AD-A237 558

DTIC
ELECTE
JUN 2 7 1991
S C D

Report BCM-NS-CNS-91-001

A Study of Neuronal Properties, Synaptic Plasticity and Network Interactions
Using a Computer Reconstituted Neuronal Network Derived from Fundamental Biophysical Principles

David C. Tam

Division of Neuroscience
Baylor College of Medicine
1 Baylor Plaza
Houston, TX 77030

(713) 798-3134

E-mail address:
dtam@next-cns.neusc.bcm.tmc.edu or
dtam@cephalo.neusc.bcm.tmc.edu

Accession For
NTIS GRA&I
DTIC TAB
Unannounced
Justification
By Per AD-A230477
Distribution/
Availability Codes
Dist | Avail and/or Special
A-1
QUALITY INSPECTED 1

June 15, 1991

Semiannual Performance (Technical) Progress Report (December, 1990 - June, 1991)

Grant number: N00014-90-J-1353

Prepared for
Office of Naval Research
Biological Intelligence Program
800 North Quincy Street
Arlington, VA 22217-5000

Progress Summary

A preliminary version of the neural simulation program *MacNeuron* is implemented on the Macintosh IIci with more extensive user-interface added to the previous version. A preliminary user manual and the software will be sent to the scientific program officer, which describes the specific run-time environment of the program.

The current version of *MacNeuron* incorporated much of the friendly user-interface for building a network of neuron. The building process can be done by (a) the menu-driven window user-interface environment; and (b) the batch-mode "script" file for describing the run-time environment and simulation parameters. An interactive built-in text editor is also included in the simulation program for entering the script-file for describing the run-time environment. The program can be run with interactive user-control (via iconic interface) or under hands-off background mode (via text-based script interface). Currently, the script file is excuted via the "macro" command, where a group of user-specified commands can be executed under a menu-command created by the user's script file.

Specific Program Progress

The major progress is made in the user-interface of the MacNeuron program. Two specific areas has been incorporated into the program: (a) the building of a network of neurons via menu-driven windowing interface; and (b) the "script" description of the commands for batch-mode operation.

(A) Window Interface for Creation of a Network of Neurons

The user-interface of the program has been redesigned to incorporate more flexible creation of a network of neuron. The construction of neurons is based on a top-down approach, where a hierarchy of brain – network – neuron – membrane – ionic-channel – chemical-concentration-compartment is followed. In other words, the description of the simulation is done in a top-down fashion. The description of the neuronal networks in this simulator is done through two processes: (1) the description of the physical parameters of neuron and membrane-patch objects, and (2) the description of the connections (or links) between the above objects.

91 6 24 110

110 91-03294

Since the description of a neuron is separated from the description of its connections, the physiological parameters of a neuron can be stored separated from the connections which describes how the neurons are connected together and how the membrane-patches are connected together to form a neuron. Thus, libraries of the physical description of a neuron can be stored in separated files, which can be retrieved easily later by different users for constructing the specific neurons and networks. Thus, the construction of a network can be done by connecting the existing library files of the neuron descriptions. The actual implementation of neuron library will be done later in the development phase. The current implementation concentrates on the development of the necessary window user-interface for describing the connectivity and physiological parameters of a neuron.

At the top level is the "brain", where a brain is composed of multiple "networks". The networks are connected together to form a brain. At the next level is the "network", where a network is composed of multiple "neurons". The neurons are connected together to form a network. At the next level is the "neuron", where a neuron is composed of multiple "membrane-patches" or "membrane-compartments". The membrane-patches are connected together for form the morphology of a neuron (including soma, dendrites and axons). At the next level is the "membrane-patch", where a membrane-patch is composed the "ionic-conductance channels". There are also "chemical-compartments" which can be separated by the membrane to form the extracellular and intracellular ionic fluid compartments. Chemical compartments can also be used to form "shells" of iso-concentration of ionic species, such as calcium.

Each of the above level of description of the neural model can be specified by the user via hierarchical windows. The brain is created by a mouse-button press to the ionic button in the window to instantiate a new brain-object. Double-clicking on the brain-object (i.e., opening the brain-object) will list all its connectivities and attributes. Similarly, a network can be instantiated and connected hierarchically within the iconic window. The connectivity can be established by "dragging" the object-items into the appropriate window. Thus, the user-interface is intuitive enough to provide a user-friendly environment for creation of the simulation network.

(B) Batch-mode Script Language for Specifying the Description of neurons and Run-time Environment

The simulator desription script language is also implemented in the current version. The script language can be used to create user-defined "macros" commands. These macro commands will be automatically installed as menu-items appended to the existing menus. Thus, the user can easily custom create his/her own commands. Thus, a single command can be used to describe and execute a series of simulation descriptions and actions.

The script language is based on the programming language syntax of Pascal. The use of Pascal-like language facilitates users' familarity with a well-defined language for control and execution. The familar "repeat-loop" and "if-the-else" statements can be used to control the flow of the simulation in batch-mode. Thus, the simulation can be executed with a series of parameter sets using a simple program-loop structure.

The neuron-objects can also be created like any data in a programming language. The user can use variables to assign to the neuron-objects, so that multiple neurons can be created using a simple assignment statement to multiple variables. For instance, if the user wants to create 1000 neurons of the same type, all it needs to be done is to use a "for-loop", i.e., *for* neuron_number:= 1 *to* 1000 *do* *new*(neuron). Similarly, multiple connections between neurons can be specified using a simple "for-loop" structure.

One of the goals of the project is to provide a neural simulator that simplifies the complicated structure of the interconnectivities in describing networks of neurons with various physiological parameters. In pursue of this goal, the neural simulator is designed with the hierarchical structure for description of the network greatly reduces the complexity of the complicated structures in desribing the interconnects and morphology as well as physiology of neurons.

Publications by the Principal Investigator during 1990 and 1991

Tam, D. C. (1991) Signal processing by multiplexing and demultiplexing in neurons. In: *Advances in Neural Information Processing Systems.* (D. S. Touretzky, ed.), Morgan Kaufmann Publishers, San Mateo, California. (*in press*)

Tam, D. C. (1991) A hybrid time-shifted neural network for analyzing biological neuronal spike trains. *Progress in Neural Networks* (O. Omidvar, ed.)Vol. 2, Ablex Publishing Corporation: Norwood, New Jersey. (*in press*)

Alkon, D. L., Vogl, T. P, Blackwell, K. T. and Tam, D. C. (1991) Memory function in neural and artificial networks. In: *Neural Network Models of Conditioning and Action.* (M. L. Commons, S. Grossberg, J. E. R. Staddon, eds.) pp. 1-11. Lawrence Erlbaum Associates: Hillsdale, New Jersey.

Tam, D. C. (1990) Decoding of firing intervals in a temporal-coded spike train using a topographically mapped neural network. *Proceedings of the International Joint Conference on Neural Networks, June, 1990.* Vol. 3, pp. III-627–632.

Tam, D. C. (1990) Temporal-spatial coding transformation: Conversion of frequency-code to place-code via a time-delayed neural network. *Proceedings of the International Joint Conference on Neural Networks* (H. Caudill, eds.), *Jan., 1990.* Vol. 1, pp. I-130–133.

Abstracts

Tam, D. C. (1990) Functional significance of bi-threshold firing of neurons. *Society for Neuroscience Abstract.* Vol. 16, p. 1091.

Tam, D. C. (1990) Hebbian synapse and its relation to cross-correlation function in associative conditioning learning. *Eighth Annual Conference on Biomedical Engineering Research in Houston.* p. 5.

Tam, D. C. (1990) Visual-motor integration: What role does cerebellar cortical neurons participate in movement control in monkeys? *Eighth Annual Conference on Biomedical Engineering Research in Houston.* p. 39.