Proceedings of the 2014 IEEE
International Conference on Robotics and Biomimetics
December 5-10, 2014, Bali, Indonesia

A portable stand-alone bi-hemispherical neuronal network model of the cerebellum for adaptive robot control

Ruben-Dario Pinzon-Morales, *Member, IEEE*,Yutaka Hirata, *Member, IEEE*

***Abstract*—Development of computational models of the brain is relevant not only for deepening our understanding of the biological system but also for potential applications to various engineering problems. In this paper the implementation of a bi-hemispherical neuronal network model of the cerebellum (biCNN) in a stand-alone, portable real time (RT) device is presented. The biCNN is tested during a control engineering application, namely, control of a highly unstable two-wheel balancing robot. The RT device considered is the National Instruments myRIO-1900, which provides flexibility and portability to the biCNN. Execution times obtained with the RT device are compared with a personal computer implementation as reference. The results demonstrate the suitability of the RT implementation of the biCNN for robot control, and provide a successful bridge between the cerebellar research and engineering.**

I. Introduction

Control of humanoid robots is one of many engineering applications that have been challenged not only by the engineering community but also by neuroscientists. They provide a unique combination of complexity and similarities with the human body that make them an appealing workbench for understanding how the brain works [1]. Computational models inspired by the brain have been shown to provide both efficient solutions to engineering problems and insights into the function of the biological systems [1], [2]. One brain structure extensively studied and modeled is the cerebellum [3], which is involved in cognition [4], [5], [6] and in motor learning and coordination [7], [8], [9], [3], [10]. Computational models of the cerebellum have been successfully employed in motor control of inverted pendulums [2], [11], robotic arms [12], [13], wheeled robots [14], humanoid robots [1], [15], [16], among other engineering applications [17].

Due to the computational power required for simulating cerebellar models, the favored target for their implementation has commonly been personal computers, computer clusters, or graphical processor units (GPUs) [12], [15]. However dedicated hardware implementations such as those using field programmable gate arrays (FPGAs) [18] or VLSI circuits [19] present extra benefits (e.g., speed, real time operation, portability, and low power consumption) that make them alluring for engineering applications. In this document we show the first implementation of a bi-hemispherical neuronal network of the cerebellum (biCNN)[2] in a stand-alone, portable real time (RT) device. The RT device considered were the National Instruments myRIO-1900. As an example application, the biCNN is tested during a control of a highly unstable two-wheel balancing robot. Comparison of the execution time required by the RT device and a personal computer with different number of neurons and synapses in the biCNN is also provided for reference.

R. D. Pinzon-Morales and Y. Hirata are with the University of Chubu, Japan. rdpinzonm@ieee.org, yutaka@isc.chubu.ac.jp

II. Materials and Methods

Fig. 1 shows, the general scheme of the biCNN used for controlling a highly unstable two-wheel balancing robot. It includes the RT device (Nation instruments myRIO-1900) where the biCNN is deployed, the control object (ZMP INC, enuvo-wheel), and a laptop for remote programming and surveillance.

A. General description of the biCNN model

We have previously configured and tested the validity of the biCNN, which is inspired by the physio-anatomical neuronal microcircuit of the cerebellar cortex [2]. Briefly, the biCNN comprises granular (gr) cells, Golgi (Go) cells, basket and stellate (ba) cells, and Purkinje (Pk) cells in two hemispheres (Fig. 1). It receives two types of inputs carried by mossy fibers (mfs) and a climbing fiber (cf) as in the real cerebellum. mfs carry desired motion, sensory errors and a copy of the control action (efference copy of motor command), whereas the cf carries the error signal required for cerebellar learning in terms of long term potentiation (LTP) and depression (LTD) of synapses between gr and Pk cells. Synaptic connections include excitatory projection from mfs to gr and Go, and from gr to ba and Go, inhibitory feedback loop between gr and Go and Go and ba, and feed-forward mutual inhibitory loop between ba and Pk cells [2]. A proportional and derivative (PD) controller that is a feedback controller widely used in industry and other applications is included to provide the non cerebellar and non adaptive contribution to produce the control action sent to the control object (Fig. 1, PD).

B. AER implementation of the biCNN

Address Event Representation (AER) is a compact and efficient (in terms of memory consumption) communication technique for sparse networks and has been successfully extrapolated to neural networks [20]. In AER four vectors are required to describe the network architecture (Fig. 2, bottom). The first vector encodes the neurons in the network assigning a unique ID, the second vector stores the number of pre-synapses for each neuron, and the third and forth vector encode the IDs of the pre-synaptic neurons and the corresponding synaptic weights in a stacked ordered way. For the example

978-1-4799-7397-2/14/$31.00 © 2014 IEEE

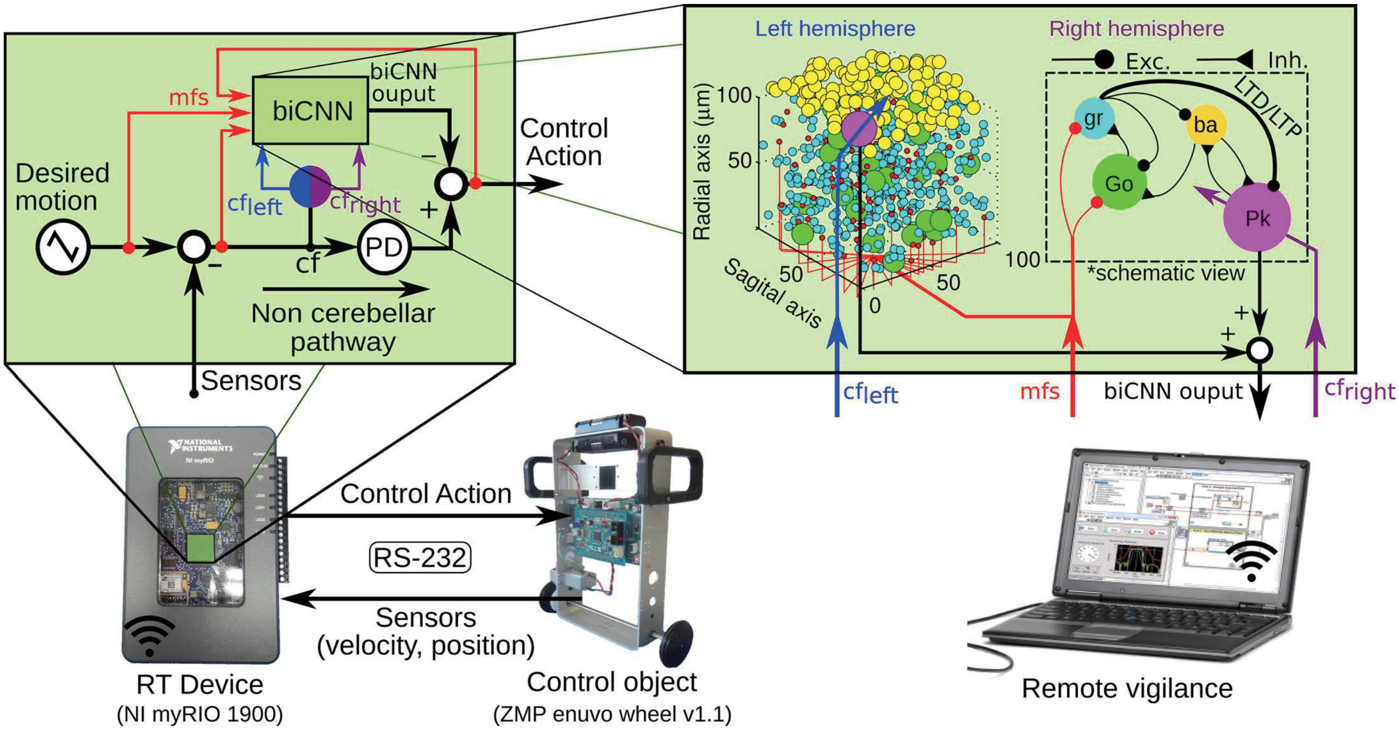

Fig. 1. Implementation of the biCNN using a NI myRIO RT device for control of a two wheel balancing robot. ba, basket/Stellate cells; Exc, excitatory synapses; Inh, inhibitory synapses; Go, Golgi cell; gr, granular cell; mfs, mossy fibers; PD, proportional and derivative controller; Pk, Purkinje cell; RT, real time.

network in Fig. 2, neuron #3 is reached by two neurons (#2, #1); this information is clearly seen in the third element of the vectors ID and number of (pre)synapses respectively (Fig. 2 red arrows). The synapses reaching this neuron and their synaptic weights can be read by first accumulating the number of synapses of the neurons that precede neuron #3 (neuron #1, 2 pre- synapses, neuron #2, 1 pre-synapse) yielding the indexes [3, 4], and then looking up those indexes in the vectors (pre)synapses IDs and synaptic weight. The AER used in the biCNN stores the pre-synaptic connections rather than the post-synaptic connections (as was originally presented in [20]) because it is more suitable for dataflow based programming frameworks such as LabVIEW (National Instruments, Austin, TX).

C. Real time device and control object

The real time device selected to implement the biCNN is the National Instruments myRIO-1900 (National Instruments, Austin, TX) shown in Fig. 1. It includes a dual-core ARM Cortex-A9 real-time processor at 667 MHz, 512 MB of DDR3 volatile memory, 2.4 GHz IEEE 802.11 port for wifi remote programming/surveillance, and several multi-purpose input/output ports. In addition the stand-alone myRIO is readily compatible with LabVIEW programming code.

The control object corresponds to a two-wheel balancing robot (enuvo wheel, ZMP INC, Tokyo) (Fig. 1), which is an inverted pendulum system widely used in control engineering for testing control strategies, because of its highly unstable dynamics. It is considered one of the most challenging control plants [21]. The robot is equipped with a motor encoder and a gyroscope for measuring wheel angle ($\phi(t)$) and body tilt angle, respectively. The robot includes a UART serial port for communication with the myRIO device at 100 Hz.

III. Results

A. Robot control experiments

The robot was commanded to follow 100 cycles of a sinusoidal desired wheel motion ($(\phi(t)_{des.} = \pi sin(2\pi 0.25t)$)

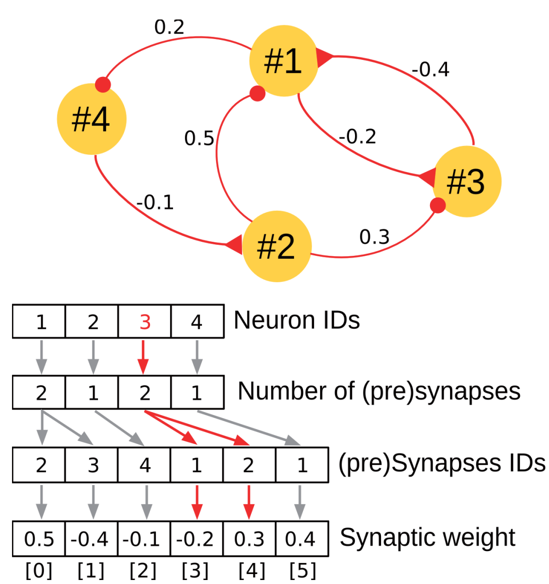

Fig. 2. Example of a generic neuronal network with four cells and its representation using AER.

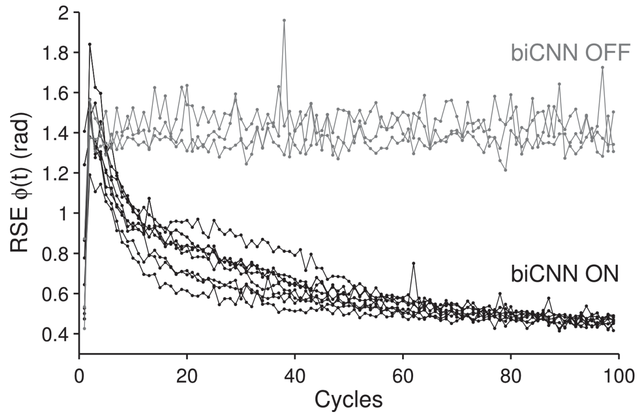

Fig. 3. Comparison of the control performance achieved in terms of RSE of $\phi(t)$ when the biCNN is enable (dark traces, N=10) and disabled (gray traces, N=3).

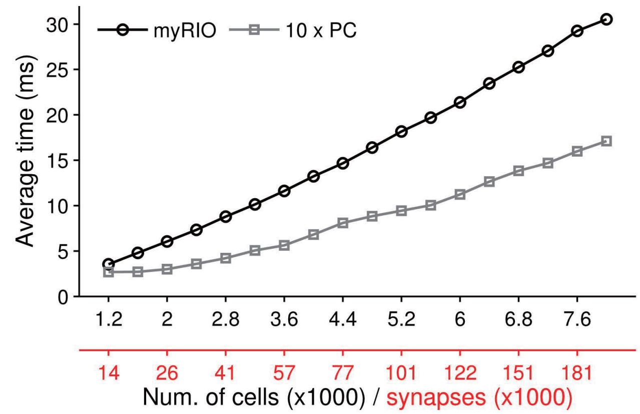

Fig. 4. Average execution time taken by the myRIO and a Windows PC versus the number of cells/synapses in the biCNN.

while the body tilt angle was desired to be constant (90 degrees with respect to the horizontal plane). The number of neurons in each hemisphere of the biCNN was set to 300 gr, 2 Go, 20 ba, 1 Pk. These numbers of neurons have been shown to be an optimal selection for the current control object [22]. Variability due to the initial conditions in the biCNN, namely, the set of initial random synaptic weights and the synaptic connections, was taken into account by creating 25 permutations of five different sets of random synaptic weights and five different AER tables (i.e., synaptic connections) [22]. Control performance was measured in terms of the root square error (RSE) of the control variable $\phi(t)$. Fig. 3 summarizes the control performance attained in two different cases. One when the biCNN was active (N=10, black traces) and the other when it was inactive (i.e., only the PD controller governed the robot, N=3, gray traces). This figure demonstrates the adaptation happening in the biCNN and the general improvement in control performance.

B. Execution time in the RT device

In the experiments presented here a total of 646 neurons were employed because it is an optimal size for the current control task and control object [22]. However, larger biCNNs can be deployed using the myRIO-1900 device. We measured the average processing time for one execution of the biCNN of different sizes using the resident Profiler tool in LabVIEW. For reference purpose, the biCNN is also executed on a Windows laptop (Intel Core i5-2430M CPU @ 2.40GHz x 4, 8 GB DDR3 RAM). LabVIEW version 2013 was used during all the experiments. Fig. 4 shows the average time required by the myRIO and the Laptop (labeled *PC* multiplied by 10) versus the number of neuron cells and synapses. It evidences that the execution time in the RT device grows linearly with the number of synapses in the biCNN, whit a steeper gradient than in the PC. This figure serves as reference for choosing the maximum number of neurons in the biCNN for a given control scenario. For instance, the two-wheel robot used here has a sampling frequency of 10 ms. The execution time has to be below the sampling frequency and therefore the biCNN could include up to 3 k neurons with more than 57 k synapses in a myRIO device or more than 20 k and 380 k synapses in a PC. Slower sampling frequencies would allow larger size of biCNNs whereas faster sampling frequencies would require smaller size.

IV. Conclusion

We presented in this document the implementation of our bi-hemispherical neuronal network model of the cerebellum in a stand-alone real time device. The biCNN, as we referred to the current implementation, was tested during control of a two-wheel balancing robot, outperforming the control performance attained when using a classical controller (PD controller). The RT device chosen (National Instruments myRIO-1900) brought flexibility and portability to the biCNN. To the best of our knowledge this is the first implementation of a realistic bi-hemispherical model of the cerebellum in a dedicated hardware device for solving a real world engineering problem. The biCNN, which is open source and available upon contact (http://nclab.solan.chubu.ac.jp/nclab/), provides a bridge between neuroscience and engineering.

Acknowledgment

This study is supported by MEXT grant-in-aid (YH). RDPM is supported by the MEXT scholarship for postgraduate studies 2011.

References

[1] M. Kawato, “From understanding the brain by creating the brain towards manipulative neuroscience,” *Philos Trans R Soc Lond B Biol Sci*, vol. 363, no. 1500, pp. 2201–2214, 2008.

[2] R. Pinzon-Morales and Y. Hirata, “A bi-hemispheric neuronal network model of the cerebellum with spontaneous climbing fiber firing produces asymmetrical motor learning during robot control,” *Frontier in Neural Circuits*, vol. 8, no. 131, 2014.

[3] M. Ito, *The Cerebellum: Brain for an Implicit Self*, ser. FT Press Science. Pearson Education, 2011.

[4] W. T. Thach, “What is the role of the cerebellum in motor learning and cognition?” *Trends in Cognitive Sciences*, vol. 2, no. 9, pp. 331 – 337, 1998.

[5] F. V. Overwalle, K. Baetens, P. Marian, and M. Vandekerckhove, “Social cognition and the cerebellum: A meta-analysis of over 350 fmri studies,” *NeuroImage*, vol. 86, pp. 554–572, 2013.

[6] M. Manto, J. D. Schmahmann, F. Rossi, D. L. Gruol, and N. Koibuchi, *Handbook of the Cerebellum and Cerebellar Disorders*. Springer Netherlands, 2013.

[7] W. T. Thach, "On the specific role of the cerebellum in motor learning and cognition: Clues from pet activation and lesion studies in man," *Behavioral and Brain Sciences*, vol. 19, pp. 411–433, 1996.

[8] R. Spencer, R. Ivry, and H. Zelaznik, "Role of the cerebellum in movements: control of timing or movement transitions?" *Experimental Brain Research*, vol. 161, no. 3, pp. 383–396, 2005.

[9] S. Highstein, J. Porrill, and P. Dean, "Report on a workshop concerning the cerebellum and motor learning," *The Cerebellum*, vol. 4, pp. 140–150, 2005.

[10] M. Manto, J. Bower, A. B. Conforto, J. Delgado-Garca, S. Guarda, M. Gerwig, C. Habas, N. Hagura, R. Ivry, P. Marin, M. Molinari, E. Naito, D. Nowak, N. Oulad Ben Taib, D. Pelisson, C. Tesche, C. Tilikete, and D. Timmann, "Consensus paper: Roles of the cerebellum in motor control: the diversity of ideas on cerebellar involvement in movement," *The Cerebellum*, vol. 11, no. 2, pp. 457–487, 2012.

[11] C.-H. Chiu, "The design and implementation of a wheeled inverted pendulum using an adaptive output recurrent cerebellar model articulation controller," *IEEE Transactions on Industrial Electronics*, vol. 57, no. 5, pp. 1814–1822, 2010.

[12] R. R. Carrillo, E. Ros, C. Boucheny, and O. J.-M. Coenen, "A real-time spiking cerebellum model for learning robot control," *Biosystems*, vol. 94, no. 12, pp. 18 – 27, 2008.

[13] C. Casellato, A. Pedrocchi, J. Garrido, N. Luque, G. Ferrigno, E. D'Angelo, and E. Ros, "An integrated motor control loop of a human-like robotic arm: Feedforward, feedback and cerebellum-based learning," in *Biomedical Robotics and Biomechatronics (BioRob), 2012 4th IEEE RAS EMBS International Conference on*, June 2012, pp. 562–567.

[14] C. Hofstotter, M. Mintz, and P. F. M. J. Verschure, "The cerebellum in action: a simulation and robotics study," *European Journal of Neuroscience*, vol. 16, no. 7, pp. 1361–1376, 2002.

[15] T. Yamazaki and J. Igarashi, "Realtime cerebellum: A large-scale spiking network model of the cerebellum that runs in realtime using a graphics processing unit," *Neural Networks*, vol. 47, pp. 103 – 111, 2013.

[16] A. Lenz, S. Anderson, A. Pipe, C. Melhuish, P. Dean, and J. Porrill, "Cerebellar-inspired adaptive control of a robot eye actuated by pneumatic artificial muscles," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 39, no. 6, pp. 1420–1433, 2009.

[17] C.-M. Wen and M.-Y. Cheng, "Development of a recurrent fuzzy cmac with adjustable input space quantization and self-tuning learning rate for control of a dual-axis piezoelectric actuated micromotion stage," *Industrial Electronics, IEEE Transactions on*, vol. 60, no. 11, pp. 5105–5115, 2013.

[18] L. Bobo, I. Herreros, and P. Verschure, "A digital neuromorphic implementation of cerebellar associative learning," in *Biomimetic and Biohybrid Systems*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2012, vol. 7375, pp. 13–25.

[19] C. Hofstoetter, M. Gil, K. Eng, G. Indiveri, M. Mintz, J. Kramer, and P. F. M. J. Verschure, "The cerebellum chip: an analog vlsi implementation of a cerebellar model of classical conditioning," in *NIPS'04*, vol. 17, 2004, pp. 577–584.

[20] C. Johansson and A. Lansner, "Towards cortex sized artificial neural systems," *Neural Networks*, vol. 20, no. 1, pp. 48–61, 2007.

[21] Z. Li, C. Yang, and L. Fan, *Advanced Control of Wheeled Inverted Pendulum Systems*. Springer, 2013.

[22] R. Pinzon-Morales and Y. Hirata, "The number of granular cells in a cerebellar neuronal network model engaged during robot control increases with the complexity of the motor task," *BMC Neuroscience*, vol. 15, no. Suppl 1, p. 143, 2014.