CHAPTER VII

A DISTRIBUTED NEURAL CONTROLLER FOR LOCOMOTION IN LINEAR MODULAR ROBOTIC CONFIGURATIONS

AVINASH RANGANATH, JUAN GONZÁLEZ-GOMEZ and LUIS MORENO LORENTE

Robotics Lab, Universidad Carlos III de Madrid, {juan.gonzalez.gomez, javier.gonzalez-quijano, mohamed.Abderrahim}@uc3m.ing.es

Artificial Neural Networks (ANNs) are well known for oscillatory output when in a closed loop with the feedback from the system it is controlling. Here we propose a new neural controller for locomotion in linear modular robotic configurations, based on oscillatory output from simple ANNs. We investigate two different methods for controlling the action of each module based on the neural output, and also evolve neural controllers which have the ability to overcome external perturbations. We use a standard Genetic Algorithm (GA) for optimizing the synaptic weights of the ANN

1 Introduction

In the past decade, research in the field of modular robotics has seen a rise. Experimental results in using such systems for dull, dirty and dangerous operations have been encouraging. With self reconfiguration capabilities, modular robots promise to be operable in unforeseen environments and terrains. Locomotion is one of the primary features for such a system to be able to function efficiently

CEBOT (Fukuda, 1988), (Fukuda, 1990) by Fukuda et al. is one of the earliest demonstration of a modular robotic system. Here the authors demonstrate locomotion in modular robots through self-reconfiguration of individual modules, where each module has the ability to move independently. PolyBot (Yim, 1994) by Mark Yim is a classical example of chain architecture reconfigurable modular robots, which has demonstrated many modes of locomotion including walking, crawling, rolling, climbing, etc.

We proceed by classifying locomotion controllers in section 2, in section 3 we explain the robotic platform on which the experiments are based. Section 4 contains experiments and results, and we finish the paper by providing concluding remarks and future work in section 5.

2 Locomotion controllers for modular robots

Locomotion in general, weather it is a gallop of an horse, or flapping wings of a bird, or humanoid bipedal walking, can be seen as repetitive and coordinated movements of limbs, which result in the emergence of locomotion gaits. And in limbless creatures like snakes, inchworm or caterpillars, it is the coordinated expansion and contraction of the body muscles. Therefore, in essence, locomotion can be seen as coordinated oscillation of limbs. Looking at locomotion as a collection of oscillation, a steady difference in phase between these oscillations can produce the required coordination. We proceed by classifying and reviewing different types of controllers implemented for locomotion in modular robots.

2.1 Classification

Locomotion controllers for modular robots can be broadly classified into two groups [Figure 1(a)]; classical controllers and bio-inspired controllers. The former come from the industrial robotics domain and it is based on inverse kinematic and trajectory generation. These kinds of controllers are hard to scale with the increase in the degrees of freedom, and they require high computation power. On the contrary the later class of controllers are inspired by biological processes. These controllers have been successfully implemented on different modular robotic platforms. Based on the method used, these controllers can be further sub-classified into Cellular Automota, Digital Hormone Method, and Oscillation based methods [Figure 1(b)].

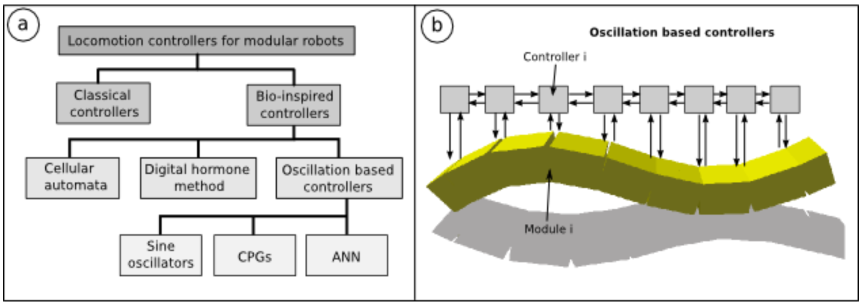

Fig. 1. (a) A classification of locomotion controllers for modular robots. (b) Schematic of a generic oscillation based locomotion controller for modular robots.

Lal et al. in (Lal, 2006) have implemented a Cellular Automata model for controlling locomotion of a five legged star shaped modular robot, where rules are evolved for controlling the actuator of each module, distributedly, based on the state of the module´s actuator and that of its immediate neighboring module´s actuators, in the previous time step.

Shen et al. have used a biologically inspired method called Digital Hormone Method (Shen, 2000), (Salemi, 2001), (Hou, 2006), for adaptive communication of state information between modules, based on which a module can decide an action from the gait table, which results in the emergence of locomotion. A particularly interesting aspect of this work is that if the configuration of the robotic organism changes during runtime, or if one or some modules fail, with adaptive communication, the locomotion gait is adapted to suit the change in configuration. Digital Hormones have been successfully implemented on two different modular robotic platforms called CONRO (Castano, 2002), (Shen, 2002) and Superbot (Salemi, 2006).

Gonzalez-Gomez et al. demonstrate in (Gonzalez-Gomez, 2005) how simple sinusoidal oscillators can be used on minimal configuration modular robots with two and three modules for generating locomotion in once and two dimensions respectively, and in (Zhang, 2009) they study the locomotion of two different kinds of caterpillar gaits, from a kinematic perspective, and replicate the same on linear configuration modular robots, again using simple sinusoidal oscillators.

In (Sproewitz, 2008) Ijspreet et al. at the Biorobotics Laboratory, EPFL, have used Central Pattern Generators (CPG) (Ijspeert, 2008) for producing locomotory oscillations on their modular robotic platform called YaMoR

(Moeckel, 2005), among other modular and non-modular robotic platforms. In (Pouya, 2010) they have tried similar CPGs for producing both oscillation and rotation in their second generation modular robotic platform called Roombots. CPGs are specialized neurons found in the spinal cord of vertebrate animals which have the capability of producing rhythmic output without rhythmic sensory or central input. The mathematical model of CPGs used for controlling locomotion in modular robots are usually one or two CPG neurons per module, which are coupled in different ways with CPGs of other modules based on the configuration. CPGs were first successfully used on a modular robotic platform by Kamimura et al. in (Kamimura, 2003), where they use CPGs to produce oscillations for adaptive locomotion on their M-TRAN modular robots.

Lal et al. in (Lal, 2007) have implemented an ANN model as a locomotion controller for their brittle star robot. Here each module is modeled as a neuron in a fully connected neural network. Neurons sum their weighted input stimulus, which is the actuator phase angle that they share locally or globally based on their location in the configuration, and use a sinusoidal activation function to determine the next step. The authors have used GA for evolving optimal synaptic weight vector of the ANN.

2.1 Proposed neural controller

We have implemented a fully connected feed-forward MLP ANN model with five input neurons, one output neuron, and a single hidden layer with five hidden neurons [Figure 2(a)]. We have tried to keep the architecture of the ANN simple, as we wanted to focus on how different inputs to the ANN would affect its output.

Each module in a given configuration has its own ANN model whose output neuron is connected to the module's actuator, making it a distributed controller. Furthermore, all the modules in a given configuration has the same ANN architecture, with exactly the same topology and weight vector, making it a homogeneous distributed controller [Figure 2(b)]. Although all the modules have identical neural model, the difference in their behavior emerges based on the difference in the input fed to their respective ANN at any given cycle. Following are the details of the neural input layer,

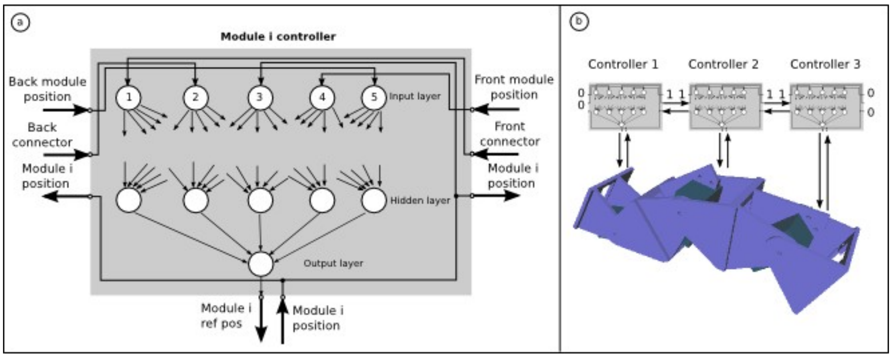

Fig. 2. (a) A schematic of the proposed neural architecture. (b) Simulated model of a three module configuration with distributed homogeneous neural based controller

- Input neurons 1-2: These neurons represent the connector information of a module. Neurons 1 and 2 represent front and back connectors respectively. These neurons are fed with binary input of either '1' if the corresponding connector is connected to another module, or '0' if not. In a given configuration, although the inputs to these two neurons may be different among different modules, they remain constant for a particular module, throughout its execution, making these two neurons bias nodes.
- Input neuron 3: This neuron is fed with the current actuator value, after being normalized between `-1´ and `+1´.
- Input neurons 4-5: These neurons are fed with the current actuator value of other directly connected modules. A '0' if the corresponding side is unconnected. Again, these inputs are normalized between `-1´ and `+1´.

Unlike CPG, which is a neural model as well, that is specifically designed for producing oscillatory or rhythmic pattern output, the proposed model has the potential to be extended for controlling more than just locomotion. With CPGs, if the configuration of the modular robotic organism changes, the coupling of the CPGs have to be changed to adapt to the new configuration, which could be a drawback if the controller is expected to work on self reconfigurable modular robots. In the proposed model, along with the ability to complexify the hidden layer of the architecture, we believe we can evolve neural architecture with the ability to control locomotion in different modular robotic configurations.

The neural model proposed in (Lal, 2007) is designed to represent every module as a neuron, unlike our model where each module has a complete ANN model. With a single neuron per module, it seems insufficient to generate any complex or adaptive behavior.

3 Robotic platform

The Y1 modules developed by Juan Gonzalez-Gomez (Gonzalez-Gomez, 2008) is an open source, low cost, flexible and easy to build modular robotic platform, which has been used as a research platform in many different work (Gonzalez-Gomez, 2005), (Zhang, 2009), (Herrero-Carron, 2011). We have implemented our proposed neural model on the simulated Y1 modules in OpenRAVE (Diankov, 2010), which is an open-source Open Dynamic Engine based robotics simulator, along with OpenMR, a modular robotics plug-in for OpenRave. We have used the open source Neural Network library called Flood (Lopez, 2010) for implementing the ANN model for the neural controller.

The Y1s [Figure 3(a)] are open-ended cube shaped modules, which has a single degree of freedom, with a rotation range of 180 degrees. The dimensions of these modules are 72x52x52 mm. The simulated modules are kept consistent with the real modules, both structure wise, and with respect to actuator features. Each module can be connected with two other modules, one each on opposite sides. We were able to successfully test the neural oscillator model on two, three, four and five module linear configurations, where in each module can actuate in the pitch axis. In this paper, we will focus more on the three module configuration to compare different methods implemented.

4 Experiments and results

With the above proposed neural controller we hope to achieve locomotory oscillations with some degree of fault tolerance when the robotic organism faces any external perturbation. For evolving neural controllers whose output resulted in stable locomotion gait, we started with a population of random individuals, and followed a fairly standard GA approach, with Roulette Wheel selection method and Intermediate Recombination method for creating new offspring. Table 1 provides the GA parameters we em-ployer for evolving our neural controller.

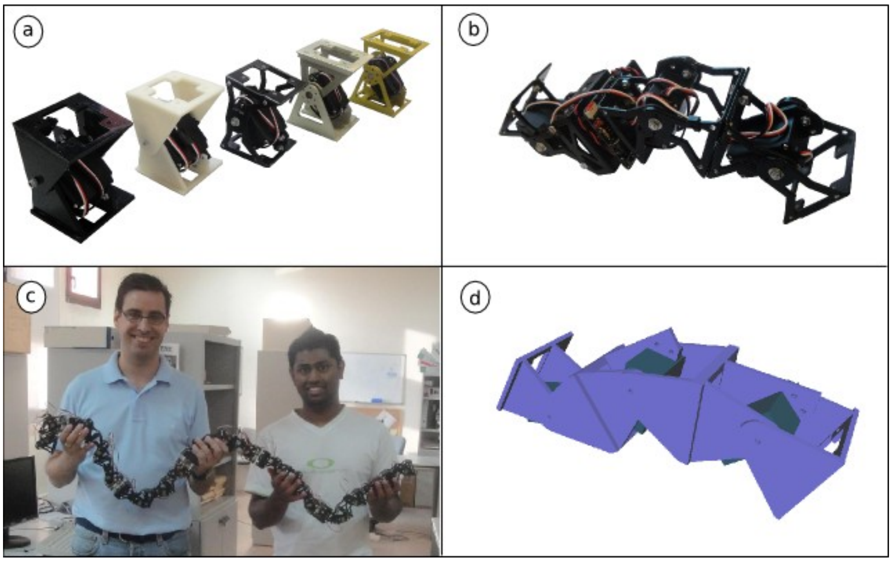

Fig. 3. (a) The Y1 family of modules. (b) A three module linear configuration. (c) Dr. Juan Gonzalez-Gomez and Avinash Ranganath with a 18 module linear configuration robot. (d) Simulated model of a three module configuration

Table 1. Parameters used for evolving neural controller

Parameters	Value
Population size	50
Generations	100
Crossover percentage	50.00%
Elite population retained	12.50%
Mutation rate	1/Size of genome

4.1 Experiment 1

Firstly, we wanted to test the validity of the proposed model for producing locomotory oscillations. So we setup a method in which after ever predetermined number of n time steps, inputs were fed to the ANN of each module, an output calculated, and the same fed to the respective module's actuator, after scaling the output value appropriately to the range of the actuator. The number of time steps between consecutive neural actuation was predetermined and fixed to a value of 1 second.

Figure 4, plots the actuator value against time of all the modules of an evolved three module configuration. The best performing individual of the final generation was able to locomot at a speed of 4.5 cms/sec. And as see in figure 4, the neural controller is able to very quickly converge and settle into a stable oscillation pattern, and with that into a stable locomotion gait within a very short period of time. We were also able to validate the same model by evolving two, four and five module linear configurations, which produced similar locomotion gait.

Figure 5 is a plot of the average frequency of the oscillations, and the speed of locomotion achieved by the best performing individual of each generation, over the course of evolution. The average frequency remains fairly constant throughout the evolution, but the speed of locomotion increases. This is because the rate of neural output is a value that is predetermined and fixed. Juan Gonzalez-Gomez in (Gonzalez-Gomez, 2008) has shown that different combinations of amplitude, phase, offset and frequency are required for achieving successful linear configuration locomotion gaits on varying terrains. To achieve a model which can adapt the frequency through evolution, we tried the model explained in the next section.

4.2 Experiment 2

To evolve oscillators whose frequency can adapt along the evolutionary process, we modified the previous model by actuating a module once it had reached its desired angle, which is the ANN output from the previous cycle, instead of at fixed intervals. Whether the actuator has reached the desired angle or not, was decided based on the satisfaction of either (1) or (2),

$$\left|\theta_{ANN} - \theta_{t_i}\right| \leq \alpha \tag{1}$$

$$\left|\theta_{t_i} - \theta_{t_{i-x}} \leq \beta\right| \tag{2}$$

Where θ_{ANN} is the desired actuator angle obtained from the ANN in the last cycle, θ_{t_i} is the angle of the actuator at the current time step. Values α, β, and x are constants.

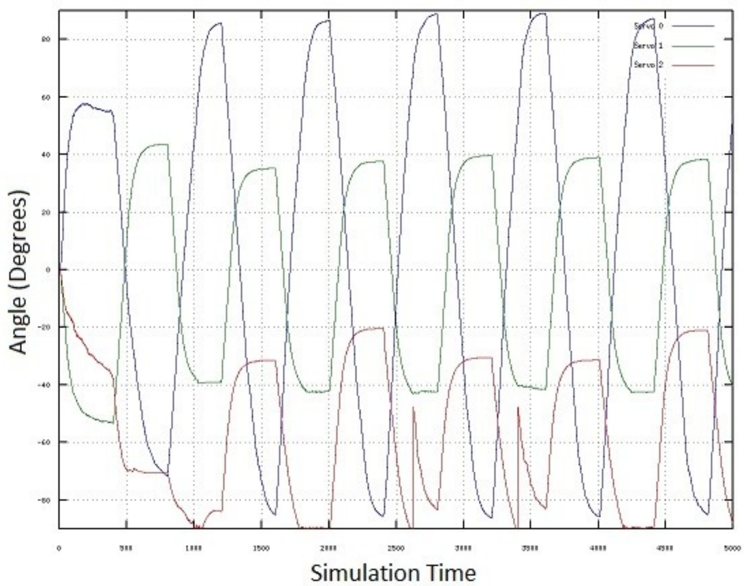

Fig. 4. Plot of the actuator values of a three module configuration with frequency controlled neural actuation.

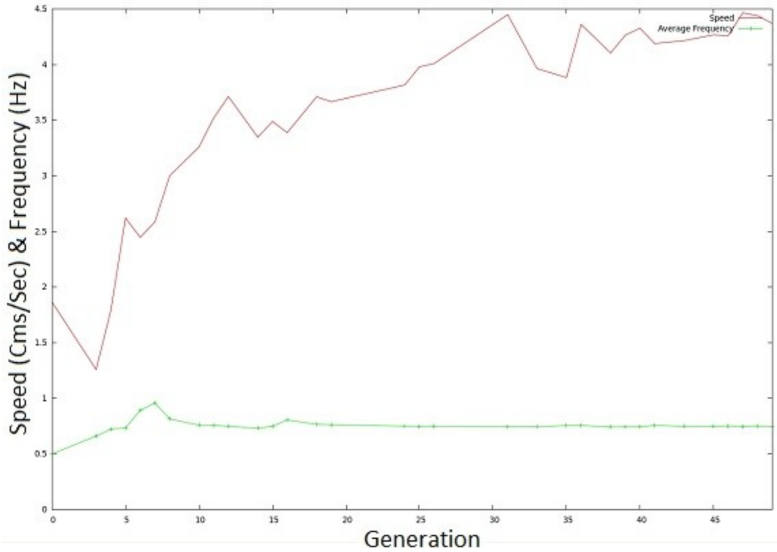

Fig. 5. Plot of the average oscillation frequency and the locomotion speed of the best performing individuals throughout the evolution of the frequency controlled neural actuation method.

With (1) we check if the current actuator angle lies within a small range of $\alpha < \theta_{ANN} < \alpha$. And with (2) we check if the rate of actuation is always above a certain threshold defined by β. The values of α, β and x were hand-coded to '3.0', '5.0', and `30` respectively, after a few experimental observations. With (2) we make sure that the neural controller would not fall into a deadlock if the actuator is unable to reach the desired angle. Figure 6 is the plot of the actuator value of an evolved three module individual. The best performing individual in this method was able to achieve a speed of over 6 cms/sec. Figure 7 is a plot of the locomotion speed achieved by the best performing individual of each generation and their corresponding average oscillation frequency, which indicates a direct correlation between the speed and frequency. But in fact, higher frequency does not always result in faster speed.

Figure 8 is a plot of locomotion speed and average oscillation frequency of three module configuration from a different evolutionary process, evolving a frequency adaptive neural controller, where the initial generations had relative success with very high frequency, but were over taken by individuals with relatively lower frequency, in later generations. Observing figure 7 and figure 8 shows that the optimal locomotion speed on flat surface for a three module linear configuration is achieved at a frequency of about 1.3Hz to 1.5 Hz.

4.3 Experiment 3

Along with stable locomotion gait, we wanted to evolve limit cycle behavior such that the neural controller would have the ability to overcome external perturbations. To achieve this, every individual was evaluated five times, starting with random actuator angles for each evaluation run, and the final fitness of an individual was calculated as an average of the five evaluations. This way, those individuals who had the ability to produce similar oscillations on each run, would be consistent in their performance, and subsequently survive and reproduce more successfully.

We were able to test this by evaluating the best performing three module individual of the final generation with simulated external perturbation. Perturbations were simulated by replacing the output of the ANN with a random value, before passing the same to the actuator.

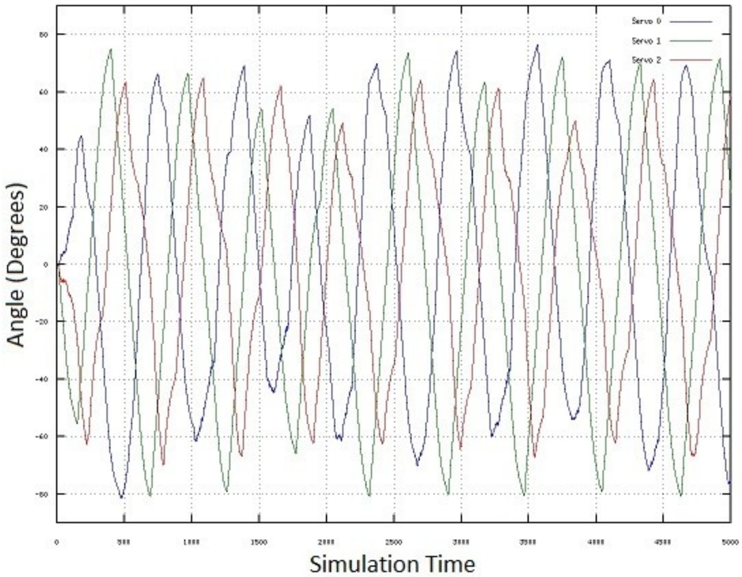

Fig. 6. Plot of the actuator values of a three module configuration with frequency adaptive neural actuation method.

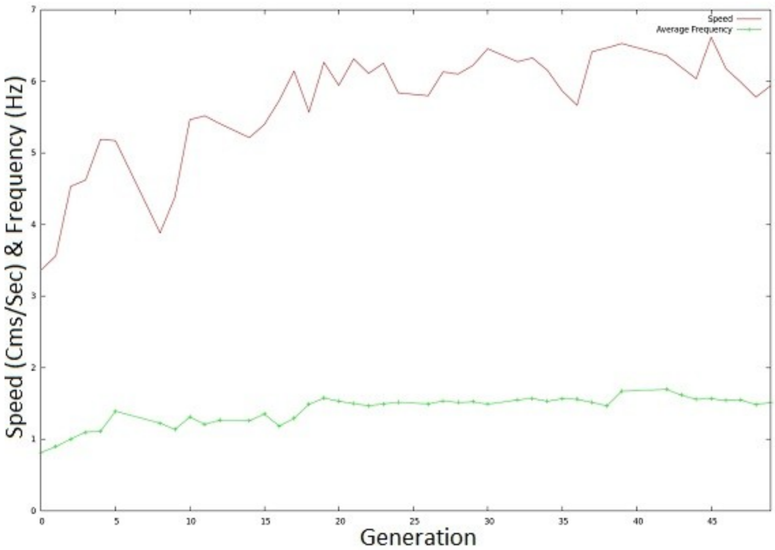

Fig. 7. Plot of the average oscillation frequency and the locomotion speed of the best performing individuals throughout the evolution of the frequency adaptive neural actuation method

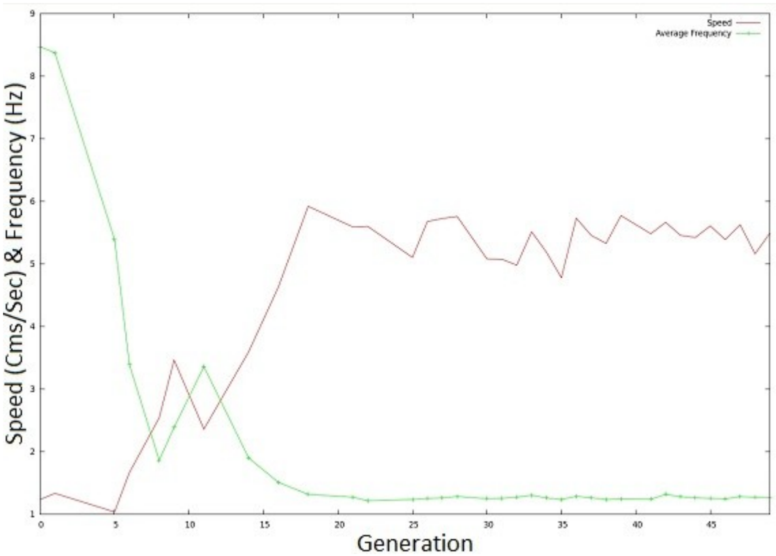

Fig. 8. Plot of the average oscillation frequency and locomotion speeed of the best performing individual of every generation.

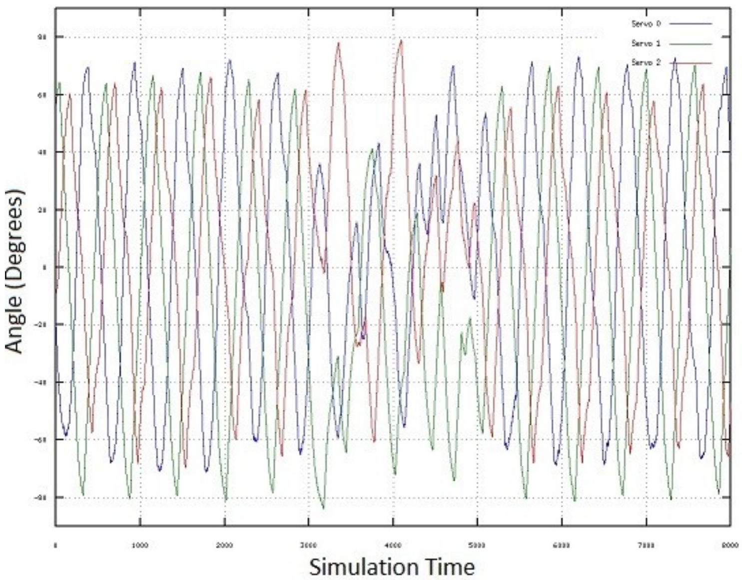

Fig. 9. Plot of the average oscillation frequency and locomotion speeed of the best performing individual of every generation.

Figure 9 contains the plot of the actuator value over time of an individual evaluated for 50 simulated seconds. External perturbations were introduced, for all the modules in the configuration, at an interval of 25 seconds (At the 3000 mark on X axis) for the next 20 consecutive cycles. As could be see, oscillations quickly converge back to their normal pattern (Within 5500 mark on the X axis), few cycles after the end of the external perturbation.

5 Conclusion and future work

In this preliminary investigation of the proposed neural model for locomotion in modular robotics, we have been able to successfully validate the model for producing locomotory oscillations in linear configurations, along with fault tolerance and limit cycle behavior. An evolved neural controller is able to consistently converge and settle into a stable oscillatory pattern, starting from a random actuator start angle, which results in a stable locomotion gait. We have been able to test out model only on a simulation environment. Going forward, we would like to validate the same on the real Y1 modular robots.

The hidden layer of the ANN in our model contains default architecture. We would like to focus on evolving both the topology and the weight of the ANN using the NEAT methodology, to both complexify the architecture for adaptive locomotion in different robotic configurations, and to investigate the minimal neural architecture required for locomotion.

References

Castano, A., Behar, A., Will, P. December 2002. "The Conro Modules for Reconfigurable Robots", IEEE/ASME Trans. Mechatronics. 7(4):403–409.

Diankov, R. September 2010. “Automated construction of robotic manipulation programs,” Ph.D. dissertation, Carnegie Mellon University, Robotics Institute.

Fukuda, T., Nakagawa, S., Kawauchi, Y., Buss, M. Nov 1988. “Self organizing robots based on cell structures-CEBOT,” in Proc. 1988 IEEE Int. Workshop Intell. Robots. New York: IEEE. pp. 145–150.

Fukuda, T., Kawauchi, Y. May 1990. “Cellular robotic system (CEBOT) as one of the realization of self-organizing intelligent universal manipulator,” in Proc. IEEE Int. Conf. Robotics and Automation, Cincinatti, OH. pp. 662–667.

Gonzalez-Gomez, J., Boemo, E. September 2005. Motion of Minimal Configurations of a Modular Robot: Sinusoidal, Lateral Rolling and Lateral Shift. Proc. of the 8th International Conference on Climbing and Walking Robots, CLAWAR, London. pp. 667-674..

Gonzalez-Gomez, J. November 2008. Modular Robotics and Locomotion: Application to Limbless robots. PhD thesis, EPS, UAM, Madrid, Spain.

Herrero-Carron, F., Rodriguez, F.B., Varona, P. 2011. "Bio-inspired design strategies for central pattern generator control in modular robotics". Bioinspiration & Biomimetics, vol. 6, no. 1, p. 016006.

Hou, F., Shen, W.-M. March 2006. Hormone-inspired Adaptive Distributed Synchronization of Reconfigurable Robots. In The 9th Intl. Conf. Intelligent and Autonomous Systems (IAS-9), Tokyo, Japan.

Ijspeert, A.J. 2008. "Central pattern generators for locomotion control in animals and robots: A review," Neural Networks, vol. 21, no. 4, pp. 642 – 653.

Kamimura, A., Kurokawa, H., Toshida, E., Tomita, K., Murata, S., Kokaji, S. 2003. Automatic locomotion pattern generation for modular robots. In IEEE International Conference on Robotics and Automation (ICRA2003).

Lal, S.P., Yamada, K., Endo, S. 2006. Studies on Motion Control of a Modular Robot Using Cellular Automata. In: Sattar, A., Kang, B.-H. (eds.) AI. LNCS (LNAI), vol. 4304, pp. 689–698. Springer.

Lal, S.P., Yamada, K., Endo, S. 2007. Evolving Motion Control for a Modular Robot. In: Ellis, R., Allen, T., Petridis, M. (eds.) Applications and Innovations in Intelligent Systems XV, pp. 245–258. Springer, London.

Lopez, R.: Flood: An Open Source Neural Networks C++ Library, www.cimne.com/flood (2010).

Moeckel, R., Jaquier, C., Drapel, K., Upegui, A., Ijspeert, A. J. 2005. YaMoR and Bluemove -- an autonomous modular robot with Bluetooth interface for exploring adaptive locomotion. Proceedings CLAWAR.

Pouya, S., van den Kieboom, J., Spröwitz, A., Ijspeert, A. J. 2010. Automatic Gait Generation in Modular Robots: to Oscillate or to Rotate? that is the question. Proceedings of IEEE/RSJ IROS, Taipei, Taiwan. 18-22.

Salemi, B., Shen, W.-M., Will, P. 2001. Hormone-Controlled Metamorphic Robots. in International Conference on Robotics and Automation. Seoul, Korea.

Salemi, B., Moll, M., Shen, W.-M. 2006. "SUPERBOT: A deployable, multifunctional, and modular self-reconfigurable robotic system," in Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst., Beijing, China. pp. 3636–3641.

Shen, W.-M., Salemi, B., Will, P. 2000. Hormones for Self-Reconfigurable Robots. in Intelligent Autonomous Systems. Venice, Italy. pp. 918–925.

Shen, W.-M., Salemi, B., Will, P. 2002. Hormone-inspired adaptive communication and distributed control for conro self-reconfigurable robots. IEEE Transactions on Robotics and Automation.

Sproewitz, A., Moeckel, R., Maye, J., Ijspeert, A.J. 2008. Learning to move in modular robots using central pattern generators and online optimization, Int. J. Rob. Res., vol. 27, num. 3-4, p. 423--443.

Yim, M. 1994. Locomotion with a unit-modular reconfigurable robot. Ph.D dissertation, in Department of Mechanical Engineering, Stanford University.

Zhang, H., González-Gómez, J., Zhang, J. 2009. "A New Application of Modular Robots on Analysis of Caterpillar-like Locomotion". Proceedings of the 2009 IEEE International Conference on Mechatronics, Malaga, Spain, 14-17 April. I.S.B.N 978-1-4244-4195-2.