

# Realtime Simulation of a Cerebellar Spiking Network Model Towards Neuroprosthesis

**Tadashi Yamazaki**

Graduate School of Informatics and Engineering

The University of Electro-Communications

1-5-1 Chofugaoka, Chofu, Tokyo 182-8585, JAPAN.

## Abstract

Neuroprosthesis aims to supersede a damaged or degenerated brain caused by accidents or aging by an artificial brain that simulates and thereby restores the impaired brain functions. To replace the real brain, the artificial brain has to simulate the same functions of the real brain, and the simulation has to be conducted in realtime. We have built a large-scale spiking network model of the cerebellum that is composed of more than 100,000 neuron units and acts as a versatile supervised learning machine for spatiotemporal information. We implement it on a graphics processing unit (GPU) to conduct the numerical simulation in realtime owing to the parallel computing capability of GPUs. We propose to use the present model towards neuroprosthesis.

## Introduction

Neuroprosthesis is an emerging area of neuroscience, which provides technology to replace a damaged or degenerated brain caused by accidents or aging with an external apparatus such as an artificial brain to simulate and thereby restoring the impaired brain functions. A typical neuroprosthesis is composed of three parts: (1) recording neural activity from the brain to provide inputs to an artificial brain (input) (2) simulating the computation of the real brain by the artificial one in response to the input signals (computation), and (3) stimulating neurons according to the output of the artificial brain (output). (1) and (3) requires electrophysiological techniques to provide interfaces between the real and artificial brains. (2) requires wide range of knowledge among computational neuroscience, machine learning, and artificial intelligence. In this article, we focus on (2).

The artificial brain must satisfy the following two conditions. First, it must compute the same function as that the real brain does. Otherwise, the output (3) is no longer reliable. Second, the computation must be made in realtime. Otherwise, the pathway from (1) to (3) does not work properly due to the delay. How can the artificial brain simulate the real brain in realtime?

The cerebellum plays an essential role in fast and smooth motor control and on-line motor adaptation (Ito 2011). Computationally, the cerebellum is a supervised learning ma-

chine that receives two inputs from mossy fibers and climbing fibers and climbing fibers as context and teacher information, respectively, and issue the output from the cerebellar nucleus (Fig. 1). The computational mechanism has been identified as equivalent to that of a liquid state machine, which is a class of supervised learning machines developed in a machine learning community (Yamazaki and Tanaka 2007). We have built a large-scale spiking network model of the cerebellum composed of more than 100,000 neurons (Yamazaki and Nagao 2012). If the cerebellar model runs in realtime, this can be used as a device for neuroprosthesis.

A Graphics Processing Unit (GPU) is a hardware accelerator for computer graphics, and now is used as a means for general-purpose parallel computation owing to its capability of creating a massive number of light-weight parallel computing streams called threads and its affordable prices. GPUs have been used in the field of computational neuroscience to scale up models and to make computer simulations faster (sometimes in realtime (Igarashi et al. 2011)). In this study, we re-implemented our previous model on a GPU. We achieved realtime computer simulations of the present model. We also succeeded to control a humanoid robot to hit a ball thrown by a pitching machine through on-line learning of a proper timing to swing a bat, suggesting the capability of the present model towards neuroprosthesis.

## Methods

The model cerebellum is the one we have already reported (Yamazaki and Nagao 2012) (Fig 1). Briefly, the model is composed of 102,400 granule cells, 1024 Golgi cells, 16 Purkinje cells, 16 basket cells, 1 nuclear cell and 1 inferior olive neuron. All neurons are modeled as conductance-based leaky integrate-and-fire units with simulated physiological parameters, and are connected based on the anatomical structure of the cerebellum. Mossy fibers and a climbing fiber provides external inputs to the network, whereas the nuclear cell provides the output. Granule and Golgi cells constitutes a random recurrent inhibitory network; granule cells excite Golgi cells, which in turn inhibit granule cells. Owing to the random connections, granule cells elicit spikes intermittently and randomly in response to mossy fiber stimuli. Different granule cells exhibit different temporal activity patterns, so that the population of

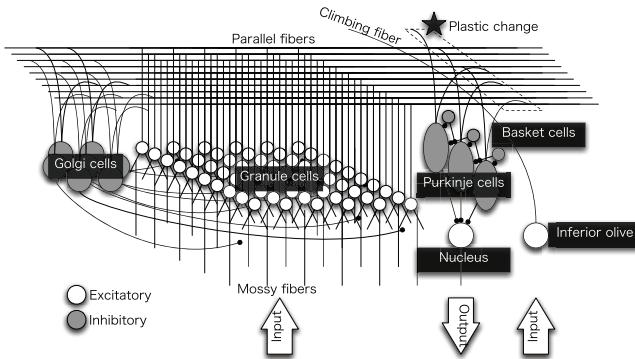


Figure 1: Schematic of the cerebellar cortical circuit

active granule cells gradually change in time without recurrence, which can code the passage of time from the onset of mossy fiber stimuli (Yamazaki and Nagao 2012). Parallel fiber-Purkinje cell synapses undergo simulated synaptic plasticity. When a parallel fiber is solely activated, the synaptic weight of the fiber is increased to simulate long-term potentiation. When a parallel fiber is activated conjunctively with climbing fiber activation, the synaptic weight is decreased to simulate long-term depression.

The program was written in C with NVIDIA CUDA Software Development Kit (NVIDIA 2011) and executed on NVIDIA GeForce GTX 580. GTX 580 has 480 processors called “cores”, on which a massive number of computing streams called “threads” are invoked. As a natural choice, we assigned one thread to one neuron and performed the calculation in parallel. Differential equations were numerically solved using 2nd order Runge-Kutta method with a fixed time step of 1 ms. Floating-point arithmetic was made in single-precision float, which is supported by a GPU.

## Results

We adopted the present cerebellar model to online learning of timing and control of a humanoid robot. We employed a hobby-purpose humanoid robot (Kondo-Kagaku KHR-3HV, 22 degrees of freedom) to swing a bat to hit a ball thrown by a hobby-use pitching machine (Fig. 2). We attached an

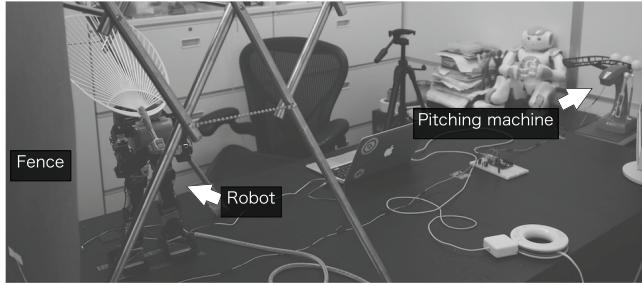


Figure 2: Setting of a robot experiment.

accelerometer to the pitching machine to detect the time to

throw a ball. We attached another accelerometer to a fence behind the robot as well to detect the failure of the robot to hit the ball and the time that the ball passed. This is an analogous to the eyeblink conditioning, in which an animal is conditioned to close the eye in response to a conditioned stimulus (CS) such as a tone by presenting the tone with an unconditioned stimulus (US) such as an airpuff to the eye (Christian and Thompson 2003). We set the firing rate of mossy fibers at 30 Hz when a ball was thrown, simulating the CS. We fed a strong transient stimulus to the inferior olive to elicit a climbing fiber signal when a ball reached to the fence, simulating the US. Purkinje cells learned to pause in advance to the US onset (Fig. 3), which triggered to play the preprogrammed swing motion of the robot. We confirmed

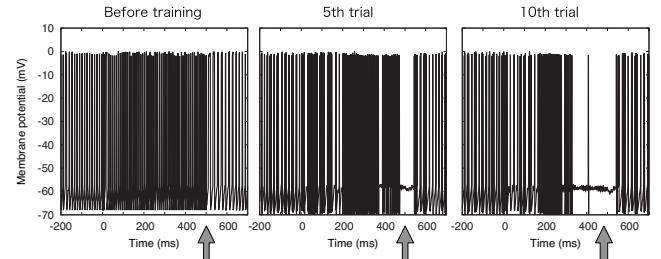


Figure 3: Purkinje cell firing during training

that a simulation for 2 s was finished within 2 s, suggesting that we were able to carry out simulations in realtime. We also confirmed that the robot was able to learn to swing a bat at a proper timing to hit a ball.

Bamford et al. have developed a VLSI model of the cerebellum and adopted it to eyeblink conditioning using rats (Bamford 2012). The present results suggest that our cerebellar model on a GPU can be used for the same purposes towards neuroprostheses.

## References

- Bamford, S. e. a. 2012. A vlsi field-programmable mixed-signal array to perform neural signal processing and neural modeling in a prosthetic system. *IEEE Trans Neural Syst Rehabil Eng* 20:455–467.
- Christian, K. M., and Thompson, R. F. 2003. Neural substrates of eyeblink conditioning: Acquisition and retention. *Learn Mem* 11:427–455.
- Igarashi, J.; Shouno, O.; Fukai, T.; and Tsujino, H. 2011. Real-time simulation of a spiking neural network model of the basal ganglia circuitry using general purpose computing on graphics processing units. *Neural Netw* 24(9):950–960.
- Ito, M. 2011. *The Cerebellum: Brain for an Implicit Self*. FT Press.
- NVIDIA. 2011. *CUDA Programming Guide 4.0*.
- Yamazaki, T., and Nagao, S. 2012. A computational mechanism for unified gain and timing control in the cerebellum. *PLOS ONE* 7(3):e33319.
- Yamazaki, T., and Tanaka, S. 2007. The cerebellum as a liquid state machine. *Neural Netw* 20:290–297.