, International Journal of Artificial Intelligence, vol. 12, no. 1, pp. 117-128, 2014.

the “ally” base is located in the “low” region, while in the B-ent the “enemy” base is located in the “low” region and the “ally” base in the “low” region as well. The altitude gradient extends throughout the environment, which is free of obstacles. To do this, an environment has been designed like the one shown in Figure 2B.

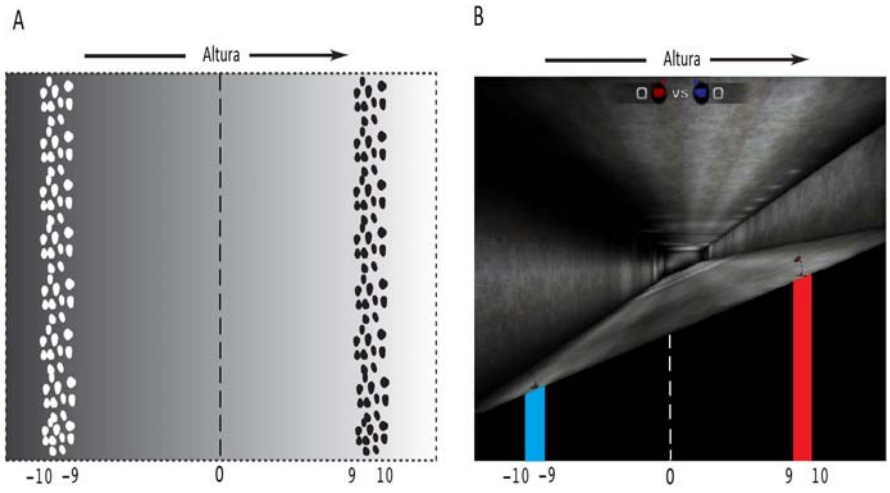


Figure 2. Simulation environment for the experiment. (A) Two dimensional theoretical simulation environment with a gradient of altitudes, in which the “enemy” base can be located in one of two regions represented by the dotted bands. (B) Simulation environment in UT2004, in which the gradient is the altitude where the bot is located and the red and blue stripes represent where the “high” and “low” bases are located. The altitude sensor may have any real value. The “base” sensor returns a value of 0 unless the bot is located in one of the bases: it will return B=1 if it is located in the “enemy” base, and B=-1 if it is located in the “ally” base.

It is expected that the bot is capable of making associations during its runtime about altitude and “enemy” bases in each of the environments described and of memorising the altitude to return to it if the bot is relocated to the centre of the map. To accomplish this, the bot will appear in the position 0 of the altitude gradient in a random orientation. From this moment on, the bot will execute 200 cycles to move around the entire environment in search of the “enemy” base and to stay in that region as efficiently as possible. Once 200 cycles have been completed, the bot is repositioned to the 0 altitude with a random orientation, and should be able to go up or down in the altitude gradient depending on if during the previous execution it learned that it was in an A-ent or B-ent environment. If the type of environment is changed, the bot needs to be able to relearn and change its altitude preference.

2.3. Evolutionary learning to obtain CTRNN

The model used in this experiment is a CTRNN model with all its neurons completely connected and interconnected. Therefore, since the neuron model for this experiment is based on the equation of the previously described CTRNN, by applying the Euler integration with a time cycle of Δt to activate the simulation nodes, the relevant equation is

This article can be cited as L. F. Castillo, M. G. Bedia, G. Isaza, J. Velez and F. Seron, Development of Believable Bots in Videogames Capable of Learning During Runtime, International Journal of Artificial Intelligence, vol. 12, no. 1, pp. 117-128, 2014.

$$\mathrm{y}_i(n+1) = y_i(n) + \frac{\Delta t}{\tau_i} * (-y_i + \sum_{j=1}^{N} w_{ji} * \sigma(y_j + \theta_j) + s_i A(\chi) + g_i B(\chi; e), \tag{6}$$

where i is an index (i =1, 2, ..., N), N is the number of neurons, y_i is the status of the neuron, Δt is the life cycle for the integration, τ_i is the time constant of the neuron activation, w_{ji} is the weight of the connection between the neurons i and j, θ is the bias term, $\sigma(x)$ is the sigmoid function, $A(x)$ is the altitude sensor, s_i is the weight of the connection of the altitude sensor, $B(x; e)$ is the base sensor (which depends on the type of environment), and g_i is the weight of the base sensor connection. For simplicity's sake, the value of Δt has been set to 0.4 seconds. Due to the complexity of the experiment, the evolutionary process is not made from scratch using Pogamut. Instead, the CTRNN obtained by Izquierdo experiment (Izquierdo et al, 2008) is used as a starting point for the behaviour of preference by the C. elegans. In this experiment, the desired behaviour in a completely interconnected and auto-connected four-neuron CTRNN after 10000 generations is used. If one intends to perform such an experiment in Pogamut for a similar CTRNN, the estimated time spent to complete the experiment is

$$Duration = (200 \frac{cycles}{\exp eriments} * 9 \frac{\exp eriments}{bots} * 320 \frac{bots}{generations} * 1000\ generations * 0{,}4 \sec / cycle) / (60 \frac{seg}{\min} * 60 \frac{\min}{hour} * 24 \frac{hours}{days} * 365\ days / year). \tag{7}$$

Therefore, it was decided to adapt the already trained CTRNN for behaviour regarding the temperature of C. elegans to the UT2004 environment, so that an initial population of 50 individuals is created. All individuals were defined by the genotype that defines the CTRNN from which the evolutionary process is carried out over 50 generations. The "fitness" function used for the evaluation throughout the evolutionary process

$$\phi = \int_{t=50}^{200} B dt, \tag{8}$$

where B is the base sensor. This function rewards the fact that the bot is located in the enemy base during the last three quarters of its lifetime.

The values returned by the "fitness" function during the evolutionary process are shown in the graphic from Figure 3. While the bot returned a maximum adaptation value of 95% in UT2004 for the original CTRNN, the best returned CTRNN after the evolutionary process reaches a "fitness" value of 98,72 % which is shown in Figure 3.

This article can be cited as L. F. Castillo, M. G. Bedia, G. Isaza, J. Velez and F. Seron, Development of Believable Bots in Videogames Capable of Learning During Runtime, International Journal of Artificial Intelligence, vol. 12, no. 1, pp. 117-128, 2014.

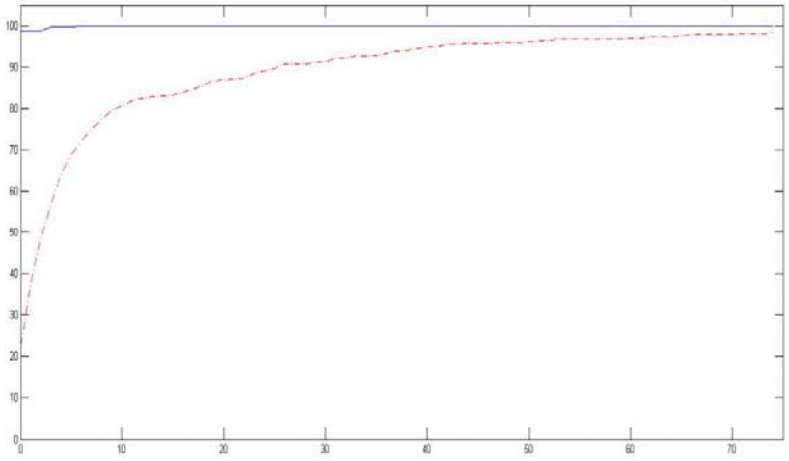
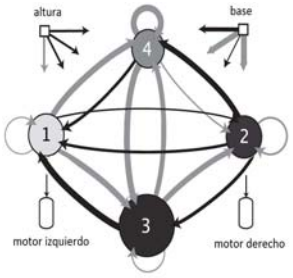


Figure 3: Graphic with the results returned for the "fitness" function after the evolutionary process for getting a CTRNN with runtime bot learning capability. The blue line shows the value returned by the best individual of each generation. The red dotted line shows the average of all the individuals in each generation.

3. Analysis of the learning behavior of the bot

If the obtained CTRNN parameters are analysed (Figure 4) three important characteristics can be seen



	y1	y2	y3	y4
τ	1,0521	1,023	5,8984	1,1077
θ	5,7900	-8,415	-9,3576	-0,7512
A	-0,621	0,6595	0,6393	0,7439
B	-87358	-9,5254	0,8064	4,8860
y1	-1,8126	2,4783	-9,2366	-9,1607
y2	3,0982	-3,7070	5,3725	6,9319
y3	9,8654	-8,6786	-2,8529	-8,1644
y4	3,4975	-2,2731	-9,9652	-9,4281

Figure 4 Parameters for the best CTRNN with 4 completely interconnected and self-connected neurons. The nodes are shaded according to their bias. The excitatory (black) and inhibitory (gray) connections is proportional to the weight of the neurons. The time constants are represented by the size of the node, the slower nerons are the larger ones.

First, one can see a high inverse bilateral symmetry: the way in which neurons 1 and 2 are connected to neurons 3 and 4 is by having similar weight but an opposite sign; both neurons are quickly activated; node 3 strongly excites neuron 1 and strongly inhibits neuron 2; both of their self-connections have small weights; the altitude sensor inhibits neuron 1 and excites neuron 2. Second, the base sensor has a large inhibitory force on the motor neurons, which will allow the bot to stop itself once it has found the enemy base camp. Lastly, and perhaps importantly, all neurons act as quickly as possible except for neuron 3, which is a lower magnitude order, which provides the CTRNN

This article can be cited as L. F. Castillo, M. G. Bedia, G. Isaza, J. Velez and F. Seron, Development of Believable Bots in Videogames Capable of Learning During Runtime, International Journal of Artificial Intelligence, vol. 12, no. 1, pp. 117-128, 2014.

a multi-scale activity over time.

Figure 5. Graphic illustration of the behaviour of a bot controlled by the CTRNN as a result of experiment 1 for the model.

The obtained neural network is used to control the behaviour of the bot in a sequence of two environmental changes and three base searches per environment, like those used for the CTRNN evolution.

1. At the beginning of the execution, the bot goes to the bottom of the map, but changes its behaviour and goes to upward before reaching the region where the base is located in the lower part of the map (between [-10,-9]). This seems to be part of the search strategy, since the bot still does not know in what kind of environment it is, and it is a phenomenon that has been observed in all executions of the bot.
2. Once the enemy base has been located in the upper area of the map, a Base value of 1 is received for finding the enemy base and the value of motor neurons quickly decreases to 0 so that the bot remains in that area as long as possible. Once the enemy base has been located, if the bot is put back in the centre, it is able to return to the enemy base directly.
3. If the environment changes (the bot changes teams), the bot returns to the top part of the map, with the difference that the sensor receives a base value of B=-1 for being in the ally base. The bot continues going up beyond the base until its behaviour gradually changes to go down in the altitude gradient and it stops once it reaches the enemy base, this time located in the lower part of the map. By placing it in the centre, the bot has learned where the new enemy base is located and goes toward it.

4. Analysis of the dynamics from the CTRNN-environment system

After having analysed both the structure of the CTRNN as well as the observable behaviour of the bot that it controls, this section will explore the system dynamics formed by the CTRNN and its environment. The aim is to understand how the bot dynamics are structured so that the location of the

This article can be cited as L. F. Castillo, M. G. Bedia, G. Isaza, J. Velez and F. Seron, Development of Believable Bots in Videogames Capable of Learning During Runtime, International Journal of Artificial Intelligence, vol. 12, no. 1, pp. 117-128, 2014.

enemy base found in the past affects the direction of the altitude gradient in which it will move. To analyse the system dynamics, the bifurcation diagrams will be analysed, which are received for all possible cases. The system dynamics change based on the altitude at which the enemy base is located and whether or not the enemy base is or is not found.

- *Analysis of the dynamics for B=0*

First, how the system equilibrium changes based on the altitude in the absence of based is checked. To do this, a case without bases, i.e. a case in which the sensor base is always B=0, is observed. Since the dynamic system defined by the CTRNN has four variables (activation of neurons) and the altitude is the parameter that changes, the bifurcation diagram is 5-dimensional. Figure 6 shows the four two-dimensional projections from the 5-dimensional diagram, one for each neuron activation value based on the altitude variable

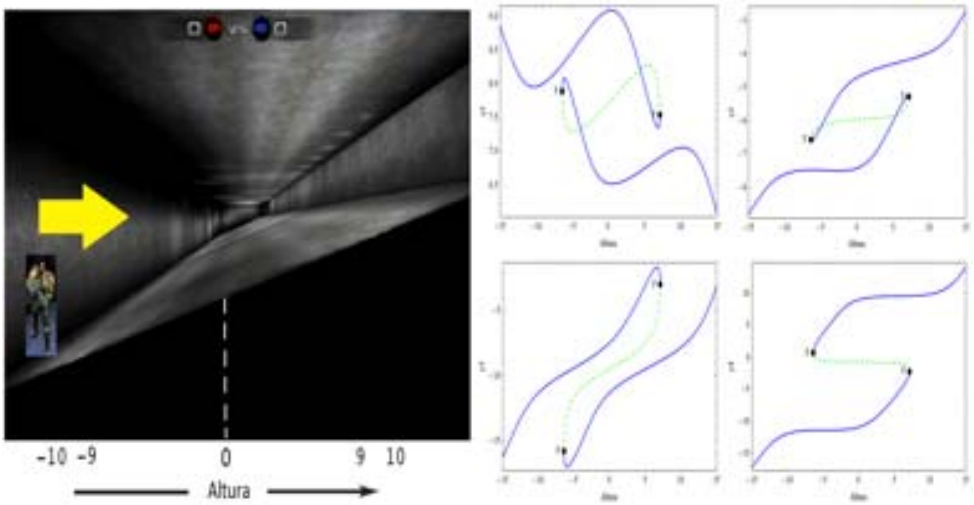


Figure 6. (A) Graphic illustration of the behavior of a bot controlled by CTRNN. (B) Bifurcation diagram in the presence of the enemy base. Four two-dimensional projections from the 5-dimensional diagram, one for each of the neurons from the CTRNN. The solid lines represent stable equilibrium points, while the dashed lines represent unstable equilibrium points. The grey verticle lines show the altitude ranges where the enemy base could be located

- *Analysis of the dynamics for B=1*

Second, the goal is to see how the system equilibrium changes based on the altitude enemy base presence, for which, although it is considered as a B=1 input in the total range of altitudes, only the shaded areas in grey are of interest, since the enemy base only can be found there. Figure 7 shows the four two-dimensional projections from the 5-dimensional diagram for this case- one for each neuron activation value based on the altitude variable. As is evidenced in the diagram, when the enemy base is located in the region corresponding to “high” altitudes, there is a single point of stable equilibrium. However, when the enemy base is located in “low” altitudes, the agent can be in one of two possible states, since the agent stays there long enough to reach the equilibrium point.

This article can be cited as L. F. Castillo, M. G. Bedia, G. Isaza, J. Velez and F. Seron, Development of Believable Bots in Videogames Capable of Learning During Runtime, International Journal of Artificial Intelligence, vol. 12, no. 1, pp. 117-128, 2014.

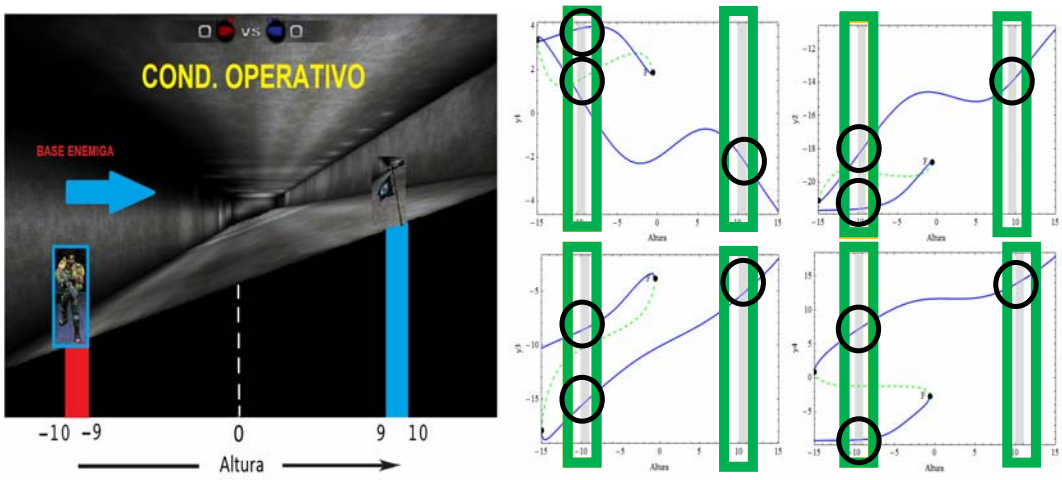


Figure 7. (A) Graphic exmple of the behavior of a bot controlled by CTRNN (B) Bifurcation diagram in the presence of the enemy base. Four two-dimensional projections from the 5-dimentional diagram, one for each one of the CTRNN neurons. The solid lines represent stable equilibrium point while the dashed lines represent unstable equilibrium points. The vertical gray lines show the altitude ranges where you can find the enemy base

Observations

The study of these diagrams suggests two main predictions:

- The first of which is that in the absence of any base, the system fall into an unlimited cycle in which the agent switches between going up and down the altitude gradient. Although the bot has not been trained for this scenario, this can be interpreted as a “high level” search behaviour of the enemy base which comes from the two “low level” behaviours for which it has been trained. This behaviour supports the fact that it is not necessary to consider the base sensor when the value returned is -1.
- Second, as a result of the bistability observed in the graphics from Figure 7, one could predict and confirm that, even after experimenting with environments with the enemy base in the “low” regions, if the agent is exposed to low altitudes in the presence of the enemy base for long enough, the agent could become reconditioned to navigate up the gradient altitudes.

5. DISCUSSION AND CONCLUSION

This paper has proven how, thanks to the feature of the CTRNNs whose neurons work with different activation times, a bot is capable of learning during runtime in the UT2004 videogame environment without changing any of the network parameters. In addition, it has been found that the behaviour that CTRNN has been trained for concerning the associate learning behaviour is not always as predictable as a priori may seem. Once the agent dynamics have been analysed with their environment, it was noted that the agent had a behaviour in which it ignored preconceptions in its

design: first, the CTRNN ignored the face that being in the ally base (indicated by a base value of =-1) and opted for ascending and descending behaviour in the altitude gradient in search of the enemy base; second, despite having designed the experiment as a model of associative learning with classical conditioning (associated altitude and enemy base), the CTRNN shows associative learning capacity with operative conditioning (associating stimuli with behaviour). As for the UT2004 simulation environment, Pogamut has proven to be of little use when the CTRNNs are evolving with many parameters (32 in this case). However, it has been shown that it is possible to successfully adapt to the UT2004 videogame environment a CTRNN obtained in another simulation environment with different characteristics, by using the Differential Evolution from an initial population in which all individuals are defined by the said CTRNN. This opens up the possibility of carrying out the evolutionary process in a simulation environment with shorter cycle times when executing the actions and in which it is possible to make the evaluations of the genetic algorithm individuals in a parallel way, thereby significantly reducing the time needed by the genetic algorithm. Once a resulted CTRNN satisfies the desired behaviour the evolutionary process moves to Pogamut for a few generations to adapt it to the UT2004 environment.

REFERENCES

Ali, M. Z., Alkhatib, K., Tashtoush, Y. (2013). Cultural algorithms: Emerging social structures for the solution of complex optimization problems,International Journal of Artificial Intelligence, 11(A13): 20-42.

Asawarungsaengkul, K., Rattanamanee, T., Wuttipornpun, T. (2013). Adaptation of the GRASP algorithm to solve a multiobjective problem using the Pareto concept, International Journal of Artificial Intelligence, 11(A13): 237-256.

Beer, R. D. (1995a). On the dynamics of small continuous-time recurrent neural networks. Adaptative Behavior, 3(4): 459-509.

Beer, R. D. (1995b). A dynamical systems perspective on agent-environment interaction. Artificial Intelligence, 72, 173–215.

Beer, R.D., Gallagher, J. (1992). Evolving dynamical neural networks for adaptive behavior. Adaptive Behavior, 1(1), 91–122.

Collins, R. J., Jefferson, D. R. (1991). Representations for artificial organisms. In J.- A. Meyer, H. Roitblat and S. Wilson, eds., From Animals to Animats 1: Proceedings of the Second International Conference on the Simulation of Adaptive Behavior (pp. 382-390). Cambridge: MIT Press.

Funahashi, K., Nakamura, Y. (1993). Approximation of dynamical systems by con- tinuous time recurrent neural networks. Neural Networks 6: 801-806.

Gao, L., Huang, J., Li, X. (2012). An effective cellular particle swarm optimization for parameters optimization of a multi-pass milling process, Applied Soft Computing, 12(11): 3490-3499.

Harvey, I., Di Paolo, E., Wood, R., Quinn, M., Tuci, E.A. (2005). Evolutionary robotics: A new scientific tool for studying cognition. Artificial Life, 11(1-2): 79-98.

Izquierdo, E.J., Harvey, I., Beer, R.D. (2008) Associative learning on a continuum in evolved dynamic neural networks. Journal of Adaptive Behavior 16(6):361-384.

Jakobi, N. (1998). Minimal Simulations For Evolutionary Robotics. PhD thesis. COGS, University of Sussex, UK.

Miller, G. F., Cliff, D. (1994). Protean behavior in dynamic games: Arguments for the coevolution of pursuit-evasion tactics. In D. Cliff, P. Husbands, J. Meyer and S. Wilson, eds., From Animals to Animats 3: Proceedings of the Second International Conference on the Simulation of Adaptive Behavior (pp. 411-420). Cambridge: MIT Press.

Precup, R.-E., David, R.-C., Petriu, E. M., Preitl, S., Radac, M.-B. (2012). Novel adaptive gravitational search algorithm for fuzzy controlled servo systems, IEEE Transactions on Industrial Informatics, 8(4): 791-800.

Spiessens, P. y Torreele, J. (1992). Massively parallel evolution of recurrent networks: An approach to temporal processing. In F.J. Varela and P. Bourgine, eds., Toward a Practice of Autonomous Systems: Proceedings of the First European Conference on Artificial Life (pp. 70-77), Cambridge: MIT Press.

Ruiz, S. M., Bedia, M. G., Castillo, L. F., Isaza, G. A. (2012). Navigation and obstacle avoidance in an unstructured environment Videogame through recurrent neural networks continuous time (CTRNN), Proceedings of 7th Colombian Computing Congress (CCC 2012), pp. 1-6, doi: 10.1109/ColombianCC.2012.6398004.

Werner, G. M., Dyer, M. G. (1991). Evolution of communication in artificial orga- nisms. En C. G. Langton, C. Taylor, J. D. Farmer and S. Rasmussen, eds., Artificial Life II (pp. 659-687). Reading, MA: Addison-Wesley.

Xu, J,-X., Deng, Z., Ji, D. (2010). Study on C. elegans behaviors using recurrent neural network model, Proceedings of 2010 IEEE Conference on Cybernetics and Intelligent Systems (CIS 2010), pp. 1-6, doi: 10.1109/ICCIS.2010.5518591.