

Open Brain Simulator Estimating Internal State of Human through External Observation towards Human Biomechatronics

Mihoko Otake, Toshihisa Takagi and Hajime Asama

Abstract—This paper presents open brain simulator, which estimates the neural state of human through external measurement for the purpose of improving motor and social skills. Macroscopic anatomical nervous systems model was built which can be connected to the musculoskeletal model. Microscopic anatomical and physiological neural models were interfaced to the macroscopic model. Neural activities of somatosensory area and Purkinje cell were calculated from motion capture data. The simulator provides technical infrastructure for human biomechatronics, which is promising for the novel diagnosis of neurological disorders and their treatments through medication and movement therapy, and for motor learning support system supporting acquisition of motor skill considering neural mechanism.

I. INTRODUCTION

Mechatronics is an applied interdisciplinary science involving mechanical engineering, electronic engineering, and software engineering. There are two technical directions in biomechatronics. One is mechatronics which supports biological systems. Another is mechatronics which learns from biological systems. The former is realized by replacement of biological function such as artificial hearts and prosthetic hands, or by augmentation and rehabilitation of existing biological function such as brain-machine interface and deep brain stimulation. The latter is also called biomimetic or biologically-inspired robots such as humanoid and snake robots. Both biomechatronic systems are not necessary the same as the biological systems but their essences of biological functions are implemented by mechatronics.

Modeling and simulation is important in order to design and control mechatronic systems. In particular, in-depth understanding and realistic modeling of biological systems is indispensable for biomechatronics. Biomechanical simulation is studied focusing on cardiovascular systems and musculoskeletal systems[1], [2] in the field of mechanical engineering and physiology, while neural network both artificial and biological ones [3], [4] are studied actively in the field of electrical and software engineering and of course in neuroscience.

Purpose of this study is to develop neural simulator that can calculate both microscopic and macroscopic states of the nervous system, since there is an urgent need for the technological development which is applicable to diagnosis, treatment and prevention of nervous diseases which cause

M. Otake and H. Asama are with Research into Artifacts, Center for Engineering The University of Tokyo, Kashiwa-no-ha 5-1-5, Kashiwa-shi, 277-8568, Japan otake@race.u-tokyo.ac.jp

T. Takagi is with the Department of Frontier Science, The University of Tokyo, Kashiwa-no-ha 5-1-5, Kashiwa-shi, 277-8568, Japan

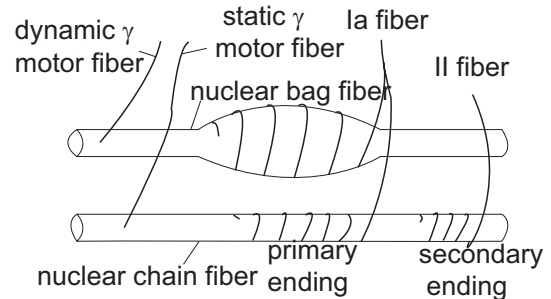


Fig. 1. Two kinds of muscle spindles: a nuclear bag fiber which detects the muscular length (top), a nuclear chain fiber which transduces the stretch speed (bottom)

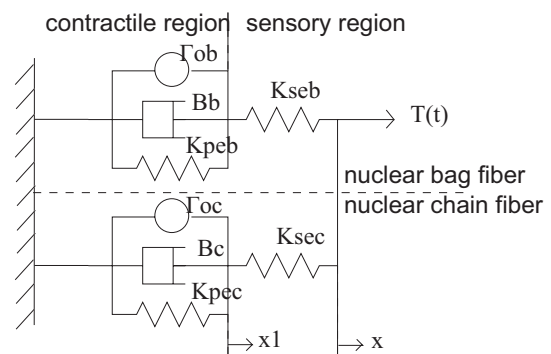


Fig. 2. The mass-spring model of muscle spindles with contractile region which regulate dynamic range, and sensing region which transduces the muscular length

movement disorders because of the rapid aging of this century. The strategy for building neural simulator is to combine top down approach and bottom up approach. This paper is organized according to these approaches.

II. TOP DOWN APPROACH: FROM MOTION CAPTURE DATA TO MACROSCALE NERVOUS SYSTEMS MODEL

Top down approach includes building macroscopic anatomical nervous systems model which can be connected to the musculoskeletal model. We can acquire neural information by analyzing whole-body muscle data, because the nervous system controls the musculoskeletal system and lengths and forces information is fed back to the nervous system. Motor information is calculated by mapping motion-capture data on to a musculoskeletal human model. Neural information represents the set of motor information on the muscles innervated by the arbitrary segment[5].

A. Computation of Muscular Information from Motion Capture Data

Human motion can be analyzed through combining motion capture system and musculoskeletal model. The body motion is mapped onto the musculoskeletal model so that the lengths and the forces of muscles are calculated[6], [7], [8], [9]. Dynamic simulation of musculoskeletal model has been studied, and it has been investigated that dynamic optimization to determine a set of muscle excitations that generate a simulation that best reproduces experimental data [10], [11]. Hundreds of time-series data of muscle lengths and forces are obtained utilizing whole body musculoskeletal model[12].

B. Computation of Input Data to Nervous Systems from Muscular Information

Muscle spindles sense the length and stretch speed of muscles and convert the set of information of input signals to the peripheral nervous systems. Motor information is transmitted to spinal cord through afferent nerve fibers called Ia or II. Muscle spindles consist of nuclear bag fibers and nuclear chain fibers. The former ones detect the stretch speed and the latter ones detect the length of the muscles (Fig. 1). The secondary endings (II) which detects length of the muscles are connected to only nuclear chain fibers, while the primary endings (Ia) which detect length and stretch speed of the muscles are connected to both nuclear bag fibers and nuclear chain fibers. The model which approximates muscle spindles as mass-spring system is shown in Fig. 2. The top one corresponds to nuclear bag fiber, whereas the bottom one corresponds to nuclear chain fiber. Both have contractile and noncontractile sites. It is known that the firing rates are proportional to the stretch length of noncontractile sites. Therefore, firing rates of the primary endings (Ia) are sum of the firing rates of nuclear bag fibers and nuclear chain fibers, and firing rates of the secondary endings (II) are sum of the firing rates of nuclear chain fibers. The equations for calculating firing rates are as follows:

$$\frac{B_b}{K_{seb}} \dot{T}_b + \left(1 + \frac{K_{peb}}{K_{seb}}\right) T_b = B_b \dot{x} + K_{peb} x + \Gamma_{0b}, \quad (1)$$

$$\nu_b = \frac{A_b}{K_{seb}} T_b, \quad (2)$$

where K_{seb} and K_{peb} are elastic modulus of sensing and contractile sites, B_b is viscous modulus, x is stretch length, Γ_{0b} is force generated by muscle spindle, A_b is proportional constant, and ν_b is firing rate. These are of nuclear bag fibers. By solving equation (1), we can obtain T_b from x . We can calculate firing rates of nuclear bag fibers ν_b from equation (2) substituting T_b . We can also obtain firing rates of nuclear chain fibers ν_c in the same manner. The firing rates of primary (Ia) and secondary (II) endings are:

$$\nu_1 = \nu_b + \nu_c, \quad (3)$$

$$\nu_2 = \nu_c. \quad (4)$$

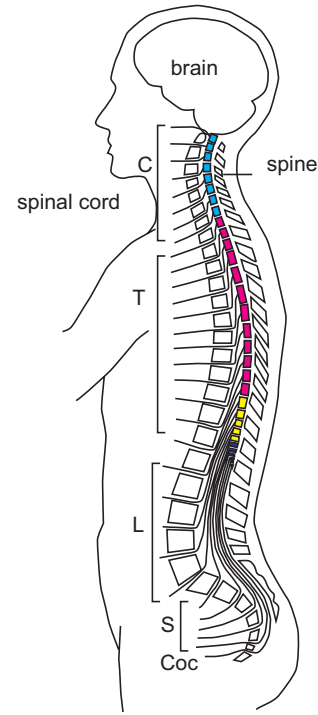


Fig. 3. Central nervous system of humans consisting of brain and spinal cord. The spinal cord is divided into 31 segments that correspond to attachments of groups of nerve roots. The skeletal muscles are classified by the innervated nerves.

We implemented the above muscle spindle model, so that we can calculate input signals to the peripheral nervous systems from the length and stretch speed of muscles. We should note that the sensitivities of the real muscle spindles are regulated by efferent nerve fibers.

C. Searching Destinations of Neural Signals using Macroscale Nervous Systems Model

Anatomy and physiology of the human spinal nervous system are summarized below[13], [14].

- The spinal cord is divided into 31 segments that correspond to attachments of groups of nerve roots (Fig. 3). The skeletal muscles are classified by the innervated nerves.
- Motor commands are sent from the ventral horn of the spinal cord. Feedback signals from the muscle spindles and the Golgi-tendon organs are received by the dorsal horn of the spinal cord. The afferent nerves are connected directly or indirectly to the efferent nerves for the stretch and inverse stretch reflexes.
- Somatotopic organization is observed at ventral horn.
- Somatotopic organization is also observed at somatosensory area.

The destination of neural signals through constructing anatomical model of the nervous systems.

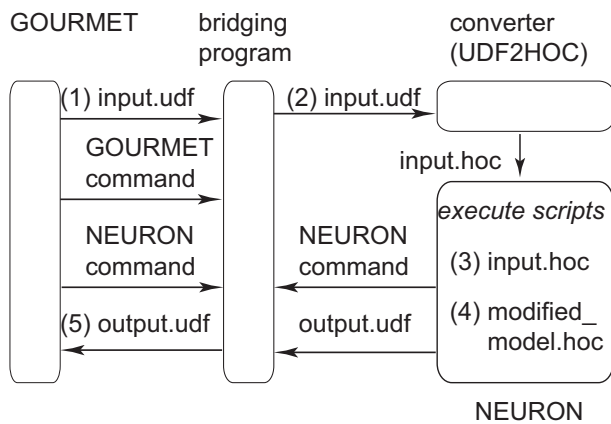


Fig. 4. System structure of the integrated simulation environment

III. BOTTOM UP APPROACH: FROM MACROSCALE NERVOUS SYSTEMS MODEL TO MICROSCALE NERVE CELL MODEL

Bottom up approach includes interfacing microscopic anatomical and physiological neural models to the macroscopic model. Fortunately, a variety of microscopic neural models are registered on neural model databases, which are created and reviewed by neuroscientists. We should be able to develop a simulator which is composed of reliable partial models, if we take advantage of them. However, reusing these models requires much time and labor, because the variables and parameters are not annotated. Also, each model doesn't consider connectivity to other models since the models are originally developed for particular experiments. We proposed a novel method for reassembly and interfacing models registered on biological model databases in order to integrate these models. The method was applied to the neural models registered on one of the typical biological model database, ModelDB[15].

In this section, we describe method for reassembly and interfacing neural models registered on the model database[16]. It is an extremely important step toward understanding the integrated function of the nervous system. The basic concept of the method is listed below.

- 1) Input and output variables and parameters should be identified systematically with minimal efforts.
- 2) Modification of the model should be kept minimal.
- 3) Simulation environments are not reconfigured.

The first strategy comes from the fact that there are over a hundred models on the database, which makes a long time to read through all files constituting the models. Therefore, we propose efficient extraction rule for identifying input/output (I/O) variables and parameters. The second one considers that the models are already reviewed and guaranteed the standalone operation. This advantage should be reserved. We add some small script to the model files for I/O. The third is based on the fact that the simulation environments are always updated. If we modify the simulators themselves, we have to rewrite them when the newer versions are released. We develop interface programs without changing the simulator.

The proposed method contains following three phases which correspond to these strategies.

A. Identification of input and output of the models

Extracting the plot commands from the model files: The models are constituted by a sequence of commands in the scripting language that creates the model itself and the GUI for a particular simulation. They contain vast numbers of variables and parameters. Some are only for internal use; others are used for plotting the graph. Each has its own range of value. If the model is used in not assumed conditions, the obtained results may be nonexistent. Basically, the GUI of the models is disturbance for integration. Conversely, if we draw attention to the variables and parameters which are plotted on the graphs, we can see that they are candidates for input and output of the models. We can also find some clues for their ranges. Therefore, the first step for identifying the input and output is to extract graph plotting commands from the model files. Retrieval terms for NEURON simulator [3] are *.addvar* and *.plot*. The variables which are plotted on the graph are candidate for output. The parameters which increment in the repetitive plotting process are candidates for input.

Extracting the units and meaning from the graphs in the papers: The simulation results of the registered models are demonstrated in the peer reviewed papers, which are listed on ModelDB[15]. The corresponding graphs can be found from these papers. The units and meaning of the variables and parameters are illustrated there. They are used for annotation. This step is not automatic but manually carried out.

B. Reassembly of the Models

Definition of Input and Output in UDF format: Now that the input and output variables and parameters are identified, they should be stored and recorded in a easily readable and understandable manner. In this study, the user definable format (UDF) is employed for data representation of the input and output, because it provides quite simple representation of experimental data[17] where overhead is kept minimum compared to XML. The format was originally developed for integrated simulator platform named GOURMET of OCTA project[18] for simulation of soft materials[17]. GOURMET has not been used other than the simulation of soft materials, although it is potentially applicable to any kinds of simulation. UDF is a plain text file, and consists of definition part and data part. In the definition part, you can define the data structure of your engine in a simple and natural way. Data are arranged in the data part according to the structure. The specific data will be explained in detail in the experiment section.

Conversion of Input and Output: The simulators cannot deal with UDF format. In order to get over the input, the program named UDF2HOC was implemented which converts UDF format into simulation script format. In the case of NEURON, the simulation scripting language is HOC. The program was executed before loading the models in the neural simulator. Then, a few lines were added in the model

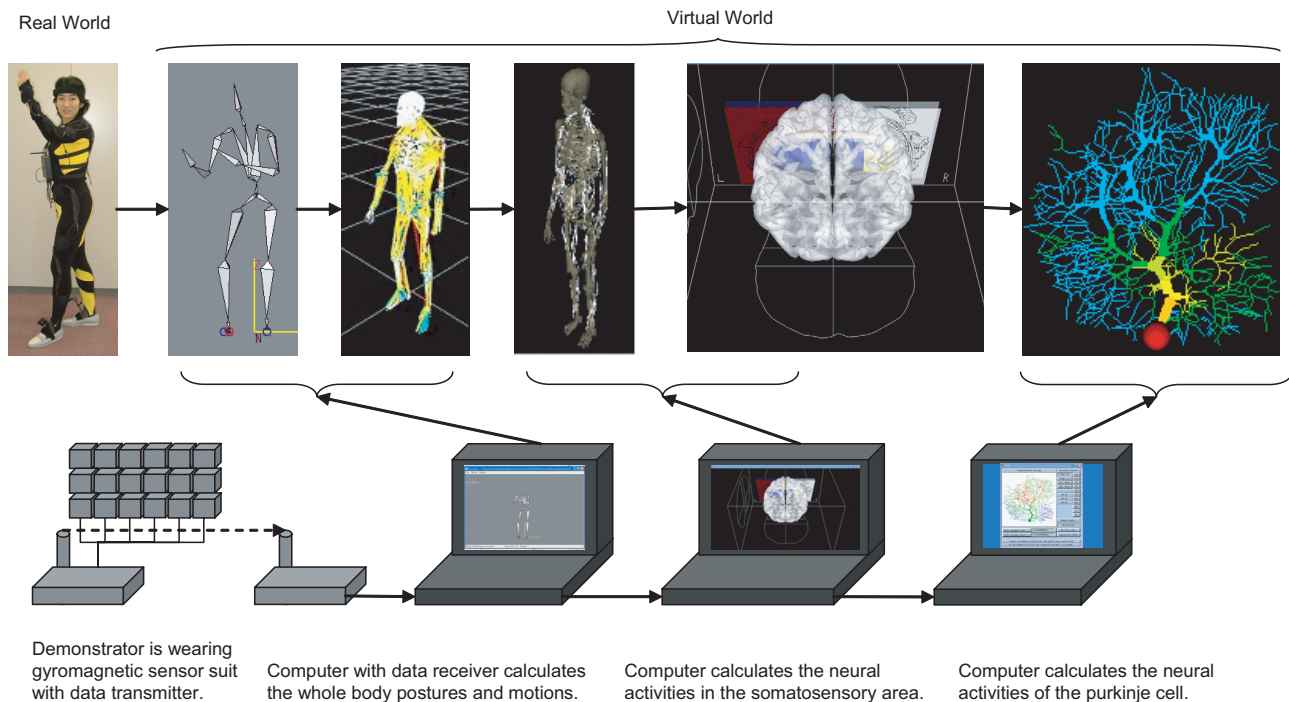


Fig. 5. Open Brain Simulator System including gyromagnetic sensor suit with data transmitter, computer with data receiver which calculates the whole body postures and motions, and computers which calculate the neural activities in the somatosensory area and Purkinje cell

script files in order to output the simulation results into UDF format.

C. Interfacing the Models

The integrated simulation platform was considered for connecting different kinds of models which run on different simulation environments. In this study, GOURMET was selected as simulation platform on which neural simulation programs run, because it provides a common GUI. GOURMET is not only a place for various simulation programs to meet and exchange the information they have, but also an editor of the input data, a viewer of the output data, a tool to make graphs and animations. The users can easily verify the simulation results because of these characteristics.

Programs which are connected to GOURMET should be able to send and receive UDF files and connected simulator commands, to receive GOURMET commands. Thus, we developed bridging program with these functions, which connects GOURMET and typical neural simulator, NEURON. In this way, we can interface the neural simulators without modifying their source codes. The operation of the bridging program is described below, which illustrated in Fig. 4.

- 1) Input data in UDF format (input.udf) is sent from GOURMET to the bridging program.
- 2) The bridging program submits the input data to the converter (UDF2HOC) and executes it. The converter generates the HOC format file (input.hoc) from input data, which can be interpreted in the NEURON simulation environment.
- 3) NEURON simulator executes the generated hoc file.
- 4) Then, NEURON runs the modified neural model so that the model generates output in UDF format.

- 5) The result (output.udf) is sent from NEURON to GOURMET through bridging program.

D. Interfacing Reassembled Microscale Nerve Cell Model to Macroscale Nervous Systems Model

Now, we have macroscopic anatomical nervous system model and reassembled microscopic neuron model. Here we reassembled the cerebellum Purkinje cell [19], [20] which works in the GENESIS simulator [4], which is another typical neural simulator as well as NEURON simulator. Then, we interfaced the Purkinje cell model to macroscopic nervous systems model.

IV. SYSTEM SETUP OF THE OPEN BRAIN SIMULATOR AND ITS SIMULATION RESULTS

Open Brain Simulator System includes gyromagnetic sensor suit with data transmitter, computer with data receiver which calculates the whole body postures and motions, and computers which calculate the neural activities in the somatosensory area and Purkinje cell (Fig. 5). Demonstrator wore gyromagnetic sensor suit with data transmitter. Computer with data receiver and gyromagnetic sensor suit with data transmitter communicated via wireless network. The following three computational modules were placed in distributed computers and were connected in order to speed up: the module calculates the whole body postures and motions, one calculates the neural activities in the somatosensory area, and one calculates the neural activities of the Purkinje cell.

Simulated voltage distribution of Purkinje cell is shown in Fig. 6 whose input neural signal to climbing fibers of the cell is calculated from motion capture data with musculoskeletal model, nervous systems model and muscle spindle model.

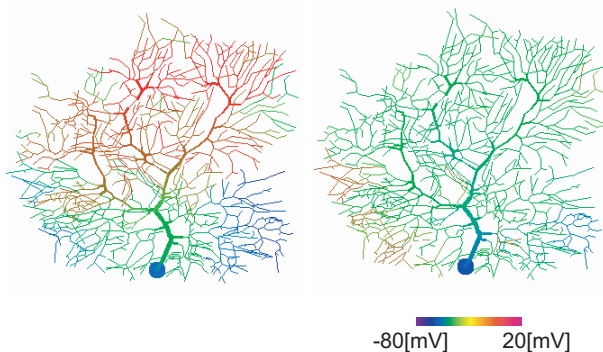


Fig. 6. Simulated voltage distribution of Purkinje cell

Lowest voltage of -80[mV] is illustrated in blue while highest voltage of 20[mV] is illustrated in red. Voltage distributions of Purkinje cell triggered by the neural signal which was directly transmitted from muscle spindle of triceps brachii muscle without modification in 4[ms] (Fig. 6 left) and 9[ms] (Fig. 6 right) after the subject started "kesagiri", a sword swinging motion are illustrated. It is known that error signals between estimated and real feedback signals are encoded in the climbing fiber input. We can consider that the simulated internal state change of Purkinje cell is the response before motor learning process, since its input is only real feedback signals from muscle spindles without modification.

V. CONCLUSION

In this study, we presented open brain simulator, which estimates the macroscopic state of somatosensory area and microscopic state of Purkinje cell of cerebellum through motion capture based on the knowledge of biomechanics, bioelectronics, and neuroscience. We integrated the computational modules which calculate from 1) whole body posture data to 2) muscular length and stretch speed, 3) responses of muscle spindles, and 4) neural activities of somatosensory areas and neurons. Musculoskeletal model, muscle spindle model, macroscopic nervous systems model, and microscopic neuron model were implemented or re-assembled and interfaced with each other so that we can calculate neural activities from motion capture data. The neuromuscular models which we have developed is only a part of the whole human brain, and obtained neural activities are theoretical. Even though the current models are limited, the simulator is scalable, which can be connected to the novel neural models which will be developed by neuroscientists in the future. The simulator provides technical infrastructure for human biomechanics, which is promising for the novel diagnosis of neurological disorders and their treatments through medication and movement therapy, and for motor learning support system supporting acquisition of motor skill considering neural mechanism.

Future work includes development of computational modules which can estimate the metabolism of the neurons and electric potential on the surface of the brain so that we can compare the simulation results to the measurement of brain activities. We will also develop mesoscopic neural models

which connect different areas of the central nervous systems in order to simulate neural activities of neurons which are far from sensory-motor systems.

ACKNOWLEDGMENTS

This work was supported by a Japan Science and Technology Agency Grant for the PRESTO program "Development of the Bilateral Multiscale Neural Simulator" (PI: Mihoko Otake) and a Grant-in-Aid for Scientific Research on priority area Systems Genomics (#014), Mobilligence (#454) and Information Explosion (#456) from the Ministry of Education, Culture, Sports, Science and Technology of Japan (MEXT).

REFERENCES

- [1] A. McCulloch, "Modeling the human cardiome in silico," *Journal of Nuclear Cardiology*, vol. 7, pp. 496–499, 2000.
- [2] E. Crampin, M. Halstead, P. Hunter, P. Nielsen, D. Noble, N. Smith, and M. Tawhai, "Computational physiology and the physiome project," *Experimental Physiology*, vol. 89, pp. 1–24, 2004.
- [3] M. L. Hines and N. T. Carnevale, "The neuron simulation environment," *Neural Computation*, vol. 9, pp. 1179–1209, 1997.
- [4] J. Bower, D. Beeman, and M. Hucka, "The genesis simulation system," in *The Handbook of Brain Theory and Neural Networks, Second edition*, M. Arbib, Ed. Cambridge, MA: MIT Press, 2003, pp. 475–478.
- [5] M. Otake and Y. Nakamura, "Spinal Information Processing and its Application to Motor Learning Support," *Journal of Robotics and Mechatronics*, vol. 17, no. 6, pp. 617–627, 2005.
- [6] S. L. Delp and P. J. Loan, "A computational framework for simulating and analyzing human and animal movement," *IEEE Computing in Science and Engineering*, vol. 2, pp. 46–55, 2000.
- [7] J. Rasmussen et al., "Anybody - a software system for ergonomic optimization," in *Fifth World Congress on Structural and Multidisciplinary Optimization*, 2003.
- [8] T. Komura and Y. Shinagawa, "Attaching physiological effects to motion-capture data," in *Proceedings of Graphics Interface*, 2001, pp. 27–36.
- [9] D. G. Lloyd and T. S. Buchanan, "Strategies of muscular support of varus and valgus isometric loads at the human knee," *Journal of Biomechanics*, vol. 34, pp. 1257–1267, 2001.
- [10] D. Davy and M. Audu, "A dynamic optimization technique for predicting muscle forces in the swing phase of gait," *Journal of Biomechanics*, vol. 20, pp. 187–201, 1987.
- [11] M. Kaplan and J. Heegaard, "Predictive algorithms for neuromuscular control of human locomotion," *Journal of Biomechanics*, vol. 34, pp. 1077–1083, 2001.
- [12] Y. Nakamura et al., "Dynamic computation of musculo-skeletal human model based on efficient algorithm for closed kinematic chains," in *Proceedings of the 2nd International Symposium on Adaptive Motion of Animals and Machines*, 2003, pp. SaP-I–2.
- [13] J. Bossy, *Atlas of Neuroanatomy and Special Sense Organs*. W.B. Saunders, 1970.
- [14] S. G. Waxman, *Correlative Neuroanatomy, 22nd eds.* a LANGE medical book, 1995.
- [15] M. Migliore, T. M. Morse, A. P. Davison, L. Marenco, G. M. Shepherd, and M. L. Hines, "Modeldb: making models publicly accessible to support computational neuroscience," *Neuroinformatics*, vol. 1, no. 1, pp. 135–9, 2003.
- [16] M. Otake and T. Takagi, "Reassembly and Interfacing Neural Models Registered on Biological Model Databases," *Genome Informatics*, vol. 16, no. 2, pp. 76–85, 2005.
- [17] M. Doi, "Challenge in polymer physics," *Pure Appl. Chem.*, vol. 75, no. 10, pp. 1359–1615, 2003.
- [18] —, *OCTA*. <http://octa.jp/>.
- [19] E. De Schutter and J. Bower, "An active membrane model of the cerebellar Purkinje cell. I. Simulation of current clamps in slice," *Journal of Neurophysiology*, vol. 71, pp. 375–400, 1994.
- [20] —, "An active membrane model of the cerebellar Purkinje cell. II. Simulation of synaptic responses," *Journal of Neurophysiology*, vol. 71, pp. 401–419, 1994.