Neural Networks 47 (2013) 103–111

Contents lists available at ScienceDirect

Neural Networks

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

2013 Special Issue

Realtime cerebellum: A large-scale spiking network model of the cerebellum that runs in realtime using a graphics processing unit

Tadashi Yamazaki [a,*], Jun Igarashi [b]

[a] RIKEN Brain Science Institute, 2-1 Hirosawa, Wako, Saitama 351-0198, Japan

[b] Computational Science Research Program, RIKEN, 2-1 Hirosawa, Wako, Saitama 351-0198, Japan

ARTICLE INFO

Keywords:
Graphics processing unit
Realtime simulation
Spiking network model
Cerebellar microcomplex
Reservoir computing
Robotics

ABSTRACT

The cerebellum plays an essential role in adaptive motor control. Once we are able to build a cerebellar model that runs in realtime, which means that a computer simulation of 1 s in the simulated world completes within 1 s in the real world, the cerebellar model could be used as a realtime adaptive neural controller for physical hardware such as humanoid robots. In this paper, we introduce "Realtime Cerebellum (RC)", a new implementation of our large-scale spiking network model of the cerebellum, which was originally built to study cerebellar mechanisms for simultaneous gain and timing control and acted as a general-purpose supervised learning machine of spatiotemporal information known as reservoir computing, on a graphics processing unit (GPU). Owing to the massive parallel computing capability of a GPU, RC runs in realtime, while reproducing qualitatively the same simulation results of the Pavlovian delay eyeblink conditioning with the previous version. RC is adopted as a realtime adaptive controller of a humanoid robot, which is instructed to learn a proper timing to swing a bat to hit a flying ball online. These results suggest that RC provides a means to apply the computational power of the cerebellum as a versatile supervised learning machine towards engineering applications.

1. Introduction

The cerebellum plays an essential role in motor learning and control. In an engineering term, the cerebellum is thought as an "adaptive control device", which observes the status of body parts by sensors continuously and calibrates the movement online to achieve fast and smooth motor control. The ability of adaptive control is unique to living organisms, and is expected its engineering applications such as humanoid robot control for flexible movements. Several attempts have been made to date. Feedback-error learning is a system-level approach to adopt the adaptive control ability for humanoid robots (Kawato & Gomi, 1992; Miyamoto, Kawato, Setoyama, & Suzuki, 1988; Shibata & Shaal, 2001). In those studies, a three-layer perceptron with rate-coding neurons was employed as a model of the cerebellum. Other studies have built spiking network models, and adopted to control a navigation robot that avoids hitting at the wall (Hofstötter, Mintz, & Verschure, 2002) and a robot arm with 2 joints to perform reaching tasks (Carrillo, Ros, Boucheny, & Coenen, 2008). Those models, however, focus primarily on engineering applications. It remains unknown whether they reproduce experimental results of, for example, Pavlovian delay eyeblink conditioning and gain adaptation of vestibulo-ocular reflex. Moreover, for the sake of computational time, the network size of those models is relatively smaller than the other cerebellar models which aim to reproduce experimental results.

On the other hand, we have built a large-scale spiking network model of the cerebellum, which is composed of more than 100,000 spiking neuron units with realistic parameters. The model has been demonstrated to reproduce experimental results of Pavlovian delay eyeblink conditioning (Yamazaki & Tanaka, 2007b) and gain adaptation of optokinetic response eye movements (Yamazaki & Nagao, 2012), suggesting that our cerebellar model can learn and control gain and timing information adaptively. Our cerebellar model could be adopted to such real-world applications as well, if the computer simulation is made in real time.

A graphics processing unit (GPU) is hardware designed and optimized for graphics, video, and visual computing in 2D and 3D (Patterson & Hennessy, 2011). The architecture consists of two components, one for graphics and the other for numerical calculation, which turns a GPU into a programmable graphics processor as well as a scalable parallel computational platform. CUDA (Compute Unified Device Architecture) (NVIDIA, 2011), a unified software development environment for GPUs, allows us to use a GPU as a highly-parallel, multi-threaded multiprocessor. GPUs have been already employed in the field of computational

* Correspondence to: Graduate School of Informatics and Engineering, The University of Electro-Communications, 1-5-1 Chofugaoka, Chofu, Tokyo 182-8585, Japan. Tel.: +81 42 443 5000.
E-mail address: nn12@neuralgorithm.org (T. Yamazaki).

0893-6080/$ – see front matter © 2013 Elsevier Ltd. All rights reserved.
doi:10.1016/j.neunet.2013.01.019

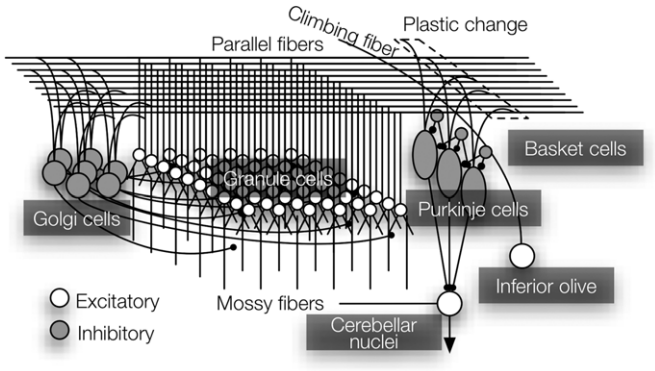

Fig. 1. Network structure of our cerebellar model.

neuroscience (Goodman & Brette, 2009; Igarashi, Shouno, Fukai, & Tsujino, 2011; Miikkulainen, Bednar, Choe, & Sirosh, 2005; Nageswaran, Dutt, Krichmar, Nicolau, & Veidenbaum, 2009). A study demonstrates realtime simulation of a detailed basal ganglia model for decision making (Igarashi et al., 2011).

In this study, we re-implement our cerebellar model on a GPU, so as to carry out the computer simulation in real time. Using some techniques on implementation, the new model, which we call "Realtime Cerebellum (RC)" can run in real time: a simulation of 1 s in the simulated world completes within 1 s in the real world. We carry out computer simulation of Pavlovian delay eyeblink conditioning, and confirm that RC reproduces qualitatively the same results with the previous model. We also adopt RC to hardware control to demonstrate the power of realtime computing and delay compensation in sensorimotor loop. We set up a robot experiment, in which a small humanoid robot is instructed to hit a flying ball thrown by a pitching machine by swinging a bat at hand. The robot gradually learns the correct timing by repetition of practice and finally succeeds to hit the ball.

2. Materials and methods

2.1. Overview of our cerebellar model

RC, the cerebellar model we implemented on a GPU in this study, is based on our previous models (Yamazaki & Nagao, 2012; Yamazaki & Tanaka, 2007b). Briefly, RC is composed of 102,400 granule cells, 1024 Golgi cells, 16 Purkinje cells, 16 basket cells, 1 inferior olive and 1 neuron in the cerebellar nucleus (Fig. 1). External inputs are fed by mossy fibers to granule cells and the nucleus. Granule cells excite Golgi cells, Purkinje cells and basket cells. In turn, Golgi cells and basket cells inhibit granule cells and Purkinje cells, respectively. All the 16 Purkinje cells inhibit the nuclear cell. Other external inputs are fed by climbing fibers to Purkinje cells. The final output of the network is generated by the nucleus. Granule cell–Purkinje cell synapses undergo plastic change (long-term depression and potentiation).

Neurons are modeled as conductance-based, leaky integrate-and-fire units.

$$C\frac{dV(t)}{dt} = -g_{\mathrm{leak}}(V(t)-E_{\mathrm{leak}}) - g_{\mathrm{ex:AMPA}}(t)(V(t)-E_{\mathrm{ex}}) - g_{\mathrm{ex:NMDA}}(t)(V(t)-E_{\mathrm{ex}}) - g_{\mathrm{inh}}(t)(V(t)-E_{\mathrm{inh}}) - g_{\mathrm{ahp}}(t-\hat{t})(V(t)-E_{\mathrm{ahp}}), \quad (1)$$

where $V(t)$ and C are the membrane potential at time t and the capacitance, respectively. The membrane potential is determined by five types of currents specified by the right-hand side of Eq. (1), namely, leak, alpha-amino-3-hydroxy-5-methyl-4-isoxazolepropionic acid receptor (AMPAR)-mediated, NMDAR-mediated and gamma-aminobutyric acid type A receptor ($GABA_AR$)-mediated currents, and the current for emulation of the after-hyperpolarization. For each type $c \in \{\mathrm{leak}, \mathrm{ex:AMPA}, \mathrm{ex:NMDA}, \mathrm{inh}, \mathrm{ahp}\}$, the current at a given time is calculated with the conductance g_c and reversal potential E_c. The conductance is calculated by the convolution of the alpha function $\alpha(t)$ and the spike event $\delta_j(t)$ of presynaptic neuron j at time t as follows:

$$g_c(t) = \sum_j w_j \int_{-\infty}^{t} \bar{g}_c \alpha(t-s)\delta_j(s)ds, \quad (2)$$

where $\bar{g}_c$ represents the maximum conductance and w_j the synaptic weight from the presynaptic neuron j. The alpha functions are defined for each current and each neuron type with different time constants. When the membrane potential of a neuron exceeds the threshold θ, the neuron is supposed to elicit a spike, followed by the after-hyperpolarization that determines a refractory period. The conductance for the after-hyperpolarization is given by

$$g_{\mathrm{ahp}}(t-\hat{t}) = \exp(-(t-\hat{t})/\tau_{\mathrm{ahp}}), \quad (3)$$

where τ_{ahp} represents the time constant of the after-hyperpolarization and $\hat{t}$ is the last firing time of the neuron. Detailed parameters are described in our previous papers (Yamazaki & Nagao, 2012; Yamazaki & Tanaka, 2007b).

For each connection between two neurons, a constant called a synaptic weight is assigned; detailed values are shown in our previous paper (Yamazaki & Nagao, 2012). These synaptic weights do not change during the whole computer simulation, except those between parallel fibers (granule cell axons) and Purkinje cells. When a parallel fiber is solely activated, the weight increases slightly, whereas when the activation is paired with that of a climbing fiber, the weight decreases slightly. This bidirectional change models long-term potentiation (LTP) (Coesmans, Weber, De Zeeuw, & Hansel, 2004; Lev-Ram, Mehta, Kleinfeld, & Tsien, 2003) and long-term depression (LTD) (Ito, 2001, 2002), respectively. The equation for LTP/LTD is shown in our previous paper (Yamazaki & Nagao, 2012; Yamazaki & Tanaka, 2007b).

We have hypothesized that, the recurrent inhibitory network composed of granule and Golgi cells generates various temporally-fluctuating spike patterns among granule cells in response to mossy fiber signals (Yamazaki & Tanaka, 2005). Because different granule cells exhibit different temporal patterns, the population of active granule cells changes gradually in time, indicating that there is a one-to-one correspondence between a granule-cell population and a time step from the onset of mossy fiber signals. Therefore, the temporal evolution of active granule-cell populations can represent the passage of time from the mossy fiber signal onset. To study how the spike patterns of granule cells evolve over time, we define two indices. Let $z_i(t)$ be the population average activity of a granule-cell cluster i, which is defined by a set of nearby granule cells sharing the same inhibitory inputs from Golgi cells via glomeruli (see Yamazaki & Nagao, 2012; Yamazaki & Tanaka, 2007b for details):

$$z_i(t) = \frac{1}{\tau_{\mathrm{PKJ}}} \sum_{s=0}^{t} \exp\left(-(t-s)/\tau_{\mathrm{PKJ}}\right) \left(\frac{1}{N_{\mathrm{granule\ per\ cluster}}} \sum_j \delta_j(s) \right), \quad (4)$$

where $\delta_j(s)$ represents the spike elicited by model granule cell j in the cluster, $N_{\mathrm{granule\ per\ cluster}}$ is the number of granule cells in a cluster (namely, 100), and τ_{PKJ} is a decay time constant of AMPAR-mediated EPSPs at Purkinje cells, which was set at 8.3 ms. We define the autocorrelation of the activity pattern at times t and $t+\tau$ as follows: