MODELLING THE SOMATIC ELECTRICAL RESPONSE OF HIPPOCAMPAL PYRAMIDAL NEURONS

by

Lyle J Borg-Graham
B.S.E.E., University of California at Berkeley

Submitted to the

DEPARTMENT OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCE

in partial fulfillment of the requirements
for the degree of

MASTER OF SCIENCE

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

May, 1987

Signature of Author _____

Department of Electrical Engineering and Computer Science. May 18. 1987

Certified by _____

Tomaso Poggio
Thesis Supervisor

Accepted by _____

Arthur C. Smith, Chairman
Committee on Graduate Students

MODELLING THE SOMATIC ELECTRICAL RESPONSE OF HIPPOCAMPAL PYRAMIDAL NEURONS

by

Lyle J Borg-Graham

Submitted to the Department of Electrical Engineering and Computer Science on May 18, 1987 in partial fulfillment of the requirements for the Degree of Master of Science in Electrical Engineering.

ABSTRACT

A modelling study of hippocampal pyramidal neurons is described. This study is based on simulations using HIPPO, a program which simulates the somatic electrical activity of these cells. HIPPO is based on a) descriptions of eleven non-linear conductances that have been either reported for this class of cell in the literature or postulated in the present study, and b) an approximation of the electrotonic structure of the cell that is derived in this thesis, based on data for the linear properties of these cells.

HIPPO is used a) to integrate empirical data from a variety of sources on the electrical characteristics of this type of cell, b) to investigate the functional significance of the various elements that underly the electrical behavior, and c) to provide a tool for the electrophysiologist to supplement direct observation of these cells and provide a method of testing speculations regarding parameters that are not accessible.

The novel results of this thesis include:

- Simulation of a wide range of electrical behavior of hippocampal pyramidal cells by using descriptions of ionic conductances (channels) whose kinetic properties are developed from both limited voltage-clamp and current-clamp data and from the theory of single-barrier gating mechanisms. This result suggests that the single-barrier gating mechanism of the Hodgkin-Huxley model for ionic channels is empirically valid for a wide variety of currents in excitable cells.

- An estimation of the linear parameters of hippocampal pyramidal cells that suggest that the membrane resistivity, and thus the membrane time constant, is non-homogeneous.

- An estimation of dendritic membrane resistivity (R_m) and cytoplasmic resistivity (R_i) that is higher than generally considered, and the conclusion that the cell is more electrically compact than previously thought. This

compactness implies that distal and proximal dendritic input have similar efficacies in generating a somatic response.

- A method for estimating the dimensions of the equivalent cable approximation to the dendritic tree based solely on histological data.
- Descriptions of three putative Na^+ currents ($I_{Na-trig}$, I_{Na-rep}, and $I_{Na-tail}$) that quantitatively reproduce the behavior generally ascribed to Na^+ currents in hippocampal pyramidal cells.
- Descriptions of two Ca^{2+} currents (I_{Ca} and I_{CaS}) and a system for regulating Ca^{2+} inside the cell that qualitatively reproduces the data for Ca^{2+}-only behavior in hippocampal pyramidal cells.
- Descriptions of six K^+ currents (a delayed rectifier K^+ current – I_{DR}, a transient K^+ current – I_A, a Ca^{2+}-mediated K^+ current – I_C, a Ca^{2+}-mediated slow K^+ current – I_{AHP}, a muscarinic K^+ current – I_M, and an anomalous rectifier K^+ current – I_Q) that are consistent with the available data on these currents and that reproduce either quantitatively or qualitatively the behavior associated to each current during the electrical response of hippocampal pyramidal cells.
- Simulations demonstrating possible computational and/or pathologic roles for the model currents.
- The design of an interactive program that simulates hippocampal pyramidal cells with a variety of models of electrotonic structure and the inclusion of Hodgkin-Huxley-like non-linear conductances at various points in the cell.

Thesis Supervisor: Prof. Tomaso Poggio, Professor of Brain and Cognitive Sciences

Acknowledgments

I would like to thank the following people for their contribution to this Thesis:

Prof. Tomaso Poggio for his continued support and thought-provoking interjections on how this work may have relevance for understanding the computational mechanisms of the brain.

Prof. Paul Adams and Dr. Johan Storm for their invaluable collaboration in both the acquisition of new data that formed the foundation of the model, for their assistance in the critical reading of the literature that was referenced to build the model, for their critique of the putative features of the work, and for their constant and enthusiastic feedback with the result that this project reflected some symbiosis between experiment and theory.

Prof. Christof Koch for providing the initial impetus towards my studying the problem presented here, and for steering my work early on so as to avoid some of the pitfalls that await the unwary neural modeller.

Prof. William Peake and Prof. Thomas Weiss for their detailed critique of this work and ideas for pursuing issues raised by the methodology.

Dr. Norberto Grzywacz for his exhaustive reading of the work and many suggestions for analyzing the computational roles of the neuron.

Christopher Graham, L. L. D. for his insightful reading and detailed critique of a draft of this Thesis, as well as moral support.

Dr. Phillipe Brou and Steve Gander for their patient consulting as I learned the fundamentals of the Symbolics Lisp environment.

Patrick O'Donnell for his plotting software and readiness to assist with advice about the Symbolics machines.

Davi Geiger and Anselm Spoerri for their good humour in the desert.

And, especially, Josette for herself.

Contents

List of Figures

List of Tables

Chapter 1

INTRODUCTION

1.1 Modelling Neurons of the Central Nervous System

Understanding the brain is a multi-level task, incorporating perspectives from molecular biology to cognitive science and psychology. At some point in this hierarchy the single cell is encountered, and the view that all information processing in the brain derives from mechanisms on this level is generally accepted; i.e. it is correct to speak of a neuron *processing signals*, rather than the neuropil being the basic functional unit for computation.

The actual role of individual neurons in information processing is open to speculation. In some systems good arguments have been advanced in support of the handling of certain tasks by specific cells. In most structures in the central nervous system (CNS), however, the role of the single cell is not well defined. Typically, descriptions of information processing in the CNS refer to anatomical structures consisting of (at least) thousands of cells, and fail to assign roles to single cells.

Thus, an investigation into information processing on the level of the single neuron is important. Over the past decade quantitative data on CNS neurons has grown considerably, and interpretation of this data is now appropriate in order to establish the role of the neuron as it receives the multitudinous signals from the neural mesh. Utilization of systematic models is a method of addressing this problem. One of the models that is an appropriate vehicle for this task is named HIPPO

1.2 The HIPPO Model of Hippocampal Pyramidal Cells

This thesis describes the development and application of the computer model, HIPPO. This model simulates the somatic electrical behavior of a stereotypical cortical integrating neuron, the mammalian hippocampal pyramidal cell (HPC). The development of HIPPO includes an estimation of the electrical structure for this cell, development of the numerical technique used in the model algorithm, integration of electrophysiological data into the model (particularly that describing the non-linear conductances reported for the HPC), and implementation of the model on a Symbolics 3600 LISP machine. The application of HIPPO includes an integration of sparse and conflicting data obtained from a variety of electrophysiological protocols. Applying HIPPO includes also testing of speculations regarding characteristics not accessible to *in vivo* or *in vitro* measurement.

As set forth this report, modelling a non-linear system as complex as the hippocampal pyramidal cell is problematic at best. The situation is complicated by both the numerous interdependencies of the mechanisms underlying electrical behavior in these neurons[1], and by the approximations and assumptions (e.g. the Hodgkin-Huxley model, ref. Chapter 4) that are required due to the present state of the data.

In light of these difficulties, this model is presented with the understanding that many of the putative mechanisms described could easily be incorrect in their details, but given the constraints imposed on the development of the model parameters (as defined throughout this Thesis), these descriptions are reasonable in that they are based on first principles and that they generate the desired behavior. At best, the descriptions will in some way reflect what is actually going on in these cells; at worst, the descriptions and the resulting behavior of the model will generate testable predictions and suggestions for postulating more accurate mechanisms.

[1]In fact, these interdependencies provide important and implicit constraints on the derivation of parameters, which in turn causes the selection of parameters to be less arbitrary than otherwise would be the case. These constraints are manifest in the cross-checking of overall model behavior, required whenever a subset of the model parameters is altered. This point will be reiterated several times in later chapters when strategies for developing various elements are reviewed.

1.3 Organization of Thesis

In this chapter Sections 1.4 through 1.7 will introduce the hippocampal pyramidal neuron and describe the motivations for modelling this cell. Some comments on the applied aspects of the program are also presented.

Chapter 2 contains a discussion of the strategy used herein in developing HIPPO and the basic structure of the model, outlining the geometry of the model and the type of circuit that it simulates. The development of HIPPO involves careful examination of the literature on hippocampal cells (and other neurons, as required) in conjunction with consultation with electrophysiologists. The techniques used by the electrophysiologist to measure the various components of the electrical behavior of a neuron are reviewed since these techniques guide the construction of the model from available data and the evaluation of any inconsistencies in that data. This chapter closes with a brief discussion of the network elements, in particular the electrochemical potentials that drive the electrical excitability of these neurons.

Chapter 3 covers the evaluation of the linear characteristics of the HPC. This analysis forms a basis for building the model of the pyramidal neuron, particularly since many of the non-linear parameters must be estimated from incomplete data. Estimating the characteristics of non-linearities in the cell is fruitless without a solid linear description to build on. Several approaches to this problem , as well as a critical review of the published data on the linear structure of the HPC, are presented. Finally, the linear parameters used for the present version of HIPPO are discussed.

The non-linear conductances in the model are all based on some variation of the classic Hodgkin and Huxley model ([21], [20], [22], [23]) of the Na^+ and K^+ conductances in the squid axon. This approach represents a major assumption in the HIPPO model, particularly since many of the non-linear conductances in HPC have not been conclusively demonstrated as being Hodgkin-Huxley-like conductances. However, in light of the paucity of data for these cells, this approach is a reasonable one, and in fact has been successful in reproducing many qualitative and quantitative aspects of HPC electrical behavior. Since the Hodgkin-Huxley model is of such basic importance to the HIPPO description, this model and the application of this model to putative HPC conductances are described in Chapter 4. In addition, the implications of the single-barrier gating interpretation of the Hodgkin-Huxley model are discussed in detail.

In the next three chapters the development of descriptions of the various non-linear currents is described, along with the simulated behavior of these

currents. In these chapters the behavior of the model is compared typically with data from cells obtained under conditions similar to those being simulated.

In Chapter 5, the three proposed Na^+ currents, $I_{Na-trig}$, I_{Na-rep}, and $I_{Na-tail}$, are presented and the HIPPO simulation of Na^+-only HPC behavior is shown.

In Chapter 6, the HIPPO description of the two Ca^{2+} currents, I_{Ca} and I_{CaS}, are presented with simulations of Ca^{2+}-only HPC behavior, as well as the HIPPO description of the dynamics of intracellular Ca^{2+} and the factors that determine the concentration of Ca^{2+} underneath the cell membrane. This latter component is important since two K^+ currents (I_C and I_{AHP}) are presumably mediated by the concentration of intracellular Ca^{2+}, and the magnitude of $[Ca^{2+}]_{shell.1}$ (ref. Chapter 6) can significantly change the reversal potential for Ca^{2+} (E_{Ca}).

In Chapter 7 the six K^+ currents in the model are presented. These currents, I_{DR}, I_A, I_C, I_{AHP}, I_M, and I_Q, display a wide range of activation/inactivation characteristics and thus modulate the HPC response in many different ways. The parameters used in the model for these currents are presented here, as well as various simulations demonstrating their behavior.

In Chapter 8 and Chapter 9 some selected simulations are presented of voltage clamp protocols and current clamp protocols, respectively. These simulations augment the ones that are presented in earlier chapters, and demonstrate the overall behavior of the model and how the model reproduces various data taken from cells. In contrast to the results presented in earlier chapters, the simulations discussed here represent speculative behavior of the HPC, given the HIPPO description of its electrical characteristics.

In Chapter 10 implications of the results obtained by the model are discussed, and the validity of both these results and the approach used in constructing HIPPO. Guidelines are also proposed regarding the application of HIPPO. In the final chapter, Chapter 11, some of the future applications of HIPPO are presented.

In Appendix A a sample simulation session is described, showing the interactive nature of the menu-driven HIPPO and the presentation of simulation results. Appendix B contains a description of the predictor-corrector algorithm used by HIPPO to solve the network equations. In Appendix C the structure of the HIPPO code will be described. Appendix D contains the software listing for HIPPO.

1.4 Hippocampal Pyramidal Neurons As A Stereotypical Cortical Integrating Neuron

The hippocampus is a part of the cerebral cortex. This structure carries and (presumably) processes signals projecting to and leaving from various regions of the neocortex. The hippocampus forms along the free medial edge of the temporal lobe of each cerebral hemisphere, extending from the several layers of neocortex, forming its characteristic spiral, which in turn consists of a single layer of pyramidal cells. Historically, the striking anatomy and connectivity of the hippocampus has made it one of the more studied areas of cortex. Although the classical role of the hippocampus as a major player in the so-called "limbic system" is now being re-evaluated, there is substantial evidence of various functional roles of this structure, including a putative role in mediating long-term memory.

The pyramidal neuron is the basic efferent cell of the cerebral cortex, integrating afferents from both intracortical and extracortical structures. The connectivity of a single pyramidal cell is typically very large, with hundreds to thousands of afferent connections. This input tends to be quite segregated, with distinct tracts originating from various structures making synapses with specific regions of the pyramidal cell's extensive dendritic tree. The pyramidal cell, as one of the major cell types in the cortex, is an important determinant of cortical function on the cellular level. The hippocampal pyramidal cell is representative of this class of neurons, and for these reasons and those set forth below, it is a cell of choice for investigations of central neuron characteristics.

The large body of knowledge for the hippocampus has been enhanced in recent years by the brain slice technique used for obtaining stable *in vitro* electrophysiological measurements with various micro-electrode techniques. In the slice technique, approximately 500 μm thick transverse slices of freshly excised hippocampus (typically rat or guinea pig) are maintained for several hours in small chambers filled with an appropriate oxygenated solution. Once set up in this manner, intracellular recordings from microelectrodes can be obtained for several hours. A related technique, which also has been developed recently, is the combination of patch clamp recording methods with pyramidal cells cultured from embryonic neurons. This technique, while clearly moving one more step away from the physiological environment, allows for higher quality measurements due to the improved electrical nad mechanical characteristics of the patch electrode over the micro-electrode.

The hippocampal pyramidal cell has therefore been chosen as the target cell for the present study. To build this model, an attempt was made to evaluate a large sample of the literature, which is quite extensive. As an initial modelling study, this effort was successful in quantifying much of the behavior of this representative cell in the CNS, and in establishing the basic aspects of somatic HPC function. These results may be extended to other cells in the CNS, especially when more data on these cells becomes available.

1.5 Application of HIPPO

An important aspect for the application of the HIPPO model as a research tool is its flexibility. The structure of HIPPO allows straightforward testing of the sensitivity of the model to changes in various parameters. In particular, estimating a parameter which is based on low-confidence experimental data can require testing of values over a wide range. One cost of this flexibility is in the execution time of a given simulation protocol. For this reason, versions of HIPPO were developed which had a relatively fixed structure and simulation protocol but executed considerably faster. In some cases the use of these quick "customized" HIPPOs was effective in developing an intuitive sense of the behavior of the model, and presumably that of the cell. For example, voltage-clamp simulations of isopotential structures involve considerably less computation than that of voltage-clamp simulations of non-isopotential structures or current-clamp simulations in general. Yet, to a first approximation, much of the data in the literature can be effectively simulated with the simplified voltage-clamp protocol. Once initial estimates of simulation parameters have been tested on the simplified HIPPO, then the more general HIPPO can be used to examine more realistic structures.

1.6 The User Interface

A substantial effort was invested in the user interface of HIPPO. Input to the model is done via a menu hierarchy (ref. Appendix A) that allows efficient manipulation of relevant parameters and a subsequently rapid set-up for a given simulation. A limited degree of automated simulation execution is also provided. Output of HIPPO is both graphical and numerical. Manipulation of the output is straightforward and non-displayed parameters are easily accessible. The user interface design has a net result of being able to use HIPPO in an interactive, self-documenting fashion.

1.7 Previous Work

Much of the program design philosophy and the approaches used in estimating model parameters were inspired by an earlier model constructed by Prof. Christof Koch and Prof. Paul Adams for the bullfrog sympathetic ganglion cell [2].

Chapter 2

MODELLING STRATEGY AND THE ELEMENTS OF HIPPO

2.1 Introduction

The goal of the HIPPO model is to give a reasonable description of a non-linear time-varying multi-variable system. To achieve this, development of the model was accomplished in stages of increasing complexity along several degrees of freedom, including the geometry of the model cell and its non-linear, time-varying properties. Since many of the network components are non-linear, superposition does not hold in general. The resulting interdependence of the parameters was a considerable problem in constructing a valid description, especially since any change in a single parameter often meant that much of the model behavior had to be checked. Careful evaluation of experimental results was essential in order to prevent generation of false-positive solutions. This chapter will discuss the general development of the model, the structure of the modelled system and its elements.

2.2 Determining the Validity of the Model Results

A key consideration in the interpretation of the HIPPO results is in determining the validity of a given version of the model. There is no clear-cut

unique solution set for the model parameters. For example, many (non-physiological) descriptions of the kinetics will yield reasonable behavior.

The basic approach considers an evaluation of as many parameters as possible under orthogonal or nearly orthogonal simulation protocols, mimicking the electrophysiologist's technique. Particular attention is paid to when experimental results reflect the overlapping of several kinetic mechanisms, particularly when superposition does not hold (when superposition does hold, then it may be exploited to extract the relevant parameters from the total response). Whenever several non-linear elements contribute to the model response the model is used iteratively to test different hypotheses for the parameters in question.

Most of the HPC currents are present over a limited range of membrane voltages. In the simplest case, involving a determination of the kinetics of a system with two currents $X1$ and $X2$, when the activation ranges for $X1$ and $X2$ are non-overlapping, then the voltage clamp protocol will have no problem quantifying each current. In practice, however, there are few currents that experience an exclusive range of activation, and therefore the situation is more complicated [1].

While more than one current may be activated at a given voltage range, different components may be distinguished if they have significantly different time courses. For example, I_A and I_{DR} are activated over the same range. Since I_A activates and inactivates much faster than I_{DR} over part of this range, however, the two currents can be distinguished by their distinct time courses in voltage clamp protocols (Segal and Barker, 1980).

Another technique to separate different currents is to exploit the pharmocological sensitivity of some currents. For example, Na^+ currents are generally believed to be blocked by the puffer fish toxin, tetradotoxin (TTX), and that channels for other ions are largely unaffected by TTX. Thus, in voltage clamp preparations TTX is commonly used to unmask currents that might otherwise be swamped by larger Na^+ currents. with similar kinetics. Other examples of selective blocking of currents include the use of tetra-ethylammonium (TEA) to block some potassium currents (e.g. I_{DR}), 4-aminopyridine (4-AP) to block I_A, and various Ca^{2+} blockers (e.g. Mn^+) or Ca^{2+} -chelators to inhibit Ca^{2+} currents , and calcium-mediated currents (I_C and I_{AHP}) (ref. Table 7.1).

[1]The main exception is I_Q, which is the only current activated at fairly hyperpolarized potentials (Chapter 7). The leak current is superimposed on the Q current, but that may be readily distinguished from the I_Q.

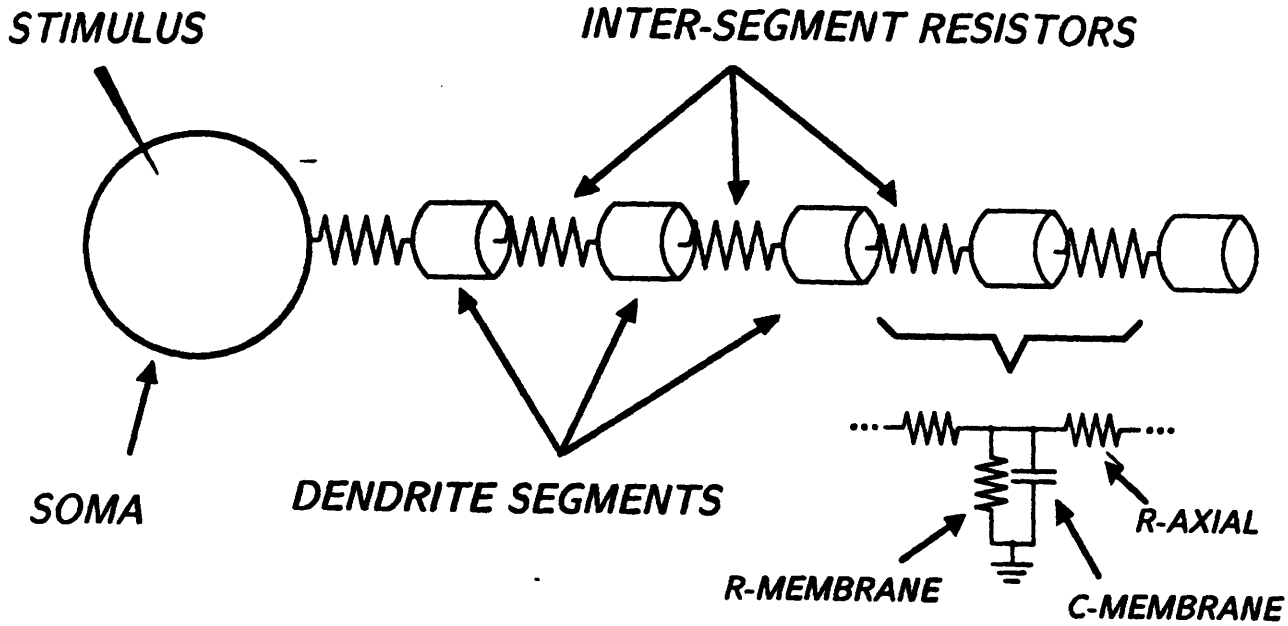


Figure 2.1: Typical geometry of HIPPO compartmental model simulation, including soma sphere joined in series to several dendritic cylindrical segments and current injection into soma. Each compartment is isopotential. One of the outputs of the simulation is the time course of each compartment voltage.

2.3 Geometry Of Model

HIPPO simulates hippocampal pyramidal neurons with a compartmental model that incorporates several isopotential compartments connected together with resistors. The simplest morphology is a single compartmental, isopotential spherical model of the entire cell, i.e. no dendritic structure or axonal process. This structure can be extended with the addition of as many as five processes, which can be configured as four dendrites and one single compartment axon. Four of the processes are represented by an arbitrary number of lumped cylindrical segments, with each segment having its own set of linear and non-linear electrical parameters. The fifth process, when included, is represented by a single isopotential cylindrical compartment. Most simulations were run with a single dendrite and no axon, as illustrated in Figure 2.1.

The physical and electrical parameters for each of the compartments -

soma sphere and process cylinders - can be set uniquely for each compartment. For example, the soma can be set up with at most eleven non-linear currents, a particular membrane resistivity and capacitance, and a particular radius. A single dendrite might be added with ten segments, with eight configured as linear cables using the same linear parameters but distinct dimensions. The remaining two segments could have two non-linear conductances in addition to their linear properties, and one of the all-linear segments could have a synapse. In the present report only the linear dendrite case will be examined. In Chapter 3 a detailed analysis and subsequent method for approximating the hippocampal pyramidal cell geometry will be presented.

2.4 HIPPO Solves A Non-Linear, Time-Varying Electrical Network

In this modelling study the HPC is analyzed as an electrical circuit. Inputs to this circuit include stimuli provided by intracellular electrodes or by synaptic-like conductance changes, and circuit outputs include voltages at various parts of the cell, specific membrane currents, the concentration of Ca^{2+} in different compartments related to the circuit, and various state variables associated with the non-linear conductances. In a general sense, HIPPO is a program for simulating a particular class of electrical networks. HIPPO is configured to handle a limited set of topologies with a specific class of network elements, as well as linear resistors and capacitors, voltage sources, and current sources. The special class of elements are non-linear voltage-dependent and time-dependent conductances that represent the behavior of ion-specific channels in the cell membrane. The electrotonic structure of neurons (as determined by the morphometrics and linear components of the cell) is extraordinarily important to their function, and much of the effort in the development of HIPPO was in the characterization of this structure as well as that of the non-linear elements.

Figure 2.2 illustrates a typical network configuration for simulation. In this particular topology the network is stimulated by a current source that injects a current pulse into the soma. This source is the system input in this particular simulation. The outputs include the voltages of each of the compartments, the currents through each of the branches of the network, and the state variables that describe the behavior of the non-linear conductances.

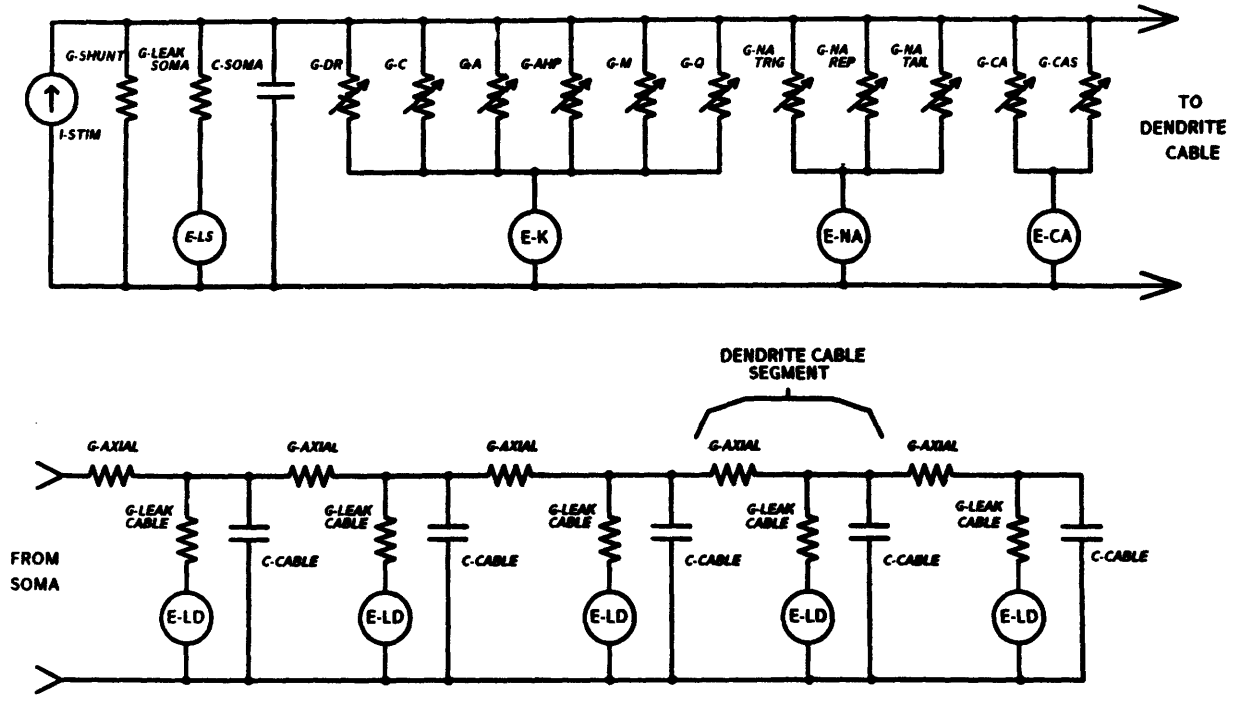


Figure 2.2: Network schematic for typical simulation protocol, similar to that shown in Figure 2.1. Voltage for each compartment is measured relative to the outside of the cell. Determining the characteristics of the circuit elements from (a) the voltage response to current source stimulus (current clamp), (b) the current response to voltage source stimulus (voltage clamp), and (c) estimation of parameters derived from basic principles is the focus of this research.

2.5 Elements of the Model

The basic task of HIPPO is the determination of the circuit elements. Characterization of some of these components is straightforward, e.g. the membrane capacitance, while most of the others are subject to considerable speculation. Because of a lack of data, some elements cannot be determined with a high degree of certainty. For these parameters, if the behavior of the cell is strongly effected, sets of simulations were conducted over the presumed range of the parameter, resulting in a range of cell responses peculiar to changes in that parameter.

HIPPO incorporates 11 non-linear, time-varying conductances in the soma, including those that underly three putative sodium currents, ($I_{Na-trig}$, $I_{Na-tail}$, and I_{Na-rep}), a delayed-rectifier potassium current (I_{DR}), a calcium current (I_{Ca}), a slow calcium current (I_{CaS}), a calcium-mediated potassium current (I_C), an after-hyperpolarization potassium current (I_{AHP}), a muscarine-inhibited potassium current (I_M), a transient potassium current (I_A), and an anomalous rectifier potassium current (I_Q).

All the compartments include the leakage current (I_L) and the capacitance current (I_{Cap}) as explicit components of the network model. In addition, the soma compartment includes a non-specific shunt conductance as may be introduced by the microelectrode.

All the parameters for the model, including the kinetics of the non-linear conductances and the linear characteristics of hippocampal pyramidal cells, were derived either from the literature or from consultation with Prof. Paul Adams [2], Dr. Johan Storm [3] and Prof. Christof Koch [4].

2.6 Reversal Potentials and Ionic Current Through Membrane Conductances

The origin of the membrane potentials will now be reviewed, as these elements are fundamental to the interpretation of the model. The reversal potentials for each conductance derive from two salient features, (1) a concentration gradient across a membrane for ionic species X and, (2) selective permeability in that conductance for X relative to any other ionic species in

[2]Department of Neurobiology and Behavior, State University of New York at Stony Brook.

[3]Ibid.

[4]Department of Biology, California Institute of Technology.

the intra- and extra-cellular medium. The concentration gradient sets up a thermodynamic potential that drives ions of X down the gradient, and across the membrane, in order to balance the concentrations. If, however, the mobility across the membrane of any of the other species in the mediums is not the same as X, movement of X will then set up a charge imbalance across the membrane. This imbalance will create a potential difference across the membrane that will oppose movement of X down the concentration gradient. At equilibrium, there will be no net flow of X across the membrane, and the electrical potential will be equal and opposite to the thermodynamic potential caused by the concentration gradient. The relationship between concentration gradient and electrical potential is described by the Nernst equation,

$$E_X = \frac{RT}{z_X F} \ln \frac{[X]_{out}}{[X]_{in}}$$

where E_X is the potential due to ionic species X (referenced to the inside of the cell), R is the gas constant, T is the temperature in degrees Kelvin, z_X is the valence of X, $[X]_{out}$ is the concentration of X outside the membrane, and $[X]_{in}$ is the concentration of X inside the membrane.

Note that if the membrane is permeable to other charge carriers in the medium, then space-charge neutrality will be maintained as counterions are dragged across the membrane with X. The concentration gradient of X will then be eliminated with no concomitment establishment of an ionic potential due to a charge imbalance from the movement of X.

The flow of ions through membrane channels has been the subject of much theoretical work, and at present there is no consensus as to the mechanisms involved (Hille, 1985). However, measurements of the intrinsic conductance of ion channels over a narrow range of membrane potentials [5] show that to a first approximation this intrinsic conductance is linear (independent of the voltage):

$$I_X = g_X(V_{membrane} - E_X)$$

where I_X is the ionic current, $V_{membrane}$ is the voltage applied across the membrane, and g_X is the conductance of X through the membrane channels.

[5] Typically in the physiological range of membrane potentials, and several millivolts away from the reversal potential of a given channel, where non-linearity of the intrinsic conductance is pronounced most.

An important assumption in the HIPPO model is that the flow of one ionic species across the membrane is independent of the flow of any other species; the different currents are linearly independent. This allows the different current paths to be represented as distinct independent conductances in parallel with each other, each driven by the appropriate ionic potential, as was illustrated in Figure 2.

In fact, it has been demonstrated that so-called "selective" channels are not absolutely selective for a given ion. Most channels are instead preferentially selective for one ion or another, and have a lower (perhaps much lower but non-zero) permeability for other species. The result is a reversal potential for a given channel that may be expressed by the Nernst-Goldman equation, including the appropriate ions and their relative permeabilities. For example, the reported reversal potentials for the (assumed) K^+ channels typically vary between -90 and -70 millivolts, whereas E_K, assuming standard values for the concentration of K^+ inside and outside the membrane, is about -98 millivolts. Likewise, data on I_{CaS}, which is advertised as a Ca^{2+} current, indicates that its reversal potential is around 0 millivolts (see Chapter 6). Finally, the resting potential in the model is assumed to be set by a non-voltage-dependent channel with a reversal potential of -70 millivolts, which implies that either there is a mixture of perfectly selective channels that contribute to give the observed "leak" channel, or there is a single channel that is permeable, to varying degrees, to more than one ion.

2.6.1 Sodium Potential - E_{Na}

In the HIPPO simulations, the sodium potential was not found to be a very critical parameter, probably since most of the activity of the cell occurs around potentials that are very hyperpolarized to the sodium potential. Changing this potential mainly affected the amplitude and rate of rise of the action potential. $[Na^+]_{in}$ is assumed to be 12 mM, and $[Na^+]_{out}$ is assumed to be 145 mM. At physiological temperature this corresponds to a potential of +63 mV.

2.6.2 Potassium Potential - E_K

The potassium potential is the most sensitive ionic potential in the model. Much of the reproduced activity takes place within 10 to 20 millivolts from rest. In addition, there is evidence that the potassium concentration adjacent to the outside membrane is substantially different than the rest of

the extracellular medium, which in turn will change E_K transiently during electrical activity. Further, the intracellular potassium concentration is not measured readily. In the model presented here, however, the concentration of K^+ inside and outside the cell is assumed constant. $[K^+]_{in}$ is set at 155mM, and $[K^+]_{out}$ is set at 4mM. At $37°C$, this corresponds to a potassium potential of -85 mV.

As shall be discussed in Chapter 7, raising the reversal potential of I_{DR} to -73 mV was necessary, in order to obtain certain features of the voltage response as mediated by this current. Different so-called K^+ conductances may, in fact, have slightly different reversal potentials, reflecting, as mentioned above, a non-ideal selectivity of a given channel. For example, the higher reversal potential of I_{DR} implies that this channel allows the passage of either a small proportion of Ca^{2+} or Na^+ as well as K^+.

2.6.3 Calcium Potential - E_{Ca}

The calcium potential was calculated from the constant extracellular Ca^{2+} concentration (4.0 mM) and the concentration of Ca^{2+} directly underneath the membrane regions where the Ca^{2+} channels are assumed to be grouped, $[Ca^{2+}]_{shell.1}$. At rest, $[Ca^{2+}]_{shell.1}$ was equal to 50 nM, resulting in a E_{Ca} of 128 mV. In Chapter 6, the dynamics of the Ca^{2+} system and the behavior of E_{Ca} are presented in detail.

Chapter 3

HPC ELECTROTONIC STRUCTURE AND DETERMINATION OF LINEAR PARAMETERS

3.1 Introduction

This chapter describes an estimation of the electrotonic structure of the hippocampal pyramidal cell model. The parameters for this structure are derived from the literature and from theoretical considerations that are developed herein. First, the basis for this development and the role it plays in the modelling effort will be described. Next, the parameters for the electrotonic structure will be defined and described, including the membrane capacitance, the cytoplasmic resistivity, and the factors underlying the membrane resistivity. The next section will begin by describing the problem of modelling the geometry of the cell. Two methods for estimating the dimensions will be presented, the first by extrapolating data used in other modelling studies, and the second based on a histological data-based technique that I have developed.

The electrical parameters of the cell will be estimated next based on reported data. When combined with the results of the previous section, some reports may be used to derive morphometric data, but not electrical parameters, and other reports may be employed for only some electrical measurements.

In order to develop a model structure that is consistent with the available (valid) data, the next section in this chapter derives the frequency response for the general structure. Next, DTFT techniques are used with the derived frequency response to examine various model structures so that the desired linear temporal response may be obtained. Several suggestions on how some parameters may be better estimated using the phase and magnitude of the frequency response are discussed, and the accuracy of the compartmental model used in the actual simulations is verified by comparison with the previously derived response of the continuous structure. Finally, the parameters of the structure used in this study are presented, along with a discussion of some of the possible functional implications of the values for these parameters.

3.2 The Importance of the Electrotonic Structure

In order to develop descriptions of non-linear elements in the pyramidal cell using the small amount of available data, building on an accurate characterization of the electrotonic structure of the cell is necessary. The term "electrotonic" refers to the cable-like characteristics of the cell as defined by the linear properties of the cell membrane , cytoplasm, and the intricate geometry of the dendritic tree.

Starting with a valid electrotonic structure is important for a few reasons. First, in the absence of complete voltage clamp data, the estimates for many of the non-linear parameters must be evaluated by current clamp simulations. In this case, subtleties in the resulting voltage records are analyzed to determine the accuracy of a given estimation. If the linear response of the model cell is different than that of the real cell, determining whether differences between simulated and actual responses are due to errors in the estimation of the non-linear parameters or to errors in the linear parameters may be impossible.

For example, one method used to derive the Na^+ currents in the hippocampal pyramidal cell involves running voltage clamp simulations on the linear model (no non-linear conductances) using an actual Na^+-only spike record as the clamp voltage. In this protocol, as will be reviewed in Chapter 5, the clamp supplies the current necessary to cancel the linear currents (leak current, soma-dendrite current, and soma capacitance current) elicited by the spike waveform. Presumably, then, the clamp current must reflect those

currents that are mediated by Na^+ channels during a Na^+-only spike. The time course of the clamp current therefore provides clues as to the non-linear processes that may underly the Na^+ currents (ref. Figure 5.2). For example, does the waveform indicate that more than one HH-like conductance is operating, and what are the magnitudes of the putative components? Models with different linear response will give different clamp currents under these conditions, so it is necessary that attention is focused on a model whose linear response most closely follows a real cell.

Another motivation to carefully develop the linear structure of the model came about when various references for this structure were consulted, including reports of measurements of cells and reports of other hippocampal pyramidal simulations. As will be reviewed later, many aspects of these reports were inconsistent, and required reviewing the assumptions inherent in these analyses and integration of the valid aspects of the reported data to obtain a more consistent description of the relevant parameters.

3.3 Building the Linear Description

Several papers on the measurement of the linear properties of hippocampal neurons were consulted to obtain the model parameters, including other modelling studies ([48], [44]), measurements of the linear properties of hippocampal neurons ([7], [52]), and references for analytical approaches to approximations of the neuron geometry ([26]). Typically these papers derive parameters via analysis of the assumed linear response to a hyperpolarizing current step. The analysis is based often on the calculated response of the soma/short-cable structure. Several methods are available to estimate a given parameter, and more than one is often used to estimate better a given parameter (e.g. [7]) .

In examining the published data, however, some problems arose when the derived parameters were checked either using the model or by running simple calculations. These inconsistencies will be addressed in this chapter.

3.4 HPC Linear Parameters

The linear parameters of the model include:

- Steady state input resistance as seen from the soma – R_{in} (Ω)
- Specific resistivity of the soma membrane – R_{m-soma} ($K\Omega\,cm^2$)
- Specific resistivity of the dendrite membrane – R_{m-dend} ($K\Omega\,cm^2$)

- Cytoplasmic resistivity – R_i ($K\Omega\,cm$)
- Specific membrane capacitance (assumed homogeneous)– C_m ($\mu f/cm^2$)
- Radius of the soma – a_{soma} (μm)
- Radius of the dendritic cable – a (μm)
- Length of the dendritic cable – l (μm)
- Length constant of the dendritic cable – λ (μm)
- Electrotonic length of the dendritic cable – L (dimensionless)
- Dendrite/Soma conductance ratio – ρ (dimensionless)
- Terminating admittance of the dendritic cable, normalized to that of a semi-infinite cable – B (dimensionless)

Some of these parameters are derived from the others, including λ and L:

$$\lambda = \sqrt{\frac{aR_{m-dend}}{2R_i}}$$

$$L = \frac{l}{\lambda}$$

Other parameters that are sometimes used for convenience include

- Cytoplasmic resistivity per unit length – r_a ($K\Omega\,cm^{-1}$)

where

$$r_a = \frac{R_i}{\pi a^2}$$

- (Typically dendritic) membrane time constant – τ_0 or τ (milliseconds)

where

$$\tau_0 = R_{m-dend}C_m$$

Many investigators refer to a homogeneous membrane resistivity, R_m, that is constant over both the soma and dendrites. This and each of the other parameters will be discussed in this chapter. The specific membrane capacitance, the cytoplasmic resistivity, the leak conductance, the electrode shunt conductance, and the leak reversal potential will now be discussed.

3.4.1 Specific Membrane Capacitance

The generally accepted value for C_m is 1 $\mu f/cm^2$. This value is comparable to the specific capacitance of .8 $\mu f/cm^2$ for a pure lipid bilayer ([19]). In some cells, however, a different value for C_m has been reported. For example,

the apparent membrane capacity for crustacean muscle fibers is 15 to 40 $\mu f/cm^2$ ([19]). If the true capacity per unit area is 1 $\mu f/cm^2$, this indicates that the membranes of these cells is quite contorted.

The capacitance of any given compartment was then calculated based on this value multiplied by the total surface area of the compartment that faced the extracellular medium. This calculation was based on several assumptions about the structure of the cell and the structure of the membrane. For example, ideal geometries were assumed when calculating the absolute value for the capacitance for any of the compartments in the HIPPO model – a sphere for the soma compartment and right cylinders for the dendritic and axonal compartments. In fact, the cell membrane is much more convoluted than this description implies, and the net result would be an underestimation of the cell capacitance. On the other hand, the value of 1 $\mu f/cm^2$ assumes a smooth membrane, without any small-scale variations. A more realistic calculation of membrane capacitance would take into account the inhomogeneity of the membrane and the variation of the membrane thickness. These factors would tend to reduce the capacitance per unit area.

In summary, the model cell incorporates a value of 1 $\mu f/cm^2$ for C_m. In addition, C_m is assumed to be constant over the entire cell (i.e. C_m is the same for both the soma and the dendrites). Some investigators have proposed larger values for C_m, for example ranging from about 2 to 4 μf/cm^2 ([52]). These values were derived from estimating the membrane time constant under assumptions that are probably not valid (e.g. homogeneous time constant over the entire cell, terminating impedance of dendrites = 0). The errors incurred under the various assumptions that have been used in other studies will be examined later, particularly when the estimation of R_m is discussed. These errors have likely contributed to an overestimation of C_m in some of these reports.

3.4.2 Cytoplasmic Resistance

The resistivity of the intracellular medium, the cytoplasm, is calculated with the assumption that the interior of the cell is homogeneous. This is clearly an assumption since the cell is packed with a myriad of cytostructural elements, organelles and inclusions. To a first approximation, however, the inhomogeneity of the cytoplasm is ignored.

Shelton [44] presents the following argument as to the size of R_i. He proposes that the lower limit of R_i is set by the conductivity of pure physiological saline solution, corresponding to a value of 50 to 60 $\Omega\, cm$. Measure-

ments of the resistivity of extracellular brain tissue are cited in the range of 50 - 600 $\Omega\,cm$, and measurements of the resistivities of axoplasm and somatic cytoplasm in other cells are quoted as being in the range of 20 - 160 $\Omega\,cm$ and 70 - 390 $\Omega\,cm$, respectively. Shelton proposes that the resistivity of the medium in which a microelectrode is immersed contributes to the effective electrode resistance due to the convergence resistance near the electrode tip. Since the microelectrode bridge circuit must be compensated to account for the electrode resistance, the compensation required as the electrode tip moves from outside to inside the cell will give an indication of the difference in the extra- and inter-cellular resistivities. Measurements along these lines indicate that the difference between these resistivities for the cerebellum and the Purkinje cell are less than 50 $\Omega\,cm$. Assuming that the cerebellar extracellular resistivity is 200 $\Omega\,cm$, Shelton then uses this result to suggests that R_i is near 250 $\Omega\,cm$.

This value of R_i is significantly higher than what is used usually in the reports analyzing the linear characteristics of the pyramidal cell. Typical values in these reports are in the range of 50 - 75 $\Omega\,cm$. Most studies do not indicate the rationale for these values, other than the supposed analogy to the resistivity of a Ringer's-type solution. An investigation of the significance of R_i was therefore of interest, in particular to see if large differences in this parameter could significantly affect the derivation of the other linear parameters.

The most obvious parameter that is a strong function of R_i is the dendritic length constant, λ, and thus the electrotonic length of a dendritic segment, L. λ is determined by R_i, R_{m-dend}, and a by the following relation:

$$\lambda = \sqrt{\frac{R_{m-dend}a}{2R_i}}$$

(Note that λ expresses the length over which the voltage from a *constant* point source attenuates by a factor of $1/e$ down an *infinite* dendritic cable.)

For a fixed value of R_{m-dend} and a, a four-fold increase in R_i (e.g. from 65 to 260 Ω cm) will correspond to a halving of λ. The manner in which R_i affects the input impedance of the cable is discussed later.

One of the assumptions of the compartmental model is that within each compartment the intracellular resistance can be neglected, so that the compartment is isopotential. The cytoplasmic resistivity is only considered in the electrical communication between dendritic compartments, where the

connecting resistor is calculated from the dimensions of the compartments and the cytoplasmic resistivity according to the formula –

$$R_{coupling} = \frac{R_i \times l_{compartment}}{\pi a^2}$$

where $l_{compartment}$ is the length of the dendrite segment.

In summary, the model incorporates an R_i of either 200 or 250 Ω cm for most of the analyses. In some cases, R_i was set to 75 Ω cm in order to evaluate data from other reports of intracellular measurements or modelling studies, but results presented in later chapters are obtained using the higher values of R_i.

3.4.3 Leak Conductance, Electrode Shunt Conductance, And Leak Reversal Potential

R_m, the specific membrane resistivity, is defined as a linear, time-independent conductance. The intrinsic leak conductance of the cell, R_{leak}, and the electrode shunt conductance, R_{shunt}, combine to form R_m when the impedance of the membrane is evaluated. R_{leak} includes the conductance of the lipid bilayer, and an ion-specific channel or channels whose conductance is or are voltage and time independent. R_{shunt} is the non-specific leak arising from the impalement of the cell with a microelectrode. Since R_{leak} is a selective conductance, it is modeled in series with a voltage source representing the leak reversal potential, E_{leak}. R_{shunt}, however, is non-selective, and therefore is modeled without a series voltage source.

Accurate determination of R_m is difficult, particularly because of the cable properties of the pyramidal cell and, as will be demonstrated, the non-homogeneity of R_m. In this section some estimates of R_{shunt} are presented as well as a background for the measurement of the intrinsic R_{leak} and estimates of E_{leak}. The estimates of R_m (actually of R_{m-soma} and R_{m-dend}) will be presented later in this chapter.

The conductance of the lipid bilayer sets an upper bound for the R_{leak} of 10^8 - 10^9 $\Omega\,cm^2$ [19]. Since estimates of R_m typically are in the range of 500 to 10^4 $\Omega\,cm^2$, ion channels or the electrode leak appears to account for the majority of the total membrane leak.

Various estimates of the leak introduced by an electrode have been made, ranging from 50 to 200 MΩ [44]. We can roughly estimate the magnitude of the leak introduced by the single electrode used in the single electrode clamp (SEC) protocol from the amount of constant "repair" hyperpolarizing

current that must be supplied to the soma in order to maintain a resting potential of -60 to -70 mv [1] . Typical values for this current range from 0.5 to 1.5 nA (Storm, personal communication). If the normal resting potential is assumed to be -70 mv, and sufficient repair current is supplied to restore this membrane voltage, then the previous range of repair current magnitudes implies electrode leaks in the range of 140 to 47 MΩ , respectively.

Estimates of pyramidal cell input impedance vary over an order of magnitude. This range is more than can be explained simply by the difference in the surface area and electrotonic structure of the measured cells. The integrity of the electrode seal is variable, and could conceivably account for a large part of the input conductance.

For many cells R_m is estimated by measuring the time constant of the cell in response to small steps of injected current with the cell at resting potential. In this case, either the cell membrane is assumed to be equipotential (in which case the response should consist of a single exponential and the single time constant is measured), or an infinite cable structure is assumed with a homogeneous membrane, and the largest time constant of the response is interpreted as the true membrane time constant. This formula shall be referred to later when some of the published estimates of R_m are examined.

Typical values for the time constants measured under these conditions for various cells (including non-neuronal cells) range from 10μ s to 1 second. This range corresponds to R_m's of 0.30 to $10^6 \Omega\,\mathrm{cm}^2$, assuming that C_m can range from 1 - 30 $\mu\mathrm{f/cm}^2$. Thus the number of channels that are open and contribute to the maintenance of the resting potential varys considerably between different cell types.

The stability of the resting potential may be investigated by perturbing the membrane voltage in the presence of active conductances. These simulations can test the validity of any calculated E_{rest}, since it is likely that the membrane voltage would be stable in the neighborhood of the actual E_{rest}, and that the spike threshold would be distinct (e.g. greater than 10mv depolarized from rest). This stability of E_{rest} is observed for non-spontaneously firing cells, and is advantageous since this behavior is directly related to the ability of the cell to reject noise (at a low level of perturbation) and the integrative ability of the cell. In the latter case, a firing threshold near E_{rest}

[1] This repair current is often only transiently required, however, as if over time the leak introduced by the micro-electrode is sealed automatically (Storm, personal communication)

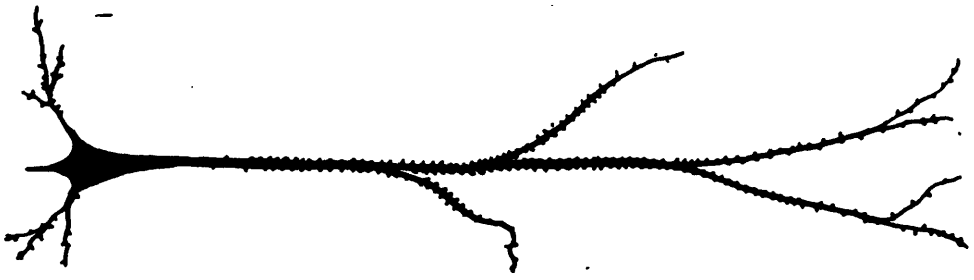


Figure 3.1: Typical hippocampal pyramidal cell. The main regions include the soma, the basal dendrites, the apical dendrites, the axonal hillock, and the axon.

would cause the cell to fire for a larger set of inputs than if the threshold was more depolarized. Modulation of the firing threshold is a possible physiological mechanism for changing the computational function of a single cell. On the other hand, threshold modulation may be a factor is some pathological states, such as epilepsy where the threshold is abnormally low leading to hyperexcitability (e.g. seizures), or in states where the threshold is too high, causing hypoexcitability (e.g. paralysis at the extreme).

In most reports, E_{rest} is assumed to be about -70 mv. Since the evidence for hippocampal pyramidal cells indicate that there is little current due to non-linear channels at rest (the exception being a small I_m, discussed in Chapter 6), I have assumed that the reversal potential for the leak conductance, E_{leak}, is equal to -70 mV.

3.5 Modelling the Cell Geometry

The shape of the hippocampal pyramidal neuron is quite complex, as Figure 3.1 illustrates.

The basic regions of the pyramidal neuron are the cell body, or soma, the basal dendrites, the apical dendrites, the axonal hillock, and the axon. Synaptic input to the cell is received at all its regions, but is primarily received on the dendritic trees. In the standard view of the HPC, the dendritic membrane is primarily linear while the somatic, axon hillock and axon membranes are active, that is contains non-linear voltage and time dependent conductances. Although recent studies show that there are non-linear conductances located on the dendrites, in the present model purely linear

dendrites are assumed.

3.5.1 Assumption of Linear Dendrites

Defining the dendrites to be linear is an important assumption for the model. There is extensive evidence of various Na^+, Ca^{2+}, and Cl^- channels in the dendrites ([53], [34], [31], [50], [5]), but the exclusion of dendritic non-linear conductances was considered reasonable as a first approximation since the present work is focused on the action of somatic currents. I assume that the behavior of the somatic non-linear conductances are relatively insensitive to regions of non-linear dendritic membrane, at least when considering cell response to somatic input. In Chapter 9, simulations of somatic response to dendritic input will be presented, in which the assumption of linear dendrites is a more restrictive one in terms of interpreting the model results.

The primary function of the dendrites is to collect and integrate synaptic input from other neurons. That input is conducted to the soma where an action potential is initiated if the soma membrane is excited above the local threshold. As far as the model is concerned, though, the contribution of the dendrites is simply that of a linear load on the soma.

3.5.2 Approaches to the Representation of HPC Structure

The possible options for representing the structure of the pyramidal cell in simulations are worthy of review. At one extreme, the entire geometry of the cell and its dendritic tree may be modeled in detail, with the dendritic tree reduced to a set of branching cylinders, perhaps including the tapering of each cylinder and the dendritic spines. The appropriate linear cable equations may then be employed to examine the steady-state input conductance of the entire tree ([52]), assuming linear dendrites. If the transient response is of interest, or if non-linear dendritic conductances are to be included, representing the cable segments with compartmental approximations and solving the network numerically is necessary ([44]).

Histological technique can supply the data necessary for this sort of representation, but the attempts to model dendritic trees at least approximate the tapering segments as right cylinders. The hippocampal pyramidal cell has been modeled in this fashion ([52]). In this study, the dendritic tree was dissected into a branching structure of right cylinders, without spines. Several cells were analyzed, with the dendritic trees modeled with 300 - 1,000 cylinders per cell. Using the equation for the input conductance of a short

cable and moving proximally from each distal termination, the steady-state input conductance of the entire tree was derived as a function of membrane resistivity and cytoplasmic resistivity. A study of a Purkinje neuron ([44]) represented the cell with 1089 coupled compartments. In this case the dynamic behavior of the linear cell was derived numerically in order to estimate its linear properties.

The next level of approximation in reducing the dendritic tree consists of collapsing branched structures into equivalent cylinders, according to the technique developed by Rall [26] (described shortly). The complexity of the resulting representation (i.e. how much will the tree be collapsed into larger cables), depends on the morphological characteristics of the dendrites and the accuracy desired by the modeller. In this model, several versions of such a geometry were used, as illustrated in Figure 3.2. For investigating somatic properties the dendritic tree was sufficiently represented as a single short cable, as shown in Figure 3.2. On the other hand, as was mentioned at the beginning of the chapter, the parameters of this approximation, the dimensions of the soma and the cable and their linear electrical properties, were critical to the response of the model, and their careful estimation is important.

At the other end of the spectrum, in representing pyramidal cell geometry, is an isopotential sphere representing the entire cell. For most modelling studies of cells with a significant dendrite tree this approach is too simplified for two reasons. First, the linear response of the sphere will consist of a simple exponential, precluding the role of the dendrites as relatively isolated stores of charge that contribute to restoring the soma voltage after short perturbations. Second, the majority of voltage-dependent ion channels are believed to be localized at the soma, and that the dendrites are either linear, or incorporate localized, lower density, non-linear conductances. Modelling the cell as an isopotential sphere prevents considering such a distribution of non-linear and linear membrane.

In summary, modelling with a detailed description of the dendritic tree is necessary if one is interested in evaluated the complex information processing that apparently occurs at the level of distinct regions of the tree. If, however, one is interested only in somatic properties, as a first approximation the tree may be collapsed so that its approximate load as that of a single short cable may be evaluated. A next step in the analysis of somatic properties may use a slightly more complicated approximation to the tree structure, as shown in Figure 3.2, and has been used by Traub and Llinas([48])(see

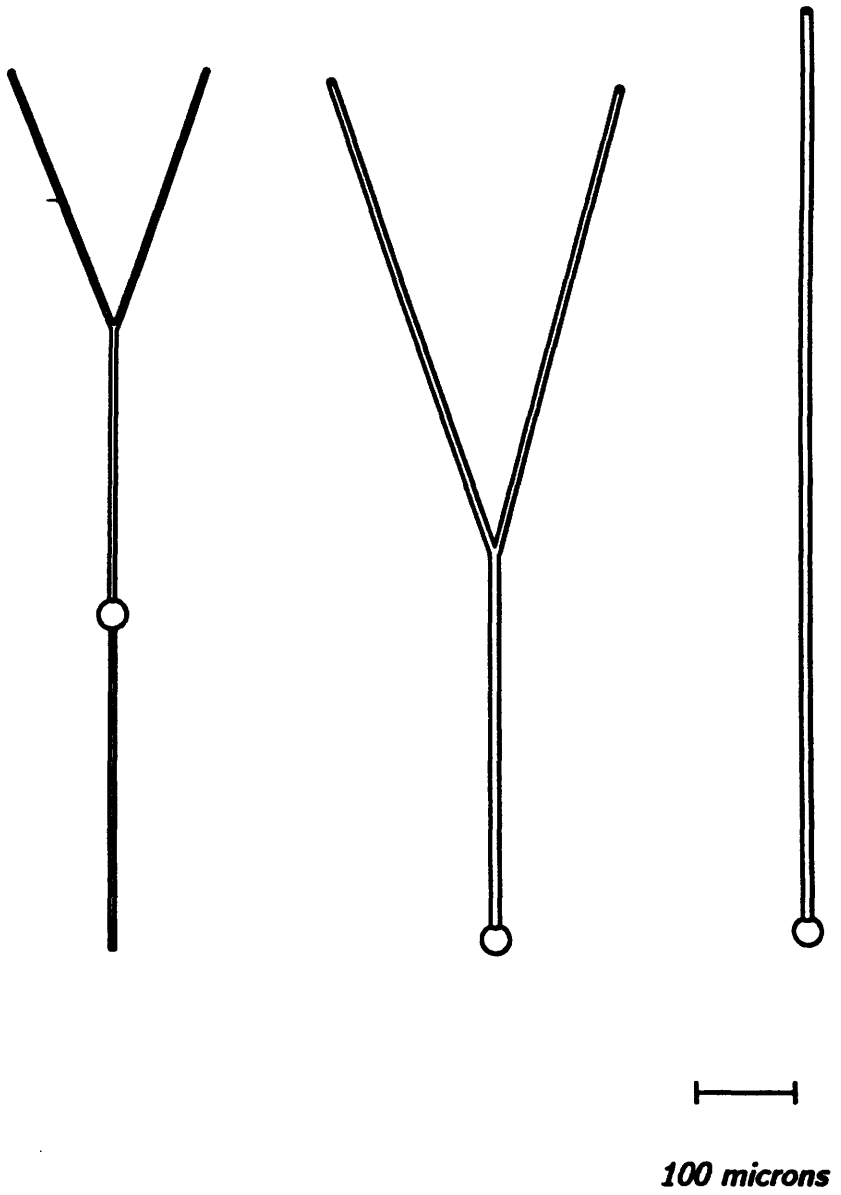


Figure 3.2: Different model geometries used to approximate hippocampal pyramidal cell in present study, drawn approximately to scale. The simple soma/short-cable structure on the right was used for the majority of the analyses.

also Figure 3.2). In these versions, the contribution of non-linear dendritic membrane may be considered, where the non-linear membrane is localized in some isolated section of a dendritic branch. The simulations of Traub and Llinas have provided some interesting results in this area, but their model parameters may not have been derived carefully enough to warrant any more than qualitative interpretations of the behavior of their model.

After the Rall method of reducing dendritic trees has been introduced, several methods for estimating the model geometry will be presented.

3.6 The Rall Reduction Of The Dendritic Tree To Equivalent Cylinders

Rall has shown that under certain conditions a set of dendritic branches emerging from the distal end of a dendritic segment may be collapsed into a single cable whose input impedance (as seen from the parent segment) is identical to that of the original set. The conditions for the reduction of each set of branches into a single cable are twofold: 1) the terminating impedance of each branch must be the same, and 2) the electrotonic length of each branch must be equal. The electrotonic length of the new equivalent branch is the same as the original branches, and its diameter, raised to the 3/2 power, is equal to the sum of the original diameters, each as well raised to the 3/2 power. The terminating impedance of the equivalent branch is equal to the terminating impedance of each of the original branches.

If the diameter of the equivalent branch or segment is equal to that of the more proximal parent segment, then the two cables connected in series are equivalent to a single longer cable. As long as the appropriate conditions hold, the entire dendritic tree can be represented by a single cable by applying the reduction algorithm repeatedly, starting from the distal branches and continuing proximally to the soma.

The constraints for the Rall reduction are rather severe, and several types of neurons have been analyzed to see if the above conditions are applicable. Remarkably enough, some neurons seem to follow the so-called "3/2 rule" (e.g. in lateral geniculate nucleus [44]), and the suggestion has been made that the Rall reduction is quantitatively valid for them (although it is not always clear if the authors of these studies of fully aware of the complete set of constraints in the reduction algorithm [Rall, personal communication]).

For hippocampal pyramidal cells, the reviews have been mixed, with quantitative studies based on detailed histological data suggesting that the

Rall constraints are not met at all well ([52]). Despite this, the reduction as described is still considered a good first approximation to the pyramidal trees, and some studies have suggested that in terms of the dendritic input impedance the Rall approximation is in good qualitative agreement with the actual tree structure (Brown et al). Later in this chapter the responses of a soma/single-dendritic-cable and a soma/double-dendritic-cable will be compared to show qualitatively that the Rall reduction is a useful one even when the electrotonic lengths of the daughter branches are not identical.

3.7 Approximation Of The Soma As An Isopotential Sphere

The so-called "soma" of the hippocampal pyramidal cell is not a sphere; it is more of tapered cylinder with rounded ends. Further, the transition between soma and dendrite is not well-defined, especially for the apical processes. The soma region is assumed to be well-defined, however, in the model approximation. This region is also assumed to be isopotential. This assumption allows the use of a sphere instead of a cylinder to represent the soma, as long as the surface area of the soma is conserved. The isopotential approximation assumes that voltage gradients are minimal, despite the finite cytoplasmic resistivity. It can be shown ([26], Ch. 3) that the spread of current from a single intra-somatic point source introduces a very small voltage gradient in the soma.

The dimensions of the soma were evaluated from the model soma used in Traub and Llinas's model, and from estimating the dimensions from micrographs. The soma used in the Traub and Llinas model was a cylinder, so the surface area of this soma was used to set the radius of the spherical soma in the present model at 17.5 μm. This value is consistent with the size of the soma region seen in micrographs.

3.8 Estimating the Dimensions of the Model Dendrites

Traub and Llinas' paper provided the default dimensions of the dendritic cable of the model as well. In their model the dendritic tree was represented by a two short cable basal dendrites and a short cable apical dendrite that terminated into two short cable apical branches (Figure 3.2). The HIPPO

model topology was initially configured the same way, using Traub and Llinas' dimensions. In their paper, however, the effect of dendritic input and localized regions of non-linear membrane in the dendritic tree were investigated, necessitating the described geometry. Since at the present time the description of soma currents is being investigated, this tree structure was collapsed into a single cable using a variant on the Rall method.

Traub and Llinas used a homogeneous $R_m = 3.0\,\mathrm{K}\Omega\,\mathrm{cm}^2$, and set $R_i = 75\Omega\,\mathrm{cm}$.

3.8.1 Deriving the Dimensions of a Single Cable That is the Approximate Equivalent of Two Cables

The first step in this approximation was collapsing the basal branches and apical branches into a single basal and apical cable. This step was straightforward since both the basal branches and the apical branches were the same electrotonic length as their partners, and further in that the diameter of the apical shaft satisfied the 3/2 rule with its daughter branches.

The second step was to combine the equivalent apical cable (ac) with the equivalent basal cable (bc). This was not straightforward since the equivalent apical and basal cables were not the same electrotonic length ($L_{ac} = 0.8, L_{bc} = 0.6$). The approach used was to calculate a according to the 3/2 rule, and then calculate l so that that the single cable would have the same steady-state input impedance as the original two cables in parallel.

First, the diameter of the single "equivalent" cable (sc) was derived from the 3/2 rule:

$$a_{sc} = (a_{ac}^{3/2} + a_{bc}^{3/2})^{2/3} \tag{3.1}$$

where a_x is the radius of the appropriate cable, yielding $a\,(= a_{sc}) = 4.3\mu\mathrm{m}$. The next step was to derive the length of the single cable, starting with the formula for the parallel input impedance of the original cables:

$$Z_{sc}(s=0) = \frac{Z_{ac}(s=0) \cdot Z_{bc}(s=0)}{Z_{ac}(s=0) + Z_{bc}(s=0)} \tag{3.2}$$

From the equation for the input impedance of a short cable (Equation 3.20, derived later, with $s = 0$ since we are interested in the steady state impedance, and $G_{soma} = 0$ and $C = 0$, since we are interested in the impedance of the isolated cable), Equation 3.2 becomes

$$\frac{1}{r_{a,sc}\lambda_{sc}} \tanh L_{sc} = \frac{1}{r_{a,ac}\lambda_{ac}} \tanh L_{ac} + \frac{1}{r_{a,bc}\lambda_{bc}} \tanh L_{bc} \qquad (3.3)$$

Now since

$$\begin{aligned} r_{a,sc} &= \frac{R_i}{\pi a^2} \\ r_{a,ac} &= \frac{R_i}{\pi a_{ac}^2} \\ r_{a,bc} &= \frac{R_i}{\pi a_{bc}^2} \end{aligned}$$

$$\begin{aligned} \lambda_{sc} &= \sqrt{\frac{aR_m}{2R_i}} \\ \lambda_{ac} &= \sqrt{\frac{a_{ac}R_m}{2R_i}} \\ \lambda_{bc} &= \sqrt{\frac{a_{bc}R_m}{2R_i}} \end{aligned}$$

$$L_{sc} = \frac{l}{\lambda_{sc}}$$

and if a is derived from Equation 3.1, then from Equation 3.3 we obtain

$$\tanh(\frac{Kl}{\sqrt{a}}) = \frac{a_{ac}^{3/2} \tanh(L_{ac}) + a_{bc}^{3/2} \tanh(L_{bc})}{a_{ac}^{3/2} + a_{bc}^{3/2}} \qquad (3.4)$$

where

$$K = \sqrt{\frac{2R_i}{R_m}}$$

By expanding the tanh term on the left side of Equation 3.4, and by making the substitution (from Equation 3.1) of

$$\sqrt{a} = (a_{ac}^{3/2} + a_{bc}^{3/2})^{1/3}$$

the length, l, of the single cable is found to be

$$l = \frac{(a_{ac}^{3/2} + a_{bc}^{3/2})^{1/3}}{2K} \ln\left(\frac{1 + \alpha_1 \tanh(L_{ac}) + \alpha_2 \tanh(L_{bc})}{1 - \alpha_1 \tanh(L_{ac}) - \alpha_2 \tanh(L_{bc})}\right)$$

where

$$\alpha_1 = \frac{1}{1 + (\frac{a_{bc}}{a_{ac}})^{3/2}}$$

$$\alpha_2 = \frac{1}{1 + (\frac{a_{ac}}{a_{bc}})^{3/2}}$$

This procedure gave $l = 850\mu$m, and from this L was calculated as $L = 0.69$. To check this reduction, the transient response to a current step of this configuration was then compared with the response of the original geometry of Traub and Llinas (Figure 3.4). The responses were nearly identical, validating the approximation *between these two geometries.*

Important inconsistencies arise, however, when the linear response of the Traub and Llinas model is compared with that of actual cells. These will be examined once the data derived from intracellular measurements has been presented. At this point, this model will be used only to establish a reasonable set of dimensions for the HIPPO model.

3.8.2 A New Method of Estimating l and a For the Equivalent Cylinder Approximation From Histological Data

In order to check the validity of the dimensions used in Traub and Llinas' model, a method was derived for estimating l and a from purely histological data, that is, without relying on estimates of electrical properties. The parameters used for this estimation include:

- Average length of the dendritic tree – l_{av} (μm)
- Average radius of the dendrite branches – a_{av} (μm)
- Radii of the i proximal dendrites where they attach to the soma – a_i (μm)

The radius of the equivalent short cable of the entire tree, a, is then set by the a_i's under the assumption that the radius of each proximal segment is the same as the radius of the equivalent cylinder approximation for the portion of the tree distal to that segment. Thus –

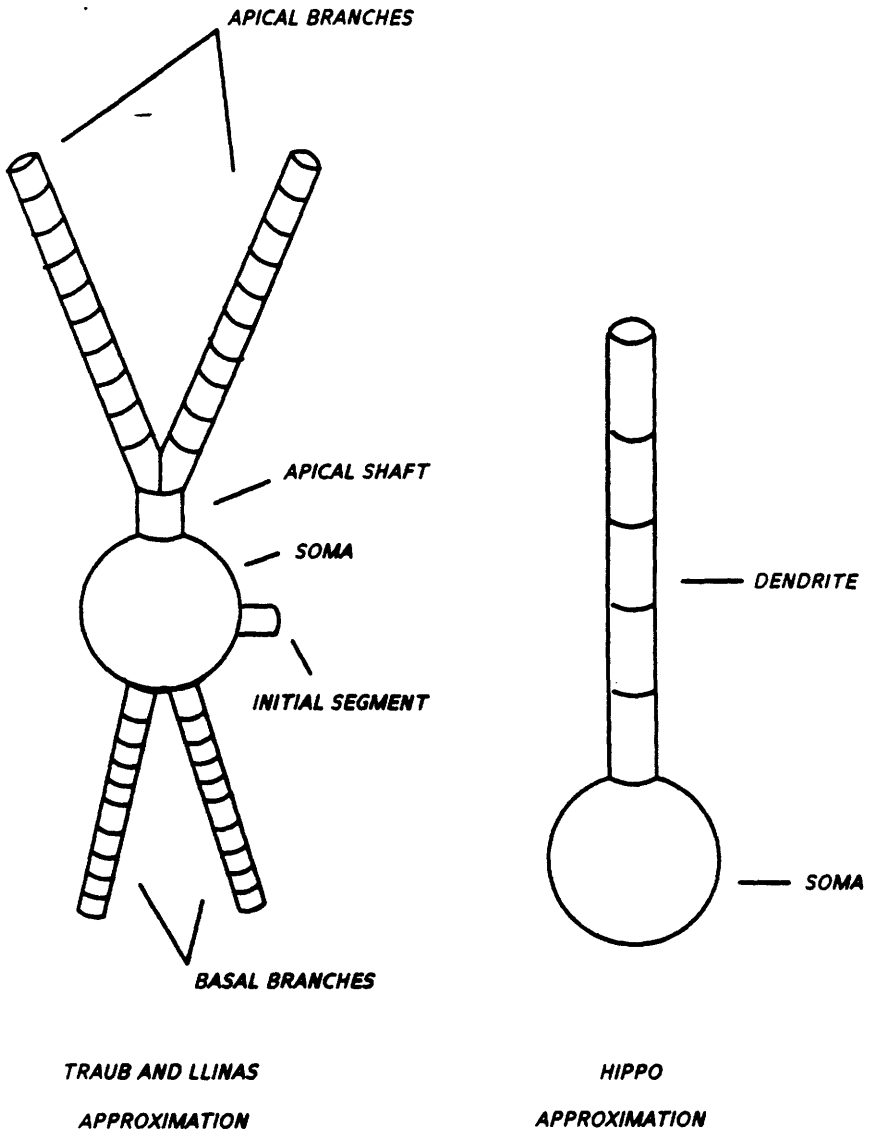


Figure 3.3: Comparison of cell geometry approximation used by Traub and Llinas and single cable approximation used in the model. Structures are not drawn to scale.

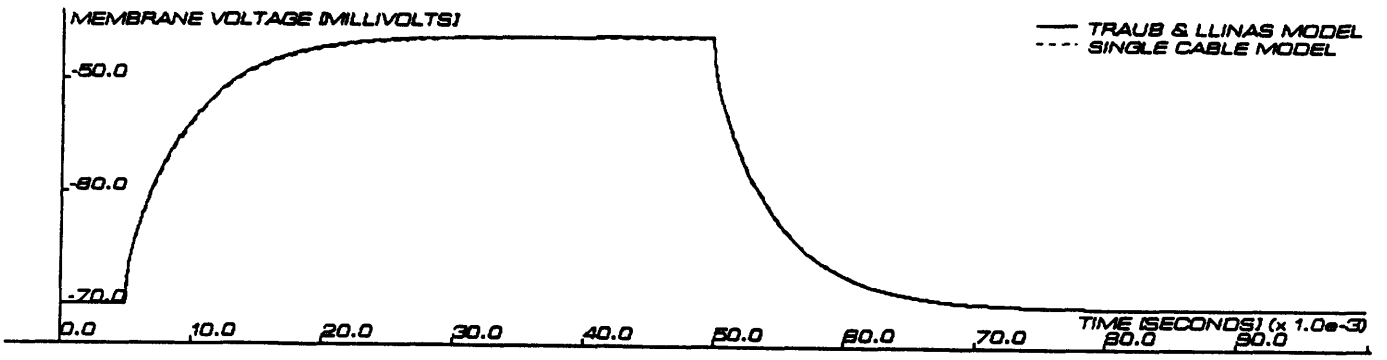


Figure 3.4: Comparison of response to current step of geometry used by Traub and Llinas and single cable approximation used in the model.

$$a = \left(\sum a_i^{3/2}\right)^{2/3}$$

The length of the equivalent cable for the entire tree, l, is a function of the average length of the dendritic tree, l_{av}, the average diameter of the dendrite branches at about the midpoint of the tree, a_{av}, and the estimated a. Here an assumption is made that the tree can be represented by a number of identical cables with radius a_{av} and length l_{av}. As previously mentioned, the Rall method requires that electrotonic length be conserved in the equivalent cylinder. The L for the cable representing the entire tree is therefore estimated as

$$L = \frac{\sqrt{\frac{a_{av}R_{m-dend}}{2R_i}}}{l_{av}} \tag{3.5}$$

Note that the numerator of the right hand side of Equation 3.5 is the space constant for the "average" cable.

Since l is equal to

$$l = \frac{\sqrt{\frac{aR_{m-dend}}{2R_i}}}{L} \tag{3.6}$$

Equations 3.5 and 3.6 can be combined to get

$$l = l_{av}\sqrt{\frac{a}{a_{av}}}$$

This estimate for l is function only of the dimensions of the tree. The significance of this estimate this is that l may be derived purely from histological measurements and does not depend on an assumed value for R_{m-dend} or R_i. The estimate of L, on the other hand, does depend on the estimated values of R_{m-dend} and R_i. Further, a_{av} is not the same thing as the average diameter of the proximal branches. a_{av} must be used as defined since the main part of the electrotonic length of the dendritic tree is determined by the finer and more numerous distal processes. Thus the diameter of *these* branches must be considered in estimating L (or l).

Typical values for a_{av} for the hippocampal pyramidal cell are in neighborhood of 0.5 - 1.0 μm. At the soma there is typically either one or two apical branches, with a diameter ranging from 3 - 10 μm (e.g. Johnston and Brown, 1983). There are usually several proximal basal branches, with a typical diameter of about 1 μm. A reasonable value for l_{av} could range from 300 - 500 μm. For example, if there are two apical dendrites originating at the soma with diameters of 3.0 μm and 4.0 μm, and there are six basal branches at the soma, each with a diameter of 1.0 μm, with the above ranges for a_{av} and l_{av}, the estimated value for a is 3.6 μm and the estimated range for l is 570 - 1300 μm.

As a second example, let us assume that there are two apical dendrites have diameters of 3.0 μm and 10.0 μm, and the six basal branches stay the same as before with diameters of 1.0 μm. Using the same ranges for a_{av} and l_{av}, the estimated value for a now is 6.8 μm, and the estimated range for l is 780 - 1800 μm.

These values can be compared with the dimensions of the equivalent cylinder derived from the Traub and Llinas model. The value for a in this report was 4.3 μm, the length of the equivalent cylinders for the basal branch and the apical branch were 555 μm and 820 μm, respectively, and the length of the single equivalent cable that was derived in this paper was 850 μm. These numbers compare well with the figures above. In fact the authors comment that their estimate for l of their model's apical cylinder was "possibly somewhat small". How the dimensions of this model were derived is not known, but presumably an analysis similar to the one just presented was employed.

To summarize, the dimensions of the Traub and Llinas model are in good agreement with the previous estimate. These results will be used both in the next section to test the validity of another report which implies a set of dimensions, and later in this chapter when the final dimensions of the model will be determined. Another estimation of the model cell dimensions, this time based on reported parameters of CA1 cells derived from intracellular electrical measurements, will now be presented.

3.9 Evaluating Reported Linear Parameters Derived from Intracellular Measurements

The report used as a basis for this analysis is that by Brown et. al. ([7]). In this paper essentially three parameters were derived from the linear responses of hippocampal pyramidal cells. These parameters included R_m, which was assumed to be homogeneous over the entire cell, L and ρ. C_m was taken to be $1.0\mu f/cm^2$, and R_i was assumed to be 75Ω cm. Analysis of the response of the cell to a current step applied to the soma was based on the assumption that the cell could be approximated by the soma/short-cable model with a homogeneous membrane time constant. According to Rall (), this step response can be represented by a linear combination of exponential terms:

$$V_f - V = \sum_{i=0,\infty} C_i \exp(-t/\tau_i)$$

where V is the response at the soma relative to rest, V_f is the steady-state soma voltage, τ_0 is the membrane time constant ($\tau_0 = R_m C_m$), and the remaining τ_i's are shorter time constants due to charge redistribution down the dendrite cable. Standard exponential peeling techniques were used to evaluate the longer τ_0, whose coefficient, C_0 was assumed to be much larger than the remaining C_i's. R_m was then derived from the measured τ_0. Three methods were used to derive L and ρ, all of which were dependent on the soma/short-cable approximation and, as before, the assumption of homogeneous R_m. This study estimated R_m as $19K\Omega\,cm^2$, ρ as 1.2, and L as 0.95. R_{in} averaged about $39M\Omega$ [2].

To evaluate these results, I constructed a model geometry that was consistent with the above values for R_{in}, ρ, τ_0, R_m, C_m, R_i, and L. The

[2] Means of measurements from CA1 cells.

parameters that we need to derive for the geometry are the radius of the soma, a_{soma}, the radius of the dendrite, a, and the length of the dendrite, l.

The first step is to derive a_{soma}. The conductance of the soma is calculated from ρ and R_{in}. Since

$$\rho = \frac{G_{dendrite}}{G_{soma}}$$

and

$$G_{dendrite} + G_{soma} = \frac{1}{R_{in}}$$

then

$$G_{soma} = \frac{1}{R_{in}}(\frac{1}{1+\rho})$$

This gives $G_{soma} = 11.8\,\text{nS}$. The radius of the soma is then calculated from G_{soma} and R_m:

$$a_{soma} = \sqrt{\frac{3R_m G_{soma}}{4\pi}}$$

This results in $a_{soma} = 73\,\mu\text{m}$. Now the formula for R_{in} is a function of l and a, given by:

$$R_{in} = \left(\frac{1}{r_a \lambda}\tanh(L) + G_{soma}\right)^{-1} \tag{3.7}$$

where

$$L = l \times \sqrt{\frac{aR_m}{2R_i}} \tag{3.8}$$

$$\lambda = \sqrt{\frac{R_{m-dend}a}{2R_i}} \tag{3.9}$$

$$r_a = \frac{R_i}{\pi a^2} \tag{3.10}$$

Equation 3.7 is derived later (Equation 3.20, with $s = 0$). Estimates for l and a were obtained by calculating R_{in}, L, and ρ, using initial estimates for l and a with Equations 3.7- 3.10, and then adjusting l and a until the desired values for R_{in}, L, and ρ were obtained. This procedure resulted in estimates for $l = 1800\mu\text{m}$ and $a = 3\mu\text{m}$.

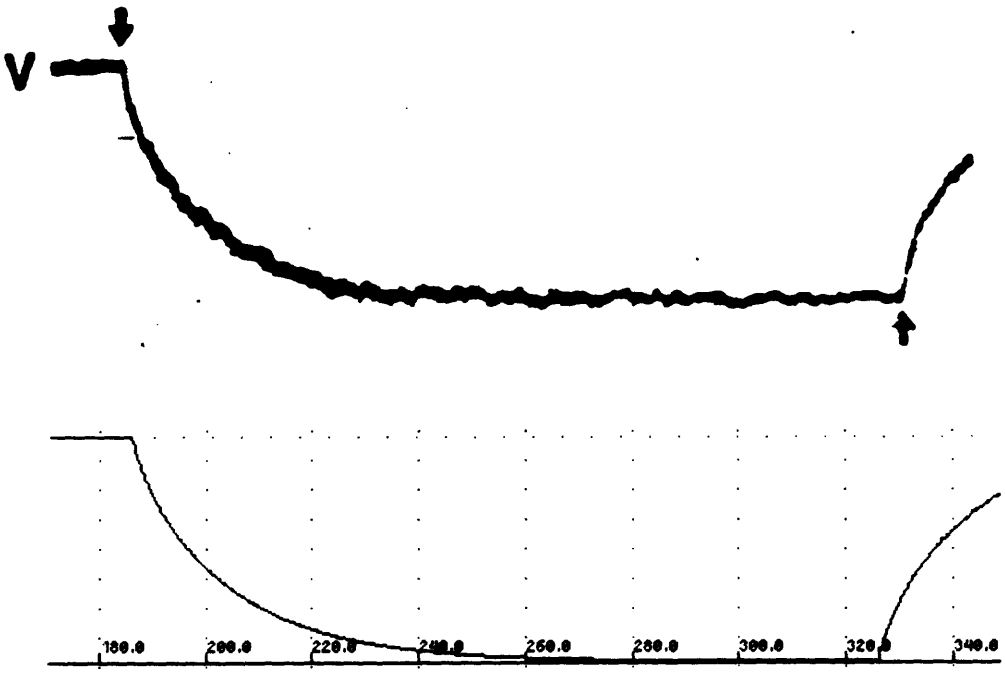
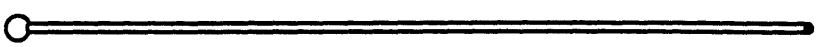


Figure 3.5: Typical response to current step ([7]) (top) and response to current step of model based on Brown et. al. parameters (bottom), where $\rho = 1.2$, $L = 0.95$, $R_{in} = 39\text{M}\Omega$, $a_{soma} = 73\mu\text{m}$, $a = 3.0\mu\text{m}$, and $l = 1800\mu\text{m}$.

The step response of the geometry just derived and a typical step response from the Brown et. al. paper is seen in Figure 3.5. These responses are in good agreement. On the other hand, note Figure 3.6, where the resulting geometry and the geometry derived in the previous section are compared. The most striking feature of the geometry derived from the Brown et. al. data is the estimated soma radius of 73 μm. This result is inconsistent with the dimensions derived earlier, where the a_{soma} was estimated to be on the order of 10 to 20μm. The dendrite radius of 3.0 μm and a dendrite length of 1800 μm of the Brown et. al. geometry is consistent with the previously derived dimensions, but these values are in the extreme of the previously proposed ranges for a and l.

HIPPOCAMPAL PYRAMIDAL CELL

BROWN ET AL GEOMETRY



HIPPO GEOMETRY

100 microns

Figure 3.6: Comparison of soma/short-cable geometries derived from data of Brown et. al. and that estimated in this chapter with camera lucida reconstruction of guinea pig hippocampal pyramidal cell ([52]).

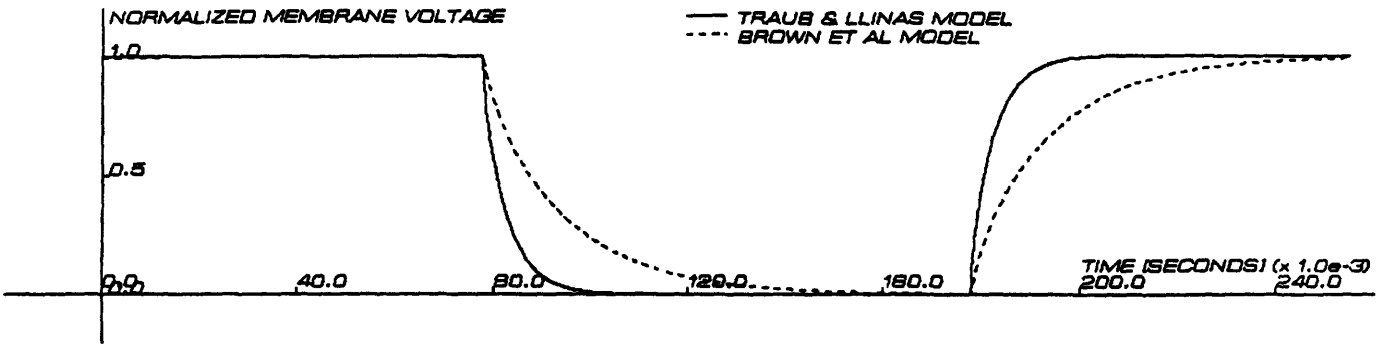


Figure 3.7: Normalized response to current step of Brown et. al. geometry and Traub and Llinas model.

3.10 Comparison of Linear Response of Traub and Llinas–Derived Model and Brown et. al.–Derived Model

While the geometry implied by the Brown et. al. report is incorrect based on my earlier analysis, the step response is assumed to be valid since this was measured directly from cells. On the other hand, while the geometry of the Traub and Llinas-derived model is a good approximation, as I have shown with my estimate based on purely histological data, the step response of this structure does not match that reported by Brown et. al. , as shown in Figure 3.7. All these reports refer to pyramidal cells, though not necessarily to the same subfield (i.e. CA1, CA3).

The first difference is the τ_0 for the two models; τ_0 for the Traub and Llinas model is about 5 milliseconds (consistent with their value of R_m, $5\,K\Omega\,\mathrm{cm}^2$), and τ_0 reported by Brown et. al. is about 19 milliseconds. The second difference is between the value of ρ for the Traub and Llinas model (approximately 20) compared to values of ρ that have been reported from intracellular measurements by Brown et. al. and others (ρ = .5 to 2). Comparing the directly measured value of ρ from the Traub and Llinas model with the estimated ρ of Brown et. al. is valid since in the latter case ρ was estimated assuming models a soma/short-cable structures with homogeneous R_m.

As has been mentioned earlier, these disparities were reconciled by introducing a distinct R_{m-soma} and R_{m-dend}. Investigating the effect of varying R_i was also useful. Before deriving a structure which was consistent with the reported data, though, deriving the analytical response of the general soma/short-cable structure (with non-homogeneous R_m) is necessary so that the full implications of varying each free parameter may be analyzed.

3.11 Derivation of the Frequency Response of Soma/Short-Cable Structure with Non-homogeneous Membrane Resistivity

So far I have presented evidence that supports using a spherical isopotential soma attached to a short dendritic cable, with each section having a distinct membrane resistivity, in order to represent the hippocampal pyramidal cell. This representation, as diagrammed in Figure 3.8, is completely specified by the parameters R_{m-soma}, R_{m-dend}, R_i, C_m, a_{soma}, a, B, and l.

Different investigators have considered the effect of the extreme values of B: $B = 1$ (infinite cable termination) and $B = 0$ (open circuit/sealed end termination). Assuming that the distal dendrite processes end rather abruptly is common, though, and therefore the sealed end assumption is used, as is done in the present analysis.

To investigate the effect of these parameters on the linear transient and steady-state response of the cell, as measured from the soma, I derived the frequency response of this circuit as follows.

We start with the equation for the linear RC cable.

$$\frac{\partial^2 V}{\partial X^2} = V + \frac{\partial V}{\partial T}$$

where V = the membrane voltage at some point, X; X = the distance along the cable from the soma, x, normalized by λ; T = the normalized time, t/τ; and $\tau = R_{m-dend}C_m$.

The Laplace transform of the second-order partial differential equation is taken then to yield the second-order ordinary differential equation

$$\frac{d^2 \hat{V}}{dX^2} = (s+1)\hat{V}$$

where $\hat{V}$ = the Laplace transform of V.

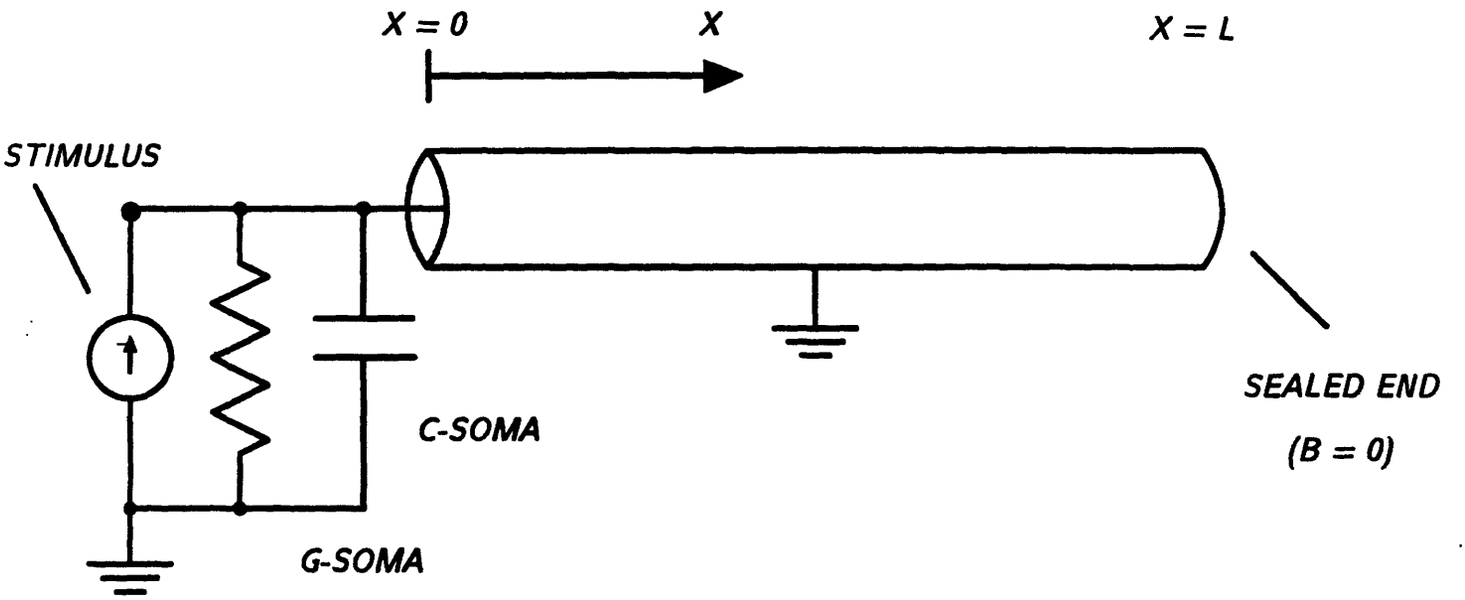


Figure 3.8: Circuit topology of a soma/short-cable structure. The structure is not drawn to scale.

The solution to this equation is

$$\hat{V} = \hat{A}e^{\sqrt{s+1}X} + \hat{B}e^{-\sqrt{s+1}X} \tag{3.11}$$

where $\hat{A}$ and $\hat{B}$ are constants that depend on the boundary conditions.

Now the Laplace transform of the axial current, $\hat{I}_a$, is equal to the change in $\hat{V}$ with X times r_a, where

$$r_a = \frac{R_i}{\pi a^2}$$

Thus

$$\hat{I}_a = \frac{-1}{r_a \lambda} \frac{d\hat{V}}{dX} \tag{3.12}$$

Note the inclusion of λ since X is the normalized distance. Solving for $\frac{d\hat{V}}{dX}$ in Equation 3.12 (using Equation 3.11), we obtain

$$\hat{I}_a = \frac{-\sqrt{s+1}}{r_a \lambda}(\hat{A}e^{\sqrt{s+1}X} - \hat{B}e^{-\sqrt{s+1}X})$$

The boundary conditions are set at $X = 0$ (at the soma), and $X = L$ (at the end of the cable). At the soma, the axial current I_a is equal to the sum of the soma currents –

$$\begin{aligned} \hat{I}_a(X=0) &= \hat{I}_{stimulus} - \hat{V}_{soma} G_{soma} - s\hat{V}_{soma} C' \\ &= \frac{-\sqrt{s+1}}{r_a \lambda}(\hat{A} - \hat{B}) \end{aligned}$$

where C' is the capacitance of the soma normalized by the dendrite time constant –

$$C' = \frac{C_m}{\tau_{dendrite} \frac{4}{3} \pi a_{soma}^2} \tag{3.13}$$

At the end of the cable since the terminating admittance = 0 then $\hat{I}_a = 0$, thus

$$\begin{aligned} \hat{I}_a(X=L) &= 0 \\ &= \frac{-\sqrt{s+1}}{r_a \lambda}(\hat{A}e^{\sqrt{s+1}L} - \hat{B}e^{-\sqrt{s+1}L}) \end{aligned}$$

so

$$\hat{A}e^{\sqrt{s+1}L} = \hat{B}e^{-\sqrt{s+1}L} \quad (3.14)$$

$$\hat{B} = \hat{A}e^{2\sqrt{s+1}L} \quad (3.15)$$

Now since

$$\begin{aligned} \hat{V}_{soma} &= \hat{V}(X=0) \\ &= \hat{A}+\hat{B} \end{aligned}$$

then

$$\hat{V}_{soma} = \hat{A}(1+e^{2\sqrt{s+1}L}) \quad (3.16)$$

Solving for $\hat{I}_{stimulus}$ –

$$\hat{I}_{stimulus} = \hat{A}(1+e^{2\sqrt{s+1}L})(G_{soma}+sC') + \hat{A}\frac{\sqrt{s+1}}{r_a\lambda}(e^{2\sqrt{s+1}L}-1) \quad (3.17)$$

Now we can find $\hat{A}$ and $\hat{B}$ from Equations 3.15 and 3.17 –

$$\hat{A} = \frac{\hat{I}_{stimulus}}{\frac{\sqrt{s+1}}{r_a\lambda}(e^{2\sqrt{s+1}L}-1)+(1+e^{2\sqrt{s+1}L})(G_{soma}+sC')} \quad (3.18)$$

$$\hat{B} = \frac{\hat{I}_{stimulus}e^{2\sqrt{s+1}L}}{\frac{\sqrt{s+1}}{r_a\lambda}(e^{2\sqrt{s+1}L}-1)+(1+e^{2\sqrt{s+1}L})(G_{soma}+sC')} \quad (3.19)$$

And finally from Equations 3.16, 3.18, and 3.19 we obtain

$$\hat{V}_{soma} = \frac{\hat{I}_{stimulus}(1+e^{2\sqrt{s+1}L})}{\frac{\sqrt{s+1}}{r_a\lambda}(e^{2\sqrt{s+1}L}-1)+(1+e^{2\sqrt{s+1}L})(G_{soma}+sC')}$$

which gives the expression for the input impedance as seen from the soma, $Z_{soma}(s)$ –

$$
\begin{aligned}
Z_{soma}(s) &= \frac{\hat{V}_{soma}}{\hat{I}_{stimulus}} \\
&= \frac{(1+e^{2\sqrt{s+1}L})}{\frac{\sqrt{s+1}}{r_a\lambda}(e^{2\sqrt{s+1}L}-1)+(1+e^{2\sqrt{s+1}L})(G_{soma}+sC')}
\end{aligned}
$$

In more compact form this is

$$
Z_{soma}(s) = \left(\frac{\sqrt{s+1}}{r_a\lambda}\tanh(\sqrt{s+1}L) + (G_{soma}+sC')\right)^{-1} \qquad (3.20)
$$

This expression for $Z_{soma}(s)$ was not amenable to attempts to perform an inverse transformation. However, when

$$
R_{m-soma} = R_{m-dend}
$$

that is for the case of a homogeneous membrane time constant, the expression for $Z_{soma}(s)$ simplifies somewhat and the inverse transform for this case has been derived. This is a rather complicated expression involving an infinite series, each term of which involves a product of exponential terms and a finite summation of the product of other exponential terms with parabolic cylinder functions ([27]).

Since an analytical expression for the inverse transform of the soma response could not be obtained, the response was analyzed in two ways – examining the frequency response directly and using DTFT techniques to estimate the temporal response (impulse response and step response). In order to evaluate the frequency response, the magnitude and phase of $Z_{soma}(s = j\tau\omega)$ were derived (note that the factor of τ is required because the Laplace transform was taken with respect to *normalized* time). So, from Equation 3.20 -

$$
Z_{soma}(j\omega) = \left[\frac{\sqrt{j\tau\omega+1}}{r_a\lambda}\left(\frac{1-e^{2\sqrt{j\tau\omega+1}L}}{1+e^{2\sqrt{j\tau\omega+1}L}}\right) + (G_{soma}+j\omega C)\right]^{-1} \qquad (3.21)
$$

The first step in this derivation was expressing the square root of $(j\tau\omega+1)$ in rectangular form -

$$\sqrt{j\tau\omega+1} = \sqrt{\frac{1+\sqrt{1+(\tau\omega)^2}}{2}} + j\sqrt{\frac{-1+\sqrt{1+(\tau\omega)^2}}{2}}$$

Letting

$$\eta = \sqrt{\frac{1+\sqrt{1+(\tau\omega)^2}}{2}}$$

$$\nu = \sqrt{\frac{-1+\sqrt{1+(\tau\omega)^2}}{2}}$$

the exponential terms may then be expanded –

$$\begin{aligned} e^{-2\sqrt{j\tau\omega+1}L} &= e^{-2L(\eta+j\nu)} \\ &= e^{-2L\eta}(\cos(2L\nu) - j\sin(2L\nu)) \end{aligned}$$

Equation 3.21 can be rearranged to give the real and imaginary parts of the frequency response –

$$\begin{aligned} Z(j\omega) &= \frac{\alpha+j\beta}{\delta+j\gamma} \\ &= \frac{\alpha\delta+\beta\gamma}{\delta^2+\gamma^2} + j\frac{(-\alpha\gamma+\beta\delta)}{\delta^2+\gamma^2} \end{aligned}$$

and thus

$$|Z(j\omega)| = \sqrt{\frac{\alpha^2+\beta^2}{\delta^2+\gamma^2}}$$

$$\text{Phase}(Z(j\omega)) = \text{atan}\left(\frac{-\alpha\gamma+\beta\delta}{\alpha\delta+\beta\gamma}\right)$$

where

$$\alpha = r_a\lambda[1+e^{-2L\eta}\cos(2L\nu)]$$

$$\beta = -r_a \lambda e^{-2L\eta} \sin(2L\nu)$$

$$\delta = \eta + r_a \lambda G_{soma} + e^{-2La}[-\eta \cos(2L\nu) - \nu \sin(2L\nu) + r_a \lambda G_{soma} \cos(2L\nu) + r_a \lambda \tau \omega C \sin(2L\nu)]$$

$$\gamma = \nu + r_a \lambda \tau \omega C + e^{-2La}[\eta \sin(2L\nu) - \nu \cos(2L\nu) + r_a \lambda \tau \omega C \cos(2L\nu) - r_a \lambda G_{soma} \sin(2L\nu)]$$

These formulas were used to see how varying some specific parameters while keeping the remainder constant changed the frequency response. In particular, these results were used in investigating how parameters that are derived from the transient response are affected when the directly measured parameters are kept fixed and some other derived parameter is varied .

To summarize the results so far, I have proposed that the following parameters are either known with a fair degree of assurance, or may be estimated: C_m, R_i, a_{soma}, R_{in}, τ_0, B, and a limited range for a and l of the equivalent dendritic cable. On the other hand, I have shown that the reported values for R_m are inconsistent with the other data available for these cells, and in fact the soma and the dendrites may be approximated as having distinct membrane resistivities.

The problem of estimating the geometry of the model is therefore determined by the following constraints – estimate for the soma radius, estimate for the range of cable diameters, estimate for the cable length, input resistance, observed time constant, estimate of membrane capacitance, and the estimate of cytoplasmic resistivity. The free parameters then include R_{m-soma}, R_{m-dend},and a. The results of this estimation will be presented in the next section.

3.12 Simulating the Step Response of the Brown et. al. Geometry with Alternative Models

Once the frequency response of the general soma/short-cable model was derived, I attempted to find different values for the membrane resistivities and the cytoplasmic resistivity that would yield step responses similar to that of the model derived from the Brown et. al. parameters.

The constraints included $a_{soma} = 17\mu$m, R_i, which was set at 200 Ω cm, $\tau_0 = 19$ms. $R_{in} = 39\text{M}\Omega$, and $C_m = 1.0\mu\text{f/cm}^2$. R_{m-dend} was then set at

either 30, 40, or 50 kΩ cm, and a, was set at either 5, 6, or 7 μm. For a given combination of R_{m-dend} and a both R_{m-soma} and l were then varied until all the above constraints were met. For this analysis the electrode shunt resistance was not specified, thus R_{soma} reflected both the leak conductance of the soma and the electrode shunt resistance.

The step response and the frequency response of the resulting structure were then compared with that derived from the Brown and Perkel model. The parameters were adjusted under these constraints to derive several structures whose time response was consistent with the data. The complete parameters for these structure are listed in Table 3.1. The responses of these structures were clustered into three groups, each group characterized by a common value for a. The responses for structures B, C, and D were almost identical to each other, as were the responses of structures E and F. Therefore, the analysis suggests that the diameter of the dendrite cable was the most sensitive parameter in determining the linear response of the soma-cable structure.

For the majority of the simulations, including all those presented in this thesis, version "C" of the model structures was chosen as representative of the family of model structures. In this case the value for R_{m-dend} of 40 KΩ cm^2 is a similar to the value for R_{m-dend} (approximately 40 KΩ cm^2) estimated by Shelton for Purkinje cells, and is much higher than the values of R_m that are quoted consistently in reports on hippocampal pyramidal cells. Also, this model has an R_i of 200 Ω cm, which is also in line with the value of R_i estimated by Shelton, as described earlier. The value of a (6.0μm) and l (1200μm) is consistent with the values estimated earlier in the chapter (Section 3.8.2).

The step responses for structures A, C, and F and that derived from the Brown et. al. data are shown in Figure 3.9. An expanded view of these response is shown in Figure 3.10. In this figure the effect of the smaller soma time constant in the model structures is seen as the response of these structures initially decay much faster than the reference structure. However, as shown in Figure 3.11, all four responses eventually settle into a single exponential decay with the same time constant of 19 ms. The magnitude of the frequency responses for the same structures are shown in Figure 3.13, and the phase of the frequency responses for these structures are shown in Figure 3.12.

There are several interesting features of these simulations. The first is that the values of ρ and L vary greatly for the different structures – between

Model	$a\,(\mu\text{m})$	$R_{m-dend}\,(\text{K}\Omega\,\text{cm}^2)$	$R_{soma}\,(\Omega\,\text{cm}^2)$	$l\,(\mu\text{m})$	$\lambda\,(\mu\text{m})$	L	ρ
A	5.0	50	720	1350	2500	0.54	0.43
B	6.0	30	1100	1200	2121	0.57	1.2
C	6.0	40	850	1200	2450	0.49	0.69
D	6.0	50	750	1200	2738	0.44	0.50
E	7.0	40	870	1050	2646	0.40	0.74
F	7.0	50	760	1050	2958	0.35	0.53

Table 3.1: Parameters of model structures derived to match the τ_0 (19 ms) and R_{in} (39 MΩ) of the Brown et. al. data, with $R_i = 200\,\Omega\,\text{cm}$, $a_{soma} = 17\,\mu\text{m}$, and $C_m = 1.0\,\mu\text{f/cm}^2$. The values listed for structure "C" were chosen for the model.

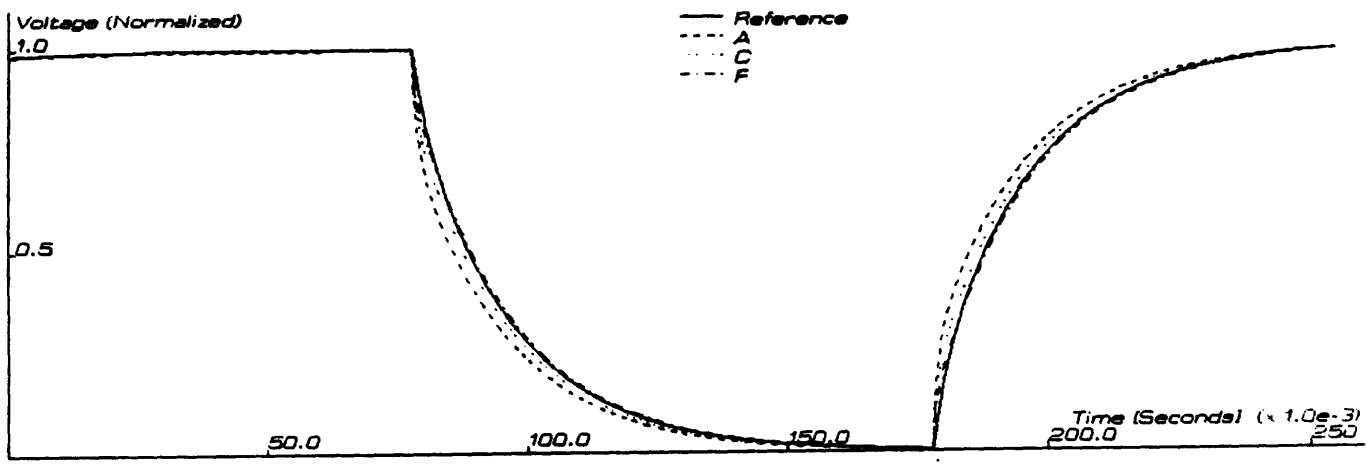


Figure 3.9: Normalized response to injection of somatic current step for Brown et. al. structure , and representative alternative structures (A, C, and F, ref. Table 3.1) consistent with histological measurements.

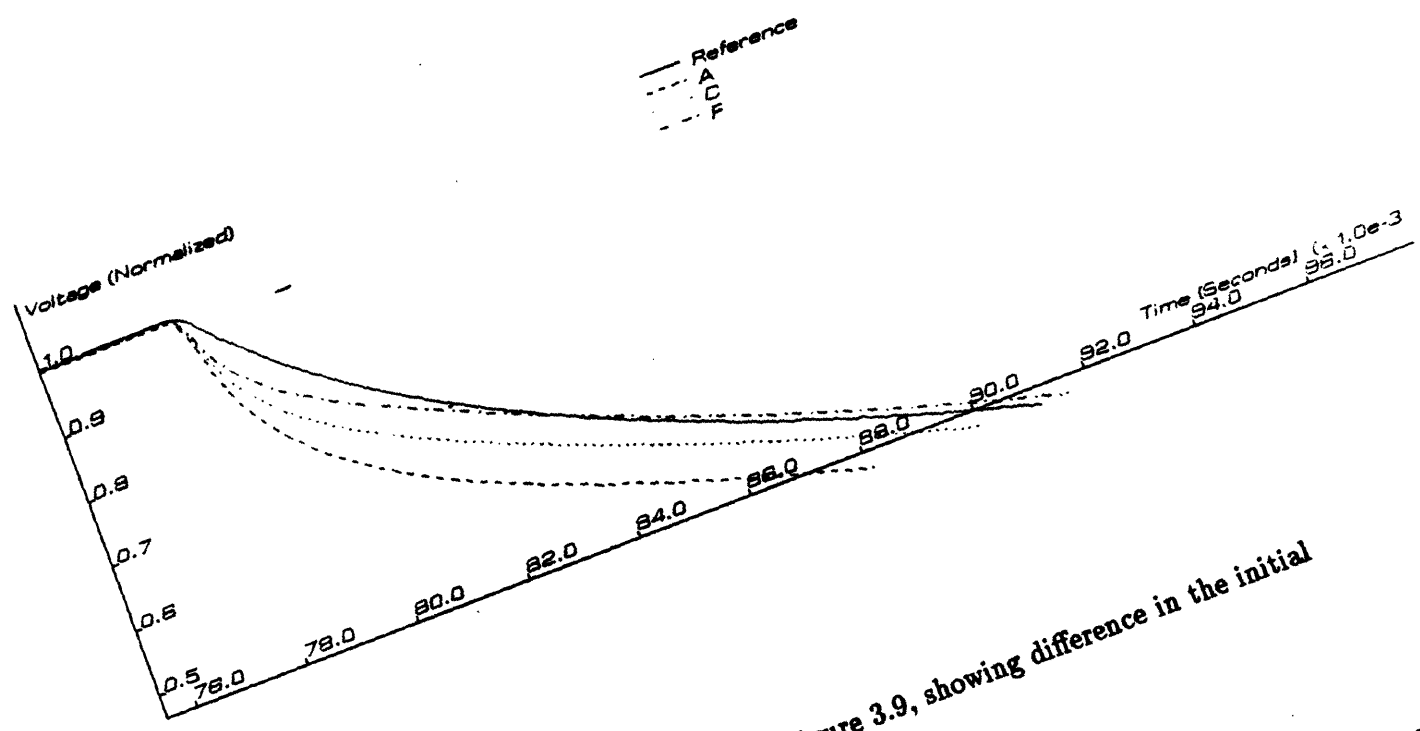


Figure 3.10: Expanded view of Figure 3.9, showing difference in the initial part of the somatic current step responses.

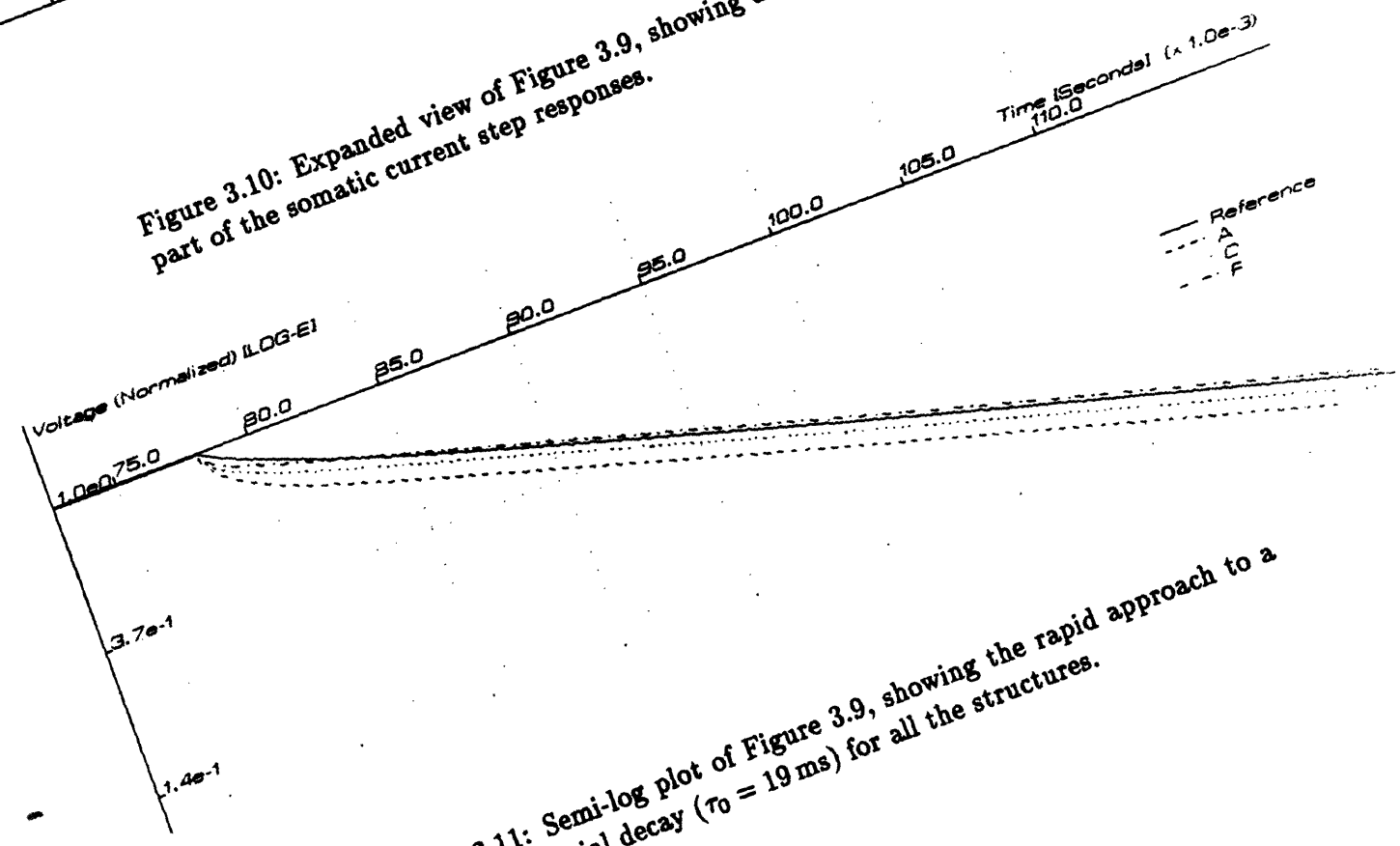


Figure 3.11: Semi-log plot of Figure 3.9, showing the rapid approach to a single exponential decay ($\tau_0 = 19$ ms) for all the structures.

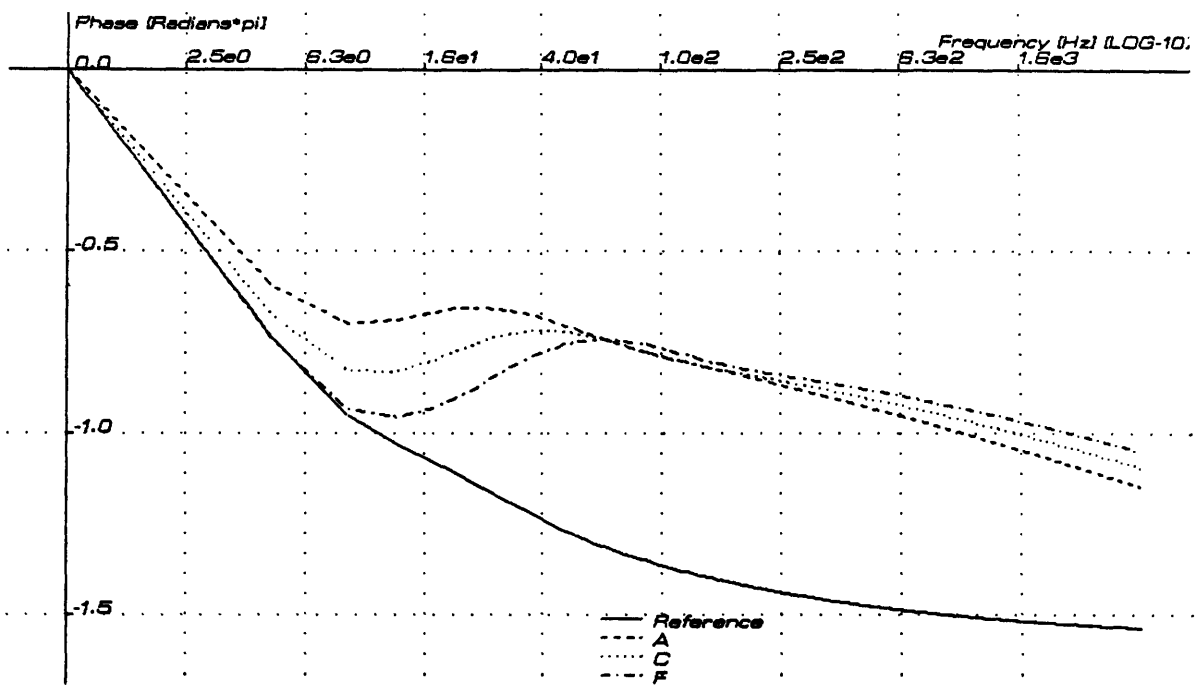


Figure 3.12: Phase of frequency response for Brown et. al. structure , and representative alternative structures (A, C, and F, ref. Table 3.1) consistent with histological measurements.

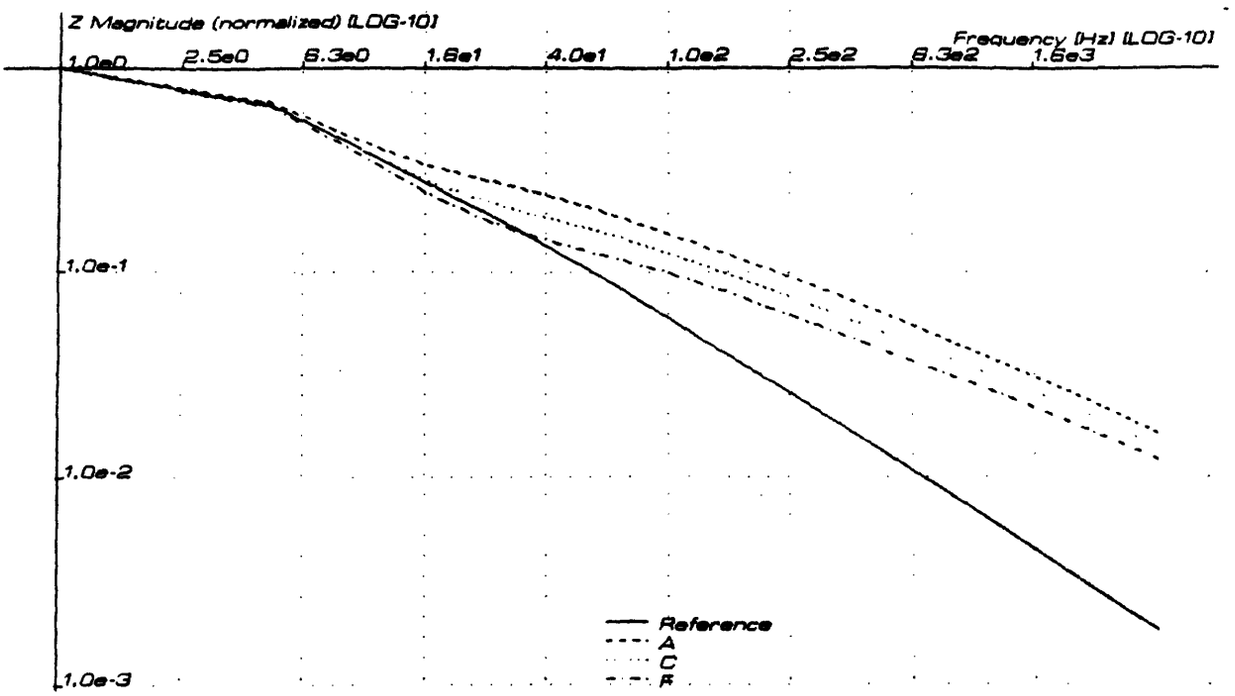


Figure 3.13: Normalized magnitude of frequency response for for Brown et. al. structure , and representative alternative structures (A, C, and F, ref. Table 3.1) consistent with histological measurements.

1.2 and .43 for ρ and between .35 and .57 for L. This shows that the methods typically used for estimating ρ and L are not reliable *unless* the cell has a homogeneous membrane time constant.

The most distinctive difference in the characteristics of the structures with a non-homogeneous membrane time constant and the structure based on the Brown et. al. data is in the phase of the frequency response. For all the structures with non-homogeneous R_m the phase deviates from that of the structure with homogeneous R_m at a frequency of about 100Hz. This difference does not manifest itself strongly on the temporal responses, however, because of the attenuated response above 100Hz.

The characteristics of the phase response for the simulated cell structures suggest that evaluation of the linear parameters discussed in this chapter may be better served by analyzing the frequency response of the cells under protocols that ensure a linear response. Since the interesting part of the phase response occurs at frequencies where the cell impedance is relatively small, spectral estimation using averaging techniques or white-noise approaches may be applicable.

The values for R_{soma} and $R_{dendrite}$ differ by about two orders of magnitude in all the derived structures. If the contribution of an electrode shunt is considered, this difference is reduced, but by only a factor of about two since the typical soma resistance (including the electrode shunt resistance) is around 70 MΩ and the electrode shunt resistance is about 100 MΩ as estimated earlier.

In summary, there are many versions of the soma/short-cable model that can give the same τ_0 and R_{in} with differences in the distribution of R_m between soma and dendrite, and realistic variations in a. Examination of the frequency response indicates that this measurement may provide a way to better estimate the electrotonic parameters of these cells, particularly under the assumption of non-homogeneous membrane resistivity. While the magnitude of the frequency responses for the various structures are rather similar, the *phase* of the frequency responses differ markedly, and this metric may be usefully exploited in order to better estimate the linear parameters of the soma/short-cable model.

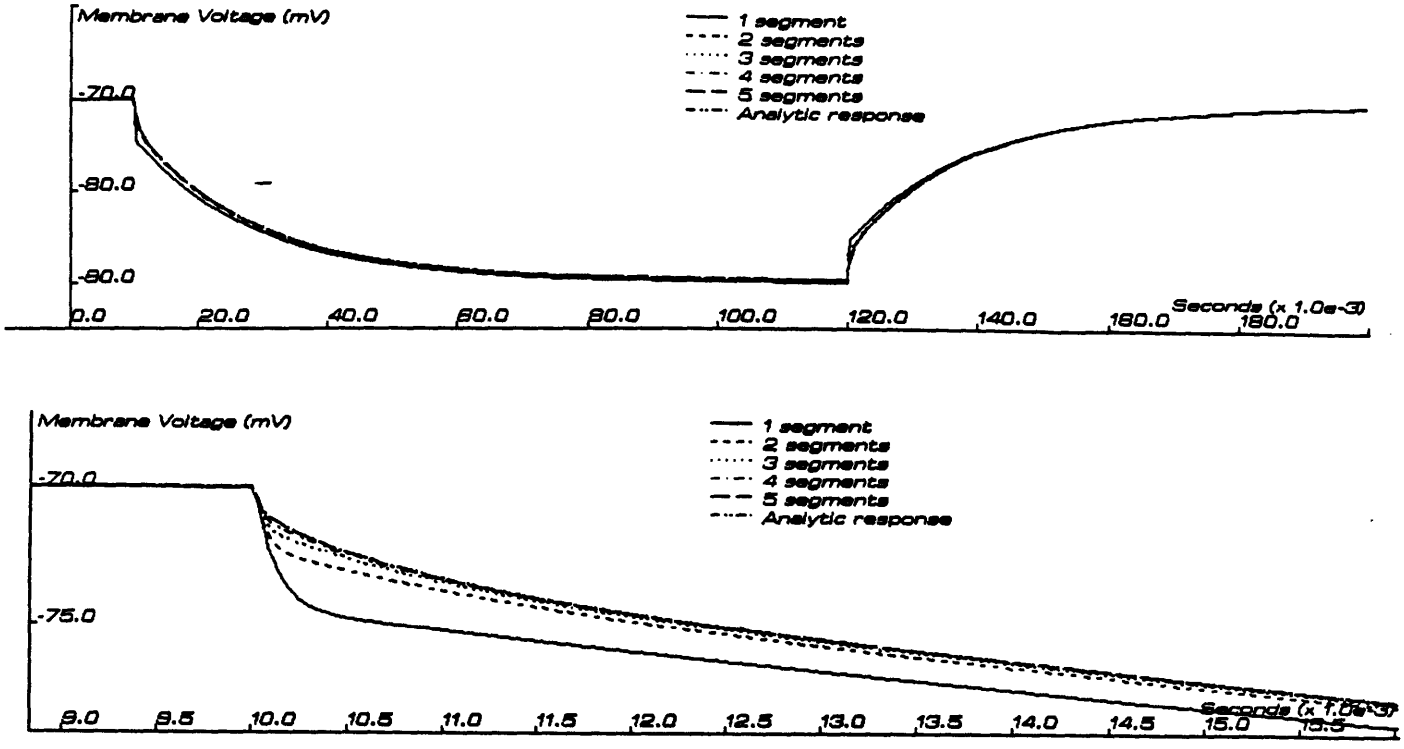


Figure 3.14: Step response of Soma-Cable Structure a) Inverse FFT of Analytic Solution b) Model with 1 segment c) Model with 2 segments d) Model with 3 segments e) Model with 4 segments f) Model with 5 segments

3.13 Discrete (Lumped) Approximation To Dendritic Cables and Comparison Of HIPPO Results To Analytical Solution Of Linear Cable - Dependence Of Segment Dimensions

Once the response to a current step of the soma/short-cable structure was derived from the inverse DFT of the analytical frequency response, the compartmental approximation of the cable was evaluated by comparing the model's response in current clamp simulations to the estimated response of the continuous cable. In Figure 3.14 the response of the model with different numbers of compartments is compared to the estimated response. As can be seen in the figure, the response of the model with 5 segments is in very good agreement with the estimated response.

3.14 Summary of Results from the Determination of Electronic Structure

To recapitulate, the HIPPO model electrotonic parameters are as follows:

- $a_{soma} = 17\mu$m
- $a = 6.0\mu$m
- $l = 1200\mu$m
- $R_{shunt} = 100\,\mathrm{M}\Omega$
- $R_{m-soma} = 850\,\Omega\,\mathrm{cm}^2$
- $R_{m-dend} = 40\,\mathrm{K}\Omega\,\mathrm{cm}^2$
- $R_i = 200\,\Omega$cm
- $C_m = 1.0\mu\mathrm{f/cm}^2$
- $B = 0$
- $\lambda = 2450\,\mu$m
- $L = .49$
- $\rho = .69$
- $\tau_0 = 19$ milliseconds
- $R_{in} = 39\,\mathrm{M}\Omega$
- $E_{leak} = -70$ millivolts

3.15 Is It Important to Capture Dendritic Morphometric Characteristics?

The results described here tend to confirm that the actual geometry of the dendritic tree may not in itself be critical to somatic response. For example, the Rall reduction is reasonably accurate even if the constraints specified in this algorithm are not met exactly. What is very important, however, is the various parameters that characterize the tree (or its equivalent single cable) as a whole, that is as a lumped element (cable). This result has been reported elsewhere ([51]). Specifically, the equivalent cylinder approximation works well even when the constraints on subsequent cable diameters and conservation of electrotonic length are not met exactly. The parameters characterizing that cylinder are important to the electrical load as seen by the soma, however, and can have a large effect on the processing of information that occurs there.

As shown, the assumption of a homogeneous membrane time constant allows the construction of a soma/short-cable approximation of the pyramidal cell whose linear response closely matches that of the real cell. On the

other hand, this construction is inconsistent with histological measurements, and a structure with a non-homogeneous membrane time constant can be proposed which successfully addresses these problems.

Although the two constructions yield models with distinct frequency responses, the significant differences occur at frequencies that are substantially attenuated in both structures, so that the step responses are rather similar. Why, then, is it important to revise the earlier model with the homogeneous membrane time constant? As shown, although the somatic responses of the two models are similar, the values of ρ and L are very different. This is important when considering the role of the dendritic tree in integrating synaptic input. In particular, the smaller L that has been suggested in the present study indicates that the dendritic tree is more electrically compact that previously thought. In functional terms, this means that there is less distinction, from the point of view of the soma, between distal and proximal dendritic input. This could enhance the computational flexibility of the dendritic tree since a fundamental limit such as linear attenuation of EPSPs and (possibly) IPSPs will be reduced by the smaller L, and selective enabling/disabling of various sections of the tree could be accomplished more effectively by non-linear mechanisms (e.g. other synapses).

The effect of the lower ρ that is indicated in the present study is to reduce the burden on the somatic conductances imposed by the dendritic load, for example during the spike depolarization and repolarization.

Chapter 4

APPLYING THE HODGKIN-HUXLEY (HH) MODEL OF IONIC CHANNELS TO PUTATIVE HIPPOCAMPAL CURRENTS

4.1 Introduction

An extension of the Hodgkin-Huxley (HH) model of ionic channels in the squid axon ([21],[20],[22],[23]) is the foundation for the description of the hippocampal pyramidal cell ion channels that are used in the model. This comes about in two ways. First, many of the currents that have been described in the literature have been fitted to HH-like models to start with. Second, when this model has been used either to augment sparse voltage clamp data on a particular current or to propose currents whose existence is defended purely on phenomenological grounds, these currents have been constructed using HH-like mechanisms. Examining the HH model in detail is therefore important in order to establish some of the key assumptions in

the HIPPO model.

4.2 Background of the HH Model

In the early 1950's Hodgkin and Huxley postulated that the electrical activity of the squid axon was due to two time-dependent non-linear conductances, one of which was specific to Na^+ ions and another which was specific to K^+ ions. This result was based of data obtained with the newly-developed voltage clamp method for measuring electrical properties of non-linear membranes. Using the voltage clamp protocol and various manipulations, including replacement of the $NaCl$ in the external medium with choline chloride to eliminate the Na^+ current, Hodgkin and Huxley measured the time-constants and the steady state values for the two conductances as a function of the membrane voltage.

Noting the sigmoidal characteristic of the activation of the Na^+ channel as the membrane was depolarized, and the fact that the channel inactivated a short time after it was activated, a model for the Na^+ channel was derived that included four "gating" particles (three so-called m activation particles and one h inactivation particle). These particles can be thought of as distinct regions of the channel protein, each of which can be in one of two stable conformations or states, conducting (open) or non-conducting (closed). For a given channel to conduct, all of its gating particles must be in the open state. The macroscopic conductance of the Na^+ channel, g_{Na}, was expressed as

$$g_{Na} = m^3 h \bar{g}_{Na}$$

where $0 < m, h < 1$ and $\bar{g}_{Na}$ is the maximum conductance for the ensemble of Na^+ channels in the membrane. Hodgkin and Huxley determined that the transition between states is governed by first order kinetics, and the rate constants for this transition are functions of voltage, as will be described later. The likelihood of whether a given particle will be open or closed is therefore also a function of voltage.

The sigmoidal activation characteristic under voltage clamp arises from the third power of the m gating particle. This number was determined by Hodgkin and Huxley by fitting powers of exponential relaxations to the observed kinetics. In a similar manner, the macroscopic conductance of the K^+ channel in the squid axon was described as being determined by four

gating particles, n. The macroscopic conductance of the K^+ channel was expressed as -

$$g_K = n^4 \bar{g}_K$$

where $\bar{g}_K$ is the absolute conductance for the ensemble of K^+ channels in the membrane.

The transient behavior of the Na^+ channel during excitation of the neuron, through its activation and subsequent inactivation, was explained by the voltage dependencies of m and h, and the different voltage-dependent functions for the time constants of m and h. The steady state value of m is a monotonically increasing function of the membrane voltage, while the steady state value of h is a monotonically decreasing function of the membrane voltage. In addition, the time constant for m is smaller than the time constant for h at a given voltage. The result is that on depolarization m will adapt to its (more open) steady state value quickly while h will lag behind in its (more open) hyperpolarized steady state. The channel will begin to conduct with the increase of m. In a short time, however, h will relax to its (more closed) steady-state value at the new (depolarized) membrane voltage. Even though the three m "particles" are in the open state, the subsequent closing of the single h "particle" will shut the channel down and turn off the Na^+ current .

Once Hodgkin and Huxley had a description of these two non-linear conductances and the linear parameters of the cell, they were able to numerically reconstruct the action potential in the squid axon. In the model of the hippocampal pyramidal neuron, several distinct currents, mediated by different ions, are described using variants on the HH model theme.

4.3 Extension of the HH Description to Pyramidal Hippocampal Cells

The Hodgkin and Huxley model approach can be extended[1] to describe some of the currents found in other electrically non-linear cells. Analysis of other currents is undertaken here under the assumption that they are based on mechanisms which undergo first-order kinetic transformations between conducting states and non-conducting states. By both qualitative and quantitative analysis, plausible mechanisms underlying non-linear currents

[1]The notion that this report "extends" on the HH model is discussed in Section 4.8.

may be deduced. These descriptions are typically based initially on voltage clamp data. As will be explained shortly, this protocol can measure the time constants and steady state values for the kinetic events controlling the conductances behind these currents, assuming that, indeed, such a kinetic description is valid.

4.3.1 The Voltage-Dependent First-Order Kinetics of HH-like Conductances

To recapitulate, in the HH model each current in the electrically-active cell is assumed to correspond to a specific type of ionic channel, which in turn is comprised of a protein conglomeration that traverses the membrane. In each channel ions travel through a luminal trans-membrane aqueous phase, driven by the driving potential for the channel. As reviewed in Chapter 2, this driving potential is a function of the membrane voltage and the trans-membrane concentration gradient of the carrier for that conductance, according to the Nernst equation or the Nernst-Goldman equation.

The transitions of the particles between states are governed by first-order kinetics. Each state or conformation corresponds to a free-energy well, with a single high-energy rate-determining barrier between the two states. Movement of the gating particles between states is assumed to be accompanied by a movement of charge, causing the state-transition kinetics to be dependent on the membrane voltage. These gating particles are regions of the protein that (a) can reversibly mediate the conductivity of the channel, possibly via steric factors, and (b) have a sufficient dipole moment and freedom of movement so that they may act as voltage-sensors, changing the conformation of the protein or protein group as a function of the electric field across the channel. The magnitude of the voltage dependence is derived from the Boltzmann equation which specifies the probabilities of state occupancies according to the free energies of the states.

In practice, voltage clamp protocols, in which the membrane relaxation currents are measured as the cell membrane is "clamped" at different potentials with a microelectrode, are used to measure the kinetics of the various currents. This technique assumes that the kinetics of different currents can be measured independently, either because different currents are activated over non-overlapping membrane voltages, because the time courses are distinct, or because the currents have distinct pharmocological sensitivities. Implicit in this approach is the assumption that different currents interact only through the membrane voltage. In fact, in the case of currents which

are dependent on the movement of Ca^{++} into the cell simple voltage-clamp measurements may give misleading results. Whatever independence exists between the different currents is exploited by the electrophysiologist as he devises protocols for intracellular measurements.

The macroscopic conductance of given type of channel is determined by the proportion of channels in the open state, the conductance of a single channel, and the total number of channels of that type in the membrane. For example, if the conductance of some channel Y is controlled by a single gating particle, and the proportion of open gating particles is x, then the macroscopic conductance of that channel type is expressed by

$$g_Y = x \cdot \bar{g}_Y$$

where g_Y is the actual conductance for the channel current I_Y, and $\bar{g}_Y$ is the maximum conductance for that current. The factor x is equivalent to the probability that the gating particle for a single channel will be in the open state. As will be shown, x is both a function of the membrane voltage and of time.

The macroscopic voltage- and time-dependence of the channel conductance arises from the first-order kinetics that the gating particles obey in their transition between their open and closed states.

$$\text{closed}\,(1-x) \overset{\alpha,\beta}{=} \text{open}\,(x)$$

Here x represents the fraction of channels in the open state, and $1 - x$ represents the fraction of channels in the closed state. α and β are the forward and backward rate constants for the reaction, respectively. This relationship yields the simple differential equation relating the derivative of $x(t)$, $\dot{x}(t)$, with the steady state value of x, x_∞, and the time constant for the reaction, τ_x -

$$\dot{x}(t) = \frac{x_\infty - x(t)}{\tau_x}$$

where x_∞ and τ_x can be expressed in terms of the rate constants α and β -

$$x_\infty = \frac{\alpha}{\alpha + \beta}$$

$$\tau_x = \frac{1}{\alpha + \beta}$$

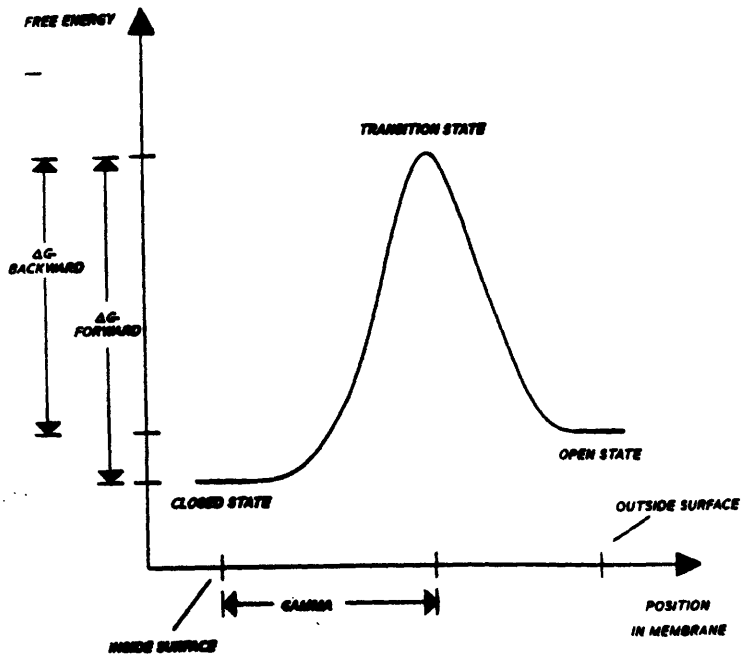


Figure 4.1: Energy diagram for gating particle states with no applied membrane voltage. The stable open and closed states correspond to the low-energy wells. The high-energy transition state is the rate-limiting step. γ, which is the relative position of the transition state within the membrane, can be between 0 and 1.

As will be discussed later in the sections on the various non-linear currents in the model, in the literature current kinetics are occasionally determined empirically in terms of an "α - β" type formulation. For most currents, however, the voltage dependence of x_∞ and τ_x is the figure that is reported.

The energy profile for a gating particle in the single barrier model is shown in Figure 4.1.

As mentioned earlier, the rate constants for the transitions from one side of the reaction coordinate to the other is given by the Boltzmann equation, which is a function of the difference between the energy of the rate-limiting step and the initial state. The expression for the forward rate constant in the absence of an applied voltage, α_0, is -

$$\alpha_0 = C e^{\frac{\Delta G}{RT}} \tag{4.1}$$

where ΔG is the free-energy difference between the closed state and the

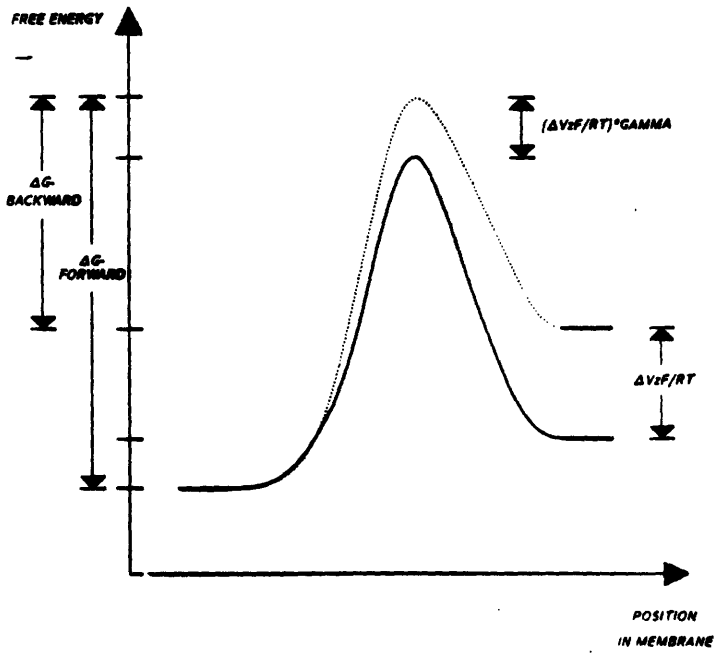


Figure 4.2: Energy diagram with applied voltage ΔV across the membrane.

transition state. R is the gas constant, T is the absolute temperature, and C is a constant.

The voltage-dependence of the kinetics arises from the distortion of the above energy diagram when a voltage is applied across the membrane, as shown in Figure 4.2. The applied voltage changes the difference in free energy between the stable states and the transition state. The effect of the voltage is reflected in the expression for the rate constants as follows -

$$\alpha = \alpha_0 \epsilon^{z\gamma\Delta\xi} \tag{4.2}$$

$$\beta = \beta_0 \epsilon^{-z(1-\gamma)\Delta\xi} \tag{4.3}$$

where

$$\Delta\xi = \frac{\Delta V F}{RT} \tag{4.4}$$

z is the effective valence of the gating particle, and γ is the position of the transition state within the membrane, normalized to the membrane thickness. ΔV is equal to $V - V_{\frac{1}{2}}$, where V is the membrane voltage and $V_{\frac{1}{2}}$ is

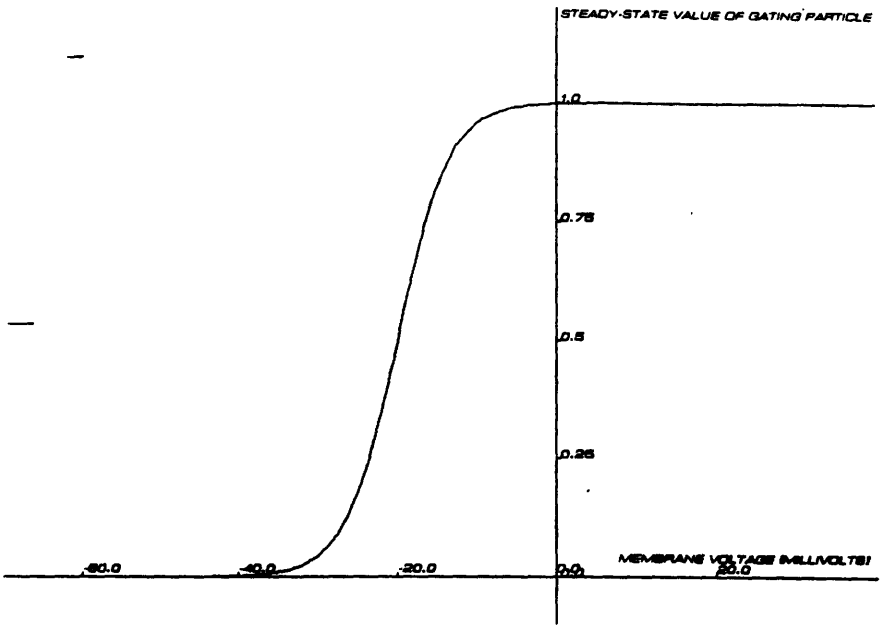


Figure 4.3: The steady-state (x_∞) curve for the hypothetical gating particle x. $z = 8$. $V_{\frac{1}{2}} = -20mV$.

the membrane voltage at which α equals α_0 and β equals β_0 ([26]). F is Faraday's constant. Normally, α_0 and β_0 are taken as equal. This can be reconciled with the different energies of the stable states as shown in Figure 4.2 by adjusting $V_{\frac{1}{2}}$.

Since the backward and forward rate constants are functions of the membrane voltage, the values for the time constant and the mean steady-state (from now on referred to as the steady-state curve) are also functions of voltage. The resulting expression for steady-state curve is a sigmoidal function -

$$x_\infty = \frac{1}{1 + e^{-z\Delta\xi}} \tag{4.5}$$

This type of characteristic is shown in Figure 4.3.

The expression for the time constant is a skewed bell-shaped function of the voltage -

$$\tau_x = \frac{e^{-z\gamma\Delta\xi}}{\alpha_0 + \beta_0 e^{-z\Delta\xi}} \tag{4.6}$$

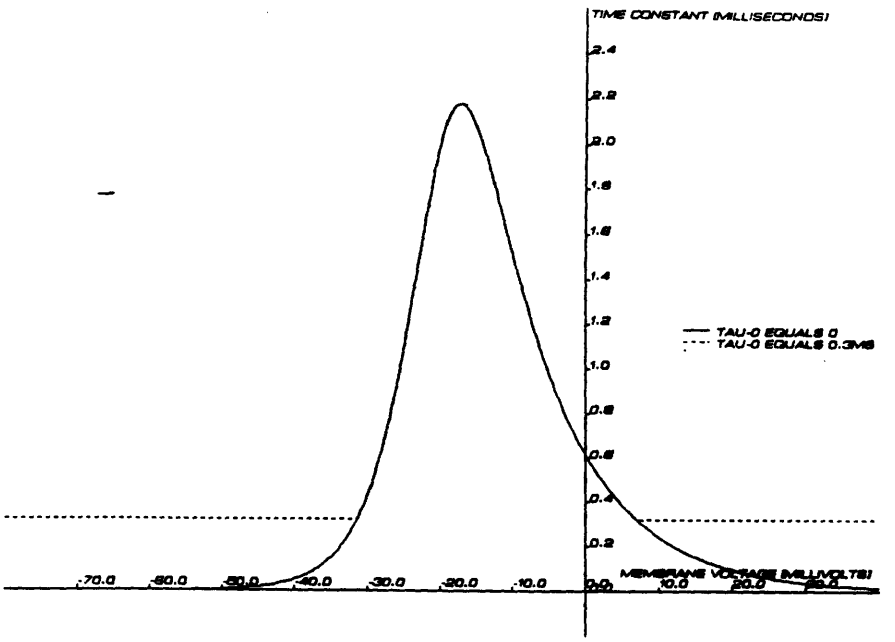


Figure 4.4: The time constant (τ_x) curve for the gating particle x, with $\tau_0 = 0$ and 0.3milliseconds. $z = 8$. $V_{\frac{1}{2}} = -20mV$. $\gamma = 0.3$.

An example of such a function for the time constant is shown in Figure 4.4. As referred to earlier, including an additional assumption of a linear rate-limiting mechanism on the gating particle transition was useful. For example, drag on the gating portion of the protein as it changes conformation will place an upper limit on the rate constants of the gating transitions. As the rate constants defined by the Boltzmann equation increase exponentially with voltage, an assumption was made that at some point other intrinsic aspects of the channel protein would prevent an arbitrarily fast transition. For the simulations this factor was, as a first approximation, taken as a specific constant minimum value for the time constant, τ_0, for each of the current's gating particles. This is illustrated in Figure 4.4.

4.3.2 Activation and Inactivation Gating Particles

There are two types of gating particles: *activation* gating particles (activation variable) and *inactivation* gating particles (inactivation variable). The steady state curve for an activation particle increases with depolarization; the steady state curve for an inactivation particle decreases with depolar-

ization. This characteristic is determined by the sign of z – positive for an activation particle and negative for an inactivation particle. The activation and inactivation particles therefore have opposite effects on the channel conductance with depolarization – the activation particle opens on depolarization and the inactivation particle closes on depolarization.

4.3.3 Transient and Persistent Channels

The type of gating particles in a channel determine whether it is a *transient* channel or a *persistent* channel. A persistent channel has only activation particles; this type of channel will stay open upon prolonged depolarization. A transient channel, on the other hand, is only open for a limited time upon depolarization; a typical scenario is that upon depolarization the (typically faster) activation particles relax to their open state and thus, along with the already open (because of the lower holding potential) inactivation particles, the channel conducts. After some delay the slower inactivation particles relax to their closed position at the depolarized level, and thus close the channel.

4.3.4 Activation/De-inactivation and Inactivation/Deactivation

Recall that for a given channel to conduct, *all* of its gating particles must be in the open position, regardless of whether they are classified as activation or inactivation particles. When describing the change of the conductance state of a channel, then, some clarification of nomenclature is useful. When a channel goes into the conducting state because of the movement of an activation particle into its open position (state), then the process is called *activation*. When a channel goes into the conducting state because of the movement of an inactivation particle into its open position, then the process is called *de-inactivation*. When a channel goes into the non-conducting state because of the movement of an inactivation particle into its closed position, then the process is called *inactivation*. And finally, when a channel goes into the non-conducting state because of the movement of an activation particle into its closed position, then the process is called *deactivation*.

4.4 Fitting the HH Parameters to Putative Current Kinetics

Fitting the HH model to the behavior of a given current began under the assumption that the channel responsible for the current had only one or two types of gating particles - either there was a single activation particle or there was an activation particle with an inactivation particle. The number of any given particle in a single channel was constrained to be at the most four, but in practice the inclusion of more than four duplicate gating particles had little effect on the overall kinetics of a channel.

The first step in formulating the expression for a given current was to determine its activation/deactivation and/or inactivation/de-inactivation properties. By examining voltage clamp and/or current clamp records, the relevant questions were as follows:

1. Does the conductance in question increase on depolarization, independent of factors such as Ca^{2+} entry? If so, then the conductance is likely controlled by at least one activation particle.

2. Is the conductance transient, i.e. is the conductance removed after activation without repolarization? If so, then the conductance is likely mediated by at least one inactivation particle.

3. Is there any relationship between the activation of Ca^{2+} and the presence of the current in question? If so, the possibility that such a relationship may mimic or mask voltage-dependent activation or inactivation must be considered.

Once the basic type of particles that govern the channel were determined, it remained to estimate the parameters for each particle. For each gating particle (for each current) the free parameters included:

- $V_{\frac{1}{2}}$ - the voltage at which α and β are equal
- γ - a measure of the symmetry of the system ($0 < \gamma < 1$)
- z - the effective valence of the gating particle (typically z ranged from 3 to 30)
- α_0 - the forward rate constant when $V = V_{\frac{1}{2}}$

- β_0 - the backward rate constant when $V = V_{\frac{1}{2}}$
- τ_0 - the minimum time constant of the gating particle (typically τ_0 ranged from 0.5 milliseconds to 4.0 milliseconds)

The first step in fitting the parameters was to adjust the steady-state activation and inactivation curves according to the available data. For some currents there was more or less complete voltage clamp measurements of these curves, while for others only the steady-state conductance as a function of voltage was available (ref. Na^+ currents, Chapter x). Note that in the latter case, if the current in question is transient then the steady-state conductance will be a measure of the product of the some power of an activation variable and some power of an inactivation variable (the window current). As referred to earlier, α_0 was taken as equal to β_0 in the estimations of current kinetics, with no loss of generality.

Adjusting the steady-state curve for a gating particle is straightforward. $V_{\frac{1}{2}}$ is simply the voltage where the steady-state curve is equal to 0.5, as implied in Equation 4.5 and shown in Figure 4.3. Once $V_{\frac{1}{2}}$ is estimated, z is then adjusted to set the steepness of the steady-state curve as required.

Unless good measurements on the time constant for a current were available, manipulating the remaining parameters to yield different functions of the time constant was often a tricky procedure. The data for each current gave different clues as to the form of this function, and for some currents it was not possible to derive a unique function until the current was re-evaluated in light of modified description of another current. In a few cases, however, a particular function for a particular variable turned out to be not critical (e.g. the y gating particle for I_{DR}, whose time constant only had to be greater than some value, irregarless of voltage).

4.4.1 Effect of Gating Particle Valence

Observing how the variation in the free parameters affects the steady-state and time constant curves is instructive since this process was integral to the development of the conductance mechanisms. Figures 4.5 and 4.6 illustrate how the different values of z change the steepness of the steady-state curve and the sharpness of the time constant curve.

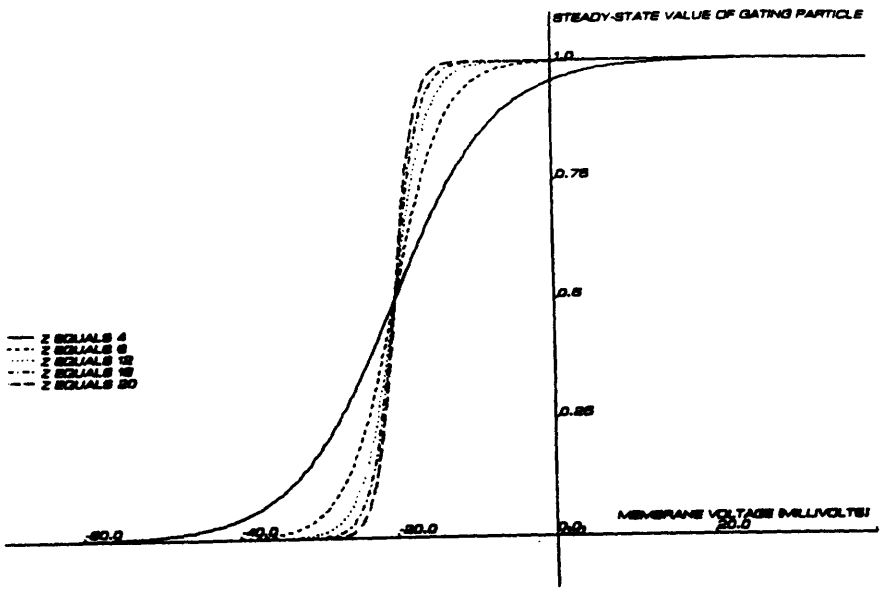


Figure 4.5: Effect of the valence, z, of the gating particle, x, on the x_∞ curve. Note that this is an activation gating particle. $V_{\frac{1}{2}} = -20mV$.

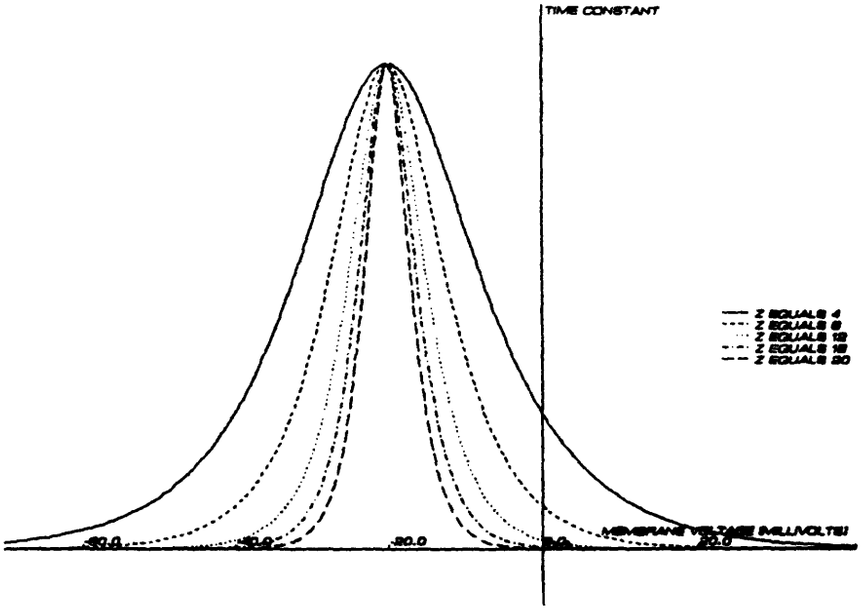


Figure 4.6: Effect of the valence, z, of the gating particle, x, on the τ_x curve. $V_{\frac{1}{2}} = -20mV$. The time scale is arbitrary since it is linearly scaled by α_0. τ_0 is set to 0. and γ is set to 0.5.

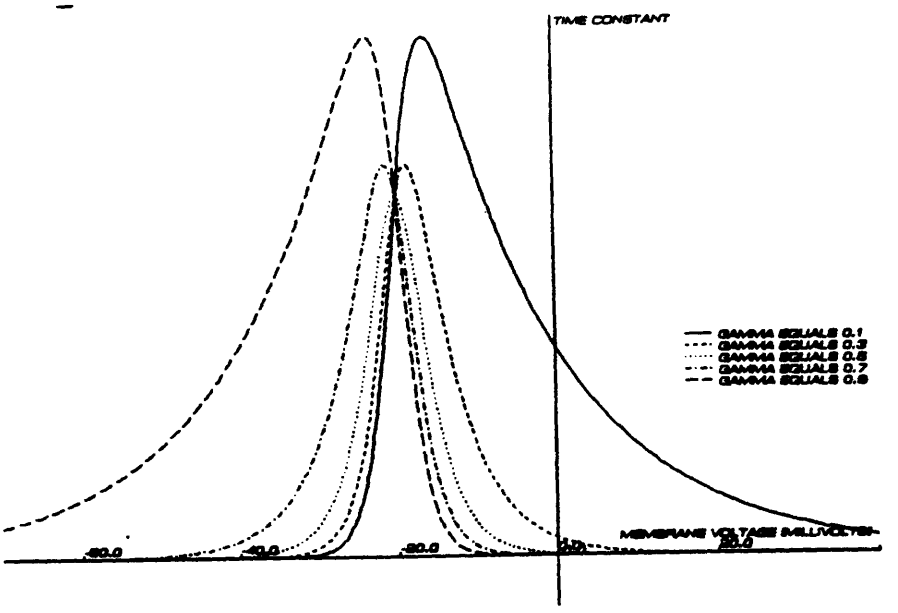


Figure 4.7: Effect of the relative position, γ, of the transition state within the membrane for the gating particle, x, on the τ_x curve. $z = 16$ and $V_{\frac{1}{2}} = -20mV$. The time scale is arbitrary, as in Figure x. τ_0 is set to 0.

4.4.2 Effect of Gating Particle Symmetry

Figure 4.7 illustrates how the symmetry of the system, as specified by γ, affects the curve for the time constant for $z = 16$. Extreme values of γ (i.e. close to 1 or 0) cause the time constant to change abruptly at some voltage so, to a first approximation, as a function of voltage the time constant is either large or relatively small. This sort of characteristic was used to advantage in constructing some of the current kinetics. For example, as will be outlined in Chapter 5, the inactivation time constant for one of the Na^+ currents, I_{Na-rep}, needed such a precipitous characteristic in order that it reproduce repetitive Na^+-only spikes.

4.4.3 Effect of the Number of Gating Particles in a Given Channel

More than one gating particle in a single channel causes a delay in the net effect of that particle type when the membrane voltage changes. This delay

increases with the number of particles in the channel. Increasing the order (the number per channel) of the particle also makes the (effective) steady-state curve steeper and more depolarized (hyperpolarized) for activation (inactivation) particles.

Another important effect of the number of gating particles is how the resulting steady-state characteristic changes. Specifically, the activation or inactivation curves measured with the voltage-clamp protocol *do not* indicate the voltage-dependent steady-state characteristics of each particle. Rather, the resulting curves reflect the behavior of the *ensemble* of particles, a point that is not often made clear in the literature. If a channel is assumed to be governed by N activation particles, for example, then the steady-state curve for a single activation particle is found by taking the Nth root of the (overall) steady-state activation characteristic.

In the following chapters the voltage-dependent steady-state curves for both the individual gating particles for each current will be illustrated. In addition, the apparent steady-state curve of the appropriate ensemble of gating particles will be illustrated (depending on the number of gating particles assigned to a given conductance), as might be measured by the voltage-clamp protocols.

4.5 Procedure for Fitting HH Parameters

The parameters governing the kinetics of each current in the model were determined according to the data for a given current. At one extreme, non-ambiguous voltage clamp data that was almost complete specified most the relevant parameters – for example the steady-state activation curves for I_Q and I_M. At the other extreme, for example for the putative Na^+ currents, only meager voltage clamp data was available, augmented by extensive, though much more ambiguous, current clamp data. In these cases the steady-state activation curve or activation and inactivation curves as appropriate, had to be estimated and then checked with steady-state voltage clamp simulations. The functions for the time constants would then be estimated, consistent with the z and $V_{\frac{1}{2}}$ parameters that had been set by the steady-state characteristics. Simulations would then be used to check the resulting kinetics and, if necessary, the functions would be modified (e.g. changing γ or α_0) to yield better behavior. For all the currents specific time constant data was either incomplete or non-existent. These functions were iteratively derived by running current clamp simulations of certain proto-

cols for which I had data to compare the model behavior with. Note that these parameters amount to verifiable predictions of the model, assuming that experimental protocols may be devised that record the time and voltage dependence of different currents in isolation.

In Chapters 5, 6 and 7 the parameters for the model currents will be presented, along with the resulting curves for the steady-state and time constant functions.

4.6 Temperature Dependence of the Gating Kinetics

Temperature dependence of the kinetics described here has several elements, all of which ultimately derive from the temperature term in the Boltzmann distribution (eqns. 4.1 and 4.4). However, some of these relationships are handled explicitly while others are estimated.

Consider the expression for the forward and backward rate constants, α and β (ref. eqns. 4.2 and 4.3). Each expression evaluates to the maximum of two expressions, a product of two terms and (in the current approximation) a voltage-independent rate-limiting term. The product is formed by a base reaction rate term that ultimately derives from a Boltzmann distribution, although the factors in this expression are not specified. The second, voltage-dependent term in the product is also a Boltzmann distribution, however, as has been shown, each term in this distribution is specified. Therefore, the temperature dependence of the base rate is undefined while this dependence for the voltage-dependent term is explicit. Likewise, the temperature dependence (if any) of the rate-limiting term is undefined.

The base rate term and the rate-limiting term the temperature dependence was therefore assumed to be similar to that generally observed for biologic reactions, where a Q_{10} of 3 is typical [2]. This factor is used to derive a coefficient for the rate constants as follows:

$$Q_{10-factor} = Q_{10}^{\frac{T_0 - T}{10}}$$

where T is the temperature and T_0 is the temperature at which $\alpha_{0,base}$ is determined. $Q_{10-factor}$ is then multiplied with the both the base rate

[2] This factor is dependent on different currents, as appropriate. The Q_{10} for I_M is set to 5, based on Halliwell and Adams, 1982 [16], and the Q_{10} for the Na^+ currents was also set to 5 in order to improve the performance of the model.

term and the rate-limiting term as a first approximation to the effect of temperature of these terms.

Note that the temperature dependence that derives from the voltage-dependent term is (by definition) voltage-dependent. The effect of the temperature on this term disappears when the membrane voltage is equal to $V_{\frac{1}{2}}$ for a given gating particle, and the effect on the voltage-dependent term increases as the membrane voltage moves away from $V_{\frac{1}{2}}$. However for most gating particles of the model this effect is smaller than the $Q_{10-factor}$, due to the small value of z.

Another temperature dependence arises from the coefficient of the exponential term of the Boltzmann expression. To a first approximation this is typically taken to be a constant (as is done in this model). However, reviewing the significance of this term is instructive. This term is the "pacemaker" for the reaction, as it denotes the effective state transition frequency, whereas the exponential term (as explained before) relates the probability of reaching a given state after a transition. According to Eyring Rate Theory ([19]) this pacemaker term is proportional to the temperature (derived from the frequency of molecular vibrations $= kT/h$, where k and h are Boltzmann's constant and Plank's constant, respectively).

This term contributes a linear temperature dependence of the rate constants, whereas the previous temperature-related terms were exponential functions of temperature. Considering that temperature is in degrees Kelvin, the linear contribution will be negligible on the rate constants when temperature ranges over ten degrees, e.g. between 298° K (25° C) and 308° K (35° C). The present assumption of a constant coefficient for the exponential terms in eq. 4.5 and eq. 4.6 is therefore justified.

4.7 Adequacy of the HH Model for Describing the Kinetics of Putative Hippocampal Channels?

The HH model of ion channels is clearly a simple one. First, assuming that channels can be described in terms of having discrete regions that can modulate channel conductance through the steric interaction of discrete, voltage-dependent conformational states, there are likely to be more than two stable states for any such "particle" (as opposed to just the open and closed HH states). Such multi-state models and other interpretations of gating have

been considered by other investigators ([9],[4],[3], [8]). For each additional stable state there will also be an additional transition state. A different transition state could become the rate-determining step over some range of membrane voltage. resulting in a non-sigmoidal voltage dependence of the rate constants over the entire voltage range. In addition, a gating particle could possibly influence channel conductance in a more graded fashion. In this case, different conformational states would not necessarily act as binary enabling/disabling mechanisms.

In fact, experimental data for many currents indicate that the simple thermodynamic description of the HH model is not sufficient for the gating mechanisms that govern those currents. For example, some currents have shown minimal or no voltage-dependence for either their activation/inactivation time constants nor their steady state values. In many of these cases whether this reflects the true kinetic nature of the currents, whether this is artifactual from the inherent limitations of the equipment, or whether there is contamination from other currents that has not been accounted for is not clear. In some cases. different measurement protocols can shed light on these questions. In other cases. simulations can help test speculations as to the true nature of the currents. Another explanation is that there is a distinct linear rate-limiting mechanism that alters the function for the time constant as would be expected from the HH model. Such a mechanism is considered in the present simulations. as will be described later.

Another complicating factor is one that reflects actual physiological mechanisms. yet is not explicitly described in the HH model. This factor is the effect of the concentration of various ions in the vicinity of the membrane. There will be an observable effect of different concentrations of the predominant ions (Na^+, K^+, Ca^{2+}, and Cl^-) on the reversal potential for these species, as expressed in the Nernst equation. given that a given ion undergoes large changes in its local concentration because of sequestering, saturation of buffering mechanisms, or active transport. The model described here assumes only passive transport of the charge carriers across the membrane: for example, maintenance of the Na^+ and K^+ concentration gradients in light of the flux of these ions during electrical activity is assumed to occur over a long time scale. In addition. there are many cases where the local concentration of some ion is a regulator of some active process - e.g. Ca^{2+} in the activation of the actin-myosin system and as a mediator in the conductance of certain channels. As will be described later, such coupling is indicated in some of the hippocampal pyramidal cell non-linear currents. the notable

example being the Ca^{2+} -activated K^+ current, I_C. In this case there is evidence that the conductance underlying this current is dependent on the concentration of Ca^{2+} underneath the membrane, as may be supplied by the Ca^{2+} currents (e.g. I_{Ca} and I_{CaS}).

On the other hand, in support of the HH approach, there is evidence that the HH description is valid for at least some ion channels. For example, the movement of charge that occurs when the postulated gating particles change state (the so-called "gating current") has been detected for some channels ([19]). The primary structure of certain channels, e.g. some Na^+ channels and acetylcholine receptors, have been sequenced, and speculations on the tertiary structure have been made on the basis of this data. There are indications in these sequences of segments with polar residues that traverse the membrane in such a way so that they maybe able to sense the membrane voltage, i.e. properties expected of putative "gating particles".

On a more empirical level, simulations of non-linear membrane using HH-like descriptions for the ion channels have been successful in reproducing the electrical activity of several electrically-active cells. In the present work, it was remarkable how well HH models were able to reproduce the behavior of several channels.

4.8 The Concept of an "Extension" of the HH Model

The descriptions for the HIPPO non-linear conductances are based on *extensions* of the HH model. This is because the HIPPO descriptions explicitly consider the implications of the single-barrier gating model proposed by Hodgkin and Huxley, especially with regard to the relationship between the parameters that define this model and the resulting voltage-dependent time constants for the gating particles. In other studies that draw on the HH model the relation between the steady-state characteristics of the gating particles and their temporal characteristics is purely empirical, and is not derived from the single-barrier model.

Chapter 5

ESTIMATING Na^+ CURRENTS

5.1 Introduction

This chapter describes the derivation of the kinetics for three proposed Na^+ currents in the hippocampal pyramidal cell. I shall begin with the background for this problem, and then I shall present the data that was used to derive the model parameters. After the motivation for using three Na^+ currents is discussed, the strategy I used to estimate the relevant parameters will be presented.

The parameters for the Na^+ currents will then be presented. Some of these parameters will be compared with the analogous parameters of the squid axon Na^+ channel and the I_{Na} of the rabbit node of Ranvier, since these latter two currents are among the few Na^+ channels for which the kinetics have been measured under voltage clamp.

5.2 Background for Evaluating I_{Na}

One of the first applications of the model has been the estimation of the Na^+ currents in hippocampal pyramidal cells, including those which underlie the depolarizing phase of the action potential. The fast Na^+ conductance necessary for the spike corresponds to the classical Na^+ current described by Hodgkin and Huxley. To initiate the action potential, this current rapidly turns on when the membrane voltage passes the firing threshold for the cell. Almost as rapidly, the fast Na^+ turns itself off as the cell depolarizes.

contributing substantially to the repolarization of the action potential.

A quantitative description of the Na^+ currents is vital because these currents are the progenitors of the action potential and therefore are some of the basic determinants of neuronal function. Also the activation/inactivation properties of the Na^+ currents set the stage for the entrance of the numerous outward currents.

There is little voltage clamp data for Na^+ currents since these currents are typically large and fast, exceeding the current sourcing ability and the temporal response of the single-electrode clamp circuit used to make the measurements. Since the data is not complete, it was necessary to look to sources of data other than that from hippocampal preparations. These included estimations of the kinetics of a fast Na^+ current in rabbit node of Ranvier ([10]) and in the bullfrog (Koch and Adams, bullfrog sympathetic ganglion simulations, personal communication. In addition parameters used in other neuron simulations were consulted ([48], hippocampal simulations).

In the HIPPO model, this problem has been approached several ways, including using the descriptions just mentioned. I also tried using I_{Na} kinetics based on measurements from rabbit node of Ranvier, with some modifications. In particular, the time constants for the m and the h variables were scaled by two, in addition to the appropriate temperature compensation ($q_{10} = 3$, Adams, personal communication)[1].

Attempts to derive the original source for the kinetics used by Traub, et al, were unsuccessful. My impression is that the kinetics used in this model are simply the ones derived by Hodgkin and Huxley for squid axon, modified slightly to yield acceptable empirical results for the simulation of some protocols. Initially I tried such an approach.

Specifically, I have attempted to derive channel kinetics that are consistent with current clamp records of Na^+-only spikes (Storm, personal communication), the steady-state Na^+ dependent current-voltage characteristic ([12], Storm ibid), and current clamp records of normal action potentials obtained under various conditions, under the assumption that any channels that conduct Na^+ may be described by the HH-like kinetics described earlier, and further that each channel may have one or two types of gating particles. The task was therefore to try to fit the behavior of this class of voltage-dependent channels to the data. I began by considering the Na^+-only spike.

[1] From scaling of time constants for I_{Na} in bullfrog myelinated nerve and bullfrog sympathetic ganglion soma.

In particular, it was desired to describe the fast Na^+ so that the cell had the capacity for a stable resting potential that in turn could be perturbed enough to result in an action potential. This implied that at the resting potential the inactivation variable (h) was turned on and that the activation variable (m) was well turned off. In addition, the time constant for the m variable had to be substantially less than the time constant for the h variable throughout most of the depolarized range of the membrane potential above rest. This insured that once threshold was reached, the m variable would have a chance to fully activate and allow Na to enter the cell before the h variable caught up with the depolarization and subsequently go into the closed state, thus shutting off the conductance.

Although a useful description was found empirically, it will be important to compare this description to actual measurements of the fast Na current kinetics whenever they become available.

5.3 Deriving Na^+ Conductance Kinetics

5.3.1 Implications of Na^+-only Spike

Current clamp records taken using hippocampal slices which had been treated with agents that blocked all potassium and calcium currents enables one to look at the behavior of the Na^+ currents and, presumably, the leak conductance in isolation. Such protocols assume that 1) all non-linear currents other than Na^+ currents are blocked, and 2) such treatment leaves the leak conductance unchanged. Figure 5.1 shows a record of a Na^+-only spike under such conditions.

This spike gives several clues about the Na^+ currents in this cell. First, the spike threshold is quite sharp. Also the subthreshold response shows very little activation of inward current. This behavior of the spike threshold implies that the activation curve for the Na^+ current underlying the initiation is steep, with the curve centered around -55 millivolts.

The second feature is the biphasic repolarization of the spike. The trajectory of the spike repolarization under the specified conditions is due to two factors - the inactivation of the Na^+ current(s) and the linear leak of the membrane. Initially, the spike repolarizes rapidly. Assuming that the major portion of the spike is due to a Na^+ current similar to the classical fast Na^+ current described in squid axon, this initial repolarization is consistent with the rapid inactivation of the channel with depolarization. At depolarized membrane potentials, the time constant for inactivation is on

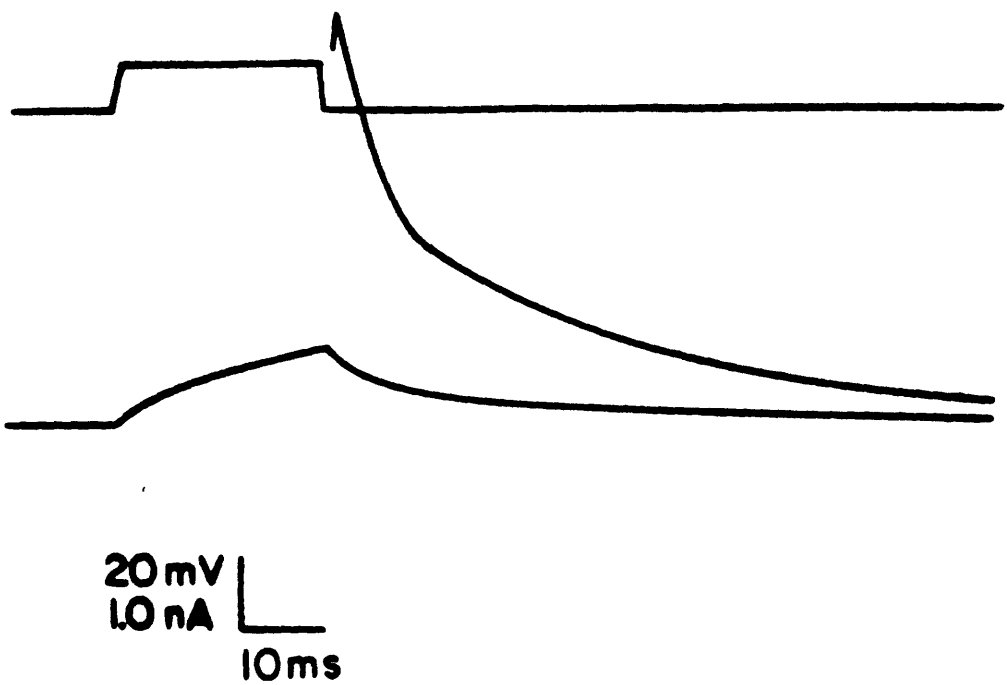


Figure 5.1: Na^{+}-only Spike and Subthreshold Response - Current clamp protocol with cesium chloride electrode. TEA. 4AP. and Mn^{++} added to block the calcium and potassium currents. Resting potential is -65 mv. Stimulus current is top trace. From Storm (unpublished data).

the order of a few milliseconds. However, approximately 7 milliseconds after the spike peak the repolarization slows drastically. This slow phase of the repolarization, which commences when the membrane voltage is about -20 millivolts, lasts approximately 60 milliseconds. Since this decay is too slow to be accounted for by the time constant of the cell, we propose that the long tail is due to an non-linear (Na^+) inward current.

We can determine whether a Na^+ tail current is likely to be present during a spike that is repolarized by outward currents. The action potential is repolarized by K^+ currents, in addition to the leak conductance and the inactivation of the Na^+ currents. If any Na^+ tail current has been activated during the fast spike, then it must be canceled by a slow residual component of the outward currents, since no long lasting depolarized tail is observed. During a normal action potential there is therefore either a completely activated slow component of the fast Na^+ current that is canceled by a slow K^+ current(s), or there is a separate slow Na^+ current that has not had a chance to be activated during the short spike, or there is some middle ground where a incompletely-activated inward current is canceled by a residual outward current.

The time course of the actual spike was used as the clamp voltage in a voltage clamp simulation using the linear cell in order to estimate the current during a Na^+-only spike. As was described in Chapter 3, the resulting simulated clamp current revealed the total current that must be supplied by non-linear conductances during the spike. Incidentally, this protocol was an example of the power of the simulation technique, since controlling an actual microelectrode voltage clamp with such a fast time-varying signal is not always possible.

The result of the voltage clamp simulation is shown in Figure 5.2 The time course of the clamp current implied that the non-linear mechanisms underlying the spike had at least two distinct components, an early, large component which quickly deactivated/inactivated, and a later small component which deactivated/inactivated slowly, remaining for approximately 100 milliseconds.

The fast component was assumed analogous to the classical fast Na^+ current of the squid axon as described by Hodgkin and Huxley.

For the repolarizing tail I considered two possible mechanisms: an abrupt slowing of inactivation of the fast Na^+ current underlying the spike, or the action of another kind of Na^+ channel. For the present this first possibility has been discounted for two reasons. First, I have not been able to derive a function for the voltage-dependent time constant for inactivation for the fast

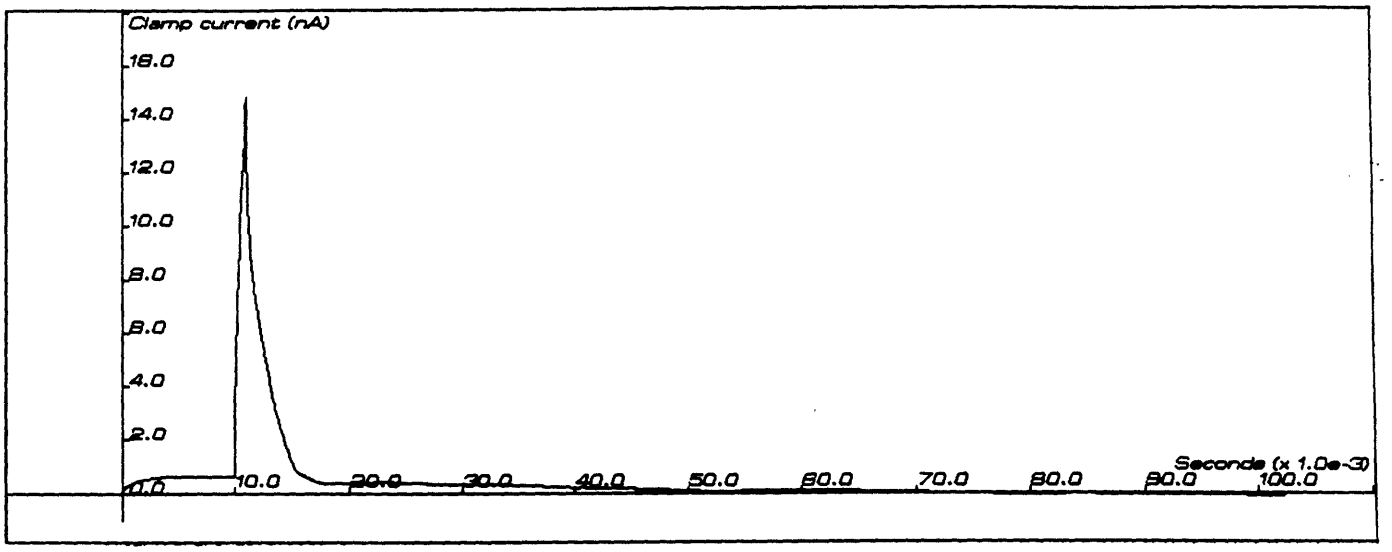


Figure 5.2: Clamp current during voltage clamp simulation using time course of Na^+-only spike (Figure 5.1) as command voltage

Na^+ that was consistent with the single-barrier gating assumptions and that had the necessary sharp increase at the appropriate voltage. Second, in light of the present assumptions regarding the behavior of the K^+ currents, it was determined that the mechanism for the slow tail would only be significantly activated during a long (e.g. non-repolarized) spike, thereby removing the requirement of an outward current that would cancel out the slow tail current after the normal spike.

In considering the possibility of a distinct tail current, the important characteristics of this current was that it had to have a high threshold and a slow onset, consistent with the lack of a long after-depolarization in normal spikes. For example, if this current had a threshold of approximately -10 millivolts with a slow activation time constant, i.e. 4 milliseconds, then during a normal spike this current will not have time to activate fully. On the other hand, during the slower repolarization that occurs without non-linear outward currents, this tail current will have time during the peak of the spike to activate more, and thereby contribute to a long repolarization. I called this current $I_{Na-tail}$. I attempted to adapt the activation data for I_{NaP} (discussed next) to account for the action of the so-called $I_{Na-tail}$, but this has been unsuccessful to date. This is primarily because the low threshold of the activation curve for I_{NaP} has thwarted attempts at deriving a function for the time constant of activation that is consistent with the single-barrier model and which in turn reproduces the Na^+-only spike.

5.3.2 Implications of Na^+-only Repetitive Firing

Repetitive firing elicited in cells in which all currents except Na^+ have been blocked offer additional clues as to the nature of the Na^+ currents in hippocampal pyramidal cells. Figure 5.3 illustrates such a record. The key features of these voltage traces are 1) higher threshold of spikes following initial spike (i.e. higher threshold of the *secondary* spikes). 2) reduced amplitude of repetitive spikes. 3) reduction of spike amplitude with increasing stimulus, 4) repetitive firing elicited only in a narrow range of membrane voltages.

5.3.3 Implications of Tetrodotoxin Sensitive Steady State Current-Voltage Characteristic

Figure 5.4 shows a steady-state current-voltage characteristic from hippocampal pyramidal cells that demonstrates a tetrodotoxin (TTX) sensitive

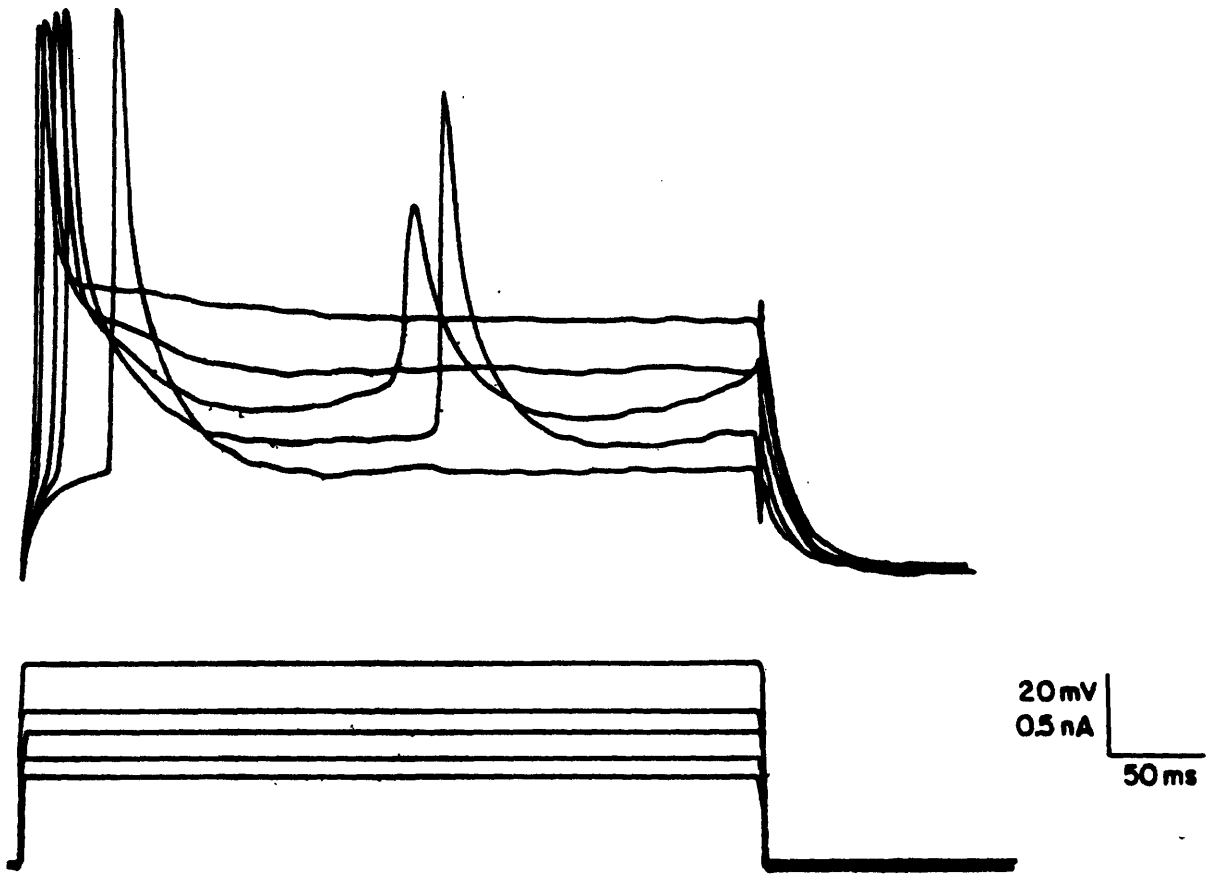


Figure 5.3: Na^+-only Repetitive Spiking - Current clamp protocol under same conditions as Figure 5.1. Current stimuli is bottom trace. From Storm (unpublished data).

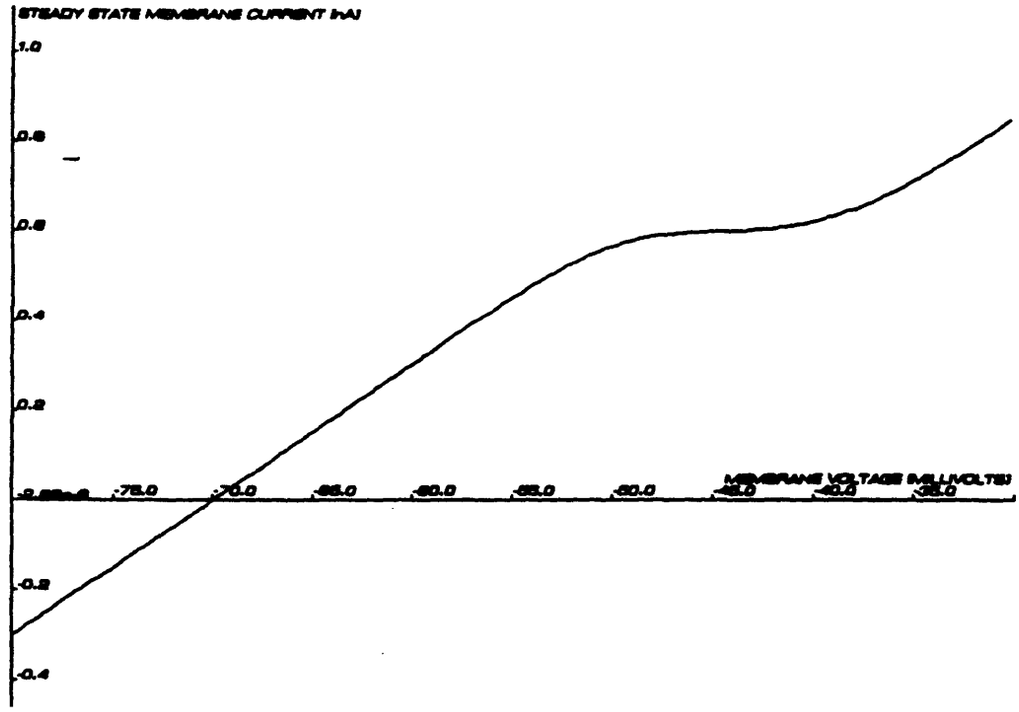


Figure 5.4: Inward Rectification by Na^+. Curve derived from steady-state activation of a persistent Na^+ current. I_{NaP} ([12]).

inward-rectification ([12]). Assuming that a sensitivity to TTX means that Na^+ currents underlie this rectification, the characteristic can be accounted for by either the "window current" of a transient Na^+ current, by a persistent (non-inactivating) Na^+ current (I_{NaP}), or by some combination of these types of channels.

5.3.4 The Role of Window Currents

Window current is due to any overlap in the voltage-dependent steady state curves of the activation and inactivation variables, thereby making a normally transient current contribute a persistent component over some range of membrane voltage. Since any overlap in the activation and inactivation curves will be limited, rectification due to a window current alone would disappear at depolarized membrane voltages. The steady-state current-voltage characteristic would then continue the linear characteristic established prior to the onset of rectification. The data for this cell, however, would not necessarily demonstrate a depolarized removal of rectification since the steady-state current-voltage curve was only measured to -35 millivolts.

Important aspects of this characteristic include the lack of inward rectification around the Na^+-only spike threshold, which implies that the m and

h curves for the current activated at the threshold do not overlap at that threshold.

5.3.5 Adding Together All of the Evidence

Taked all together, the data presented so far implies several characteristics of any TTX-sensitive (presumably Na^+-carried) currents. These may be summarized as follows:

1. Na^+ mediated repetitive firing in cells depolarized from the resting potential implies that the inactivation curve for the current underlying the higher threshold spikes is non-zero at the depolarized level.

2. The lack of inward-rectification at the lower spike threshold contradicts the earlier conclusion that the activation curve for the fast Na^+ current is steep at the lower threshold.

3. A steep activation curve at the lower threshold taken with the non-zero inactivation at depolarized membrane potentials would result in an appreciable window current. This window current in turn would contribute to inward rectification starting at the lower spike threshold of -55 millivolts. This is inconsistent with the data.

To explain these phenomena, I suggest that there is an additional fast Na^+ channel whose threshold for firing is depolarized from that of the original fast Na^+ channel, and whose activation and inactivation kinetics are such that it might mediate Na^+-only repetitive firing. In the absence of repolarization from any non-linear outward currents, simulations indicated that there must be a finite overlap of the activation and inactivation curves of any HH-like Na^+ channel that can mediate repetitive firing. This overlap will result in a finite window current, and thus a steady state inward rectification. I was able to adjust this rectification to qualitatively reproduce the onset of the observed rectification discussed earlier. Because it mediates repetitive Na^+-only spikes, I named the high threshold current I_{Na-rep}. Since I deduced that the original fast Na^+ current had a sensitive, low threshold for initiating the action potential, I called this current $I_{Na-trig}$.

The steady-state Na^+ mediated rectification also constrains the behavior of the $I_{Na-tail}$. In particular, if this current contributed any window current then such a window current could only activate above -30 millivolts, in order to be consistent with the steady-state IV characteristic.

The implications of a sharp threshold for the Na^+ -only spike. and a small subthreshold response implies steep and/or activation characteristics for the current underlying the initiation of the spike. On the other hand, presumed modulation of the spike threshold by outward (K^+) currents (see Chapter 7) which in turn do not greatly effect the slope or amplitude of the action potential implies that around threshold Na^+ activation is not instantaneous. in other words a small outward conductance would be able to counter the sub-threshold inward rectification of the Na^+ current sufficiently to raise the firing threshold. Note that the faster the Na^+ current activated around threshold, the larger the outward current would have to be to suppress the initiation of the spike. Since threshold is only about 30 millivolts above E_K, the small driving force for an outward K^+ current means that a large conductance is required. However, a large K^+ conductance that is enabled immediately prior to the spike would allow a large outward current on the upstroke of the spike, due to the increasing driving force that the K^+ ions see. An alternative explanation is that the threshold-modulating K^+ current shuts off prior to or during the upstroke of the spike. and thus a K^+ conductance of sufficient size to transiently counteract a quickly activating Na^+ current would not then serve to attenuate the spike itself. A final alternative is that the size of the spike current is large enough that a sub-threshold activated outward conductance would not attenuate the spike noticeably.

5.4 Strategy for Determining Na^+ Current Kinetics

Once it was determined that three Na^+ currents might model the observed behavior. the following strategy used to derive their kinetics:

1. Estimate the absolute Na^+ conductance for the fast Na^+ currents ($I_{Na-trig}$ and I_{Na-rep}) by running voltage clamp simulations using the Na^+-only spike as the command voltage.

2. A reasonable set of equations governing the kinetics (backward and forward rate constants for the activation gating particle m and inactivation gating particle h) for the three putative Na^+ currents was determined. The free parameters for each function include the free energy changes between the stable states and the transition state. the

location of the transition state within the membrane, and the effective valence of the gating particle. Voltage-dependent functions of the time-constants and steady-state values of the gating particles are then derived from the appropriate rate functions.

3. Run (current clamp) simulations of the Na^+-only single and repetitive spike protocols.

4. Compare the simulations with the data.

5. Adjust the appropriate rate-constant functions and repeat the simulations.

6. Once a good match between the current clamp simulations and the data was reached, the steady state current-voltage characteristic of the cell with all three Na^+ currents activated was derived to measure the inward-rectification generated by the estimated currents.

7. This characteristic was compared with that of one from the model with the derived Na^+ currents replaced by the reported persistent Na^+ current.

8. If needed, return to step 5. in order to obtain a good fit to all the available data.

This process eventually converged to yield a model description that was in good qualitative agreement with the data pertaining to Na^+-only behavior. The derived Na^+ currents were then tested by running simulations in which various K^+ currents were added, once they were derived. This led to a modification of some of the parameters of the Na^+ currents, while preserving the Na^+-only behavior, which provided a rigorous set of constraints on the parameter adjustment. The entire process was and is one of adding one piece of information at a time to the model, and then running simulations to find out how the new data affects the model's behavior.

5.5 Results

5.5.1 Simulation of Na^+-Mediated Inward-Rectification and Spikes

Figure 5.5 compares the steady-state current-voltage characteristic of the model with 1) the reported I_{NaP}, and 2) the $I_{Na-trig}$, $I_{Na-tail}$, I_{Na-rep} cur-

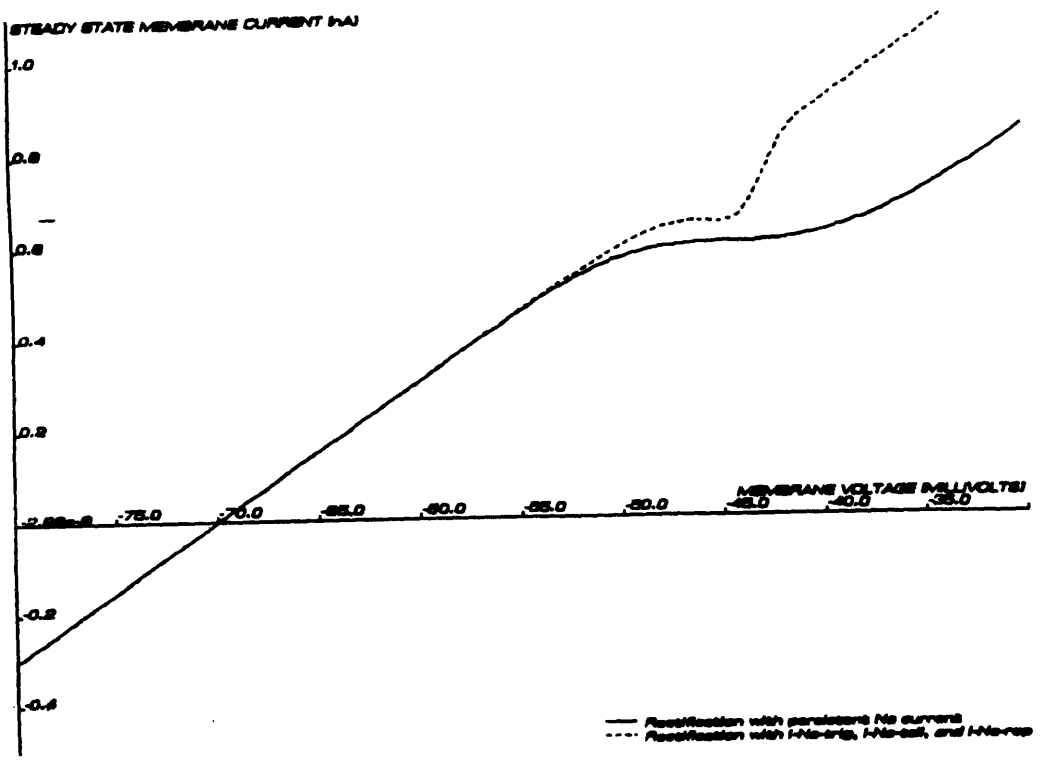


Figure 5.5: Current-voltage characteristics of model showing inward-rectification mediated by I_{NaP} and by $I_{Na-trig}$, $I_{Na-tail}$, and I_{Na-rep}, currents.

rents. The model currents cause an onset of inward rectification that is in qualitative agreement with the published data. However, since this steady-state inward current is mainly due to the transient I_{Na-rep} window current, the rectification only occurs over a limited range of membrane voltages. This is not necessarily inconsistent with the characteristic of I_{NaP} because of the limited range over which this current was measured, as explained earlier. Possibly the so-called persistent Na^+ current is actually a transient current which would demonstrate removal of inward-rectification at more depolarized membrane potentials. Given more data, the derived characteristics of the so-called I_{Na-rep} might be adjusted to better match the steady-state current-voltage relationship of the model.

Figure 5.6 illustrates a simulation of the Na^+-only single spike. The model's behavior is in good agreement with the data, in particular in regards to the sharp threshold of the spike, the time course of the depolarizing phase, the initial fast repolarization, and the slower late repolarization. Also in the figure are the three model Na^+ currents that underlie the Na^+-only spike. In this figure the initial activation of $I_{Na-trig}$, the subsequent recruitment of the higher threshold I_{Na-rep}, and the slow time course of $I_{Na-tail}$ after the first two currents have inactivated can be seen.

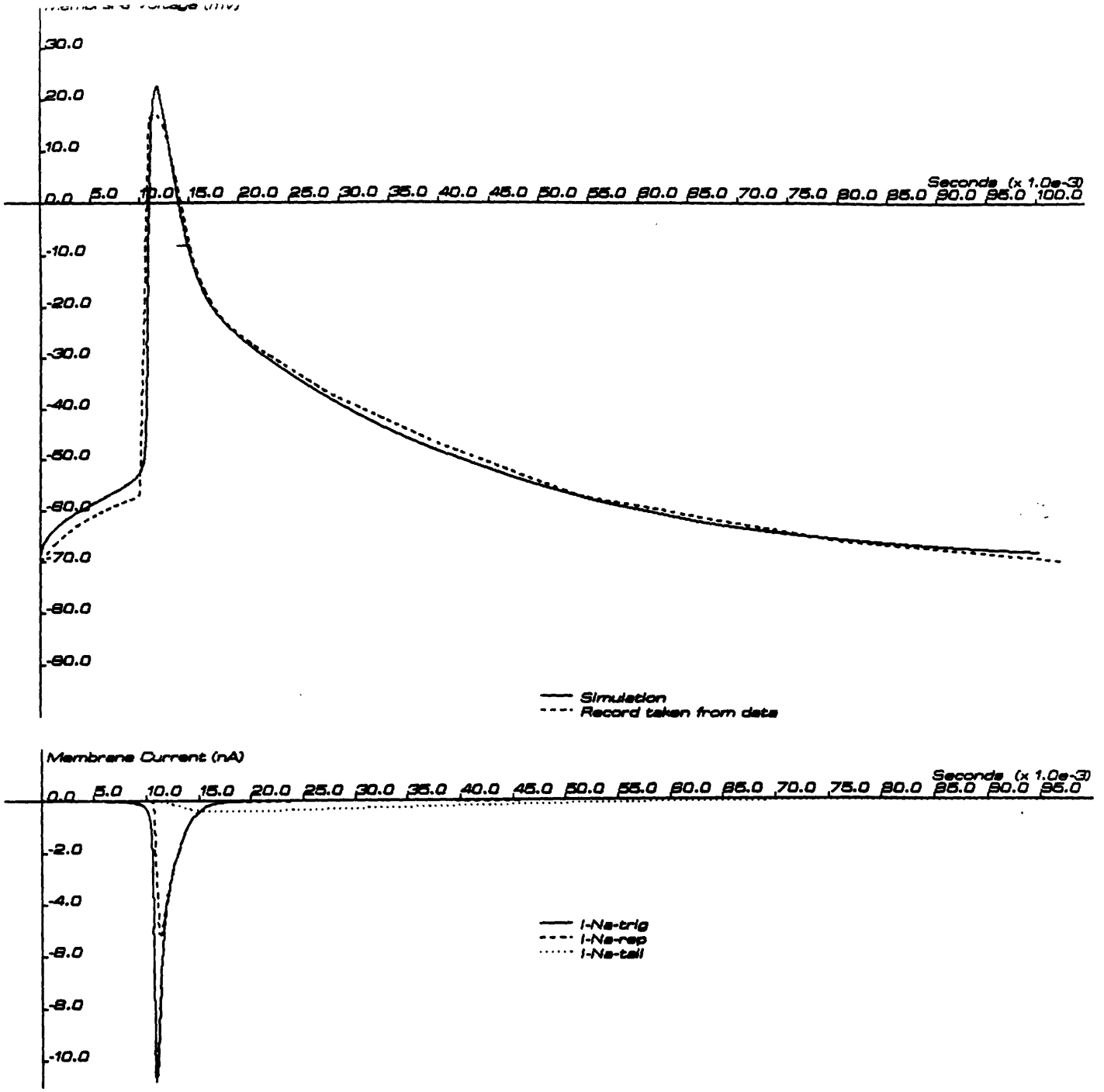


Figure 5.6: Current clamp simulation of Na^+-only single spike. Spike stimulus - 0.78 nA. Top - Simulation of spike compared with record taken from data. Bottom - $I_{Na-trig}$, $I_{Na-tail}$, I_{Na-rep} currents during spike.

Figure 5.7 illustrates a simulation of Na^+-only repetitive firing under different constant current inputs. At the bottom of the figure are the Na^+ currents underlying the marked spike train. After the first spike, the initiation of later spikes is mediated completely by I_{Na-rep}.

5.6 Parameters of the Three Putative Na^+ Conductances

The parameters for the three proposed hippocampal Na^+ currents will now be presented in detail. Some of these parameters will then be compared with the analogous parameters of the squid axon Na^+ channel and the I_{Na} of the rabbit node of Ranvier.

All parameters were set assuming a temperature of 24°C. It was necessary to use a high value for the q_{10} (= 5) for these currents since simulations of action potential repolarization at the higher temperature used for interpreting most of the K^+ current data (32°C) indicated that significantly faster activation/inactivation was required. Figure 5.8 shows the resulting effect of different temperatures on the Na^+-only spike. The striking effect of temperature in these simulations suggest that measuring the temperature dependence of Na^+-only spikes in HPC may provide a good test for the model description of the Na^+ currents.

5.7 Parameters of $I_{Na-trig}$

$I_{Na-trig}$ is based on the classical I_{Na} of the squid axon. Important differences were required, however, so that $I_{Na-trig}$ would have a sharp threshold with very little subthreshold activation. Also, it was necessary to adjust some parameters to obtain the desired characteristics during normal repetitive firing.

5.7.1 Results

First, the valence of both the m and h particles is large ($z_m = 20$, $z_m = 30$), which makes them steep functions of voltage. Likewise, the m_∞ and h_∞ curves for $I_{Na-trig}$ do not overlap as they do in the squid axon I_{Na} (ref. Figure 5.15).

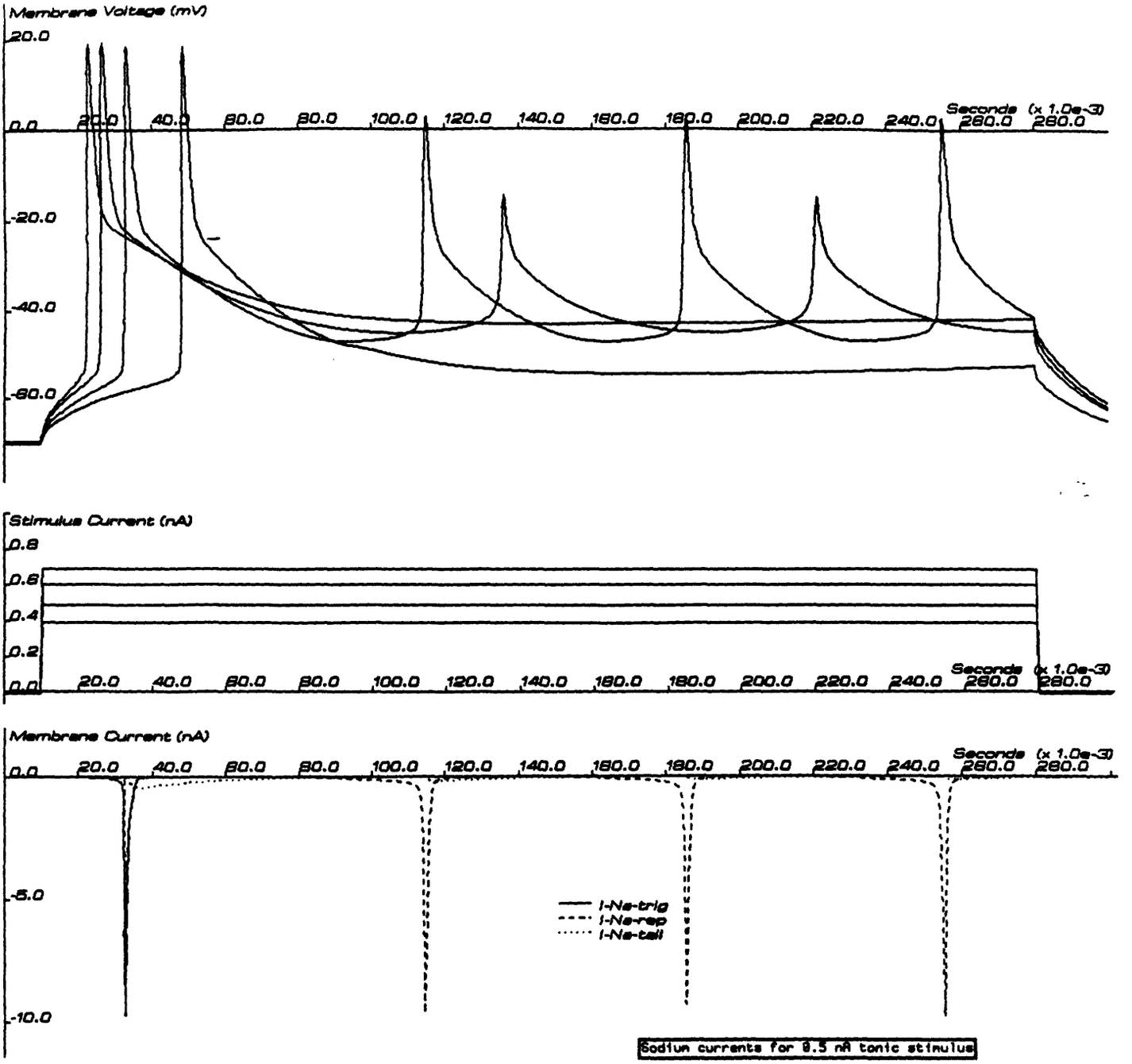


Figure 5.7: Current clamp simulation of Na^+-only repetitive spike protocol. Top - Spike trains in response to different stimulus strengths. Middle - $I_{Na-trig}$, $I_{Na-tail}$, I_{Na-rep} currents during trace marked with the arrow. Bottom - Stimulus currents.

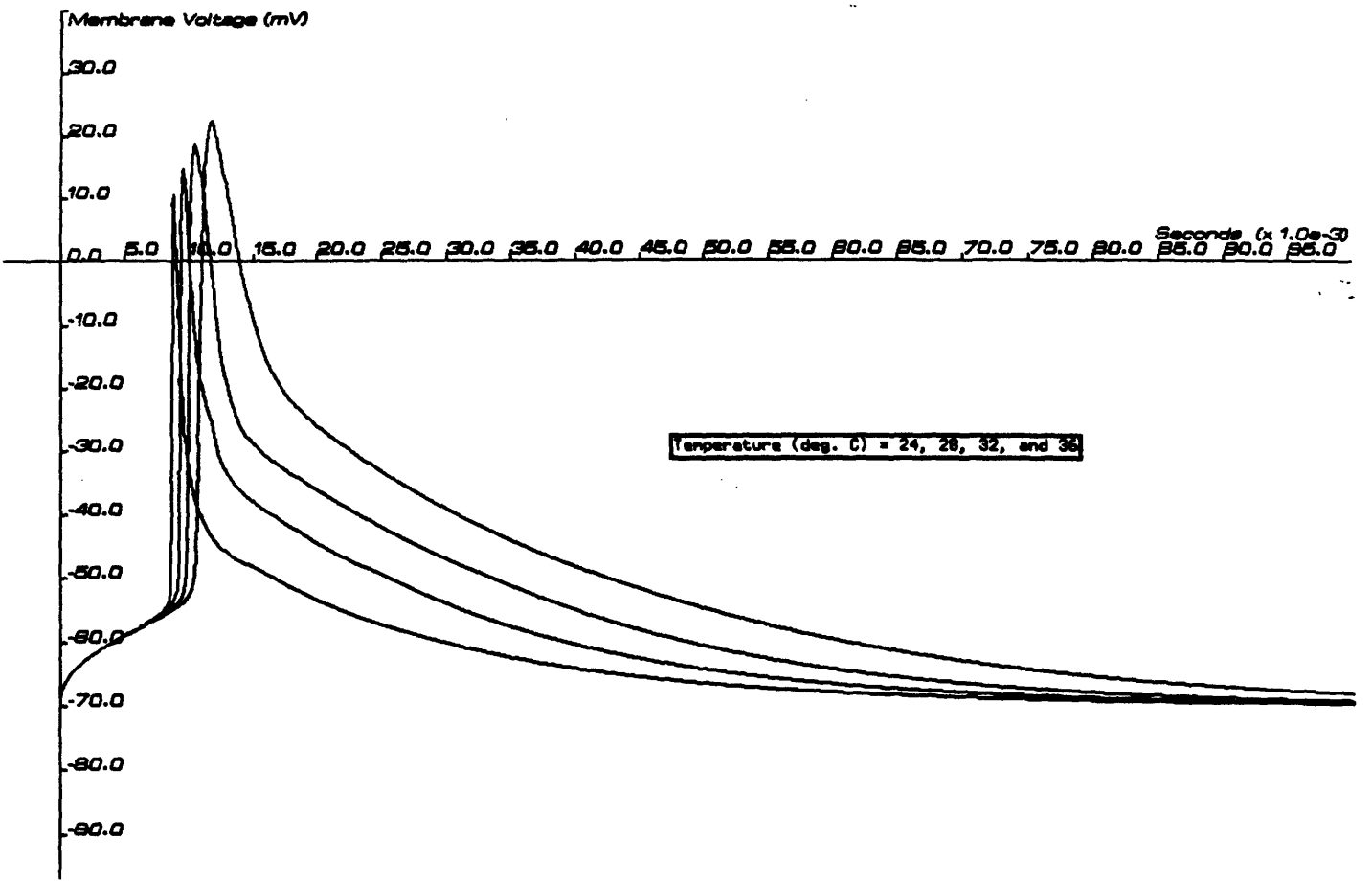


Figure 5.8: Current clamp simulations of Na^+-only single spike at different temperatures. q_{10} for the three Na^+ current kinetics is set to 5. q_{10} for the absolute conductances is set to 1.5 . With increasing temperature the spike threshold drops, the depolarizing slope is steeper, and the repolarization (due to inactivation/deactivation) is faster.

The position on the voltage axis, as determined by $V_{\frac{1}{2},m,Na-trig}$ and $V_{\frac{1}{2},h,Na-trig}$, were set to make the firing threshold equal to about -55 millivolts. This threshold was made slightly higher than indicated by the data in order to allow subthreshold activation of I_A (ref. Chapter 7).

Setting the order of the inactivation particle h and the the $\tau_{h,Na-trig}$ magnitudes involved compromising between formulations that met a) the observed width (about 1.7 milliseconds at 0 millivolts) and b) the observed height (about 15 millivolts) of Na^+-only spike. The formula I have used includes two h particles and setting $\tau^o_{h,Na-trig} = 2.0$ milliseconds so that the current would not inactivate too quickly at the top of the spike. When a single h particle was used it was necessary to adjust $\tau^o_{h,Na-trig} = 1.5$ milliseconds to maintain the width of the spike; however. this formula made the peak amplitude too high.

The curve for $\tau_{m,Na-trig}$ was symmetrical ($\gamma_{m,Na-trig} = 0.5$). but when normal repetitive firing was simulated using the K^+ currents. it was necessary to make the curve for $\tau_{h,Na-trig}$ skewed to the right (depolarized) ($\gamma_{h,Na-trig} = 0.2$) so that removal of inactivation was fast enough near rest to allow for rapid firing.

The value for $\bar{g}_{dens,Na-trig}$ ($= 40\text{mS/cm}^2$) was set in order to obtain an initial slope of the action potential of approximately 130 volts per second (measured from threshold to 0 millivolts). This value was dependent on $\bar{g}_{dens,Na-rep}$ as well . since I_{Na-rep} is activated within a few tenths of a millisecond after the beginning of the spike and therefore I_{Na-rep} contributes substantially to the upstroke of the action potential (see Figure 5.6).

The equation for $I_{Na-trig}$ is -

$$I_{Na-trig} = \bar{g}_{Na-trig}\, m_{Na-trig}\, h^2_{Na-trig}\,(V - E_{Na^+})$$

where

$$\bar{g}_{Na-trig} = 0.53\,\mu\text{S}$$

$$\bar{g}_{dens,Na-trig} = 40.0\,\text{mS/cm}^2$$

Table 5.1 lists the parameters for the $I_{Na-trig}$ gating variables. These are the rate functions for the activation variable, m, of $I_{Na-trig}$ -

$$\alpha_{m,Na-trig} = 0.3 \exp\left(\frac{(V + 47)0.5 \cdot 20 \cdot F}{RT}\right)$$

Gating Variable	z	γ	α_0	$V_{\frac{1}{2}}$ (mV)	τ_0 (ms)
m (activation)	20	0.5	0.3	-47.0	0.5
h (inactivation)	-30	0.2	0.003	-54.0	2.0

Table 5.1: Parameters of $I_{Na-trig}$ Gating Variables

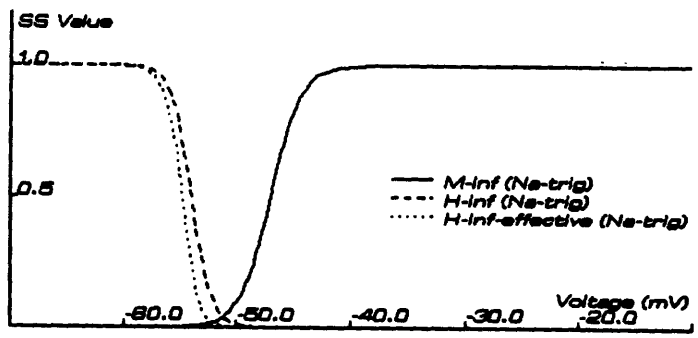


Figure 5.9: Steady-state curves (m_∞ and h_∞) for $m_{Na-trig}$ and $h_{Na-trig}$ and effective curves as would be measured by voltage-clamp experiments.

$$\beta_{m,Na-trig} = 0.3 \exp\left(\frac{(-47 - V)0.5 \cdot 20 \cdot F}{RT}\right)$$

These are the rate functions for the inactivation variable, h, of $I_{Na-trig}$

$$\alpha_{h,Na-trig} = 0.01 \exp\left(\frac{(V + 61)0.7 \cdot -30 \cdot F}{RT}\right)$$

$$\beta_{h,Na-trig} = 0.01 \exp\left(\frac{(-61 - V)0.3 \cdot -30 \cdot F}{RT}\right)$$

Figures 5.9 and 5.10 show the voltage dependence on the steady-state values and the time constants for the $m_{Na-trig}$ and $h_{Na-trig}$ variables.

5.8 Parameters of I_{Na-rep}

The kinetics of I_{Na-rep}, like $I_{Na-trig}$, is similar to the squid axon I_{Na} kinetics. In order that I_{Na-rep} be able to generate repetitive $Na-only$ spikes,

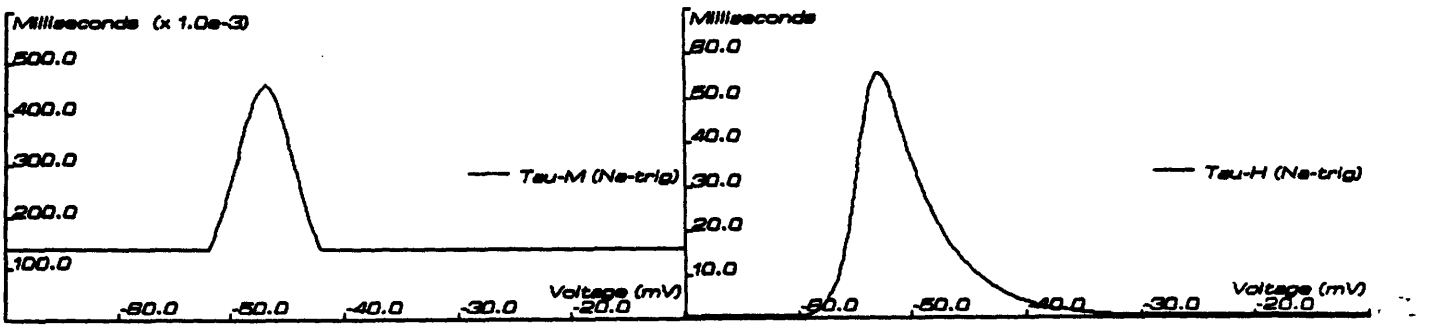


Figure 5.10: Time constant curves (τ_m and τ_h) for $m_{Na-trig}$ and $h_{Na-trig}$.

however, it was necessary to adjust the parameters for this conductance very carefully. Experimentation revealed that a key requirement for getting high threshold $Na-only$ repetitive spikes was that the m_∞ and h_∞ curves for I_{Na-rep} overlap. In addition, the curve for h_∞ had to be very steep and the curve for τ_h had to be sharp on the hyperpolarized side. These characteristics were needed so that during the repolarization after a spike, removal of inactivation would occur while m_{Na-rep} was large enough to allow enough current for another spike. On the other hand, h could not be so fast that there was $Na-only$ repetitive firing without tonic stimulation.

5.8.1 Results

Experimentation with the order of m and h resulted in the assignment of two m and three h particles to the I_{Na-rep} conductance. The high order of h accentuated the steepness of the h_∞ curve so that when the cell repolarized slowly (with a tonic current stimulus) the removal of inactivation would occur abruptly enough to allow repetitive firing. A single m particle did not provide enough positive feedback on the initiation of secondary spikes to get the observed magnitudes (e.g. between -20 and 5 millivolts). Three m particles did not allow the channel to retain sufficient activation after the initial spike to initiate subsequent spikes.

The value for $\bar{g}_{dens,Na-rep}$ had various effects. In particular, the value for $\bar{g}_{dens,Na-rep}$ modulated the role of $I_{Na-trig}$ during the initial slope of the action potential. As introduced previously, both the value of $\bar{g}_{dens,Na-rep}$ and $\bar{g}_{dens,Na-trig}$ determined this slope.

A second consequence of $\bar{g}_{dens.Na-rep}$ was that it had to be large enough to support regenerative firing when $I_{Na-trig}$ was inactivated because of the depolarized membrane. On the other hand, $\bar{g}_{dens.Na-rep}$ could not be too large since this would give a significant depolarizing hump after the initial spike when the tonic stimulus is too small for repetitive firing - such a hump is not observed experimentally (Figure 5.3).

On a more subtle level, the relationship between stimulus magnitude and the second spike during Na^+-only repetitive firing is such that initially (from below threshold to just above threshold for the tonic stimulus) the greater the stimulus the sooner the second spike. However, past a certain point the greater the stimulus the *later* the second spike occurs, until the stimulus is too large to promote Na^+-only repetitive firing. During my simulations I found that this behavior was dependent on $\bar{g}_{dens.Na-rep}$ - if $\bar{g}_{dens.Na-rep}$ was too large, then there was no range of stimulus strengths in which a larger stimulus caused the second spike to occur earlier.

In practice the most critical test of $\bar{g}_{dens.Na-rep}$ was the latter relation between $\bar{g}_{dens.Na-rep}$ and the timing of the second spike during Na^+-only repetitive firing. Once the desired relationship was achieved the other characteristics were matched primarily by the adjustment of other relevant parameters.

The overlap for the m_∞ and h_∞ curves resulted in the steady state Na^+ mediated inward rectification discussed earlier.

In summary, the parameters for I_{Na-rep} were among the most sensitive of the model, and a substantial amount of effort was needed to derive them.

The equation for I_{Na-rep} is -

$$I_{Na-rep} = \bar{g}_{Na-rep} m^2_{Na-rep} h^3_{Na-rep} (V - E_{Na+})$$

where

$$\bar{g}_{Na-rep} = 0.50\,\mu\text{S}$$

$$\bar{g}_{dens.Na-rep} = 35.0\,\text{mS/cm}^2$$

Table 5.2 lists the parameters for the I_{Na-rep} gating variables. These are the rate functions for the activation variable, m, of I_{Na-rep} -

$$\alpha_{m.Na-rep} = 0.67 \exp\left(\frac{(V + 34)0.5 \cdot 6 \cdot F}{RT}\right)$$

Gating Variable	z	γ	α_0	$V_{\frac{1}{2}}$ (mV)	τ_0 (ms)
m (activation)	6	0.5	0.67	-34.0	5.0
h (inactivation)	-30	0.17	0.0023	-42.5	3.0

Table 5.2: Parameters of I_{Na-rep} Gating Variables

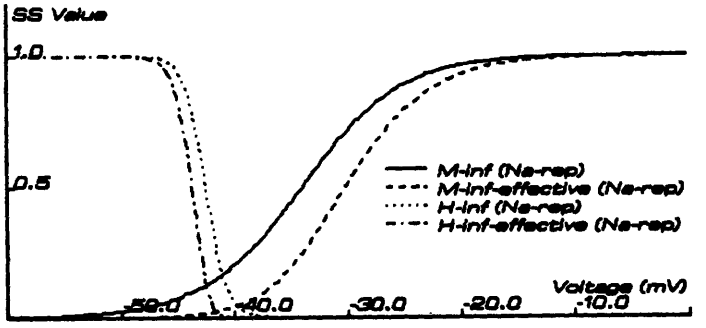


Figure 5.11: Steady-state curves (m_∞ and h_∞) for m_{Na-rep} and h_{Na-rep} and effective curves as would be measured by voltage-clamp experiments.

$$\beta_{m,Na-rep} = 0.67 \exp\left(\frac{(-34 - V)0.5 \cdot 6 \cdot F}{RT}\right)$$

These are the rate functions for the inactivation variable, h, of I_{Na-rep} -

$$\alpha_{h,Na-rep} = 0.0023 \exp\left(\frac{(V + 42.50.83 \cdot -30 \cdot F}{RT}\right)$$

$$\beta_{h,Na-rep} = 0.0023 \exp\left(\frac{(-42.5 - V)0.17 \cdot 30 \cdot F}{RT}\right)$$

Figures 5.11 and 5.12 show the voltage dependence on the steady-state values and the time constants for the m_{Na-rep} and h_{Na-rep} variables.

5.9 Parameters of $I_{Na-tail}$

The key features that I defined for the proposed $I_{Na-tail}$ include significant activation only when there are prolonged spikes, e.g. when there is no

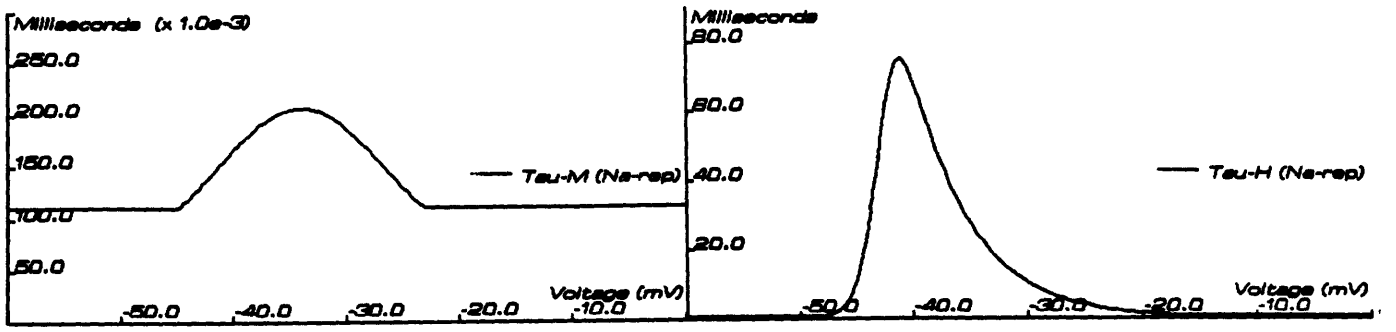


Figure 5.12: Time constant curves (τ_m and τ_h) for m_{Na-rep} and h_{Na-rep}.

repolarization due to non-linear outward currents. Also, this current was derived to be a transient current, with no contribution to steady state inward rectification.

5.9.1 Results

The steady state curve for m was adjusted so that activation commenced only for very depolarized levels (Figure 5.13.). On the other hand, the time constant for m was derived so that once m was open it was slow to relax to the closed state as the membrane repolarized (Figure 5.14.)

Determining the parameters for $\tau_{m,Na-tail}$ and $\bar{g}_{dens,Na-tail}$ was done together, since both of these factors determined the slow repolarization inward current.

The curves for h were not so critical - the main requirement was that at rest h was fully open so that $I_{Na-tail}$ could be turned on with the spike. However, h_∞ had to be 0 at levels depolarized from rest so that there would be no window current component from $I_{Na-tail}$. The curve for $\tau_{h,Na-tail}$ was set so that on one hand h did not change much during spiking, leaving the m variable in control of this current, and, on the other hand, fast enough so that $I_{Na-tail}$ would not have an apparent persistent characteristic because of a sluggish inactivation.

There was no need for the delayed state transition characteristics of more than one m or h particle for $I_{Na-tail}$; therefore the order of each was set to one.

Given that in general the requirements for $I_{Na-tail}$ were not as rigid

Gating Variable	z	γ	α_0	$V_{\frac{1}{2}}$ (mV)	τ_0 (ms)
m (activation)	8	0.95	0.025	-5.0	5.0
h (inactivation)	-6	0.2	0.0017	-47.0	3.0

Table 5.3: Parameters of $I_{Na-tail}$ Gating Variables

as other currents. i.e. I_{Na-rep}, the derived parameters were not the only set that would demonstrate the desired behavior. For example, h could be faster, as long as either τ_m was likewise decreased and/or $\bar{g}_{Na-tail}$ was increased to compensate for the resulting increase in inactivation of $I_{Na-tail}$ during the spike.

The equation for $I_{Na-tail}$ is -

$$I_{Na-tail} = \bar{g}_{Na-tail} m_{Na-tail} h_{Na-tail} (V - E_{Na+})$$

where

$$\bar{g}_{Na-tail} = 0.013\,\mu\text{S}$$

$$\bar{g}_{dens,Na-tail} = 1.0\,\text{mS/cm}^2$$

Table 5.3 lists the parameters for the $I_{Na-tail}$ gating variables. These are the rate functions for the activation variable, m, of $I_{Na-tail}$ -

$$\alpha_{m,Na-tail} = 0.025 \exp\left(\frac{(V+5)0.95 \cdot 8 \cdot F}{RT}\right)$$

$$\beta_{m,Na-tail} = 0.025 \exp\left(\frac{(-5-V)0.05 \cdot 8 \cdot F}{RT}\right)$$

These are the rate functions for the inactivation variable, h, of $I_{Na-tail}$

$$\alpha_{h,Na-tail} = 0.0017 \exp\left(\frac{(V+47)0.8 \cdot -6 \cdot F}{RT}\right)$$

$$\beta_{h,Na-tail} = 0.0017 \exp\left(\frac{(-47-V)0.2 \cdot -6 \cdot F}{RT}\right)$$

Figures 5.13 and 5.14 show the voltage dependence on the steady-state values and the time constants for the $m_{Na-tail}$ and $h_{Na-tail}$ variables.

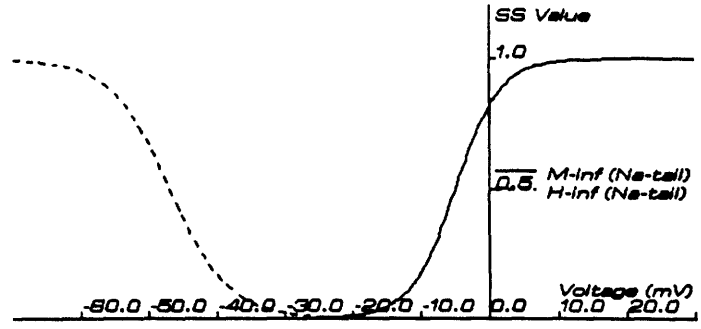


Figure 5.13: Steady-state curves (m_∞ and h_∞) for $m_{Na-tail}$ and $h_{Na-tail}$.

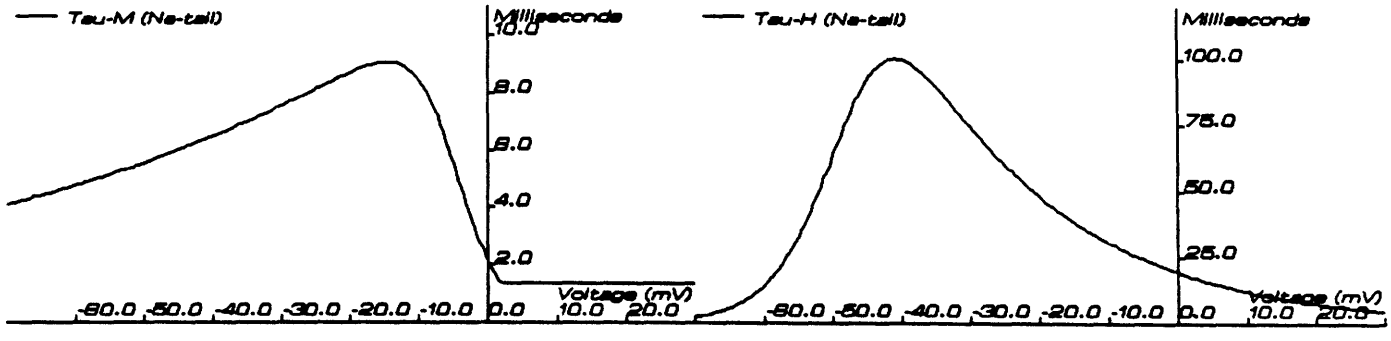


Figure 5.14: Time constant curves (τ_m and τ_h) for $m_{Na-tail}$ and $h_{Na-tail}$.

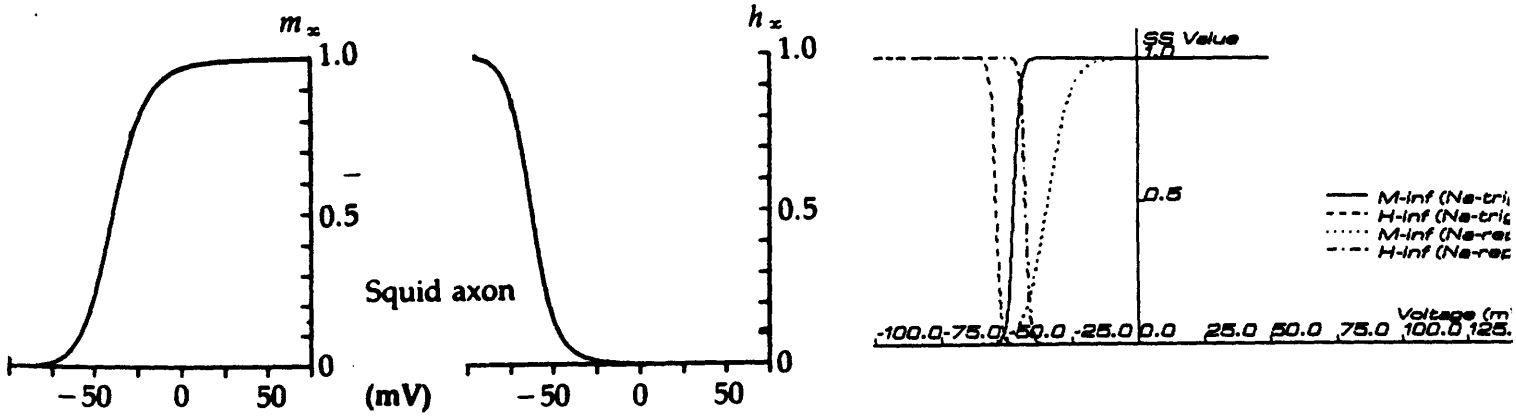


Figure 5.15: Steady-state curves (m_∞ and h_∞) for squid axon I_{Na} and the hippocampal $I_{Na-trig}$ and I_{Na-rep}.

5.10 Comparison of $I_{Na-trig}$ and I_{Na-rep} Kinetics With Those of Squid Axon I_{Na} and Rabbit Node of Ranvier I_{Na}

Comparing the characteristics of the squid axon I_{Na} kinetics with that of $I_{Na-trig}$ and I_{Na-rep} is interesting. Figures 5.15 and 5.16 illustrate the m and h steady-state and time constant curves for these three currents. The salient differences include the substantial overlap (giving a large window current) in the squid m_∞ and h_∞ curves and the much lower valence of the respective squid I_{Na} gating particles implied by these curves, as compared to the HIPPO curves.

5.11 Discussion of Functional Roles of the Proposed Na^+ Currents

Once we have constructed the three model currents that successfully reproduce the data, it is important to ask what roles these currents might play in the pyramidal cell. Consider $I_{Na-trig}$. The characteristics of this current allow for a sharp firing threshold from resting potential. The advantage of this is that the neuron is more tuned to a specific input firing level: there is a higher noise margin in regards to the firing efficacy of a given pattern of synaptic input. In addition, the lack of a window current for $I_{Na-trig}$ means that at rest or at subthreshold membrane potentials there will be

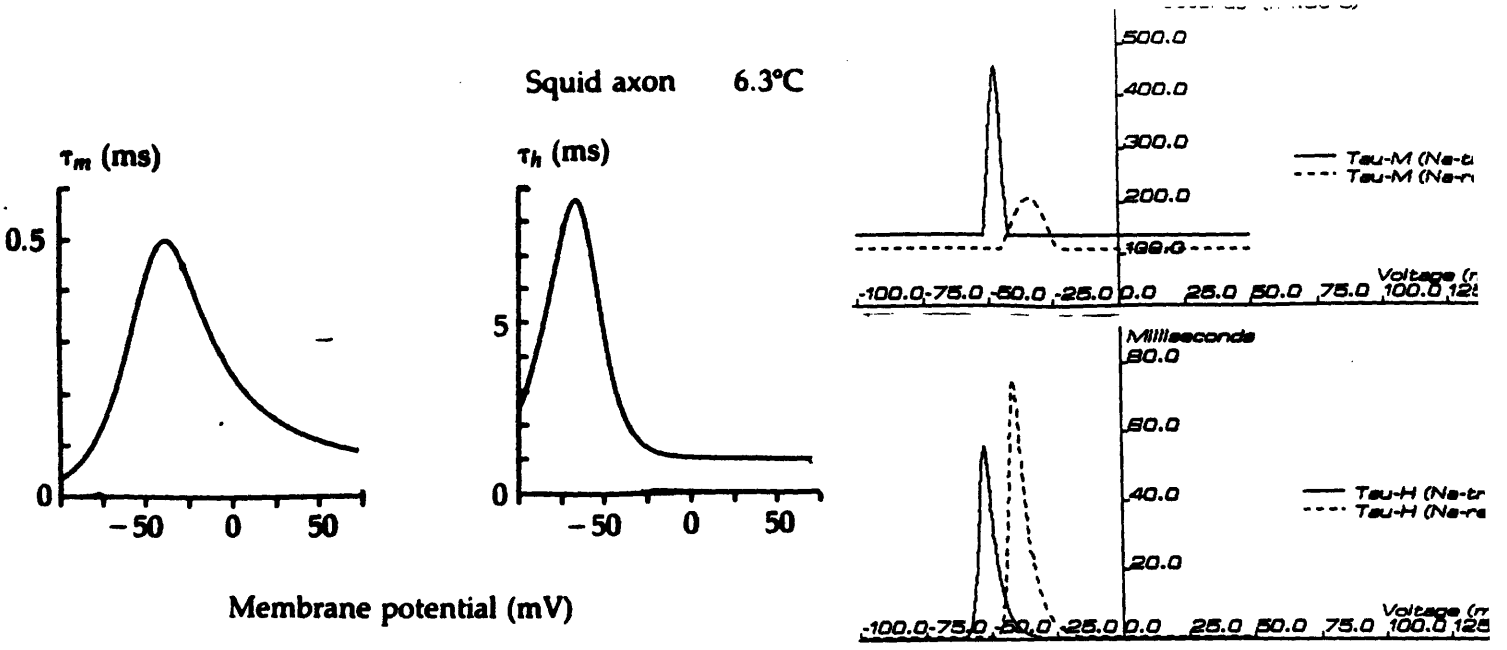


Figure 5.16: Time constant curves (τ_m and τ_h) for squid axon I_{Na} and the hippocampal $I_{Na-trig}$ and I_{Na-rep}.

little "wasted" Na^+ current. This is metabolically favorable as the cell does not have to remove the buildup of Na^+ resulting from such a background current. Likewise, any constant inward current at rest would have to be balanced by an outward (presumably K^+) current in order to maintain the resting potential. Again, this loss adds to the energy requirements of the cell at "rest".

Given these characteristics of $I_{Na-trig}$, a regenerative, higher threshold Na^+ current is necessary in order to mediate the higher threshold spikes that are observed under various conditions, including bursting on top of a (presumably) Ca^{2+} depolarizing hump, and repetitive Na^+-only firing.

What could be the advantage of this second Na^+ current? Such a higher threshold Na^+ current on top of a sharp, lower threshold Na^+ current could relax the requirements of the repolarization mechanism during a train of spikes in response to some tonic depolarization. An I_{Na-rep}-type current could mediate later action potentials without the requirement that the cell repolarize to below the threshold of an $I_{Na-trig}$-type current - all that is needed is that the cell repolarize to somewhere below the threshold of I_{Na-rep}. Simulation of repetitive firing (Figure 5.17) shows how I_{Na-rep} could furnish the major portion of depolarizing current for spikes after the first spike of a train.

Allowing the cell to fire again from a higher threshold reduces the amount of outward current needed to sustain multiple spikes, which in turn impose less of burden on the cell's machinery for maintaining the K^+ concentration gradient. In addition, the overlap of the activation and inactivation curves of

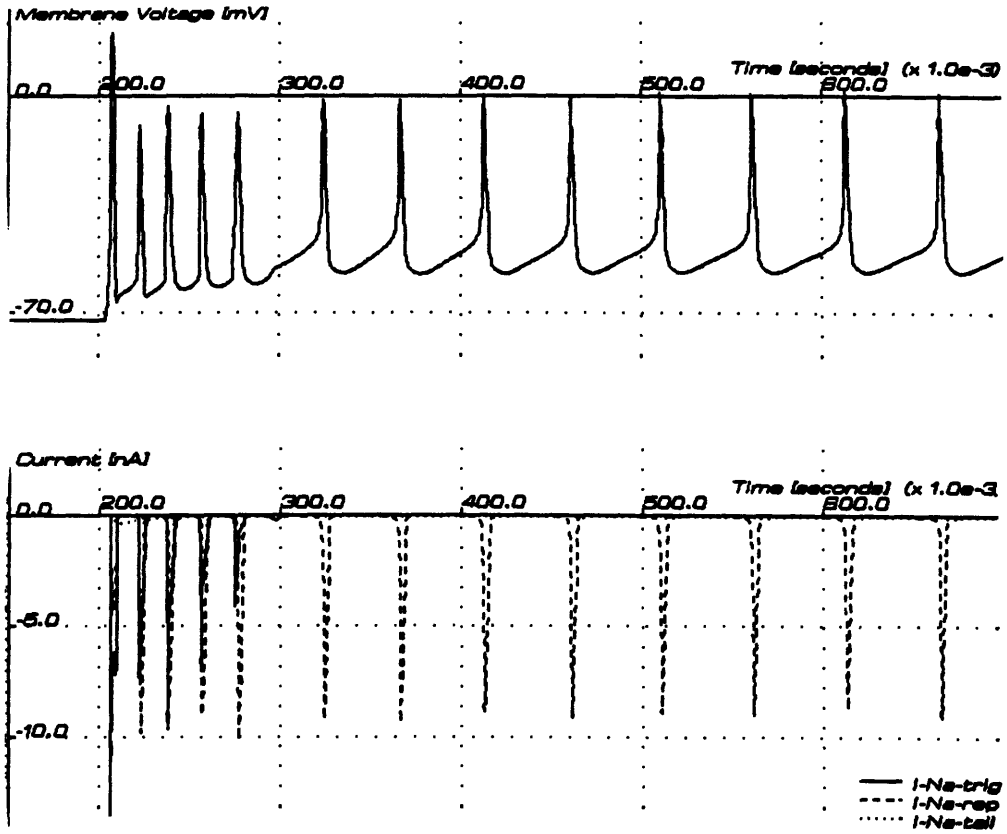


Figure 5.17: Current clamp simulation of normal repetitive firing in response to 1.0 nA tonic stimulus. Note the fast after-hyperpolarization (fAHP) after first spike, mediated by I_C (Storm, ibid) (Chapter 7). Slowing of firing is due to activation of I_{AHP} (Chapter 7). In lower part of figure are the $I_{Na-trig}$, $I_{Na-tail}$, I_{Na-rep} currents during the train.

I_{Na-rep} results in an ill-defined threshold for repetitive firing, allowing for a greater flexibility in modulating the frequency of firing by other mechanisms, e.g. distinct actions of the various K^+ currents.

On the other hand, when this current is blocked, there is a degenerate response to large tonic stimuli, as will be demonstrated in Chapter 9 (ref. Figure 9.9). As will be discussed later, whether this dependence of repetitive firing on I_{Na-rep} is physiological or pathological is not obvious.

What about the proposed $I_{Na-tail}$? As constructed, this current contributes to a small after-depolarization during a normal spike that must be countered by an outward current. In our simulations, this is accomplished by I_{DR}. For now the function this slowly-inactivating Na^+ current might have is not clear. Perhaps this current may be inhibited in certain circumstances, allowing it to play a role in mediating repetitive firing. Such speculation awaits further evidence of such a $I_{Na-tail}$ in actual cells. An important related question is whether or not the $I_{Na-tail}$ (if it exists) is either physiologically modulated by factors that do not affect the other currents, or is its role in shaping the response of the cell a constant one?

In Chapter 9 the effect of $I_{Na-tail}$ on repetitive firing will be compared with that of other currents.

Chapter 6

ESTIMATING Ca^{++} CURRENTS AND ACCUMULATION OF Ca^{2+} IN THE CELL

6.1 Introduction

This chapter describes the two calcium (Ca^{2+}) currents that have been described for the HPC, I_{Ca} and I_{CaS}, and possible mechanisms that establish the concentration of free Ca^{2+} in various regions underneath the cell membrane.

For the current version of the model the goals set for the characterization of the Ca^{2+} phenomena were quantitatively relatively modest and based partly on heuristics. In summary, the desired behavior of the system included :

- Generation of Ca^{2+}-only spikes that were qualitatively similar to actual Ca^{2+}-only spikes.
- Voltage and time-dependent changes in $[Ca^{2+}]$ underneath the membrane -$[Ca^{2+}]_{shell.1}$ and $[Ca^{2+}]_{shell.2}$ so as to mediate two K^+ currents (I_C and I_{AHP}).
- Response to voltage clamp protocols in qualitative agreement with the available data.

The effect of the change in $[Ca^{2+}]_{shell.1}$ on E_{Ca} was also considered, assuming that E_{Ca} is determined by the concentration gradient across the membrane in the vicinity of the I_{Ca} channels.

6.2 Calcium Current - I_{Ca}

Many workers have reported Ca^{2+} currents in HPC ([6], [13], [15], [18], [28], [33], [38], [40], [41], [53], [55]). The fast Ca^{2+} current in the model, I_{Ca}, which underlies Ca^{2+}-only spikes has kinetics similar to that of $I_{Na-trig}$, except that the curves for the gating variables are less steep and the time constants are about one order of magnitude slower. These kinetics were originally based on those used by Traub and Llinas [49], [48] in their hippocampal and motorneuron models.

In deriving the kinetics of I_{Ca} I attempted to reproduce current clamp records from cells in which both Na^+ currents and I_{DR} were blocked with TTX and TEA, respectively ([41]). In these cells slow Ca^{2+}-mediated "spikes" were elicited by long depolarizing current steps. Spike threshold was dependent on the holding potential prior to the current stimulus. Paradoxically, the higher the holding potential the lower the threshold. At the extreme , a holding potential of -70 millivolts resulted in elimination of a regenerative response after the stimulus (though some inward current was activated during the stimulus). On the other hand, a holding potential of -40 millivolts resulted in a firing threshold for the Ca^{2+} spike of about -30 millivolts. This behavior is contrary to what might be expected from a current with activation/inactivation properties similar to a fast Na^+ current, in which case a lower holding potential would cause the inactivation to be more completely removed, thereby lowering the firing threshold. Segal and Barker proposed that the observed behavior of the Ca^{2+} spike was due to the action of the transient K^+ current I_A (Chapter 7.); when the cell was held at the lower potential, the inactivation of I_A was removed so that the subsequent depolarization allowed the activation I_A to counter the activation of I_{Ca}. Holding the cell at the higher potential inactivated I_A, thereby allowing the later depolarizing current pulse to elicit the Ca^{2+} spike. The formulation for the kinetics of I_{Ca} was therefore tied somewhat to the description of I_A in the model.

Another action of I_{Ca} that I attempted to reproduce was its apparent role in the slow depolarizing hump that is observed in some cells which exhibit burst firing ([48]), as I mentioned in the previous chapter in the dis-

cussion of I_{Na-rep} function during repetitive and burst firing. At this point the model does not exhibit such behavior. In fact, such a hump between -60 and -40 millivolts is inconsistent with the apparent I_{Ca} (slow) activation at approximately -40 millivolts. The supposed Ca^{2+}-mediated hump is possibly due to I_{Ca} channels in the dendrites, rather than somatic I_{Ca}. In the dendrite current input local or distal to the site of the I_{Ca} channels could activate the channels without raising the soma voltage beyond 10 millivolts or so above rest. Once activated, the dendritic Ca^{2+} conductances could supply enough long-lasting inward current to cause the somatic hump in question. In future studies with HIPPO, such conductances will be placed on the dendrites to test this hypothesis (ref. Chapter 11).

Another requirement for the kinetics of I_{Ca} was that this current not be significantly activated during the normal action potential. This is based on the assumption that the effect of Ca^{2+} blockers on the shape of the action potential is due mainly to the subsequent inhibition of I_C and I_{AHP}. This was accomplished by including two activation particles, s, in order to force a delay in activation with depolarization, and likewise adjusting the curves for s_∞ and τ_s so that during the regular spike s would change little, while during the sustained depolarization required to elicit the Ca^{2+} spike s would have enough time to move to the open position.

In addition, it was necessary to set the order of s to three so that subthreshold activation of s during regular spikes did not allow significant (in terms of membrane depolarization) I_{Ca}.

An important characteristic of Ca^{2+} spikes is the abruptly-biphasic repolarization (see [41]). The initial decay after the peak of the spike is relatively slow, presumably due to residual I_{Ca}, until the membrane potential reaches about -10 millivolts. The membrane then rapidly repolarizes to the resting (or holding) potential, as if I_{Ca} was suddenly turned off.

Since this knee occurs well depolarized from the spike threshold (between -40 and -30 millivolts), it cannot be due to complete de-activation of the activation gating particle (s) that underlies the threshold.

Also, I was not able to adjust either the number of nor the kinetics of the inactivation particle (w) so that a delayed yet abrupt inactivation could account for the knee. However, by adjusting the steepness of the s_∞ curve so that the effective steady state activation (in the hyperpolarizing direction) began to drop off around -5 millivolts, the start of de-activation as the Ca^{2+}-only spike repolarized to this level contributed a moderate knee in the simulated spike. With the present version of I_{Ca}, simulated Ca^{2+} spikes (Figure 6.1 have an analogous repolarization knee, but this is not as steep

Gating Variable	z	γ	α_0	$V_{\frac{1}{2}}$ (mV)	τ_0 (ms)
s (activation)	4	0.5	0.10	-24.0	2.0
w (inactivation)	-12	0.2	0.001	-35.0	5.0

Table 6.1: Parameters of I_{Ca} Gating Variables

and does not occur quite as depolarized from the spike threshold as some of the reported Ca^{2+}-only spikes.

In these simulations inactivation of the w variables contribute to the repolarization knee. Future versions of the I_{Ca} description may include either more than one inactivation or activation gating variables, or may use a gating variable with a more complicated state domain (e.g. more than two stable states). Also to be considered is the possibility that in these reports un-blocked outward currents also are involved, particularly because different data suggest that the repolarizing phase of Ca^{2+}-only spikes is quite long and without the described knee (Storm, personal communication).

With present description of $I_C$alc and Ca^{2+} accumulation underneath the membrane the amount of Ca^{2+} that flows across the membrane during regular action potentials changes E_{Ca} by at most 20 millivolts (see Figure 6.8. The Ca^{2+} influx during the pure Ca^{2+} spike, however, is enough to change E_{Ca} so that at the peak of the spike E_{Ca} drops to about 10 millivolts (ref. Figure 6.8). The reduction of E_{Ca} during Ca^{2+}-only spikes is a contributor to the reduction of I_C , and in fact is the limiting factor as to the magnitude of the Ca^{2+}-only spike. These results suggest that measurement of E_{Ca} [1] during Ca^{2+}-only spikes can help validate the description of Ca^{2+} -accumulation underneath the membrane described here or suggest alternative descriptions.

The equation for I_{Ca} is -

$$I_{Ca} = \bar{g}_{Ca} s_{Ca}^2 w_{Ca}^4 (V - E_{Ca})$$

where

$$\bar{g}_{Ca} = .64\,\mu\text{S}$$

[1] For example by using hybrid clamp protocol in which the reversal potential for the spike current is measured at different points of a Ca^{2+}-only spike by switching from current clamp to voltage clamp.

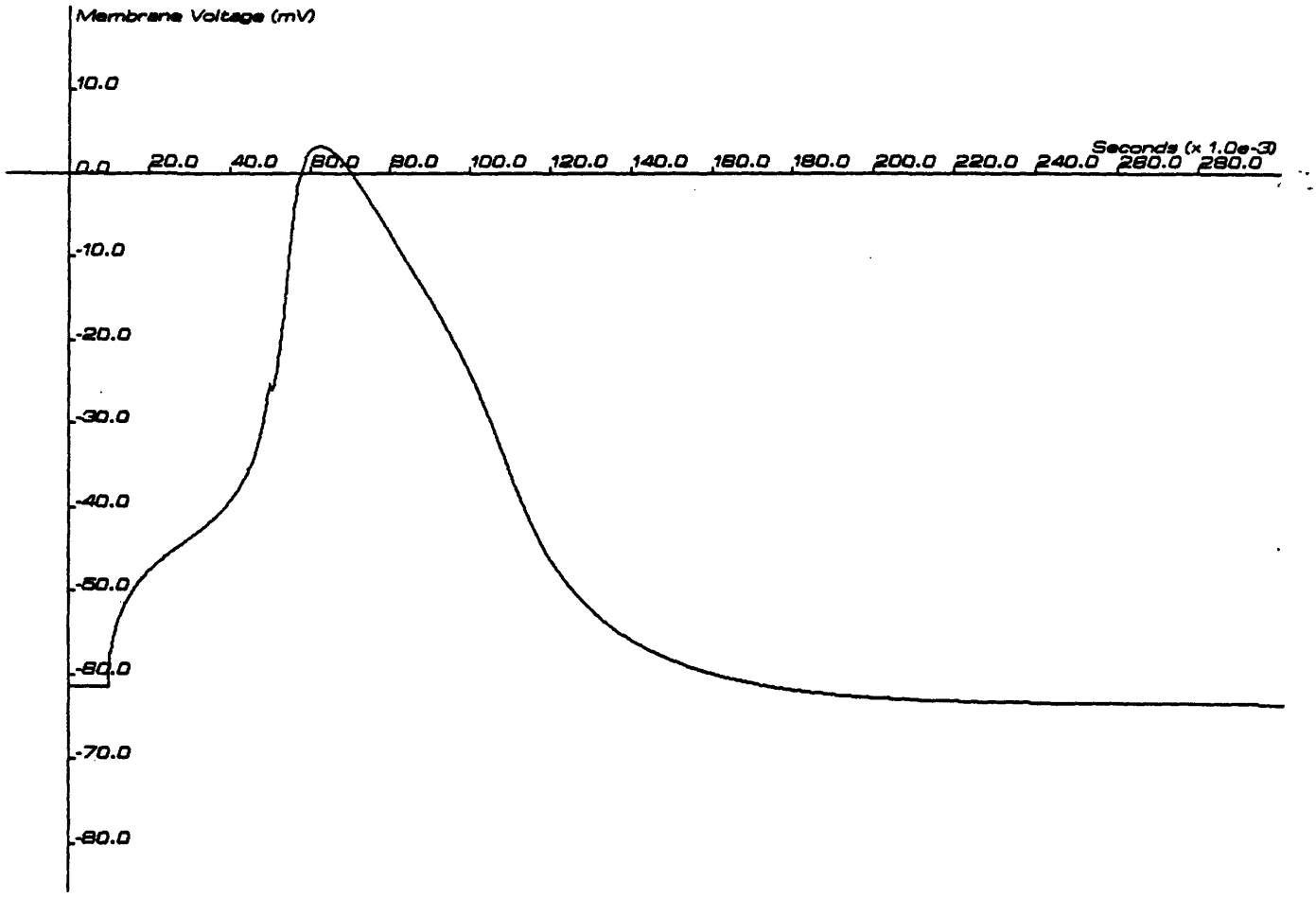


Figure 6.1: Current clamp simulation of Ca^{2+} spike. Non-linear currents include I_{Ca} I_{AHP}, and I_A.

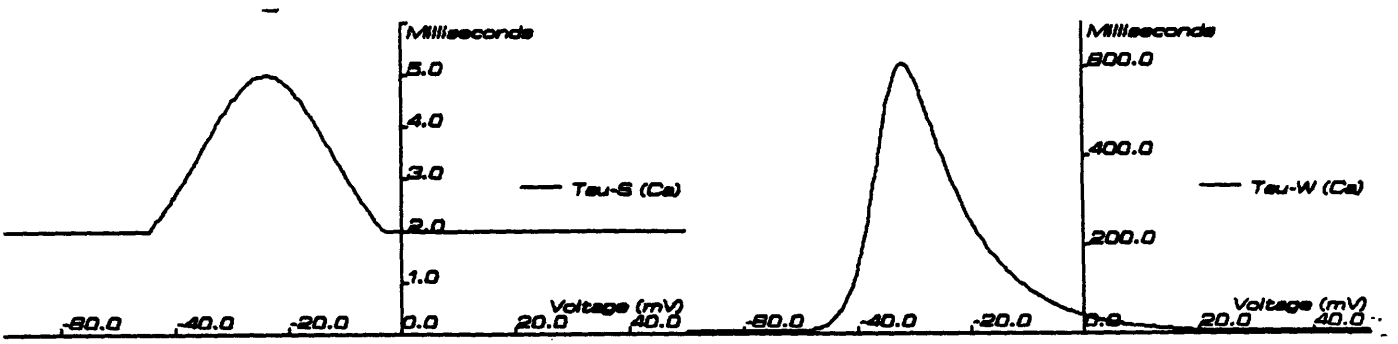


Figure 6.2: Steady-state curves (x_∞ and w_∞) for s_{Ca} and w_{Ca} and effective curves as would be measured by voltage-clamp experiments.

Table 6.1 lists the parameters for the I_{Ca} gating variables. These are the rate functions for the activation variable, s, of I_{Ca} -

$$\alpha_{s,Ca} = 0.1 \exp\left(\frac{(V + 24)0.5 \cdot 4 \cdot F}{RT}\right)$$

$$\beta_{s,Ca} = 0.1 \exp\left(\frac{(-24 - V)0.5 \cdot 4 \cdot F}{RT}\right)$$

These are the rate functions for the inactivation variable, w, of I_{Ca} -

$$\alpha_{w,Ca} = 0.001 \exp\left(\frac{(V + 35)0.2 \cdot -12 \cdot F}{RT}\right)$$

$$\beta_{w,Ca} = 0.001 \exp\left(\frac{(-35 - V)0.8 \cdot -12 \cdot F}{RT}\right)$$

Figure 6.2 and Figure 6.3 show the voltage dependence on the steady-state values and the time constants for the x_{CA} and y_{CA} variables.

6.3 Slow Calcium Current - I_{CaS}

I_{CaS} is a slow, non-inactivating current (e.g. [28]). While it has been reported that this current is a true Ca^{2+} current, careful examination of the data for I_{CaS} suggests that the reversal potential for this current is around 0 millivolts, implying that I_{CaS} is a mixed carrier current.

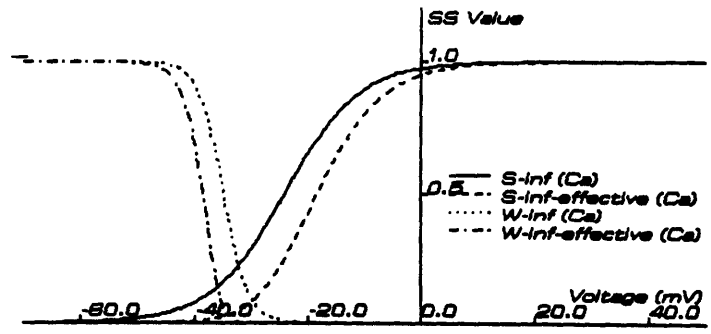


Figure 6.3: Time constant curves (τ_s and τ_w) for s_{Ca} and w_{Ca}.

Gating Variable	z	γ	α_0	$V_{\frac{1}{2}}$ (mV)	τ_0 (ms)
x (activation)	25	0.5	4.0	-30.0	0.1

Table 6.2: Parameters of I_{CaS} Gating Variables

The small conductance of this current (0.08 μS), combined with the slow onset of its activation variable x (τ_x is reported to range from 50 to 100 milliseconds) suggest that I_{CaS} has only a small functional role during repetitive firing. At this stage of the model, such a role has not been demonstrated.

The equation for I_{CaS} is -

$$I_{CaS} = \bar{g}_{CaS} x_{CaS} (V - E_{CaS})$$

where

$$E_{CaS} = 0\text{millivolts}$$

$$\bar{g}_{CaS} = 0.08\,\mu\text{S}$$

Table 6.2 lists the parameters for the I_{CaS} gating variables.

These are the rate functions for the activation variable, x, of I_{CaS} -

$$\alpha_{x,CaS} = 4.0 \exp\left(\frac{(V + 30)0.5 \cdot 25 \cdot F}{RT}\right)$$

$$\beta_{x,CaS} = 4.0 \exp\left(\frac{(-30 - V)0.5 \cdot 25 \cdot F}{RT}\right)$$

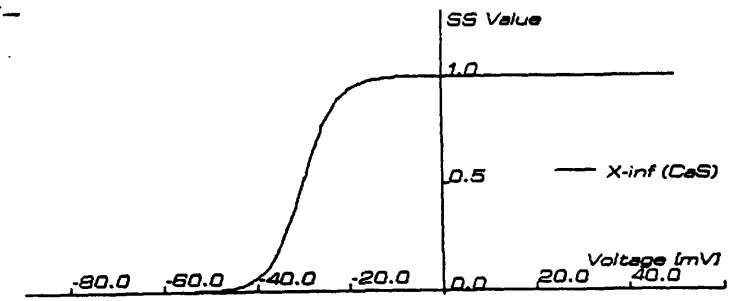


Figure 6.4: Steady-state curve (x_∞ for x_{CaS}.

Figure 6.4 show the voltage dependence on the steady-state value for the x_{CaS} variable.

6.4 Mechanisms Regulating $[Ca^{2+}]_{shell.1}$ and $[Ca^{2+}]_{shell.2}$

To recapitulate, there were three reasons to consider the accumulation of Ca^{2+} underneath the membrane as a result of the influx of Ca^{2+} currents:

1. Activation of I_{AHP} is presumed to be dependent on the influx of I_{Ca}.
2. Activation of I_C is presumed to be dependent on the influx of I_{Ca}.
3. The very low resting value of $[Ca^{2+}]_{in}$ (typically assumed to be about 50nM) and the low resting value of $[Ca^{2+}]_{out}$ (on the order of a few mM) implies that the influx of Ca^{2+} from I_{Ca} can significantly change the ratio of the extra-cellular and intra-cellular $[Ca^{2+}]$, changing E_{Ca}, resulting in negative feedback via reduction of the driving force for the Ca^{2+} currents.

For the activation of I_{AHP} and I_C , the observed Ca^{2+} dependence is assumed to involve some mechanism between free intracellular Ca^{2+} and the individual channels [2] The simple relationship that is used in the present model assumes that activation of both I_{AHP} and I_C is (partially) dependent on Ca^{2+} -binding gating particles in these two types of channels. The

[2] Many versions of this mechanism have been proposed ([19]). In this study a fairly simple mechanism is employed.

binding of Ca^{2+} to the gating particles is reversible and the behavior of the particles obey first order kinetics (Chapter 7).

The time course of I_C and I_{AHP} set some constraints on the kinetics of Ca^{2+} accumulation underneath the soma membrane. As shall be discussed in detail in Chapter 7, Ca^{2+}-mediated activation and inactivation of I_C must be sudden and complete, in accordance with the sudden onset of I_C and the apparent removal of I_C prior to subsequent spikes in a spike train. Given the sigmoidal relationship between the Ca^{2+} -dependent gating particles and the log of the concentration of Ca^{2+} (ref. Chapter 7, Figure X), this means that the $[Ca^{2+}]$ that mediates I_C must rise and then fall quickly with every spike. On the other hand, Ca^{2+}-mediated activation of I_{AHP} is gradual, getting stronger with each spike in a train, and then gradually decaying over one second or longer.

In order to accommodate these two patterns of Ca^{2+}-mediated behavior, a two-region shell, single core model was developed. In this model both I_{Ca} and I_C channels communicate with a distinct part of a shell underneath the soma surface, *shell*.1. I_{AHP} channels, on the other hand, communicate directly with the remainder of the soma shell, *shell*.2. Ca^{2+} flows between the two shell regions and between each shell region and the soma core by simple diffusion.

The physical relationship between the different soma shell regions, the relevant channels, and the soma core is illustrated in Figure 6.5. Figure 6.6 shows a view of the soma membrane surface illustrating the proposed segregation of Ca^{2+} channels and I_C and I_{AHP} channels. Figure 6.7 shows the compartmental model based on this arrangement which is used to determine the concentration of Ca^{2+} in the shell regions.

The model therefore includes a shell of thickness d_{shell} on the intracellular face of the membrane. A portion of this shell is assigned to *shell*.1 and the remainder is assigned to *shell*.2. The concentration of free Ca^{2+} in *shell*.1, $[Ca^{2+}]_{shell.1}$, is a function of the two Ca^{2+} currents, I_{Ca} and I_{CaS} and movement of Ca^{2+} between *shell*.1 and *shell*.2 and between *shell*.1 and the core. Likewise, the concentration of free Ca^{2+} in *shell*.2 is determined by the flow of Ca^{2+} between *shell*.1 and *shell*.2 and between *shell*.2 and the core. The concentration of Ca^{2+} in the core is assumed to be a constant since the volume of the core is much larger than the volume of the two shells.

The movement of Ca^{2+} between the three compartments can be described as follows. Let X_1, X_2, and X_3 equal the amount of Ca^{2+} (nanomoles) in *shell*.1 (compartment 1), *shell*.2 (compartment 2), and the core (compart-

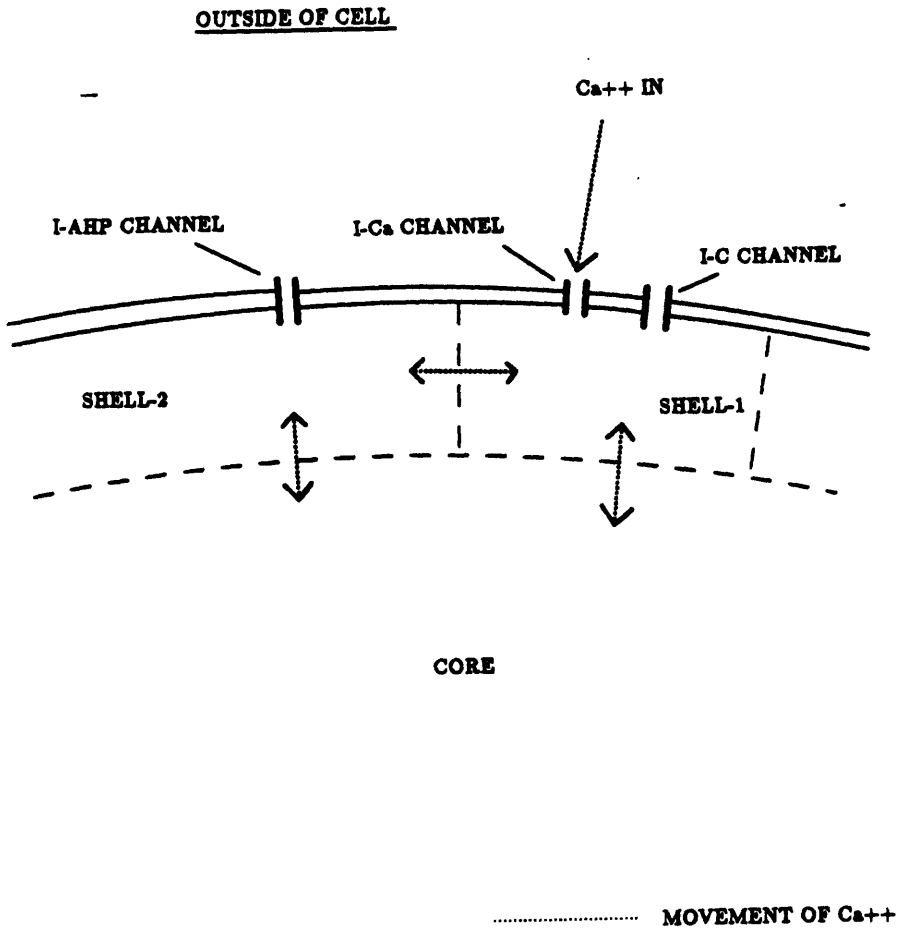


Figure 6.5: Diagram of localization of I_{Ca}, I_{CaS}, I_C, and I_{AHP} channels in distinct regions of the soma membrane, as postulated by the model. This scheme assumes that the I_{Ca}, I_{CaS}, and I_C channels are all in close proximity (i.e. *shell*.1), such that the immediate change in $[Ca^{2+}]$ in the vicinity of the Ca^{2+} current channels when these channels conduct is sensed by the I_C channels. Likewise, the I_{AHP} channels are assumed to reside in a relatively distant area of the soma membrane, such that the rise in local $[Ca^{2+}]$ around these channels is delayed from the onset of the Ca^{2+} currents.

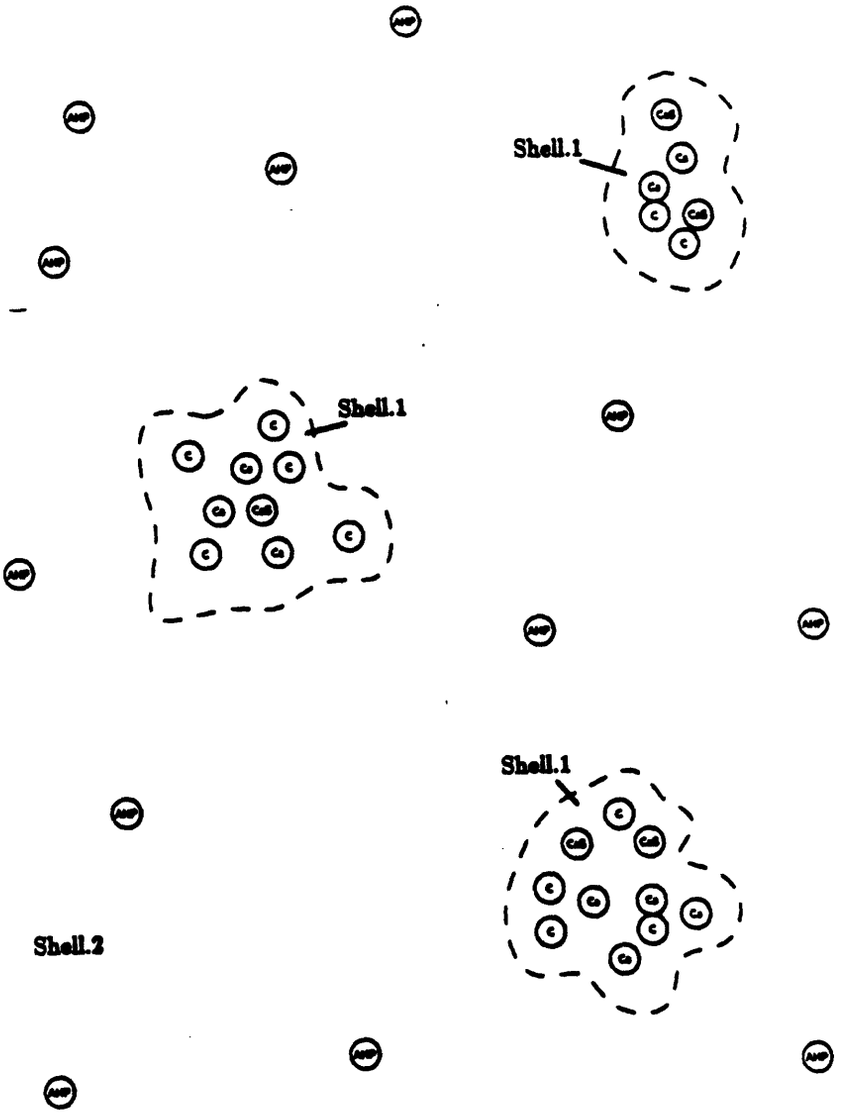


Figure 6.6: Proposed segregation of I_{Ca}, I_{CaS}, I_C, and I_{AHP} channels as seen looking onto the surface of the soma. The effective area for diffusion between the two shell regions, A_{12}, is determined by the total length of the dotted line in the figure as well as the depth, d_{shell}, of the shell. In the model A_{12} was lumped with D_{12} to yield an effective diffusion constant for the entire flow between the regions (see text). The empirical adjustment of this metric to give the desired kinetics is then equivalent to adjusting this length (i.e. the amount of communicating surface area). Also, the dotted line does not represent a distinct physical barrier but rather a boundary for approximating the continuous diffusion case with compartments.

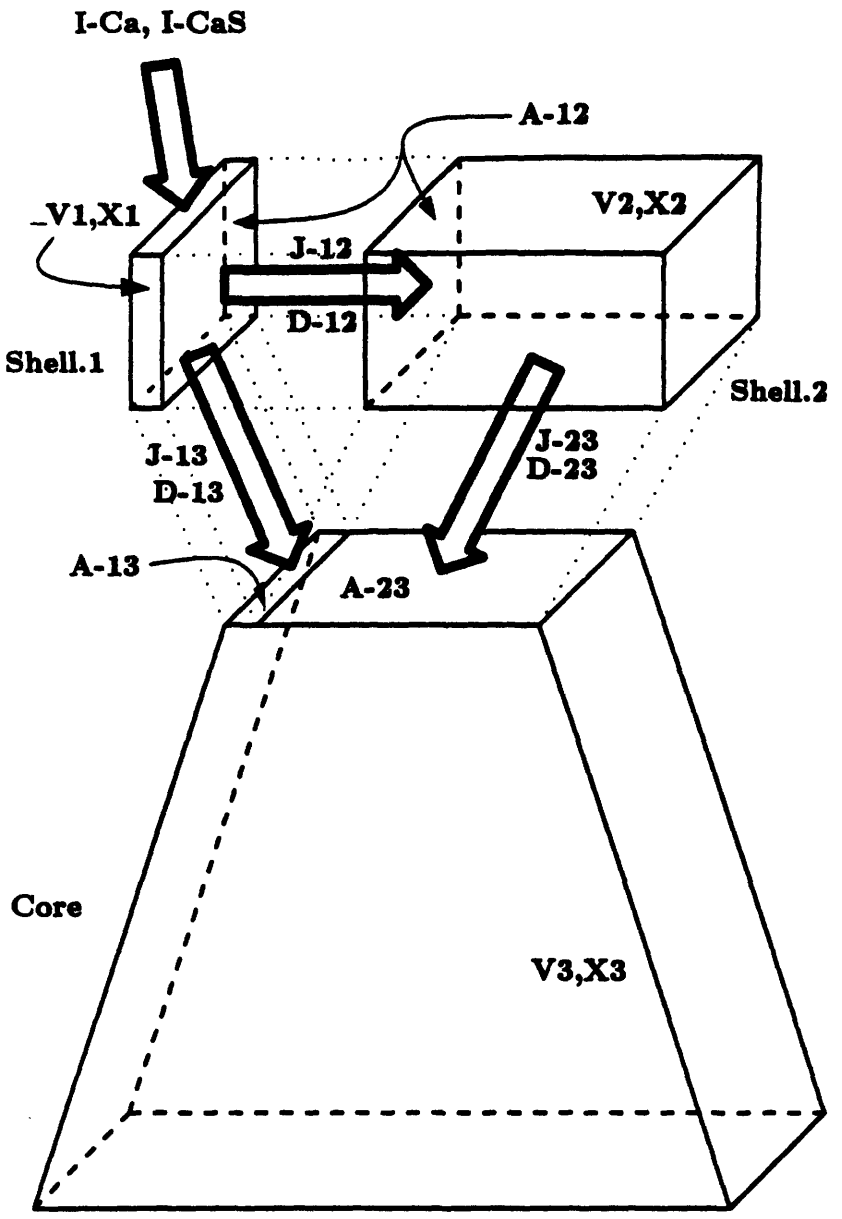


Figure 6.7: 3-compartment model of Ca^{2+} influx and accumulation, based on structure suggested in Figures 6.5 and 6.6. Parameters of this model are given in the text.

ment 3). respectively. In the same manner let V_1, V_2, and V_3 be the volumes (ml) and C_1, C_2, and C_3 be the concentrations of Ca^{2+} (mM) in the three compartments. Let J_{12}, J_{13} and J_{23} be the flux of Ca^{2+} (nanomoles per second per square cm) between *shell*.1 and *shell*.2. *shell*.1 and the core, and *shell*.2 and the core, respectively, and let D_{ij} be the diffusion constant (cm per second) for the flux J_{ij}. The area for Ca^{2+} diffusion between any two compartments i and j is given by A_{ij} (square cm).

The change in the amount of Ca^{2+} in each of the compartments is as follows:

$$\begin{aligned}
\dot{X}_1 &= -J_{12}A_{12} - J_{13}A_{13} - \frac{I_{Ca} + I_{CaS}}{2 \times 10^3 F} \\
\dot{X}_2 &= J_{12}A_{12} - J_{23}A_{23} \\
\dot{X}_3 &= J_{13}A_{13} + J_{23}A_{23}
\end{aligned}$$

where F is Faraday's constant and the currents are in nano-amps. The two Ca^{2+} currents contribute only to the change in the amount of Ca^{2+} in *shell*.1. There is a factor of 2 in the Ca^{2+} current term since each Ca^{2+} ion carries two charges, and there is a minus sign preceding this term, since the inward currents are defined as negative.

The flux of Ca^{2+} from compartment i to compartment j is given by Fick's law, as follows:

$J_{ij} = D_{ij}(C_i - C_j)$

Since the concentration in compartment i, C_i, is given by X_i/V_i, then, incorporating Fick's law, the time derivative of the concentration of each compartment is as follows:

$$\begin{aligned}
\dot{C}_1 &= \frac{\dot{X}_1}{V_1} \\
&= \frac{1}{V_1}\left(-A_{12}D_{12}(C_1 - C_2) - A_{13}D_{13}(C_1 - C_3) - \frac{I_{Ca} + I_{CaS}}{2 \times 10^3 F}\right) \\
\dot{C}_2 &= \frac{\dot{X}_2}{V_2} \\
&= \frac{1}{V_2}\left(A_{12}D_{12}(C_1 - C_2) - A_{23}D_{23}(C_2 - C_3)\right) \\
\dot{C}_3 &= \frac{\dot{X}_3}{V_3}
\end{aligned}$$

$$= \frac{1}{V_3}\Big(A_{13}D_{13}(C_1 - C_3) + A_{23}D_{23}(C_2 - C_3)\Big)$$

The volume of the shell is determined by d_{shell} and the surface area of the soma. *shell*.1 is set to cover 0.1% of the soma surface, with *shell*.2 comprising the remainder of the surface. If A equals the surface area of the soma (square cm), the areas for Ca^{2+} flow between each shell region and the core are given by:

$$\begin{aligned} A_{13} &= 0.001A \\ A_{23} &= 0.999A \end{aligned}$$

The volume of each shell region is given by:

$$\begin{aligned} V_1 &= A_{13}d_{shell} \\ V_2 &= A_{23}d_{shell} \end{aligned}$$

The volume of the core is set equal to the soma volume, since d_{shell} is much smaller than the soma radius.

D_{13} and D_{23} are equal, since each shell region is assumed has the same proximity to the core. Let

$D_{13} = D_{23} = D_{sh-cr}$

D_{12} can be considered as equal to the previous two diffusion constants without any loss of generality since the value for A_{12} may be adjusted to allow the shells to equilibrate much faster with each other than with the (low concentration) of the core. This area is the effective diffusion area for Ca^{2+} between the (intertwined) regions of the shell. For convenience, let us define

$D'_{sh-sh} = A_{12}D_{12}$

The previous expressions can now be used to give the following equations for $\dot{C}_1$, $\dot{C}_2$, and $\dot{C}_3$:

$$\begin{aligned} \dot{C}_1 &= \frac{1}{d_{shell}}\left[-\frac{D'_{sh-sh}}{0.001A}(C_1 - C_2) - D_{sh-cr}(C_1 - C_3) + \frac{I_{Ca} + I_{CaS}}{0.001A \times 2 \times 10^3 F}\right] \\ \dot{C}_2 &= \frac{1}{d_{shell}}\left[\frac{D'_{sh-sh}}{0.999A}(C_1 - C_2) - D_{sh-cr}(C_2 - C_3)\right] \\ \dot{C}_3 &= \frac{D_{sh-cr}}{V_3}\Big[A_{13}(C_1 - C_2) + A_{23}(C_2 - C_3)\Big] \end{aligned}$$

Now we assume that

$$\frac{V_3}{AD_{sh-cr}} >> 1 \text{ second}$$

so

$\dot{C}_3 \approx 0$

and C_3 is set to a constant 50 nM. At each time step $[Ca^{2+}]_{shell.1}$ and $[Ca^{2+}]_{shell.2}$ are calculated by integrating the above differential equations.

The relevant parameters were adjusted so that, given the previously estimated kinetics for I_{Ca}, during single and repetitive firing the concentrations of Ca^{2+} in the two sub-membrane compartments had the time courses and relative magnitudes discussed earlier in this section. An additional constraint was that $[Ca^{2+}]_{shell.1}$ could not change so much during either normal action potentials or, especially, Ca^{2+} -only spikes so that E_{Ca} would be reduced too quickly, wiping out the Ca^{2+} driving force before the spike was complete.

The following parameters satisfied the reported constraints:

$$d_{shell} = 0.25\mu\text{m}$$

$$D'_{sh-sh} = 2.0 \times 10^{-11}(\text{cm}^3/\text{millisecond})$$

$$D_{sh-cr} = 4.0 \times 10^{-7}(\text{cm/millisecond})$$

The remaining parameters needed to calculate the concentration of intracellular Ca^{2+} derive directly from the previously presented soma dimensions and the Ca^{2+} current kinetics.

This description is somewhat similar to that used in other modelling studies ([48], [2]), in particular the idea that local accumulation of Ca^{2+} in a limited space underneath the membrane can mediate other processes, and that the kinetics of the Ca^{2+} in this region is governed by first-order mechanisms.

Figures 6.8 shows how the concentration of Ca^{2+} changes in the two shell regions during a single action potential. Figures 9.19, 9.23, and 9.24 show how the concentration of Ca^{2+} changes in the two shell regions during a train of action potentials. Again in Chapter 7 the relationship between these concentrations and the activation of I_C and I_{AHP} will be defined in more detail.

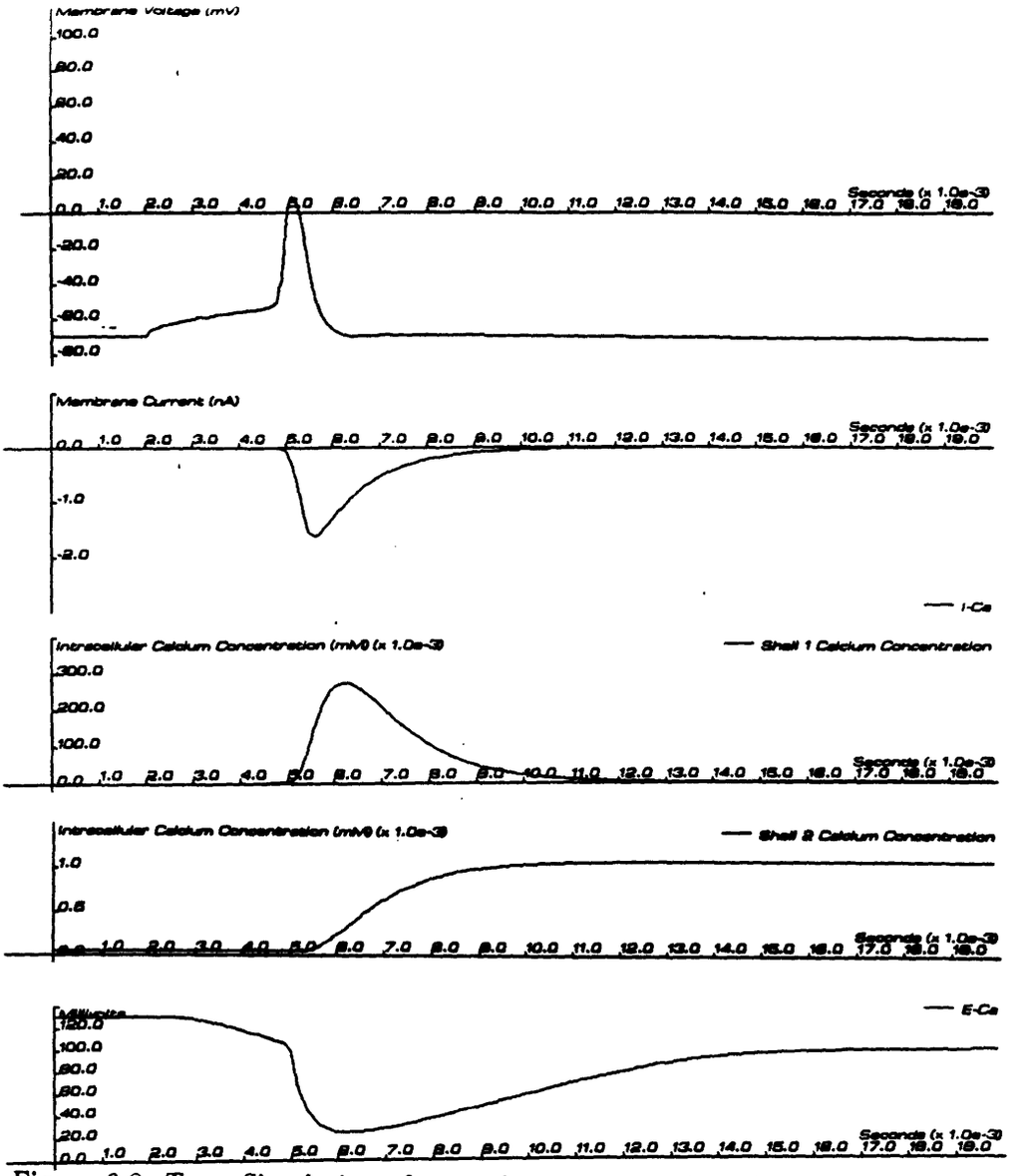


Figure 6.8: Top - Simulation of normal action potential. Top Middle - I_{Ca} during spike. Bottom Middle - $[Ca^{2+}]_{shell.1}$ during spike. Bottom - E_{Ca} during spike.

6.5 Calculation of E_{Ca}

As mentioned earlier E_{Ca} was calculated at each time step from the Nernst equation, using the current $[Ca^{2+}]_{shell.1}$ and the fixed $[Ca^{2+}]_{out}$ as the relevant concentrations for the E_{Ca} equation. The change in $[Ca^{2+}]_{shell.1}$ and E_{Ca} during a single action potential is illustrated in Figure 6.8. $[Ca^{2+}]_{shell.1}$ and E_{Ca} during a Ca^{2+}-only spike is shown in Figure 6.9. During the Ca^{2+}-only spike the subsequent fall of E_{Ca} contributes to the reduction of I_{Ca}.

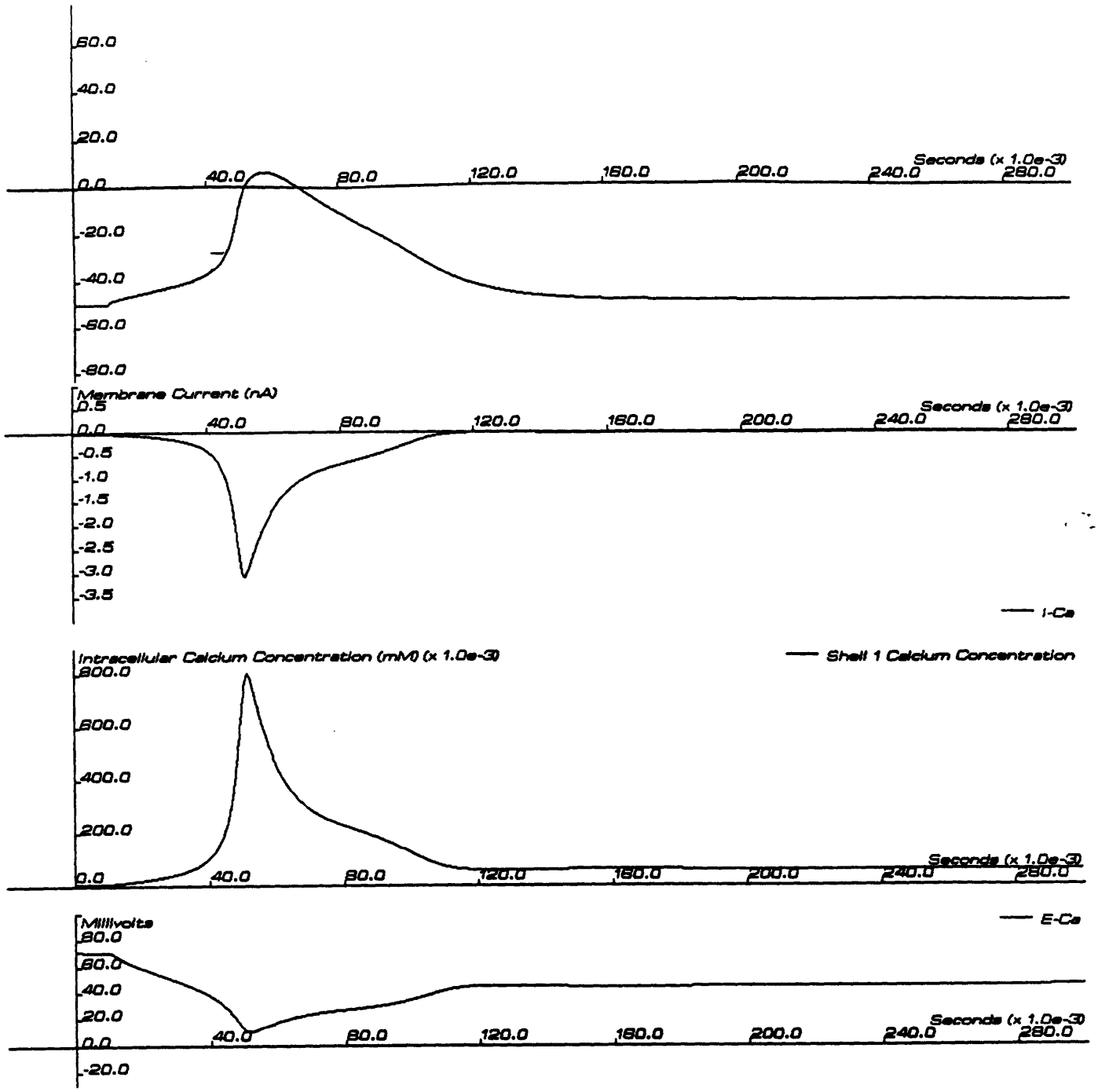


Figure 6.9: Top - Simulation of Ca^{2+} -only spike. Top Middle - I_{Ca} during spike. Bottom Middle - $[Ca^{2+}]_{shell.1}$ during spike. Bottom - E_{Ca} during spike.

6.6 Discussion

The Ca^{2+} system parameters described here are highly speculative but are based on valid physical mechanisms. The net result is that the model successfully reproduces a wide variety of Ca^{2+}-related behavior. Many of the parameters were developed in parallel with the development other parameters (e.g. K^+ current parameters, in particular those defining I_C and I_{AHP}), and this interdependence constrains the overall problem.

For example, including the two shell regions may appear somewhat artificial; yet given the nature of I_{Ca} [3] (as determined by Ca^{2+}-only spikes and other relatively independent evidence) the characteristics of these compartments are constrained by (a) the dimensions of the soma; (b) the amount of I_{Ca} entering the cell during Ca^{2+}-only spikes, which in turn effects E_{Ca}, providing negative feedback; (c) the amount of I_{Ca} entering the cell during regular action potentials; and (d) the *a prior* Ca^{2+}-mediated characteristics of I_C and I_{AHP}.

In sum, modification of any one parameter typically resulted in a widespread effect due to the numerous feedback loops in the system, and these loops helped to constrain the overall modelling problem. Clearly alternative mechanisms may be suggested for the model features described here (e.g. more complicated kinetics for the Ca^{2+}-mediated gating particles of I_C or I_{AHP}), but at the very least such alternatives would have to be as physically plausible as those suggested here and would also be subject to the same constraints, since these constraints are inherent in the system being modelled.

[3] In the results presented here the contribution of I_{Ca2} to model behavior is minimal.

Chapter 7

ESTIMATING K^+ CURRENTS

7.1 Introduction

This chapter presents the six K^+ currents in the model - I_{DR}, I_A, I_C, I_{AHP}, I_M, and I_Q. We begin by reviewing the strategy for evaluating the K^+ currents data, and the guidelines that constrain the development of the model descriptions. Next, the classical "Delayed Rectifier" K^+ current, I_{DR}, and the so-called "A" K^+ current, I_A, are described. Following this, a brief description of the Ca^{2+}-mediated w gating particles incorporated in the model description of I_C and I_{AHP} is presented, followed by the discussion of these two Ca^{2+}-mediated K^+ currents. The chapter closes with descriptions of two more K^+ currents, I_M and I_Q. In this chapter the action of each of the K^+ currents on specific features of the single spike and/or repetitive spikes will be demonstrated, primarily with comparisons between simulations and data.

7.2 Review of Strategy for Evaluating K^+ Currents

As described in Chapter 2, forming a plan for building the model was not trivial, given that the quality of data for the currents varied greatly and that the action of some currents was mainly seen in concert with other currents, thereby complicating the parameter estimation for a (presumably) unique

conductance.

The data for K^+ currents ranges from complete to marginal, in terms of the voltage-dependence of each current's kinetics. the absolute magnitude of each conductance, and the relationship of a given current with other factors (in particular intracellular Ca^{2+}). In addition, for some currents (e.g. I_A and I_C). although a plethora of data may be available much of it is inconsistent with each other. It was very difficult to sort through this body of information and decide what data could be applied, which should be discounted, and what assumptions should be used to fill in the gaps. Often consultation with Drs. Adams and Storm provided some insight for this problem.

In order to make progress a set of references had to be chosen as a "gold standard", particularly when data from different reports were inconsistent. The primary standard that I used was the data from Storm, 1986. Using this data as a first reference had the advantage that I could both examine the original data of Dr. Storm's and, when necessary, obtain insights from him as to the implications of the data. In this chapter and others many of the comparisons between simulation and experimental data are done using data from this report.

In summary, the data for I_{DR}, I_A, I_M, and I_Q is more complete than that for I_C and I_{AHP}. For I_{DR} and I_A, estimations of steady state activation/inactivation parameters from voltage clamp are available, although the associated time constant data is not as complete. Also, there is strong evidence as to these currents' specific roles from various current clamp protocols. On the other hand, much of the data used to evaluate these currents are taken under conditions in which several other currents are simultaneously active, making it difficult to separate each contribution. For I_M and I_Q the situation is similar in that there is good data on steady state activation (the evidence shows that these currents do not inactivate) from voltage clamp studies, with sparse estimates on the time constant parameters. However, evaluating the behavior of I_M and I_Q is somewhat easier than doing so for I_{DR} and I_A since these currents are activated in relative isolation with respect to the other currents.

In the case of I_C and I_{AHP}, little voltage clamp data is available for either their steady state or temporal properties of any presumed activation/inactivation parameters. In addition, describing these currents is complicated by the fact that they are presumably mediated by intracellular Ca^{2+}. Little quantitative data is available on this interaction for either current, and there is at present no consensus among workers in this field as to

the mechanisms involved. As introduced in the previous chapter and which shall be expanded upon later, I have made the simple assumption (like that used by other workers, e.g. [48]) that I_C and I_{AHP} are dependent on a power of the concentration of Ca^{2+} either directly beneath the membrane or in a secondary "compartment". This is a highly speculative model, as discussed in the previous chapter. The parameters of this description are based primarily on heuristics, specifically the simulation of the fAHP and the AHP that is observed in HPC. Making the situation more difficult is the fact that there are no protocols to date in which I_C or I_{AHP} are activated without the concomitment presence of other currents, thereby inextricably tying the behavior of any set of estimated parameters for these currents to those of other currents.

In light of the above situation, I developed the present description of the K^+ currents in the following way [1]:

1. Begin with the data on I_M and I_Q, with estimates on the time constants derived from the data and the HH single barrier model. For I_Q its parameters did not affect the later development since this current is only activated at potentials lower than that generally considered in the simulations.

2. Develop an estimation of I_{DR} based on the available voltage clamp data and the simulation of data on action potentials in which I_{DR} is presumably the only repolarizing current.

3. Develop an estimation of I_A based on the available voltage clamp data and simulation of action potentials in which presumably the only K^+ currents are I_{DR} and I_A.

4. Re-evaluate the description of I_M with simulations that reflect the contribution of I_M to the action of I_{DR} and I_A.

5. To a first approximation, the actions of I_C and I_{AHP} are independent of one another. I_C is transient over a time span of a few milliseconds during the spike, and the evidence indicates that this a large current. On the other hand, I_{AHP} activates more slowly, is small, and may last from 0.5 to several seconds. However, since both these currents are

[1] For each K^+ current, as with the Na^+ and Ca^{2+} currents, building the description of the current began with estimating the number and type of activation and/or inactivation and/or Ca^{2+}-mediated activation variables governing the conductance.

dependent on Ca^{2+} entry, their estimation was tied to the description of I_{Ca} and the mechanisms regulating $[Ca^{2+}]_{shell.1}$ and $[Ca^{2+}]_{shell.2}$. Therefore, while the behavior of the I_C or I_{AHP} descriptions could be evaluated independently, whenever the Ca^{2+} mechanisms were modified to alter one of the current's action, the effect of the modification on the other current had to be checked.

6. As the descriptions for I_C and I_{AHP} evolved, the impact of a given version on the behavior of the other currents had to be continually re-evaluated. At times, this feedback resulted in modifications of one of the other currents. In these cases modifications were made which stayed within the envelope of parameters that had been already established. For example, modification of some aspect of I_C might indicate that the parameters of $I_{Na-trig}$ had to be changed. However, this change could not alter the aspects of $I_{Na-trig}$ that had been fixed by earlier simulations (e.g. the threshold of $I_{Na-trig}$).

As described in Chapter 5, the estimation of the K^+ currents involved many iterations, many of which caused re-evaluation of either the Na^+ currents or Ca^{2+} system parameters. The linear parameters of the model, however, were kept constant, since these parameters were established based on data from cells in which all non-linear currents had been inhibited.

As mentioned earlier, certain agents are assumed to mediate selective blocking of specific currents, in accordance with the generally accepted conclusions in the literature. These agents and their actions are summarized in Table 7.1. Any blocking agent used experimentally probably does not act with perfect selectivity, particularly given the wide variety of mechanisms that have proposed for their action (e.g. receptors-site mediated, blockage of the channel lumen, secondary block of Ca^{2+}-dependent K^+ channels via block of Ca^{2+} channels). However, as a first approximation, perfect selectivity is often assumed when evaluating the data (for example application of 4-AP blocks only I_A, leaving the remaining currents untouched).

7.3 Delayed Rectifier Potassium Current - I_{DR}

The delayed-rectifier potassium current is similar to the classical delayed rectifier for the squid axon as described by Hodgkin and Huxley. The parameters for this current were initially taken from [42], who identified I_{DR} in voltage clamp studies as a large, slowly-activating ($\sim$ 100 milliseconds),

Current	NA	TEA	4-AP	ACH	Ba	Musc	Ca^{2+}-blk
I_{DR}		++		+? (1)	+? (1)		
I_A	+ (2)	- (3)	++				
I_C		++ (4)					++ (4)
I_{AHP}	++ (2,4)	- (4)					++ (4)
I_M					++ (1)	++ (1)	- (1)
I_Q	++ (1)				- (1)		

Table 7.1: Typical chemical agents used to block specific currents, as reported by different investigators. (+) indicates reduction, (++) indicates blocking, (-) indicates no effect. NA = Norepinephrine. TEA = Tetra-ethyl ammonium, 4-AP = 4-Aminopyridine, Ach = Acetylcholine, Ba = Barium, Musc = Muscarine, Ca^{2+}-blk = Ca^{2+}-blockers (e.g. Cadmium, EGTA). (1) - [16]. (2) - [39]. (4) - [30]. (3) - [43]

very-slowly inactivating ($\sim$ 3 seconds), TEA-sensitive K^+ current. However, the voltage clamp was only taken to -35 millivolts, so it is possible that only the beginning of the I_{DR} characteristics were measured. In particular, I propose that the time constant for activation, τ_x, drops to about 1-2 milliseconds at membrane potentials greater than -20 millivolts.

My description of this current is based on the data of [42], specifically the reported steady-state activation/inactivation curves. In the model I_{DR} is constructed so that it may function as a major repolarizing component during the action potential. Such a role is indicated by current clamp experiments in which the spike is quickly repolarized by a TEA-sensitive component in the presence of Ca^{2+} blockers. These blockers, which disable the Ca^+ currents, presumably also disable any Ca^+-mediated K^+ currents, in particular I_C. In summary, the main actions that I determined I_{DR} served included:

- Repolarize the action potential fully when all other K^+ currents have been blocked
- Reduce in the presence of other repolarizing currents so that no extra hyperpolarization is observed

- Mediate the medium after-hyperpolarization (mAHP), independent of any other K^+ currents
- Repolarize the cell sufficiently during tonic stimulation so that repetitive firing at the threshold of $I_{Na-trig}$ could occur
- Activate independently of the width of spike since the mAHP is unaffected by the slower repolarization with 4-AP or Ca^{2+} blockers

As will be discussed shortly, there is evidence that I_C plays a major role in spike repolarization under certain conditions, and in fact it has been suggested that this current is the major repolarizing current in bullfrog sympathetic neurons. Since action potentials are quickly repolarized in hippocampal pyramidal neurons under conditions that would eliminate I_C, however, it was thought that the characteristics of I_{DR} would allow it to reprise it classical role when I_C has been disabled.

7.3.1 Results

This effort was successful in simulating the TEA-sensitive repolarization of the action potential, as shown in Figure 7.1. In addition, this formulation of I_{DR} kinetics was able to simulate the voltage clamp results as reported by [42].

Three activation particles (x) were used in the formula for the I_{DR} conductance so that activation of this current would be delayed after the initial rise of the action potential. A single inactivation particle (y) was used since it has been reported that this current does indeed inactivate ([42]). However, the time constant for y is quite slow over most of the physiological range of membrane voltages (ref. Figure 7.3. so that during the action potential and afterwards, the demise of I_{DR} is primarily due to removal of activation rather than inactivation. Removal of I_{DR} by inactivation after the spike is consistent with the mechanism of I_{DR} in the squid axon as described in [21], [20]. [22]. [23].

The valences and the $V_{\frac{1}{2},x,DR}$ and $V_{\frac{1}{2},y,DR}$ for x and y was determined by the x_∞ and y_∞ curves reported by [42]. In the case of the x particle the third power of x_{infty} was matched to the [42] data.

The curve for $\tau_{x,DR}$ was skewed to the left ($\gamma_{x,DR} = 0.9$) so that I_{DR} would remain activated after the spike long enough to cause the mAHP, and so that $\tau_{x,DR}$ was consistent with the reported values of approximately 180 milliseconds. $V < -30$ millivolts, approximately 6 milliseconds otherwise.

Gating Variable	z	γ	α_0	$V_{\frac{1}{2}}$ (mV)	τ_0 (ms)
x (activation)	12	0.95	0.008	-28.0	0.5
y (inactivation)	-9	0.8	0.0004	-45.0	6.0

Table 7.2: Parameters of I_{DR} Gating Variables

The curve for $\tau_{y,DR}$ was skewed to the left ($\gamma_{y,DR} = 0.2$) in order to approximate the reported approximate value of 4 seconds for y (between -50 and -30 millivolts, [42]).

On the other hand, in order that activation be independent of the width of the spike, as described above, it was necessary to set the base rate for $\tau_{x,DR}$ to 0.5 milliseconds. Later in this chapter and Chapter 10 the role of I_{DR} in concert with I_A and I_C will be demonstrated, including examination of I_{DR}'s role in mediating the mAHP.

Another parameter that was important to set in regards to I_{DR} was its reversal potential. The standard value of -85 mV for E_K caused I_{DR} to be too strong near threshold, specifically, on repolarization of the spike the I_{DR} wiped out the ADP seen in the data. To reconcile this problem without significantly altering the time course and strength of I_{DR} during the initial stage of the spike repolarization and the later mAHP, it was necessary to set a reversal potential for this current distinct from the general E_K. E_{DR} was set to -73 millivolts, which proved successful in obtaining the desired behavior. This was felt to be a reasonable adjustment, since (as mentioned in Chapter 2) a given channel is not necessarily perfectly selective for one species of ion – an E_{DR} of -73 mV implies that I_{DR} is slightly contaminated with an occasional Na^+ or Ca^{2+} ion hitching along with the predominantly K^+ flow.

All I_{DR} parameters were determined at 30°C.

The equation for I_{DR} is -

$$I_{DR} = g_{DR} x_{DR}^3 y_{DR} (V - E_{DR})$$

where

$$\bar{g}_{DR} = 0.7\,\mu\text{S}$$

$$E_{DR} = -73\text{mV}$$

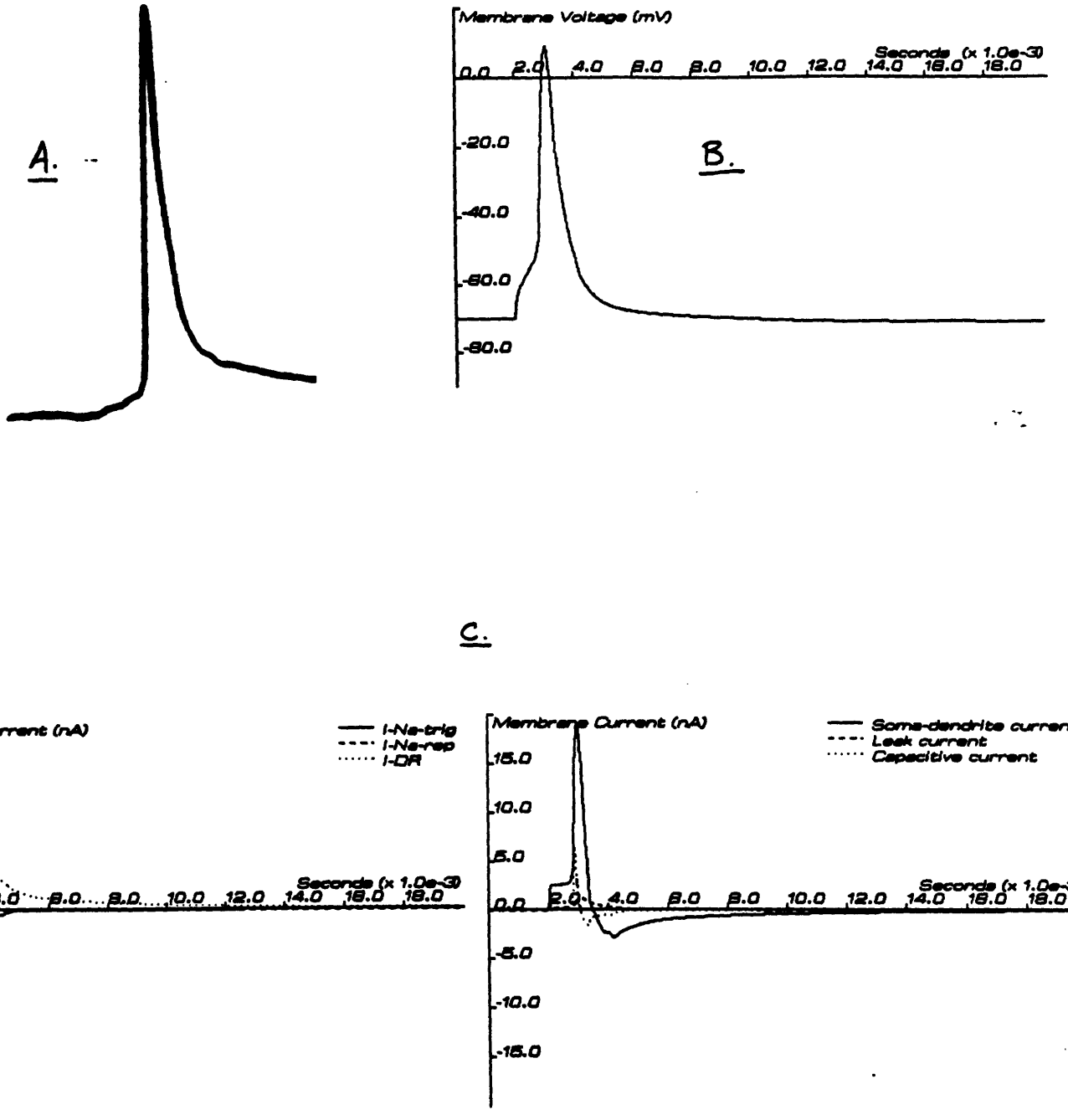


Figure 7.1: A. Action potential with Ca^{2+} blockers and 4-AP in external medium. B. Current clamp simulation of (A.). C. Main currents in simulation, including I_{DR}, $I_{Na-trig}$, I_{Na-rep}, and $I_{Na-tail}$.

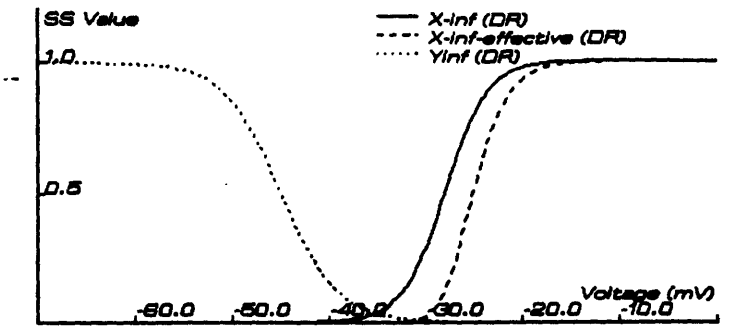


Figure 7.2: Steady-state curves (x_∞ and y_∞) for x_{DR} and y_{DR} and effective curves as would be measured by voltage-clamp experiments.

Table 7.2 lists the parameters for the I_{DR} gating variables. These are the rate functions for the activation variable, x, of I_{DR} -

$$\alpha_{x,DR} = 0.008 \exp\left(\frac{(V + 28)0.95 \cdot 12 \cdot F}{RT}\right)$$

$$\beta_{x,DR} = 0.008 \exp\left(\frac{(-28 - V)0.05 \cdot 12 \cdot F}{RT}\right)$$

These are the rate functions for the inactivation variable, y, of I_{DR} -

$$\alpha_{y,DR} = 0.0004 \exp\left(\frac{(V + 45)0.8 \cdot -9 \cdot F}{RT}\right)$$

$$\beta_{y,DR} = 0.0004 \exp\left(\frac{(-45 - V)0.2 \cdot -9 \cdot F}{RT}\right)$$

Figure 7.2 and Figure 7.3 show the voltage dependence on the steady-state values and the time constants for the x_{DR} and y_{DR} variables.

7.4 A-Current Potassium Current - I_A

I_A is a transient K^+ current whose classical role, first defined for molluscan neurons, is to modulate excitability. In particular, this current is selectively blocked by 4-AP, and the convulsant action of this drug is attributed to its inhibition of I_A.

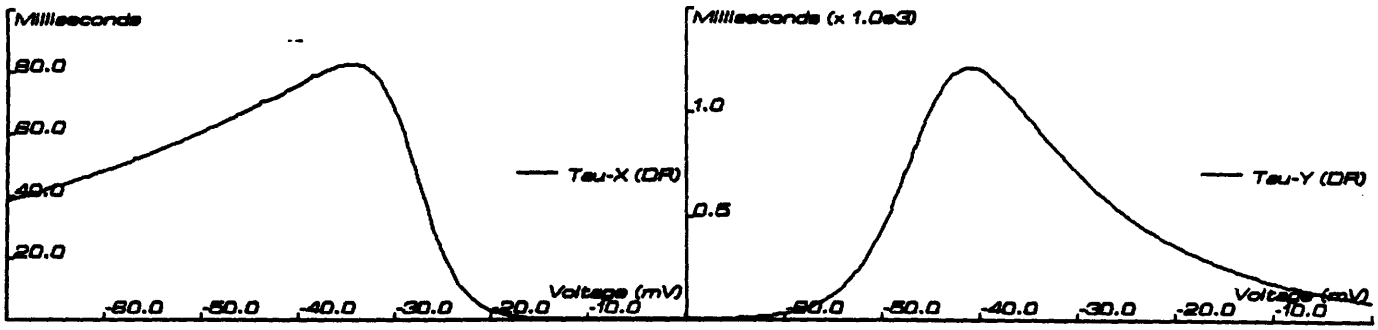


Figure 7.3: Time constant curves (τ_x and τ_y) for x_{DR} and y_{DR}.

Several workers have reported an I_A in HPC. However. the data obtained by voltage clamp differs somewhat in different reports. and the functional effect of I_A (inferred from current clamp stimulation with and without 4-AP or other I_A-agonists) varies considerably. In general, I_A has been reported to modulate the width of the action potential and influences the excitability of the cell. References which report voltage-clamp measurements of this current include Segal and Barker, 1984 [42]. Halliwell et al. 1986 [17]. Zbicx and Weight, 1985 [54]. Gustafsson et al, 1982 [14], and Segal et al. 1984 [43]. In addition. the action of I_A on spike repolarization is reported in Storm. 1986b [47]. Some of these reports will now be summarized.

Segal et al measured I_A in cultured rat hippocampal cells (subfield not specified). Making their measurements at 21- 24°C, they report that I_A is half-inactivated at rest (-70 mV), has a $V_{1/2}$ for activation at about -20 to -30 mV. and (apparently) is described by ba^4 kinetics, where b is inactivation and a is activation. The maximum conductance for I_A was estimated to be greater than .5 μS. The time constant of decay at (24°C) is about 24 ms. independent of voltage. The time constant to peak was within 10 ms. Application of 4-AP lowered spike threshold from -44 to -50 mV. but this procedure *did not* broaden the spike. [2].

Interestingly. the current clamp record shown in this report demonstrated spikes with a) a high threshold (-50 to -44 millivolts. as compared to typical thresholds of about -55 millivolts) and b) small amplitudes. peaking at about +5 millivolts. as compared to typical action potential peaks of

[2]This is contrary to the data of [47] although it is possible that in the [43] report they did not look at the spike carefully enough.

approximately +20 millivolts. There are several implications of this data. First, the lowering in threshold with 4-AP implies that for a spike-producing stimulus that is not too large, over a voltage range of about 5 millivolts and for almost 10 ms there is little inactivation of the threshold Na^+ current. If this "trigger" current is the spike initiation current here, then this means that at either threshold the steady state inactivation is practically complete. For no inactivation to take place during the approach to the higher threshold spike, this means that the time constant for inactivation of the Na^+ current in this range must be greater than 20 ms. Likewise, the fact that 4-AP does not change the amplitude of the spike, but (probably) changes (slightly) the width of the spike implies that either a) in the control I_A transiently activates and is gone during the upstroke of the spike, only to reactivate during repolarization in order to contribute to the repolarization, or b) I_A is present during the entire spike, but the slower onset to the threshold as mediated by I_A allows stronger activation of the Na^+ current, which in turn cancels out the effect of I_A during the upstroke and peak of the spike. The first possibility is not likely because removal of inactivation for I_A cannot take place during repolarization since steady state inactivation is complete at -50mv.

Halliwell et al. investigated CA1 cells in slices of rat and guinea pig hippocampus, measuring, at 28°C, the effects of dendrodotoxin (DTX) and 4-AP. They report an I_A which is sensitive to both these agents, has a very fast onset and an activation curve that starts near -60mV. The DTX-sensitive component was .5nA at a -40mV clamp voltage (v-holding = -76mV). Inactivation starts at about -60 mV, and was linear to -100 mV. The time constant for decay of the DTX-sensitive component was 20 ms at -40 mV, and seemed to slow at lower potentials; a faster decaying outward component which was resistant to 4AP or DTX (perhaps I_C) decays within about 10 ms.

Gustafsson et al measured guinea pig CA3 pyramidal cells from slice at 33°C or 26° C. This report shows activation and inactivation characteristics similar to that reported for the cultured cells in Segal et al. 1984, with a peak current at -30 mV of 5 nA. A faster decaying outward component of similar size remained after application of 4AP, and this component may in fact have two components; two time constants of the faster component were measured - about 10ms and about 1sec (26° C). This might partially reflect contribution of I_C .

Zbicx and Weight also measured guinea pig CA3 pyramidal cells from slice, this time at either 32°C or 33° C. These workers report a decay time

constant = 200 - 400 mS. The activation to peak was within 10 mS. independent of voltage. Peak current at -35 mV was 4 nA. Threshold for activation of I_A was -55 to -50 mV. It appeared that above about -40 mV inactivation had two components, a fast one with a time constant of about 100 mS and (after about 100 mS) a slow one of about 380 mS. Lack of 4-AP-sensitive tail current below -54 mV suggests that this current deactivates very rapidly upon hyperpolarization.

Finally, Storm reports that I_A mediates a rapid onset, pre-spike transient (several hundred milliseconds) outward rectification that delays onset of repetitive firing for a narrow range of tonic stimulus strengths. This I_A does not, however, alter the frequency of firing once the spike train starts. This data implies that under the reported protocol I_A inactivates during the IR (initial ramp). These experiments were done with Mn, which presumably will block the Ca^{2+} currents or the Ca^{2+}-mediated currents. Also, [47], reports that 4-AP broadens the repolarization of single spikes, but does not effect the fAHP or the mAHP. Under some protocols addition of 4-AP (or Cd) caused a second (almost twice as broad) spike to be fired spontaneously within 10 milliseconds of the first spike. The second spike was also about 5-10 millivolts smaller and lacked a fAHP under either 4-AP or Cd. Pre-hyperpolarizations (-80mV for 900ms) enhanced the effect of 4-AP on spike repolarization; pre-depolarizations (-58mV for 900 ms) reduced effect by about half that of the hyperpolarizing protocol, implying that inactivation is not complete at -58mv. In the current study the records from this report [47], are the primary ones used to compare the simulations with actual data.

A related report describes the putative role of I_A at the post-synaptic terminal. Application of 4-AP has been described as enhancing synaptic transmission [45]. Irregular firing subsequent to the IR reported by [46] may therefore be partially due to enhancement of spontaneous EPSPs from inhibition of synaptic I_A.

To summarize, the so-called I_A has been reported by different investigators to:

1. Delay onset of spike
2. Raise spike threshold
3. Mediate transient "initial ramp" (strong outward rectification) prior to initial spike in response to tonic stimulus without strong role during later spikes (particularly frequency of later spikes)

4. Selectively modulate repolarization of single spike without effecting mAHP or sAHP and have a minimal effect on spike amplitude

The data includes fairly complete measurement of the steady state activation and inactivation curves, but there is not complete data on the voltage dependence of the appropriate time constants. Current clamp data showing the previously mentioned effects of I_A are available, however. Therefore in order to simulate this current I began with the reported steady-state curves and then derived functions for the time constants that were consistent with the voltage clamp data and that reproduced the current clamp results.

One of the key features of this current that had to be matched in the simulations was the fact that during the spike the appearance of the I_A was exquisitely timed to influence just the main part of the repolarizing phase. As previously mentioned, experiments in which spikes were elicited with and without 4-AP showed that blockage of I_A did not influence the ADP or mAHP ([47]), thus indicating that the I_A was fully deactivated/inactivated within a few milliseconds of its onset.

7.4.1 Results

The results for the derived kinetics are shown in Figure 7.9, which includes the reported steady state curves for the activation variable x and the inactivation variable y.

The channel was configured with three activation gating particles (x) to obtain a delay in activation with depolarization. The effect of I_A is seen only 1 to 2 milliseconds *after* the peak of the spike. Raising the power of x was necessary to obtain the required delay consistent with the position of x_∞ curve on the voltage axis, as reported by [42].

On the other hand, given the y_∞ curve in the same report, no delay was necessary for the inactivation of I_A, and only one y variable was used in the channel formulation.

Figure 7.4 illustrates the contribution of I_A on the repolarization of the single action potential in the presence of Ca^{2+} blockers (which will inhibit the contribution of I_C on the repolarization) and without these blockers. The experimental data was taken by measuring the response with and without 4-AP.

Figures 7.5, 7.6, and 7.7 illustrate the data from [46] and simulations of this data which demonstrate the role of I_A in mediating the IR prior to repetitive firing. In the simulations the IR is very sensitive to stimulus

Gating Variable	z	γ	α_0	$V_{\frac{1}{2}}$ (mV)	τ_0 (ms)
x (activation)	3.5	0.8	0.2	-52.0	1.0
y (inactivation)	-7	0.4	0.0015	-72.0	24.0

Table 7.3: Parameters of I_A Gating Variables

strength, and that beyond a narrow range this response is quite diminished. This characteristic is consistent with data taken under similar conditions (Storm, personal communication).

The reported action of I_A related to the increased excitability of the cell with the addition of 4-AP is shown in Figure 7.8. In this figure recordings from Segal et al are compared with simulations of similar protocols. The delay in firing in the cell without 4-AP is similar to that demonstrated earlier with the simulations of Storm's data.

All I_A parameters were determined at 30°C.

The equation for I_A is -

$$I_A = 0.5 x_A^3 \, y_A \, (V - E_K)$$

where

$$\bar{g}_A = 0.5 \, \mu\text{S}$$

Table 7.3 lists the parameters for the I_A gating variables. These are the rate functions for the activation variable, x, of I_A -

$$\alpha_{x,A} = 0.2 \exp\left(\frac{(V + 52)0.8 \cdot 3.5 \cdot F}{RT}\right)$$

$$\beta_{x,A} = 0.2 \exp\left(\frac{(-52 - V)0.2 \cdot 3.5 \cdot F}{RT}\right)$$

These are the rate functions for the inactivation variable. y. of I_A -

$$\alpha_{y,A} = 0.0015 \exp\left(\frac{(V + 72)0.6 \cdot -7 \cdot F}{RT}\right)$$

$$\beta_{y,A} = 0.0015 \exp\left(\frac{(-72 - V)0.4 \cdot -7 \cdot F}{RT}\right)$$

Figure 7.9 and Figure 7.10 show the voltage dependence on the steady-state values and the time constants for the x_A and y_A variables.

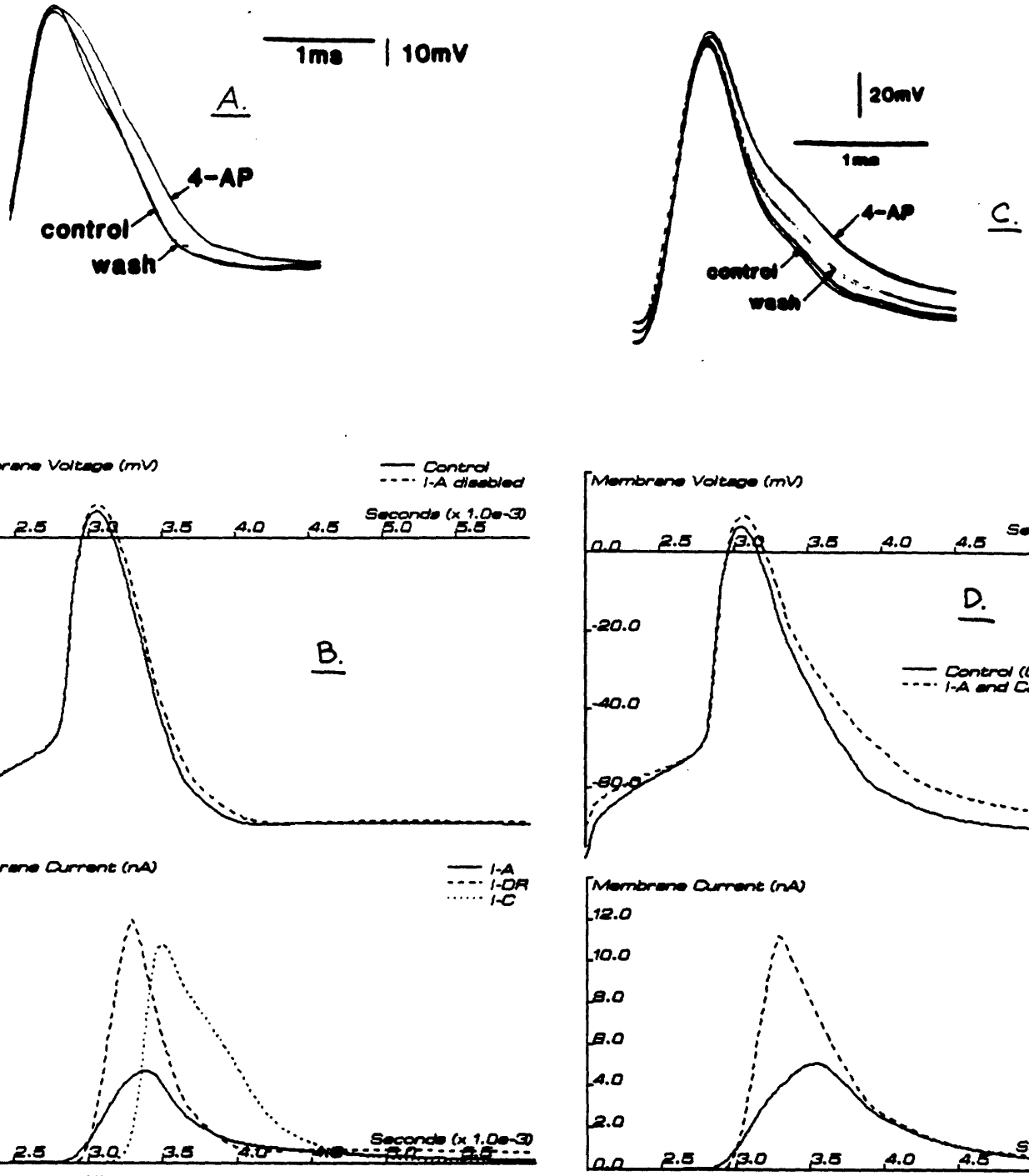


Figure 7.4: Current clamp data of action potentials which demonstrate the role of I_A during repolarization. A. Action potential with and without 4-AP. B. Current clamp simulation of (A.). Non-linear currents here include I_A, I_{DR}, I_C, I_{Ca}, $I_{Na-trig}$, I_{Na-rep}, and $I_{Na-tail}$. C. Action potential with and without 4-AP (Ca^{2+} blockers added). D. Current clamp simulation of (C.). Non-linear currents here include I_A, I_{DR}, $I_{Na-trig}$, I_{Na-rep}, and $I_{Na-tail}$.

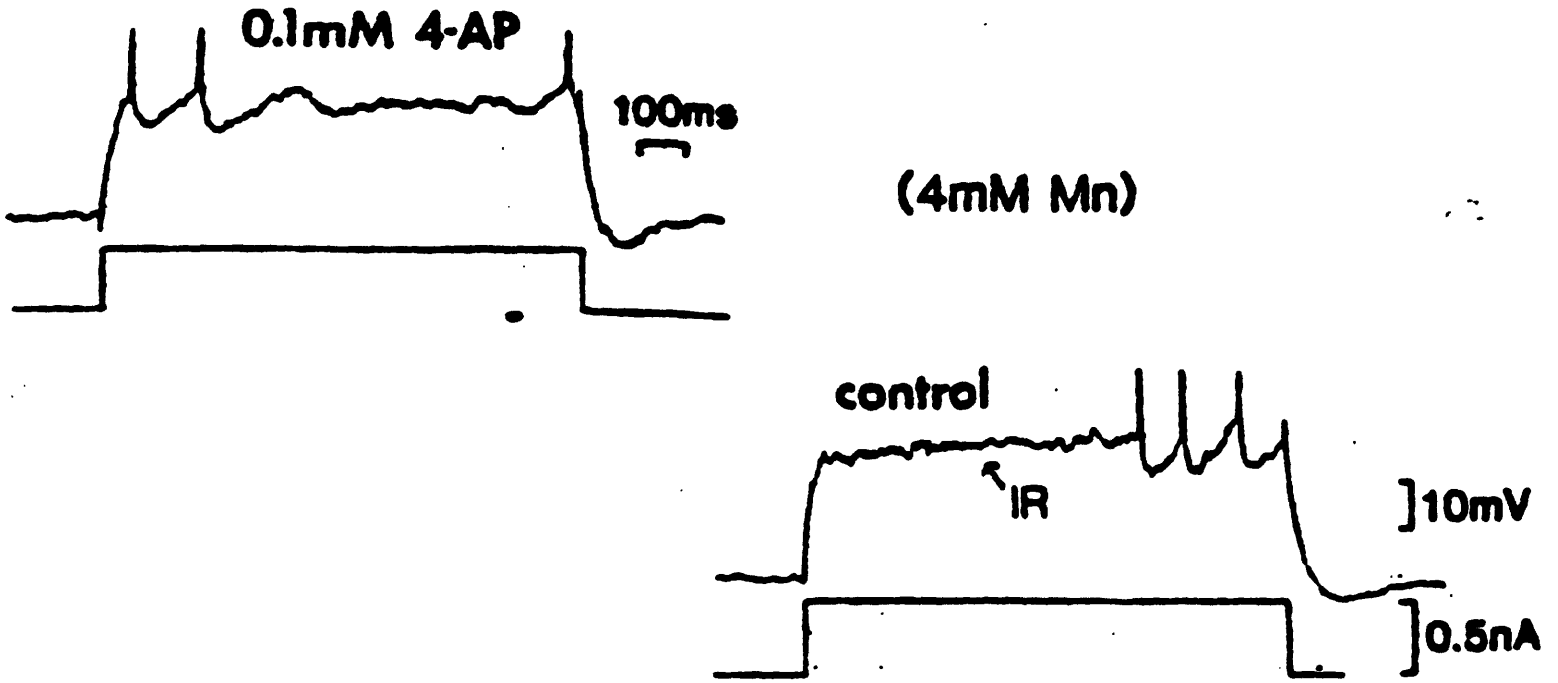


Figure 7.5: Current clamp records (Storm. 1986a) of repetitive firing in response to tonic, small amplitude current stimulus, with and without application of 4-AP. Mn added. With 4-AP on the left; control on the right. Each stimulus began with a long hyperpolarizing prepulse, which presumably removed most of the resting inactivation of I_A.

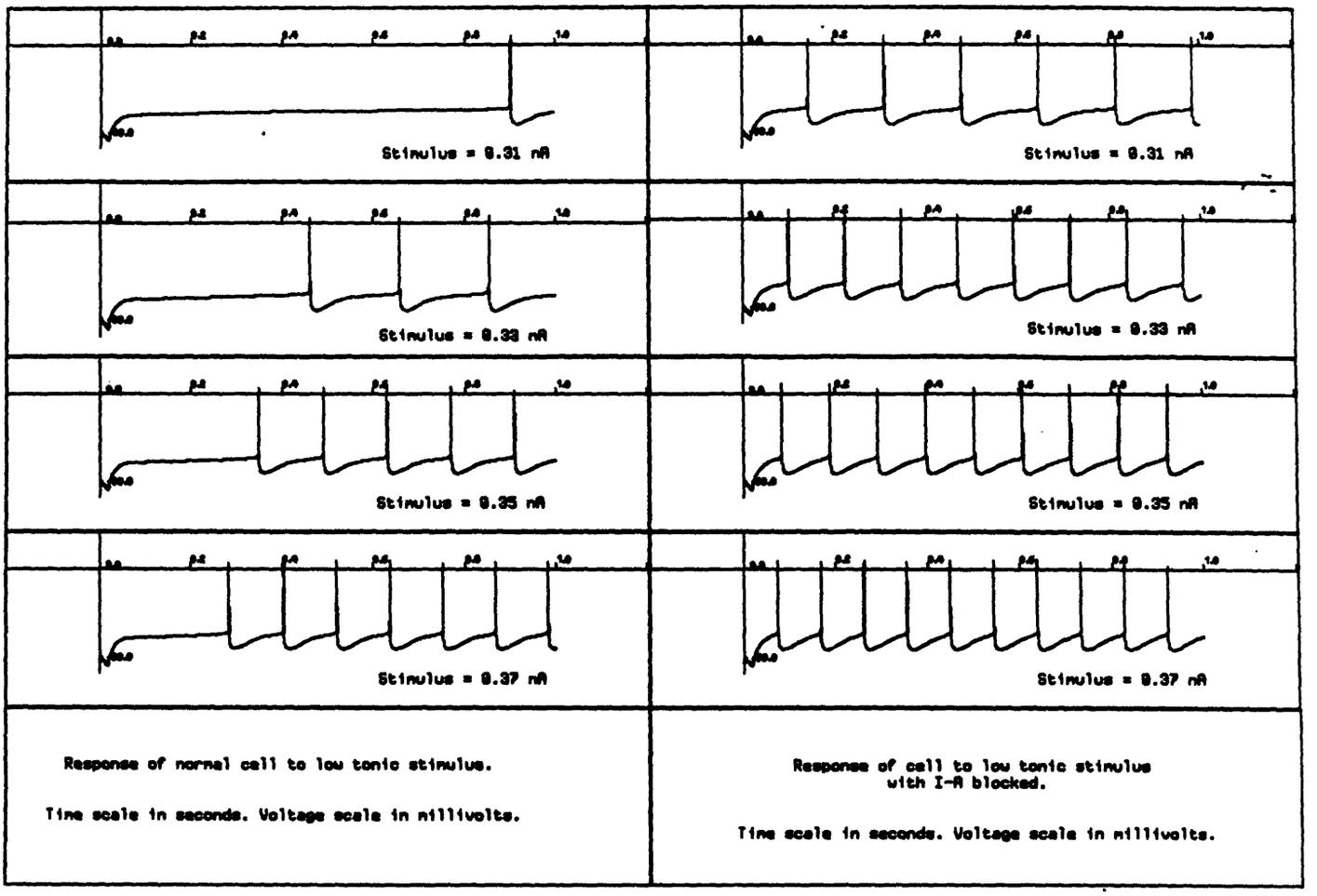


Figure 7.6: Current clamp simulation of repetitive firing in response to tonic, small amplitude current stimulus, with and without I_A. No Ca^{2+} currents. Simulations with I_A are on the left: without I_A on the right. Each stimulus begins with a 20 ms -0.5 nA pulse (hyperpolarizing) in order to remove most of the resting inactivation of I_A. These simulations follow from the records shown in Figure 7.5.

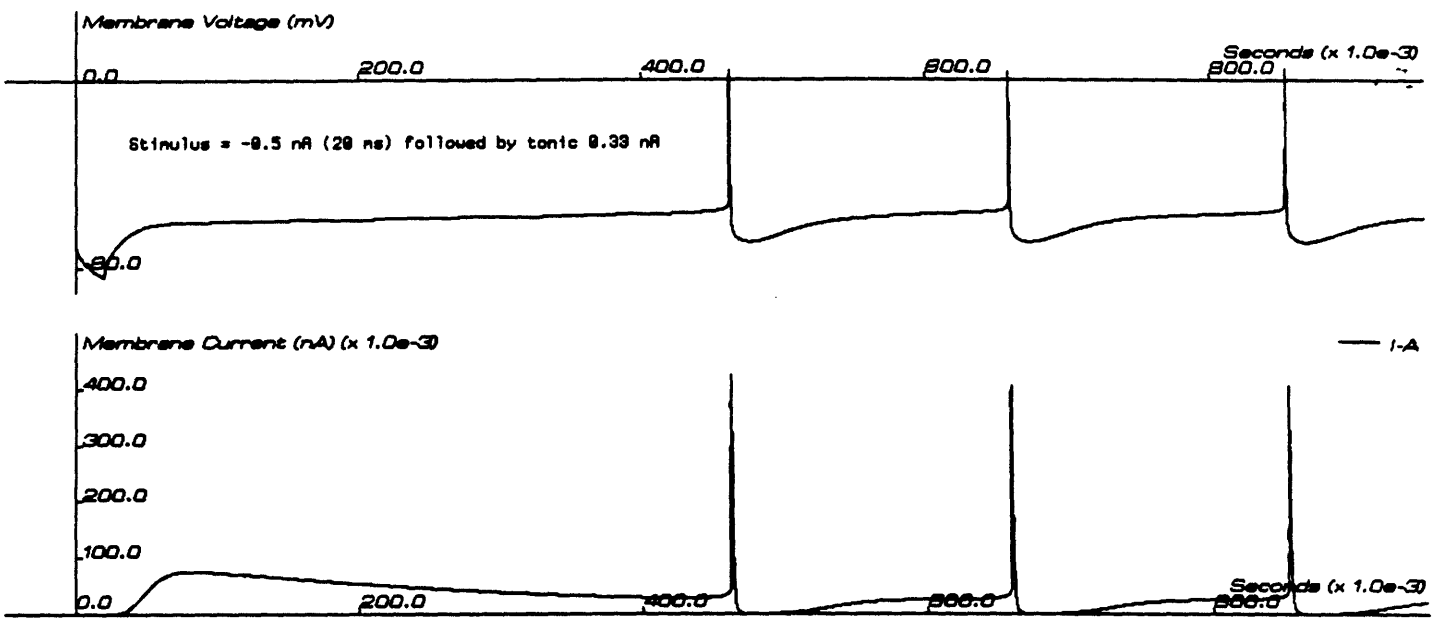


Figure 7.7: Record of I_A during response to 0.33 nA stimulus, as shown in previous figure. The pre-threshold activation of I_A serves to delay the initial spike.

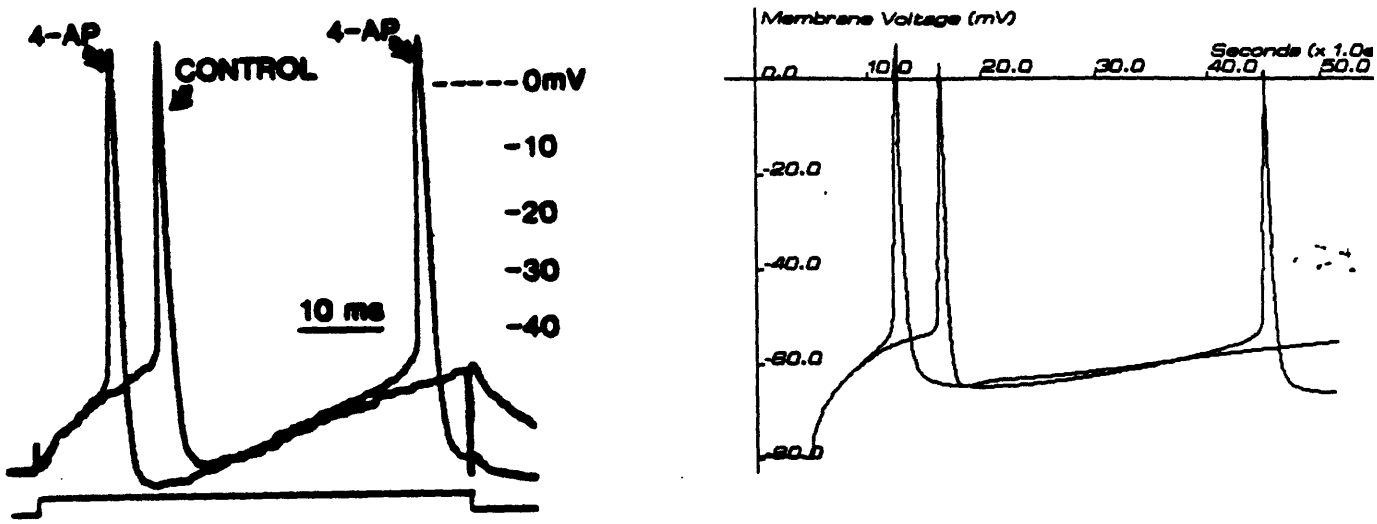


Figure 7.8: Current clamp data of action potentials which demonstrate the role of I_A during strong tonic stimulus. Left - Record of successive spikes in response to tonic stimulus before and after application of 4-AP ([43]). Note the advance of spike and slight lowering of threshold with application of 4-AP. Right - Current clamp simulation of record on left. Stimulus is 2 nA.

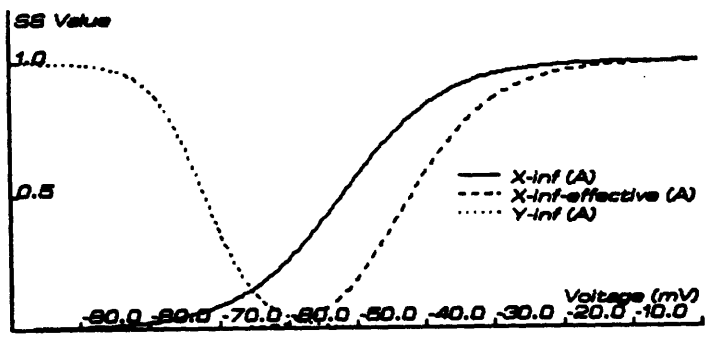


Figure 7.9: Steady-state curves (x_∞ and y_∞) for x_A and y_A and effective curves as would be measured by voltage-clamp experiments.

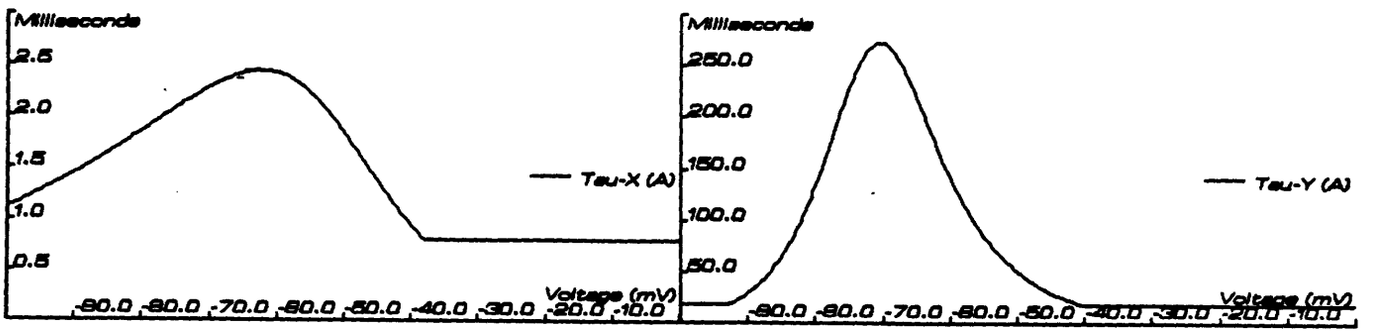


Figure 7.10: Time constant curves (τ_x and τ_y) for x_A and y_A.

7.5 Ca^{2+}-Mediation of K^+ Currents by Ca^{2+} - binding Gating Particle w

In order to cause I_C and I_{AHP} to be mediated by intracellular Ca^{2+}, I incorporated a Ca^{2+}-binding gating particle in the expressions for both of these currents. Several workers have postulated mechanisms for such an interaction between intracellular Ca^{2+} and different ion channels, ranging from complex multi-state kinetic models based on experimental data to very simple descriptions for modelling studies ([48]).

In light of the paucity of quantitative data on such mechanisms in HPC, my goals for the description of a putative, generic Ca^{2+}-binding gating particle were as follows:

- Relationship between Ca^{2+} concentration and particle activation allowing for non-degenerate kinetics considering the range of Ca^{2+} concentrations during various cell responses.
- Binding kinetics based on a simple but reasonable model.
- Kinetic description that could be easily modified to yield significantly different behavior, that is a description that could be modified to suit a wide range of desired behaviors.

To this end the following description for a Ca^{2+} -binding gating particle, w, was used. Each w particle can be in one of two states, open or closed, just as the case for the Hodgkin-Huxley-like voltage-dependent activation

and inactivation gating particles. Each w particle is assumed to have n Ca^{2+} binding sites. all of which must be bound in order for the particle to be in the open state. Binding is cooperative in a sense that reflects the two states available to a given particle. i.e. either a particle has no Ca^{2+} ions bound to it. and therefore it is in the closed state, or all n binding sites are filled. and the particle is in the open state. The state diagram for this reaction is as follows:

$$w_{closed} + n\,Ca_{in}^{2+} \overset{\alpha,\beta}{=} w_{open}^{*}$$

where the $*$ notation means that the particle is bound to all n (intracellular) Ca^{2+} ions. α and β are the forward and backward rate constants, respectively.

This scheme results in the following differential equation for w, where now w is the fraction of particles in the open state. assuming that the concentration of Ca^{2+} is large enough that the reaction does not significantly change the store of intracellular Ca^{2+}:

$$\frac{dw}{dt} = \alpha(1 - w)[Ca^{2+}]_{in}^{n} - \beta w$$

The steady state value for w (the fraction of particles in the open state) as a function of the intracellular Ca^{2+} concentration is then:

$$w_{\infty} = \frac{\alpha[Ca^{2+}]_{in}^{n}}{\alpha[Ca^{2+}]_{in}^{n} + \beta}$$

The time constant for the differential equation is:

$$\tau_w = (\alpha[Ca^{2+}]_{in}^{n} + \beta)^{-1}$$

The order of the binding reaction.n. that is the number of Ca^{2+} binding sites per w particle. determines the steepness of the previous two expressions, as a function of $[Ca^{2+}]_{in}$. Given the constraints on the range for $[Ca^{2+}]_{shell.1}$ and $[Ca^{2+}]_{shell.2}$ during single and repetitive firing. n was set to three for both the I_C w particle and the I_{AHP} w particle. On the other hand. as shall be presented shortly. the range of Ca^{2+} concentrations for which the I_{AHP} w particle is activated is set to about one order of magnitude lower than that for the I_C w particle. since I_C was exposed to the larger $[Ca^{2+}]_{shell.1}$.

7.6 C-Current Potassium Current - I_C

I_C is a Ca^{2+}-dependent K^+ current that plays a large role during an single, isolated action potential. It has been proposed that this current, which is apparently inhibited when Ca^{2+} blockers are added, is the underlying current of the fast-afterhyperpolarization (fAHP) which is observed in single spikes and (sometimes) to a lesser degree after spikes of repetitive trains. Studies of bullfrog sympathetic ganglion neurons suggest that I_C is a major repolarizing current during the spike in these cells (Adams et al).

Limited voltage clamp data was available for this current (Segal and Barker, Madison et al), and in many of the reports measurements of a reputed I_C was likely corrupted by I_{AHP}, since both are identified by, among other methods, sensitivity to Ca^{2+} blockers.

On the other hand, [47] has demonstrated well the role of I_C during the fast repolarization of the action potential, distinct from the much slower role of I_{AHP} as a hyperpolarizing current. In addition, I_C is selectively blocked with TEA at concentrations much lower than that required to block I_{DR}. Storm has tentively isolated the role of I_C in current clamp protocols, and this data was the primary standard I used in the estimation of the I_C parameters.

In my simulations of the role of I_C the challenge can be summarized as follows:

- Formulate the kinetics of I_C so that the fAHP is reproduced
- Devise a Ca^{2+} and voltage dependence for I_C such that I_C is activated significantly only during spike repolarization.
- Adjust I_{DR} parameters so that activation of I_C does not inhibit activation of I_{DR}- otherwise the presumably fast inactivation of I_C would allow the residual inward currents to immediately depolarize the cell following the fAHP. Also the mAHP, which is mediated by I_{DR}, is insensitive to Ca^{2+} -blockers, further indicating that activation of I_{DR} is not affected by the faster spike repolarization mediated by I_C.

Although the estimation of *every* current necessitated the re-evaluation of every other current to a greater or lesser degree, not only was the estimation of I_C dependent on the characteristics of I_{DR} and vica versa, but also the dependence of I_C on Ca^{2+} meant that estimation of the I_{Ca} and $[Ca^{2+}]_{shell,1}$-system parameters took into account the behavior of I_C.

The first step in analyzing I_C was estimating with simulations the current that is necessary to generate the observed fAHP. The results of these simulations indicating that I_C had to have two salient characteristics - very fast activation/inactivation and large maximum conductance (on par with I_{DR}). In fact, the time course of the fAHP-producing current was a sharp and large spike reminiscent of the Na^+ currents that initiated the action potential.

The characteristics that I have derived for I_C are different from those proposed in the literature, in particular the kinetics described here are somewhat faster than those reported elsewhere. However, as mentioned earlier, the interpretations of the voltage clamp data for I_C is possibly complicated by the activation of I_{AHP}, which is also Ca^{2+}-dependent. In fact, the simulations described here indicate that *if* I_C is the current responsible for the fAHP, then I_C *must* have the fast activation/inactivation/deactivation kinetics proposed here.

7.6.1 Results

The Ca^{2+}-dependence of I_C was constructed so that after only a small delay the influx of Ca^{2+} into *shell.1* would activate I_C. The conductance of the I_C channel was therefore mediated in part by a single w particle, implying that each I_C channel has three independent Ca^{2+}-binding sites on a single gating particle, each of which, in turn, must be bound to Ca^{2+} in order for the channel to conduct.

The very fast turn off of I_C necessary to obtain a significant ADP after the fAHP was accomplished by incorporating four x particles, by making the steady state curve for activation steep and centered only a few millivolts above the resting potential, and by making the time constant for x very fast, especially when the x_∞ curve goes to zero. As the spike is repolarized past about -60 millivolts the x particles quickly relax to their closed state, shutting off I_C leaving a minimal tail that, if larger, could otherwise wipe out the ADP.

A voltage-dependent inactivation particle, y, was included since the available voltage clamp data indicate that I_C is a transient current at depolarized potentials, with a time constant on the order of greater than several milliseconds. Simulations indicate, however, that during normal activity removal of I_C is accomplished by de-activation of either the x or the w particles.

In summary, the turning on of I_C is mediated by influx of Ca^{2+}; removal

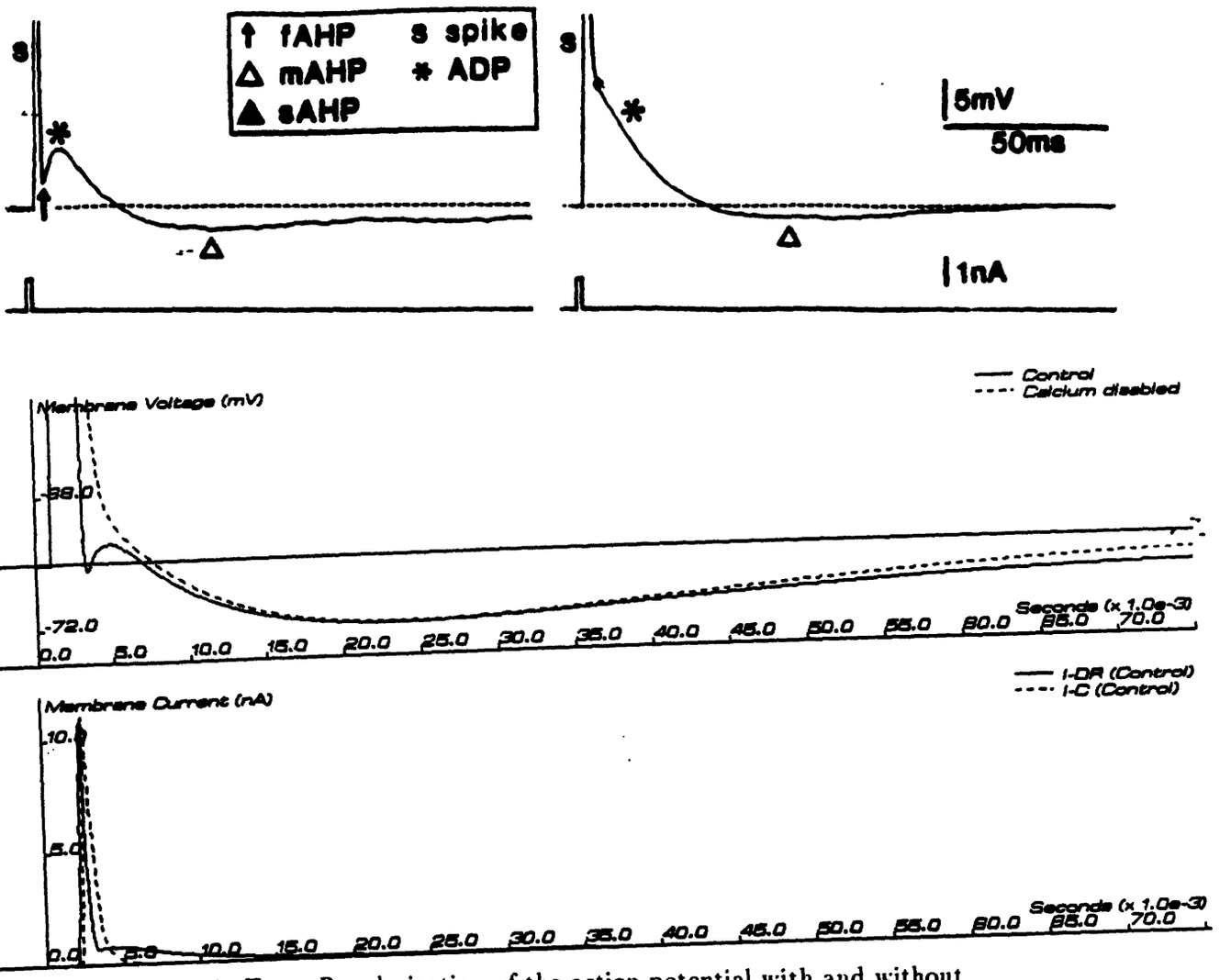


Figure 7.11: Top - Repolarization of the action potential with and without Ca^{2+} blockers (Storm. 1986b). Note absence of fAHP when Ca^{2+} blockers are added. Bottom - Simulation of protocol.

of I_C is mediated by the repolarization of the spike. During the spike, the x particles turn on first with the depolarization of the beginning of the spike. As the I_{Ca} channels open (slower than the $I_{Na-trig}$ and I_{Na-rep} channels), the subsequent influx of Ca^{2+} into $shell.1$ raises $[Ca^{2+}]_{shell.1}$ to turn on the w, turning on I_C. As the cell repolarizes, the four x particles close quickly, shutting off I_C abruptly enough to allow the observed ADP. Soon after the spike (within 30 ms) $[Ca^{2+}]_{shell.1}$ drops as Ca^{2+} flows into $shell.2$ and the core, thereby shutting off w so that activation of x on the upstroke of a subsequent spike does not turn on I_C too soon (see Figure 9.27).

Figure 7.11 illustrates the contribution of the fully activated I_C to the repolarization of the action potential and how the fAHP is eliminated when I_C is blocked. In the next section an expanded view of this simulation will be presented.

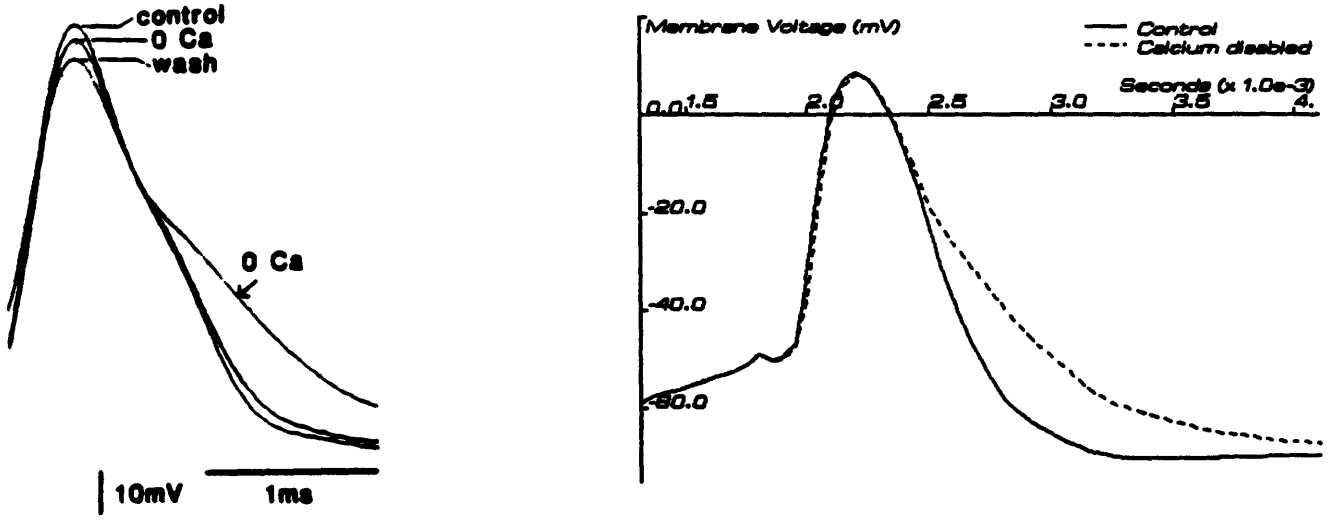


Figure 7.12: Left – Expanded view of the action potential with and without Ca^{2+} blockers (Storm), showing spike broadening with Ca^{2+} blockers. Right - Simulation of protocol.

Gating Variable	z	γ	α_0	$V_{\frac{1}{2}}$ (mV)	τ_0 (ms)	α^{*}_{Ca}	β^{**}_{Ca}
x (activation)	25	0.2	0.007	-65.0	0.25	-	-
y (inactivation)	-20	0.2	0.003	-60.0	15	-	-
w (Ca^{2+}-activation)	-	-	-	-	-	1000	0.125

Table 7.4: Parameters of I_C Gating Variables. $* = (mS^{-1}mM^{-3})$, $** = (mS^{-1})$

The equation for I_C is -

$$I_C = \bar{g}_C \, x_C^4 \, y_C \, w_C \, (V - E_K)$$

where

$$\bar{g}_C = 0.4 \, \mu S$$

Table 7.4 lists the parameters for the I_C gating variables. These are the rate functions for the activation variable, x, of I_C-

$$\alpha_{x,C} = 0.007 \exp\left(\frac{(V + 65)0.2 \cdot 25 \cdot F}{RT}\right)$$

$$\beta_{x,C} = 0.007 \exp\left(\frac{(-65 - V)0.8 \cdot 25 \cdot F}{RT}\right)$$

These are the rate functions for the activation variable, y, of I_C-

$$\alpha_{y,C} = 0.003 \exp\left(\frac{(V + 60)0.2 \cdot 20 \cdot F}{RT}\right)$$

$$\beta_{y,C} = 0.003 \exp\left(\frac{(-60 - V)0.8 \cdot 20 \cdot F}{RT}\right)$$

Figure 7.13 and Figure 7.14 show the voltage dependence on the steady-state values and the time constants for the x_C and y_C variables.

As mentioned above, each w particle was assumed to have three non-competitive Ca^{2+} binding sites, all of which were either empty (corresponding to the closed state) or filled (corresponding to the open state). Figure 7.15 shows the dependence of the steady-state value of the w_C variable on $[Ca^{2+}]_{shell}$. Figure 7.16 shows the dependence of the time constant for the w_C variable on $[Ca^{2+}]_{shell}$.

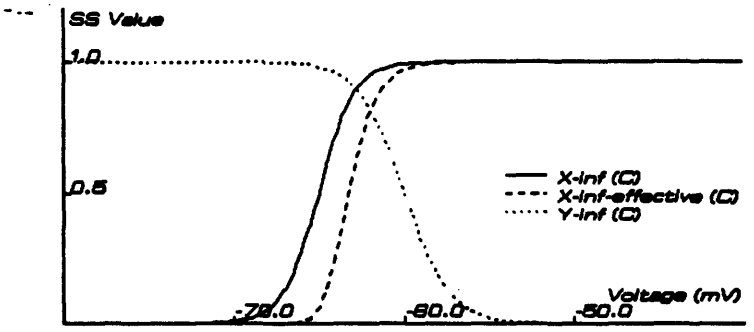


Figure 7.13: Steady-state curves (x_∞ and y_∞) for x_C and y_C and effective curves as would be measured by voltage-clamp experiments.

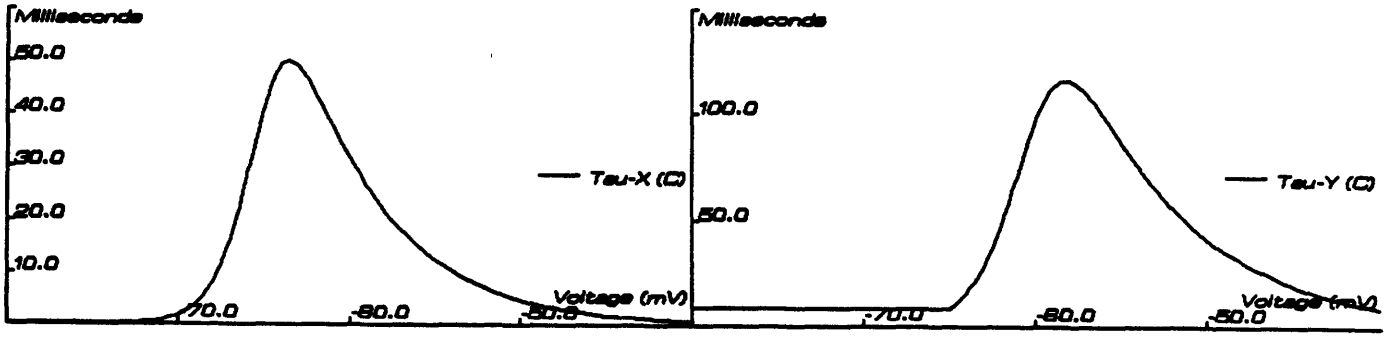


Figure 7.14: Time constant curves (τ_x and τ_y) for x_C and y_C.

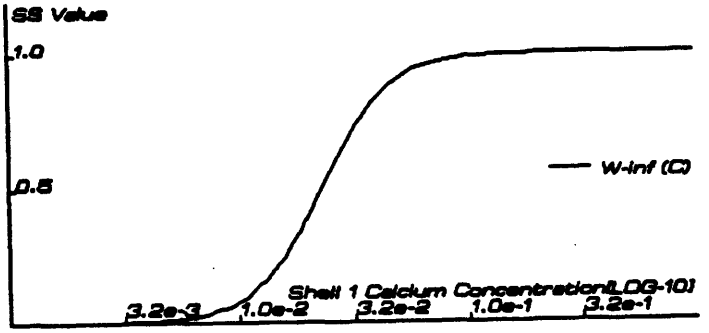


Figure 7.15: Relation between w_∞ and $[Ca^{2+}]_{shell}$ for w_C.

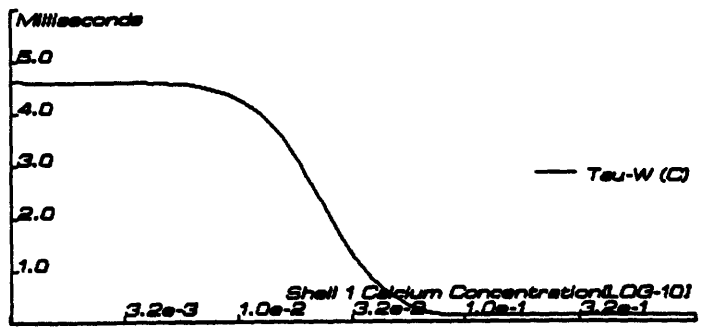


Figure 7.16: Relation between τ_w and $[Ca^{2+}]_{shell}$ for w_C.

7.7 AHP Potassium Current - I_{AHP}

I_{AHP} is a slow, Ca^{2+}-mediated K^+ current that underlies the long after-hyperpolarization (AHP). Typically the AHP is about 1 to 2 millivolts and lasts from 0.5 - 3 seconds after a single spike. Adding Ca^{2+} blockers or noradrenaline to the extracellular medium eliminates the AHP, and likewise markedly reduces the cell's accommodation to tonic stimulus.

Since most of the data on the proposed I_{AHP} is derived from various current clamp protocols, the model description of this current is based on that used in other models (Koch and Adams, 1986) and from heuristics derived from the properties of other currents, in particular I_{Ca} and I_{DR}. The important relationship between the I_{AHP} and I_{DR} parameters arose when I attempted to simulate both the mAHP (mediated by I_{DR}) and the AHP according to data from Storm ( ). In addition, since I_{AHP} is dependent on Ca^{2+} entry, the derivation of this current and the dynamics of $[Ca]_{shell.1}$ and $[Ca]_{shell.2}$ was done simultaneously. In fact, it was determined that in order for the activation of I_{AHP} to be delayed from the onset of the spike, it was necessary to introduce the second intracellular space (shell) that was described in Chapter 6. Such a relationship between Ca^{2+} influx and the subsequent delayed activation of I_{AHP} has been suggested in the literature ([30]).

7.7.1 Results

I propose that the conductance underlying I_{AHP} is dependent both on Ca^{2+} and voltage. The Ca^{2+} dependence of this current is clearly demonstrated

since the AHP is removed when Ca^{2+} blockers are added. and construction of a reasonable model of Ca^{2+} dynamics such that I_{AHP} may be dependent on this is possible.

The mechanism that I use for Ca^{2+}-mediation of I_{AHP} is similar to that for I_C. that is the I_{AHP} channel includes a single Ca^{2+}-binding w particle, with the same binding reaction as shown in Equation x.

Voltage-clamp studies ([30]) indicate that there is no voltage-dependent activation of I_{AHP}, however. This puts a greater constraint on the Ca^{2+}-mediated mechanism for this current since the activation necessary to underly the long, small hyperpolarization after a single spike is significantly less than that required to squelch rapid spikes after some delay in response to tonic stimulus. In particular, these requirements provided rather restricted constraints on the buildup of Ca^{2+} during each spike in region of the I_{AHP} channels, *shell*.2, and likewise the dependence of the I_{AHP} w particle on this localized concentration of Ca^{2+} .

On the other hand I have included two inactivation gating particles. y and z. The rationale for the y particle is based on two pieces of evidence. First, it has been reported that Ca^{2+} spikes are insensitive to noradrenaline in protocols where I_{DR} and I_A have been blocked by TEA and 4-AP, respectively (Segal and Barker). The fact that these spikes are unchanged with the addition of noradrenaline implies that under this protocol I_{AHP} is inactivated by some other mechanism, since presumably I_{AHP} has not been disabled. Since the protocol involves a long (approximately 30 milliseconds) depolarization of the cell before the Ca^{2+} spike, it was possible to include an inactivation particle for I_{AHP} that was (a) fast enough to disable I_{AHP} under these conditions, but (b) was slow enough so that normal spiking did not cause the y particle to change states.

A second indication for the voltage-dependent inactivation particle y is consistent with the previous evidence. that is the amplitude and rate of rise of action potentials singly or in trains appears independent of the presence I_{AHP}. In particular, the size of the I_{AHP} conductance necessary to repress repetitive firing is large enough to significantly effect the spike once threshold is achieved if this conductance remained during the spike. Such a role for I_{AHP} has not been demonstrated. y therefore causes I_{AHP} to shut off during an action potential so that this current does not reduce the amplitude of the spike.

The second inactivation particle, z, was included to account for the delayed peak seen in the large afterhyperpolarization that occurs after a long (greater than 100 ms) stimulus (Madison and Nicoll, 1982 and others). At

rest, z is partially closed. With a large, lengthy hyperpolarization the z particle becomes more open, thereby slowly increasing I_{AHP} and the magnitude of the sAHP, until the Ca^{2+} in *shell.2* eventually drains down to its resting level and subsequently shutting off w. The time constant for z was set very slow above rest so that it did not change appreciably during firing. Below about -75 mV, however, the time constant approaches 120 milliseconds so that the desired role of z during the sAHP is obtained.

No voltage-dependence for I_{AHP} has been noted in the literature. However, the dependence of I_{AHP} on Ca^{2+} influx may have precluded voltage-clamp experiments which might verify the voltage-dependencies indicated by the simulations.

With the present formulation for I_{AHP}, this current plays an important role during repetitive firing by shutting off the spike train after several hundred milliseconds. This occurs primarily through the dependence of I_{AHP} on $[Ca]_{shell.2}$, which slowly increases during repetitive firing. Eventually the rise of $[Ca]_{shell.2}$ causes I_{AHP} to provide sufficient outward rectification for counter-acting the stimulus current and thus stop the cell from firing (Figure 7.19). The fact that I_{AHP} is strongly activated by this protocol is indicated by the long hyperpolarization at the end of the stimulus (Madison and Nicoll, 1982, and see simulation of their results in Figure 7.19). Madison and Nicoll, 1982 [32] report that noradrenaline blocks accommodation by selectively blocking I_{AHP}.

The characteristics demonstrated by the model I_{AHP} are in qualitative agreement with many of the characteristics reported in the literature (e.g. [30] , [41])., including the increased activation of I_{AHP} with increasing numbers of spikes in a single train, delayed activation from onset of Ca^{2+} influx, the role of I_{AHP} in modulating repetitive firing, time constant for inactivation/deactivation of greater than one second, the apparent voltage insensitivity (the transition of y and z with sub-threshold depolarization is slow, and once x is activated deactivation takes several seconds.

The action of I_{AHP} resulting in the sAHP is illustrated in Figure 7.17. Expanded view of this figure (same simulation as Figure 7.11) is shown in Figure 7.18.

The equation for I_{AHP} is -

$$I_{AHP} = y_{AHP}^2 z_{AHP} w_{ahp} (V - E_K)$$

where

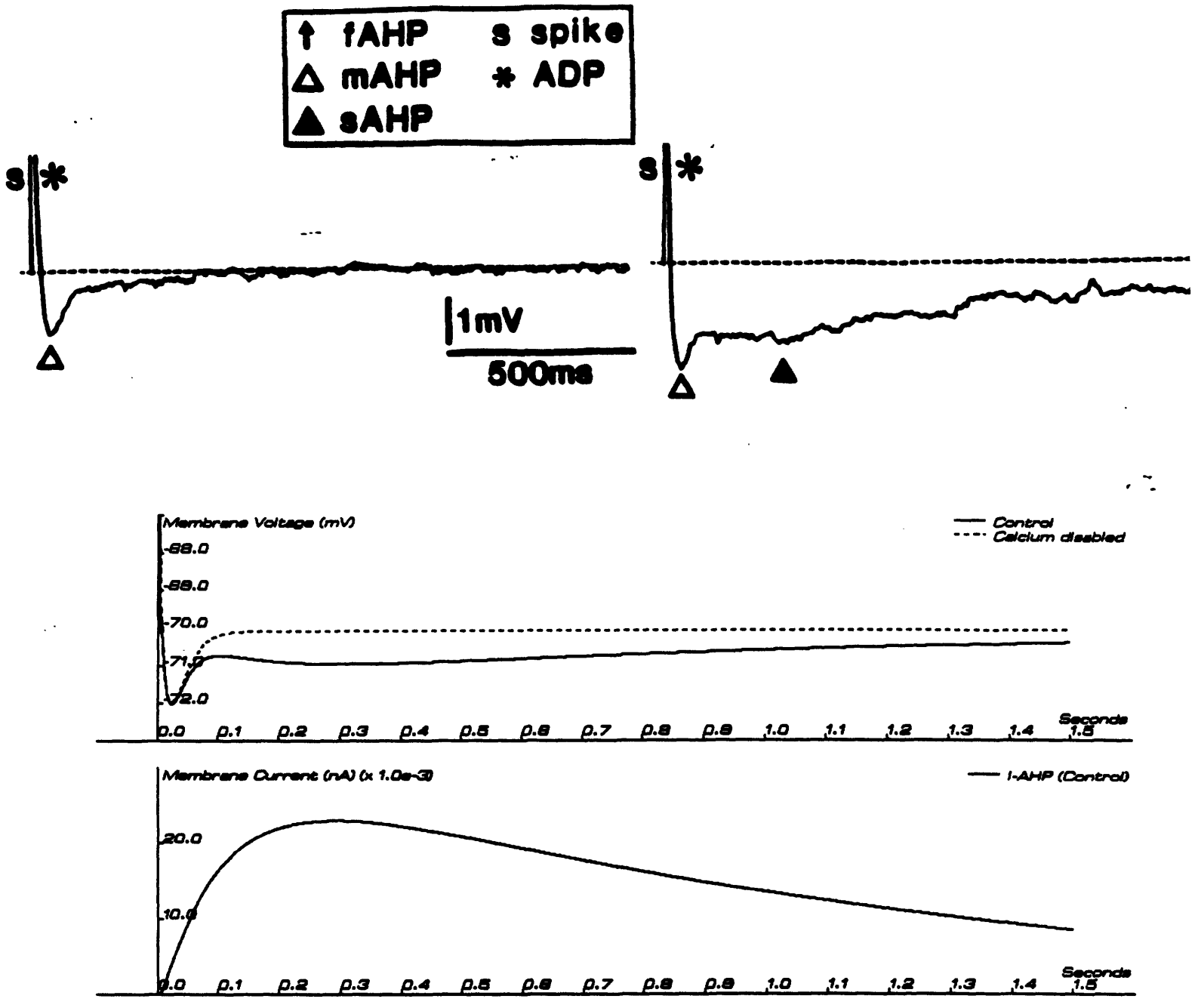


Figure 7.17: Top - Repolarization of the action potential with and without Ca^{2+} blockers (Storm). Note absence of sAHP when Ca^{2+} blockers are added. Bottom - Simulation of protocol.

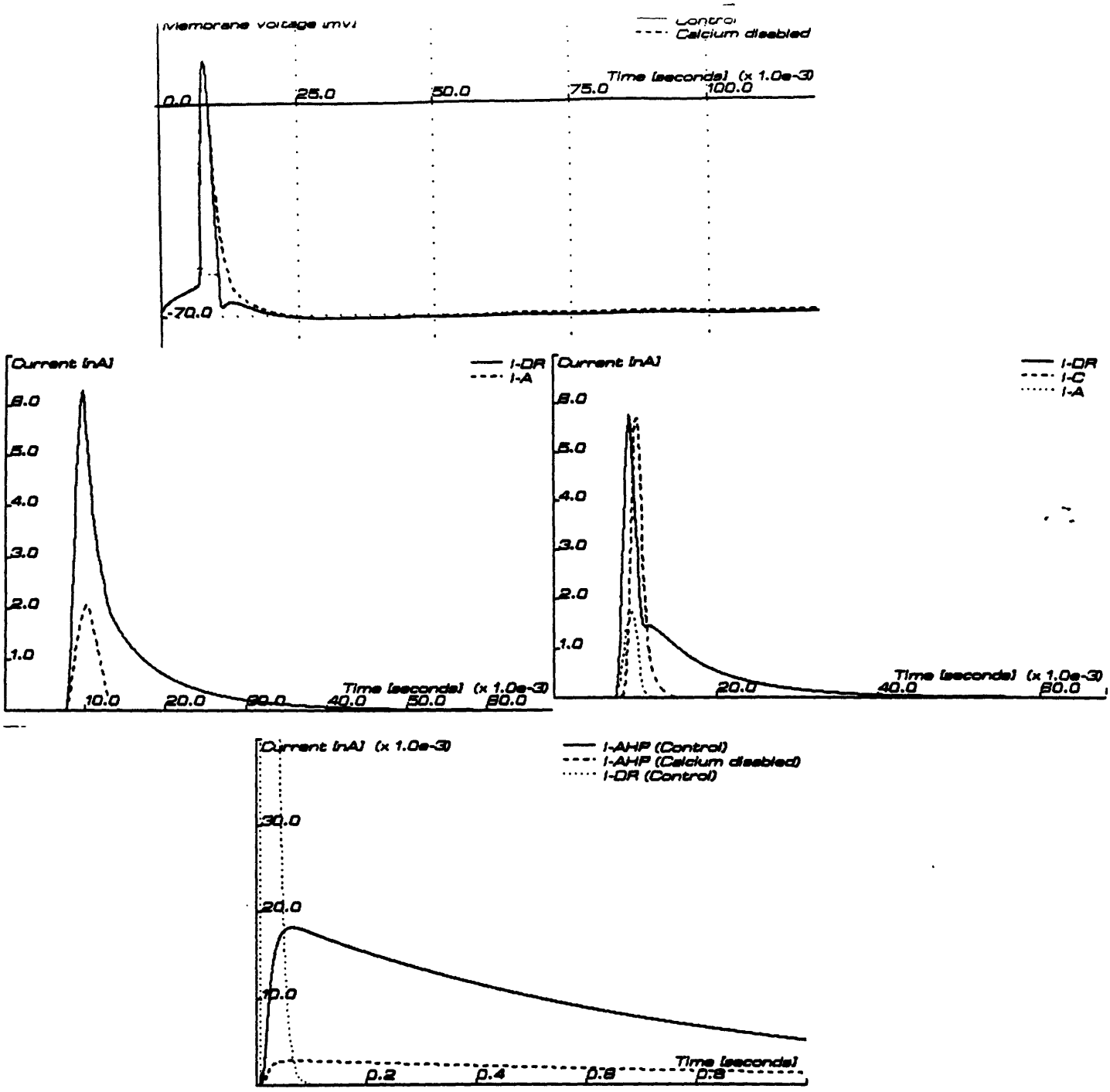


Figure 7.18: A. Simulation of Figure 7.11 showing entire action potential, with and without Ca^{2+} blockers. B. I_C, I_{DR} and I_A when Ca^{2+} blockers absent. C. I_{DR} and I_A when Ca^{2+} blockers present. D. I_{AHP} and I_{DR} with and without Ca^{2+} blockers. I_C and I_{DR} are the principle repolarizing currents when there are no Ca^{2+} blockers, and I_{DR} increases when I_C is disabled. Also the fast time course of I_C allows it to produce the sharp fAHP.

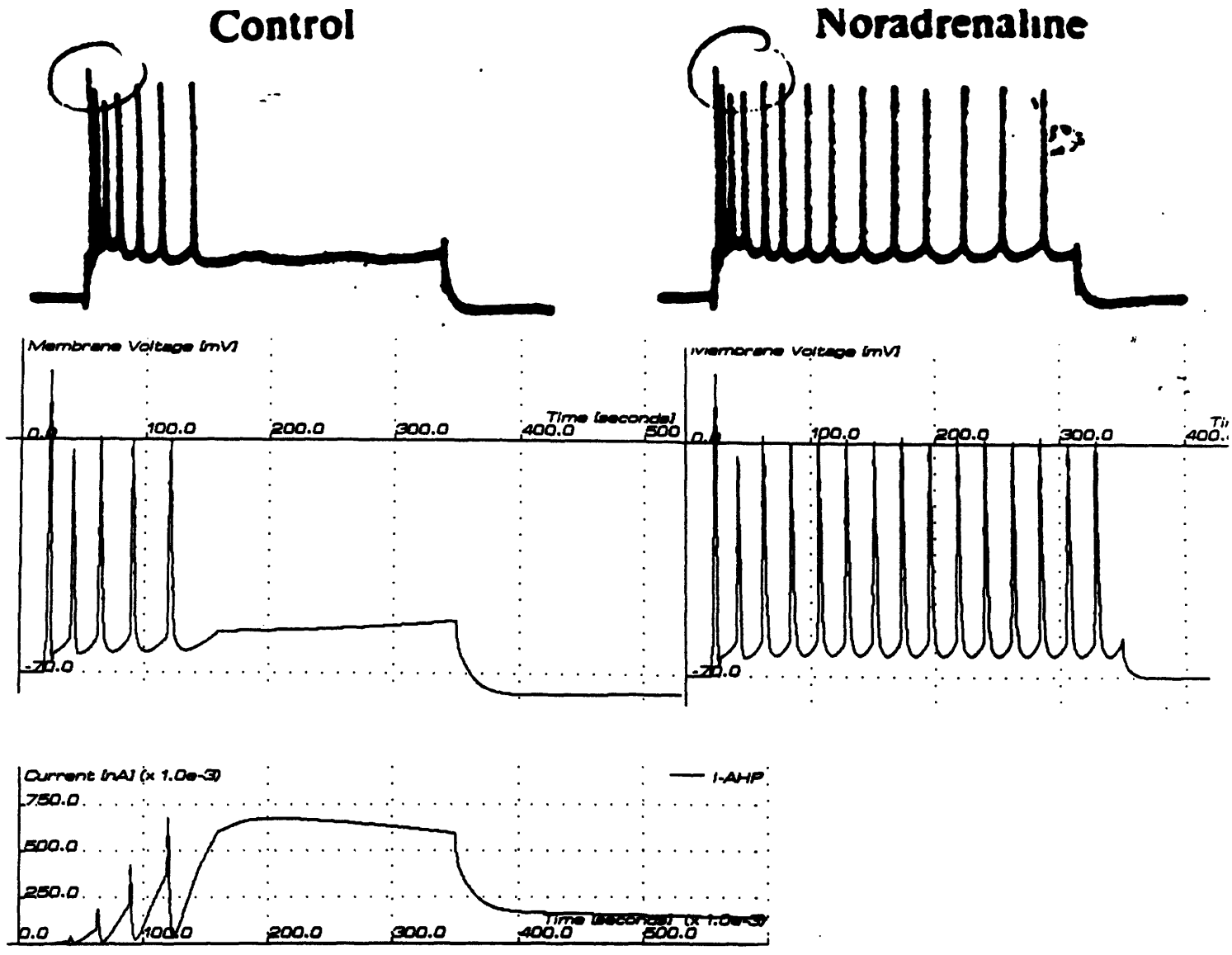


Figure 7.19: Influence of I_{AHP} on accommodation. Top - Repetitive firing in response to tonic depolarizing stimulus with and without noradrenaline [32]. Middle - Simulation of above responses. Bottom - I_{AHP} during simulations. When I_{AHP} is enabled there is a gradual rise in this conductance with each spike until subsequent spikes are blocked.

Gating Variable	z	γ	α_0	$V_{\frac{1}{2}}$ (mV)	τ_0 (ms)	α^*_{Ca}	β^{**}_{Ca}
y (inactivation)	-15	0.8	0.015	-50.0	2.5	-	-
z (inactivation)	-12	0	0.0002	-72.0	120.0	-	-
w (Ca^{2+}-activation)	-	-	-	-	-	10^5	0.005

Table 7.5: Parameters of I_{AHP} Gating Variables. $* = (mS^{-1}mM^{-3})$, $** = (mS^{-1})$

$$\bar{g}_{AHP} = 0.35\,\mu\text{S}$$

Table 7.5 lists the parameters for the I_{AHP} gating variables. These are the rate functions for the activation variable, x, of I_{AHP}-

$$\alpha_{y\_AHP} = 0.015 \exp\left(\frac{(V+50)0.8 \cdot -15 \cdot F}{RT}\right)$$

$$\beta_{y\_AHP} = 0.015 \exp\left(\frac{(-50-V)0.2 \cdot -15 \cdot F}{RT}\right)$$

These are the rate functions for the activation variable, y, of I_{AHP}-

$$\alpha_{z\_AHP} = 0 (\gamma = 0)$$

$$\beta_{z\_AHP} = 0.0002 \exp\left(\frac{(-72-V) \cdot -12 \cdot F}{RT}\right)$$

Figure 7.20 and Figure 7.21 show the voltage dependence on the steady-state values and the time constants for the x_{AHP} and y_{AHP} variables.

Again, each w particle was assumed to have three non-competitive Ca^{2+} binding sites, all of which were either empty (corresponding to the closed state) or filled (corresponding to the open state). Figure 7.22 shows the dependence of the steady-state value of the w_{AHP} variable on $[Ca^{2+}]_{shell.2}$. Figure 7.23 shows the dependence of the time constant for the w_{AHP} variable on $[Ca^{2+}]_{shell.2}$.

7.8 M-Current Potassium Current - I_M

I_M is a small persistent K^+ current that is activated near rest and that is selectively inhibited by muscarinic agonists ([16]). This current has been

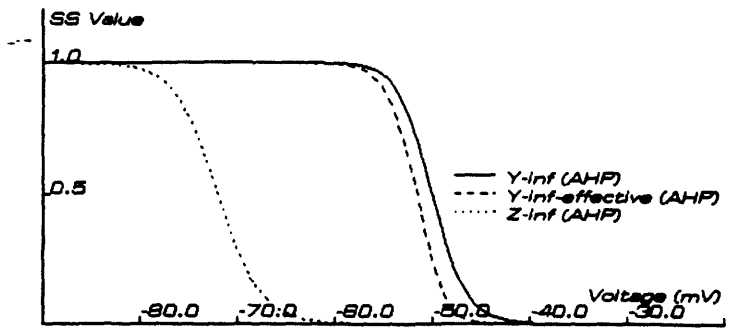


Figure 7.20: Steady-state curves (x_∞, y_∞) for x_{AHP} and y_{AHP} and effective curves as would be measured by voltage-clamp experiments.

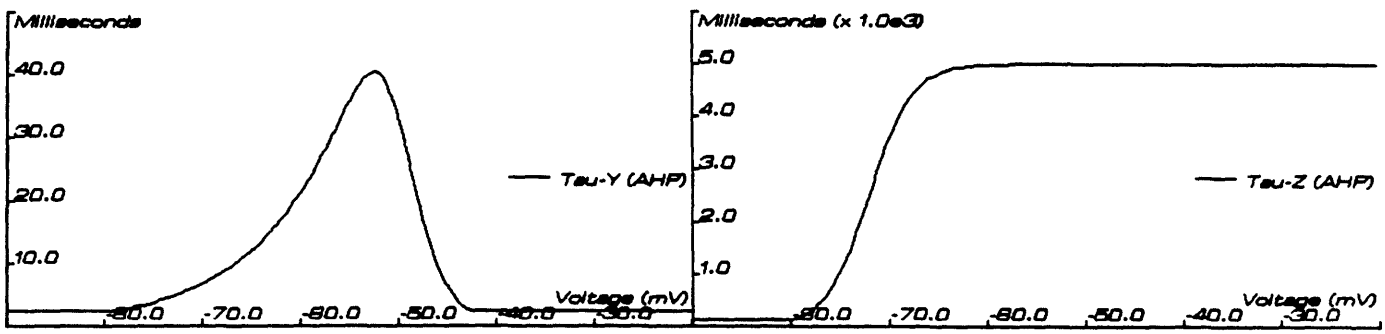


Figure 7.21: Time constant curve (τ_x, τ_y) for x_{AHP} and y_{AHP}.

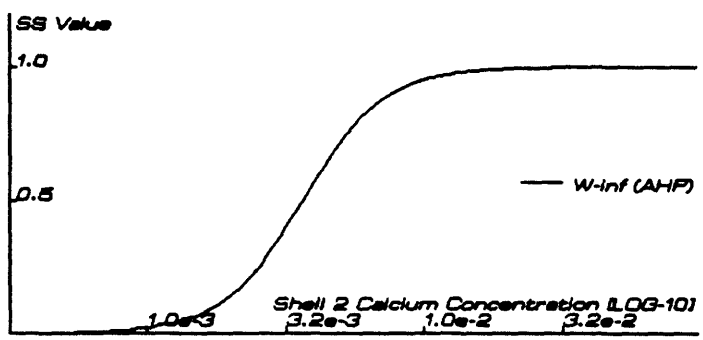


Figure 7.22: Relation between w_∞ and $[Ca^{2+}]_{shell.2}$ for w_{AHP}.

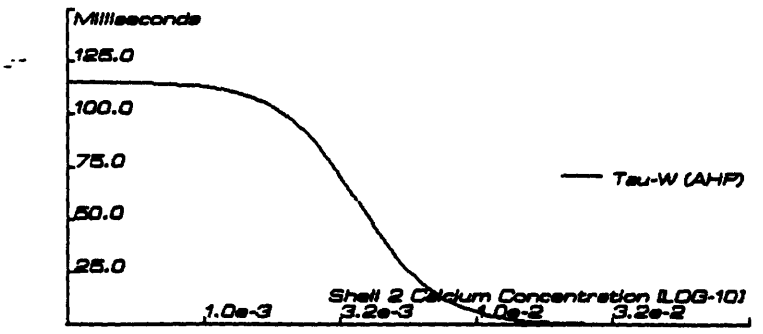


Figure 7.23: Relation between τ_w and $[Ca^{2+}]_{shell.2}$ for w_{AHP}.

reported to contribute to cell excitability and to the mediation of repetitive firing in different species [1].

There is evidence for massive cholinergic projection from the medial septum to the hippocampus ([16], Kuhar, in [24], [25]) so the mediation of HPC behavior by cholinergic agonists via specific currents is potentially a very important mechanism for modulation of either single HPCs or local populations of HPCs.

7.8.1 Results

For the model the steady state parameters of the activation variable of I_M, x, were inferred from [16]. Data on the temporal properties of this current is sparse. The time constant for activation for I_M has been estimated at being between 50 and 300 milliseconds within 20 millivolts of rest. The Q_{10} for I_M has been estimated at 5 ([16]). This means that the I_M is much faster at physiological temperature than would be indicated by the reported data, which was measured at 23°C. Since the activation is slow and no clear recordings of the time course of activation are available, I included a single activation particle, x, in the formula for the I_M conductance.

As mentioned in Chapter 3, originally I assumed that at the resting potential the only conductances that were open were linear, and therefore E_{leak} was set equal to E_{rest} ($= -70$mV). However, the data suggests that at rest a small amount of I_M is activated. In the model, then, inclusion of I_M shifted E_{rest} slightly from -70.0 mV to -72.3 mV.

Current clamp simulations suggest that I_M has two roles: 1) changing

Gating Variable	z	γ	α_0	$V_{\frac{1}{2}}$ (mV)	τ_0 (ms)
x (activation)	5	0.5	0.0015	-45.0	10

Table 7.6: Parameters of I_M Gating Variable

the current stimulus threshold for $I_{Na-trig}$ mediated spikes, and 2) modulating repetitive firing in response to tonic stimulus. The first characteristic comes about since I_M is partially activated at the resting potential, and therefore decreases the input impedance of the cell. Figure 7.24 illustrates that blocking I_M increases the firing frequency of the cell in response to tonic stimulus. However, this increase is much less than that usually reported for cholenergic modification of HPC firing. This result implies that the major effector of the cholinergic response is I_{AHP}, as has been demonstrated earlier.

The equation for I_M is -

$$I_M = \bar{g}_M x_M (V - E_K)$$

Table 7.6 lists the parameters for the I_M gating variable.

These are the rate functions for I_M-

$$\alpha_{x,M} = 0.0015 \exp\left(\frac{(V+45)0.5 \cdot 5 \cdot F}{RT}\right)$$

$$\beta_{x,M} = 0.0015 \exp\left(\frac{(-45-V)0.5 \cdot 5 \cdot F}{RT}\right)$$

Figure 7.25 and Figure 7.26 show the voltage dependence on the steady-state values and the time constants for the x_M and y_M variables.

7.9 Q-Current Potassium Current - I_Q

I_Q is a small current that is activated when the cell is hyperpolarized with respect to resting potential. The reversal potential for the I_Q has been estimated at approximately -80 millivolts. Since E_K is about -90 millivolts, this current might be due to a mixed conductance.

At the present time the functional characteristics of I_Q have not been investigated.

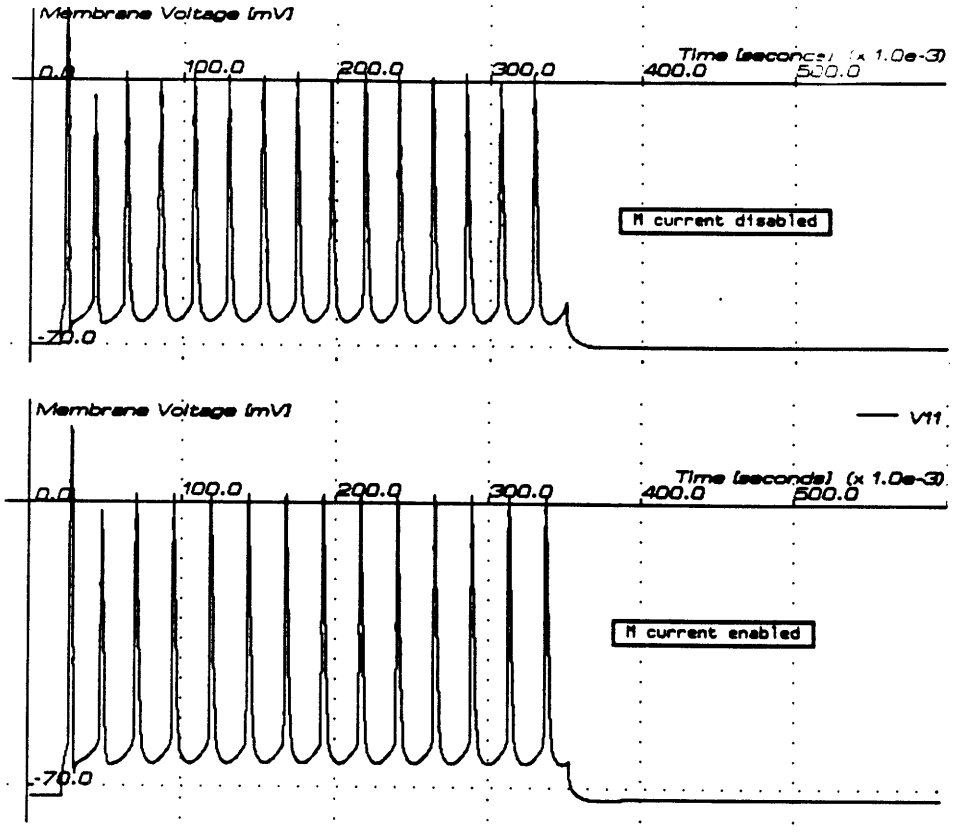


Figure 7.24: Influence of of removal of I_M on accommodation. I_{AHP} blocked to allow immediate repetitive firing. Stimulus 1.0 nA for 300 milliseconds. Top - I_M enabled. Bottom - I_M disabled. Disabling I_M increases the firing rate by about 7 % in this protocol.

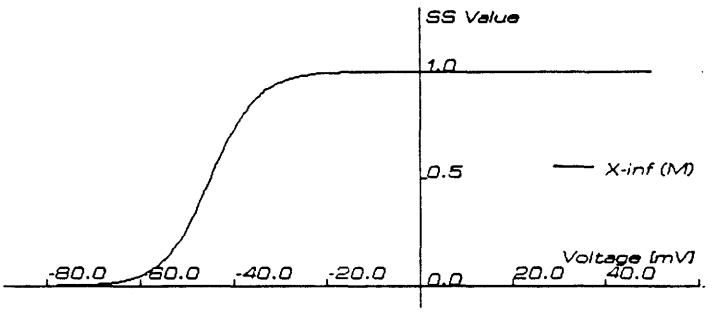


Figure 7.25: Steady-state curve (x_∞) for x_M.

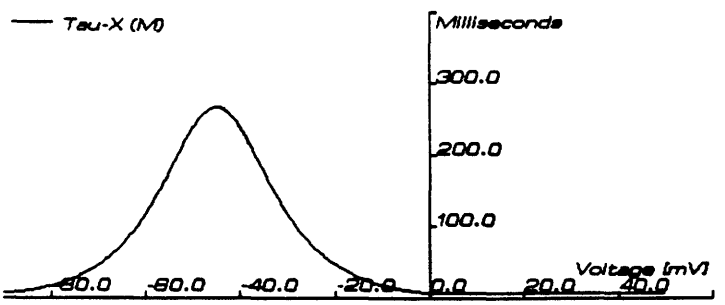


Figure 7.26: Time constant curve (τ_x) for x_M.

Gating Variable	z	γ	α_0	$V_{\frac{1}{2}}$ (mV)	τ_0 (ms)
x (activation)	15	0.98	0.0003	-45.0	6.0

Table 7.7: Parameters of I_Q Gating Variable

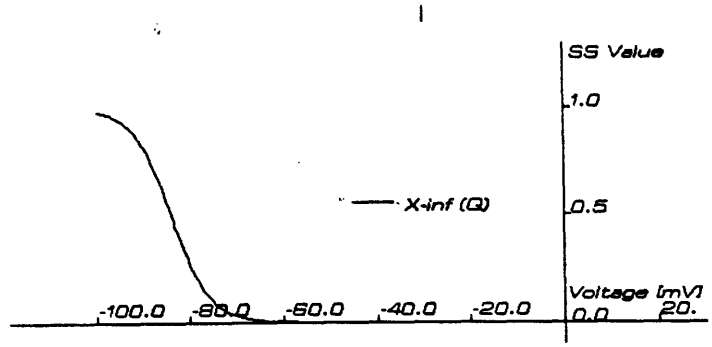


Figure 7.27: Steady-state curve (x_∞) for x_Q.

7.9.1 Results

The current description for I_Q is based solely on the data from [16].

The equation for I_Q is -

$$I_Q = \bar{g}_Q x_Q (V - E_K)$$

where

$$\bar{g}_Q = 0.002\,\mu S$$

Table 7.7 lists the parameters for the I_Q gating variable.

These are the rate functions for the activation variable, x, of I_Q-

$$\alpha_{x,Q} = 0.0003 \exp\left(\frac{(V + 45)0.98 \cdot 15 \cdot F}{RT}\right)$$

$$\beta_{x,Q} = 0.0003 \exp\left(\frac{(-45 - V)0.02 \cdot 15 \cdot F}{RT}\right)$$

Figure 7.27 and Figure 7.28 show the voltage dependence on the steady-state values and the time constants for the x_Q and y_Q variables.

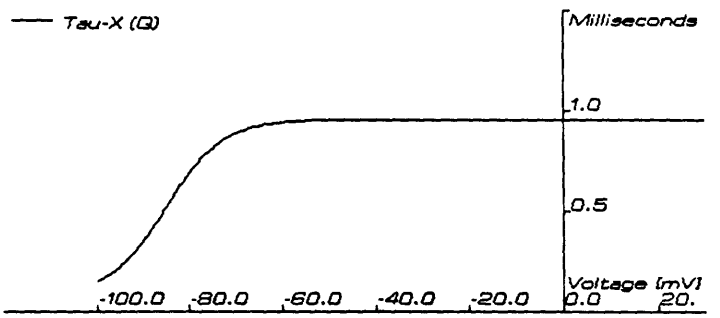


Figure 7.28: Time constant curve (τ_x) for x_Q.

Chapter 8

VOLTAGE CLAMP SIMULATIONS

8.1 Introduction

Simulating voltage clamp data was an important verification of the parameters derived for both currents and the linear structure of the model. In particular, these simulations estimated how much the current flow due to the unclamped dendrites contributed to errors in parameters derived with voltage clamp protocols.

8.2 Non-Ideal Space Clamp

The non-zero R_i means that a voltage clamp applied at the soma will not clamp the dendrites ideally. This distortion of the clamp signal is shown in Figure 8.1, where the soma has been clamped to -50 mV from a resting potential of -70 mV for 50 milliseconds. The distortion of the clamp voltage has two components. First, the rise time of the clamp signal gets progressively longer further down the dendrite. Second, the final voltage reached is lower the further down the dendrite.

If we assume that the non-linear conductances are perfectly segregated in the soma, with the dendrite being linear, then this situation is not intractable. In this case the protocol will perform the correct voltage clamp on the non-linear conductances, and the current that passes through them will be a function of only the time, the holding potential, and the clamp potential. There will be a component of the clamp current due to the non-

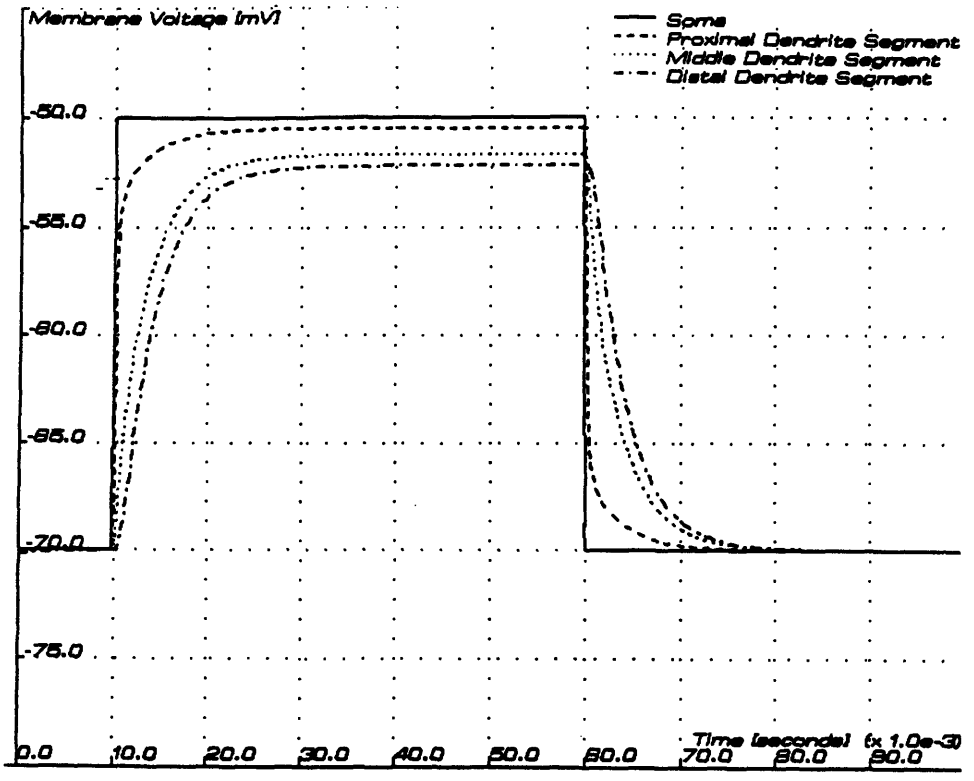


Figure 8.1: Voltage clamp simulation with clamp applied to the soma. The voltage step seen by the different parts of the dendrite membrane is distorted.

ideal clamping of the dendrite, but this current may be compensated for by estimating the linear properties of the dendrite.

On the other hand, no clear cut distribution of non-linear conductances is more likely: there may be significant non-linear membrane in dendrite that is at a significant electrotonic distance from the soma. Referring to Figure 8.1, if there is any non-linear conductance in the proximal dendrite (in this simulation this refers to the proximal 240 μm of dendrite), then the steady state voltage caused by the clamp is not very different than the soma voltage (-50.5 mV and -50.0 mV, respectively). However, during the initial 10 milliseconds of the clamp the voltage in the proximal dendrite is significantly different, and therefore any non-linear membrane in this region will be poorly clamped. This will be a problem if the conductances are fast in this region, e.g. activation or inactivation on the order of several milliseconds. Although the results of the analysis presented in Chapter 3 indicates that the dendrite are electrically compact, the high value for R_i that has been proposed causes the dendrite voltage to lag significantly behind the soma voltage.

8.3 Contamination of Na^+ Parameters Derived by Voltage Clamp

The problem of unclamped dendrites is most severe for the faster currents. This can be demonstrated by examining the putative Na^+ currents with voltage clamp simulations.

The simulated voltage clamp protocol in which all currents except for Na^+ currents have been blocked is shown in Figure 8.2.

Ideally, the soma-dendrite current may be measured in isolation by running the voltage clamp on a cell where all the non-linear currents have been blocked. This current may then be subtracted from the clamp current when the all none-Na^+ components are blocked or disabled in order to estimate the Na^+ currents.

Figure 8.3 illustrates the various components of the response of a -70 to 0 mV voltage clamp of just the Na^+ currents. The soma-dendrite current contributes substantially to the clamp current. If this current is not taken into account then the estimated Na^+ component will be significantly smaller and faster than the actual Na^+ component.

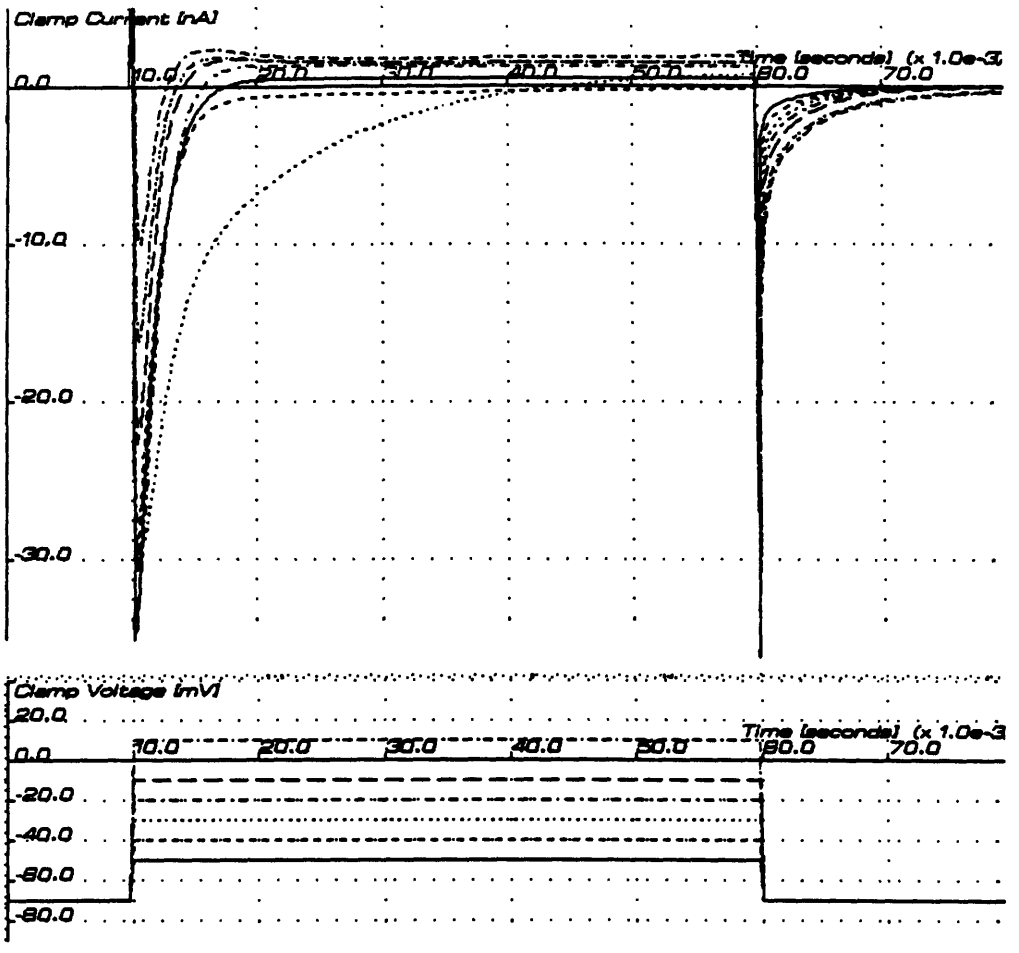


Figure 8.2: Voltage clamp simulations in which all currents have been blocked except for the Na^+ currents.

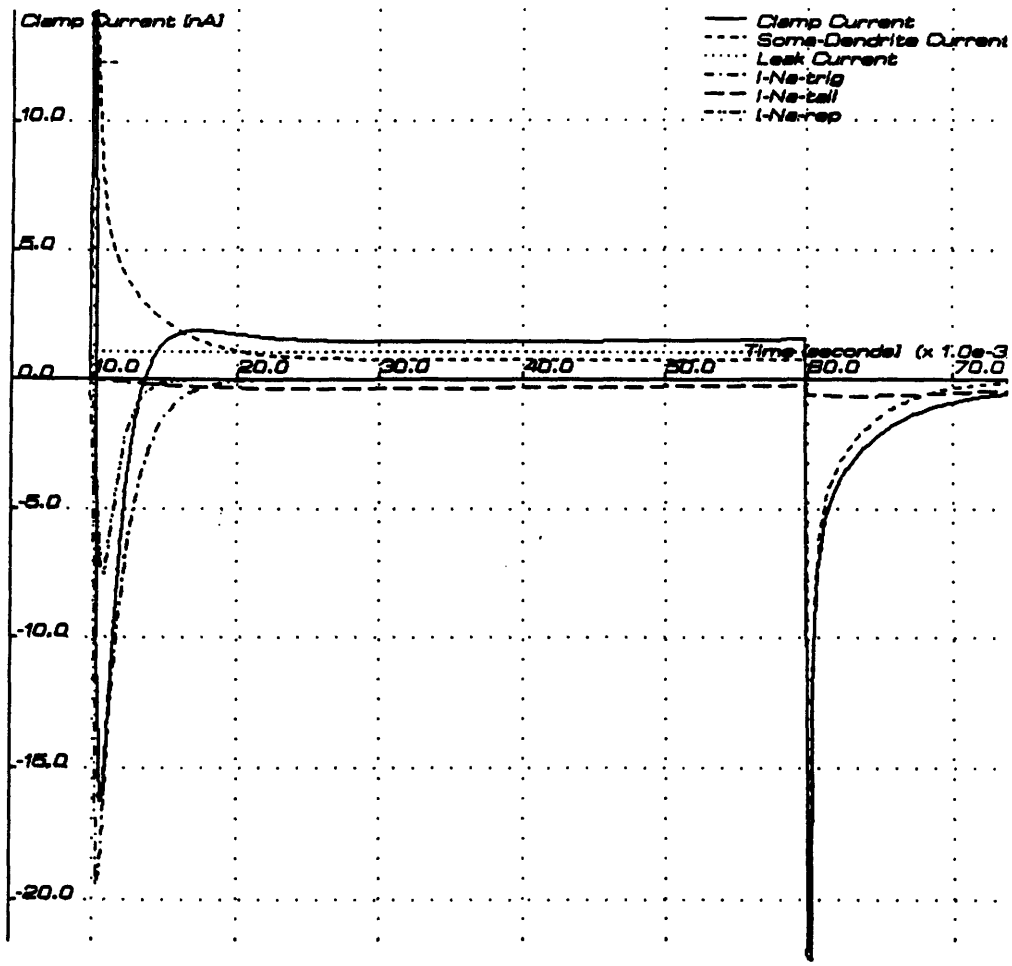


Figure 8.3: Voltage clamp simulation under same conditions of Figure 8.2 illustrating different components of the clamp current. A considerable portion of the clamp current is due to delayed charging of the dendrite's distributed capacitance. Clamp step is from -70 mV to 0 mV for 50 milliseconds.

Chapter 9

CURRENT CLAMP SIMULATIONS

9.1 Introduction

This chapter presents some illustrative simulations of current clamp protocols that demonstrate the overall behavior of the model. The behavior of the model under various protocols is examined, assuming that the current descriptions based on simulation of actual data, as demonstrated in Chapters 5, 6, and 7, sets a realistic stage for more speculative simulations.

Finally, a typical simulation will be presented along with the records of the major currents and the time course of their gating variables in order to demonstrate the full output of HIPPO.

9.2 Regulation of Repetitive Firing – Effect of Blockade of Specific Currents

Consider Figure 9.1. In this simulation the response to long tonic stimuli of different strengths is demonstrate with all the HIPPO currents present. For most of the stimuli the cell responds with an initial burst of action potentials followed by a slow train of spikes.

As we have seen in the previous chapters the major mechanisms mediating theses responses is the buildup of intracellular Ca^{2+}, the subsequent activation of I_{AHP} and the high, broad threshold of I_{Na-rep}. The burst phase is mediated by $I_{Na-trig}$, but once I_{AHP} becomes large enough.

another $I_{Na-trig}$-mediated spike is delayed by the outward rectification of I_{AHP}.

Investigating what happens under the same stimulus protocols when specific currents are blocked illustrates some of the predictions of the model. First, note Figures 9.2, 9.3, and 9.4, which show spike trains in which I_A, I_M, and $I_{Na-tail}$ have been blocked, respectively. The model results suggest that although I_A can have a significant role in delaying the onset of firing and in modulating the width of the spike (ref. Chapter 7), this current does not alter repetitive firing once it has been initiated. I_M and $I_{Na-tail}$, on the other hand, appear to have little functional role under any of the protocols presented. If these currents (I_M and $I_{Na-tail}$) are assumed to be in the HPC for a reason, then this result suggests that either the model description for them is incomplete or that their site of action is primarily non-somatic (see Section 10.2.4).

In Figures 9.5, 9.6, and 9.7, the response of the cell to tonic stimulus is shown where the Ca^{2+} currents, I_C, and I_{AHP} have been blocked, respectively. Here some marked departures from the response of Figure 9.1 can be seen, in particular the change in accommodation. When all Ca^{2+} activity is blocked, as in Figure 9.5, the frequency of repetitive firing is constant, that is the cell reaches a steady-state condition immediately with the first action potential.

Figures 9.6 and 9.7 show how I_C and I_{AHP} *both* contribute to the accommodation response. When I_C is blocked the initial accommodation is quite similar to the normal response. However (especially for the stronger stimuli), later in the spike train the frequency of firing begins to increase slightly, as if I_{AHP} was partially wearing out in its role as headmaster. When I_{AHP} is selectively blocked, on the other hand, accommodation is immediately compromised and the burst phase lasts for many spikes. Eventually a reduced accommodation starts, though, slowly putting on the brakes to prevent excessive spiking.

This accommodation is due to I_C, as can be seen in Figure 9.8. In this figure the response to a 2.2 nA tonic stimulus as was shown in the previous figure is reproduced along with the time course of the intracellular Ca^{2+} concentrations and I_C. Here it can be seen that at the beginning of the train I_C fulfills its normal role as a transient repolarizing agent, active only during the spike. When the subsequent spikes come too fast, however, the concentration of Ca^{2+} in *shell*.2 has a chance to build up, thereby raising the basal level of $[Ca^{2+}]_{shell.1}$ between spikes. This rise is enough to activate I_C

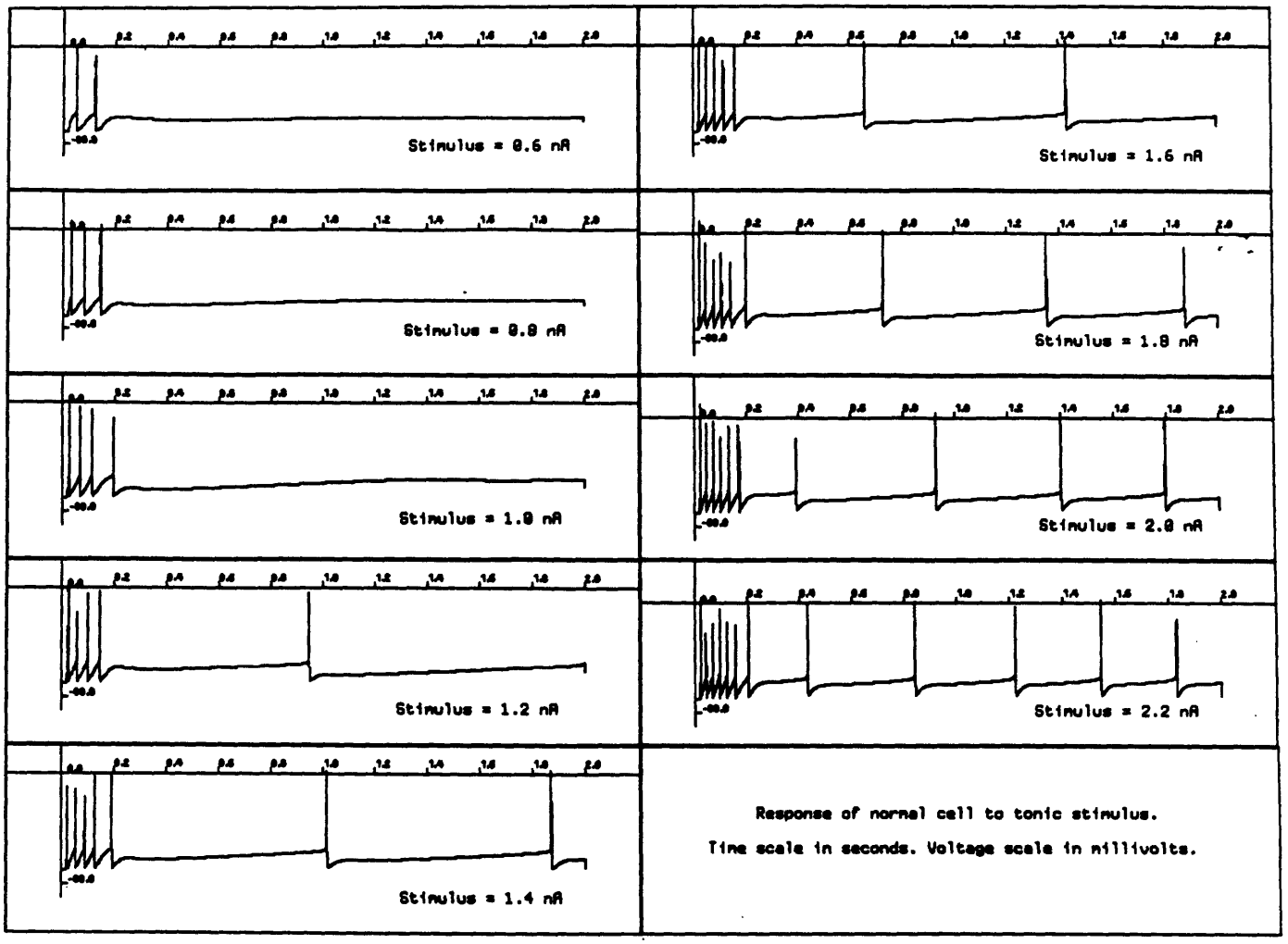


Figure 9.1: Simulations of response to tonic somatic input. Stimulus strengths include 0.6 nA. 0.8 nA. 1.0 nA. 1.2 nA. 1.4 nA. 1.6 nA. 1.8 nA. 2.0 nA. and 2.2 nA.

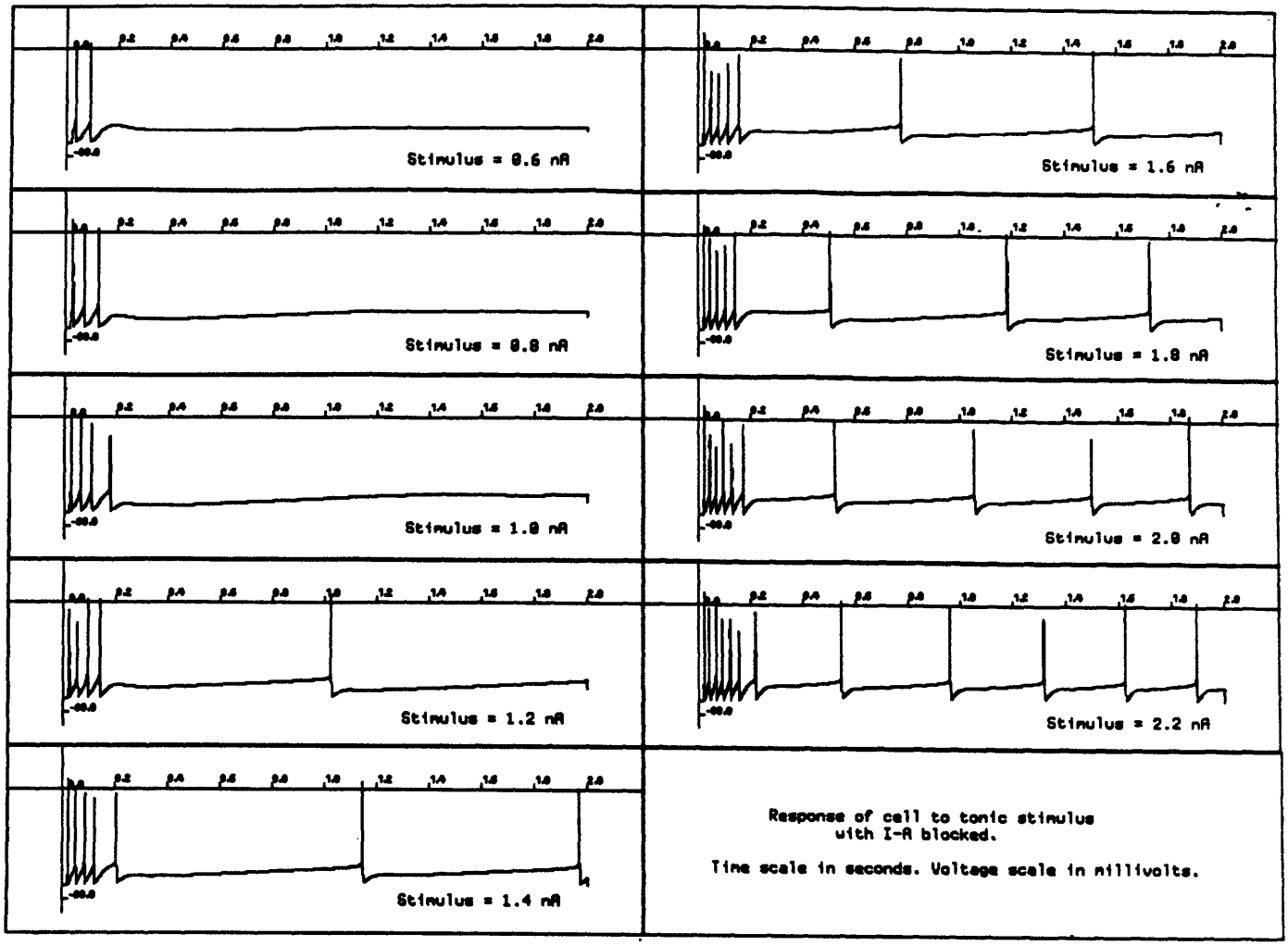


Figure 9.2: Simulations of response to tonic somatic input with I_A disabled. Stimulus strengths include 0.6 nA. 0.8 nA. 1.0 nA. 1.2 nA. 1.4 nA. 1.6 nA. 1.8 nA. 2.0 nA. and 2.2 nA.

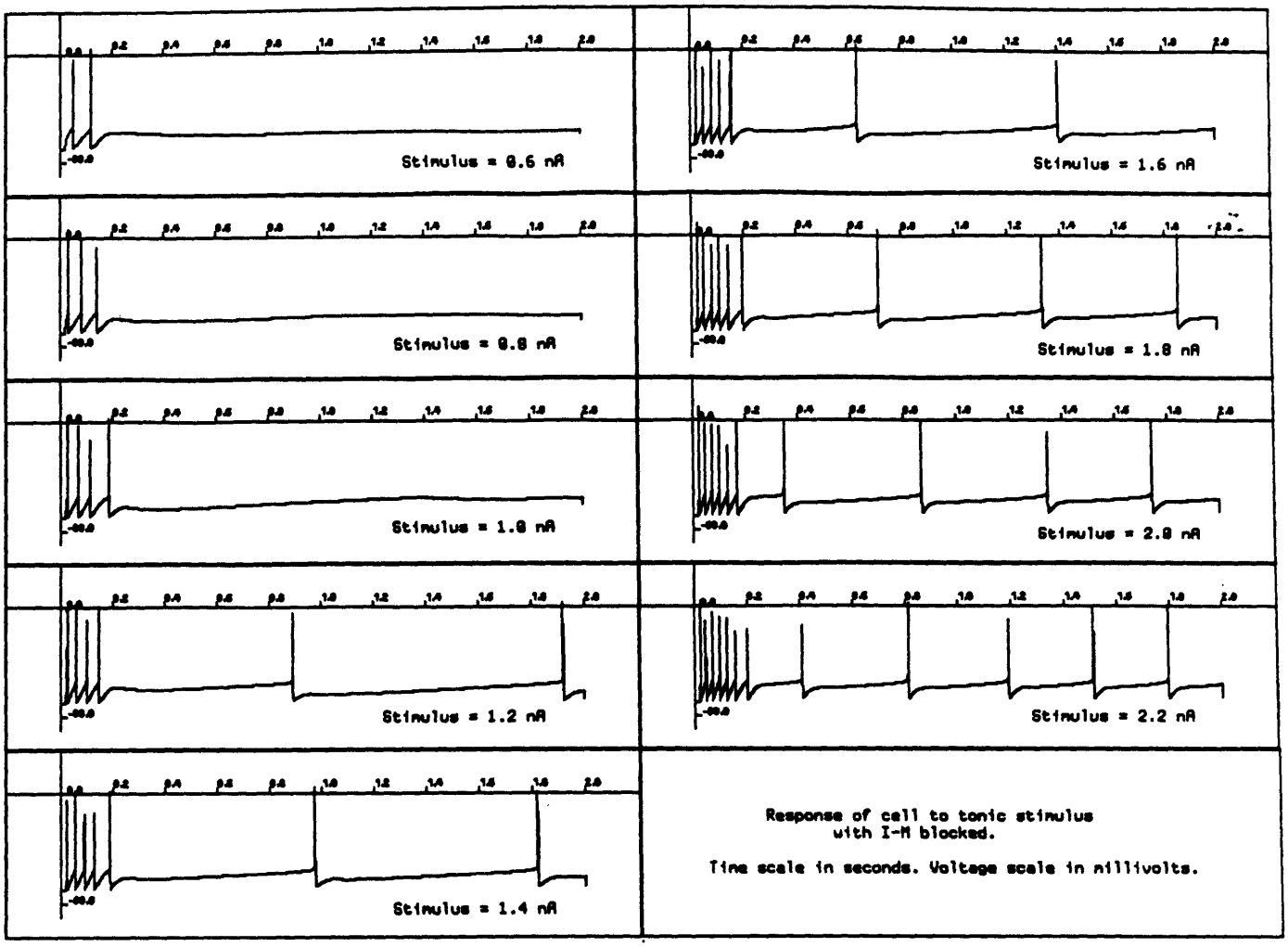


Figure 9.3: Simulations of response to tonic somatic input with I_M disabled. Stimulus strengths include 0.6 nA. 0.8 nA. 1.0 nA. 1.2 nA. 1.4 nA. 1.6 nA. 1.8 nA. 2.0 nA. and 2.2 nA.

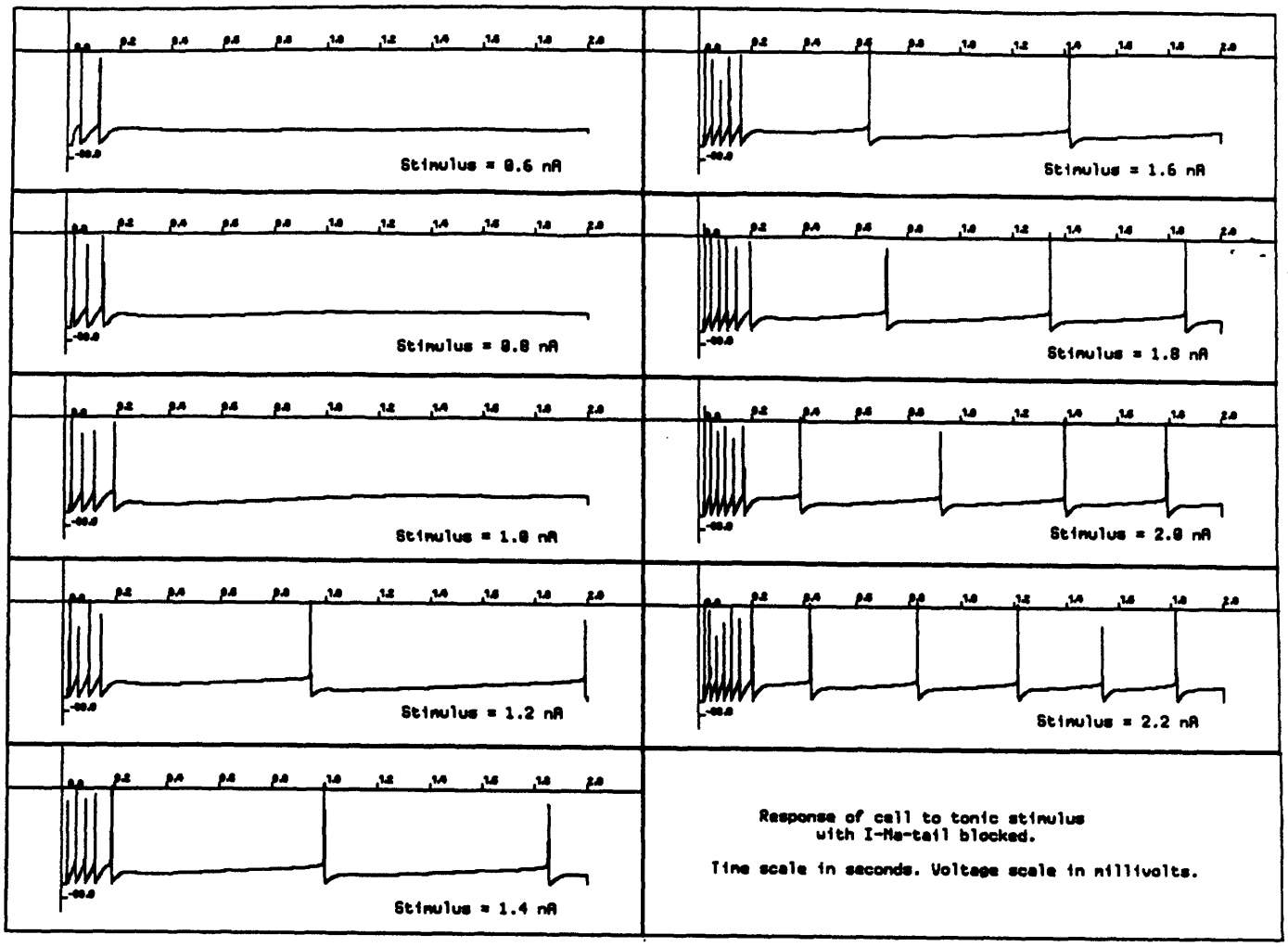


Figure 9.4: Simulations of response to tonic somatic input with $I_{Na-tail}$ disabled. Stimulus strengths include 0.6 nA, 0.8 nA, 1.0 nA, 1.2 nA, 1.4 nA, 1.6 nA, 1.8 nA, 2.0 nA, and 2.2 nA.

interictally so that this current provides a suppressing influence to the latter part of the train. The simulations suggest, among other things, that I_C may have a dual role - under "normal" conditions I_C just modulates the width of the individual spike , and under conditions when I_{AHP} is blocked (which easily could be physiological considering I_{AHP} is inhibited by cholenergic agonists) I_C steps in to provide a controlling influence suppressing strong repetitive firing.

Finally, Figure 9.9 shows the response of the cell with just I_{Na-rep} blocked. Here a fairly bizzare response is seen, since it seems that this current would only contribute to the strength of individual spikes and the extension of the effective range for firing threshold. However, in these simulations removal of I_{Na-rep} has an additional (possibly pathologic) effect. At low stimuli, the standard burst/accommodation response is seen, but as the stimulus intensity is increased the cell response degenerates into a series of low amplitude Ca^{2+} spikes followed by a cessation of activity - the cell effectively becomes mute.

In Figures 9.10 through 9.16, this response is examined more closely and compared with the response of the normal cell to the same stimulus. In Figure 9.10 the two responses and their current records are compared over the entire 2 seconds. At this level the most striking difference is the large I_{DR} and I_{Ca} (also $[Ca^{2+}]_{shell.1}$ and $[Ca^{2+}]_{shell.2}$) in the latter portion of the I_{Na-rep}-blocked response. Examining the first part of the response in detail (Figures 9.12 and 9.13), the differences are not as obvious. However, even though $I_{Na-trig}$ is about the same for the two protocols, the blocking of I_{Na-rep} causes a significant reduction in the spike amplitude. The result is that I_{DR} is not activated as strongly as in the normal case, thereby reducing the interictal hyperpolarization and increasing the frequency of firing. This is shown more clearly in Figures 9.14 and 9.15, where the initial spikes for both responses are shown. Other than these changes, however, it appears that nothing degenerate is occurring.

The situation changes, though, as accommodation (mediated by I_{AHP}) sets in, as shown in Figure 9.16, where the activated I_{AHP} reduces the amplitude of the later spikes even further, which in turn prevents the full turning on of I_{DR}. The net result is that the cell becomes more depolarized on the average, allowing I_{Ca} to fully activate. This inward current, while now superseding $I_{Na-trig}$ as the "spike" current, cannot depolarize the cell enough to activate I_{DR} fully, which could repolarize the cell back to E_{rest} and reset the firing cycle. Eventually, then, after a few oscillatory Ca^{2+}-spikes,

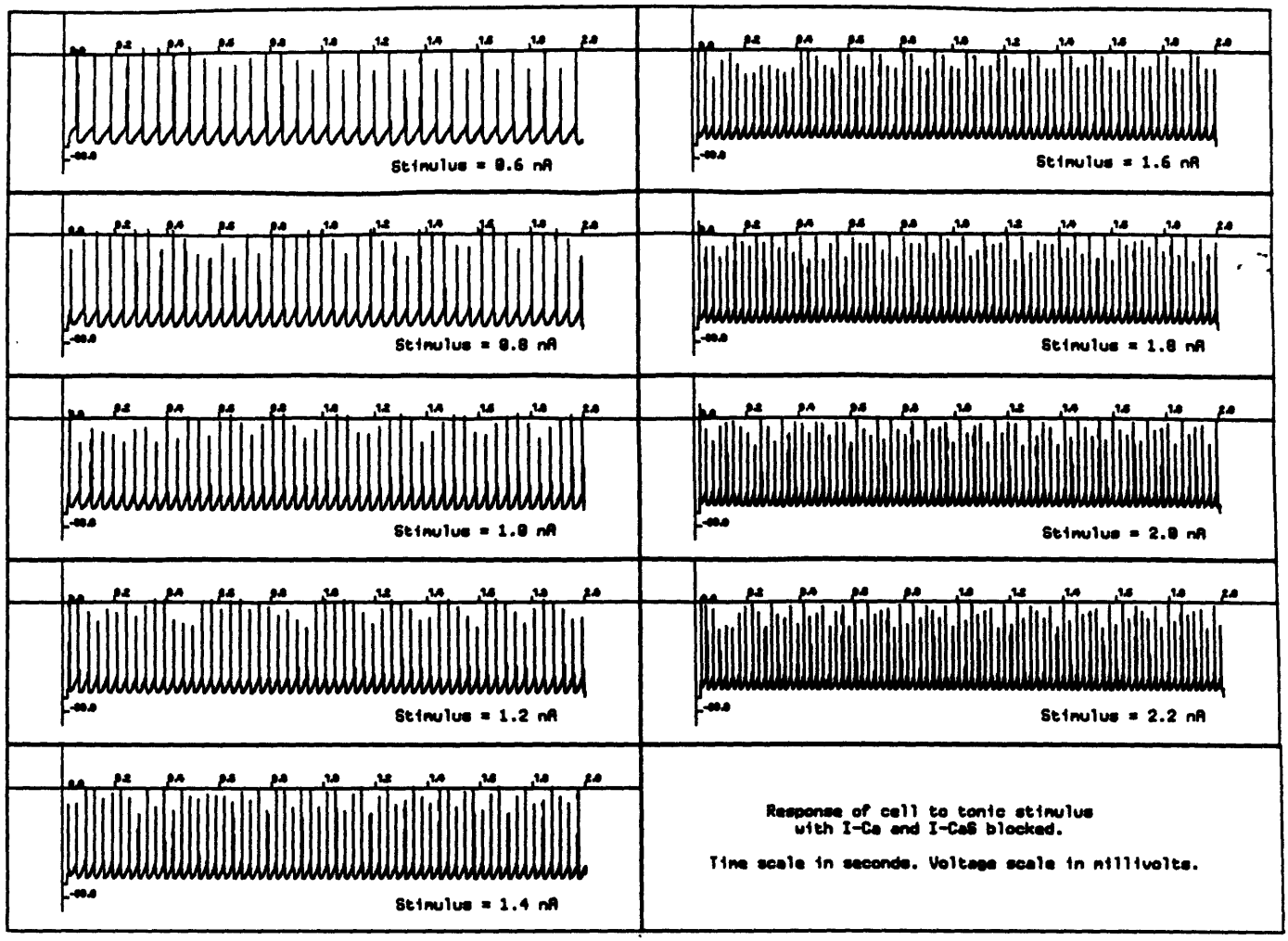


Figure 9.5: Simulations of response to tonic somatic input with I_{Ca} and I_{CaS} disabled. Stimulus strengths include 0.6 nA, 0.8 nA, 1.0 nA, 1.2 nA, 1.4 nA, 1.6 nA, 1.8 nA, 2.0 nA, and 2.2 nA.

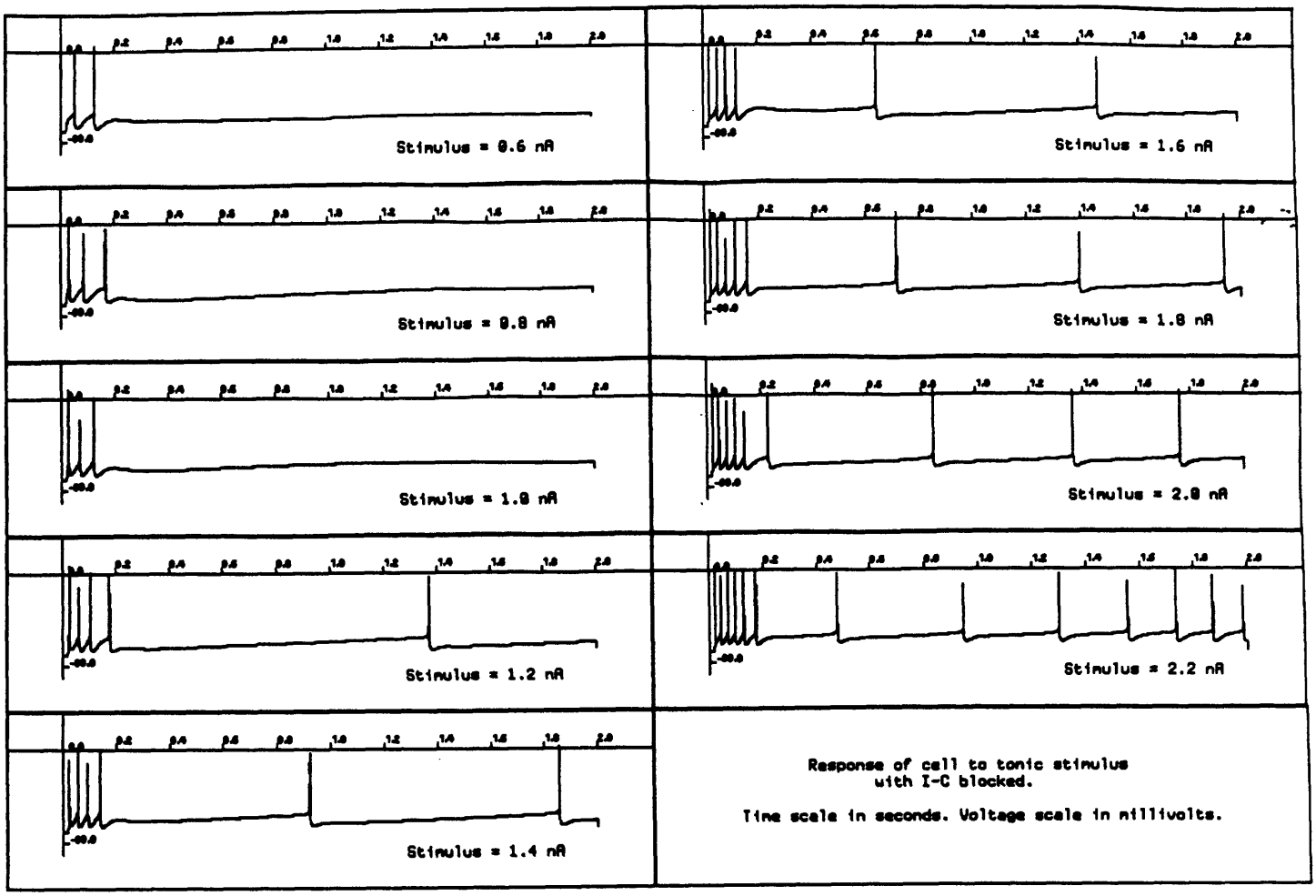


Figure 9.6: Simulations of response to tonic somatic input with I_C disabled. Stimulus strengths include 0.6 nA. 0.8 nA. 1.0 nA. 1.2 nA. 1.4 nA. 1.6 nA. 1.8 nA. 2.0 nA. and 2.2 nA.

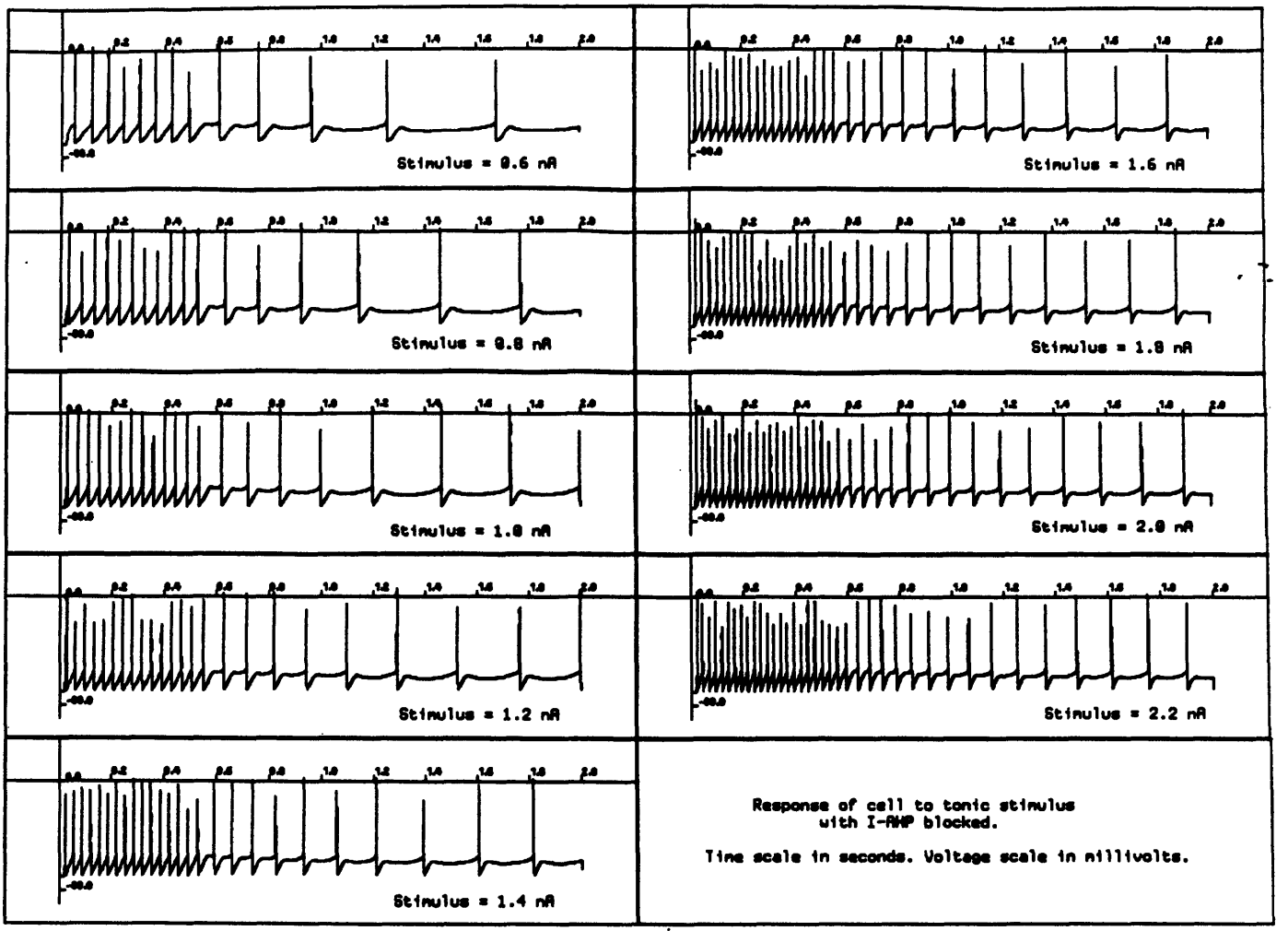


Figure 9.7: Simulations of response to tonic somatic input with I_{AHP} disabled. Stimulus strengths include 0.6 nA. 0.8 nA. 1.0 nA. 1.2 nA. 1.4 nA. 1.6 nA. 1.8 nA. 2.0 nA. and 2.2 nA.

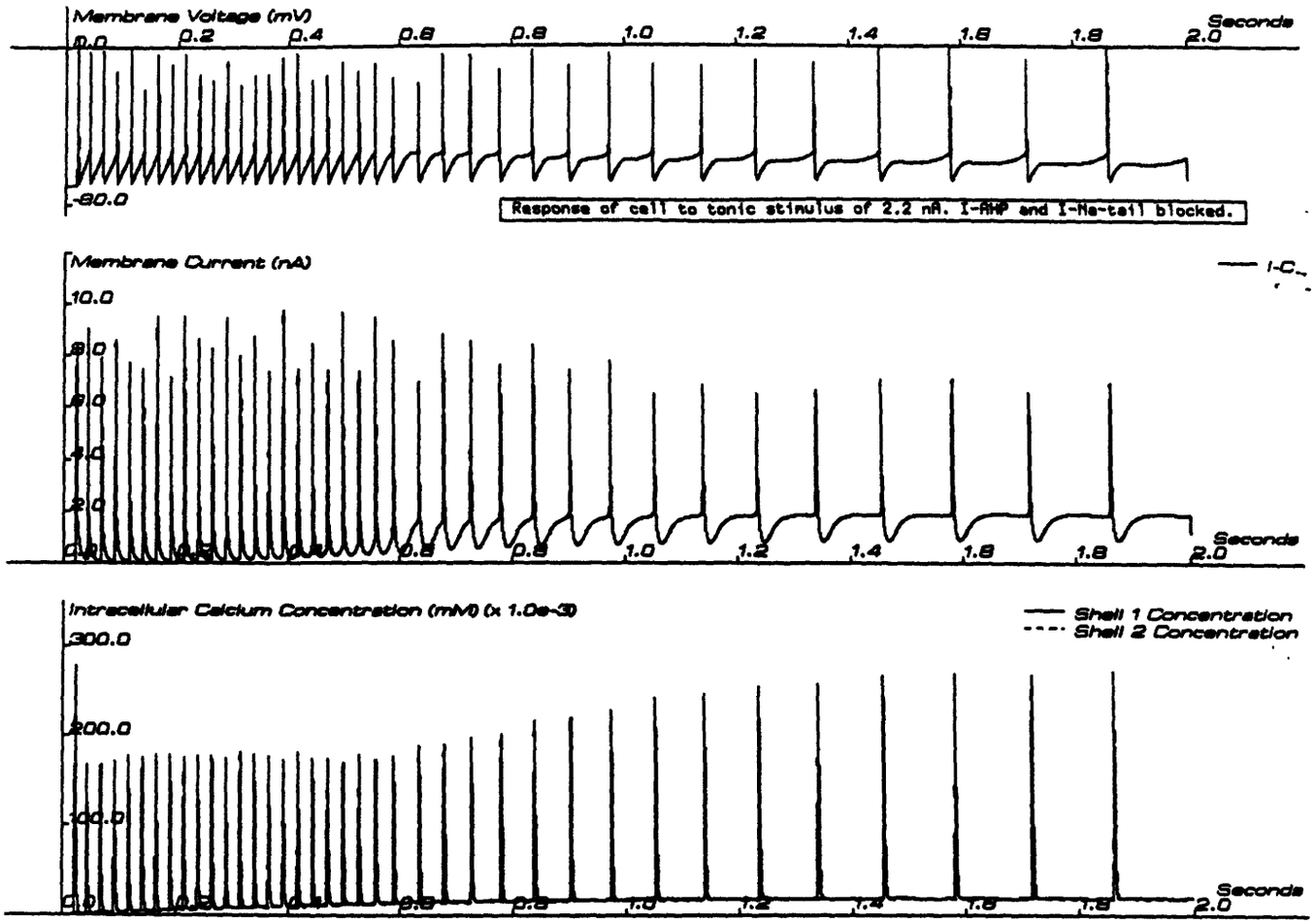


Figure 9.8: Role of I_C during spike train in response to 2.2 nA stimulus with I_{AHP} blocked. Top - Soma response. Middle - I_C during train. Bottom - $[Ca^{2+}]_{shell.1}$ and $[Ca^{2+}]_{shell.2}$ during train. The buildup of $[Ca^{2+}]_{shell.2}$ as a result of the sustained high rate of repetitive firing, causes a rise in $[Ca^{2+}]_{shell.1}$ between spikes and subsequent persistent activation of I_C.

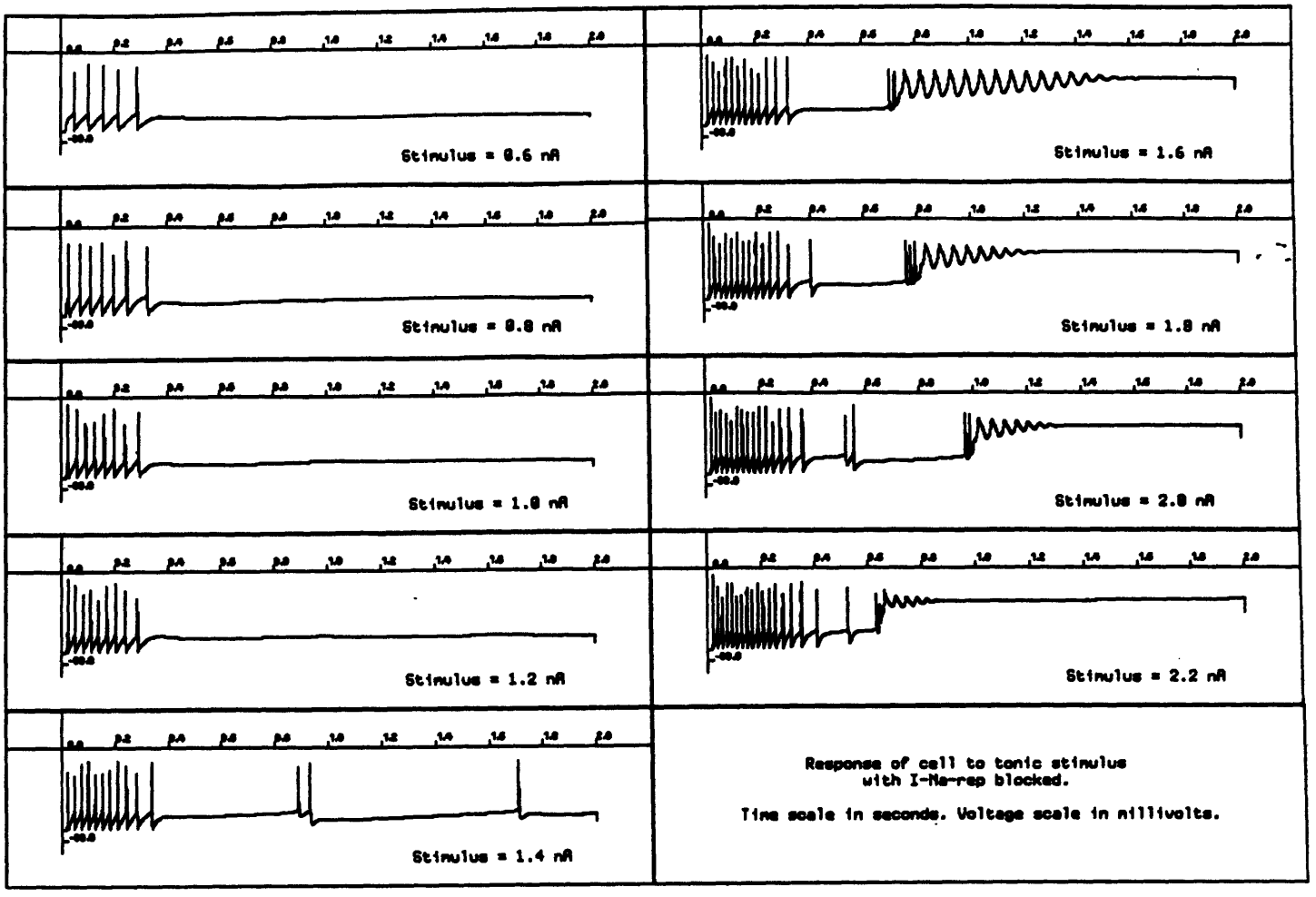


Figure 9.9: Simulations of response to tonic somatic input with I_{Na-rep} disabled. Stimulus strengths include 0.6 nA. 0.8 nA. 1.0 nA. 1.2 nA. 1.4 nA. 1.6 nA. 1.8 nA. 2.0 nA, and 2.2 nA.

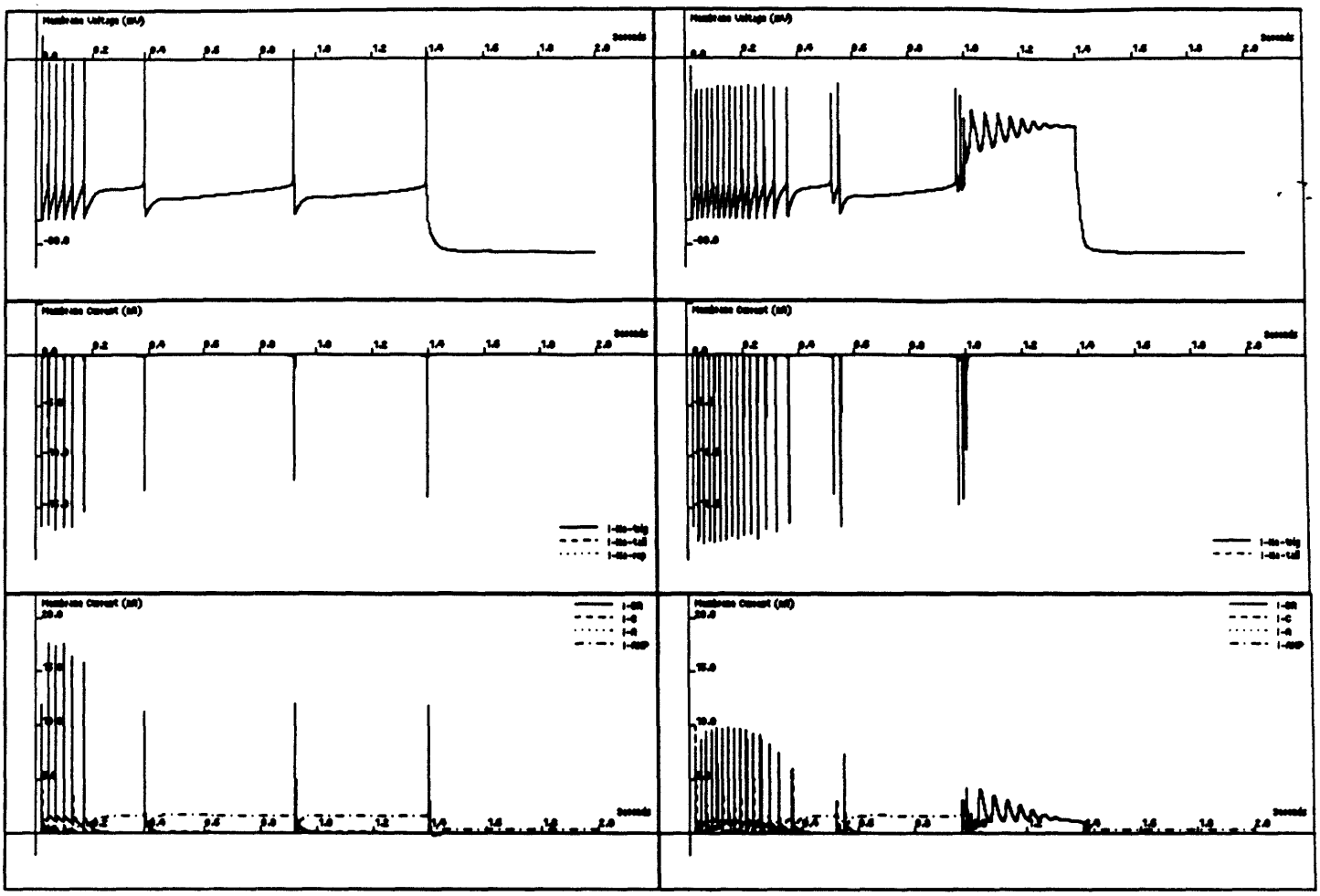


Figure 9.10: Comparison of spike train with and without I_{Na-rep} - Soma response and Na^+ and K^+ currents.

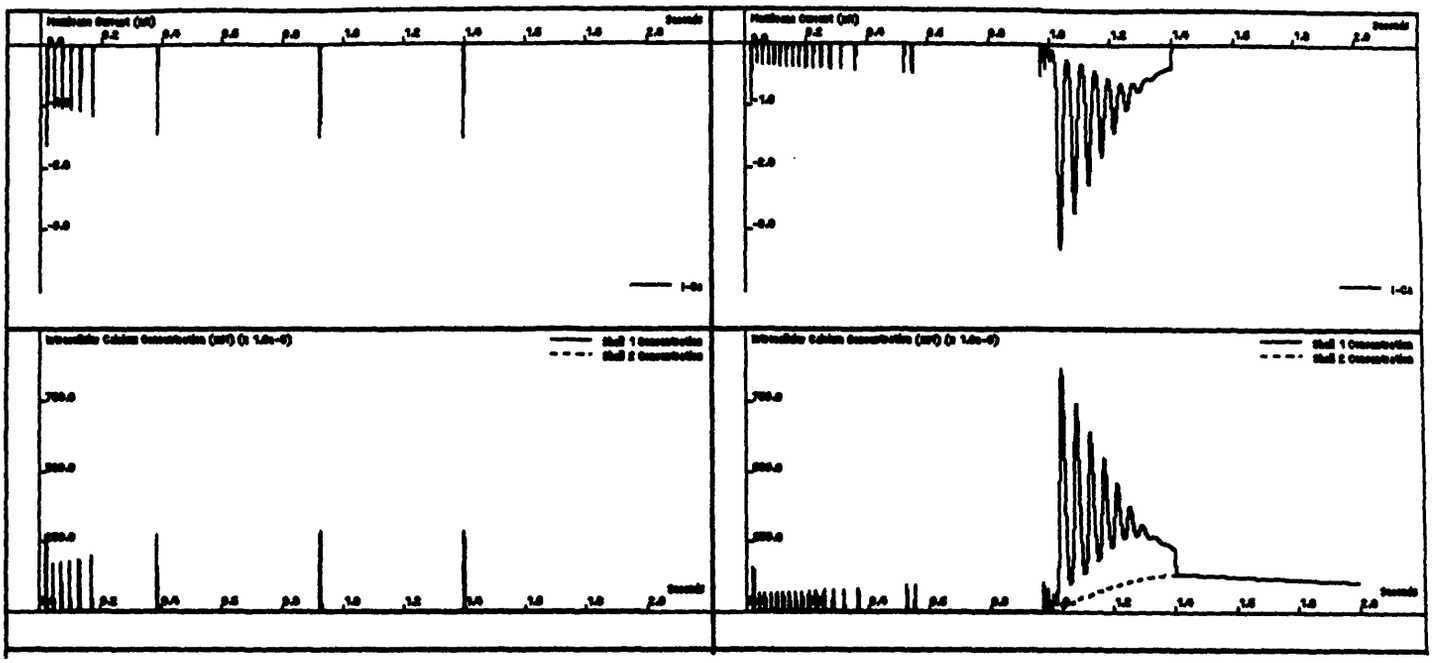


Figure 9.11: Comparison of a spike train with and without $I_{Na-rep} - I_{Ca}$ and intracellular Ca^{2+} concentrations.

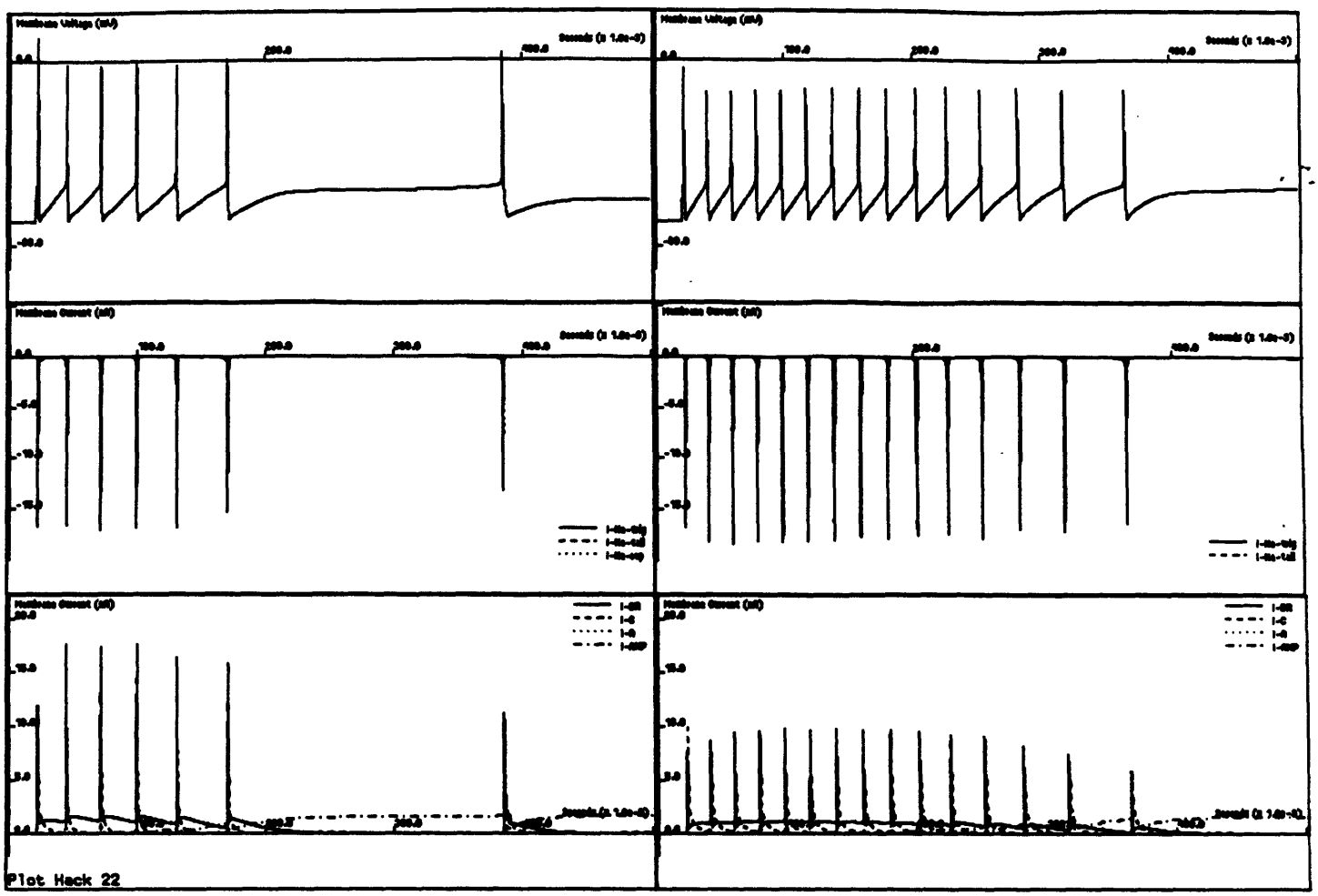


Figure 9.12: Comparison of beginning of spike train with and without I_{Na-rep} – Soma response and Na^+ and K^+ currents.

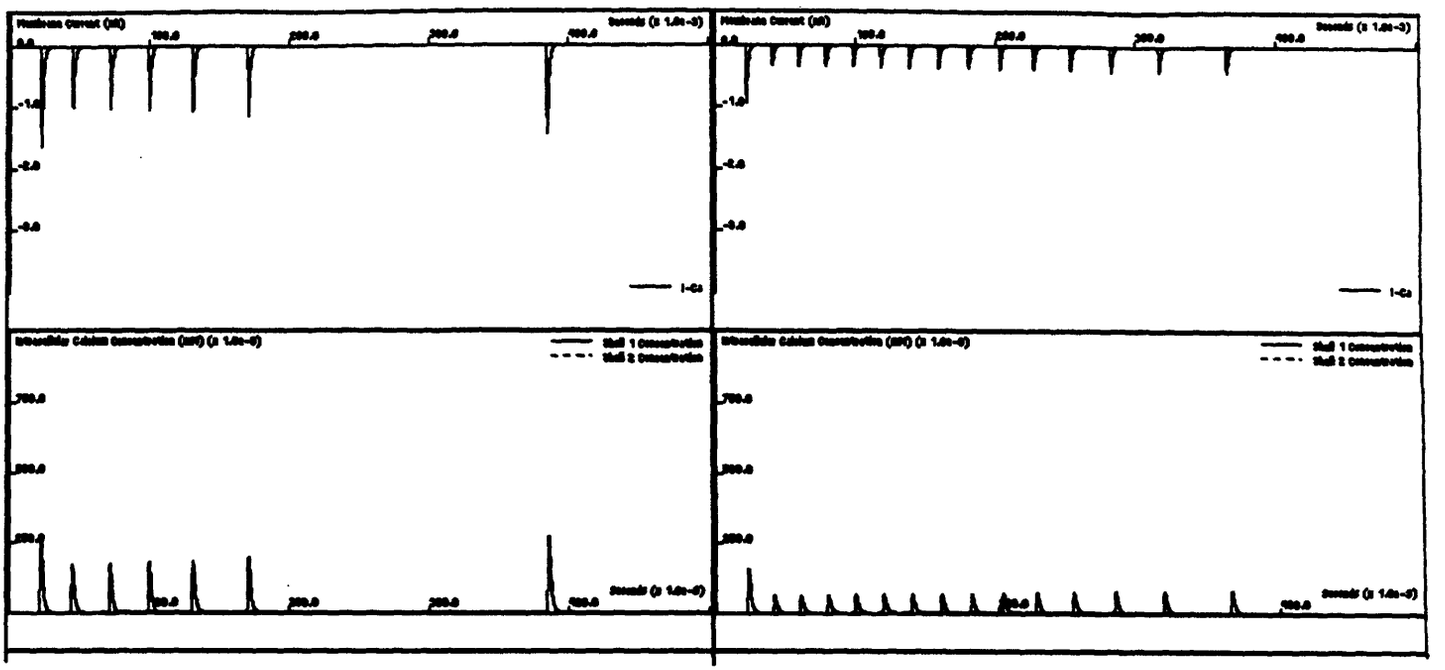


Figure 9.13: Comparison of beginning of spike train with and without $I_{Na-rep} - I_{Ca}$ and intracellular Ca^{2+} concentrations.

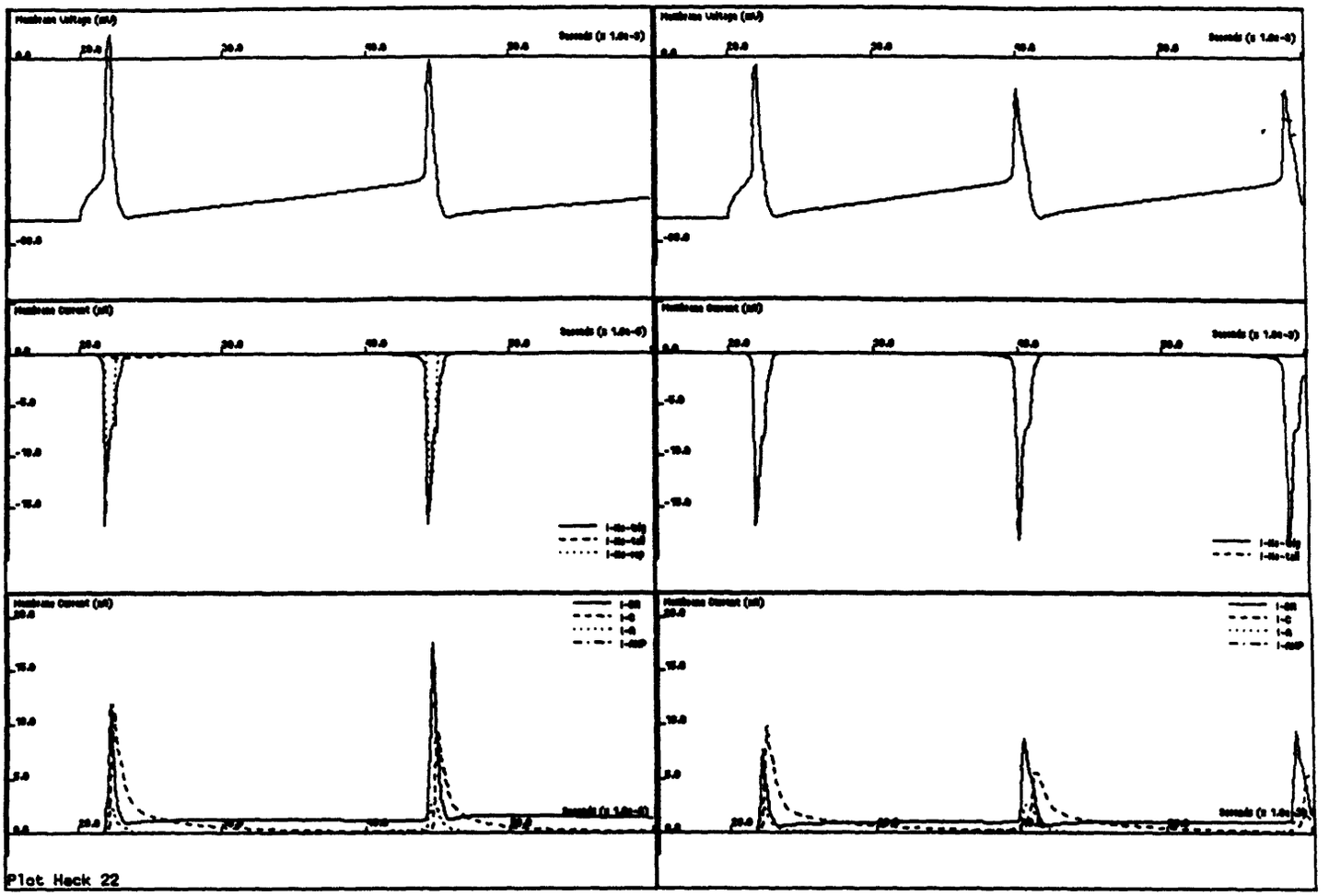


Figure 9.14: Comparison of initial spikes of spike train with and without I_{Na-rep} - Soma response and Na^+ and K^+ currents.

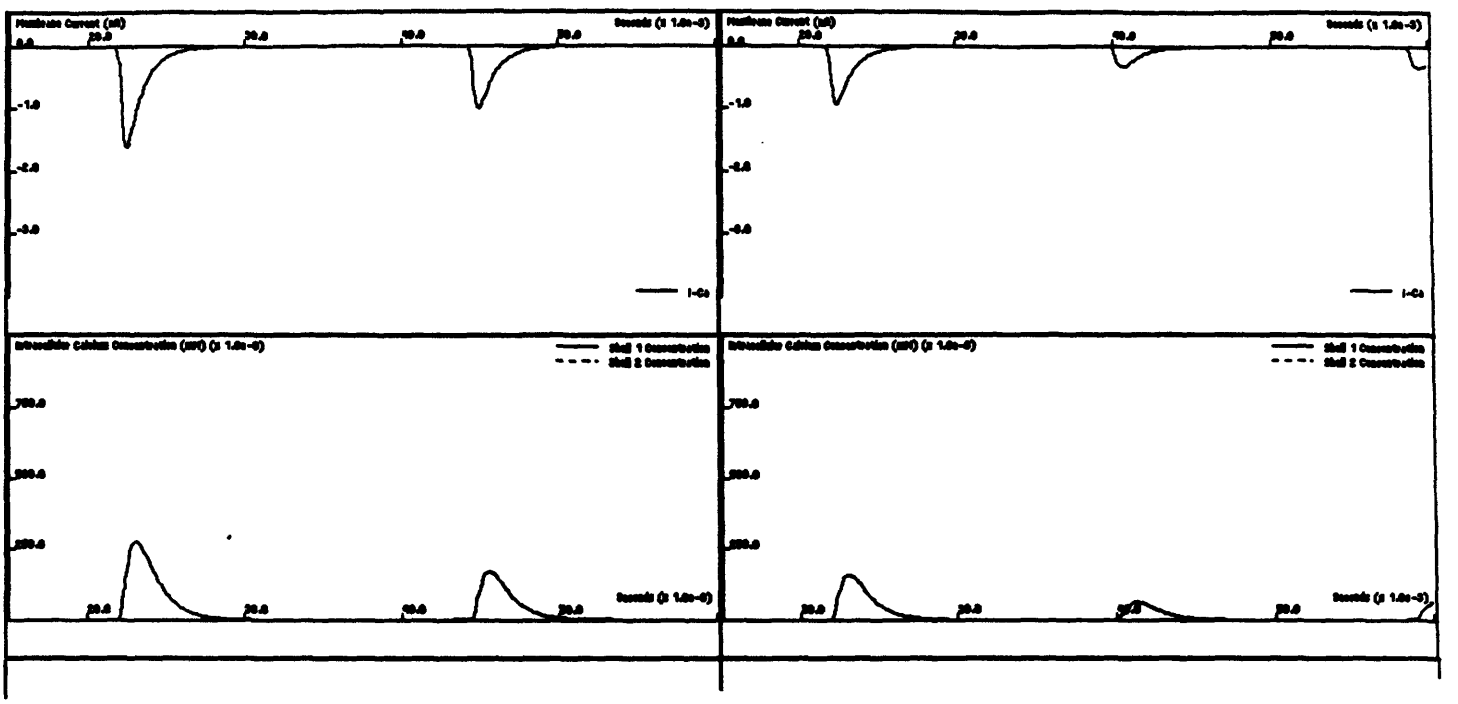


Figure 9.15: Comparison of initial spikes of spike train with and without $I_{Na-rep} - I_{Ca}$ and intracellular Ca^{2+} concentrations.

the cell reaches a stable, depolarized level until the stimulus is removed.

9.3 Conduction of Dendritic Input to Soma

Figure 9.17 illustrates a current clamp simulation in which a series of four 1.3 nA, 3 millisecond current pulses was injected into the distal dendrite. These pulses approximate excitatory synaptic input, assuming that an excitatory synapse consists of a selective conductance with a reversal potential around -25 mV, a total conductance on the order of 60 nS, and an activation period of 3 milliseconds. As seen in the figure, these events propagate down the dendrite and sum at the soma until spike threshold is reached.

An interesting detail from this simulation is the shape of the repolarization of this spike. Recall that single spikes evoked by *somatic* stimuli display a distinct fAHP (ref. Figure 7.18). In Figure 9.17 the bottom of the fAHP is elevated so that there no longer is a short depolarizing phase prior to the ADP. This is caused by the increased soma-dendrite gradient, which in turn results from the fact that the original depolarization is due to the dendrites rather than from soma input [1].

As mentioned in Chapter 5 a characteristic of $I_{Na-trig}$ is its sharp threshold. This characteristic is demonstrated in Figure 9.18. Here simulations with two inputs are illustrated. One input consists of the previous pulse train. The second input is identical except that the interval between pulses has been increased by 1 millisecond. This input evokes essentially a linear response (compare with the step responses in Chapter 3), demonstrating the fine tuning of $I_{Na-trig}$.

9.4 Demonstration of the Full Output of the HIPPO Simulations

I shall now present a typical simulation protocol in order to show the collection of variables that underly the behavior of the model. Figure 9.19 shows the overall response of the model to the soma stimulus shown in the lower part of the figure. An initial hyperpolarizing current step is applied

[1] In the earlier simulation of the single action potential the repolarization of the spike also caused the dendrite to momentarily be at a higher potential than the soma due to the charge stored in the dendritic capacitance, which in turn contributed to the ADP, but here this potential difference is greater as explained above.

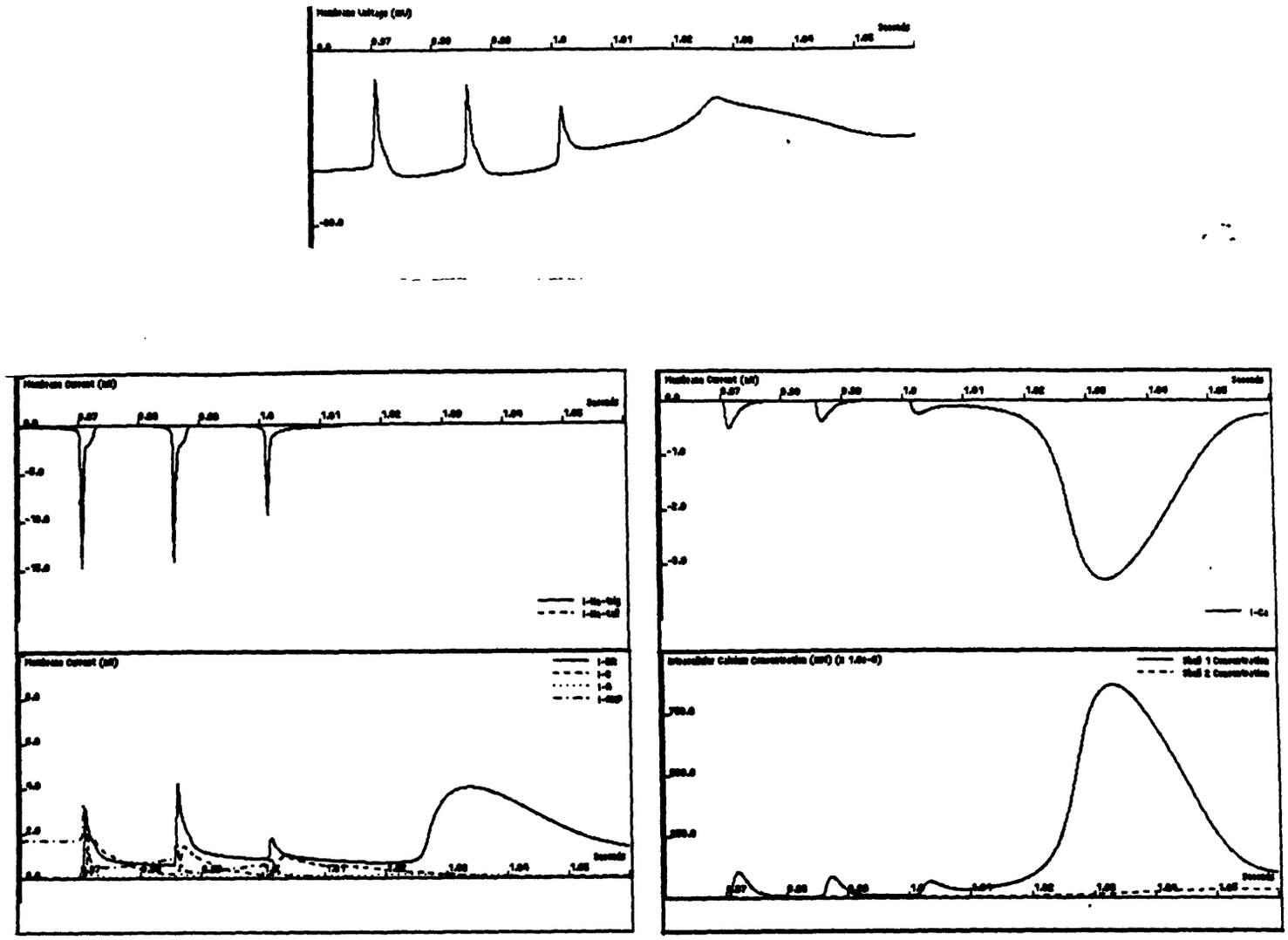


Figure 9.16: Degeneration of spike train without I_{Na-rep} – Soma response. Na^+ currents. K^+ currents. I_{Ca} , and intracellular Ca^{2+} concentrations. Note the eventual strong activation of I_{Ca} preceding stable depolarization.

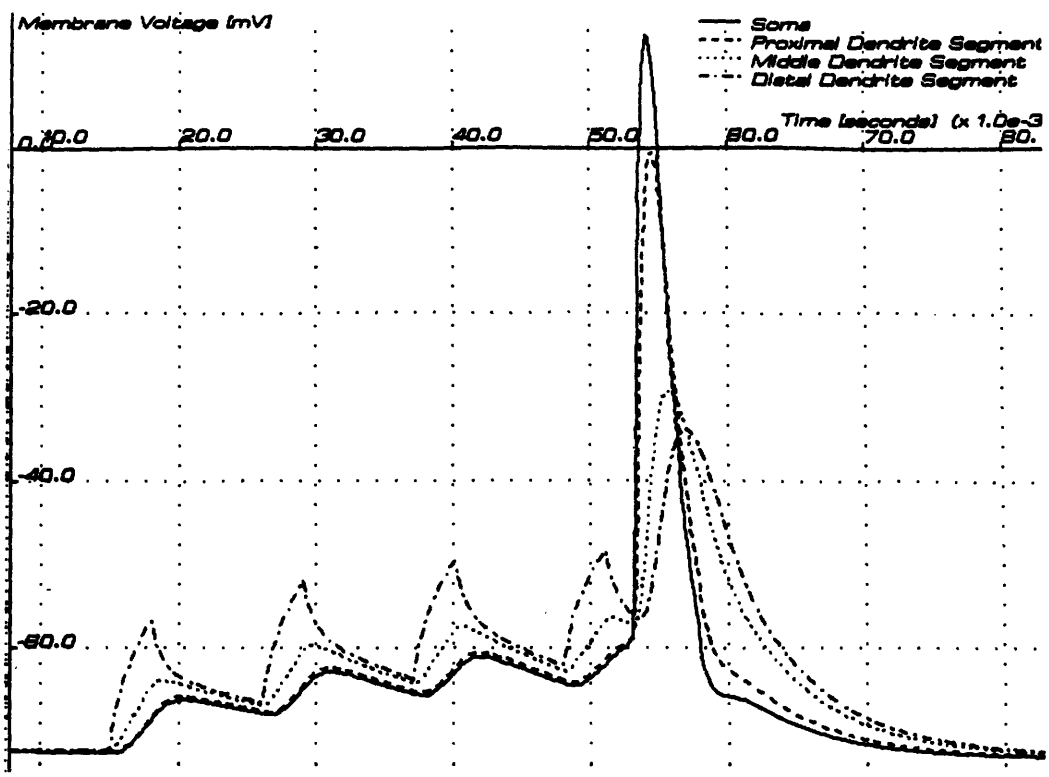


Figure 9.17: Propagation of spike-producing dendritic input down cable to soma. Current pulse train injected into distal dendritic segment (ref. Figure 9.18 Stimulus 1 (top)).

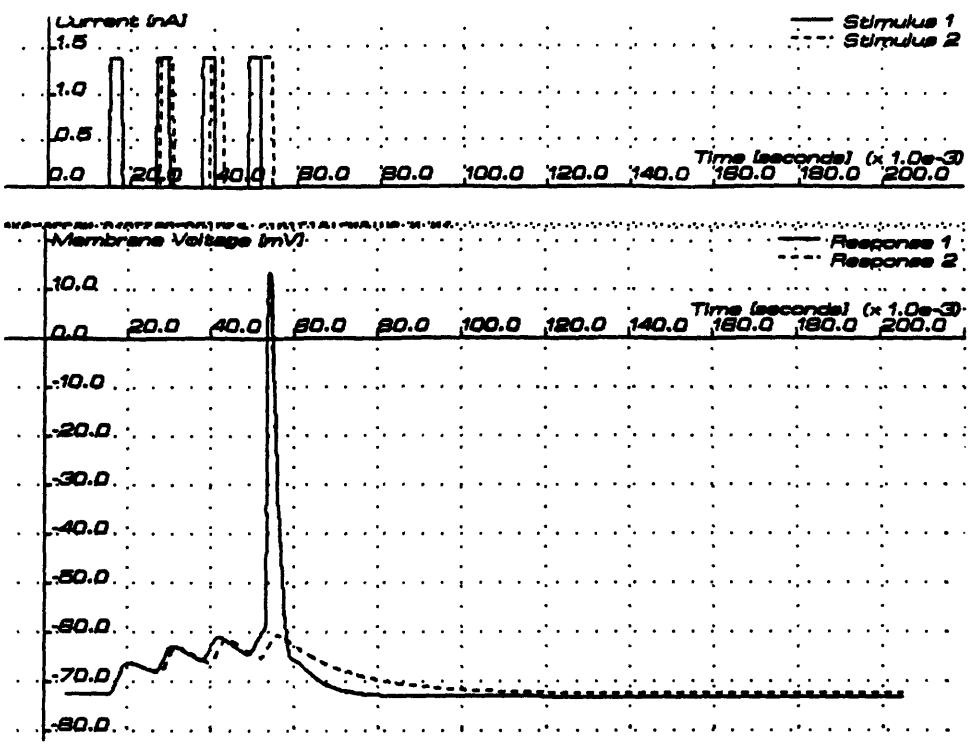


Figure 9.18: Superimposed response to the stimulus used in the previous figure (Stimulus 1, Response 1) and a subthreshold stimulus (Stimulus 2, Response 2). Note the near linear response of the subthreshold stimulus. Both voltage traces are the soma response (bottom). Current pulse trains injected into the distal dendritic segment (top).

to the soma, bringing the soma voltage down from its resting potential of about -72 mV to about -83 mV. Next, a 220 millisecond 1.8nA depolarizing current step is applied, resulting in the characteristic burst of action potentials, whose frequency just begins to reduce as the action of I_{AHP} starts. After the stimulus, the beginning of the long-lasting after-hyperpolarization is seen.

In Figures 9.20, 9.21 9.22 the Na^+ currents and their associated gating variables during the response of Figure 9.19 are illustrated (note change of time scale).

In Figures 9.23, and 9.24 I_{Ca} its gating variables and the time course of the intracellular Ca^{2+} concentrations are shown during the response of Figure 9.19 are illustrated (note change of time scale).

In Figures 9.25, 9.26, 9.27, 9.28, and 9.29 the K^+ currents and their associated gating variables during the response of Figure 9.19 are illustrated (note change of time scale). The relationship between the activation of the Ca^{2+}-mediated gating variables (w) for I_C and I_{AHP} and the time courses for $[Ca^{2+}]_{shell.1}$ and $[Ca^{2+}]_{shell.2}$ shown in Figure 9.24 is clear. The conductance underlying I_M remains relatively constant (the time course of I_M closely matches the time course of the voltage), and therefore while this current is almost as large as I_{AHP}, it has the relatively uncolorful role of mediating repetitive firing only slightly by changing the effective (linear) impedance of the cell.

Finally, in Figures 9.30 and 9.31 the linear components of the somatic response are shown, i.e. the capacitive and leak soma currents and the dendritic voltages, respectively.

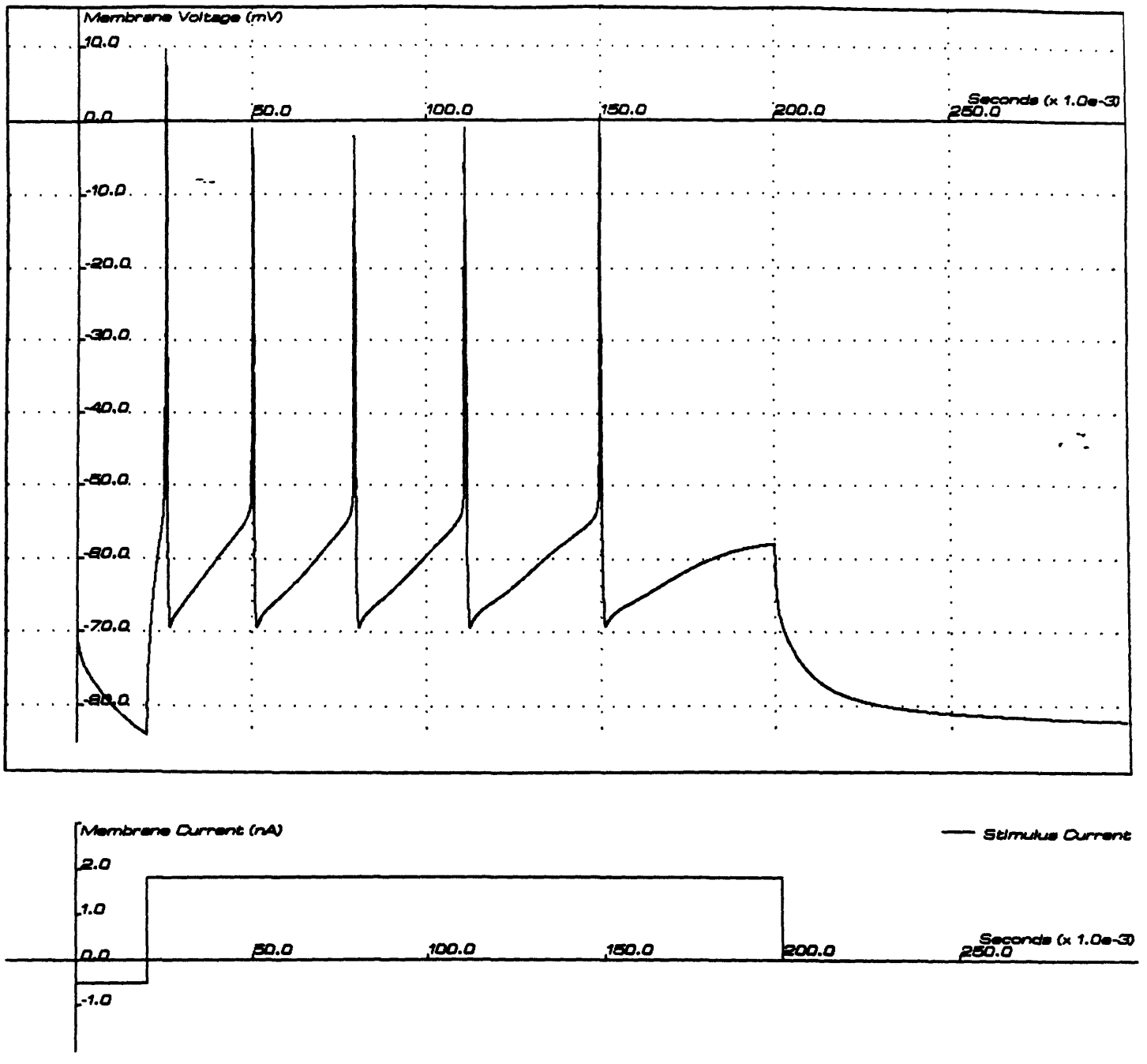


Figure 9.19: Typical simulated response of the cell (top) to short hyperpolarizing current stimulus followed by longer depolarizing step (bottom).

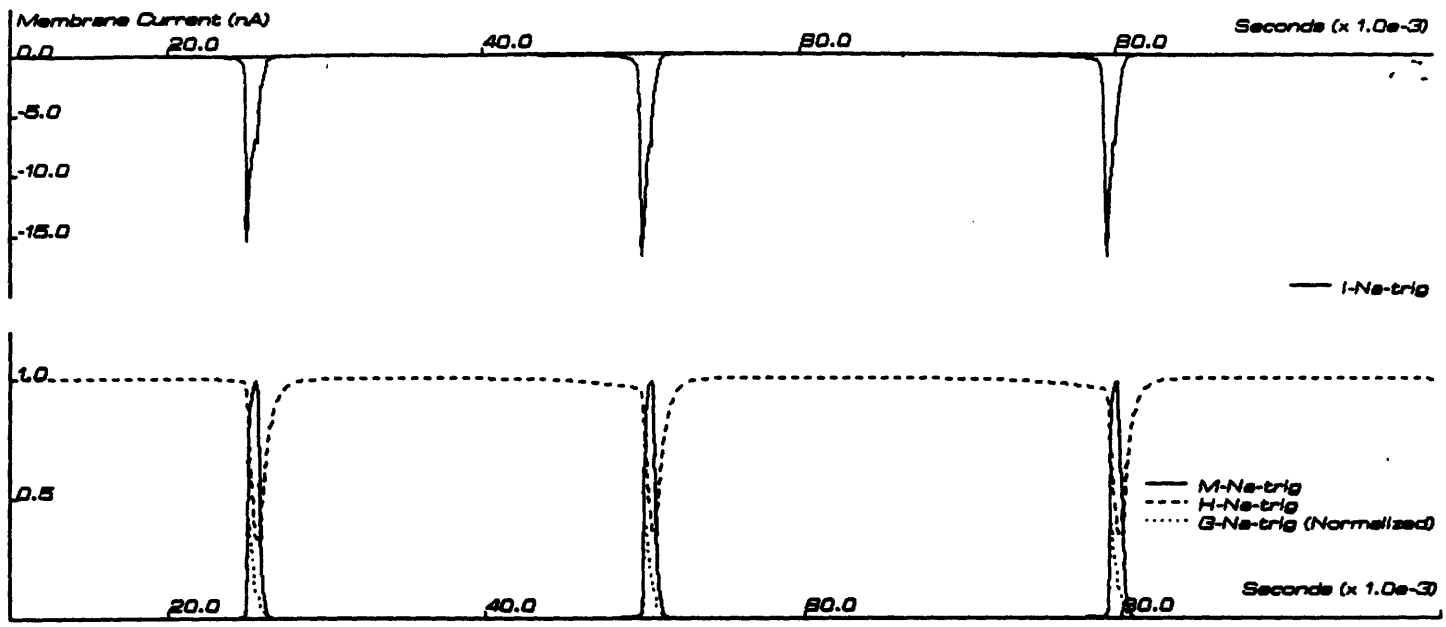


Figure 9.20: $I_{Na-trig}$ and its gating variables during response shown in Figure 9.19.

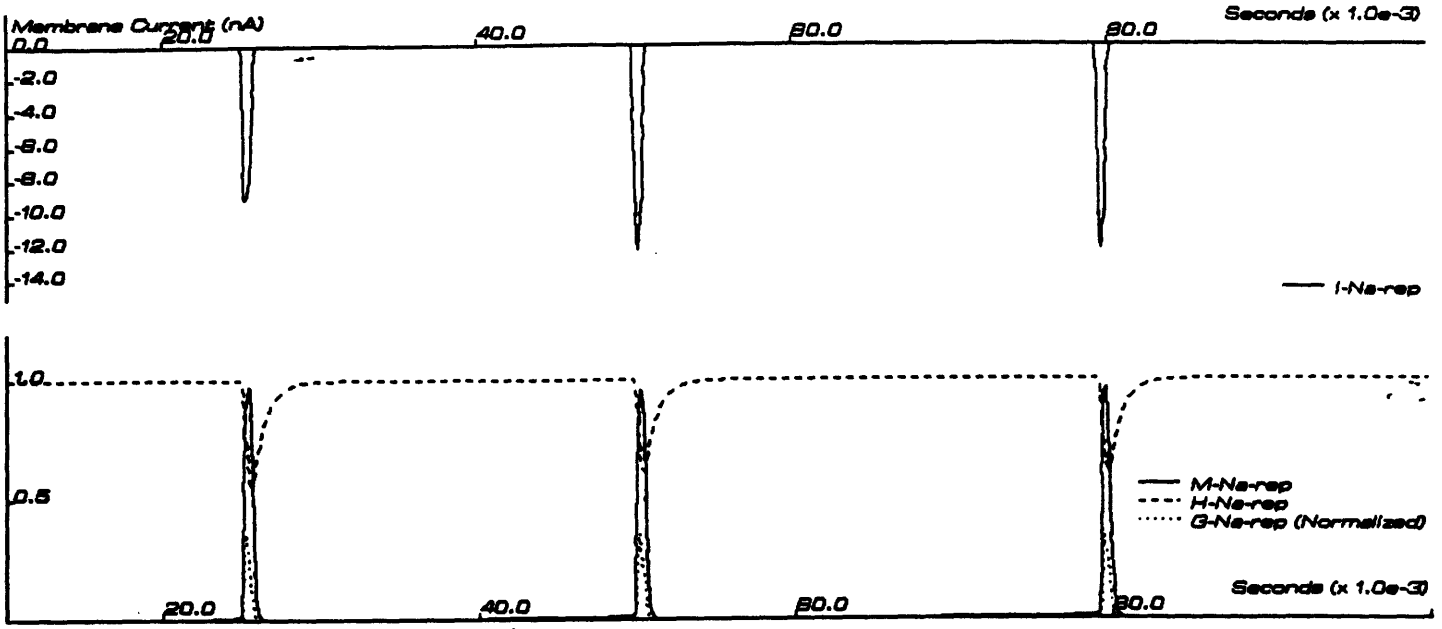


Figure 9.21: I_{Na-rep} and its gating variables during response shown in Figure 9.19.

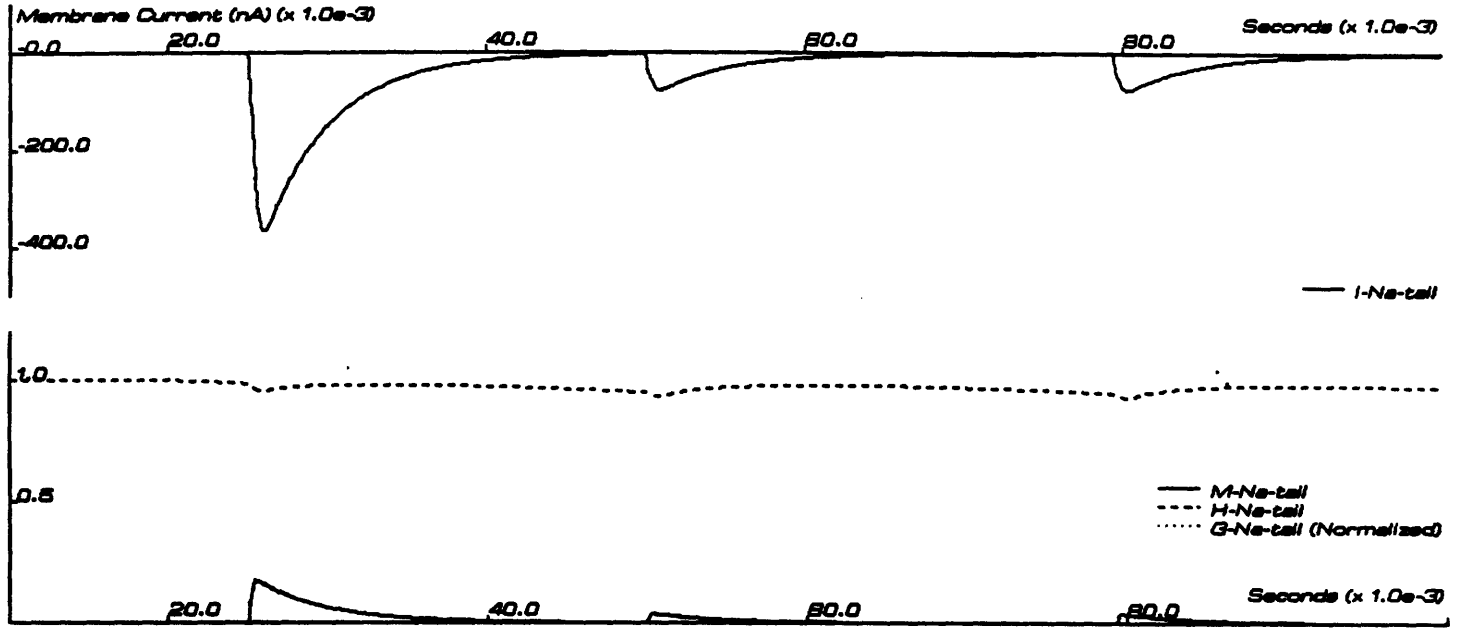


Figure 9.22: $I_{Na-tail}$ and its gating variables during response shown in Figure 9.19.

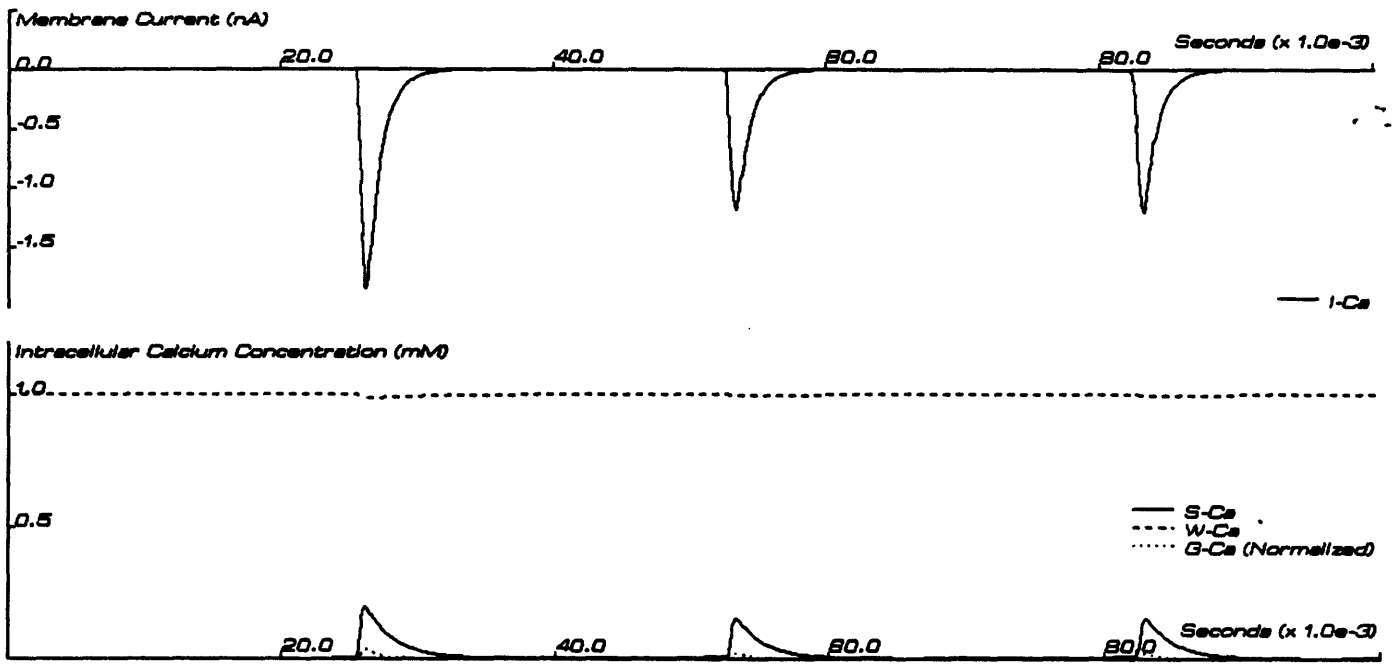


Figure 9.23: I_{Ca} and its gating variables during response shown in Figure 9.19.

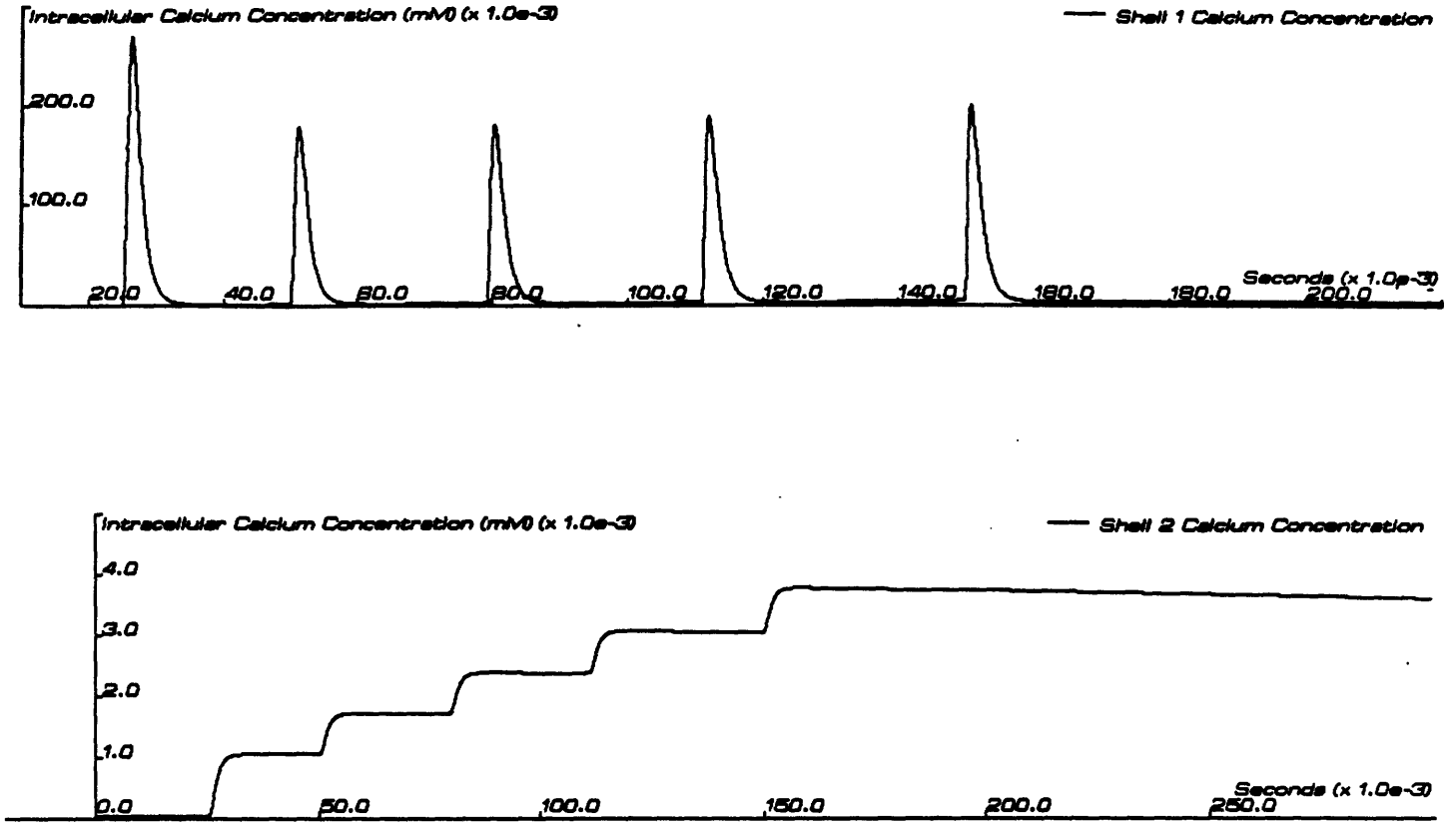


Figure 9.24: $[Ca^{2+}]_{shell.1}$ and $[Ca^{2+}]_{shell.2}$ during response shown in Figure 9.19.

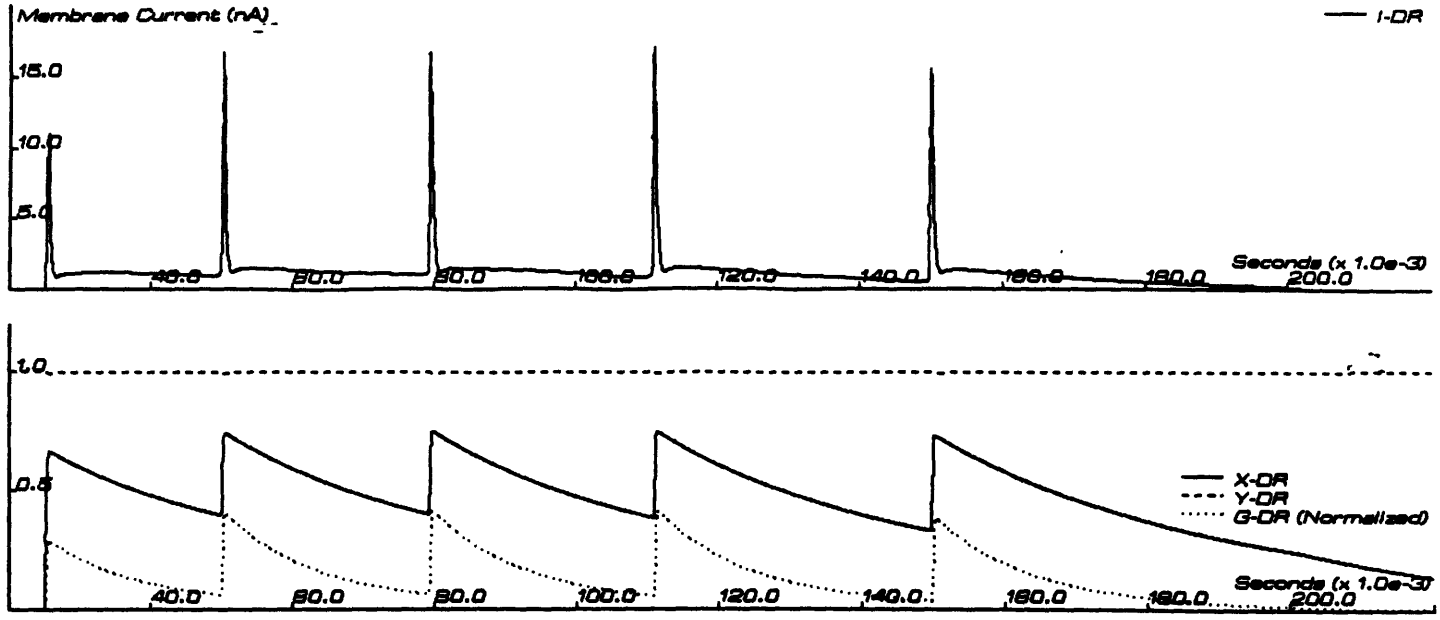


Figure 9.25: I_{DR} and its gating variables during response shown in Figure 9.19.

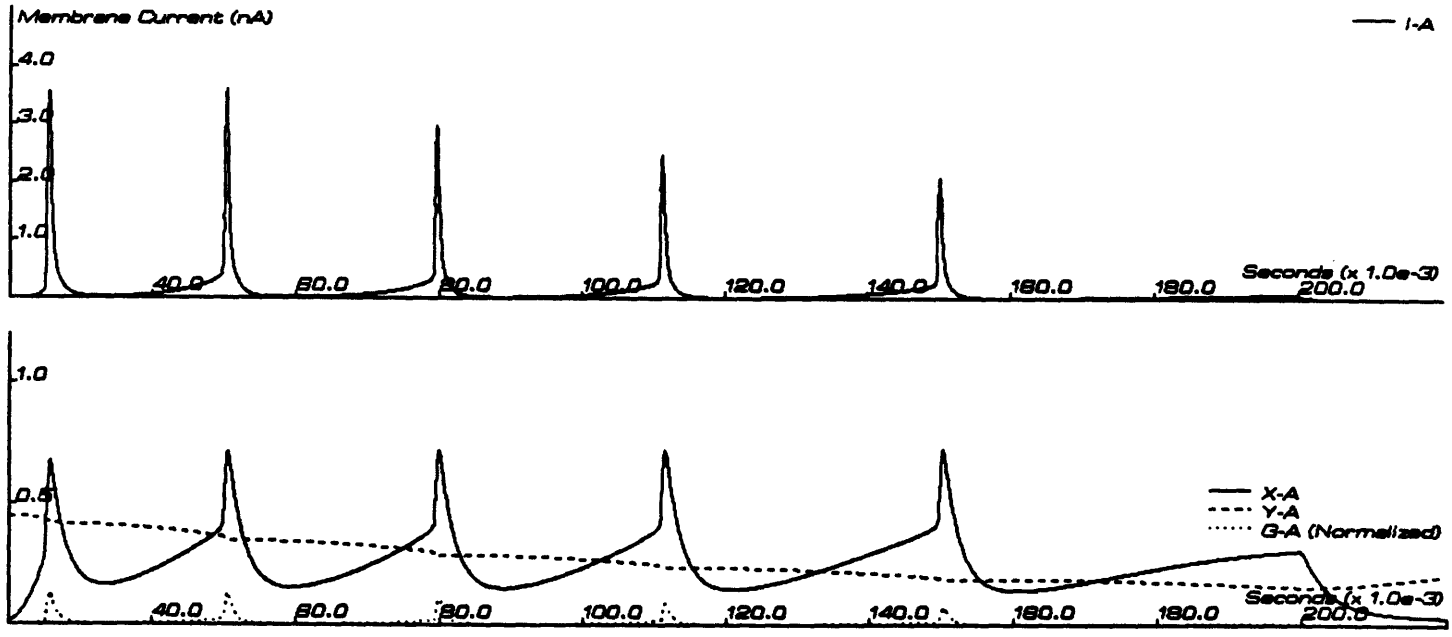


Figure 9.26: I_A and its gating variables during response shown in Figure 9.19.

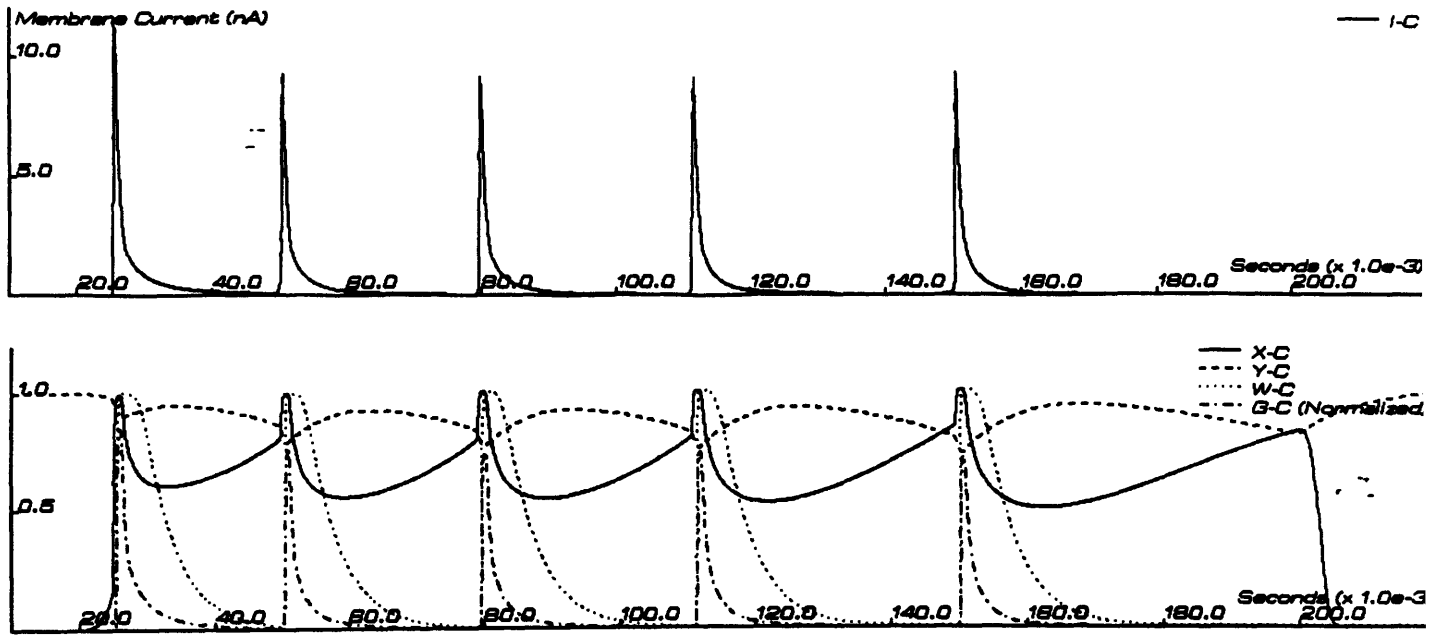


Figure 9.27: I_C and its gating variables during response shown in Figure 9.19.

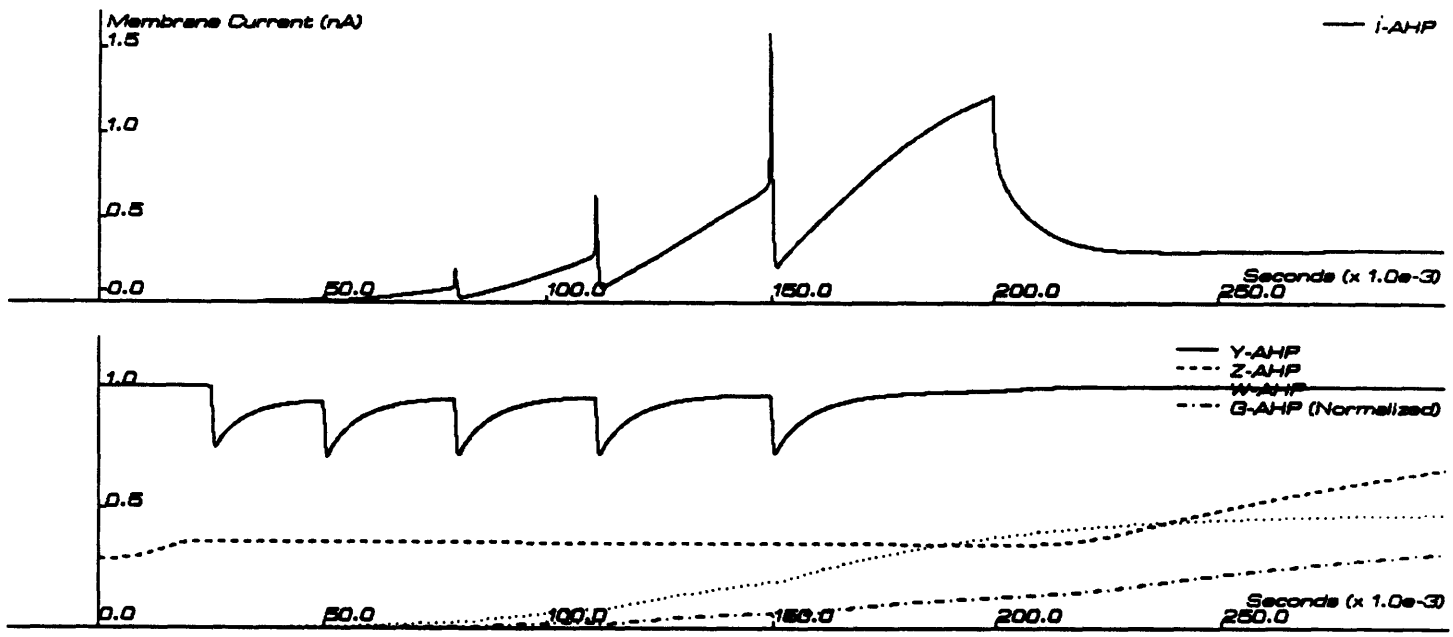


Figure 9.28: I_{AHP} and its gating variables during response shown in Figure 9.19.

Figure 9.29: I_M during response shown in Figure 9.19.

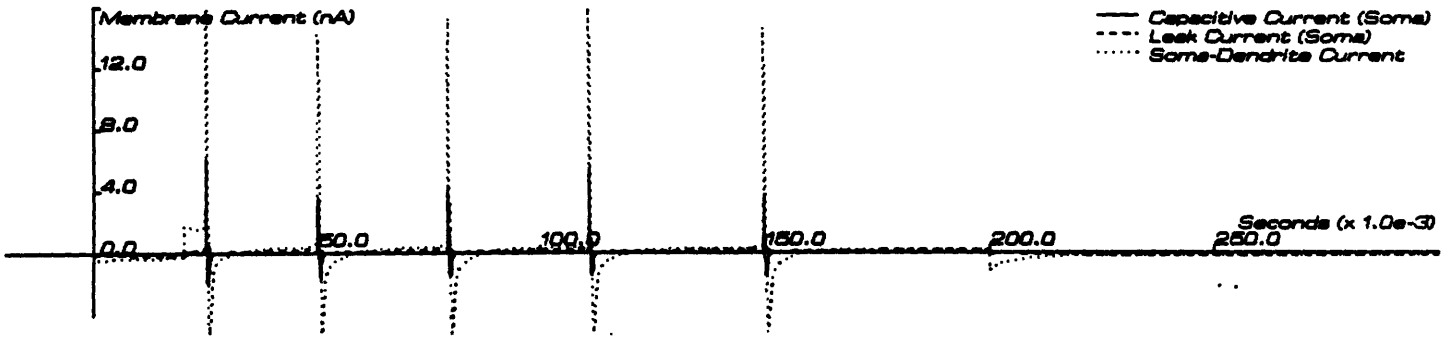


Figure 9.30: The soma capacitive and leak currents during response shown in Figure 9.19.

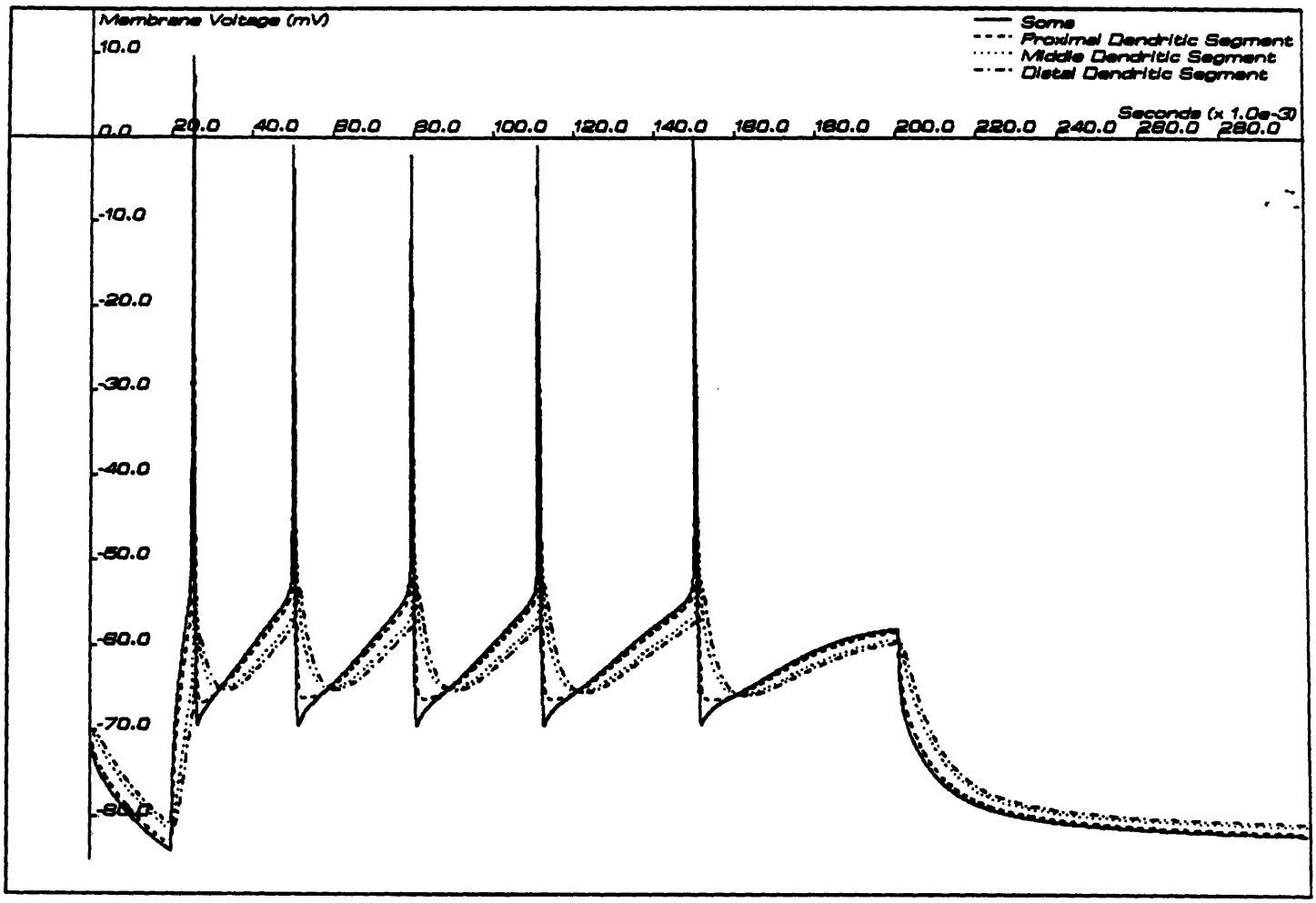


Figure 9.31: The dendritic voltages during response shown in Figure 9.19.

Chapter 10

DISCUSSION

10.1 Introduction

This chapter will address some of the more general implications and conclusions derived from the model, in particular those issues not covered in earlier chapters. The overall question remains that if neural nets are realizable with elements that just exhibit integrative all-or-nothing responses that are connected with regenerative conductors, then why are all the channels needed? The results of the model suggest some rationale as well as some specific questions addressed at the apparent role of many of the currents described.

10.2 Physiological Roles of Specific Currents in Information Processing

How can the different currents described here contribute to the information processing capability of the pyramidal cell? The first step in answering this question is primarily mechanical, that is we need to show how a given current shapes the response to a repertoire of inputs. At this stage, the repertoire considered has been very basic - short depolarizing current steps that evoke single spikes, long lasting depolarizing current steps that evoke spike trains, and (to a lesser degree) simple dendritic input consisting of depolarizing current steps applied to the distal portion of the dendritic cable. By examining the response to these inputs the functional roles of the model currents can be grouped into three (non-exclusive) categories:

Current	Spike Shape	Threshold	Freq-Inten
$I_{Na-trig}$	+	+++	
I_{Na-rep}	+	++	+++
$I_{Na-tail}$			+
I_{Ca}	(++)	(+)	+ (+++)
I_{CaS}	?	?	?
I_{DR}	++	+	++
I_A	+	++	++
I_C	+		+++
I_{AHP}	-	++	+++
I_M	-	+	+
I_Q	?	?	?

Table 10.1: Functional roles of hippocampal somatic currents. Entries in parentheses indicate secondary role, e.g. Ca^{2+} activation of K^+ current. "?" means that the role is unknown.

1. Modulation of shape of single action potential (Spike Shape).
2. Modulation of firing threshold, both for single and repetitive spikes (Threshold).
3. Modulation of repetitive firing, specifically the relationship between strength of tonic input and frequency of initial burst and later "steady state" spike train (Freq-Inten).

Table 10.1 summarizes the main roles for each of the described currents as indicated by the simulations.

10.2.1 Possible Roles for the Modulation of the FI Characteristic

Traditionally neural information is assumed to be encoded by frequency modulation (specifically PCM), that is the number of spikes per second encompasses the message of a neuron. For example, the strength of contraction for a muscle fiber is, over some range, a linear function of the spike frequency of its efferent neuron. If action potential propagation is assumed to be a stereotyped phenomena, then clearly the only way to modulate neuronal output is by changing the spike frequency. If the inhibition of a

specific current changes the FI characteristic, this allows the modulation of that neuron's information processing by various agents.

10.2.2 Possible Roles for the Modulation of the Threshold of the Somatic Action Potential

The setting of the somatic threshold will determine the minimal input for eliciting a spike, and in effect change the sensitivity of a cell. For example, if $I_{Na-trig}$ was blocked by some endogenous agent, then the firing threshold for that cell will be raised by about 10 millivolts. This would cause the cell to ignore a wide variety of input patterns that would otherwise generate soma spikes. Even subtle changes in soma threshold, as might for example be mediated by selective inhibition of I_M, could significantly alter the overall transfer function of a local population of neurons, assuming that the cholinergic input is spread out over that population and not just directed at a single cell ([24], [25],[35], [36]).

There are actually two aspects of the "threshold" for a cell - static and dynamic. In other words the rate at which the soma membrane approaches threshold is as important as the absolute level of that threshold. In general the threshold rises with a slower approach because there is a small range for which sub-threshold activation of $I_{Na-trig}$ is possible. The most striking evidence for this was demonstrated in Chapter 7, where the role of I_A in delaying spike initiation by (effectively) slow stimulus was shown. I_A therefore may help to distinguish tonic dendritic (particularly distal) input versus tonic somatic input. For input that eventually will supply the same depolarizing current at the soma, dendritic input will have a slower onset due to the cable properties. This slow onset could allow I_A to transiently delay the onset of the spike or spikes, as was shown. A similar depolarizing current of somatic origin (e.g. somatic synapses) would have a faster onset such that I_A would not be activated in time on the depolarizing phase to delay the spike. Extending this possibility further, blocking I_A could have the physiological or pathological result of reducing the ability of the soma to discriminate proximal versus distal inputs.

10.2.3 Possible Roles for the Modulation of the Shape of the Somatic Action Potential

How important is the *shape* of an individual spike at the soma? In general this question has not been addressed in the literature, but we can speculate

on the possibilities. First, we can assume that spike shape, in particular spike width, is unimportant to a first degree at the soma – once the soma fires, it fires. However, the role of the spike beyond the soma may or may not depend on the spike shape, and this possibility is dependent on to what extent spike propagation is a linear or non-linear phenomena.

This in turn will determine the degree to which an axonal termination "see's" the actual time course of the somatic event. At one extreme, the proximal axon could transmit the spike a purely non-linear fashion – once threshold was reached, the classic "all-or-nothing" response would transmit a stereotyped action potential down the axon whose shaped would be completely independent of the (immediate) post-threshold behavior at the soma. At the other extreme, i.e. if the axonal membrane were purely linear, the propagation of the somatic event at any point down the axon would be a convolution of the entire somatic signal, rather than just a function of when the soma potential passed some threshold.

The situation in the brain probably lies somewhere between these limits, that is electrical activity at the axon terminal is somewhat dependent on the shape of the somatic spike. The extent to which this is true will in turn be dependent on the wavelength of the propagated spike. For example, consider a typical un-myelinated axon of an HPC with a diameter of 1 micron and a conduction velocity on the order of 10 meters/second. For this axon a 1 millisecond action potential will have a wavelength on the order of 10 millimeters. Since the distance between the soma and an axon terminal may fall in this range, the post-threshold waveform at the soma may influence the pre-synaptic waveform, despite the non-linearity of the axon.

Consider what happens if the axon is myelinated. Myelination means that its C_m will be much less and its R_m will be much greater. This results in (a) the conduction velocity increasing (which increases the wavelength proportionally) and (b) a reduction of the attenuation of the somatic signal as it travels down the axon, in particular the high-frequency components of the signal. In sum, if the HPC axon is myelinated, the electrical activity at its terminals will even more likely depend on the time course of the somatic waveform, despite the excitable membrane at (in particular) the axon's nodes of Ranvier.

So, given the possibility that the shape of the somatic action potential may modulate the signal at the pre-synaptic terminal, what role could this serve? There are at least two possibilities. First, it has been demonstrated that the release of transmitter at the pre-synaptic terminal is not an all-or-nothing event, that is the amount of transmitter released is a function of the

time course of the terminal spike ( ). For example, modulation of the somatic spike width may in turn determine how much transmitter is released down the line, thereby allowing a mechanism for changing the effective strength of the spike as seen by the distal neuron. Second, pyramidal cell axons often project collaterals back to the originating cell, forming axo-somatic synapses, resulting in a feedback loop. In this case, modulation of the somatic spike could affect this feedback in complicated ways, particularly since the length of the collaterals is not large.

There may also be a role for the somatic spike shape during the transmission of an action potential at axonal branch points. For example, consider a axonal branch point with an impedance mismatch and where there is one thin and one thick proximal branch. In this case an orthodromic spike that is too narrow may not be able to depolarize the thick branch sufficiently for transmission of the spike down that branch, and as a result the spike would propagate only down the thin branch. If this is possible, then modulation of the somatic spike shape could be used to direct the cell's output in a time-varying way, i.e. some times allowing blanket transmission to all the cell axon's destinations, and at other times allowing reception of that output by only a limited set of the proximal neurons.

To summarize, encoding information as spike frequency is clearly part of the story, but it may not be the whole story. Modulation of somatic spike width could be equivalent to a modulation of the "loudness" of a given neuron's message. As mentioned previously, considering that some of the currents may be modulated by non-cell-specific factors (e.g. local, non-synaptic release of cholinergic agonists), the "message" being turned up or down may be one being broadcast from a local population of cells, not just a single cell.

In order to further examine the above scenarios, it will be necessary to investigate the relationship between somatic spike shape and pre-synaptic potential, in particular the effect of axon length, diameter, etc. We have to answer the question of whether the pre-synaptic membrane (and, more importantly, the post-synaptic membrane via modulation of transmitter release or gap-junction interactions) see what is happening at the soma? We also have to analyze at what point does axonal transmission reduce to a stereotyped all-or-nothing action potential such that the pre-synaptic response is independent of the soma potential beyond threshold.

10.2.4 Other Implications of Somatic Currents

In this thesis the *somatic* response of the HPC has been modelled, under the assumption that the dendrites present a linear load to the soma. and that the data on HPC currents reflect the activity of channels localized at the soma. The assumption of a linear dendritic tree has already been discussed (Chapter 2). However, the idea that currents measured at the soma reflect channels whose functional role is defined at the soma may be questioned as follows. Specifically, all channel proteins, regardless of there final (functional) destination are manufactured at the soma. Some of the so-called somatic channels may actually be vestiges of channels intended for axonal and/or pre-synaptic membrane. Some percentage of the channels which are manufactured at the soma for eventual export may be expressed in somatic membrane either on their way to final destination or when they are transported back to the soma for recycling. For example, it has been demonstrated that application of 4-AP modulates post-synaptic events (enhancement of EPSPs []). Does this mean that I_A, which has been tacitly assigned a primarily somatic role in this report. actually does most of its work at synaptic membrane sites on the dendrites? This question should be addressed in order to fully establish the functional role of the currents in the HPC.

However. if the spike-shaping channels are intended for pre-synaptic membrane. then modulation by endogenous factors (e.g. ACH) obviously takes place at target neuron. Now this is disadvantageous if we want factor to act selectively on some afferent tract. On the other hand, perhaps in a given dendritic field only some afferents have certain channels. so there still could be some selectivity.

10.3 Why Do the HPC Currents Span Such a Broad Kinetic Spectrum?

A related question is what could be the usefulness of several types of currents with a range of activation/inactivation characteristics for information processing function at the single cell level. Again. these currents primarily define somatic integration: the role of the dendritic tree will further complicate matters.

For example. do current kinetics serve to stabilize the cell. that is is the cell response relatively sensitive or insensitive to variations in a) chan-

nel configurations or b) channel kinetics/voltage-dependencies? The model suggests that many of the parameters have a strong effect on cell behavior. Now, the question remains as to what is the functionally important aspects of HPC response. For example, does a delay to onset of repetitive firing due to tonic dendritic input as opposed to somatic input (ref. I_A) have any functional aspect? Considering that this delay can be on the order of several hundred milliseconds, then the delay may have a very important functional role.

A crude analogy to a computer may be instructive (adapted from [37]). Cognitive processes execute on the order of hundreds of milliseconds, thus a delay of this magnitude, as demonstrated by the action of, for example, I_{AHP}, could correspond to an "instruction cycle" delay mechanism. Likewise, some currents seen to function as delay mechanisms on the order of a "machine cycle" (about tens of milliseconds), for example I_C. Along these lines, a tentative categorization of the described currents is as follows:

- I_A - can differentiate tonic dendritic input from somatic input
- I_{AHP} - can terminate initial train of repetitive firing
- I_C - just modulates spike width
- I_M - helps set threshold in general, may effect F-I
- I_{DR} - basal repolarizer
- $I_{Na-tail}$ - modulates repetitive firing
- I_{Na-rep} - allows repetitive firing with lower metabolic cost
- $I_{Na-trig}$ - basal spike current
- I_Q - ?

10.4 Pathological Roles of Specific Currents

Are specific currents mediated in isolation under certain pathologic conditions? The selective action of neurotransmitters on some of the currents, e.g. muscarine on I_M, noradrenaline on I_{AHP}, supports this possibility. Other examples include reports of various endogenous substances found *in vitro* that selectively affect distinct currents, e.g. the role of ethyl alcohol on mediation of I_A in *Aplysia* (Biophysics Abstracts, 1987).

As shown in Chapter 9, blocking the putative I_{Na-rep} has the surprising effect of causing the cell to "latch-up" in response to certain strengths of tonic stimulus that would otherwise elicit well-bounded stable spike trains. Although the existence of this current is problematical, the possibility of selective blocking for it raises the possibility of an intriguing pathology, in which neurons stimulated over a certain threshold will simply give up and remain silent until the stimulus stops. On the other hand, it just as likely that this would be a physiological response, that is under some conditions putting an upper bound, not just a lower one, on the intensity of a cell's input may be advantageous.

The relationship between intracellular Ca^{2+} and I_C and I_{AHP} can also indicate possible pathologic mechanisms. One role for these Ca^{2+} -mediated outward currents that may be important is that they limit Ca^{2+} influx by repolarizing the cell when Ca^{2+} currents are turned on. Intracellular Ca^{2+} is an important messenger for several mechanism, for example muscular contraction, but excessive $[Ca^{2+}]_{in}$ is a noxious agent. There are thus at least three negative feedback mechanisms for limiting the flow of Ca^{2+} – first, voltage-dependent inactivation (e.g. the w particle of I_{Ca}) of Ca^{2+} currents; second, reduction of E_{Ca} with Ca^{2+} influx; and finally, the just mentioned Ca^{2+} -mediation of repolarizing currents. These mechanisms suggest possible pathologic roles for some of the mechanisms. For example, as shown in Chapter 9 blocking of I_{AHP} causes I_C to step in and eventually limit further repetitive firing. On the other hand, if both these currents are blocked repetitive firing may go unchecked, with a subsequent larger buildup of $[Ca^{2+}]_{in}$ to, perhaps, pathological levels.

10.5 Why Model?

Why a model provides more information than that which is put into it, particularly when the model attempts to describe a fairly complicated system, is not always obvious. However, there are some compelling reasons to employ this approach, including the following:

- Modelling helps answer the question as to whether or not the collection of currents described experimentally for this cell is sufficient to account for the observed behavior.

- Limited data for a non-linear system cannot uniquely specify the system. Modelling is a way to generate plausible mechanisms that can

then be tested as more data becomes available.

- Modelling provides the experimentalist with a way to examine high-uncertainty data and can stimulate alternative explanations when experimental results are inconsistent with the current body of knowledge, as is embodied in the model.

- More specific to the results discussed here, if modelling indicates that some currents only affect spike shape then this is evidence for some interesting role for spike shape modulation. This in turn can give suggest new ideas as to how information is encoded in CNS.

10.6 Questions Posed by the Model in Regard to Current Mechanisms and Kinetics

Does it really matter what the time constant for decay is at potentials greater than about -40 mV, as long as it is much greater than the time constant for activation, considering that the spike will be repolarized before inactivation can take place? Also, what is the usefulness of inactivation mechanisms for some currents, in particular for the K^+ currents? As demonstrated by the model, during normal activity these currents are removed primarily by the removal of activation. So far, a clear role for inactivation mechanisms has not been established, but finding such a role is tempting, if one assumes that these mechanisms do not exist solely for the complication of voltage-clamp protocols.

Since we do not see all aspects of current-specific behavior in all HPC (e.g. do all HPC exhibit Ca^{2+}-dependent fAHP?) the question remains as to why some cells have certain characteristics while others don't.

10.7 Interpreting the Model Behavior

Given the speculative nature of many of the currents that I have presented in the model, any results that reflect the interaction of many of the model elements must be regarded as preliminary. None the less, there are a few interpretations that we can draw that may reflect mechanisms in actual cells.

A key question to be answered for any of the currents is whether or not a given current is modulated *in vivo*, either physiologically or pathophysiologically. From an evolutionary standpoint, for a current to have a

physiological role via selective control of that current, clearly the controlling factor must be present under physiological conditions. On the other hand, in certain pathophysiological states a specific current may be modulated in order to compensate for the problem. One would suspect that if a current has evolved (that is survived) there must be a motivation for its presence that is manifested in either physiologic conditions (e.g. as a computational mechanism) or pathologic conditions (e.g. as a compensatory or protective mechanism, or as well a computational mechanism).

For several of the currents described here such endogenous factors have been identified. For example, I_M is inhibited by muscarinic (physiologically, cholenergic) agonists. I_A has been reported to be inhibited by acetylcholine (Nakajima et al, 1986), and I_{AHP} is inhibited by muscarinic agonists (Madison et al.1987) and noradrenaline (Madison and Nicoll,1986). Speculation as to whether there are as yet undiscovered mechanisms *in vivo* for modulating some of the other currents, for example the three proposed Na^+ currents, is interesting.

10.8 The Effect of Populations of Neurons as Distinct from Single Cells, and the Implications for Graded Inhibition of HPC Currents

We have considered the all-or-nothing contribution of the various currents, i.e. either a given current is present at its normal strength or it is blocked completely. This description may be oversimplified in two ways. First, the mechanism that blocks a given current may have a graded effect with respect to a single neuron. For example, cholenergic input may be diffuse over the soma, and at a given time only part of these afferents may be activated and, subsequently, only a portion of the I_M channel or I_{AHP} channel population inhibited. Second, inhibition of a given current must be thought of not only in terms of a single cell but of a local population of cells, the size of the population depending on the neuro-architecture of a given region and the efferents of interest. Activation of a cholenergic tract which terminates in a localized area in CA3 may impinge on thousands of HPCs. Assuming that (worst case) the I_M of a given HPC in the area is then either turned on or off completely, the behavior of the population is such that there will be a graded response. This graded response will in turn depend on the strength of the cholenergic tract activity.

The key point here is that thinking about the information processing

properties of single neurons only in isolation deals with just part of the problem. Rather, considering how a *population* of neurons behaves is imperative. No single cell is an island, and removal of a single pyramidal cell from the hippocampus will probably have zero functional effect.

On the other hand, understanding the spectrum of behavior inherent in the individual functional unit (in this case the single neuron[1] is vital to deriving the behavior of the group, particularly when the size of that group varies depending on the system being considered.

10.9 Other Issues Suggested by the Modelling Approach

One interesting possibility posed by the model is that Ca^{2+}-mediated currents might be used as a fast-response transducer for monitoring intracellular Ca^{2+}. Previously, this problem has been addressed by different methods, including via measurement with microelectrodes [29], with questionable results.

In order to use Ca^{2+}-mediated currents as a transducer, it will be required to verify the relationships between activation of these currents and Ca^{2+} concentration appropriate for these currents, for example by using patch clamp protocols. Modelling can then be used to extract estimates of the time course of Ca^{2+} concentration given limited data, since the simulation of current clamp protocols establish useful constraints between the relevant parameters. In the results presented here, the time course of intracellular Ca^{2+} was tightly linked to both the membrane voltage and the different currents.

[1] Of course the definition of what constitutes the "individual functional unit" is not fixed - this may range from single channels to specific areas of a dendritic tree to the single cell to subfields to fields on up through the main systems in the CNS.

Chapter 11

FUTURE DIRECTIONS

11.1 Introduction

The model presented here is a preliminary one: at this point there are only a few conclusions that may be drawn from it with confidence regarding the functional aspects of the entire cell. The data base, at present, is sparse, and it was necessary to augment the available information with reasonable speculations on unknown mechanisms. In some respects this effort has been successful in reproducing the qualitative aspects of HPC response. Other aspects have not been simulated well, and it remains to obtain additional data from cells in order to fill in the gaps.

11.2 Some Experiments for the Future

Some experiments that are suggested by the model results include the following:

- Validate assumptions regarding electrotonic structure using frequency domain techniques.
- Evaluate the method for estimating the electrotonic parameters of the dendritic tree from histological data that was presented in Section 3.8.2.
- Investigate contribution of apparent soma leak by microelectrode. If contribution is significant during electrophysiological measurements, then use model to determine behavior of undamaged cell.

- Validate presence of proposed Na^+ channels.
- Validate voltage-dependence of I_C and I_{AHP}.
- Determine Ca^{2+}-dependence of I_C and I_{AHP}.
- Develop versions of model for different hippocampal sub-fields or different species.
- Further investigation of the relationship between various parameters and functional sensitivity, e.g. does changing I_{Na-rep} parameters affect firing patterns.
- In general, devise voltage-clamp protocols to validate assumptions for current parameters.
- Test description of Ca^{2+} system.
- Investigate more quantitively the temperature-dependence on HPC parameters.
- Run experiments to check the model predictions, as possible, for the various patterns of repetitive firing as shown in Chapter 9.

11.3 Testing the "Super" Cell Assumption

During the analysis of the HPC literature it became apparent that developing an experimental protocol in which evaluation of *several* currents *and* the linear response for a single preparation would be very valuable. A significant handicap in the building of HIPPO was that the available data was derived from a vast variety of cells. On the other hand, the HIPPO description tacitly assumes that all the currents/characteristics reviewed could be expressed in a single cell, and in fact this (probably fictional) "super" HPC is the system being modelled. Indeed, one of the more remarkable aspects of the model is that it was possible to derive a single system description that simulated such a wide range of responses.

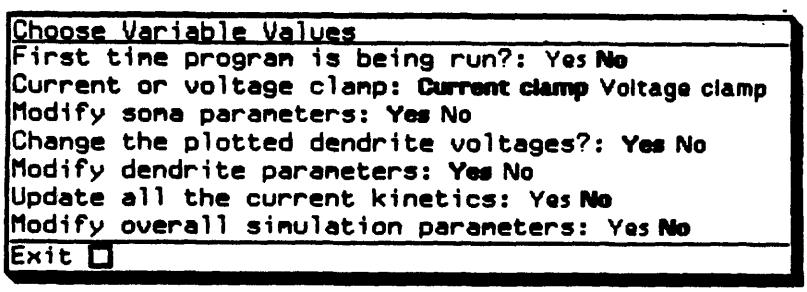
On the other hand, a single real cell may not embody every detailed response presented here, and a given cell probably expresses only some limited subset of the reported behavior. Running future experiments with this in mind, and to design a suitable protocol that would shed light on the complete behavior of a given preparation in order to test the conclusions of the model will be useful.

Appendix A

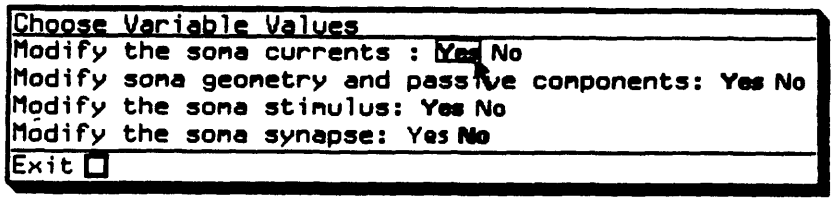
A SAMPLE SIMULATION SESSION

In this appendix a typical simulation session will be demonstrated. The first step in running HIPPO is configuring the LISP environment with the proper window frame. This is done by calling the function STARTUP. Next, the function CLAMP is called. The first task of CLAMP is to present the user with a series of menus that set the parameters for the current simulation. These menus will be illustrated below.

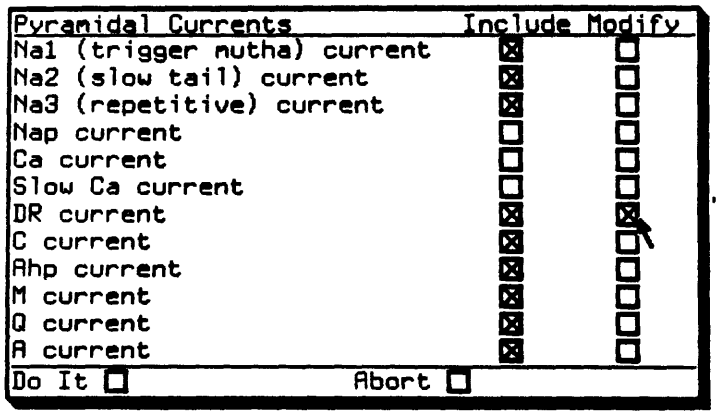
The first menu to appear is -



In this simulation all the options will be selected. The next menu to appear asks which soma parameters will be modified -



The next menu asks which soma currents will be included and/or modified -



In this case I_{Ca} and I_{CaS} will be killed (as if Ca^{2+} blockers had been added to the cell medium) and the parameters for I_{DR} will be modified.

The next menu changes the parameters for I_{DR} -

```
Delayed-Rectifier Potassium Current
DR-current absolute conductance [micro-S]: 0.7
Block some fraction of absolute conductance [0-1]; 1.0

 ** X Variable Kinetics **

V12 for Dr x: -18
Alpha-base value for Dr x at V12: 0.008
Valence for Dr x: 12
Gamma for Dr x: 0.95
Minimum value for time constant [ms]: 0.5
```

Here the kinetics of I_{DR} have been shifted +10mV along the voltage axis, thereby increasing the threshold for the activation of this current.

The next menu allows for modification of the passive soma parameters -

```
Passive components
Soma sphere radius [micrometers] : 17.5
Leakage battery [mV]    : -70.0
Na reversal potential [mV]    : 50.0
K reversal potential [mV]    : -85.0
Ca reversal potential [mV]    : 110.0
Calculate C-mem from geometry (yes) or use input capacitance (no): Yes No
Membrane capacitance [microfaradssq-cm]    : 1.0
Input capacity [nF] : 0.15
Calculate *RS-MEM from geometry (yes) or use input impedance (no): Yes No
Membrane resistance [ohm-cm-cm] : 850.0
Input impedance [MOhm] (used to substitute for soma and dendrite Rin only) : 39.0
Temperature of experiment [Celsius]: 30
Q-10 [Rate constant coefficient per 10 degrees]: 3.0
Q-10 [Ionic conductance coefficient per 10 degrees]: 1.5
Include electrode shunt conductance (if no the g-shunt will be ignored)?: Yes No
Electrode shunt [Mohms]: 1.0e7
Constant current injected [nA]: -0.25
```

In this case the temperature of the simulation has been set to 30°C.

Now the menu for the soma current stimulus comes up. This is set to inject 1nA into the soma for 5 milliseconds at the beginning of the simulation run -

```
Setting Up Current Clamp
Do you want current injected into the soma?: Yes No
Current clamp by : Command array Entered steps
Enter name of current command array -: NIL
Step 1 amplitude [na]: 1
       For how long [ms]: 5
Step 2 amplitude [na]: 0
       For how long [ms]: 0
Step 3 amplitude [na]: 0.0
       For how long [ms]: 0.0
Step 4 amplitude [na]: 0.0
       For how long [ms]: 0.0
Step 5 amplitude [na] : 0.0
For how long (this will change the duration of the simulation)[ms]: 60
Exit
```

Now the dendrite will be set up -

```
  ***  SETTING UP THE DENDRITES  ***

 -- DENDRITE STRUCTURE --

How many apical dendrite shaft segments? : 5
Include apical dendrite shaft: Yes No
Modify it?: Yes No
How many apical dendrite left branch segments? : 0
Include apical dendrite left branch: Yes No
Modify it?: Yes No
How many apical dendrite right branch segments? : 0
Include apical dendrite right branch: Yes No
Modify it?: Yes No
How many basal dendrite segments? : 0
Include basal dendrite: Yes No
Modify it?: Yes No

  -- DENDRITE CHARACTERISTICS --

Modify dendrite passive components: Yes No
Modify the dendrite current stimulus: Yes No
Modify the dendrite synapse: Yes No
Modify the currents of the modified dendrites: Yes No

Exit
```

```
Choose Variable Values
Do all the  5 apical shaft segments have the same geometry?: Yes No
Exit
```

```
Choose Variable Values
Length of segment [micrometers]: 240
Diameter of segment [micrometers]: 12.0
Exit
```

The passive characteristics of the dendrite segments will now be set -

```
Passive Properties of dendrite Segments
axon membrane capacitance [microfaradssq-cm]    : 0.1
axon membrane resistance [ohm-cm-cm] : 50000.0
axon axoplasm resistance [ohm-cm] : 25.0
dendrite membrane capacitance [microfaradssq-cm]    : 1.0
dendrite membrane resistance [ohm-cm-cm] : 40000.0
dendrite axoplasm resistance [ohm-cm] : 200.0
dendritic leak potential [mv] : -70.0
Plot all the voltages in solid lines: Yes No
Exit
```

And current - 2na from 25 milliseconds to 30 milliseconds into the simulation - will be injected into the distal dendritic segment -

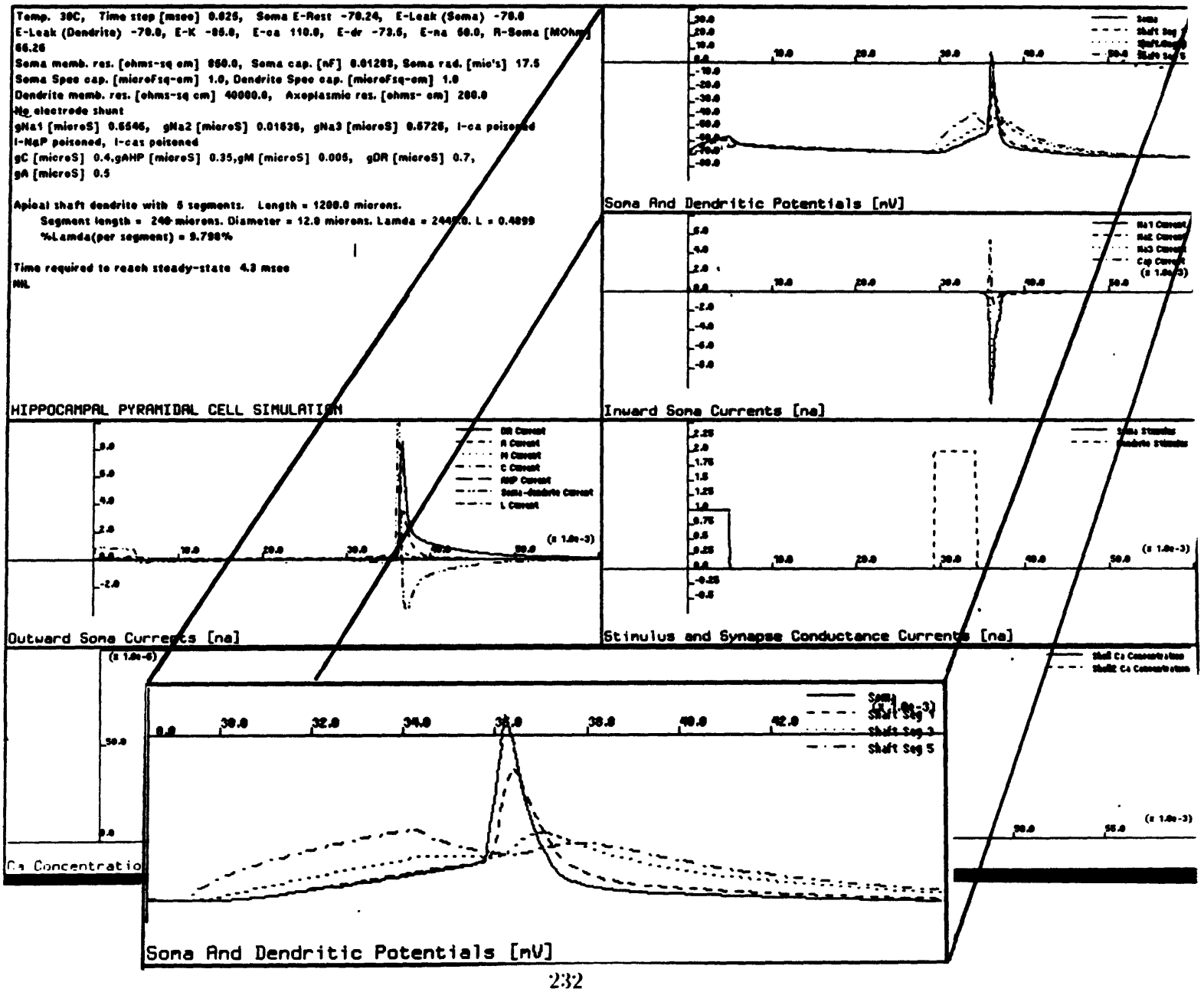
```
Choose Variable Values
Do you want current injected into the  apical dendrite shaft?: Yes No
Segment to inject current into - : 5
Step 1 amplitude [na]: 0.0
       For how long [ms]: 25
Step 2 amplitude [na]: 2
       For how long [ms]: 30
Step 3 amplitude [na]: 0.0
       For how long [ms]: 0.0
Step 4 amplitude [na]: 0.0
       For how long [ms]: 0.0
Step 5 amplitude [na]: 0.0
       For how long [ms]: 0.0
```

HIPPO now begins calculating the network response. While the simulation is running, the elapsed time is displayed -

```
Length of simulation + time for steady state - 76.07ms Current time - 12.2ms
HIPPOCAMPAL PYRAMIDAL CELL SIMULATION
```

When the simulation is complete, the relevant voltages and currents are plotted, along with a printout of the parameters. The output of the simulation run is shown.

Various characteristics of the simulation can be examined more closely as desired. For example, suppose the response to the dendritic stimulus is of interest. The relevant portions of the plots can be readily zoomed as shown.



Appendix B

HIPPO ALGORITHM

In this model a spherical-soma/dendritic-cable approximation of the pyramidal cell is reduced to an electrical network. HIPPO calculates the response of the network using a modified predictor-corrector scheme, based on that used by Cooley and Dodge. [11]. At any given time step this algorithm finds the set of solutions by a iteration from the previous set. The inputs to the network include:

- intrinsic non-linear conductances and their equilibrium potentials
- current injected into one or more compartments
- controlled voltage source placed in parallel in the soma
- synaptic conductances

The outputs of the network include:

- voltage and the derivative of the voltage
- state variable values and their derivatives
- individual branch currents

These values are found for every compartment in the network.

The program first calculates the steady state of the network (if one exists) for the current set of simulation conditions. If a steady state does not exist (e.g the cell fires spontaneously) a quasi-steady-state solution is used as the initial values for the simulation. The algorithm then proceeds as follows for each time increment:

1. Estimate the voltage of each compartment by open integration using the value at the last time step, the value of the derivative at the last time step, and the time step. If this is the first time step, then the previous voltage is the steady-state voltage.

$$V'(n\Delta t) = V([n-1]\Delta t) + (\Delta t \times \dot{V}([n-1]\Delta t))$$

where $V'(n\Delta t)$ is the estimate of the voltage of a given compartment. $V([n-1]\Delta t)$ is the voltage at the previous time step, Δt is the size of the time increment, and $\dot{V}([n-1]\Delta t)$ is the time derivative of the voltage at the last time step.

2. Estimate the steady state values of the state variables and their time constants at the current time using the voltage estimates. For example the steady state value of the $I_{Na-trig}$ m variable is estimated as:

$$m'_{Na-trig,\infty}(n\Delta t) = f(V'(n\Delta t))$$

Likewise the time constant for the $I_{Na-trig}$ m is estimated :

$$\tau'_{m,Na-trig}(n\Delta t) = g(V'(n\Delta t))$$

where $f()$ gives the steady state value ($m_{Na-trig,\infty}$) of $m_{Na-trig}$ at a given voltage, and $g()$ gives the time constant ($\tau_{m,Na-trig}$) at a given voltage. Similar equations are used for the activation and inactivation variables for all the currents that are included in any given compartment. Note that functions like ($f()$ and $g()$) are among the key results either measurements of cell parameters or the estimates derived with the model.

3. Estimate the present value for the state variables by trapezoidal approximation, using the old values for the state variables and their derivatives, the estimates for their current steady state value, and the estimates for their time constants. For example,

$$m'_{Na-trig}(n\Delta t) = \frac{m_{Na-trig}([n-1]\Delta t) + \frac{\Delta t}{2}(\dot{m}_{Na-trig}([n-1]\Delta t) + \frac{m'_{Na-trig,\infty}(n\Delta t)}{\tau'_{m,Na-trig}(n\Delta t)})}{1 - \frac{\Delta t}{2\tau'_{m,Na-trig}(n\Delta t)}}$$

4. The conductances are estimated from the state variable estimates. For example, for the $I_{Na-trig}$ conductance the estimates for $m_{Na-trig}$ and $h_{Na-trig}$ are used as follows -

$$g'_{Na-trig}(n\Delta t) = m'_{Na-trig}(n\Delta t)h'^{2}_{Na-trig}(n\Delta t)\overline{g}_{Na-trig}$$

where $g'_{Na-trig}(n\Delta t)$ is the current estimate for the $I_{Na-trig}$ conductance, and $\overline{g}_{Na-trig}$ is the total conductance for the $I_{Na-trig}$ current. The Ca^{2+}-dependent K^{+} conductances are estimated using the values of $[Ca^{2+}]_{shell.1}$ and $[Ca^{2+}]_{shell.2}$ calculated at the previous time step.

5. Conservation of currents at circuit nodes (KCL) and the appropriate branch equations are used to calculate the estimated capacitive current for each compartment at the current time. This current is then used with the value of the capacitance of the compartment to calculate the derivative of the compartment voltage, given the estimates for the conductances, the estimates for the voltages in adjacent compartments, and the value of any injected current into the compartment. For the circuit topology most often used in the simulations (see Figure 1.1) the expression of KCL for the estimated soma currents is :

$$\begin{aligned} I'_{stimulus} + I'_{capacity} + I'_{Na-trig} + I'_{Na-rep} + I'_{Na-tail} \\ + I'_{A} + I'_{DR} + I'_{C} + I'_{M} + I'_{AHP} + I'_{Q} \\ + I'_{Ca} + I'_{CaS} + I'_{L} + I'_{dendrite-soma} &= 0 \end{aligned}$$

The relevant branch equations are

$$I'_{capacity} = C_{soma}V'_{soma}(n\Delta t)$$

$$I'_{Na-trig} = g'_{Na-trig}(n\Delta t) \times (E_{Na} - V'_{soma}(n\Delta t))$$

$$I'_{Na-rep} = g'_{Na-rep}(n\Delta t) \times (E_{Na} - V'_{soma}(n\Delta t))$$

$$I'_{Na-tail} = g'_{Na-tail}(n\Delta t) \times (E_{Na} - V'_{soma}(n\Delta t))$$

$$I'_A = g'_A(n\Delta t) \times (E_K - V'_{soma}(n\Delta t))$$

$$I'_{AHP} = g'_{AHP}(n\Delta t) \times (E_K - V'_{soma}(n\Delta t))$$

$$I'_{DR} = g'_{DR}(n\Delta t) \times (E_K - V'_{soma}(n\Delta t))$$

$$I'_C = g'_C(n\Delta t) \times (E_K - V'_{soma}(n\Delta t))$$

$$I'_M = g'_M(n\Delta t) \times (E_K - V'_{soma}(n\Delta t))$$

$$I'_Q = g'_Q(n\Delta t) \times (E_K - V'_{soma}(n\Delta t))$$

$$I'_{Ca} = g'_{Ca}(n\Delta t) \times (E_{Ca} - V'_{soma}(n\Delta t))$$

$$I'_{CaS} = g'_{CaS}(n\Delta t) \times (E_{Ca} - V'_{soma}(n\Delta t))$$

$$I'_L = g_L \times (E_{Leak} - V'_{soma}(n\Delta t))$$

$$I'_{Shunt} = g_{Shunt} \times V'_{soma}(n\Delta t)$$

$$I'_{dendrite-soma} = g_{dendrite-soma} \times (V'_{dendrite-segment1}(n\Delta t) - V'_{soma}(n\Delta t))$$

Rearranging to get an expression for $V'_{soma}(n\Delta t)$.

$$V'_{soma}(n\Delta t) = \frac{1}{C_{membrane}} \times$$

$$\Big((g'_{Na-trig}(n\Delta t) + g'_{Na-rep}(n\Delta t) + g'_{Na-tail}(n\Delta t)) \times (E_{Na} - V'_{soma}(n\Delta t))$$

$$+(g'_A(n\Delta t) + g'_{AHP}(n\Delta t) + g'_{DR}(n\Delta t)$$
$$+g'_C(n\Delta t) + g'_M(n\Delta t) + g'_Q(n\Delta t)) \times (E_K - V'_{soma}(n\Delta t))$$
$$+(g'_{Ca}(n\Delta t) + g'_{CaS}(n\Delta t)) \times (E_{Ca} - V'_{soma}(n\Delta t))$$
$$+g_L \times (E_{Leak} - V'_{soma}(n\Delta t)) + g_{Shunt} \times -V'_{soma}(n\Delta t)$$
$$+g_{dendrite-soma} \times (V'_{dendrite-segment1}(n\Delta t) - V'_{soma}(n\Delta t))\Big)$$

Similar equations are derived for all the other compartment voltage derivatives.

6. A second estimate of the compartment voltages is made with trapezoidal approximation using the previous values for the voltages and their derivatives and the estimate for the present derivative. For example, the second estimate of the soma voltage is derived as follows-

$$V''(n\Delta t) = V([n-1]\Delta t) + \frac{1}{2}(\dot{V}([n-1]\Delta t) + \dot{V}'(n\Delta t))$$

where $V''(n\Delta t)$ is the second estimate for the compartment voltage. Recall that both the voltage, $V([n-1]\Delta t)$, and its derivative, $\dot{V}([n-1]\Delta t)$, were stored as results from the previous time step.

7. The new voltage estimates are compared with the previous voltage estimates. If any of these estimates is not within some convergence criteria ϵ, then the algorithm goes back to step 2 using the *mean* of the previous and present voltage estimates.

8. If all the voltage estimates are within the convergence criteria then these estimates are taken as the present values for the voltages. A final estimate of the state variables and the derivatives of the voltages are then calculated, once again using steps 2 through 6. The derivatives of the state variables are also calculated using the appropriate differential equations (ref eq. 1.). These values are also stored as the state of the network at the present time.

9. $[Ca^{2+}]_{shell.1}$ and $[Ca^{2+}]_{shell.2}$ for the current time step is calculated using the current value of I_{Ca} and I_{CaS} and the appropriate differential equations (see Chapter 6).

10. Increment the time and continue simulation.

Note that $[Ca^{2+}]_{shell.1}$ and $[Ca^{2+}]_{shell.2}$. and thus the interaction between these concentrations and I_C and I_{AHP}. are calculated *out* of the predictor-corrector loop in order to speed up execution time. This is reasonable since the time constants for the influx of Ca^{2+} and the change in the compartment concentrations are much slower than the typical time step used in the simulations (0.05 milliseconds).

The stability of the algorithm was primarily a function of the time step and the state variable with the fastest kinetics. Runs for a given simulation were done with the largest time step that resulted in a convergent solution. Typically simulations were run with a time step of 0.05 milliseconds. and an ϵ of 0.1 millivolts. The accuracy of key simulations was checked by re-running the simulation with a small time step and a small epsilon (typically 0.01 milliseconds and 0.01 millivolts. respectively). Running time for simulations with a 0.05 millisecond time step and $\epsilon = 0.1$ millivolts was between 0.5 and 1 second (real time) per millisecond of simulation.

Appendix C

OVERVIEW OF THE HIPPO CODE

The HIPPO program is written in ZetaLisp, a dialect of Lisp that is implemented on the Symbolics 3600-type computer. Although this code will not run under Common Lisp, converting it should not be very difficult.

The output of HIPPO assumes that a plotting package written by Patrick O'Donnell has been loaded into the machine. Again, this part of the program could be readily modified to run on another system.

Some of the features of this code include the evaluation of the voltage-dependent gating variable functions and storage of the results in arrays *before* simulation runs so as to speed run times. In addition, it is relatively straightforward to add new conductances to the model due to the modularity of the program.

Copies of this code are available on cartridge tape from the author.

Appendix D

HIPPO LISTING

```
;; -*- Mode: LISP; Syntax: Zetalisp; Package: (HIPPO); Base: 10; Fonts: CPTFONT,CPTFONTI; Hardcopy-Fonts: FIX10,TIM
ESROMAN10BI -*-;;
;;;
;;; ************ HIPPO - SIMULATION OF HIPPOCAMPAL PYRAMIDAL NEURONS ***************
;;;
;;;      **** UNITS ****
;;;
;;; All dimensions are as follows -
;;;
;;; Time (milliseconds) except for "time" output array where the units are seconds.
;;; Distance (micrometers)
;;; Surface area (square-cm)
;;; Volume (cubic micrometers)
;;;
;;; Voltage (millivolts)
;;; Current (nano-amps)
;;;
;;; Specific membrane capacitance (microfarads per square-cm)
;;; Specific membrane resistivity (ohms-cm-cm)
;;; Capacitance (nanofarads)
;;; Resistance (mega-ohms)
;;;
;;; Conductance (microsiemans)
;;; Conductance density (millisiemans per square-cm)
;;;
;;;
;;;      **** Define all the global variables and arrays ****
;;;
;;; The naming convention for the variables is as follows -
;;;
;;; variable-name    = local variable
;;; *variable-name   = global variable
;;; VARIABLE-NAME    = global array
;;; variable-name$   = slot location in array
;;; *variable-name*  = global (output) list


;;; These arrays hold all the state variables for each compartment. DENDRITE is a two dimensional array to
;;; include all the dendritic segments.
(defvar BASAL-DENDRITE (make-array '(50 200) :initial-value 1.0))
(defvar APICAL-1-DENDRITE (make-array '(50 200) :initial-value 1.0))
(defvar APICAL-2-DENDRITE (make-array '(50 200) :initial-value 1.0))
(defvar APICAL-SHAFT-DENDRITE (make-array '(50 200) :initial-value 1.0))
(defvar *total-segments 10)
(defvar SOMA (make-array 200 :initial-value 0.0))
(defvars-w-value (*dendrite-synapse-step 0)(*soma-synapse-step 0)
  (*start-dendrite-synapse 10.0)(*start-soma-synapse 10.0))

(defvar m-na1-inf-array (make-array 1700 :initial-value 1.0))
(defvar h-na1-inf-array (make-array 1700 :initial-value 1.0))
(defvar t-m-na1-array (make-array 1700 :initial-value 1.0))
(defvar t-h-na1-array (make-array 1700 :initial-value 1.0))
(defvar m-na2-inf-array (make-array 1700 :initial-value 1.0))
(defvar h-na2-inf-array (make-array 1700 :initial-value 1.0))
(defvar t-m-na2-array (make-array 1700 :initial-value 1.0))
(defvar t-h-na2-array (make-array 1700 :initial-value 1.0))
(defvar m-na3-inf-array (make-array 1700 :initial-value 1.0))
(defvar h-na3-inf-array (make-array 1700 :initial-value 1.0))
(defvar t-m-na3-array (make-array 1700 :initial-value 1.0))
(defvar t-h-na3-array (make-array 1700 :initial-value 1.0))

(defvar x-nap-inf-array (make-array 1700 :initial-value 1.0))
(defvar t-x-nap-array (make-array 1700 :initial-value 1.0))

(defvar s-ca-inf-array (make-array 1700 :initial-value 1.0))
(defvar w-ca-inf-array (make-array 1700 :initial-value 1.0))
(defvar t-s-ca-array (make-array 1700 :initial-value 1.0))
(defvar t-w-ca-array (make-array 1700 :initial-value 1.0))
(defvar x-cas-inf-array (make-array 1700 :initial-value 1.0))
(defvar t-x-cas-array (make-array 1700 :initial-value 1.0))

(defvar x-a-inf-array (make-array 1700 :initial-value 1.0))
(defvar y-a-inf-array (make-array 1700 :initial-value 1.0))
(defvar t-x-a-array (make-array 1700 :initial-value 1.0))
(defvar t-y-a-array (make-array 1700 :initial-value 1.0))

(defvar x-c-inf-array (make-array 1700 :initial-value 1.0))
(defvar y-c-inf-array (make-array 1700 :initial-value 1.0))
(defvar t-x-c-array (make-array 1700 :initial-value 1.0))
(defvar t-y-c-array (make-array 1700 :initial-value 1.0))
```

```
(defvar x-dr-inf-array (make-array 1700 :initial-value 1.0))
(defvar y-dr-inf-array (make-array 1700 :initial-value 1.0))
(defvar t-x-dr-array (make-array 1700 :initial-value 1.0))
(defvar t-y-dr-array (make-array 1700 :initial-value 1.0))
(defvar x-m-inf-array (make-array 1700 :initial-value 1.0))
(defvar t-x-m-array (make-array 1700 :initial-value 1.0))
(defvar x-q-inf-array (make-array 1700 :initial-value 1.0))
(defvar t-x-q-array (make-array 1700 :initial-value 1.0))

(defvar z-ahp-inf-array (make-array 1700 :initial-value 1.0))
(defvar t-z-ahp-array (make-array 1700 :initial-value 1.0))
(defvar y-ahp-inf-array (make-array 1700 :initial-value 1.0))
(defvar t-y-ahp-array (make-array 1700 :initial-value 1.0))

(defvar voltage-array (make-array 1700 :initial-value 1.0))     ;Use this array for plotting variable curves


;;; Set up labels for the various dendrite-segment and soma array slots.
(defvars-w-value
  (label$ 0)

  ;;The last permanent values for the voltage and the derivative of the voltage.
  (voltage$ 1) (voltage-dot$ 2)

  ;;"-est1", the present estimate of the voltage and "-est1-dot", the present estimate of the derivative of the
 ;;voltage,which are to be used in calculating the next estimate of the voltage.
  (voltage-est1$ 3)(voltage-est1-dot$ 4)

  ;;"-est2", the next estimate of the voltage. When this is calculated it will then be compared with the
 ;;previous estimate to see if the two values are within the convergence criterium.
  (voltage-est2$ 5)

  ;;Passive parameters.
  (capacitance$ 10)(length$ 11)(diameter$ 12)(e-rest$ 13)(total-segments$ 14)(include-me$ 15)(plot-me$ 16)
  (ca-conc-shell$ 17)  (ca-conc-shell-dot$ 18)
    (ca-conc-shell2$ 117)  (ca-conc-shell2-dot$ 118)

  ;;The last permanent values for the state variables and their derivatives, calculated with the last permanent
 ;;value for the voltage.
  (m-na1$ 20)(m-na1-dot$ 21)  (h-na1$ 22)(h-na1-dot$ 23)
  (m-na2$ 120)(m-na2-dot$ 121)  (h-na2$ 122)(h-na2-dot$ 123)
  (m-na3$ 124)(m-na3-dot$ 125)  (h-na3$ 126)(h-na3-dot$ 127)
  (s-ca$ 24)(s-ca-dot$ 25)  (w-ca$ 26)(w-ca-dot$ 27)
  (x-a$ 28)(x-a-dot$ 29)  (y-a$ 30)(y-a-dot$ 31)
  (x-dr$ 32)(x-dr-dot$ 33)  (y-dr$ 34)(y-dr-dot$ 35)
  (x-m$ 36)(x-m-dot$ 37)
  (x-q$ 38)(x-q-dot$ 39)
  (x-nap$ 40)(x-nap-dot$ 41)
  (x-c$ 42)(x-c-dot$ 43)  (y-c$ 44)(y-c-dot$ 45) (w-c$ 54)(w-c-dot$ 55)
  (x-cas$ 46)(x-cas-dot$ 47)
  (z-ahp$ 48)(z-ahp-dot$ 49)(y-ahp$ 50)(y-ahp-dot$ 51)(w-ahp$ 52)(w-ahp-dot$ 53)



  ;;Absolute conductances.
  (g-axial$ 60)(g-synapse$ 61)
  (g-leak$ 62)(gbar-na1$ 63)
  (gbar-na2$ 163)
  (gbar-na3$ 164)
  (gbar-ca$ 64)
  (gbar-k$ 65)
  (gbar-m$ 66)(gbar-dr$ 67)(gbar-c$ 68)(gbar-q$ 69)(gbar-nap$ 70)(gbar-a$ 71)
  (gbar-cas$ 72)
  (gbar-ahp$ 73)

  ;;Flags for the dendrite currents.
  (include-na$ 80)
  (include-ca$ 81))


;;; These are used to store the steady state values and the time constants for the voltage clamp protocol.
(defvars *xqinf *tq *xcinf *txc *ycinf *tyc *wcinf *twc
         *xminf *tm
         *mna1inf *tmna1 *hna1inf *thna1
         *mna2inf *tmna2 *hna2inf *thna2
         *mna3inf *tmna3 *hna3inf *thna3
         *mcainf *tmca *hcainf *thca
         *xdrinf *txdr *xcasinf *txcas *zahpinf *tzahp *yahpinf *tyahp *wahpinf *twahp
```

```
        *ydrinf *tydr *xainf *txa *yainf *tya *xnapinf *txnap )

;;; These are the arrays which hold the soma and dendrite synaptic conductances [micro-S]
(defvar SOMA-SYNAPSE (make-array 10000))
(defvar DENDRITE-SYNAPSE (make-array 10000))

;;; Various global variables
(defvars *gs-na1-est *gs-na2-est *gs-na3-est
        *gs-nap-est  *gs-ca-est  *gs-a-est  *gs-c-est  *gs-m-est  *gs-dr-est *gs-cas-est *gs-ahp-est
        *gs-q-est *gs-leak *gs-synapse *gs-coupling *time *clamp-voltage *vstep *time-for-steady-state
        DENDRITE-ARRAY *g-electrode)

(defvars *a-m-na1 *b-m-na1 *a-h-na1 *b-h-na1)
(defvars *a-m-na2 *b-m-na2 *a-h-na2 *b-h-na2)
(defvars *a-m-na3 *b-m-na3 *a-h-na3 *b-h-na3)
(defvar *vclamp-command-flag 1)
(defvar *voltage-command* nil)
(defvar *iclamp-command-flag 1)
(defvar *current-command* nil)
(defvar *stim-seg 4)
(defvar *syn-seg 4)
(defvars *plot-list1 *label-list1 *plot-list2 *label-list2 *plot-list3 *label-list3 *plot-list4 *label-list4
        *plot-list5 *label-list5)

(defvars-w-value (*i-stim-1 0.0)(*i-stim-2 0.0)(*i-stim-3 0.0)(*i-stim-4 0.0)(*i-stim-5 0.0)
                 (*t-stim-1 0.0)(*t-stim-2 0.0)(*t-stim-3 0.0)(*t-stim-4 0.0)(*t-stim-5 0.0)
                 (*i-den-stim-1 0.0)(*i-den-stim-2 0.0)(*i-den-stim-3 0.0)(*i-den-stim-4 0.0)
                 (*i-den-stim-5 0.0)
                 (*i-den-stim-6 0.0)(*i-den-stim-7 0.0)(*i-den-stim-8 0.0)(*i-den-stim-9 0.0)(*i-den-stim-10 0.0)
                 (*t-den-stim-1 0.0)(*t-den-stim-2 0.0)(*t-den-stim-3 0.0)(*t-den-stim-4 0.0)
                 (*t-den-stim-5 0.0)
                 (*t-den-stim-6 0.0)(*t-den-stim-7 0.0)(*t-den-stim-8 0.0)(*t-den-stim-9 0.0)(*t-den-stim-10 0.0)
                 (*current-stimulus-segment 5) (*i-stim 0.0)(*i-den-stim 0.0)
                 (*time-step 0)(*duration 50)(*include-soma-current t)(*include-dendrite-current nil)
                 (*plot-dendrite t)(*calculate-steady-state t)(*first-run t)(*steady-state-run nil))

(defvar *qten 3.0)                              ;Temperature dependance of rate constants. The rate constants
                                                ;are multiplied by *QTEN raised to (T-Tbase)/10, where T is the
                                                ;temperature of the simulation, and Tbase is the temperature of
                                                ;the of the experiment that measured the rate constants.

(defvars-w-value (*qten-factor-at-25 1.0) (*qten-factor-at-32 1.0)(*qten-factor-at-24 1.0)
                 (*qten-factor-at-22 1.0) (*qten-factor-at-37 1.0)
                 (*qten-factor-at-14 1.0) (*qten-factor-at-25-m 1.0))

(defvars-w-value (*soma-synapse-tau 1.0)(*soma-synapse-amplitude 1.0)(*e-synapse -25.0)
                 (*dendrite-synapse-tau 1.0) (*dendrite-synapse-amplitude 1.0) (*synapse-segment 5)
                 (*total-apical-1-segments 0) (*total-apical-2-segments 0) (*total-apical-shaft-segments 5)
                 (*total-basal-segments 0)
                 (*i-constant-injection 0.0))

(defvars-w-value (*axonal-cap-mem .1)(*axonal-r-mem 50000.0)(*axonal-r-int 25.0))

;;; Miscellaneous flags
(defvars-w-value (*modify-soma-passive-components t)(*modify-soma-stimulus t)(*modify-soma-synapse nil)
                 (*modify-soma-currents t)(*modify-soma t)(*modify-dendrite t)
                 (*segments-all-the-same t)(*simulation-flag t)(*include-soma-synapse nil)
                 (*include-dendrite-synapse nil)
                 (*modify-dendrite-synapse nil)(*modify-dendrite-stimulus NIL)(*include-dendrite t)
                 (*modify-dendrite-geometry NIL)(*modify-dendrite-passive-components NIL)
                 (*modify-dendrite-currents NIL)(*plot-voltages-solid t)(*overlay-simulations nil)
                 (*change-plot-dendrite NIL)
                 (*update-apical-1 nil)
                 (*update-apical-2 nil)
                 (*update-apical-shaft NIL)
                 (*update-basal nil)
                 (*include-apical-1 nil)
                 (*include-apical-2 nil)
                 (*include-apical-shaft t)
                 (*include-basal nil)
                 (*modify-dendrite-currents nil)
                 (*plot-results t)
                 (*update-all-kinetics nil))

(defvars-w-value (*plot-as1 nil) (*plot-as3 nil) (*plot-as5 nil) (*plot-as10 nil)
  (*plot-al1 nil) (*plot-al4 nil) (*plot-ar1 nil) (*plot-ar4 nil)
                 (*plot-b1 nil) (*plot-b4 nil))
;;; Flags for the currents
(defvars-w-value (*include-na1 T)
                 (*include-na2 T)
                 (*include-na3 T)
                 (*include-nap nil)(*include-a T)(*include-ahp nil)(*include-cas nil)
```

```
                (*include-dr T)(*include-k nil)(*include-c nil)(*include-m nil)(*include-kinetics nil)
                (*include-ca nil)(*include-q nil)(*include-shunt t))
(defvars-w-value  (*na1-mod nil) (*na2-mod nil) (*na3-mod nil) (*nap-mod nil) (*ca-mod nil) (*c-mod nil)
                  (*dr-mod nil) (*a-mod nil) (*m-mod nil) (*q-mod nil) (*cas-mod nil)(*ahp-mod nil))

;;; The simulation time step
(defvar *dt .01)                ;msecs

;;; *epsilon is the convergence criteria for the compartment voltages; *dot-epsilon is the convergence criteria
;;; for the derivative of the compartment voltages when trying to calculate the steady state.
(defvar *epsilon .01)
(defvar *dot-epsilon .01)

(defvars-w-value (*v1 -60.0)(*v2 20.0)(*v3 -60.0)(*v4 -60.0)(*v5 -60.0))

(defvar *plot-step 1)
(defvar *point-index 1)
(Defvar *plot-points 300)
(defvar *vclamp-run)
(defvar *iclamp-run)
(defvar *clamp-type 1)
;;; Output arrays
(defvars *i-current* *caps-current* *current* *time* *voltage* *na1-current* *na2-current* *na3-current*
         *nap-current* *cas-current*
         *dr-current* *a-current* *k-current* *m-current* *c-current* *ahp-current* *e-eff*
         *m-na1* *h-na1*  *m-na2* *h-na2* *m-na3* *h-na3*
         *x-dr* *y-dr* *x-a* *y-a* *g-a* *g-dr* *n-k* *x-m* *x-c* *y-c* *w-c* *m-na-dot* *g-na* *i-stim* *g-c*
         *g-c-conc*  *soma-synapse-current* *nad1-current* *cad1-current* *coupling-current* *stim-current*
         *dendrite-stim-current* *as1voltage* *as3voltage* *as5voltage* *as10voltage*
         *a11voltage* *a14voltage* *ar1voltage* *ar4voltage*
         *b1voltage* *b4voltage* *shunt-current*
         *dendrite-synapse-current* *ca-current* *soma-synapse-conductance* *dendrite-synapse-conductance*
         *q-current* *x-nap* *z-ahp* *y-ahp* *w-ahp* *g-ahp* *s-ca* *w-ca* *g-ca* *g-na1* *g-na2*  *g-na3*
         *ca-conc-shell* *ca-conc-shell2*
         *e-ca*
         *x-nap*)


(defun set-up-output-arrays (plotted-points)
  (cond-every
    (t (setq *i-stim* (make-array plotted-points :initial-value 0.0)
             *ca-conc-shell* (make-array plotted-points :initial-value 0.0)
             *ca-conc-shell2* (make-array plotted-points :initial-value 0.0)
             *e-ca* (make-array plotted-points :initial-value 0.0)
             *i-current* (make-array plotted-points :initial-value 0.0)
             *caps-current* (make-array plotted-points :initial-value 0.0)
             *current* (make-array plotted-points :initial-value 0.0) *time* (make-array plotted-points :initial-va
lue 0.0)
             *voltage* (make-array plotted-points :initial-value 0.0) *e-eff* (make-array plotted-points :initial-v
alue 0.0)
             *shunt-current* (make-array plotted-points :initial-value 0.0)
             *coupling-current* (make-array plotted-points :initial-value 0.0)
             *stim-current* (make-array plotted-points :initial-value 0.0)
             *dendrite-stim-current* (make-array plotted-points :initial-value 0.0)
             *as1voltage* (make-array plotted-points :initial-value 0.0)
             *as3voltage* (make-array plotted-points :initial-value 0.0)
             *as5voltage* (make-array plotted-points :initial-value 0.0)
             *as10voltage* (make-array plotted-points :initial-value 0.0)))
    (*include-na1 (setq *na1-current* (make-array plotted-points :initial-value 0.0)))
    (*include-na2 (setq *na2-current* (make-array plotted-points :initial-value 0.0)))
    (*include-na3 (setq *na3-current* (make-array plotted-points :initial-value 0.0)))
    (*include-ca (setq  *ca-current* (make-array plotted-points :initial-value 0.0)))
    (*include-cas (setq *cas-current* (make-array plotted-points :initial-value 0.0)))
    (*include-dr (setq *dr-current* (make-array plotted-points :initial-value 0.0)))
    (*include-a (setq *a-current* (make-array plotted-points :initial-value 0.0)))
    (*include-m (setq *m-current* (make-array plotted-points :initial-value 0.0)))
    (*include-ahp (setq *ahp-current* (make-array plotted-points :initial-value 0.0)))
    (*include-q (setq *q-current* (make-array plotted-points :initial-value 0.0)))
    (*include-c (setq *c-current* (make-array plotted-points :initial-value 0.0)))
    (*include-kinetics
      (cond-every
        (*include-na1 (setq *m-na1* (make-array plotted-points :initial-value 0.0)
                            *h-na1* (make-array plotted-points :initial-value 0.0)
                            *g-na1* (make-array plotted-points :initial-value 0.0)))
        (*include-na2 (setq *m-na2* (make-array plotted-points :initial-value 0.0)
                            *h-na2* (make-array plotted-points :initial-value 0.0)
                            *g-na2* (make-array plotted-points :initial-value 0.0)))
        (*include-na3 (setq *m-na3* (make-array plotted-points :initial-value 0.0)
                            *h-na3* (make-array plotted-points :initial-value 0.0)
                            *g-na3* (make-array plotted-points :initial-value 0.0)))
        (*include-ca (setq *s-ca* (make-array plotted-points :initial-value 0.0)
```

```
                        *w-ca* (make-array plotted-points :initial-value 0.0)
                        *g-ca* (make-array plotted-points :initial-value 0.0)))
        (*include-dr (setq *x-dr* (make-array plotted-points :initial-value 0.0)
                        *y-dr* (make-array plotted-points :initial-value 0.0)
                        *g-dr* (make-array plotted-points :initial-value 0.0)))
        (*include-a (setq *x-a* (make-array plotted-points :initial-value 0.0)
                        *y-a* (make-array plotted-points :initial-value 0.0)
                        *g-a* (make-array plotted-points :initial-value 0.0)))
        (*include-m (setq *x-m* (make-array plotted-points :initial-value 0.0)))
        (*include-ahp (setq *z-ahp* (make-array plotted-points :initial-value 0.0)
                        *y-ahp* (make-array plotted-points :initial-value 0.0)
                        *w-ahp* (make-array plotted-points :initial-value 0.0)
                        *g-ahp* (make-array plotted-points :initial-value 0.0)))
        (*include-c (setq *x-c* (make-array plotted-points :initial-value 0.0)
                        *y-c* (make-array plotted-points :initial-value 0.0)
                        *w-c* (make-array plotted-points :initial-value 0.0)
                        *g-c* (make-array plotted-points :initial-value 0.0)
                        *g-c-conc* (make-array plotted-points :initial-value 0.0)))))))


(defvars *plot-pane-1 *plot-pane-2 *plot-pane-3 *plot-pane-4 *plot-pane-5 *plot-pane-6)



;;; SETUP-MENU4 Sets up all the parameters for the current run.
(defun setup-menu4 ()
  (tv:choose-variable-values
    '((*first-run "First time program is being run?" :boolean)
      (*clamp-type "Current or voltage clamp" :choose ("Current clamp" "Voltage clamp"))
      (*modify-soma  "Modify soma parameters" :boolean)
      (*change-plot-dendrite "Change the plotted dendrite voltages?" :boolean)
      (*modify-dendrite  "Modify dendrite parameters" :boolean)
      (*update-all-kinetics "Update all the current kinetics" :boolean)
      (*simulation-flag "Modify overall simulation parameters" :boolean)))
  (if *first-run (initialize-dendrites))
  (if (equal *clamp-type "Current clamp")
      (setq *iclamp-run t *vclamp-run nil)
      (setq *iclamp-run nil *vclamp-run t))
  (cond-every (*simulation-flag (menu-for-simulation))
              (*modify-soma (menu-for-soma))
              (*modify-dendrite (menu-for-dendrite))
              (*change-plot-dendrite (menu-for-dendrite-plotting))
              ((or *change-plot-dendrite *change-plot-points *change-include-kinetics *modify-soma-currents)
               (set-up-output-arrays *plot-points)
               (format t "Made new output arrays"))
              (*update-all-kinetics (variable-array-setup))))


(defun menu-for-dendrite-plotting ()
  (tv:choose-variable-values
    '((*plot-as1 "Plot shaft segment 1" :boolean)
      (*plot-as3 "Plot shaft segment 3" :boolean)
      (*plot-as5 "Plot shaft segment 5" :boolean)
      (*plot-as10 "Plot shaft segment 10" :boolean)
      (*plot-al1 "Plot left segment 1" :boolean)
      (*plot-al4 "Plot left segment 4" :boolean)
      (*plot-ar1 "Plot right segment 1" :boolean)
      (*plot-ar4 "Plot right segment 4" :boolean)
      (*plot-b1 "Plot basal segment 1" :boolean)
      (*plot-b4 "Plot basal segment 4" :boolean))
      ':LABEL "CHOOSE PLOTTED SEGMENTS")
  (aset (if *plot-as1 t nil) APICAL-SHAFT-DENDRITE 0 plot-ms$)
  (aset (if *plot-as3 t nil) APICAL-SHAFT-DENDRITE 2 plot-ms$)
  (aset (if *plot-as5 t nil) APICAL-SHAFT-DENDRITE 4 plot-ms$)
  (aset (if *plot-as10 t nil) APICAL-SHAFT-DENDRITE 9 plot-ms$)
  (aset (if *plot-al1 t nil) APICAL-1-DENDRITE 0 plot-ms$)
  (aset (if *plot-al4 t nil) APICAL-1-DENDRITE 3 plot-ms$)
  (aset (if *plot-ar1 t nil) APICAL-2-DENDRITE 0 plot-ms$)
  (aset (if *plot-ar4 t nil) APICAL-2-DENDRITE 3 plot-ms$)
  (aset (if *plot-b1 t nil) BASAL-DENDRITE 0 plot-ms$)
  (aset (if *plot-b4 t nil) BASAL-DENDRITE 3 plot-ms$))


;;; MENU-FOR-SOMA Sets up all the parameters for the current run.
(defun menu-for-soma ()
    (tv:choose-variable-values
   '((*modify-soma-currents "Modify the soma currents " :boolean)
       (*modify-soma-passive-components  "Modify soma geometry and passive components" :boolean)
       (*modify-soma-stimulus "Modify the soma stimulus" :boolean)
```

```
      (*modify-soma-synapse "Modify the soma synapse" :boolean)))
  (cond-every (*modify-soma-currents (menu-for-soma-currents))
              (*modify-soma-passive-components (menu-for-soma-geometry-and-passive-components ))
              ((and *iclamp-run *Modify-soma-stimulus) (menu-for-soma-current-stimulus))
              ((and *vclamp-run *Modify-soma-stimulus) (menu-for-soma-voltage-stimulus))
              (*modify-soma-synapse (menu-for-soma-synapse))))

;;; MENU-FOR-DENDRITE Sets up all the parameters for the current run.
(defun menu-for-dendrite ()
  (tv:choose-variable-values
    '(""
      " -- DENDRITE STRUCTURE --"
      ""
      (*total-apical-shaft-segments "How many apical dendrite shaft segments? " :number)
      (*include-apical-shaft "Include apical dendrite shaft" :boolean)
      (*update-apical-shaft "Modify it?" :boolean)
      (*total-apical-1-segments "How many apical dendrite left branch segments? " :number)
      (*include-apical-1 "Include apical dendrite left branch" :boolean)
      (*update-apical-1 "Modify it?" :boolean)
      (*total-apical-2-segments "How many apical dendrite right branch segments? " :number)
      (*include-apical-2 "Include apical dendrite right branch" :boolean)
      (*update-apical-2 "Modify it?" :boolean)
      (*total-basal-segments "How many basal dendrite segments? " :number)
      (*include-basal "Include basal dendrite" :boolean)
      (*update-basal "Modify it?" :boolean)
      ""
      "  -- DENDRITE CHARACTERISTICS --"
      ""
      (*modify-dendrite-passive-components "Modify dendrite passive components" :boolean)
      (*modify-dendrite-stimulus "Modify the dendrite current stimulus" :boolean)
      (*Modify-dendrite-synapse "Modify the dendrite synapse" :boolean)
      (*modify-dendrite-currents "Modify the currents of the modified dendrites" :boolean)
      "")
    ':label "  ***  SETTING UP THE DENDRITES  ***  ")
  (set-dendrite-segments-and-flags)
  (cond-every
    (*update-apical-1 (aset *total-apical-1-segments   APICAL-1-DENDRITE 0 total-segments$)
                      (menu-for-dendrite-geometry APICAL-1-DENDRITE)
                      (if *modify-dendrite-currents
                          (menu-for-dendrite-currents APICAL-1-DENDRITE)))
    (*update-apical-2 (aset *total-apical-2-segments   APICAL-2-DENDRITE 0 total-segments$)
                      (menu-for-dendrite-geometry APICAL-2-DENDRITE)
                      (if *modify-dendrite-currents
                          (menu-for-dendrite-currents APICAL-2-DENDRITE)))
    (*update-apical-shaft (aset *total-apical-shaft-segments   APICAL-SHAFT-DENDRITE 0 total-segments$)
                          (menu-for-dendrite-geometry APICAL-SHAFT-DENDRITE)
                          (if *modify-dendrite-currents
                              (menu-for-dendrite-currents APICAL-SHAFT-DENDRITE)))
    (*update-basal (aset *total-basal-segments   BASAL-DENDRITE 0 total-segments$)
                   (menu-for-dendrite-geometry BASAL-DENDRITE)
                   (if *modify-dendrite-currents
                       (menu-for-dendrite-currents BASAL-DENDRITE)))
    (*modify-dendrite-passive-components (menu-for-dendrite-passive-components))
    (*Modify-dendrite-stimulus (menu-for-dendrite-stimulus))
    (*Modify-dendrite-synapse (menu-for-dendrite-synapse))))

;;; SET-DENDRITE-SEGMENTS-AND-FLAGS
(defun set-dendrite-segments-and-flags ()
  (aset  *total-apical-1-segments  APICAL-1-DENDRITE 0 total-segments$)
  (aset  *total-apical-2-segments  APICAL-2-DENDRITE 0 total-segments$)
  (aset  *total-apical-shaft-segments  APICAL-SHAFT-DENDRITE 0 total-segments$)
  (aset  *total-basal-segments  BASAL-DENDRITE 0 total-segments$)
  (if *include-apical-shaft (aset t APICAL-SHAFT-DENDRITE 0 include-me$)
      (aset nil APICAL-SHAFT-DENDRITE 0 include-me$))
  (if *include-apical-1 (aset t APICAL-1-DENDRITE 0 include-me$)
      (aset nil APICAL-1-DENDRITE 0 include-me$))
  (if *include-apical-2 (aset t APICAL-2-DENDRITE 0 include-me$)
      (aset nil APICAL-2-DENDRITE 0 include-me$))
  (if *include-basal (aset t BASAL-DENDRITE 0 include-me$)
      (aset nil BASAL-DENDRITE 0 include-me$)))

;;; INITIALIZE-DENDRITES Initializes the length and diameter of the dendrite segments, and sets the Na and Ca
;;; current to NIL.
(defun initialize-dendrites ()
  (aset "left apical branch" APICAL-1-DENDRITE 0 label$)
  (aset "right apical branch" APICAL-2-DENDRITE 0 label$)
  (aset "apical shaft" APICAL-SHAFT-DENDRITE 0 label$)
  (aset "basal branch" BASAL-DENDRITE 0 label$)
  (aset  10  APICAL-1-DENDRITE 0 total-segments$)
  (aset  0  APICAL-2-DENDRITE 0 total-segments$)
  (aset  5  APICAL-SHAFT-DENDRITE 0 total-segments$)    ; Default - just 5 apical segments.
  (aset  10  BASAL-DENDRITE 0 total-segments$)
```

```
  (aset t APICAL-SHAFT-DENDRITE 0 include-me$)
  (aset nil APICAL-1-DENDRITE 0 include-me$)
  (aset nil APICAL-2-DENDRITE 0 include-me$)
  (aset nil BASAL-DENDRITE 0 include-me$)
  (dolist (DENDRITE-ARRAY (list BASAL-DENDRITE APICAL-SHAFT-DENDRITE APICAL-1-DENDRITE APICAL-2-DENDRITE))
    (if (aref DENDRITE-ARRAY 0 include-me$)
        (do ((segment 0 (incf segment)))
            ((= segment 50))                    ; Just set up for 50 segments in each cable by default.
          (aset 240.0 DENDRITE-ARRAY segment length$)   ;
          (aset 12.0 DENDRITE-ARRAY segment diameter$)  ;
          (aset nil DENDRITE-ARRAY segment include-na$)
          (aset nil DENDRITE-ARRAY segment include-ca$)))))


(defvar *new-plot-points)
(defvar *now-include-kinetics nil)
(defvar *change-plot-points nil)
(defvar *change-currents nil)
(defvar *change-include-kinetics nil)
(defvar steady)
;;; MENU-FOR-SIMULATION Set up the overall simulation parameters.
(defun menu-for-simulation ()
  (setq steady (if *calculate-steady-state "Re-calculate" "Old value")
        *new-plot-points *plot-points
        *now-include-kinetics *include-kinetics)
    (tv:choose-variable-values
      '((steady "Calculate steady state?" :choose ("Re-calculate" "Old value"))
        (*now-include-kinetics "Interested in kinetics?" :boolean)
        (*new-plot-points "Number of points to plot (if ft-clamp will be run, then enter 2048) - " :integer)
        (*dt "Set *time step [ms]" :number)
        (*duration "Length of simulation [ms]" :number)
        (*plot-results "Plot results?" :boolean)
        (*overlay-simulations "Plot over previous data?" :boolean)
        (*epsilon "Convergence criteria in pred/corr" :number)
        (*e-holding "Holding voltage for initialization [mV]" :number))
      ':label "Setting up stimulus conditions for clamp")
    (setq *calculate-steady-state (if (equal "Re-calculate" steady) T nil)
          *plot-step (// (fixr (// *duration *dt)) *new-plot-points)
          *change-plot-points (neq *new-plot-points *plot-points)
          *plot-points *new-plot-points
          *change-include-kinetics (and *now-include-kinetics (not *include-kinetics))  ;only set *change-include-kineti
cs
                                                        ;if last time *include-kinetics was nil and now it is set
          *include-kinetics *now-include-kinetics ))


(defvar *length)
(defvar *diameter)
;;; MENU-FOR-DENDRITE-GEOMETRY Set up the geometry components of the segments
(defun menu-for-dendrite-geometry (DENDRITE-ARRAY)
  (let ((total-segments (aref DENDRITE-ARRAY 0 total-segments$))(list1)(list2))
    (setq list1 (format nil "Do all the ~2d ~A segments have the same geometry?"
                        total-segments (aref DENDRITE-ARRAY 0 label$)))
    (tv:choose-variable-values
      '((*segments-all-the-same ,list1 :boolean)))
    (cond (*segments-all-the-same
           (setq *length (aref DENDRITE-ARRAY 0 length$)
                 *diameter (aref DENDRITE-ARRAY 0 diameter$))
           (tv:choose-variable-values
             '((*length "Length of segment [micrometers]"  :number)
               (*diameter "Diameter of segment [micrometers]"  :number)))
           (do ((segment 0 (incf segment)))
               ((= segment total-segments))
             (aset *length DENDRITE-ARRAY segment length$)
             (aset *diameter DENDRITE-ARRAY segment diameter$)))
          (t
           (do ((segment 0 (incf segment)))
               ((= segment total-segments))
             (setq *length (aref DENDRITE-ARRAY segment length$) *diameter (aref DENDRITE-ARRAY segment diameter$)
                   list1 (format nil "Length of ~A segment~2d [micrometers] "
                                 (aref DENDRITE-ARRAY 0 label$) (+ 1 segment))
                   list2 (format nil "Diameter of ~A segment~2d [micrometers] "
                                 (aref DENDRITE-ARRAY 0 label$) (+ 1 segment)))
             (tv:choose-variable-values
               '((*length ,list1 :number)
                 (*diameter ,list2 :number)))
             (aset *length DENDRITE-ARRAY segment length$)
             (aset *diameter DENDRITE-ARRAY segment diameter$)))))
  (update-dendrite DENDRITE-ARRAY))

;;; MENU-FOR-DENDRITE-PASSIVE-COMPONENTS
(defun menu-for-dendrite-passive-components ()
```

```
  (tv:choose-variable-values
    '((*axonal-cap-mem   "axon membrane capacitance [microfarads/sq-cm]   " :number)
      (*axonal-r-mem  "axon membrane resistance [ohm-cm-cm] " :number)
      (*axonal-r-int "axon axoplasm resistance [ohm-cm] " :number)
      (*capd-mem   "dendrite membrane capacitance [microfarads/sq-cm]   " :number)
      (*rd-mem  "dendrite membrane resistance [ohm-cm-cm] " :number)
      (*rd-int "dendrite axoplasm resistance [ohm-cm] " :number)
      (*ed-l "dendritic leak potential [mv] " :number)
      (*plot-voltages-solid "Plot all the voltages in solid lines" :boolean))
    ':label    "Passive Properties of dendrite Segments")
  (dolist (DENDRITE-ARRAY (list BASAL-DENDRITE APICAL-SHAFT-DENDRITE APICAL-1-DENDRITE APICAL-2-DENDRITE))
    (if (aref DENDRITE-ARRAY 0 include-me$)
        (do ((segment 0 (incf segment)))
            ((= segment (aref DENDRITE-ARRAY 0 total-segments$)))
          (update-dendrite DENDRITE-ARRAY)))))

;;; UPDATE-DENDRITE Updates dendrite structure before run with new parameters.
(defun update-dendrite (DENDRITE-ARRAY)
  (let ((total-segments (aref DENDRITE-ARRAY 0 total-segments$))
        (rdmem (if (eq DENDRITE-ARRAY "BASAL-DENDRITE") *rd-mem *rd-mem))
        (rdint (if (eq DENDRITE-ARRAY "BASAL-DENDRITE") *rd-int *rd-int))
        (capdmem (if (eq DENDRITE-ARRAY "BASAL-DENDRITE") *axonal-cap-mem *capd-mem)))
    (do ((segment 0 (incf segment)))
        ((= segment total-segments))
      (aset (// (* (aref DENDRITE-ARRAY segment length$) 3.14159
                   (aref DENDRITE-ARRAY segment diameter$)
                                1.0e-2)
                rdmem)
            DENDRITE-ARRAY segment
            g-leak$)
      (aset (* (aref DENDRITE-ARRAY segment length$) 3.14159
               (aref DENDRITE-ARRAY segment diameter$)
               capdmem 1.0e-5)
            DENDRITE-ARRAY segment capacitance$)
      (aset (// (* 3.14159
                   (* 0.5 (aref DENDRITE-ARRAY segment diameter$))
                   (* 0.5 (aref DENDRITE-ARRAY segment diameter$))
                   100.0)
                (* rdint
                   (aref DENDRITE-ARRAY segment length$)))
            DENDRITE-ARRAY segment
            g-axial$))))

;;; MENU-FOR-DENDRITE-CURRENTS Sets up all the dendrite currents for the current run.
(defun menu-for-dendrite-currents (DENDRITE-ARRAY)
  (let ((total-segments (aref DENDRITE-ARRAY 0 total-segments$)))
    (do ((segment 0 (incf segment)))
        ((= segment total-segments))
      (let ((list1 (list (list 'Na   "Na current in this segment"
                               (list (list :include (aref DENDRITE-ARRAY segment include-na$)) :modify))
                         (list 'Ca   "Ca current in this segment"
                               (list (list :include (aref DENDRITE-ARRAY segment include-ca$)) :modify)))))
        (let ((result (tv:multiple-choose
                        (format nil "Currents in ~A dendrite segment~2d" (aref DENDRITE-ARRAY 0 label$)
                                (+ 1 segment))
                        list1
                        '((:include "Include" nil nil nil (:modify))
                          (:modify "Modify" (:include) nil)))))
          (loop for item in result
                do (progn (if (not (memq :include item))
                              (selectq (car item)
                                (Na (aset nil DENDRITE-ARRAY segment include-na$))
                                (Ca (aset nil DENDRITE-ARRAY segment include-Ca$))))
                          (if (memq :modify item)
                              (selectq (car item)
                                (Na (menu-for-Nad-current DENDRITE-ARRAY segment))
                                (Ca (menu-for-Cad-current DENDRITE-ARRAY segment))))
                          (if (memq :include item)
                              (selectq (car item)
                                (Na (aset t DENDRITE-ARRAY segment include-na$))
                                (Ca (aset t DENDRITE-ARRAY segment include-Ca$)))))))))))

;;; MENU-FOR-SOMA-CURRENTS Sets up all the soma currents for the current run.
(defun menu-for-soma-currents ()
  (let ((flag nil)
        (list1 (list (list 'Na1  "Na1 (trigger mutha) current" (list (list :include  *include-na1) :modify))
                     (list 'Na2  "Na2 (slow tail) current" (list (list :include  *include-na2) :modify))
                     (list 'Na3  "Na3 (repetitive) current" (list (list :include  *include-na3) :modify))
                     (list 'nap   "Nap current" (list (list :include  *include-nap) :modify))
```

```
                    (list 'ca   "Ca current" (list (list :include  *include-ca) :modify))
                    (list 'cas   "Slow Ca current" (list (list :include  *include-cas) :modify))
                    (list 'dr   "DR current" (list (list :include  *include-dr) :modify))
                    (list 'c   "C current" (list (list :include  *include-c) :modify))
                    (list 'ahp   "Ahp current" (list (list :include  *include-ahp) :modify))
                    (list 'm   "M current" (list (list :include  *include-m) :modify))
                    (list 'q   "Q current" (list (list :include  *include-q) :modify))
                    (list 'a   "A current" (list (list :include  *include-a) :modify)))))
    (let ((result (tv:multiple-choose "Pyramidal Currents"
                                      list1
                                      '((:include "Include" nil nil nil (:modify))
                                        (:modify "Modify" (:include) nil)))))
      (loop for item in result
            do (progn (if (not (memq :include item))
                          (selectq (car item)
                            (Na1 (setq *include-na1 nil))
                            (Na2 (setq *include-na2 nil))
                            (Na3 (setq *include-na3 nil))
                            (Nap (setq *include-nap nil))
                            (Ca (setq *include-Ca nil))
                            (Cas (setq *include-Cas nil))
                            (K (setq *include-k nil))
                            (DR (setq *include-DR nil))
                            (C (setq *include-C nil))
                            (AHP (setq *include-AHP nil))
                            (M (setq *include-M nil))
                            (Q (setq *include-Q nil))
                            (A (setq *include-A nil))))
                      (if (memq :modify item)
                          (selectq (car item)
                            (Na1 (and (setq flag t *na1-mod t)(menu-for-Na-current)))
                            (Na2 (and (setq flag t *na2-mod t)(menu-for-Na-current)))
                            (Na3 (and (setq flag t *na3-mod t)(menu-for-Na-current)))
                            (Nap (and (setq flag t *nap-mod t)(menu-for-Nap-current)))
                            (Ca (and (setq flag t *ca-mod t)(menu-for-Ca-current)))
                            (DR (and (setq flag t *dr-mod t)(menu-for-DR-current)))
                            (C (and (setq flag t *c-mod t)(menu-for-C-current)))
                            (AHP (and (setq flag t *ahp-mod t)(menu-for-AHP-current)))
                            (Q (and (setq flag t *q-mod t)(menu-for-Q-current)))
                            (A (and (setq flag t *a-mod t)(menu-for-A-current)))))
                      (if (memq :include item)
                          (selectq (car item)
                            (Na1 (setq *include-na1 T))
                            (Na2 (setq *include-na2 T))
                            (Na3 (setq *include-na3 T))
                            (Nap (setq *include-nap T)) (Ca (setq *include-Ca T))
                            (Cas (setq *include-Cas T))
                            (K (setq *include-k T)) (DR (setq *include-DR T)) (C (setq *include-C T))
                            (AHP (setq *include-AHP T))
                            (M (setq *include-M T))
                            (Q (setq *include-Q T))
                            (A (setq *include-A T)))))))
    (cond (flag  (variable-array-setup)(print "Updated current kinetics")))
    (update-gbars)))



(defun update-gbars ()
  (aset (gbar-na 1 (surf-area *soma-radius)) SOMA gbar-na1$)
  (aset (gbar-na 2 (surf-area *soma-radius)) SOMA gbar-na2$)
  (aset (gbar-na 3 (surf-area *soma-radius)) SOMA gbar-na3$)
  (aset (* *qten-g-32 (gbar-ca (surf-area *soma-radius))) SOMA gbar-ca$)
  (aset *gbar-cas SOMA gbar-cas$)
  (aset (* *qten-g-30 *gbar-a) SOMA gbar-a$)
  (aset *gbar-ahp SOMA gbar-ahp$)
  (aset *gbar-m SOMA gbar-m$)
  (aset *gbar-c SOMA gbar-c$)
  (aset (* *qten-g-30 *gbar-dr) SOMA gbar-dr$)
  (aset *gbar-q SOMA gbar-q$))


(defvar *v)
;;; VARIABLE-ARRAY-SETUP Before the clamp run this fills (or updates) the state variable arrays for the currents that are
;;; currently enabled.
(defun variable-array-setup ()
  (do ((voltage -100.0 (+ voltage 0.1))
       (i 0 (+ i 1)))
      ((= 1500 i))
    (setq *v voltage)
    (aset voltage voltage-array i)
```

```
    (cond-every ((or *update-all-kinetics *na1-mod)
                 (setq *a-m-na1 (a-m-na 1 voltage) *b-m-na1 (b-m-na 1 voltage)
                       *a-h-na1  (a-h-na 1 voltage) *b-h-na1 (b-h-na 1 voltage))
                 (aset (m-na-inf 1) m-na1-inf-array i)
                 (aset (h-na-inf 1) h-na1-inf-array i)
                 (aset (t-m-na 1) t-m-na1-array i)
                 (aset (t-h-na 1) t-h-na1-array i))
                ((or *update-all-kinetics *na2-mod)
                 (setq *a-m-na2 (a-m-na 2 voltage) *b-m-na2 (b-m-na 2 voltage)
                       *a-h-na2  (a-h-na 2 voltage) *b-h-na2 (b-h-na 2 voltage))
                 (aset (m-na-inf 2) m-na2-inf-array i)
                 (aset (h-na-inf 2) h-na2-inf-array i)
                 (aset (t-m-na 2) t-m-na2-array i)
                 (aset (t-h-na 2) t-h-na2-array i))
                ((or *update-all-kinetics *na3-mod)
                 (setq *a-m-na3 (a-m-na 3 voltage) *b-m-na3 (b-m-na 3 voltage)
                       *a-h-na3  (a-h-na 3 voltage) *b-h-na3 (b-h-na 3 voltage))
                 (aset (m-na-inf 3) m-na3-inf-array i)
                 (aset (h-na-inf 3) h-na3-inf-array i)
                 (aset (t-m-na 3) t-m-na3-array i)
                 (aset (t-h-na 3) t-h-na3-array i))
                ((or *update-all-kinetics *nap-mod)
                 (aset (x-nap-inf voltage) x-nap-inf-array i)
                 (aset (t-x-nap voltage) t-x-nap-array i))
                ((or *update-all-kinetics *ca-mod)
                 (aset (s-ca-inf voltage) s-ca-inf-array i)
                 (aset (w-ca-inf voltage) w-ca-inf-array i)
                 (aset (t-s-ca voltage) t-s-ca-array i)
                 (aset (t-w-ca voltage) t-w-ca-array i))
                ((or *update-all-kinetics *cas-mod)
                 (aset (x-cas-inf voltage) x-cas-inf-array i)
                 (aset (t-x-cas voltage) t-x-cas-array i))
                ((or *update-all-kinetics *a-mod)
                 (aset (x-a-inf voltage) x-a-inf-array i)
                 (aset (y-a-inf voltage) y-a-inf-array i)
                 (aset (t-x-a voltage) t-x-a-array i)
                 (aset (t-y-a voltage) t-y-a-array i))
                ((or *update-all-kinetics *ahp-mod)
                 (aset (z-ahp-inf voltage) z-ahp-inf-array i)
                 (aset (t-z-ahp voltage) t-z-ahp-array i)
                 (aset (y-ahp-inf voltage) y-ahp-inf-array i)
                 (aset (t-y-ahp voltage) t-y-ahp-array i))
                ((or *update-all-kinetics *c-mod)
                 (aset (x-c-inf voltage) x-c-inf-array i)
                 (aset (y-c-inf voltage) y-c-inf-array i)
                 (aset (t-x-c voltage) t-x-c-array i)
                 (aset (t-y-c voltage) t-y-c-array i))
                ((or *update-all-kinetics *dr-mod)
                 (aset (x-dr-inf voltage) x-dr-inf-array i)
                 (aset (y-dr-inf voltage) y-dr-inf-array i)
                 (aset (t-x-dr voltage) t-x-dr-array i)
                 (aset (t-y-dr voltage) t-y-dr-array i))
                ((or *update-all-kinetics *m-mod)
                 (aset (x-m-inf voltage) x-m-inf-array i)
                 (aset (t-x-m voltage) t-x-m-array i))
                ((or *update-all-kinetics *q-mod)
                 (aset (x-q-inf voltage) x-q-inf-array i)
                 (aset (t-x-q voltage) t-x-q-array i))))
  (setq *na1-mod nil *na2-mod nil *na3-mod nil *nap-mod nil *ca-mod nil *cas-mod nil *c-mod nil
        *dr-mod nil *m-mod nil *q-mod nil  *a-mod nil *ahp-mod nil))
;;;(+ 1000 (fixr (* 10 voltage)))

(defvar command)
;;; MENU-FOR-SOMA-VOLTAGE-STIMULUS
(defun menu-for-soma-voltage-stimulus  ()
  (setq command (if (= 1 *vclamp-command-flag) "Command array" "Entered steps"))
  (tv:choose-variable-values
    '((command "Voltage clamp by " :choose ("Command array" "Entered steps"))
      (*voltage-command* "Enter name of voltage command array -" :eval-form)
      (*v1 "Step 1 amplitude [mv]" :number)
      (*t-stim-1 "       For how long [ms]" :number)
      (*v2 "Step 2 amplitude [mv]" :number)
      (*t-stim-2 "       For how long [ms]" :number)
      (*v3 "Step 3 amplitude [mv]"  :number)
      (*t-stim-3 "       For how long [ms]" :number)
      (*v4 "Step 4 amplitude [mv]" :number)
      (*t-stim-4 "       For how long [ms]" :number)
      (*v5 "Step 5 amplitude [mv]" :number)
      (*duration "For how long (this will change the duration of the simulation) [ms]" :number))
    ':label "Setting Up Voltage Clamp")
  (setq *vclamp-command-flag (if (equal command "Command array") 1 2)))
```

```
;;; MENU-FOR-SOMA-CURRENT-STIMULUS Sets the current stimulus to the soma.
(defun menu-for-soma-current-stimulus  ()
  (setq command (if (= 1 *iclamp-command-flag) "Command array" "Entered steps"))
  (tv:choose-variable-values
    '((*include-soma-current "Do you want current injected into the soma?" :boolean)
      (command "Current clamp by " :choose ("Command array" "Entered steps"))
      (*current-command* "Enter name of current command array -" :eval-form)
      (*i-stim-1 "Step 1 amplitude [na]" :number)
      (*t-stim-1 "       For how long [ms]" :number)
      (*i-stim-2 "Step 2 amplitude [na]" :number)
      (*t-stim-2 "       For how long [ms]" :number)
      (*i-stim-3 "Step 3 amplitude [na]"  :number)
      (*t-stim-3 "       For how long [ms]" :number)
      (*i-stim-4 "Step 4 amplitude [na]" :number)
      (*t-stim-4 "       For how long [ms]" :number)
      (*i-stim-5 "Step 5 amplitude [na] " :number)
      (*duration "For how long (this will change the duration of the simulation)[ms]" :number))
       ':label "Setting Up Current Clamp")
  (setq *iclamp-command-flag (if (equal command "Command array") 1 2)))


;;; MENU-FOR-DENDRITE-STIMULUS Sets the current stimulus to the dendrite and the segment that is to be injected.
(defun menu-for-dendrite-stimulus  ()
  (tv:choose-variable-values
    '((*include-dendrite-current "Do you want current injected into the  apical dendrite shaft?" :boolean)
      (*stim-seg "Segment to inject current into - " :number)
      (*i-den-stim-1 "Step 1 amplitude [na]" :number)
      (*t-den-stim-1 "       For how long [ms]" :number)
      (*i-den-stim-2 "Step 2 amplitude [na]" :number)
      (*t-den-stim-2 "       For how long [ms]" :number)
      (*i-den-stim-3 "Step 3 amplitude [na]"  :number)
      (*t-den-stim-3 "       For how long [ms]" :number)
      (*i-den-stim-4 "Step 4 amplitude [na]" :number)
      (*t-den-stim-4 "       For how long [ms]" :number)
      (*i-den-stim-5 "Step 5 amplitude [na]" :number)
      (*t-den-stim-5 "       For how long [ms]" :number)
      (*i-den-stim-6 "Step 6 amplitude [na]" :number)
      (*t-den-stim-6 "       For how long [ms]" :number)
      (*i-den-stim-7 "Step 7 amplitude [na]" :number)
      (*t-den-stim-7 "       For how long [ms]" :number)
      (*i-den-stim-8 "Step 8 amplitude [na]"  :number)
      (*t-den-stim-8 "       For how long [ms]" :number)
      (*i-den-stim-9 "Step 9 amplitude [na]" :number)
      (*t-den-stim-9 "       For how long [ms]" :number)
      (*i-den-stim-10 "Step 10 amplitude [na]" :number)
      (*duration "For how long (this will change the duration of the simulation)[ms]" :number)))
  (setq *current-stimulus-segment (- *stim-seg 1)))

;;; MENU-FOR-DENDRITE-SYNAPSE Sets the alpha function for the dendrite synapse.
(defun menu-for-dendrite-synapse ()
  (tv:choose-variable-values
    '((*include-dendrite-synapse "Do you want a synapse on the left apical dendrite branch?" :boolean)
      (*dendrite-synapse-tau "Dendrite alpha synapse time constant [ms] " :number)
      (*dendrite-synapse-amplitude "Dendrite synapse amplitude [micro-S] " :number)
      (*syn-seg "Segment with synapse - " :number)
      (*start-dendrite-synapse "Start the synapse conductance [ms] " :number)
    :label "Setting up synapse input into segment 10"))
  (setq *synapse-segment (- *syn-seg 1))
  (set-up-synapse DENDRITE-SYNAPSE *dendrite-synapse-tau *dendrite-synapse-amplitude))

;;; MENU-FOR-SOMA-SYNAPSE Sets the alpha function for the soma synapse.
(defun menu-for-soma-synapse ()
  (tv:choose-variable-values
    '((*include-soma-synapse "Do you want a soma synapse?" :boolean)
      (*soma-synapse-tau "Soma alpha synapse time constant [ms] " :number)
      (*soma-synapse-amplitude "Soma synapse amplitude [micro-S] " :number)
      (*start-soma-synapse "Start the synapse conductance [ms] " :number)
    :label "Setting up synapse input into soma."))
  (set-up-synapse SOMA-SYNAPSE *soma-synapse-tau *soma-synapse-amplitude))


;;; SET-SOMA-VOLTAGE-STIMULUS Set up the voltage clamp.
(defun set-soma-voltage-stimulus (time-step)
  (setq *clamp-voltage  (if (= 1 *vclamp-command-flag)
                            (aref *voltage-command* *time-step)
                            (cond ((= time-step 0) *v1)
                                  ((= time-step (fixr (// *t-stim-1 *dt))) *v2)
                                  ((= time-step (fixr (// *t-stim-2 *dt))) *v3)
                                  ((= time-step (fixr (// *t-stim-3 *dt))) *v4)
                                  ((= time-step (fixr (// *t-stim-4 *dt))) *v5)
                                  (T *clamp-voltage)))))
```

```
;;; SET-SOMA-CURRENT-STIMULUS Set up the injected current to the soma, if any.
(defun set-soma-current-stimulus ()
;  (let ((time (- *time *time-for-steady-state)))
  (let ((time (- *time 0.0)))
    (if *include-soma-current
        (setq *i-stim  (if (= 1 *iclamp-command-flag)
                           (aref *current-command* *time-step)
                           (cond ((< time *t-stim-1) *i-stim-1)
                                 ((< time *t-stim-2) *i-stim-2)
                                 ((< time *t-stim-3) *i-stim-3)
                                 ((< time *t-stim-4) *i-stim-4)
                                 (t *i-stim-5))))
        (setq *i-stim 0.0))))

;;; SET-DENDRITE-CURRENT-STIMULUS Set up the injected current to the dendrite , if any.
(defun set-dendrite-current-stimulus ()
  (let ((time (- *time *time-for-steady-state)))
    (if *include-dendrite-current
        (setq *i-den-stim (cond ((< time *t-den-stim-1) *i-den-stim-1)
                                ((< time *t-den-stim-2) *i-den-stim-2)
                                ((< time *t-den-stim-3) *i-den-stim-3)
                                ((< time *t-den-stim-4) *i-den-stim-4)
                                ((< time *t-den-stim-5) *i-den-stim-5)
                                ((< time *t-den-stim-6) *i-den-stim-6)
                                ((< time *t-den-stim-7) *i-den-stim-7)
                                ((< time *t-den-stim-8) *i-den-stim-8)
                                ((< time *t-den-stim-9) *i-den-stim-9)
                                (t *i-den-stim-10)))
        (setq *i-den-stim 0.0))))

;;; SET-UP-SYNAPSE Fills the array argument with an alpha function that is set by the time-constant and amplitude
;;; arguments. The array is 10000 dt's long.
(defun set-up-synapse (array tau amplitude)
  (dotimes (i 10000)
    (if (< (// i (- (// tau *dt))) -10.0) (aset 0.0 array i)
        (aset (* i *dt amplitude (exp (// i (- (// tau *dt))))) array i))))


;;; INITIALIZE-SOMA-VOLTAGE Sets the initial soma voltage to the holding voltage.
(defun initialize-soma-voltage ()
  (aset *e-holding SOMA voltage$)
  (aset *e-holding SOMA voltage-est1$)
  (aset *e-holding SOMA voltage-est2$))

;;; INITIALIZE-SOMA-STATES sets all the state variables under the assumption that the system is at steady state,
;;; with the membrane voltage at the holding voltage (now this is *e-holding). For example, the activation and
;;; inactivation variables are set to their "inf" values for the holding potential, and the derivatives are set
;;; to 0. Once the state variables are set, the derivative of the membrane voltage is calculated using the
;;; circuit equation.
(defun initialize-soma-states ()
  (setq *soma-synapse-step 0)
  (let* ((g-na1)(g-na2)(g-na3)
         (g-a)(g-m)(g-dr)(g-nap)(g-ca)(g-cas)(g-ahp)
         (g-q)(g-total)(e-eff)(left-voltage)(right-voltage-1)(right-voltage-2)
         (this-voltage)(g-shunt)
         (g-c)(g-l)(g-coupling-left)(g-coupling-right-1)(g-coupling-right-2)
         (voltage (aref SOMA voltage$))(voltage-index (+ 1000 (fixr (* 10 voltage)))))
    ;;First set all the states to their steady state value at the holding potential, and their derivatives to 0.
    ;;Note that all the variables will be initialized even if their respective currents have not been enabled.
    (aset *ca-conc-shell1-rest SOMA ca-conc-shell$)
    (aset 0.0 SOMA ca-conc-shell-dot$)
    (aset *ca-conc-shell2-rest SOMA ca-conc-shell2$)
    (aset 0.0 SOMA ca-conc-shell2-dot$)
    (aset (aref m-na1-inf-array voltage-index) SOMA m-na1$)
    (aset (aref h-na1-inf-array voltage-index) SOMA h-na1$)
    (aset 0.0 SOMA m-na1-dot$)
    (aset 0.0 SOMA h-na1-dot$)                    ;Fast Na 1 current (Trig)
    (aset (aref m-na2-inf-array voltage-index) SOMA m-na2$)
    (aset (aref h-na2-inf-array voltage-index) SOMA h-na2$)
    (aset 0.0 SOMA m-na2-dot$)
    (aset 0.0 SOMA h-na2-dot$)                    ;Fast Na 2 current (Tail)
    (aset (aref m-na3-inf-array voltage-index) SOMA m-na3$)
    (aset (aref h-na3-inf-array voltage-index) SOMA h-na3$)
    (aset 0.0 SOMA m-na3-dot$)
    (aset 0.0 SOMA h-na3-dot$)                    ;Fast Na 3 current (Rep)
    (aset (aref x-a-inf-array voltage-index) SOMA x-a$)
    (aset (aref y-a-inf-array voltage-index) SOMA y-a$)
    (aset 0.0 SOMA x-a-dot$)
    (aset 0.0 SOMA y-a-dot$)                      ;A current
    (aset (aref x-dr-inf-array voltage-index) SOMA x-dr$)
    (aset (aref y-dr-inf-array voltage-index) SOMA y-dr$)
    (aset 0.0 SOMA x-dr-dot$)
```

```
(aset 0.0 SOMA y-dr-dot$)                    ;DR current
(aset (aref x-m-inf-array voltage-index) SOMA x-m$)
(aset 0.0 SOMA x-m-dot$)                     ;M current
(aset (aref z-ahp-inf-array voltage-index) SOMA z-ahp$)
(aset 0.0 SOMA z-ahp-dot$)
(aset (aref y-ahp-inf-array voltage-index) SOMA y-ahp$)
(aset 0.0 SOMA y-ahp-dot$)
(aset (w-ahp-inf (aref SOMA ca-conc-shell$)) SOMA  w-ahp$)
(aset 0.0 SOMA w-ahp-dot$)                   ;AHP current
(aset (w-c-inf (aref SOMA ca-conc-shell$)) SOMA  w-c$)
(aset 0.0 SOMA w-c-dot$)
(aset (aref x-c-inf-array voltage-index) SOMA x-c$)
(aset 0.0 SOMA x-c-dot$)
(aset (aref y-c-inf-array voltage-index) SOMA y-c$)
(aset 0.0 SOMA y-c-dot$)                     ;C current
(aset (aref x-q-inf-array voltage-index) SOMA x-q$)
(aset 0.0 SOMA x-q-dot$)                     ;Q current
(aset (aref x-nap-inf-array voltage-index) SOMA x-nap$)
(aset 0.0 SOMA x-nap-dot$)                   ;Nap current
(aset (aref x-cas-inf-array voltage-index) SOMA x-cas$)
(aset 0.0 SOMA x-cas-dot$)                   ;Cas current
(aset (aref s-ca-inf-array voltage-index) SOMA s-ca$)
(aset (aref w-ca-inf-array voltage-index) SOMA w-ca$)
(aset 0.0 SOMA s-ca-dot$)
(aset 0.0 SOMA w-ca-dot$)                    ;Ca current
(aset *caps SOMA capacitance$)
(aset 0.0 SOMA g-synapse$)
;;Now calculate the conductances based on the state variable values. If a given current has been disabled,
;;then the appropriate conductance is set to 0.
(setq g-na1 (if *include-na1 (g-na1 (aref SOMA gbar-na1$)(aref SOMA m-na1$)(aref SOMA h-na1$))
                0.0)
      g-na2 (if *include-na2 (g-na2 (aref SOMA gbar-na2$)(aref SOMA m-na2$)(aref SOMA h-na2$))
                0.0)
      g-na3 (if *include-na3 (g-na3 (aref SOMA gbar-na3$)(aref SOMA m-na3$)(aref SOMA h-na3$))
                0.0)



      g-c (if *include-c (g-c (aref SOMA gbar-c$) (aref SOMA x-c$) (aref SOMA y-c$)(aref SOMA w-c$)) 0.0)
      g-ahp (if *include-ahp (g-ahp *gbar-ahp (aref SOMA z-ahp$)(aref SOMA y-ahp$)(aref SOMA w-ahp$)) 0.0)

      g-m (if *include-m (* (aref SOMA gbar-m$) (aref SOMA x-m$)(aref SOMA x-m$)) 0.0)
      g-a (if *include-a (g-a (aref SOMA gbar-a$) (aref SOMA x-a$)(aref SOMA y-a$)) 0.0)
      g-dr (if *include-dr (g-dr (aref SOMA gbar-dr$) (aref SOMA x-dr$)(aref SOMA y-dr$)) 0.0)
      g-cas (if *include-cas (* (aref SOMA gbar-cas$)
                                (aref SOMA x-cas$)) 0.0)

      g-ca (if *include-ca (g-ca (aref SOMA gbar-ca$) (aref SOMA s-ca$)(aref SOMA w-ca$)) 0.0)

      g-nap (if *include-nap (* (aref SOMA gbar-nap$) (aref SOMA x-nap$)) 0.0)
      g-q (if *include-q (* (aref SOMA gbar-q$) (aref SOMA x-q$)(aref SOMA x-q$)) 0.0)
      g-l (aref SOMA g-leak$)
      g-shunt (if *include-shunt *g-electrode 0.0)
      g-total (+ g-na1 g-na2 g-na3 g-a g-nap g-cas g-m g-dr g-c g-ahp g-l g-shunt)
      e-eff (// (+ (* (+ g-nap g-na1 g-na2 g-na3) *e-na)
                   (* g-cas *e-cas) (* g-ca (e-ca ))
                   (* (+ g-a g-m g-c  g-ahp) *e-k)
                   (* g-l *e-l)
                   (* g-dr *e-dr))
                g-total)
      ;;Now calculate the dertvative of the voltage based on the steady-state values for the state variables at
      ;;the holding potential.
      left-voltage (if (aref BASAL-DENDRITE 0 include-me$)
                       (aref BASAL-DENDRITE 1 voltage$)
                       (aref SOMA voltage$))
      g-coupling-left (if (aref BASAL-DENDRITE 0 include-me$)
                          (g-parallel (aref BASAL-DENDRITE 1 g-axial$) (aref SOMA g-axial$))
                          0.0)
      this-voltage (aref SOMA voltage$)
      right-voltage-1 (if (aref APICAL-SHAFT-DENDRITE 0 include-me$)
                          (aref APICAL-SHAFT-DENDRITE 1 voltage$)
                          (aref SOMA voltage$))
      g-coupling-right-1 (if (aref APICAL-SHAFT-DENDRITE 0 include-me$)
                             (g-parallel (aref APICAL-SHAFT-DENDRITE 1 g-axial$)
                                         (aref SOMA g-axial$))
                             0.0)
      right-voltage-2 (aref SOMA voltage$)
      g-coupling-right-2 0.0)
;;"V-DOT" gets the dertvative of the voltage. Then store it as the last permanent value.
(aset (v-dot g-coupling-left g-coupling-right-1 g-coupling-right-2 g-total
             e-eff left-voltage this-voltage right-voltage-1 right-voltage-2
             *caps (+ *i-constant-injection *i-stim))
```

```
        SOMA voltage-dot$)))

;;; INITIALIZE-DENDRITE-VOLTAGES Sets the initial dendrite segment voltages to the dendrite leak potential.
(defun initialize-dendrite-voltages ()
  (dolist (DENDRITE-ARRAY (list BASAL-DENDRITE APICAL-SHAFT-DENDRITE APICAL-1-DENDRITE APICAL-2-DENDRITE))
    (do ((segment 0 (incf segment)))
        ((= segment (aref DENDRITE-ARRAY 0 total-segments$)))
      (aset *ed-l DENDRITE-ARRAY segment voltage$)
      (aset *ed-l DENDRITE-ARRAY segment voltage-est1$)
      (aset *ed-l DENDRITE-ARRAY segment voltage-est2$))))

;;; INITIALIZE-DENDRITE-STATES sets all the state variables under the assumption that the system is at steady
;;; state, with the membrane voltages at the holding voltage (now this is *ed-l). For example, the activation
;;; and inactivation variables are set to their "inf" values for the holding potential, and the derivatives are
;;; set to 0. Once the state variables are set, the derivatives of the membrane voltages are calculated using
;;; the circuit equation.
(defun initialize-dendrite-states ()
  (dolist (DENDRITE-ARRAY (list BASAL-DENDRITE APICAL-SHAFT-DENDRITE APICAL-1-DENDRITE APICAL-2-DENDRITE))
    (let ((*dendrite-synapse-step 0)(e-eff)(g-total)(g-na)(g-ca)(g-l)
          (i-stim)(total-segments (aref DENDRITE-ARRAY 0 total-segments$)))
      (if (aref DENDRITE-ARRAY 0 include-na$)    ; make sure this part even exists
          (do ((segment 0 (incf segment)))
              ((= segment total-segments))
            (let* ((voltage (aref DENDRITE-ARRAY segment voltage$))
                   (voltage-index (+ 1000 (fixr (* 10 voltage))))
                   (include-na (aref DENDRITE-ARRAY segment include-na$))
                   (include-ca (aref DENDRITE-ARRAY segment include-ca$)))
              ;;If a current for the segment has been enabled, then initialize its state variables.
              (cond-every
                (include-na
                  (aset (aref m-na1-inf-array voltage-index) DENDRITE-ARRAY segment m-na1$)
                  (aset (aref h-na1-inf-array voltage-index) DENDRITE-ARRAY segment h-na1$)
                  (aset 0.0 DENDRITE-ARRAY segment m-na1-dot$)
                  (aset 0.0 DENDRITE-ARRAY segment h-na1-dot$))
                (include-ca
                  (aset (aref s-ca-inf-array voltage-index) DENDRITE-ARRAY segment s-ca$)
                  (aset (aref w-ca-inf-array voltage-index) DENDRITE-ARRAY segment w-ca$)
                  (aset 0.0 DENDRITE-ARRAY segment s-ca-dot$)
                  (aset 0.0 DENDRITE-ARRAY segment w-ca-dot$)))
              ;;Now calculate the conductances according to the appropriate state variables. If a given current has not
              ;;been included in the segment, then its conductance is set to 0.
              (setq g-na (if include-na
                             (* (aref DENDRITE-ARRAY segment gbar-na1$) (aref DENDRITE-ARRAY segment m-na1$)
                                (aref DENDRITE-ARRAY segment m-na1$)(aref DENDRITE-ARRAY segment h-na1$)) 0.0)
                    g-ca (if include-ca
                             (g-ca (aref DENDRITE-ARRAY segment gbar-ca$) (aref DENDRITE-ARRAY segment s-ca$)
                                   (aref DENDRITE-ARRAY segment w-ca$)) 0.0)
                    g-l (aref DENDRITE-ARRAY segment g-leak$)
                    i-stim (if (and (EQ DENDRITE-ARRAY APICAL-SHAFT-DENDRITE)
                                    (= segment *current-stimulus-segment))      ;Include current injection
                               *i-den-stim 0.0)
                    g-total (+ g-na g-ca g-l)
                    e-eff (// (+ (* g-na *e-na) (* g-ca *e-ca ) (* g-l *ed-l))
                              g-total))
              ;;Now calculate the derivative of the voltage based on the steady-state values for the state variables at
              ;;the holding potential. Note that the arguments to "V-DOT" have to be adjusted for the distal and
              ;;the proximal dendritic segment.
              (aset (dendrite-derivative DENDRITE-ARRAY total-segments segment i-stim g-total e-eff)
                    DENDRITE-ARRAY segment voltage-dot$)))))))

;;; DENDRITE-DERIVATIVE This function finds the derivative of the dendrite segment voltage given the current estimated
;;; state.
(defun dendrite-derivative (DENDRITE-ARRAY total-segments segment i-stim g-total e-eff)
  (let ((this-voltage (aref DENDRITE-ARRAY segment voltage-est1$))
        (left-voltage)(right-voltage-1)(right-voltage-2)
        (g-coupling-left)(g-coupling-right-1)(g-coupling-right-2))
    (cond ((= total-segments 1)                    ; Just one segment in this part of the tree
           (if (or (EQ DENDRITE-ARRAY APICAL-1-DENDRITE) (EQ DENDRITE-ARRAY APICAL-2-DENDRITE)) ; l or r branch
                                                   ; is abutting the distal part of the shaft
               (setq left-voltage (AREF APICAL-SHAFT-DENDRITE (- *total-apical-shaft-segments 1) voltage-est1$)
                     g-coupling-left (g-parallel
                                      (aref APICAL-SHAFT-DENDRITE (- *total-apical-shaft-segments 1) g-axial$)
                                      (aref DENDRITE-ARRAY 0 g-axial$)))
                                                   ; otherwise this is the apical or basal shaft, which abuts the soma
               (setq left-voltage (aref SOMA voltage-est1$)
                     g-coupling-left (g-parallel (aref SOMA g-axial$) (aref DENDRITE-ARRAY 0 g-axial$))))
           (if (and (EQ DENDRITE-ARRAY APICAL-SHAFT-DENDRITE) *include-apical-1)
                                                   ; shaft with l branch gets r-v-l as l
               (setq right-voltage-1 (aref APICAL-1-DENDRITE 0 voltage-est1$)
                     g-coupling-right-1 (g-parallel (aref APICAL-1-DENDRITE 0 g-axial$)
```

```
                                                    (aref DENDRITE-ARRAY 0 g-axial$)))
                                                ; else no segments distal
                    (setq right-voltage-1 (aref DENDRITE-ARRAY segment voltage-est1$)
                          g-coupling-right-1 0.0))
                   (if (and (EQ DENDRITE-ARRAY APICAL-SHAFT-DENDRITE) *include-apical-2)
                                                ; shaft with r branch gets r-v-2 as r
                    (setq right-voltage-2 (aref APICAL-2-DENDRITE 0 voltage-est1$)
                          g-coupling-right-2 (g-parallel (aref APICAL-2-DENDRITE 0 g-axial$)
                                                         (aref DENDRITE-ARRAY 0 g-axial$)))
                                                ; else no segments distal
                    (setq right-voltage-2 (aref DENDRITE-ARRAY segment voltage-est1$)
                          g-coupling-right-2 0.0)))
                  ((= total-segments (+ 1 segment))     ; At the distal segment of this part of the dendrite, and there is more
                                                ; than one segment, so l will be segment proximal in same part of tree
                   (setq left-voltage (aref DENDRITE-ARRAY (- segment 1) voltage-est1$)
                         g-coupling-left (g-parallel (aref DENDRITE-ARRAY segment g-axial$)
                                                     (aref DENDRITE-ARRAY (- segment 1) g-axial$)))
                   (if (and (EQ DENDRITE-ARRAY APICAL-SHAFT-DENDRITE) *include-apical-1)
                                                ; if shaft w◊l branch then r-v-1 will be proximal seg of l
                    (setq right-voltage-1 (aref APICAL-1-DENDRITE 0 voltage-est1$)
                          g-coupling-right-1 (g-parallel (aref APICAL-1-DENDRITE 0 g-axial$)
                                                         (aref DENDRITE-ARRAY segment g-axial$)))
                                                ; else no r-v-1, no matter which part of tree
                    (setq right-voltage-1 (aref DENDRITE-ARRAY segment voltage-est1$)
                          g-coupling-right-1 0.0))
                   (if (and (EQ DENDRITE-ARRAY APICAL-SHAFT-DENDRITE) *include-apical-2)
                                                ; if shaft w◊l branch then r-v-1 will be proximal seg of l
                    (setq right-voltage-2 (aref APICAL-2-DENDRITE 0 voltage-est1$)
                          g-coupling-right-2  (g-parallel (aref APICAL-2-DENDRITE 0 g-axial$)
                                                          (aref DENDRITE-ARRAY segment g-axial$)))
                                                ; else no r-v-2, no matter which part of tree
                    (setq right-voltage-2 (aref DENDRITE-ARRAY segment voltage-est1$)
                          g-coupling-right-2 0.0)))
                  ((= segment 0)                ; At the proximal segment of part of the tree with at least 2 segments
                   (if (or (EQ DENDRITE-ARRAY APICAL-1-DENDRITE) (EQ DENDRITE-ARRAY APICAL-2-DENDRITE))
                                                ; prox seg of either l or r branch will get distal shaft as l-v
                    (setq left-voltage (AREF APICAL-SHAFT-DENDRITE (- *total-apical-shaft-segments 1) voltage-est1$)
                          g-coupling-left (g-parallel
                                           (aref APICAL-SHAFT-DENDRITE (- *total-apical-shaft-segments 1) g-axial$)
                                           (aref DENDRITE-ARRAY segment g-axial$)))
                                                ; prox seg of apical or basal shaft will get soma as l-v
                    (setq left-voltage (aref SOMA voltage-est1$)
                          g-coupling-left (g-parallel (aref SOMA g-axial$) (aref DENDRITE-ARRAY 1 g-axial$))))
                                                ; prox seg of all parts of tree w◊ at least 2 segs will get seg 2 from same
                                                ; part as r-v-1, no r-v-2
                   (setq right-voltage-1 (aref DENDRITE-ARRAY 1 voltage-est1$)
                         g-coupling-right-1 (g-parallel (aref DENDRITE-ARRAY 1 g-axial$)
                                                        (aref DENDRITE-ARRAY 1 g-axial$))
                         right-voltage-2 (aref DENDRITE-ARRAY 0 voltage-est1$)
                         g-coupling-right-2 0.0))
                  (t                            ; At some middle seg of some part of the tree that has at least 3 segs
                   (setq left-voltage (aref DENDRITE-ARRAY (- segment 1) voltage-est1$) ;l-v is seg prox from same part of tree
                         g-coupling-left (g-parallel (aref DENDRITE-ARRAY (- segment 1) g-axial$)
                                                     (aref DENDRITE-ARRAY segment g-axial$))
                         right-voltage-1 (aref DENDRITE-ARRAY (+ 1 segment) voltage-est1$)      ;r-v-1 is seg distal from same pa
rt
                         g-coupling-right-1  (g-parallel (aref DENDRITE-ARRAY (+ 1 segment) g-axial$)
                                                         (aref DENDRITE-ARRAY segment g-axial$))
                         right-voltage-2 (aref DENDRITE-ARRAY segment voltage-est1$)    ;no r-v-2
                         g-coupling-right-2 0.0)))
    ;;"V-DOT" gets the derivative of the segment voltage. Then store it as the last permanent value.
    (v-dot g-coupling-left g-coupling-right-1 g-coupling-right-2 g-total
           e-eff left-voltage this-voltage right-voltage-1 right-voltage-2
           (aref DENDRITE-ARRAY segment capacitance$) i-stim)))



;;; LOAD-FIRST-ESTIMATES starts the estimation loop with an initial "latest" estimate (est2) of the voltages,
;;; calculated from the previous voltage, the previous derivative of the voltage, and the time step.
;;; The initial "est2" is calculated using an open integration formula.
(defun load-first-estimates ()
  ;;First get all the dendrite segments.
  (dolist (DENDRITE-ARRAY (list BASAL-DENDRITE APICAL-SHAFT-DENDRITE APICAL-1-DENDRITE APICAL-2-DENDRITE))
    (if (aref DENDRITE-ARRAY 0 include-me$)
        (do ((segment 0 (incf segment)))
            ((= segment (aref DENDRITE-ARRAY 0 total-segments$)))
          (aset (+ (aref DENDRITE-ARRAY segment voltage$)       ; Estimated voltage = Previous volt. +
                                                ; (dt * Previous-volt.-dot)
                   (* *dt (aref DENDRITE-ARRAY segment voltage-dot$)))
                DENDRITE-ARRAY segment voltage-est2$))))
```

```
  ;;Now for the soma.
  (cond ((or *steady-state-run *iclamp-run)
         (aset (+ (aref SOMA voltage$) (* *dt (aref SOMA voltage-dot$))) SOMA voltage-est2$))
        ;; For the voltage clamp the soma voltage is always the current clamp voltage.
        (*vclamp-run
         (aset (if (= 1 *vclamp-command-flag) (aref *voltage-command* *time-step) *clamp-voltage)
               SOMA voltage-est1$)
         (aset (if (= 1 *vclamp-command-flag) (aref *voltage-command* *time-step) *clamp-voltage)
               SOMA voltage-est2$))))

(defvar flag t)

;;; TEST-ESTIMATES - If ANY of the estimated voltages are not within the convergence criterium, then immediately
;;; exit with "nil", otherwise return "t".
(defun test-estimates ()
  (setq flag t)
  (and (test (aref SOMA voltage-est1$)
             (aref SOMA voltage-est2$))
       (and (dolist (DENDRITE-ARRAY
                     (list BASAL-DENDRITE APICAL-SHAFT-DENDRITE APICAL-1-DENDRITE APICAL-2-DENDRITE "end"))
              (if (equal DENDRITE-ARRAY "end") (return T))
              (if (not (if (aref DENDRITE-ARRAY 0 INCLUDE-ME$)
                           (do ((segment 0 (incf segment)))
                               ((= segment (aref DENDRITE-ARRAY 0 total-segments$)) T)
                             (if (not (test (aref DENDRITE-ARRAY segment voltage-est1$)
                                            (aref DENDRITE-ARRAY segment voltage-est2$)))
                                 (return nil)))
                           T))
                  (return nil))))))


;;; TEST-FOR-RESTING-STATE Checks to see if the compartment voltages have settled down to a quasi-resting state by
;;; testing all the stored derivatives of the compartment voltages. If the magnitude of ALL the derivatives are
;;; less than *dot-epsilon, then return "t". The system is then assumed to be at steady state.
(defun test-for-resting-state ()
  (setq flag t)
  (and (< (abs (aref SOMA voltage-dot$)) *dot-epsilon)
       (or
         (dolist (DENDRITE-ARRAY
                  (list BASAL-DENDRITE APICAL-SHAFT-DENDRITE APICAL-1-DENDRITE APICAL-2-DENDRITE))
           (if (aref DENDRITE-ARRAY 0 INCLUDE-ME$)
               (do ((segment 0 (incf segment)))
                   ((= segment (aref DENDRITE-ARRAY 0 total-segments$)))
                 (if (not
                       (< (abs (aref DENDRITE-ARRAY segment voltage-dot$)) *dot-epsilon))
                     (setq flag nil)))))
         flag)))

;;; G-PARALLEL
(defun g-parallel (g1 g2)
  (// (* g1 g2 4.0) (+ (* 2.0 g1)(* 2.0 g2))))

;;; STORE-NEW-SOMA-ESTIMATE Make the old soma voltage estimate (est1) equal to the new voltage estimate (est2).
(defun store-new-soma-estimate ()
  (aset (* 0.5 (+ (aref SOMA voltage-est1$)(aref SOMA voltage-est2$))) SOMA voltage-est1$))


;;; STORE-NEW-DENDRITE-ESTIMATES Make the old dendrite voltage estimate (est1) equal to the new voltage estimate
;;; (est2).
(defun store-new-dendrite-estimates ()
  (dolist (DENDRITE-ARRAY (list BASAL-DENDRITE APICAL-SHAFT-DENDRITE APICAL-1-DENDRITE APICAL-2-DENDRITE))
    (if (aref DENDRITE-ARRAY 0 INCLUDE-ME$)
        (do ((segment 0 (incf segment)))
            ((= segment (aref DENDRITE-ARRAY 0 total-segments$)))
          (aset (* 0.5 (+ (aref DENDRITE-ARRAY segment voltage-est1$)
                          (aref DENDRITE-ARRAY segment voltage-est2$)))
                DENDRITE-ARRAY segment voltage-est1$)))))

;;; STORE-NEW-SOMA-VOLTAGE Make the old soma voltage equal to the new voltage estimate (est2).
(defun store-new-soma-voltage ()
  (aset (aref SOMA voltage-est2$) SOMA voltage$))


;;; STORE-NEW-DENDRITE-VOLTAGES Make the old dendrite voltage equal to the new voltage estimate (est2).
(defun store-new-dendrite-voltages ()
  (dolist (DENDRITE-ARRAY (list BASAL-DENDRITE APICAL-SHAFT-DENDRITE APICAL-1-DENDRITE APICAL-2-DENDRITE))
    (if (aref DENDRITE-ARRAY 0 INCLUDE-ME$)
        (do ((segment 0 (incf segment)))
            ((= segment (aref DENDRITE-ARRAY 0 total-segments$)))
          (aset (aref DENDRITE-ARRAY segment voltage-est2$) DENDRITE-ARRAY segment voltage$)))))
```

```
;;; SET-DENDRITE-STATES-AND-V-DOTS Calculates the conductances based on the current voltage estimate (est1).
;;; Based on these values the derivative of the voltage (est1-dot) is calculated according to KCL. If the
;;; loop-done flag is set, then the derivative of the voltage, the state variables and their derivatives are
;;; stored as the final estimates (voltage-dot$, m-na$, m-na-dot$, etc.).
(defun set-dendrite-states-and-v-dots (loop-done)
  (dolist (DENDRITE-ARRAY (list BASAL-DENDRITE APICAL-SHAFT-DENDRITE APICAL-1-DENDRITE APICAL-2-DENDRITE))
    (if (aref DENDRITE-ARRAY 0 INCLUDE-ME$)
        (let ((total-segments (aref DENDRITE-ARRAY 0 total-segments$)))
          (do ((segment 0 (incf segment)))
              ((= segment total-segments))
            (let* ((v-est (aref DENDRITE-ARRAY segment voltage-est1$))  ;"V-EST" is the current voltage estimate
                   (voltage-index (+ 1000 (fixr (* 10 v-est))))
                   (include-na (aref DENDRITE-ARRAY segment include-na$))
                   (include-ca (aref DENDRITE-ARRAY segment include-ca$))
                   (m-na-est)(h-na-est)
                   (s-ca-est)(w-ca-est)
                   (e-eff)(g-na-est)(g-ca-est)(g-synapse)(i-stim)(g-leak)(g-total-est))
              (if include-na
                  (setq m-na-est (trap-approx (aref DENDRITE-ARRAY segment m-na1$)(aref DENDRITE-ARRAY segment m-na
1-dot$)
                                              (aref m-na1-inf-array voltage-index) (aref t-m-na1-array voltage-inde
x))
                        h-na-est (trap-approx (aref DENDRITE-ARRAY segment h-na1$)(aref DENDRITE-ARRAY segment h-na
1-dot$)
                                              (aref h-na1-inf-array voltage-index) (aref t-h-na1-array voltage-inde
x))
                        g-na-est (* (aref DENDRITE-ARRAY segment gbar-na1$) m-na-est m-na-est h-na-est))
                  (setq g-na-est 0.0))
              (if include-ca
                  (setq s-ca-est (trap-approx (aref DENDRITE-ARRAY segment s-ca$)(aref DENDRITE-ARRAY segment s-ca-
dot$)
                                              (aref s-ca-inf-array voltage-index) (aref t-s-ca-array voltage-index)
)
                        w-ca-est (trap-approx (aref DENDRITE-ARRAY segment w-ca$)(aref DENDRITE-ARRAY segment w-ca-
dot$)
                                              (aref w-ca-inf-array voltage-index) (aref t-w-ca-array voltage-index)
)
                        g-ca-est (g-ca (aref DENDRITE-ARRAY segment gbar-ca$)  s-ca-est w-ca-est))
                  (setq g-ca-est 0.0))
              (if (and *include-dendrite-synapse (= segment *synapse-segment)   ;Include synapse
                       (and (> *time *start-dendrite-synapse)   ;if segment has a synapse and time is right.
                            (< *dendrite-synapse-step 10000)))  ;Total length of synapse is 10000 dt's
                  (setq g-synapse (aref DENDRITE-SYNAPSE *dendrite-synapse-step)
                        *dendrite-synapse-step (+ *dendrite-synapse-step 1))
                  (setq g-synapse 0.0))
              (setq i-stim (if (and (EQ DENDRITE-ARRAY APICAL-SHAFT-DENDRITE)
                                    (= segment *current-stimulus-segment))      ;Include current injection
                               *i-den-stim 0.0)
                    g-leak (aref DENDRITE-ARRAY segment g-leak$)
                    g-total-est (+ g-na-est g-ca-est g-leak g-synapse)
                    e-eff (// (+ (* g-na-est *e-na)(* g-ca-est *e-ca )
                                 (* g-leak *ed-1)(* g-synapse *e-synapse))
                              g-total-est))
              ;;If still in the loop, then voltage derivative is just an estimate, otherwise it is stored.
              (cond ((not loop-done)
                     (aset (dendrite-derivative DENDRITE-ARRAY total-segments segment i-stim g-total-est e-eff)
                           DENDRITE-ARRAY segment  voltage-est1-dot$))
                    (t
                     (aset (dendrite-derivative DENDRITE-ARRAY total-segments segment i-stim g-total-est e-eff)
                           DENDRITE-ARRAY segment  voltage-dot$)
                     ;;Likewise, store the final values for the state variables and their derivatives after the loop
                     ;;Has finished.
                     (cond-every
                       (include-na
                         (aset m-na-est DENDRITE-ARRAY segment m-na1$)
                         (aset (dxdt-eq m-na-est (aref m-na1-inf-array voltage-index)(aref t-m-na1-array voltage-in
dex))
                               DENDRITE-ARRAY segment m-na1-dot$)
                         (aset h-na-est DENDRITE-ARRAY segment h-na1$)
                         (aset (dxdt-eq h-na-est (aref h-na1-inf-array voltage-index)(aref t-h-na1-array voltage-in
dex))
                               DENDRITE-ARRAY segment h-na1-dot$))
                       (include-ca
                         (aset s-ca-est DENDRITE-ARRAY segment s-ca$)
                         (aset (dxdt-eq s-ca-est (aref s-ca-inf-array voltage-index)(aref t-s-ca-array voltage-inde
x))
                               DENDRITE-ARRAY segment s-ca-dot$)
                         (aset w-ca-est DENDRITE-ARRAY segment w-ca$)
                         (aset (dxdt-eq w-ca-est (aref w-ca-inf-array voltage-index)(aref t-w-ca-array voltage-inde
x))
                               DENDRITE-ARRAY segment
                               w-ca-dot$))))))))))))
```

```
;;; SET-SOMA-STATES-AND-V-DOT-FOR-CURRENT-CLAMP Calculates the conductances based on the current voltage estimate.
;;; Based on these values the derivative of the voltage (...est1-dot) is calculated according to KCL. If the
;;; loop-done flag is set, then the derivative of the voltage, the state variables and their derivatives are
;;; stored as the final estimates (voltage-dot$, m-na$, m-na-dot$, etc.).
;;;
(defun set-soma-states-and-v-dot-for-current-clamp (loop-done)
  (let* ((v (aref SOMA voltage-est1$))          ;"V" is the current voltage estimate
         (m-na1-est)(h-na1-est)(g-na1-est)
         (m-na2-est)(h-na2-est)(g-na2-est)
         (m-na3-est)(h-na3-est)(g-na3-est)
         (x-nap-est)(g-nap-est)
         (s-ca-est)(w-ca-est)(g-ca-est)(x-cas-est)(g-cas-est)(new-ca-conc-shell)(new-ca-conc-shell2)
         (x-a-est)(y-a-est)(g-a-est)
         (x-c-est)(y-c-est)(w-c-est)(g-c-est)
         (z-ahp-est)     (y-ahp-est) (w-ahp-est)
         (g-ahp-est)
         (x-m-est)(g-m-est)
         (x-dr-est)(y-dr-est)(g-dr-est)
         (x-q-est)(g-q-est)
         (g-leak)(g-shunt)(g-coupling-left)(g-coupling-right-1)(g-coupling-right-2)
         (e-eff)(left-voltage)(this-voltage)(right-voltage-1)(right-voltage-2)
         (g-synapse)(g-total-est)(voltage-index (+ 1000 (fixr (* 10 v)))))
    (cond-every
      (*include-na1 (setq m-na1-est (trap-approx (aref SOMA m-na1$) (aref SOMA m-na1-dot$)
                                                 (aref m-na1-inf-array voltage-index) (aref t-m-na1-array voltage-i
ndex))
                          h-na1-est (trap-approx (aref SOMA h-na1$) (aref SOMA h-na1-dot$)
                                                 (aref h-na1-inf-array voltage-index) (aref t-h-na1-array voltage-i
ndex))
                          g-na1-est (g-na1 (aref SOMA gbar-na1$) m-na1-est h-na1-est)))
      ((not *include-na1) (setq g-na1-est 0.0))
      (*include-na2 (setq m-na2-est (trap-approx (aref SOMA m-na2$) (aref SOMA m-na2-dot$)
                                                 (aref m-na2-inf-array voltage-index) (aref t-m-na2-array voltage-i
ndex))
                          h-na2-est (trap-approx (aref SOMA h-na2$) (aref SOMA h-na2-dot$)
                                                 (aref h-na2-inf-array voltage-index) (aref t-h-na2-array voltage-i
ndex))
                          g-na2-est (g-na2 (aref SOMA gbar-na2$)  m-na2-est h-na2-est)))
      ((not *include-na2) (setq g-na2-est 0.0))
      (*include-na3 (setq m-na3-est (trap-approx (aref SOMA m-na3$) (aref SOMA m-na3-dot$)
                                                 (aref m-na3-inf-array voltage-index) (aref t-m-na3-array voltage-i
ndex))
                          h-na3-est (trap-approx (aref SOMA h-na3$) (aref SOMA h-na3-dot$)
                                                 (aref h-na3-inf-array voltage-index) (aref t-h-na3-array voltage-i
ndex))
                          g-na3-est (g-na3 (aref SOMA gbar-na3$) m-na3-est  h-na3-est )))
      ((not *include-na3) (setq g-na3-est 0.0))
      (*include-nap (setq x-nap-est (trap-approx (aref SOMA x-nap$) (aref SOMA x-nap-dot$)
                                                 (aref x-nap-inf-array voltage-index) (aref t-x-nap-array voltage-i
ndex))
                          g-nap-est (* (aref SOMA gbar-nap$) x-nap-est )))
      ((not *include-nap) (setq g-nap-est 0.0))
      (*include-cas (setq x-cas-est (trap-approx (aref SOMA x-cas$) (aref SOMA x-cas-dot$)
                                                 (aref x-cas-inf-array voltage-index) (aref t-x-cas-array voltage-i
ndex))
                          g-cas-est (* (aref SOMA gbar-cas$) x-cas-est)))
      ((not *include-cas) (setq g-cas-est 0.0))
      (*include-ca  (setq s-ca-est (trap-approx (aref SOMA s-ca$) (aref SOMA s-ca-dot$)
                                                (aref s-ca-inf-array voltage-index) (aref t-s-ca-array voltage-inde
x))
                          w-ca-est (trap-approx (aref SOMA w-ca$) (aref SOMA w-ca-dot$)
                                                (aref w-ca-inf-array voltage-index) (aref t-w-ca-array voltage-inde
x))
                          g-ca-est (g-ca (aref SOMA gbar-ca$) s-ca-est w-ca-est)))
      ((not *include-ca) (setq g-ca-est 0.0))
      (*include-a (setq x-a-est (trap-approx (aref SOMA x-a$) (aref SOMA x-a-dot$) (aref x-a-inf-array voltage-inde
x)
                                             (aref t-x-a-array voltage-index))
                        y-a-est (trap-approx (aref SOMA y-a$) (aref SOMA y-a-dot$) (aref y-a-inf-array voltage-inde
x)
                                             (aref t-y-a-array voltage-index))
                        g-a-est (g-a (aref SOMA gbar-a$) x-a-est y-a-est)))
      ((not *include-a) (setq g-a-est 0.0))
      (*include-c (setq x-c-est (trap-approx (aref SOMA x-c$) (aref SOMA x-c-dot$) (aref x-c-inf-array voltage-inde
x)
                                             (aref t-x-c-array voltage-index))
                        y-c-est (trap-approx (aref SOMA y-c$) (aref SOMA y-c-dot$) (aref y-c-inf-array voltage-inde
x)
                                             (aref t-y-c-array voltage-index))
                        w-c-est (trap-approx (aref SOMA w-c$) (aref SOMA w-c-dot$)
                                             (w-c-inf (aref SOMA ca-conc-shell$))
                                             (t-w-c (aref SOMA ca-conc-shell$)))
```

```
                    g-c-est  (g-c (aref SOMA gbar-c$) x-c-est y-c-est w-c-est)))
     ((not *include-c) (setq g-c-est 0.0))
     (*include-ahp (setq z-ahp-est (trap-approx (aref SOMA z-ahp$) (aref SOMA z-ahp-dot$) (aref z-ahp-inf-array vo
ltage-index)
                                                (aref t-z-ahp-array voltage-index))
                         y-ahp-est (trap-approx (aref SOMA y-ahp$) (aref SOMA y-ahp-dot$) (aref y-ahp-inf-array vo
ltage-index)
                                                (aref t-y-ahp-array voltage-index))
                         w-ahp-est (trap-approx (aref SOMA w-ahp$) (aref SOMA w-ahp-dot$)
                                                (w-ahp-inf (aref SOMA ca-conc-shell2$))
                                                (t-w-ahp (aref SOMA ca-conc-shell2$)))
                         g-ahp-est  (g-ahp *gbar-ahp z-ahp-est y-ahp-est w-ahp-est)))
     ((not *include-ahp) (setq g-ahp-est 0.0))

     (*include-m (setq x-m-est (trap-approx (aref SOMA x-m$) (aref SOMA x-m-dot$) (aref x-m-inf-array voltage-inde
x)
                                            (aref t-x-m-array voltage-index))
                       g-m-est (* (aref SOMA gbar-m$) x-m-est)))
     ((not *include-m) (setq g-m-est 0.0))
     (*include-dr (setq x-dr-est (trap-approx (aref SOMA x-dr$) (aref SOMA x-dr-dot$)
                                              (aref x-dr-inf-array voltage-index) (aref t-x-dr-array voltage-index
))
                        y-dr-est (trap-approx (aref SOMA y-dr$) (aref SOMA y-dr-dot$)
                                              (aref y-dr-inf-array voltage-index) (aref t-y-dr-array voltage-index
))
                       g-dr-est (g-dr (aref SOMA gbar-dr$) x-dr-est y-dr-est)))
     ((not *include-dr) (setq g-dr-est 0.0))
     (*include-q (setq x-q-est (trap-approx (aref SOMA x-q$) (aref SOMA x-q-dot$) (aref x-q-inf-array voltage-inde
x)
                                            (aref t-x-q-array voltage-index))
                       g-q-est (* (aref SOMA gbar-q$) x-q-est)))
     ((not *include-q) (setq g-q-est 0.0))
     ((and *include-soma-synapse              ;Include soma synapse if enabled,
           (and (> *time *start-soma-synapse)  ;and if time is right.
                (< *soma-synapse-step 10000))) ;Total length of synapse is 10000 dt's
      (setq g-synapse (aref SOMA-SYNAPSE *soma-synapse-step))
      (aset g-synapse SOMA g-synapse$)
      (setq *soma-synapse-step (+ *soma-synapse-step 1)))
     (t (setq g-synapse 0.0)
        (aset g-synapse SOMA g-synapse$)))
   (setq g-leak (aref SOMA g-leak$)
         g-shunt (if *include-shunt *g-electrode 0.0)
         g-total-est (+ g-na1-est g-na2-est g-na3-est
                        g-ca-est g-nap-est g-a-est g-c-est g-ahp-est g-dr-est g-m-est
                        g-q-est g-leak g-synapse g-shunt)
         e-eff (// (+ (* (+ g-na1-est g-na2-est g-na3-est g-nap-est) *e-na)
                      (* (+ g-a-est  g-c-est g-m-est g-q-est g-ahp-est) *e-k)
                      (* g-ca-est (e-ca))
                      (* g-leak *e-l)
                      (* g-dr-est *e-dr)
                      (* g-synapse *e-synapse))
                   g-total-est)
         left-voltage (if (aref BASAL-DENDRITE 0 include-ms$)
                          (aref BASAL-DENDRITE 0 voltage-est1$)
                          (aref SOMA voltage-est1$))
         g-coupling-left (if (aref BASAL-DENDRITE 0 include-ms$)
                             (g-parallel (aref BASAL-DENDRITE 0 g-axial$)
                                         (aref SOMA g-axial$))
                             0.0)
         this-voltage (aref SOMA voltage-est1$)
         right-voltage-1 (if (aref APICAL-SHAFT-DENDRITE 0 include-ms$)
                             (aref APICAL-SHAFT-DENDRITE 0 voltage-est1$)
                             (aref SOMA voltage-est1$))
         g-coupling-right-1 (if (aref APICAL-SHAFT-DENDRITE 0 include-ms$)
                                (g-parallel (aref APICAL-SHAFT-DENDRITE 0 g-axial$)
                                            (aref SOMA g-axial$))
                                0.0)
         right-voltage-2 (aref SOMA voltage-est1$)      ; Soma never has a second righthand connection.
         g-coupling-right-2 0.0)


   ;;"V-DOT" gets the derivative of the voltage. Then store it as the last permanent value.
   (cond
     ;;If still in the loop, then voltage derivative is just an estimate
     ((not loop-done)
      (aset (v-dot g-coupling-left g-coupling-right-1 g-coupling-right-2 g-total-est
                   e-eff left-voltage this-voltage right-voltage-1 right-voltage-2
                   *caps (+ *i-constant-injection *i-stim))
            SOMA voltage-est1-dot$))
     ;;Otherwise, calculated derivative becomes the stored derivative.
```

```
(t
 (aset (v-dot g-coupling-left g-coupling-right-1 g-coupling-right-2 g-total-est
              e-eff left-voltage this-voltage right-voltage-1 right-voltage-2
              *caps (+ *i-constant-injection *i-stim))
       SOMA voltage-dot$)
 ;;Likewise, all the state variables and their derivatives are calculated according to the current
 ;;voltage estimate, and stored.
 (cond-every
   (T
     (setq new-ca-conc-shell (+ (* *dt (aref SOMA ca-conc-shell-dot$))(aref SOMA ca-conc-shell$)))
     (if (< new-ca-conc-shell *core-conc) (setq new-ca-conc-shell *core-conc))
     (setq new-ca-conc-shell2 (+ (* *dt (aref SOMA ca-conc-shell2-dot$))(aref SOMA ca-conc-shell2$)))
     (if (< new-ca-conc-shell2 *core-conc) (setq new-ca-conc-shell2 *core-conc))
     (aset new-ca-conc-shell SOMA ca-conc-shell$)
     (aset new-ca-conc-shell2 SOMA ca-conc-shell2$)
     (aset (ca-conc-shell-dot
             (+ (* g-ca-est (- v (e-ca)))
                (* g-cas-est (- v *e-cas)))
            new-ca-conc-shell new-ca-conc-shell2)
           SOMA ca-conc-shell-dot$)
     (aset (ca-conc-shell2-dot  new-ca-conc-shell new-ca-conc-shell2)
           SOMA ca-conc-shell2-dot$))
   (*include-na1
     (aset m-na1-est SOMA m-na1$)
     (aset (dxdt-eq m-na1-est (aref m-na1-inf-array voltage-index) (aref t-m-na1-array voltage-index))
           SOMA m-na1-dot$)
     (aset h-na1-est SOMA h-na1$)
     (aset (dxdt-eq h-na1-est (aref h-na1-inf-array voltage-index) (aref t-h-na1-array voltage-index))
           SOMA h-na1-dot$))
   (*include-na2
     (aset m-na2-est SOMA m-na2$)
     (aset (dxdt-eq m-na2-est (aref m-na2-inf-array voltage-index) (aref t-m-na2-array voltage-index))
           SOMA m-na2-dot$)
     (aset h-na2-est SOMA h-na2$)
     (aset (dxdt-eq h-na2-est (aref h-na2-inf-array voltage-index) (aref t-h-na2-array voltage-index))
           SOMA h-na2-dot$))
   (*include-na3
     (aset m-na3-est SOMA m-na3$)
     (aset (dxdt-eq m-na3-est (aref m-na3-inf-array voltage-index) (aref t-m-na3-array voltage-index))
           SOMA m-na3-dot$)
     (aset h-na3-est SOMA h-na3$)
     (aset (dxdt-eq h-na3-est (aref h-na3-inf-array voltage-index) (aref t-h-na3-array voltage-index))
           SOMA h-na3-dot$))
   (*include-a
     (aset x-a-est SOMA x-a$)
     (aset (dxdt-eq x-a-est (aref x-a-inf-array voltage-index) (aref t-x-a-array voltage-index))
           SOMA x-a-dot$)
     (aset y-a-est SOMA y-a$)
     (aset (dxdt-eq y-a-est (aref y-a-inf-array voltage-index) (aref t-y-a-array voltage-index))
           SOMA y-a-dot$))
   (*include-c
     (aset x-c-est SOMA x-c$)
     (aset (dxdt-eq x-c-est (aref x-c-inf-array voltage-index) (aref t-x-c-array voltage-index))
           SOMA x-c-dot$)
     (aset y-c-est SOMA y-c$)
     (aset (dxdt-eq y-c-est (aref y-c-inf-array voltage-index) (aref t-y-c-array voltage-index))
           SOMA y-c-dot$)
     (aset w-c-est SOMA w-c$)
     (aset (dxdt-eq w-c-est (w-c-inf (aref SOMA ca-conc-shell$))(t-w-c  (aref SOMA ca-conc-shell$)))
           SOMA w-c-dot$))

   (*include-ahp
     (aset z-ahp-est SOMA z-ahp$)
     (aset (dxdt-eq z-ahp-est (aref z-ahp-inf-array voltage-index) (aref t-z-ahp-array voltage-index))
           SOMA z-ahp-dot$)
     (aset y-ahp-est SOMA y-ahp$)
     (aset (dxdt-eq y-ahp-est (aref y-ahp-inf-array voltage-index) (aref t-y-ahp-array voltage-index))
           SOMA y-ahp-dot$)
     (aset w-ahp-est SOMA w-ahp$)
     (aset (dxdt-eq w-ahp-est (w-ahp-inf (aref SOMA ca-conc-shell2$))(t-w-ahp  (aref SOMA ca-conc-shell2$)))
           SOMA w-ahp-dot$))

   (*include-m
     (aset x-m-est SOMA x-m$)
     (aset (dxdt-eq x-m-est (aref x-m-inf-array voltage-index) (aref t-x-m-array voltage-index))
           SOMA x-m-dot$))
   (*include-q
     (aset x-q-est SOMA x-q$)
     (aset (dxdt-eq x-q-est (aref x-q-inf-array voltage-index) (aref t-x-q-array voltage-index))
           SOMA x-q-dot$))
   (*include-dr
     (aset x-dr-est SOMA x-dr$)
```

```
          (aset (dxdt-eq x-dr-est (aref x-dr-inf-array voltage-index) (aref t-x-dr-array voltage-index))
                SOMA x-dr-dot$)
          (aset y-dr-est SOMA y-dr$)
          (aset (dxdt-eq y-dr-est (aref y-dr-inf-array voltage-index) (aref t-y-dr-array voltage-index))
                SOMA y-dr-dot$))
        (*include-ca
          (aset s-ca-est SOMA s-ca$)
          (aset (dxdt-eq s-ca-est (aref s-ca-inf-array voltage-index) (aref t-s-ca-array voltage-index))
                SOMA s-ca-dot$)
          (aset w-ca-est SOMA w-ca$)
          (aset (dxdt-eq w-ca-est (aref w-ca-inf-array voltage-index) (aref t-w-ca-array voltage-index))
                SOMA w-ca-dot$))
        (*include-cas
          (aset x-cas-est SOMA x-cas$)
          (aset (dxdt-eq x-cas-est (aref x-cas-inf-array voltage-index) (aref t-x-cas-array voltage-index))
                SOMA x-cas-dot$))
        (*include-nap
          (aset x-nap-est SOMA x-nap$)
          (aset (dxdt-eq x-nap-est (aref x-nap-inf-array voltage-index) (aref t-x-nap-array voltage-index))
                SOMA x-nap-dot$)))))))



;;; SET-SOMA-STATES-AND-V-DOT-FOR-VOLTAGE-CLAMP
(defun set-soma-states-and-v-dot-for-voltage-clamp ()
  (let* ((new-ca-conc-shell)(new-ca-conc-shell2)
         (s-ca)(w-ca)(x-cas)
         (voltage-index (+ 1000 (fixr (* 10.0 *clamp-voltage)))))
    (cond ((or *vstep (= 1 *vclamp-command-flag))
           (cond-every

             (*include-ahp (setq *zahpinf (aref z-ahp-inf-array voltage-index)
                                 *tzahp (aref t-z-ahp-array voltage-index)
                                 *yahpinf (aref y-ahp-inf-array voltage-index)
                                 *tyahp (aref t-y-ahp-array voltage-index)
                                 *wahpinf (w-ahp-inf (aref SOMA ca-conc-shell2$))
                                 *twahp (t-w-ahp  (aref SOMA ca-conc-shell2$))))
             (*include-c (setq *xcinf (aref x-c-inf-array voltage-index)
                               *txc (aref t-x-c-array voltage-index)
                               *ycinf (aref y-c-inf-array voltage-index)
                               *tyc (aref t-y-c-array voltage-index)
                               *wcinf (w-c-inf (aref SOMA ca-conc-shell$))
                               *twc (t-w-c  (aref SOMA ca-conc-shell$))))
             (*include-q (setq *xqinf (aref x-q-inf-array voltage-index) *tq (aref t-x-q-array voltage-index)))
             (*include-m (setq *xminf (aref x-m-inf-array voltage-index) *tm (aref t-x-m-array voltage-index)))
             (*include-na1 (setq *mna1inf (aref m-na1-inf-array voltage-index) *tmna1 (aref t-m-na1-array voltage-i
ndex)
                                 *hna1inf (aref h-na1-inf-array voltage-index) *thna1 (aref t-h-na1-array voltage-i
ndex)))
             (*include-na2 (setq *mna2inf (aref m-na2-inf-array voltage-index) *tmna2 (aref t-m-na2-array voltage-i
ndex)
                                 *hna2inf (aref h-na2-inf-array voltage-index) *thna2 (aref t-h-na2-array voltage-i
ndex)))
             (*include-na3 (setq *mna3inf (aref m-na3-inf-array voltage-index) *tmna3 (aref t-m-na3-array voltage-i
ndex)
                                 *hna3inf (aref h-na3-inf-array voltage-index) *thna3 (aref t-h-na3-array voltage-i
ndex)))
             (*include-ca (setq *mcainf (aref s-ca-inf-array voltage-index) *tmca (aref t-s-ca-array voltage-index)
                                *hcainf (aref w-ca-inf-array voltage-index) *thca (aref t-w-ca-array voltage-index)
))
             (*include-nap (setq *xnapinf (aref x-nap-inf-array voltage-index) *txnap (aref t-x-nap-array voltage-i
ndex)))
             (*include-cas (setq *xcasinf (aref x-cas-inf-array voltage-index) *txcas (aref t-x-cas-array voltage-i
ndex)))
             (*include-dr (setq *xdrinf (aref x-dr-inf-array voltage-index) *txdr (aref t-x-dr-array voltage-index)
                                *ydrinf (aref y-dr-inf-array voltage-index) *tydr (aref t-y-dr-array voltage-index)
))
             (*include-a (setq *xainf (aref x-a-inf-array voltage-index) *txa  (aref t-x-a-array voltage-index)
                               *yainf (aref y-a-inf-array voltage-index) *tya (aref t-y-a-array voltage-index))))
            (aset (// (- *clamp-voltage (aref SOMA voltage$)) *dt) SOMA voltage-dot$))
           (T (aset 0.0 SOMA voltage-dot$)))
      (cond-every
        (*include-cas (setq x-cas (vclamp-new-x *dt (aref SOMA x-cas$) *xcasinf *txcas))
                      (aset x-cas SOMA x-cas$))
        ((not *include-cas) (setq x-cas 0))
        (*include-ca (setq s-ca (vclamp-new-x *dt (aref SOMA s-ca$) *mcainf *tmca)
                           w-ca (vclamp-new-x *dt (aref SOMA w-ca$) *hcainf *thca))
                     (aset s-ca SOMA s-ca$) (aset w-ca SOMA w-ca$))
        ((not *include-ca) (setq s-ca 0 w-ca 0))
        (T
          (setq new-ca-conc-shell (+ (* *dt (aref SOMA ca-conc-shell-dot$))(aref SOMA ca-conc-shell$)))
```

```
      (aset new-ca-conc-shell SOMA ca-conc-shell$)
      (setq new-ca-conc-shell2 (+ (* *dt (aref SOMA ca-conc-shell2-dot$))(aref SOMA ca-conc-shell2$)))
      (aset new-ca-conc-shell2 SOMA ca-conc-shell2$)
      (aset (ca-conc-shell-dot (+ (* *gbar-ca s-ca s-ca w-ca (- *clamp-voltage (e-ca)))
                                  (* *gbar-cas x-cas (- *clamp-voltage *e-cas))) new-ca-conc-shell new-ca-conc-she
112)
            SOMA ca-conc-shell-dot$)
      (aset (ca-conc-shell2-dot new-ca-conc-shell new-ca-conc-shell2)
            SOMA ca-conc-shell2-dot$))

    (*include-ahp (aset (vclamp-new-x *dt (aref SOMA z-ahp$) *zahpinf *tzahp) SOMA z-ahp$)
                  (aset (vclamp-new-x *dt (aref SOMA y-ahp$) *yahpinf *tyahp) SOMA y-ahp$)
                  (aset (vclamp-new-x *dt (aref SOMA w-ahp$) *wahpinf *twahp) SOMA w-ahp$))
    (*include-c (aset (vclamp-new-x *dt (aref SOMA x-c$) *xcinf *txc) SOMA x-c$)
                  (aset (vclamp-new-x *dt (aref SOMA y-c$) *ycinf *tyc) SOMA y-c$)
                  (aset (vclamp-new-x *dt (aref SOMA w-c$) *wcinf *twc) SOMA w-c$))
    (*include-q  (aset (vclamp-new-x *dt (aref SOMA x-q$) *xqinf *tq) SOMA x-q$))
    (*include-m  (aset (vclamp-new-x *dt (aref SOMA x-m$) *xminf *tm) SOMA x-m$))
    (*include-nap  (aset (vclamp-new-x *dt (aref SOMA x-nap$) *xnapinf *txnap) SOMA x-nap$))
    (*include-dr  (aset (vclamp-new-x *dt (aref SOMA x-dr$) *xdrinf *txdr) SOMA x-dr$)
                  (aset (vclamp-new-x *dt (aref SOMA y-dr$) *ydrinf *tydr) SOMA  y-dr$))
    (*include-a  (aset (vclamp-new-x *dt (aref SOMA x-a$) *xainf *txa) SOMA x-a$)
                 (aset (vclamp-new-x *dt (aref SOMA y-a$) *yainf *tya) SOMA y-a$))
    (*include-na1  (aset (vclamp-new-x *dt (aref SOMA m-na1$) *mna1inf *tmna1) SOMA m-na1$)
                   (aset (vclamp-new-x *dt (aref SOMA h-na1$) *hna1inf *thna1) SOMA h-na1$))
    (*include-na2  (aset (vclamp-new-x *dt (aref SOMA m-na2$) *mna2inf *tmna2) SOMA m-na2$)
                   (aset (vclamp-new-x *dt (aref SOMA h-na2$) *hna2inf *thna2) SOMA h-na2$))
    (*include-na3  (aset (vclamp-new-x *dt (aref SOMA m-na3$) *mna3inf *tmna3) SOMA m-na3$)
                   (aset (vclamp-new-x *dt (aref SOMA h-na3$) *hna3inf *thna3) SOMA h-na3$)))))

;     (T (setq new-ca-conc-shell (+ (* *dt (aref SOMA ca-conc-shell-dot$))(aref SOMA ca-conc-shell$)))
;       (aset new-ca-conc-shell SOMA ca-conc-shell$)
```

;;; ESTIMATE-SOMA-VOLTAGE Gets and stores new estimate (...est2$) of soma voltage using the previous voltage
;;; (voltage$), the previous derivative (voltage-dot$), and the current estimate of the derivative
;;; (...est1-dot$). The function "APPROX-X" is used for the trapezoidal approximation. Note that this function is
;;; only required by the current clamp protocol.

```
;;;
(defun estimate-soma-voltage ()
  (aset
    (approx-x (aref SOMA voltage$)
              (aref SOMA voltage-dot$)
              (aref SOMA voltage-est1-dot$))
    SOMA voltage-est2$))
```

;;; ESTIMATE-DENDRITE-VOLTAGES Gets and stores new estimate (...est2) of dendrite *voltages using the previous*
;;; voltage (voltage$), the previous derivative (voltage-dot$), and the current estimate of the derivative
;;; (...est1-dot$). The function "APPROX-X" is used for the trapezoidal approximation.

```
(defun estimate-dendrite-voltages ()
  (dolist (DENDRITE-ARRAY (list BASAL-DENDRITE APICAL-SHAFT-DENDRITE APICAL-1-DENDRITE APICAL-2-DENDRITE))
    (if (aref DENDRITE-ARRAY 0 INCLUDE-ME$)
        (do ((segment 0 (incf segment)))
            ((= segment (aref DENDRITE-ARRAY 0 total-segments$)))
          (aset (approx-x (aref DENDRITE-ARRAY segment voltage$)
                          (aref DENDRITE-ARRAY segment voltage-dot$)
                          (aref DENDRITE-ARRAY segment voltage-est1-dot$))
                DENDRITE-ARRAY segment voltage-est2$)))))
```

;;; UPDATE-OUTPUT-LISTS Stores the latest values in the enabled lists.

```
(defun update-output-lists ()
  (let ((soma-voltage (aref SOMA voltage$)))
    (update-soma-lists soma-voltage)
    (if *include-dendrite (update-dendrite-lists soma-voltage))
    (if *vclamp-run (update-vclamp-list soma-voltage))))
```

;;; UPDATE-DENDRITE-LISTS

```
(defun update-dendrite-lists (soma-voltage)
  (update-coupling-current-list soma-voltage)
  (aset *i-den-stim *dendrite-stim-current* *point-index)
  (cond-every
    (*include-dendrite-synapse (update-dendrite-synapse-current-list))
    ((aref APICAL-SHAFT-DENDRITE 0 include-me$)
     (cond-every
       ((aref APICAL-SHAFT-DENDRITE 0 plot-me$)
        (aset (aref APICAL-SHAFT-DENDRITE 0 voltage$) *asivoltage* *point-index))
```

```
      ((and (aref APICAL-SHAFT-DENDRITE 2 plot-me$) (>= (aref APICAL-SHAFT-DENDRITE 0 total-segments$) 3))
       (aset (aref APICAL-SHAFT-DENDRITE 2 voltage$) *as3voltage* *point-index))
      ((and (aref APICAL-SHAFT-DENDRITE 4 plot-me$) (>= (aref APICAL-SHAFT-DENDRITE 0 total-segments$) 5))
       (aset (aref APICAL-SHAFT-DENDRITE 4 voltage$) *as5voltage* *point-index))
      ((and (aref APICAL-SHAFT-DENDRITE 9 plot-me$) (>= (aref APICAL-SHAFT-DENDRITE 0 total-segments$) 10))
       (aset (aref APICAL-SHAFT-DENDRITE 9 voltage$) *as10voltage* *point-index))))
   ((aref APICAL-1-DENDRITE 0 include-me$)
    (cond-every
      ((aref APICAL-1-DENDRITE 0 plot-me$)
       (aset (aref APICAL-1-DENDRITE 0 voltage$) *al1voltage* *point-index))
      ((and (aref APICAL-1-DENDRITE 3 plot-me$) (>= (aref APICAL-1-DENDRITE 0 total-segments$) 4))
       (aset (aref APICAL-1-DENDRITE 3 voltage$) *al4voltage* *point-index))))
   ((aref APICAL-2-DENDRITE 0 include-me$)
    (cond-every
      ((aref APICAL-2-DENDRITE 0 plot-me$)
       (aset (aref APICAL-2-DENDRITE 0 voltage$) *ar1voltage* *point-index))
      ((and (aref APICAL-2-DENDRITE 3 plot-me$) (>= (aref APICAL-2-DENDRITE 0 total-segments$) 4))
       (aset (aref APICAL-2-DENDRITE 3 voltage$) *ar4voltage* *point-index))))
   ((aref BASAL-DENDRITE 0 include-me$)
    (cond-every
      ((aref BASAL-DENDRITE 0 plot-me$)
       (aset (aref BASAL-DENDRITE 0 voltage$) *b1voltage* *point-index))
      ((and (aref BASAL-DENDRITE 3 plot-me$) (>= (aref BASAL-DENDRITE 0 total-segments$) 4))
       (aset (aref BASAL-DENDRITE 3 voltage$) *b4voltage* *point-index))))
   ((aref APICAL-SHAFT-DENDRITE 0 include-na$)
    (aset (na1-current (gbar-nad (* (aref APICAL-SHAFT-DENDRITE 0 length$)
                                    3.14159e-8 (aref APICAL-SHAFT-DENDRITE 0 diameter$)))
                       (aref APICAL-SHAFT-DENDRITE 0 m-na1$)
                       (aref APICAL-SHAFT-DENDRITE 0 h-na1$)
                       (aref APICAL-SHAFT-DENDRITE 0 voltage$))
          *nad1-current* *point-index))
   ((aref APICAL-SHAFT-DENDRITE 2 include-ca$)
    (aset (ca-current (gbar-cad (* (aref APICAL-SHAFT-DENDRITE 2 length$)
                                   3.14159e-8  (aref APICAL-SHAFT-DENDRITE 2 diameter$)))
                      (aref APICAL-SHAFT-DENDRITE 2 s-ca$)
                      (aref APICAL-SHAFT-DENDRITE 2 w-ca$)
                      (aref APICAL-SHAFT-DENDRITE 2 voltage$))
          *cad1-current* *point-index*))))

;;; UPDATE-COUPLING-CURRENT-LIST
(defun update-coupling-current-list (soma-voltage)
  (aset  (+ (* (if (aref BASAL-DENDRITE 0 include-me$)
                   (g-parallel (aref BASAL-DENDRITE 0 g-axial$)
                               (aref SOMA g-axial$))
                   0.0)
               (- soma-voltage (aref BASAL-DENDRITE 0 voltage$)))
            (* (if (aref APICAL-SHAFT-DENDRITE 0 include-me$)
                   (g-parallel (aref APICAL-SHAFT-DENDRITE 0 g-axial$)
                               (aref SOMA g-axial$))
                   0.0)
               (- soma-voltage (aref APICAL-SHAFT-DENDRITE 0 voltage$))))
         *coupling-current* *point-index*))

;;; UPDATE-DENDRITE-SYNAPSE-CURRENT-LIST
(defun update-dendrite-synapse-current-list ()
  (aset (* (aref APICAL-1-DENDRITE *synapse-segment g-synapse$)
           (- (aref APICAL-1-DENDRITE *synapse-segment voltage$)
              *e-synapse))
        *dendrite-synapse-current* *point-index)
  (aset (aref APICAL-1-DENDRITE *synapse-segment g-synapse$)
        *dendrite-synapse-conductance* *point-index*))

;;; UPDATE-VCLAMP-LIST
(defun update-vclamp-list (soma-voltage)
  (aset (+ (if *include-dr (dr-current (aref SOMA x-dr$)(aref SOMA y-dr$) soma-voltage)
               0)

           (if *include-c (c-current (aref SOMA x-c$)(aref SOMA y-c$)(aref SOMA w-c$)
                                     soma-voltage) 0 )
           (if *include-ahp (ahp-current (aref SOMA z-ahp$)(aref SOMA y-ahp$)(aref SOMA w-ahp$)
                                     soma-voltage) 0 )
           (if *include-q (* (aref SOMA gbar-q$)(aref SOMA x-q$)
                             (- soma-voltage *e-k)) 0.0)
           (if *include-m (m-current (aref SOMA x-m$) soma-voltage) 0 )
           (if *include-a (a-current (aref SOMA x-a$) (aref SOMA y-a$)
                                     soma-voltage) 0 )
           (if *include-na1 (na1-current (aref SOMA gbar-na1$)
                                         (aref SOMA m-na1$)(aref SOMA h-na1$) SOMA-voltage) 0)
           (if *include-na2 (na2-current (aref SOMA gbar-na2$)
                                         (aref SOMA m-na2$)(aref SOMA h-na2$) SOMA-voltage) 0)
           (if *include-na3 (na3-current (aref SOMA gbar-na3$)
```

```
                                    (aref SOMA m-na3$)(aref SOMA h-na3$) SOMA-voltage) 0)
          (if *include-nap (nap-current (aref SOMA gbar-nap$) (aref SOMA x-nap$)
                                        soma-voltage) 0 )
          (if *include-cas (cas-current (aref SOMA x-cas$) soma-voltage) 0 )
          (if *include-ca (ca-current (aref SOMA gbar-ca$)
                                      (aref SOMA s-ca$)
                                      (aref SOMA w-ca$) soma-voltage) 0.0 )
          (+ (* (if (aref BASAL-DENDRITE 0 include-me$)
                    (* 2.0 (aref BASAL-DENDRITE 0 g-axial$))
                    0.0)
                (- soma-voltage (aref BASAL-DENDRITE 0 voltage$)))
             (* (if (aref APICAL-SHAFT-DENDRITE 0 include-me$)
                    (* 2.0 (aref APICAL-SHAFT-DENDRITE 0 g-axial$))
                    0.0)
                (- soma-voltage (aref APICAL-SHAFT-DENDRITE 0 voltage$))))
          (l-current soma-voltage)
          (if *include-shunt (* *g-electrode soma-voltage) 0)
          (* *caps (aref SOMA voltage-dot$)))
        *current* *point-index*))

;;; UPDATE-SOMA-LISTS
(defun update-soma-lists (soma-voltage)
  (cond-every
    (*include-kinetics
      (cond-every
        (*include-na1
          (aset (aref SOMA m-na1$) *m-na1* *point-index)        (aset (aref SOMA h-na1$) *h-na1* *point-index)
          (aset (g-na1 1.0 (aref SOMA m-na1$)(aref SOMA h-na1$)) *g-na1* *point-index))
        (*include-na2
          (aset (aref SOMA m-na2$) *m-na2* *point-index)        (aset (aref SOMA h-na2$) *h-na2* *point-index)
          (aset (g-na2 1.0 (aref SOMA m-na2$)(aref SOMA h-na2$)) *g-na2* *point-index))
        (*include-na3
          (aset (aref SOMA m-na3$) *m-na3* *point-index)        (aset (aref SOMA h-na3$) *h-na3* *point-index)
          (aset (g-na3 1.0 (aref SOMA m-na3$)(aref SOMA h-na3$)) *g-na3* *point-index))
        (*include-ca
          (aset (aref SOMA s-ca$) *s-ca* *point-index) (aset (aref SOMA w-ca$) *w-ca* *point-index)
          (aset (g-ca  1.0 (aref SOMA s-ca$)(aref SOMA w-ca$)) *g-ca* *point-index))
        (*include-dr
          (aset (aref SOMA x-dr$) *x-dr* *point-index)  (aset (aref SOMA y-dr$) *y-dr* *point-index)
          (aset (g-dr 1.0 (aref SOMA x-dr$)(aref SOMA y-dr$)) *g-dr* *point-index))
        (*include-a
          (aset (aref SOMA x-a$) *x-a* *point-index) (aset (aref SOMA y-a$) *y-a* *point-index)
          (aset (g-a  1.0 (aref SOMA x-a$)(aref SOMA y-a$)) *g-a* *point-index))

        (*include-ahp
          (aset (aref SOMA z-ahp$) *z-ahp* *point-index) (aset (aref SOMA y-ahp$) *y-ahp* *point-index)
          (aset (aref SOMA w-ahp$) *w-ahp* *point-index)
          (aset (g-ahp 1.0 (aref SOMA z-ahp$)(aref SOMA y-ahp$)(aref SOMA w-ahp$)) *g-ahp* *point-index))
        (*include-c
          (aset (aref SOMA x-c$) *x-c* *point-index) (aset (aref SOMA y-c$) *y-c* *point-index)
          (aset (aref SOMA w-c$) *w-c* *point-index)
          (aset (g-c 1.0 (aref SOMA x-c$)(aref SOMA y-c$)(aref SOMA w-c$)) *g-c* *point-index))))
    (t
      (aset (* 1.0e-3 *time) *time* *point-index) (aset soma-voltage *voltage* *point-index)
      (aset (l-current soma-voltage) *l-current* *point-index)
      (aset (* *caps (aref SOMA voltage-dot$)) *caps-current* *point-index)
      (aset (+ *i-constant-injection *i-stim) *stim-current* *point-index)
      (aset (e-ca) *e-ca* *point-index)
      (aset (aref SOMA ca-conc-shell$) *ca-conc-shell* *point-index)
      (aset (aref SOMA ca-conc-shell2$) *ca-conc-shell2* *point-index))
    (*include-shunt
      (aset (* *g-electrode soma-voltage) *shunt-current* *point-index))
    (*include-na1
      (aset (na1-current (aref SOMA gbar-na1$)(aref SOMA m-na1$) (aref SOMA h-na1$) SOMA-voltage)
            *na1-current* *point-index))
    (*include-na2
      (aset (na2-current (aref SOMA gbar-na2$)(aref SOMA m-na2$) (aref SOMA h-na2$) SOMA-voltage)
            *na2-current* *point-index))
    (*include-na3
      (aset (na3-current (aref SOMA gbar-na3$)(aref SOMA m-na3$) (aref SOMA h-na3$) SOMA-voltage)
            *na3-current* *point-index))
    (*include-ca
      (aset (ca-current (aref SOMA gbar-ca$)(aref SOMA s-ca$) (aref SOMA w-ca$) soma-voltage)
            *ca-current* *point-index))
    (*include-nap
      (aset (nap-current (aref SOMA gbar-nap$)(aref SOMA x-nap$) soma-voltage)
            *nap-current* *point-index))
    (*include-cas
      (aset (cas-current (aref SOMA x-cas$) soma-voltage)
            *cas-current* *point-index))
    (*include-c
      (aset (c-current (aref SOMA x-c$)(aref SOMA y-c$)(aref SOMA w-c$) soma-voltage)
```

```
        *c-current* *point-index))
    (*include-ahp
      (aset (ahp-current (aref SOMA z-ahp$)(aref SOMA y-ahp$)(aref SOMA w-ahp$) soma-voltage)
        *ahp-current* *point-index))
    (*include-m
      (aset (m-current (aref SOMA x-m$) soma-voltage) *m-current* *point-index))
    (*include-dr
      (aset (dr-current (aref SOMA x-dr$)(aref SOMA y-dr$) soma-voltage) *dr-current* *point-index))
    (*include-a
      (aset (a-current (aref SOMA x-a$) (aref SOMA y-a$) soma-voltage)
        *a-current* *point-index))
    (*include-q
      (aset (* (aref SOMA gbar-q$)(aref SOMA x-q$)(- soma-voltage *e-k))
        *q-current* *point-index))
    (*include-soma-synapse
      (aset (* (aref SOMA g-synapse$)(- soma-voltage *e-synapse))
        *soma-synapse-current* *point-index)
      (aset (aref SOMA g-synapse$) *soma-synapse-conductance* *point-index*))))


;;; V-DOT Gives the derivative of the membrane voltage given the current conductances, the membrane capacitance,
;;; the adjacent voltages, the local voltage, and any injected current.
(defun v-dot (g-coupling-left g-coupling-right-1 g-coupling-right-2 g-membrane driving-voltage
              voltage-left voltage voltage-right-1 voltage-right-2 capacitance i-injected)
  (if (= capacitance 0.0) 0.0
      (// (+ (* g-membrane (- driving-voltage voltage))
             i-injected
             (* g-coupling-left (- voltage-left voltage))
             (* g-coupling-right-1 (- voltage-right-1 voltage))
             (* g-coupling-right-2 (- voltage-right-2 voltage)))
          capacitance)))

;;; TRAP-APPROX Computes the trapezoidal approximation of a state variable that is described by first order
;;; kinetics given the old value of the variable, the old value of its derivative, the new value of its steady
;;; state, and the new value of its time constant.
(defun trap-approx (x-old x-old-dot x-inf-new tau-x-new)
  (// (+ x-old (* (// *dt 2.0) (+ x-old-dot (// x-inf-new tau-x-new))))
      (+ 1.0 (// *dt (* 2.0 tau-x-new)))))

;;; VCLAMP-NEW-X
(defun vclamp-new-x (dt x-old x-inf tau-x)
  (+ x-inf (* (- x-old x-inf) (exp (- (// dt tau-x))))))

;;; APPROX-X Calculates x(n+1) using trapezoidal approximation with the arguments
;;; x(n), x(n)-dot, and x(n+1)-dot.
(defun approx-x (x0 x0-dot x1-dot)
  (+ x0 (* (// *dt 2.0)(+ x0-dot x1-dot))))

;;; DXDT-EQ calculates the derivative of the state variable (x-dot) according to original differential equation
;;;       characterizing the gating variables.
(defun dxdt-eq (x x-inf t-x)
  (// (- x-inf x) t-x))


;;; TEST returns true if difference between arguments is less than "epsilon",
;;;       nil otherwise.
(defun test (x y)
  (cond ((> (abs (- x y)) *epsilon) nil)
        (T T)))


;;; PRINT-PARAMETERS Print all the parameters for the currrent run in the interaction pane.
(defun print-parameters ()
  (send terminal-io :refresh) (send terminal-io :home-cursor) (send terminal-io :set-font-map '(fonts:hl8))
  (format t "Temp. ~2dC,   " *temperature) (format t "Time step [msec]  ~4f,   " *dt )
  (format t "Soma E-Rest  ~4f,   " (aref SOMA e-rest$)) (format t "E-Leak (Soma)  ~4f~%" *e-l)
  (format t "E-Leak (Dendrite)  ~4f,  " *ed-l)
  (format t "E-K  ~4f,   " *e-k) (format t "E-ca  ~4f,   " *e-ca) (format t "E-dr  ~4f,   " *e-dr)
  (format t "E-na  ~4f,  " *e-na) (format t "R-Soma [MOhm]  ~4f ~%" (// 1.0 *gs-l))
  (format t "Soma memb. res. [ohms-sq cm]  ~4f,  " *rs-mem)
  (format t "Soma cap. [nF]  ~4f, " *caps)
  (format t "Soma rad. [mic's]  ~4f  ~%" *soma-radius)
  (format t "Soma Spec cap. [microF/sq-cm]  ~4f, " *caps-mem)
  (format t "Dendrite Spec cap. [microF/sq-cm]  ~4f ~%" *capd-mem)
  (format t "Dendrite memb. res. [ohms-sq cm]  ~4f,   " *rd-mem)
  (format t "Axoplasmic res. [ohms- cm]  ~4f   ~%" *rd-int)
  (if  *include-shunt (format t "Electrode shunt [Mohm]  ~2f ~%" *r-electrode)
       (format t "No electrode shunt ~%"))
  (if  *include-na1 (format t "gNa1 [microS]  ~4f,   " (aref SOMA gbar-na1$))
       (format t "I-na1 poisoned,  ")
```

```
(if  *include-na2 (format t "gNa2 [microS]  ~4f,  " (aref SOMA gbar-na2$))
     (format t "I-na2 poisoned, "))
(if  *include-na3 (format t "gNa3 [microS]  ~4f,  " (aref SOMA gbar-na3$))
     (format t "I-na3 poisoned, "))
(if  *include-ca (format t "gCa [microS]  ~4f~%" (aref SOMA gbar-ca$))
     (format t "I-ca poisoned~%"))
(if  *include-nap (format t "gNaP [microS]  ~4f,   " (aref SOMA gbar-nap$))
     (format t "I-NaP poisoned,  "))
(if  *include-cas (format t "gCas [microS]  ~4f  ~%" (aref SOMA gbar-cas$))
     (format t "I-cas poisoned ~%"))
(if  *include-c (format t "gC [microS]  ~4f," (aref SOMA gbar-c$))
     (format t "I-C poisoned, "))
(if  *include-ahp (format t "gAHP [microS]  ~4f," (aref SOMA gbar-ahp$))
     (format t "I-AHP poisoned, "))
(if  *include-m (format t "gM [microS]  ~4f,   " (aref SOMA gbar-m$))
     (format t "I-M poisoned, "))
(if  *include-dr (format t "gDR [microS]  ~4f,   " (aref SOMA gbar-dr$))
     (format t "I-DR poisoned, "))
(if (and *include-dr (< *dr-block 1.0)) (format t "DR block = ~4f" *dr-block))
(if  *include-a (format t "~%gA [microS]  ~4f~%~%" (aref SOMA gbar-a$))
     (format t "I-A poisoned~%~%"))
(cond-every
  ((aref APICAL-SHAFT-DENDRITE 0 include-me$)
   (let* ((lamda (^ (// (* *rd-mem (// (aref APICAL-SHAFT-DENDRITE 0 diameter$) 2.0) 10000.0)
                        (* 2.0 *rd-int)) 0.5))
          (L (// (* (aref APICAL-SHAFT-DENDRITE 0 total-segments$)
                    (aref APICAL-SHAFT-DENDRITE 0 length$)) lamda)))
     (format t "Apical shaft dendrite with ~2d segments.   Length = ~4f microns.~%"
             (aref APICAL-SHAFT-DENDRITE 0 total-segments$)
             (* (aref APICAL-SHAFT-DENDRITE 0 total-segments$)
                (aref APICAL-SHAFT-DENDRITE 0 length$)))
     (format t "      Segment length = ~4d microns. Diameter = ~2f microns. Lamda = ~4f. L = ~4f~%"
             (aref APICAL-SHAFT-DENDRITE 0 length$)
             (aref APICAL-SHAFT-DENDRITE 0 diameter$)
             lamda
             L)
     (format t "      %Lamda(per segment) = ~4f%"
             (* 100.0 (// (aref APICAL-SHAFT-DENDRITE 0 length$) lamda)) )))
  ((aref APICAL-1-DENDRITE 0 include-me$)
   (let* ((lamda (^ (// (* *rd-mem (// (aref APICAL-1-DENDRITE 0 diameter$) 2.0) 10000.0)
                        (* 2.0 *rd-int)) 0.5))
          (L (// (* (aref APICAL-1-DENDRITE 0 total-segments$)
                    (aref APICAL-1-DENDRITE 0 length$)) lamda)))
     (format t "~%Left apical dendrite branch with ~2d segments.   Length = ~4f microns.~%"
             (aref APICAL-1-DENDRITE 0 total-segments$)
             (* (aref APICAL-1-DENDRITE 0 total-segments$)
                (aref APICAL-1-DENDRITE 0 length$)))
     (format t "      Length = ~4d microns. Diameter = ~2f microns.Lamda = ~4f. L = ~4f~%"
             (aref APICAL-1-DENDRITE 0 length$)
             (aref APICAL-1-DENDRITE 0 diameter$)
             lamda
             L )
     (format t "      %Lamda(per segment) = ~4f%"
             (* 100.0 (// (aref APICAL-1-DENDRITE 0 length$) lamda)) )))
  ((aref APICAL-2-DENDRITE 0 include-me$)
   (let* ((lamda (^ (// (* *rd-mem (// (aref APICAL-2-DENDRITE 0 diameter$) 2.0) 10000.0)
                        (* 2.0 *rd-int)) 0.5))
          (L (// (* (aref APICAL-2-DENDRITE 0 total-segments$)
                    (aref APICAL-2-DENDRITE 0 length$)) lamda)))
     (format t "~%Right apical dendrite branch with ~2d segments.   Length = ~4f microns.~%"
             (aref APICAL-2-DENDRITE 0 total-segments$)
             (* (aref APICAL-2-DENDRITE 0 total-segments$)
                (aref APICAL-2-DENDRITE 0 length$)))
     (format t "      Length = ~4d microns. Diameter = ~2f microns.Lamda = ~4f. L = ~4f~%"
             (aref APICAL-2-DENDRITE 0 length$)
             (aref APICAL-2-DENDRITE 0 diameter$)
             lamda
             L )
     (format t "       %Lamda(per segment) = ~4f%"
             (* 100.0 (// (aref APICAL-2-DENDRITE 0 length$) lamda)) )))
  ((aref BASAL-DENDRITE 0 include-me$)
   (let* ((lamda (^ (// (* *rd-mem (// (aref BASAL-DENDRITE 0 diameter$) 2.0) 10000.0)
                        (* 2.0 *rd-int)) 0.5))
          (L (// (* (aref BASAL-DENDRITE 0 total-segments$)
                    (aref BASAL-DENDRITE 0 length$)) lamda)))
     (format t "~%Basal dendrite with ~2d segments.   Length = ~4f microns.~%"
             (aref BASAL-DENDRITE 0 total-segments$)
             (* (aref BASAL-DENDRITE 0 total-segments$)
                (aref BASAL-DENDRITE 0 length$)))
     (format t "      Length = ~4d microns. Diameter = ~2f microns.Lamda = ~4f. L = ~4f~%"
             (Aref BASAL-DENDRITE 0 length$)
             (aref BASAL-DENDRITE 0 diameter$)
```

```
                lamda
                L )
        (format t "       %Lamda(per segment) = ~4f%"
                (* 100.0 (// (aref BASAL-DENDRITE 0 length$) lamda)) ))))
    (format t "~%~%Time required to reach steady-state  ~4f msec  " *time-for-steady-state))

;;; MAKE-LIST1
(defun make-list1 ()
  (setq *plot-list1 nil
        *label-list1 nil)
  (setq *plot-list1 (nconc *plot-list1 (list '(*as1voltage* . *time*)))
        *label-list1 (nconc *label-list1 (list (format nil "Soma"))))
  (cond-every
    ((aref APICAL-SHAFT-DENDRITE 0 include-me$)
     (cond-every
       ((aref APICAL-SHAFT-DENDRITE 0 plot-me$)
        (setq *plot-list1 (nconc *plot-list1 (list '(*as1voltage* . *time*)))
              *label-list1 (nconc *label-list1 (list (format nil "Shaft Seg 1")))))
       ((and (aref APICAL-SHAFT-DENDRITE 2 plot-me$) (>=  (aref APICAL-SHAFT-DENDRITE 0 total-segments$) 3))
        (setq *plot-list1 (nconc *plot-list1 (list '(*as3voltage* . *time*)))
              *label-list1 (nconc *label-list1 (list (format nil "Shaft Seg 3")))))
       ((and (aref APICAL-SHAFT-DENDRITE 4 plot-me$) (>=  (aref APICAL-SHAFT-DENDRITE 0 total-segments$) 5))
        (setq *plot-list1 (nconc *plot-list1 (list '(*as5voltage* . *time*)))
              *label-list1 (nconc *label-list1 (list (format nil "Shaft Seg 5")))))
       ((and (aref APICAL-SHAFT-DENDRITE 9 plot-me$) (>=  (aref APICAL-SHAFT-DENDRITE 0 total-segments$) 10))

        (setq *plot-list1 (nconc *plot-list1 (list '(*as10voltage* . *time*)))
              *label-list1 (nconc *label-list1 (list (format nil "Shaft Seg 10")))))))
    ((aref APICAL-1-DENDRITE 0 include-me$)
     (cond-every
       ((aref APICAL-1-DENDRITE 0 plot-me$)
        (setq *plot-list1 (nconc *plot-list1 (list '(*al1voltage* . *time*)))
              *label-list1 (nconc *label-list1 (list (format nil "Left Segment 1")))))
       ((and (aref APICAL-1-DENDRITE 3 plot-me$) (>=  (aref APICAL-1-DENDRITE 0 total-segments$) 4))
        (setq *plot-list1 (nconc *plot-list1 (list '(*al4voltage* . *time*)))
              *label-list1 (nconc *label-list1 (list (format nil "Left Segment 4")))))))
    ((aref APICAL-2-DENDRITE 0 include-me$)
     (cond-every
       ((aref APICAL-2-DENDRITE 0 plot-me$)
        (setq *plot-list1 (nconc *plot-list1 (list '(*ar1voltage* . *time*)))
              *label-list1 (nconc *label-list1 (list (format nil "Right Segment 1")))))
       ((and (aref APICAL-2-DENDRITE 3 plot-me$) (>=  (aref APICAL-2-DENDRITE 0 total-segments$) 4))
        (setq *plot-list1 (nconc *plot-list1 (list '(*ar4voltage* . *time*)))
              *label-list1 (nconc *label-list1 (list (format nil "Right Segment 4")))))))
    ((aref BASAL-DENDRITE 0 include-me$)
     (cond-every
       ((aref BASAL-DENDRITE 0 plot-me$)
        (setq *plot-list1 (nconc *plot-list1 (list '(*b1voltage* . *time*)))
              *label-list1 (nconc *label-list1 (list (format nil "Basal Segment 1")))))
       ((and (aref BASAL-DENDRITE 3 plot-me$) (>=  (aref BASAL-DENDRITE 0 total-segments$) 4))
        (setq *plot-list1 (nconc *plot-list1 (list '(*b4voltage* . *time*)))
              *label-list1 (nconc *label-list1 (list (format nil "Basal Segment 4")))))))))

;;; MAKE-LIST3
(defun make-list3 ()
  (setq *plot-list3 nil
        *label-list3 nil)
  (if *include-dr
      (setq *plot-list3 (nconc *plot-list3 (list '(*dr-current* . *time*)))
            *label-list3 (nconc *label-list3 (list (format nil "DR Current")))))
  (if *include-a
      (setq *plot-list3 (nconc *plot-list3 (list '(*a-current* . *time*)))
            *label-list3 (nconc *label-list3 (list (format nil "A Current")))))
  (if *include-m
      (setq *plot-list3 (nconc *plot-list3 (list '(*m-current* . *time*)))
            *label-list3 (nconc *label-list3 (list (format nil "M Current")))))
  (if *include-c
      (setq *plot-list3 (nconc *plot-list3 (list '(*c-current* . *time*)))
            *label-list3 (nconc *label-list3 (list (format nil "C Current")))))
  (if *include-ahp
      (setq *plot-list3 (nconc *plot-list3 (list '(*ahp-current* . *time*)))
            *label-list3 (nconc *label-list3 (list (format nil "AHP Current")))))
  (if *include-q
      (setq *plot-list3 (nconc *plot-list3 (list '(*q-current* . *time*)))
            *label-list3 (nconc *label-list3 (list (format nil "Q Current")))))
  (if *include-dendrite
      (setq *plot-list3 (nconc *plot-list3 (list '(*coupling-current* . *time*)))
            *label-list3 (nconc *label-list3 (list (format nil "Soma-dendrite Current")))))
  (setq *plot-list3 (nconc *plot-list3 (list '(*l-current* . *time*)))
        *label-list3 (nconc *label-list3 (list (format nil "L Current"))))
  (if *include-shunt
      (setq *plot-list3 (nconc *plot-list3 (list '(*shunt-current* . *time*)))
```

```
          *label-list3 (nconc *label-list3 (list (format nil "Shunt Current"))))))

;;; MAKE-LIST2
(defun make-list2 ()
  (setq *plot-list2 nil
        *label-list2 nil)
  (cond (*include-na1
         (setq *plot-list2 (nconc *plot-list2 (list '(*na1-current* . *time*)))
               *label-list2 (nconc *label-list2 (list (format nil "Na1 Current"))))))
  (cond (*include-na2
         (setq *plot-list2 (nconc *plot-list2 (list '(*na2-current* . *time*)))
               *label-list2 (nconc *label-list2 (list (format nil "Na2 Current"))))))
  (cond (*include-na3
         (setq *plot-list2 (nconc *plot-list2 (list '(*na3-current* . *time*)))
               *label-list2 (nconc *label-list2 (list (format nil "Na3 Current"))))))
  (cond (*include-nap
         (setq *plot-list2 (nconc *plot-list2 (list '(*nap-current* . *time*)))
               *label-list2 (nconc *label-list2 (list (format nil "NaP Current"))))))
  (cond (*include-cas
         (setq *plot-list2 (nconc *plot-list2 (list '(*cas-current* . *time*)))
               *label-list2 (nconc *label-list2 (list (format nil "Cas Current"))))))
  (cond (*include-ca
         (setq *plot-list2 (nconc *plot-list2 (list '(*ca-current* . *time*)))
               *label-list2 (nconc *label-list2 (list (format nil "Ca Current"))))))
  (setq *plot-list2 (nconc *plot-list2 (list '(*caps-current* . *time*)))
               *label-list2 (nconc *label-list2 (list (format nil "Cap Current")))))

;;; MAKE-LIST4
(defun make-list4 ()
  (setq *plot-list4 nil
        *label-list4 nil)
  (if *include-dendrite-synapse
      (setq *plot-list4 (nconc *plot-list4 (list '(*dendrite-synapse-current* . *time*)))
            *label-list4 (nconc *label-list4 (list (format nil "Dendrite Synapse")))))
  (if *include-soma-synapse
      (setq *plot-list4 (nconc *plot-list4 (list '(*soma-synapse-current* . *time*)))
            *label-list4 (nconc *label-list4 (list (format nil "Soma Synapse")))))
  (setq *plot-list4 (nconc *plot-list4 (list '(*stim-current* . *time*)))
        *label-list4 (nconc *label-list4 (list (format nil "Soma Stimulus"))))
  (setq *plot-list4 (nconc *plot-list4 (list '(*dendrite-stim-current* . *time*)))
        *label-list4 (nconc *label-list4 (list (format nil "Dendrite Stimulus")))))

;;; MAKE-LIST5
(defun make-list5 ()
  (setq *plot-list5 nil
        *label-list5 nil)
  (setq *plot-list5 (nconc *plot-list5 (list '(*ca-conc-shell* . *time*)))
            *label-list5 (nconc *label-list5 (list (format nil "Shell Ca Concentration"))))
      (setq *plot-list5 (nconc *plot-list5 (list '(*ca-conc-shell2* . *time*)))
            *label-list5 (nconc *label-list5 (list (format nil "Shell2 Ca Concentration")))))


;;; PLOT-RESULTS Plot all the output lists automatically.
(defun plot-results ()
  (cond (*iclamp-run (plot-current-clamp))
        (*vclamp-run (plot-voltage-clamp))))

;;; PLOT-CURRENT-CLAMP
(defun plot-current-clamp ()
  (make-list1)  (make-list2)  (make-list3)  (make-list4)  (make-list5)
  (send *plot-pane-1 :plot "Soma And Dendritic Potentials [mV] "
        *plot-list1
        *label-list1
        :all-solid-lines *plot-voltages-solid
        :y-min -90
        :y-max 40
        :y-interval 10
        :overlay *overlay-simulations
        :leave-window *overlay-simulations)
  (send *plot-pane-2 :plot "Inward Soma Currents [na]"
        *plot-list2
        *label-list2
        :y-min -10
        :y-max 8
        :y-interval 2
        :overlay *overlay-simulations
        :leave-window *overlay-simulations)
  (send *plot-pane-3 :plot "Outward Soma Currents [na]"
        *plot-list3
        *label-list3
        :y-min -4
        :y-max 10
```

```
        :y-interval 2
        :overlay *overlay-simulations
        :leave-window *overlay-simulations)
  (send *plot-pane-4 :plot "Stimulus and Synapse Conductance Currents [na]"
        *plot-list4
        *label-list4
        :y-interval .25
        :overlay *overlay-simulations
        :leave-window *overlay-simulations)
;   (if (or (aref APICAL-SHAFT-DENDRITE 3 include-na$) (aref APICAL-SHAFT-DENDRITE 3 include-ca$))
;       (send *plot-pane-5 :plot "Dendrite Currents [na]"
  (send *plot-pane-5 :plot "Ca Concentration in Shell"
        *plot-list5
        *label-list5
;           :y-min -10
;           :y-max 10
;           :y-interval 2
        :overlay *overlay-simulations
        :leave-window *overlay-simulations)
      nil)

;;; PLOT-VOLTAGE-CLAMP
(defun plot-voltage-clamp ()
  (make-list1)  (make-list2)  (make-list3)
  (send *plot-pane-1 :plot "Soma And Dendritic Potentials [mV] "
        *plot-list1
        *label-list1
        :all-solid-lines *plot-voltages-solid
        :y-min -90
        :y-max 40
        :y-interval 10
        :overlay *overlay-simulations
        :leave-window *overlay-simulations)
 (send *plot-pane-5 :plot "Voltage Clamp Soma Potential [mV] "
        '((*voltage* . *time*))
        (list (format nil "Soma clamp voltage"))
        :all-solid-lines *plot-voltages-solid
        :y-min -90
        :y-max 40
        :y-interval 10
        :overlay *overlay-simulations
        :leave-window *overlay-simulations)
  (send *plot-pane-3 :plot "Outward Soma Currents [na]"
        *plot-list3
        *label-list3
        :y-min -4
        :y-max 10
        :y-interval 2
        :overlay *overlay-simulations
        :leave-window *overlay-simulations)
  (send *plot-pane-2 :plot "Inward Soma Currents [na]"
        *plot-list2
        *label-list2
        :y-min -10
        :y-max 8
        :y-interval 2
        :overlay *overlay-simulations
        :leave-window *overlay-simulations)
  (send *plot-pane-4 :plot "Total Clamp Current [na]"
        '((*current* . *time*))
        (list (format nil "Soma current "))
        :overlay *overlay-simulations
        :leave-window *overlay-simulations))

(defflavor plot-frame
        ()
        (tv:bordered-constraint-frame)
  :settable-instance-variables
  (:default-init-plist
   :activate-p t
   :expose-p t
   :save-bits t))

(tv:add-select-key #\h 'plot-frame "Hippocampus" '(startup) t)

;;; STARTUP Setup up the simulation frame with 6 plot panes for the relevant ouput lists and one lisp listener
;;; pane for input and parameter printing.
(defflavor tv:plotter-pane () (g:plot-hack tv:pane-mixin))
(defun startup ()
  (tv:make-window
    'plot-frame
```

```
    ':panes '((*plot-pane-1 tv:plotter-pane
                            :label "Voltages In Soma And dendrites")
              (*plot-pane-2 tv:plotter-pane
                            :label "Outward Currents In Soma")
              (*plot-pane-3 tv:plotter-pane
                            :label "Inward Currents In Soma")
              (*plot-pane-4 tv:plotter-pane
                            :label  "Stimulus and Synapse Conductance Currents [na]")
              (*plot-pane-5 tv:plotter-pane
                            :label "Calcium Concentrations")
              (interaction-pane tv:lisp-listener
                                :label "HIPPOCAMPAL PYRAMIDAL CELL SIMULATION"))
    ':configurations '((c1
                        (:layout
                          (c1 :column r1 r2 *plot-pane-1)
                          (r1 :row interaction-pane c2)
                          (r2 :row *plot-pane-2 *plot-pane-3)
                          (c2 :column *plot-pane-5 *plot-pane-4))
                        (:sizes
                          (c1 (*plot-pane-1 200)
                                  :then (r2 200)
                                  :then (r1 :even))
                          (r1 (interaction-pane .500)
                              :then (c2 :even))
                          (r2 (*plot-pane-2 .50)
                              :then (*plot-pane-3 :even))
                          (c2 (*plot-pane-5 .5)
                              :then (*plot-pane-4 :even)))))
    ':expose-p t)
;  (name-plot-panes)
  (variable-array-setup))

;;; NAME-PLOT-PANES Name the plot windows. They must be already in place
;;; But first go the the corresponding Lisp window
(defvar interaction-pane)
(defun name-plot-panes ()
  (loop for pane in (send (send tv:selected-window :superior) :inferiors)
        with count = 1
        when (typep pane 'tv:plotter-pane)
          do (set (intern (format nil "*PLOT-PANE--D" count)) pane)
             (incf count)))




;;; SETUP-STIMULUS Updates the stimulus current to the soma and the dendrites for current clamp, or the clamp
;;; voltage for voltage clamp.
(defun setup-stimulus ()
  (cond (*iclamp-run (set-soma-current-stimulus)        ;Set up the stimulus current to the soma.
                     (set-dendrite-current-stimulus))   ;Set up stimulus current to the  dendrite.
        (*vclamp-run (set-soma-voltage-stimulus *time-step))))




;;; CLAMP
(defun clamp ()
  (setup-menu4)                          ;Set up current clamp run. Sets *vclamp-run and *iclamp-run.
  (without-floating-underflow-traps
    (if *calculate-steady-state
        (initialize-w-new-steady-state)
        (initialize-w-old-steady-state))
    (setq *time 0.0)
;    (cl:time (run-clamp))
    (run-clamp)
    (beep)    (beep)
;    (reverse-lists)
    (if *plot-results (plot-results))
    (print-parameters))
      (beep)
  (setq *first-run nil))


(defun autoclamp ()
  (setup-menu4)
  (autoclamp2 (list  *voltage2-norm2* *voltage3-norm2* *voltage4-norm2*))
  (setq *include-a nil)
```

```
  (autoclamp2 (list  *voltage2-wout-a2* *voltage3-wout-a2*  *voltage4-wout-a2*))
  (plot-results)
  (print-parameters))


(defun autoclamp2 (second-vlist)
    (do ((stimulus-list '(0.33 0.35 0.37 ) (cdr stimulus-list ))
         (voltage-list second-vlist (cdr  voltage-list))
         (voltage-array))
        ((null voltage-list))
      (setq voltage-array (car voltage-list)
            *i-stim-1 -0.5
            *t-stim-1 20.0
            *t-stim-2 1000.0
            *i-stim-2 (car stimulus-list))
      (without-floating-underflow-traps
        (if *calculate-steady-state
            (initialize-w-new-steady-state)
            (initialize-w-old-steady-state))
        (setq *time 0.0)
        (run-clamp)
        (fillarray voltage-array *voltage* ))))



;;; INITIALIZE-W-OLD-STEADY-STATE
(defun initialize-w-old-steady-state ()
;  (clear-lists)
  (setq *time 0.0
        *point-index 0)
  (if *first-run
      (and (initialize-soma-voltage)            ;If first run, set voltage for the soma to *e-holding.
           (if *include-dendrite
               (initialize-dendrite-voltages))) ;If first run, set voltage$ for all the dendrite compartments to
                                                ;*ed-l.
      (and (aset (aref SOMA e-rest$) SOMA voltage$)     ;Otherwise, set voltage$ for all compartments to their
                                                ;appropriate e-rest$.
           (aset (aref SOMA e-rest$) SOMA voltage-est1$)
           (aset (aref SOMA e-rest$) SOMA voltage-est2$)
           (dolist
             (DENDRITE-ARRAY (list BASAL-DENDRITE APICAL-SHAFT-DENDRITE APICAL-1-DENDRITE APICAL-2-DENDRITE))
             (if (aref DENDRITE-ARRAY 0 INCLUDE-ME$)
                 (do ((segment 0 (incf segment)))
                     ((= segment (aref DENDRITE-ARRAY 0 total-segments$)))
                   (aset (aref DENDRITE-ARRAY segment e-rest$) DENDRITE-ARRAY segment voltage$)
                   (aset (aref DENDRITE-ARRAY segment e-rest$) DENDRITE-ARRAY segment voltage-est1$)
                   (aset (aref DENDRITE-ARRAY segment e-rest$) DENDRITE-ARRAY segment voltage-est2$))))))
  (initialize-dendrite-states)              ;Set up dendrite segments with new configuration parameters.
  (initialize-soma-states)                  ;Set up soma with new parameters.
;  (update-output-lists)
  (setq *time-for-steady-state 0.0))


;;; INITIALIZE-W-NEW-STEADY-STATE This function runs the current clamp simulation with 0 injected current in order
;;; to calculate the steady state voltages of all the compartments. The soma starts out at *e-holding and the
;;; dendrite compartments start out at *ed-l
(defun initialize-w-new-steady-state ()
  (setq *i-stim 0.0 *i-den-stim 0.0 *steady-state-run t)
  (initialize-w-old-steady-state)
  (and (aset *e-holding SOMA voltage$)
       (aset  *e-holding SOMA voltage-est1$)
       (aset  *e-holding SOMA voltage-est2$))
  (do
    ((time 0 (+ time *dt))                  ;*TIME keeps track of the elapsed time
     (time-step 0 (+ time-step 1)))         ;*TIME-STEP keeps track of the number of increments
    ((test-for-resting-state))              ;End of clamp.
    (setq *time-for-steady-state time *time-step time-step *time time)
    ;; Before the evaluation loop, evaluate the first approximations to the voltages using previous values and
    ;; their derivatives. Also load in dummy values for next estimates in order to force initial interation.
    (load-first-estimates)
    ;; This evaluation loop performs successive approximations to the model voltages at the present time step
    ;; until the all the estimates satisfy the convergence criterium.
    (do ()
        ((test-estimates))                  ;True if ALL estimates are within epsilon (v-est1's & v-est2's)
      ;; Set v-est1's = v-est2's to prepare for generating new estimate (est2).
      (store-new-soma-estimate)
      (if *include-dendrite (store-new-dendrite-estimates))
      ;; Estimate (trap. approx.) state variables (m-na-est, etc.) based on voltage estimates (v-est1) and
      ;; previous states (m-na, m-na-dot, etc.), and then estimate v-dot (v-est1-dot) w/ state estimates
      ;; (e.g. m-na-est) & current voltage estimates (v-est1's) using KCL.
```

```
    (set-soma-states-and-v-dot-for-current-clamp nil)
    (if *include-dendrite (set-dendrite-states-and-v-dots nil))
    ;; Estimate (trap. approm.) voltage (v-est2) w/ new v-est1-dot, previous voltage (voltage) and the
    ;; previous derivative (voltage-dot).
    (estimate-soma-voltage)
    (if *include-dendrite (estimate-dendrite-voltages)))
  ;; Set new voltage (voltage) to last estimate (voltage-est2). Also set v-est1 to v-est2 for find
  ;; "SET-SOMA-STATES-AND-V-DOTS" and "SET-DENDRITE-STATES-AND-V-DOTS".
  (store-new-soma-voltage)
  (store-new-soma-estimate)
  (cond (*include-dendrite (store-new-dendrite-voltages) (store-new-dendrite-estimates)))
  ;; Calculate (trap. approm.) final estimates of state variables (m-na, etc.) based on stored voltage value
  ;; (v-est1) and previous states (m-na, m-na-dot, etc.). Calculate state variable derivatives (e.g. m-na-dot)
  ;; w/ diff. eq., the final state estimates (e.g. m-na) and final voltage estimate (voltage). Update
  ;; (replace) the stored values for the state variables and their derivatives. Calculate v-dot (voltage-dot)
  ;; w/ states (e.g. m-na) & current voltages (voltage and voltage's) using circuit equation (KCL), and store
  ;; the values.
  (set-soma-states-and-v-dot-for-current-clamp T)
  (if *include-dendrite (set-dendrite-states-and-v-dots T))
  ;; Print simulation time and concantenate the just calculated variables to the appropriate output lists.
  (cond ((zerop (\ *time-step *plot-step))
         (send terminal-io :home-down)(send terminal-io :clear-rest-of-line)
         (format t "Finding steady state; Current time - ~4fms" *time-for-steady-state))))
;          (update-output-lists))))
 (store-steady-state)                          ;Store the steady state values for repeat runs.
 (setq *steady-state-run nil
       *time-step 0))

;;; STORE-STEADY-STATE
(defun store-steady-state ()
  (aset (aref SOMA voltage$) SOMA e-rest$)
  (dolist
    (DENDRITE-ARRAY (list BASAL-DENDRITE APICAL-SHAFT-DENDRITE APICAL-1-DENDRITE APICAL-2-DENDRITE))
    (if (aref DENDRITE-ARRAY 0 INCLUDE-ME$)
        (do ((segment 0 (incf segment)))
            ((= segment (aref DENDRITE-ARRAY 0 total-segments$)))
          (aset (aref DENDRITE-ARRAY segment voltage$) DENDRITE-ARRAY segment e-rest$)))))


;;; RUN-CLAMP
(defun run-clamp ()
  ;; This is the main loop which generates the state variables and voltages for each time increment.
  (do
;    ((time *time-for-steady-state (+ time *dt))          ;*TIME keeps track of the elapsed time
   ((time 0.0 (+ time *dt))     ;In this version ignore time to steady state
    (time-step 0 (+ time-step 1)))                ;*TIME-STEP keeps track of the number of increments
   ((> *point-index (- *plot-points 1)))                  ;End of clamp.
   (setq *time time *time-step time-step)
   (setup-stimulus)                               ;Update up the current or voltage stimulus.
   ;; Before the evaluation loop, evaluate the first approximations to the voltages using previous values and
   ;; their derivatives. Also load in dummy values for next estimates in order to force initial interation. If
   ;; voltage clamp run then both est1 and est2 of soma are set to the current *clamp-voltage.
   (load-first-estimates)
   ;; For the voltage clamp the soma states are only a function of the current soma voltage and the past soma
   ;; states. First set *vstep which says that the clamp voltage has changed.
   (cond (*vclamp-run (setq *vstep (or (= time-step 0)
                                       (= time-step (fixr (// *t-stim-1 *dt)))
                                       (= time-step (fixr (// *t-stim-2 *dt)))
                                       (= time-step (fixr (// *t-stim-3 *dt)))
                                       (= time-step (fixr (// *t-stim-4 *dt)))))
                      (set-soma-states-and-v-dot-for-voltage-clamp)))
   ;; This evaluation loop performs successive approximations to the model voltages at the present time step
   ;; until the all the estimates satisfy the convergence criterium.
   (do ((i 1 (incf i)))
       ((and (> i 1)(test-estimates)))                ;True if ALL estimates are within epsilon (v-est1's & v-est2's)
     ;; Set v-est1's = v-est2's to prepare for generating new estimate (est2).

     (if *iclamp-run (store-new-soma-estimate))
     (if *include-dendrite (store-new-dendrite-estimates))
     ;; Estimate (trap. approm.) state variables (m-na-est, etc.) based on voltage estimates (v-est1) and
     ;; previous states (m-na, m-na-dot, etc.), and then estimate v-dot (v-est1-dot) w/ state estimates
     ;; (e.g. m-na-est) & current voltage estimates (v-est1's) using KCL.
     (if *iclamp-run (set-soma-states-and-v-dot-for-current-clamp nil))
     (if *include-dendrite (set-dendrite-states-and-v-dots nil))       ;
     ;; Estimate (trap. approm.) voltage (v-est2) w/ new v-est1-dot, previous voltage (voltage) and the
     ;; previous derivative (voltage-dot).
     (if *iclamp-run (estimate-soma-voltage))
     (if *include-dendrite (estimate-dendrite-voltages)))
   ;; Set new voltage (voltage) to last estimate (voltage-est2). Also set v-est1 to v-est2 for find
   ;; "SET-SOMA-STATES-AND-V-DOTS" and "SET-DENDRITE-STATES-AND-V-DOTS".
   (store-new-soma-voltage)
   (if *iclamp-run (store-new-soma-estimate))
```

```
    (cond (*include-dendrite (store-new-dendrite-voltages) (store-new-dendrite-estimates)))
    ;; Calculate (trap. approx.) final estimates of state variables (m-na, etc.) based on stored voltage value
    ;; (v-est1) and previous states (m-na, m-na-dot, etc.). Calculate state variable derivatives (e.g. m-na-dot)
    ;; w/ diff. eq., the final state estimates (e.g. m-na) and final voltage estimate (voltage). Update
    ;; (replace) the stored values for the state variables and their derivatives. Calculate v-dot (voltage-dot)
    ;; w/ states (e.g. m-na) & current voltages (voltage and voltages) using circuit equation (KCL), and store
    ;; the values.
    (if *iclamp-run (set-soma-states-and-v-dot-for-current-clamp T))
    (if *include-dendrite (set-dendrite-states-and-v-dots T))
    ;; Print simulation time and concantenate the just calculated variables to the appropriate output lists.
    (cond ((zerop (\ *time-step *plot-step))
           (send terminal-io :home-down)(send terminal-io :clear-rest-of-line)
           (format t "Length of simulation + time for steady state - ~4fms Current time - ~4fms"
                   (+ *time-for-steady-state *duration) *time)
           (update-output-lists)
           (setq *point-index (+ 1 *point-index))))))


(defvars *v-start *v-final *voltage-normal*)
;;; NORMALIZE
(defun normalize (v-start v-final)
  (setq *v-start v-start
        *v-final v-final
        *voltage-normal* (mapcar #'normalize-op *voltage* ))
  t)
;;; NORMALIZE-OP
(defun normalize-op (voltage)
  (// (- voltage *v-final)
      (- *v-start *v-final)))




(defvars *iv-voltage* *iv-current* *volts*)


;;;   Soma geometry ****************
;;;
;;;   Assume soma is a sphere
;;;

(defvar *soma-radius 17.5)                    ;micrometers

(defun surf-area (radius)                     ;sphere surface area is in sq-cm - argument is in micrometers
  (* (// 4.0 3.0) 3.14159 (* radius radius) 1.0e-8))

;;; ************** PASSIVE COMPONENTS

(defvar *temperature 27.0)
(defvar *e-na 50.0)                           ;mvolts
(defvar *e-ca 110.0)                          ;mvolts
(defvar *e-k  -85.0)                          ;mvolts
(defvar *e-holding -70)                       ;mvolts

(defvar *e-l -70.0)                           ;constant leakage battery (mV)
(defvar *ed-l -70.0)

(defvar *faraday 9.648e4)                     ;Coulombs/mole
(defvar *R 8.314)                             ;Gas constant -- (Volts*Coulombs)/(DegreesKelvin*mole)

(defvar *ca-conc-extra 1.8)                   ;Extra-cellular Ca++concentration [ mmol/liter]
                                              ;Hille says 1.5 mM Ca out, <10e-7 mM in.
                                              ;Segal and Barker, 1986 use 4.0 mM Ca out
                                              ;Madison and Nicoll, 1982 use 2.5 mM Ca out
                                              ;Blaxter et al, 1986 use ACSF with 3.25 mM Ca
                                              ;Wong and Prince, 1981 use 2.0 mM Ca

;;; Electrode shunt resistance (Mohm)
(defvar *r-electrode 10000000.0)
;;; Soma input resistance (Mohm)
(defvar *a-l 39.0)
;;; Soma membrane resistance (ohm-cm-cm)
(defvar *rs-mem 850.0)

;;; Dendrite membrane resistance (ohm-cm-cm)
(defvar *rd-mem 40000.0)

;;; Dendrite axoplasmic resistance (ohm-cm)
(defvar *rd-int 200.0)
```

```
;;; Dendrite membrane capacitance (microfarads/sq-cm)
(defvar *capd-mem 1.0)

;;; Input capacitance of soma (nf)
(defvar *caps-in 0.150)
;;; Soma membrane capacitance (microfarads/sq-cm)
(defvar *caps-mem 1.0)
;;; Total capacity of soma (nF)
(defvar *caps)


(defvar *soma-area)                          ;Soma surface area - ***** sq. microns *****
(defvar *gs-l)

;;; MENU-FOR-SOMA-GEOMETRY-AND-PASSIVE-COMPONENTS
(defvar *c-calc T)
(defvar *r-calc T)
(defvar *qten-ionic 1.5)
(defun menu-for-soma-geometry-and-passive-components ()
  (tv:choose-variable-values
    '((*soma-radius      "Soma sphere radius [micrometers] " :number)
      (*e-l   "Leakage battery [mV]   " :number)
      (*e-na  "Na reversal potential [mV]   " :number)
      (*e-k   "K reversal potential [mV]   " :number)
      (*e-ca   "Ca reversal potential [mV]   " :number)
      (*c-calc "Calculate C-mem from geometry (yes) or use input capacitance (no)" :boolean)
      (*caps-mem   "Membrane capacitance [microfarads/sq-cm]   " :number)
      (*caps-in "Input capacity [nF] " :number)
      (*r-calc "Calculate *RS-MEM from geometry (yes) or use input impedance (no)" :boolean)
      (*rs-mem  "Membrane resistance [ohm-cm-cm] " :number)
      (*a-l      "Input impedance [MOhm] (used to substitute for soma and dendrite Rin only) " :number)
      (*temperature "Temperature of experiment [Celsius]" :number)
      (*qten  "Q-10 [Rate constant coefficient per 10 degrees]" :number)
      (*qten-ionic  "Q-10 [Ionic conductance coefficient per 10 degrees]" :number)
      (*include-shunt "Include electrode shunt conductance (if no the g-shunt will be ignored)?" :boolean)
      (*r-electrode "Electrode shunt [Mohms]" :number)
      (*i-constant-injection "Constant current injected [nA]" :number))
    :label    "Passive components")
  (setq *soma-area (* 1.0e8 (surf-area *soma-radius))
        *gs-l (if *r-calc (// (surf-area *soma-radius) (* *rs-mem 1.0e-6))
                  (// 1.0 *a-l ))
        *g-electrode (// 1.0 *r-electrode)
        *caps (if *c-calc (* (surf-area *soma-radius) *caps-mem 1.0e3) *caps-in))
  (aset *gs-l SOMA g-leak$)
  (aset 100.0 SOMA g-axial$)                 ; Assume that soma has essentially zero axoplasmic resistivity.
  (update-qtens)
  (update-gbars))



(defun update-qtens ()
  (setq *qten-factor-at-25 (qten-tau-factor 25.0 *temperature *qten)
        *qten-factor-at-25-m (qten-tau-factor 25.0 *temperature *qten-m)
        *qten-factor-at-32 (qten-tau-factor 32.0 *temperature *qten)     ;Ca kinetics
        *qten-factor-at-30 (qten-tau-factor 30.0 *temperature *qten)     ;DR and A kinetics
        *qten-factor-at-27 (qten-tau-factor 27.0 *temperature *qten)
        *qten-factor-at-22 (qten-tau-factor 22.0 *temperature *qten)
        *qten-factor-at-24 (qten-tau-factor 24.0 *temperature 5.0)       ;Na kinetics.
        *qten-g-24 (qten-rate-factor 24.0 *temperature *qten-ionic)      ;Qten for ionic conductance of Na currents.
        *qten-g-30 (qten-rate-factor 30.0 *temperature *qten-ionic)      ;Qten for ionic conductance of DR and
                                                  ;A currents.
        *qten-g-32 (qten-rate-factor 32.0 *temperature *qten-ionic)      ;Qten for ionic conductance of Ca currents.
        *qten-factor-at-37 (qten-tau-factor 37.0 *temperature *qten)
        *qten-factor-at-14 (qten-tau-factor 14.0 *temperature *qten)))

;;;QTEN-TAU-FACTOR This calculates the qten factor for time constants (as temperature goes up, tau goes down).
(defun qten-tau-factor (reference-temp temp qten)
  (^ qten (// (-  reference-temp temp ) 10.0)))

;;;QTEN-RATE-FACTOR This calculates the qten factor for rate constants (as temperature goes up, so does rate).
(defun qten-rate-factor (reference-temp temp qten)
  (^ qten (// (-  temp reference-temp  ) 10.0)))

(defvars-w-value (*qten-g-24 1.0)(*qten-factor-at-27 1.0))
(defvars-w-value (*qten-g-30 1.0)(*qten-factor-at-30 1.0))
(defvars-w-value (*qten-g-32 1.0)(*qten-factor-at-32 1.0))
```

```
;;; L-CURRENT The leakage current.
(defun l-current (v)
  (* *gs-l (- v *e-l)))



;;; ALPHA and BETA These functions give the voltage dependant rate constants for the single barrier model, where alpha is the
;;; forward rate constant and beta is the backward rate constant. "v-half" is the voltage at which the forward
;;; and backward rate constants are equal. Note that there are two aspects of the temperature dependence of
;;; these rate constants. The first derives from the voltage-dependent Boltzmann distribution, which is
;;; explicitely calculated in these functions. The second arises in a lumped "Qten" factor that is strictly a
;;; coefficient for the "base-rate", that is the rate derived from the original free-energy changes of the gating
;;; particle. Since this factor depends on each current, the Qten factor is not included here but in the time constant
;;; functions for each gating variable.

(defun alpha (voltage v-half base-rate valence gamma)
  (let ((exponent (// (* (-  voltage v-half ) 1.0e-3 valence faraday gamma) (* R (+ *temperature 273.0)))))
    (setq exponent (cond ((> exponent 20.0) 20.0) ((< exponent -10.0) -10.0)
                         (t exponent)))
    (* base-rate (exp exponent))))

(defun beta (voltage v-half base-rate valence gamma)
  (let ((exponent (// (* (-  v-half voltage) 1.0e-3 valence faraday (- 1.0 gamma)) (* R (+ *temperature 273.0)))))
    (setq exponent (cond ((> exponent 20.0) 20.0)
                         ((< exponent -10.0) -10.0)
                         (t exponent)))
    (* base-rate (exp exponent))))



;;;   I-C-current **********
;;;
;;;  The Ca dependent K-current
;;;
;;; For now make it analogous to the A current, except that the C current is faster and is dependant on the
;;; concentration of Ca++in the shell in the same way as the AHP current.

;;; conductance in micro-siemens
(defvar *gbar-c 0.40)

(defvars-w-value (*v-half-cx -65.0) (*alpha-base-rate-cx 0.007)  (*valence-cx 25.0) (*gamma-cx 0.20)
                 (*v-half-cy -60.0) (*alpha-base-rate-cy 0.003) (*valence-cy 20.0) (*gamma-cy 0.2)
                 (*base-tcx 0.25)(*base-tcy 15))

(defvars-w-value (*alpha-c 1.0)  (*beta-c 1.0))
(defvars-w-value (*tau-alpha-c 0.0001)  (*tau-beta-c 8.0))


;;; W-C-INF w-c is calcium-dependent gating variable for C-current
(defun w-c-inf (calc-conc-shell)
  (// (* *alpha-c calc-conc-shell calc-conc-shell calc-conc-shell)
      (+ *beta-c (* *alpha-c calc-conc-shell calc-conc-shell calc-conc-shell ))))

;;; T-W-C
(defun t-w-c (calc-conc-shell)
  (let ((tau  (// 1.0 (+ *beta-c (* *alpha-c calc-conc-shell calc-conc-shell calc-conc-shell)))))
    (* *qten-factor-at-27 (if (< tau 0.2) 0.20
    tau))))

;;; A-X-C
(defun a-x-c (voltage)
  (alpha voltage *v-half-cx *alpha-base-rate-cx *valence-cx *gamma-cx))

;;; B-X-C
(defun b-x-c (voltage)
  (beta voltage *v-half-cx *alpha-base-rate-cx *valence-cx *gamma-cx))

;;; A-Y-C
(defun a-y-c (voltage)
  (beta voltage *v-half-cy *alpha-base-rate-cy *valence-cy *gamma-cy))

;;; B-Y-C
(defun b-y-c (voltage)
  (alpha voltage *v-half-cy *alpha-base-rate-cy *valence-cy *gamma-cy))

;;; X-C-INF
;;;       x-inf is activation variable for C-current
(defvar *x-c-inf-midpoint 0.0)
(defun x-c-inf (voltage)
  (// (a-x-c voltage) (+ (a-x-c voltage) (b-x-c voltage))))
```

```
;;; Y-C-INF
;;;       y-inf is inactivation variable for c-current
(defvar *y-c-inf-midpoint 5.0)
(defun y-c-inf (voltage)
  (// (a-y-c voltage) (+ (a-y-c voltage) (b-y-c voltage))))
;  (let ((steepness 2.0))        ;Segal and Barker; Segal, Rogawski, and Barker
;    (// 1.0     (+ 1.0 (exp (// (- voltage *y-c-inf-midpoint) steepness))))))

;;; T-X-C
;;;       tau-C-current(activation) - msec       (estimate)
(defvar *t-x-c .50)
(defun t-x-c (&optional voltage)
  (let ((tx (// 1.0 (+ (a-x-c voltage) (b-x-c voltage)))))
    (* *qten-factor-at-27 (if (< tx *base-tcx) *base-tcx tx))))
;(defun t-x-c (voltage)                          ;tau tail current (Brown and Griffith) (msec)
;  (cond ((< voltage -30.0) (* 20.0 (exp (// (+ voltage 40.0) 18.0))))
;        (t (* 20.0 (exp (// (- 40.0 (+ voltage 60.0)) 18.0))))))

;;; T-Y-C
;;;       tau-C-current(inactivation) - msec
(defvar *t-y-c 1.0)
(defun t-y-c (&optional voltage)
  (let ((ty (// 1.0 (+ (a-y-c voltage) (b-y-c voltage)))))
    (* *qten-factor-at-27 (if (< ty *base-tcy) *base-tcy ty))))

;;; MENU-FOR-C-CURRENT
(defvar *c-shift 0.0)
(defun menu-for-c-current ()
  (tv:choose-variable-values
    '((*gbar-c "C-current conductance [micro-S]" :number)
      ""
      " X Variable Kinetics "
      ""
      (*v-half-cx "V/12 for x" :number)
      (*alpha-base-rate-cx "Alpha-base value for x at V1/2" :number
                           :documentation "Increase makes it faster")
      (*valence-cx "Valence for x" :number)
      (*gamma-cx "Gamma for x" :number)
      (*base-tcx "Minimum value for time constant [ms]" :number)
      ""
      " Y Variable Kinetics "
      ""
      (*v-half-cy "V/12 for y" :number)
      (*alpha-base-rate-cy "Alpha-base value for y at V1/2" :number
                                                :documentation "Increase makes it faster")
      (*Valence-cy "Valence for y" :number)
      (*gamma-cy "Gamma for Na 1 y" :number)
      (*base-tcy "Minimum value for time constant [ms]" :number)
      ""

      ""
      ""
      " W Variable Kinetics "
      ""
      (*tau-alpha-c "Forward time constant for Ca++-binding to W particle" :number)
      (*tau-beta-c "Backward time constant for Ca++-binding to W particle" :number)
      ))
  (setq *alpha-c (// 1.0 *tau-alpha-c)
        *beta-c (// 1.0 *tau-beta-c)))

;;; C-CURRENT Function to calculate the C current.
(defun c-current (x-c y-c w-c v)
  (* (g-c *gbar-c x-c y-c w-c)
     (- v *e-k)))

;;; G-C
(defun g-c (gbar-c x-c y-c w-c)
  (if (< x-c 0.01) (setq x-c 0.0))
  (* gbar-c x-c  x-c x-c x-c y-c   w-c))

;;;X-C-EFF, Y-C-EFF
(defun x-c-eff (x-c)
  (if (< x-c 0.01) 0.0
  (^ x-c 4.0)))
(defun y-c-eff (y-c)
  (^ y-c 1.0))

;;; C-PLOT
(defvars *x-c-inf* *y-c-inf* *x-c-eff* *y-c-eff*
  *t-x-c* *t-y-c* *g-c-inf*)
(defun c-plot ()
```

```
  (menu-for-c-current)
  (setq *volts* nil *y-c-inf* nil *x-c-inf* nil  *y-c-eff* nil *x-c-eff* nil
        *t-x-c* nil *t-y-c* nil *g-c-inf* nil)
  (do   ((v -100.0 (+ v 0.5)))
        ((> v 50))
    (setq
      *y-c-inf*  (nconc *y-c-inf* (list (y-c-inf v)))
      *x-c-inf* (nconc *x-c-inf* (list (x-c-inf v)))
      *y-c-eff*  (nconc *y-c-eff* (list (y-c-eff (y-c-inf v))))
      *x-c-eff* (nconc *x-c-eff* (list (x-c-eff (x-c-inf v))))
      *t-x-c* (nconc *t-x-c* (list (t-x-c v)))
      *t-y-c* (nconc *t-y-c* (list (t-y-c v)))
      *volts* (nconc *volts* (list v))
      *g-c-inf* (nconc *g-c-inf* (list (g-c 1.0 (x-c-inf v)(y-c-inf v) 1.0 ))))))

;;; W-C-PLOT
(defvars *w-c-inf* *t-w-c* *calconc*)
(defun w-c-plot ()
  (menu-for-c-current)
  (setq *w-c-inf* nil *calconc* nil *t-w-c* nil)
  (do   ((ca 1.0e-6 (* ca 1.2 )))((> ca 10.0))
    (setq *w-c-inf* (nconc *w-c-inf* (list (w-c-inf ca)))
          *t-w-c* (nconc *t-w-c* (list (t-w-c ca)))
          *calconc* (nconc *calconc* (list ca)) ) ))


;;;    I-M current ********************
;;;
;;;  The muscarinic -sensitive K current of Paul Adams

;;; M-CURRENT
(defun m-current (x-m v)
  (* *gbar-m x-m (- v *e-k)))

;;; I-M conductance - Only activate between -70mv and -30mv (micro-siemens)
(defvar *gbar-m .005)

(defvars-w-value
  (*m-block 1.0)  (*base-tmx 10)    (*v-half-mx -45.0) (*base-rate-mx 0.0015) (*valence-mx 5) (*gamma-mx .5))

(defvar *qten-m 5.0)                              ;as reported by Paul

;;; I-M time constant - from two values given by Paul (msec)
;;; Constant says ~125 ms @ -40 mv (olfactory cortical cells)
;;; T-X-M
(defun t-x-m (voltage)
  (let* ((b (alpha voltage *v-half-mx *base-rate-mx *valence-mx *gamma-mx))
         (a (beta voltage *v-half-mx *base-rate-mx *valence-mx *gamma-mx))
         (tx (// *qten-factor-at-25 (+ a b))))
    (if (< tx (* *qten-factor-at-25 *base-tmx)) (* *qten-factor-at-25  *base-tmx) tx)))

;;; X-M-INF      x-inf is activation variable for M-current
(defun x-m-inf (voltage)
  (let ((b (beta voltage *v-half-mx *base-rate-mx *valence-mx *gamma-mx))
        (a (alpha voltage *v-half-mx *base-rate-mx *valence-mx *gamma-mx)))
    (// a (+ a b))))

;;; MENU-FOR-M-CURRENT
(defun menu-for-m-current ()
  (tv:choose-variable-values
    '((*gbar-m "M-current absolute conductance [micro-S]" :number)
      (*m-block "Block some fraction of absolute conductance [0-1]" :number)
      ""
           " ** X Variable Kinetics ** "
      ""
      (*v-half-mx "V/12 for M x" :number)
      (*base-rate-mx "Alpha-base value for M x at V1/2" :number)
      (*valence-mx "Valence for M x" :number)
      (*gamma-mx "Gamma for M x" :number)
      (*base-tmx "Minimum value for time constant [ms]" :number)
      "")
      :label "M Potassium Current"
      ))

;;; M-PLOT
(defvars *x-m-inf* *t-x-m*)
(defun m-plot ()
  (menu-for-m-current)
  (setq *x-m-inf* nil *volts* nil *t-x-m* nil)
  (do   ((v -100.0 (+ v 0.5)))
        ((> v 50.0))
    (setq *x-m-inf* (nconc *x-m-inf* (list (x-m-inf v)))
```

```
          *t-x-m* (nconc *t-x-m* (list (t-x-m v)))
          *volts* (nconc *volts* (list v)) ) ))

;;;   I-Q current ********************
;;
;;; This is the outward "anomalous rectifier" current that is activated by hyperpolarizing the cell.
;;; Ref. - Segal and Barker, Halliwell and Adams

;;; E-Q    Q current may be a mixed conductance.
(defvar *e-q -65.0)

;;; I-Q conductance (micro-siemans)
(defvar *gbar-q .002)                          ;About 2nS at full activation (Paul)

;;; Q-CURRENT
(defun q-current (x-q v)
  (* *gbar-q x-q (- v *e-q)))

;;; X-Q-INF
(defun x-q-inf (v)
  (// 1 (+ 1 (exp (// (+ v 84.0) 4.0)))))

;;; T-X-Q
(defun t-x-q (v)
  (* *qten-factor-at-25-m                      ;Paul reports Q-10 for both M and Q currents to be ~5.
     (* 1200.0 (+ (// 1 (+ 1 (exp (// (+ v 85.0) -6.0)))) .1))))

;;; MENU-FOR-Q-CURRENT
(defun menu-for-q-current ()
  (tv:choose-variable-values
    '((*gbar-q "Q-current conductance [micro-S]" :number)
      (*e-q "Q current reversal potential [mV]" :number))))


;;; Q-PLOT
(defvars *x-q-inf* *t-x-q*)
(defun q-plot ()
  (setq *x-q-inf* nil *volts* nil *t-x-q* nil)
  (do   ((v -100.0 (+ v 0.5)))
        ((> v 50.0))
    (setq *x-q-inf* (nconc *x-q-inf* (list (x-q-inf v)))
          *t-x-q* (nconc *t-x-q* (list (t-x-q v)))
          *volts* (nconc *volts* (list v)) ) ))


;;;       DR-current **********************

(defvars-w-value
  (*dr-block 1.0)
  (*base-txdr 0.50)  (*base-tydr 6.0)
  (*v-half-drx -28.0)
  (*base-rate-drx 0.008) (*valence-drx 12) (*gamma-drx .95)
  (*v-half-dry -45.0)
  (*base-rate-dry 0.0004) (*valence-dry 9) (*gamma-dry 0.2))

(defvar *e-dr -73.50)                          ;I-DR reversal potential

;;; DR conductance (microsiemans)
(defvar *gbar-dr 0.7)                          ;Segal reports 0.350

;;; Y-DR-INF      y-inf is inactivation variable for DR-current
;;;               Segal and Barker
(defun y-dr-inf (voltage)
  (let ((b (alpha voltage *v-half-dry *base-rate-dry *valence-dry *gamma-dry))
        (a (beta voltage *v-half-dry *base-rate-dry *valence-dry *gamma-dry)))
    (// a (+ a b))))

;;; T-Y-DR        tau-DR-current(inactivation) -
;;;               Segal and Barker 40 msec
(defun t-y-dr (voltage)
  (let* ((b (alpha voltage *v-half-dry *base-rate-dry *valence-dry *gamma-dry))
         (a (beta voltage *v-half-dry *base-rate-dry *valence-dry *gamma-dry))
         (ty (// *qten-factor-at-30 (+ a b))))
     (if (< ty (* *qten-factor-at-30 *base-tydr)) (* *qten-factor-at-30  *base-tydr) ty)))

;;; X-DR-INF
;;;       x-inf is activation variable for DR-current
;;;               Segal and Barker
(defun x-dr-inf (voltage)
    (let ((a (alpha voltage *v-half-drx *base-rate-drx *valence-drx *gamma-drx))        ;
          (b (beta voltage *v-half-drx *base-rate-drx *valence-drx *gamma-drx)))
      (// a (+ a b))))
```

```
;;; T-X-DR
;;;       tau-DR-current(activation) - msec
;;;       Segal and Barker 180 ms < -30mv,6 ms else

(defun t-x-dr (voltage)
    (let* ((a (alpha voltage *v-half-drx *base-rate-drx *valence-drx *gamma-drx))
           (b (beta voltage *v-half-drx *base-rate-drx *valence-drx *gamma-drx))
           (tx (// *qten-factor-at-30 (+ a b))))
      (if (< tx (* *qten-factor-at-30 *base-txdr)) (* *qten-factor-at-30 *base-txdr) tx)))


;;; MENU-FOR-DR-CURRENT
(defun menu-for-dr-current ()
  (tv:choose-variable-values
    '((*gbar-dr "DR-current absolute conductance [micro-S]" :number)
      (*dr-block "Block some fraction of absolute conductance [0-1]" :number)
      ""
            " ** X Variable Kinetics ** "
      ""
      (*v-half-drx "V/12 for Dr x" :number)
      (*base-rate-drx "Alpha-base value for Dr x at V1/2" :number)
      (*valence-drx "Valence for Dr x" :number)
      (*gamma-drx "Gamma for Dr x" :number)
      (*base-txdr "Minimum value for time constant [ms]" :number)
      ""
      " ** Y Variable Kinetics **"
      ""
      (*v-half-dry "V/12 for Dr y" :number)
      (*base-rate-dry "Alpha-base value for Dr y at V1/2" :number)
      (*valence-dry "Valence for Dr y" :number)
      (*gamma-dry "Gamma for Dr y" :number)
      (*base-tydr "Minimum value for time constant [ms]" :number)
      "")
      :label "Delayed-Rectifier Potassium Current"
      ))


;;; DR-CURRENT
(defun dr-current (x-dr y-dr v)
  (* (g-dr (aref SOMA gbar-dr$) x-dr y-dr) (- v *e-dr)))

;;; G-DR
(defun g-dr (gbar-dr x-dr y-dr)
  (* gbar-dr *dr-block
     x-dr x-dr   x-dr y-dr ))

;;;X-DR-EFF, Y-DR-EFF
(defun x-dr-eff (x-dr)
  (^ x-dr 3.0))
(defun y-dr-eff (y-dr)
  (^ y-dr 1.0))

;;; DR-PLOT
(defvars *x-dr-inf* *y-dr-inf* *x-dr-eff* *y-dr-eff*
  *t-x-dr* *t-y-dr* *g-dr-inf*)
(defun dr-plot ()
  (menu-for-dr-current)
  (setq *volts* nil *y-dr-inf* nil *x-dr-inf* nil  *y-dr-eff* nil *x-dr-eff* nil
        *t-x-dr* nil *t-y-dr* nil *g-dr-inf* nil)
  (do   ((v -100.0 (+ v 0.5)))
        ((> v 50))
    (setq
      *y-dr-inf*  (nconc *y-dr-inf* (list (y-dr-inf v)))
      *x-dr-inf* (nconc *x-dr-inf* (list (x-dr-inf v)))
      *y-dr-eff*  (nconc *y-dr-eff* (list (y-dr-eff (y-dr-inf v))))
      *x-dr-eff* (nconc *x-dr-eff* (list (x-dr-eff (x-dr-inf v))))
      *t-x-dr* (nconc *t-x-dr* (list (t-x-dr v)))
      *t-y-dr* (nconc *t-y-dr* (list (t-y-dr v)))
      *volts* (nconc *volts* (list v))
      *g-dr-inf* (nconc *g-dr-inf* (list (g-dr 1.0 (x-dr-inf v)(y-dr-inf v)))))))



;;;       AHP-current **********************
;;;
;;; Iahp will have two voltage-dependent  inactivation particles, y and z, and
;;; a calcium-depedent gating particle, w.

;;; AHP-CURRENT
(defun ahp-current (z-ahp y-ahp w-ahp  v)
```

```
  (* (g-ahp *gbar-ahp  z-ahp y-ahp w-ahp)
     (- v *e-k)))

;;; AHP conductance (microsiemens)
(defvar *gbar-ahp  0.35)

;;; G-AHP - new version
(defun g-ahp (gbar-ahp  z-ahp y-ahp w-ahp)
  (* gbar-ahp 1.0  y-ahp y-ahp w-ahp z-ahp))

(defun y-ahp-eff (y-ahp)
  (^ y-ahp 2.0))

(defvars *alpha-ahp *beta-ahp)
(defvars-w-value (*tau-alpha-ahp 1.0e-5)  (*tau-beta-ahp 200.0))
(defvars-w-value (*v-half-ahpz -72.0) (*alpha-base-rate-ahpz 2.0e-4)
  (*valence-ahpz 12.0) (*gamma-ahpz 0)
  (*v-half-ahpy -50.0) (*alpha-base-rate-ahpy 0.015)
  (*valence-ahpy 15.0) (*gamma-ahpy 0.8)
  (*base-tahpz 120.0)(*base-tahpy 2.5))

;;; W-AHP-INF w-ahp is calcium-dependent gating variable for AHP-current
(defun w-ahp-inf (calc-conc-shell )
  (// (* *alpha-ahp calc-conc-shell calc-conc-shell calc-conc-shell )
      (+ *beta-ahp (* *alpha-ahp calc-conc-shell calc-conc-shell calc-conc-shell ))))

;;; T-W-AHP
(defun t-w-ahp (calc-conc-shell)
  (let ((tau  (// 1.0 (+ *beta-ahp (* *alpha-ahp calc-conc-shell calc-conc-shell calc-conc-shell)))))
    (* *qten-factor-at-27 (if (< tau 0.002) 0.0020
    tau))))

;;; Y-AHP-INF    y-inf is inactivation variable for AHP-current
(defun y-ahp-inf (voltage)
  (let ((b (alpha voltage *v-half-ahpy *alpha-base-rate-ahpy *valence-ahpy *gamma-ahpy))
        (a (beta voltage *v-half-ahpy *alpha-base-rate-ahpy *valence-ahpy *gamma-ahpy)))
    (// a (+ a b))))

;;; T-Y-AHP      tau-AHP-current(inactivation) - msec
(defun t-y-ahp (voltage)
  (let* ((b (alpha voltage *v-half-ahpy *alpha-base-rate-ahpy *valence-ahpy *gamma-ahpy))
         (a (beta voltage *v-half-ahpy *alpha-base-rate-ahpy *valence-ahpy *gamma-ahpy))
         (ty (// 1.0 (+ a b))))
     (if (< ty *base-tahpy) *base-tahpy ty)))


;;; Z-AHP-INF
;;;      z-inf is activation variable for AHP-current
(defun z-ahp-inf (voltage)
    (let ((b (alpha voltage *v-half-ahpz *alpha-base-rate-ahpz *valence-ahpz *gamma-ahpz))      ;
          (a (beta voltage *v-half-ahpz *alpha-base-rate-ahpz *valence-ahpz *gamma-ahpz)))
      (// a (+ a b))))

;;; T-Z-AHP
;;;      tau-AHP-current(activation) - msec
(defun t-z-ahp (voltage)
    (let* ((b (alpha voltage *v-half-ahpz *alpha-base-rate-ahpz *valence-ahpz *gamma-ahpz))
           (a (beta voltage *v-half-ahpz *alpha-base-rate-ahpz *valence-ahpz *gamma-ahpz))
           (tz (// 1.0 (+ a b))))
      (if (< tz *base-tahpz) *base-tahpz tz)))

;;; MENU-FOR-AHP-CURRENT
(defun menu-for-ahp-current ()
  (tv:choose-variable-values
    '((*gbar-ahp "AHP-current conductance [micro-S]" :number)
      ""
      " Z Variable Kinetics "
      ""
      (*v-half-ahpz "V/12 for Na 1 m" :number)
      (*alpha-base-rate-ahpz "Alpha-base value for Na 1 m at V1/2" :number
                             :documentation "Increase speeds up gating particle")
      (*valence-ahpz "Valence for Na 1 m" :number)
      (*gamma-ahpz "Gamma for Na 1 m" :number)
      (*base-tahpz "Minimum value of time constant [msec]" :number)
      ""
      ""
      " Y Variable Kinetics "
      ""
      (*v-half-ahpy "V/12 for Na 1 h" :number)
      (*alpha-base-rate-ahpy "Alpha-base value for Na 1 h at V1/2" :number
                             :documentation "Increase speeds up gating particle")
      (*valence-ahpy "Valence for Na 1 h" :number)
```

```
      (*gamma-ahpy "Gamma for Na 1 h" :number)
      (*base-tahpy "Minimum value of time constant [msec]" :number)

      ""
      ""
      " W Variable Kinetics "
      ""
      (*tau-alpha-ahp "Forward time constant for Ca++-binding to W particle" :number)
      (*tau-beta-ahp "Backward time constant for Ca++-binding to W particle" :number)
      ))
  (setq *alpha-ahp (// 1.0 *tau-alpha-ahp)
        *beta-ahp (// 1.0 *tau-beta-ahp)))

;;; AHP-PLOT
(defvars *z-ahp-inf* *t-z-ahp* *y-ahp-inf* *t-y-ahp*  *y-ahp-eff*)
(defun ahp-plot ()
  (setq *t-z-ahp* nil *t-y-ahp* nil *volts* nil *z-ahp-inf* nil *y-ahp-inf* nil *y-ahp-eff* nil  )
  (menu-for-ahp-current)
  (do ((v -100.0 (+ v 0.5))) ((> v 50))
    (setq *z-ahp-inf*  (nconc *z-ahp-inf* (list (z-ahp-inf v)))
          *y-ahp-inf*  (nconc *y-ahp-inf* (list (y-ahp-inf v)))
          *y-ahp-eff* (nconc *y-ahp-eff* (list (y-ahp-eff (y-ahp-inf v))))
          *t-z-ahp*  (nconc *t-z-ahp* (list (t-z-ahp v)))
          *t-y-ahp*  (nconc *t-y-ahp* (list (t-y-ahp v)))
          *volts* (nconc *volts* (list v)) ) ))

;;; W-AHP-PLOT
(defvars *w-ahp-inf*  *t-w-ahp* *calconc*)
(defun w-ahp-plot ()
  (menu-for-ahp-current)
  (setq *w-ahp-inf* nil *calconc* nil *t-w-ahp* nil)
  (do   ((ca 1.0e-6 (* ca 1.2 ))) ((> ca 10.0))
    (setq *w-ahp-inf* (nconc *w-ahp-inf* (list (w-ahp-inf ca)))
          *t-w-ahp* (nconc *t-w-ahp* (list (t-w-ahp ca)))
          *calconc* (nconc *calconc* (list ca)) ) ))


;;;       A-current ************
;;;
;;; Zbicz and Weight report that I-a activates in <10ms and decays over
;;; several hundred ms (380ms @-50 to -40mv) (32 degreesC). However, 4-AP sensitive tail
;;; currents which have
;;; time constants of a few hundred ms in reponse to hyperpolarizing pulses to -54mv suddenly
;;; disappear when the clamp is below -54mv, suggesting that the time constant for inactivation
;;; is very rapid for potentials below -54mv, i.e. "The failure to observe any 4-AP-sensitive
;;; tail currents negative to -54mv suggests that the 4-AP-sensitive transient current
;;; deactivates very rapidly upon hyperpolarization."

;;; MENU-FOR-a-CURRENT
(defun menu-for-a-current ()
  (tv:choose-variable-values
    '((*gbar-a "a-current conductance [micro-S]" :number)
      ""
           " X Variable Kinetics "
      ""
      (*v-half-ax "V/12 for a x (sb=-30,zw=-45)" :number)
      (*base-rate-ax "Alpha-base value for a x at V1/2" :number)
      (*valence-ax "Valence for a x (sb=3.67,zw=8.5)" :number)
      (*gamma-ax "Gamma for a x" :number)
      (*base-txa "Minimum value for time constant [ms]" :number)
      ""
      " Y Variable Kinetics "
      ""
      (*v-half-ay "V/12 for a y (sb=-70,zw=-55)" :number)
      (*base-rate-ay "Alpha-base value for a y at V1/2" :number)
      (*valence-ay "Valence for a y (sb=4.28,zw=8)" :number)
      (*gamma-ay "Gamma for a y" :number)
      (*base-tya "Minimum value for time constant [ms]" :number)
      ""
      )))
(defvars-w-value
  (*base-txa 1.0)  (*base-tya 24.0)
  (*v-half-ax -52.0) (*base-rate-ax 0.2) (*valence-ax 3.5) (*gamma-ax 0.8)
  (*v-half-ay -72.0) (*base-rate-ay 0.0015) (*valence-ay 7) (*gamma-ay 0.4))

;;; A-current conductance (microsiemens)
(defvar *gbar-a  .50 )

;;; X-A-INF
;;;       x-inf is activation variable for A-current (- not confirmed sigmoid)
(defun x-a-inf (voltage)
;Segal and Barker; Segal, Rogawski, and Barker -  z=3.67,vhalf=-30
```

```
;Zbicz and Weight - z=8.5,vhalf=-45
    (let ((a (alpha voltage *v-half-ax *base-rate-ax *valence-ax *gamma-ax))    ;
          (b (beta voltage *v-half-ax *base-rate-ax *valence-ax *gamma-ax)))
      (// a (+ a b))))

;;; Y-A-INF
;;;       y-inf is inactivation variable for A-current
(defun y-a-inf (voltage)
;Segal and Barker; Segal, Rogawski, and Barker - z=4.28,vhalf=-70
;Z & W = z=8,vhalf=-55
  (let ((b (alpha voltage *v-half-ay *base-rate-ay *valence-ay *gamma-ay))
        (a (beta voltage *v-half-ay *base-rate-ay *valence-ay *gamma-ay)))
    (// a (+ a b))))

;;; T-X-A
;;;       tau-A-current(activation) - msec       (estimate)
(defvars-w-value (*t-x-a-1 3.0)(*t-x-a-2 5.0)(*t-y-a-1 5.0))
(defun t-x-a (&optional voltage)
;Segal and Barker; Segal, Rogawski, and Barker. Measured from V-holding = -70mv to steps up to -20mv
;Z  & W Probably more of an estimate, i.e. "< 10ms"".
    (let* ((a (alpha voltage *v-half-ax *base-rate-ax *valence-ax *gamma-ax))
           (b (beta voltage *v-half-ax *base-rate-ax *valence-ax *gamma-ax))
           (tx (// *qten-factor-at-30 (+ a b))))
      (if (< tx (* *qten-factor-at-30 *base-txa)) (* *qten-factor-at-30 *base-txa) tx)))

;;; T-Y-A
;;;       tau-A-current(inactivation) - msec
(defun t-y-a (&optional voltage)
;Segal and Barker; Segal, Rogawski, and Barker
;Z & W Supposedly very rapid below -54mv(5ms) ~380 ms otherwise.
  (let* ((b (alpha voltage *v-half-ay *base-rate-ay *valence-ay *gamma-ay))
         (a (beta voltage *v-half-ay *base-rate-ay *valence-ay *gamma-ay))
         (ty (// *qten-factor-at-30 (+ a b))))
      (if (< ty (* *qten-factor-at-30 *base-tya)) (* *qten-factor-at-30  *base-tya) ty)))

;;; A-PLOT
(defvars *x-a-inf* *y-a-inf* *x-a-eff* *y-a-eff*
  *t-x-a* *t-y-a* *g-a-inf*)
(defun a-plot ()
  (menu-for-a-current)
  (setq *volts* nil *y-a-inf* nil *x-a-inf* nil  *y-a-eff* nil *x-a-eff* nil
        *t-x-a* nil *t-y-a* nil *g-a-inf* nil)
  (do   ((v -100.0 (+ v 0.5)))
        ((> v 50))
    (setq
      *y-a-inf*  (nconc *y-a-inf* (list (y-a-inf v)))
      *x-a-inf* (nconc *x-a-inf* (list (x-a-inf v)))
      *y-a-eff*  (nconc *y-a-eff* (list (y-a-eff (y-a-inf v))))
      *x-a-eff* (nconc *x-a-eff* (list (x-a-eff (x-a-inf v))))
      *t-x-a* (nconc *t-x-a* (list (t-x-a v)))
      *t-y-a* (nconc *t-y-a* (list (t-y-a v)))
      *volts* (nconc *volts* (list v))
      *g-a-inf* (nconc *g-a-inf* (list (g-a 1.0 (x-a-inf v)(y-a-inf v))))
) ))

;;; A-CURRENT
(defun a-current (x-a y-a v)
  (* (g-a (aref SOMA gbar-a$) x-a y-a) (- v *e-k)))

;;; G-A
(defun g-a ( gbar-a x-a y-a)
  (* gbar-a x-a  x-a x-a y-a  ))

;;;X-A-EFF, Y-A-EFF
(defun x-a-eff (x-a)
  (^ x-a 3.0))
(defun y-a-eff (y-a)
  (^ y-a 1.0))


;;; Persistent Slow Na current ***************
;;;
;;; As reported by French and Gage, 1985
;;;
;;; For cat neocortical cells, Staffstrom Schwindt Chubb and Crill (1985) report Inap
;;;   Activates within 2 to 4 mS, over the range measured (~-70 to ~-30mV)
;;;   Activates at about 3-4mV above rest.

(defvar *gbar-nap .01)                          ;Max. conductance [microS], as measured by French and Gage.

(defvars-w-value (*v-half-napx -49.0) (*alpha-base-rate-napx 0.04)
  (*beta-base-rate-napx 0.04) (*valence-napx 6.0) (*gamma-napx 0.30)(*base-txnap 1.0))
```

```
;;; MENU-FOR-NAP-CURRENT
(defun menu-for-NaP-current ()
  (tv:choose-variable-values
    '((*gbar-nap "Persistant Na current conductance [microS]" :number)
      ""
           " X Variable Kinetics "
      ""
      (*v-half-napx "V/12 for Nap x" :number)
      (*alpha-base-rate-napx "Alpha/beta-base value for Nap x at V1/2" :number)
      (*valence-napx "Valence for Nap x" :number)
      (*gamma-napx "Gamma for Nap x" :number)
      (*base-txnap "Base value for tau nap x" :number)))
  (setq *beta-base-rate-napx  *alpha-base-rate-napx)
  (aset *gbar-nap SOMA gbar-nap$))

;;; GBAR-NAP
(defun gbar-nap (area)                          ;total nap-channel conductance (microS)
  (* *gbar-nap-dens area 1.0e3))                ;(area is in sq-cm)

;;; X-NAP-INF
(defun x-nap-inf (voltage)
    (let ((a (alpha voltage *v-half-napx *alpha-base-rate-napx *valence-napx *gamma-napx))      ;
          (b (beta voltage *v-half-napx *alpha-base-rate-napx *valence-napx *gamma-napx)))
      (// a (+ a b))))
;(let ((midpoint -49.0)(slope 4.5))
;  (// 1.0 (+ 1.0 (exp (// (- midpoint voltage) slope))))))

;;; T-X-NAP
(defun t-x-nap (voltage)
    (let* ((a (alpha voltage *v-half-napx *alpha-base-rate-napx *valence-napx *gamma-napx))
           (b (beta voltage *v-half-napx *alpha-base-rate-napx *valence-napx *gamma-napx))
           (tx (// 1.0 (+ a b))))
      (if (< tx *base-txnap) *base-txnap tx)))
;  (cond ((> voltage 0.0)
;         (* *qten-factor-at-22 1.0))
;        ((> voltage -24.0)
;         (* *qten-factor-at-22 4.0))
;        (t
;         (* *qten-factor-at-22 40.0)))) ;approx. 18mS - F&G fig.1

;;; NAP-CURRENT
(defun nap-current (gbar-nap x-nap v)
  (* gbar-nap x-nap  (- v *e-na)))

;;; PLOT-NAP
(defvars *x-nap-inf* *t-x-nap*)
(defun plot-nap ()
  (menu-for-nap-current)
  (setq *x-nap-inf* nil *volts* nil *t-x-nap* nil)
  (let ((dv .50))
    (do ((v -100.0 (+ v dv)))
        ((> v 50.0))
      (setq *x-nap-inf* (nconc *x-nap-inf* (list (x-nap-inf v)))
            *t-x-nap* (nconc *t-x-nap* (list (t-x-nap v)))
            *volts* (nconc *volts* (list v)) ) )))


;;;  Fast Na-current **********

;;; Original estimates for the parameters of the three conductances are derived from single Na only spike record
;;; (24 degrees C) and the Na only repetitive records (27 degrees C and 32 degrees C). All Qten's are derived
;;; from a reference of 24 degrees C. Gating particle kinetics have a Qten of 3; conductance Qten's are set to
;;; 1.5.

(defvar *gbar-na1-dens 40.0)                    ;conductance density, mS⦶(cm-squared)
(defvar *gbar-na2-dens 1)                       ;conductance density, mS⦶(cm-squared)
(defvar *gbar-na3-dens 35.0)                    ;conductance density, mS⦶(cm-squared)
(defvar *gbar-nad-dens 20.0)                    ;dendrite conductance density, mS⦶(cm-squared)

(defvars-w-value (*na-choose 3))

(defvars-w-value (*v-half-m1 -47.0) (*base-rate-m1 0.3)  (*valence-m1 20.0) (*gamma-m1 0.50)
                 (*v-half-h1 -54.0) (*base-rate-h1 0.003)  (*valence-h1 30.0) (*gamma-h1 0.2)

                 (*v-half-m2 -5.0) (*base-rate-m2 0.025)  (*valence-m2 8) (*gamma-m2 .95)
                 (*v-half-h2 -47) (*base-rate-h2 0.0016667)  (*valence-h2 6) (*gamma-h2 0.2)

                 (*v-half-m3 -34.0) (*base-rate-m3 0.6667)  (*valence-m3 6.0) (*gamma-m3 0.50)
                 (*v-half-h3 -42.50) (*base-rate-h3 0.0023333)  (*valence-h3 30.0) (*gamma-h3 0.17)

                 (*base-tm1 0.50)(*base-th1 2.0) (*base-tm2 5)(*base-th2 3.00) (*base-tm3 0.40)(*base-th3 3.0))
```

```
;;; MENU-FOR-NA-CURRENT
(defun menu-for-Na-current ()
  (tv:choose-variable-values
    '(""
      "  ******************** NA 1 CURRENT ******************"
      ""
      (*gbar-na1-dens "Na 1 current conductance density (std =35) [mS/sq-cm]" :number)
      ""
      " M Variable Kinetics "
      ""
      (*v-half-m1 "V/12 for Na 1 m" :number)
      (*base-rate-m1 "Alpha-base value for Na 1 m at V1/2" :number)
      (*valence-m1 "Valence for Na 1 m" :number)
      (*gamma-m1 "Gamma for Na 1 m" :number)
      (*base-tm1 "Minimum value of time constant [msec]" :number)
      ""
      " H Variable Kinetics "
      ""
      (*v-half-h1 "V/12 for Na 1 h" :number)
      (*base-rate-h1 "Alpha-base value for Na 1 h at V1/2" :number)
      (*valence-h1 "Valence for Na 1 h" :number)
      (*gamma-h1 "Gamma for Na 1 h" :number)
      (*base-th1 "Minimum value of time constant [msec]" :number)
     ""
      "  ******************** NA 2 CURRENT ******************"
      ""
      (*gbar-na2-dens "Na 2 current conductance density (std =1) [mS/sq-cm]" :number)
      ""
      " M Variable Kinetics "
      ""
      (*v-half-m2 "V/12 for Na 2 m" :number)
      (*base-rate-m2 "Alpha-base value for Na 2 m at V1/2" :number)
      (*valence-m2 "Valence for Na 2 m" :number)
      (*gamma-m2 "Gamma for Na 2 m" :number)
      (*base-tm2 "Minimum value of time constant [msec]" :number)
      ""
      " H Variable Kinetics "
      ""
      (*v-half-h2 "V/12 for Na 2 h" :number)
      (*base-rate-h2 "Alpha-base value for Na 2 h at V1/2" :number)
      (*valence-h2 "Valence for Na 2 h" :number)
      (*gamma-h2 "Gamma for Na 2 h" :number)
      (*base-th2 "Minimum value of time constant [msec]" :number)
      ""
      "  ******************** NA 3 CURRENT ******************"
      ""
      (*gbar-na3-dens "Na 3 current conductance density (std =35) [mS/sq-cm]" :number)
      ""
      " M Variable Kinetics "
      ""
      (*v-half-m3 "V/12 for Na 3 m" :number)
      (*base-rate-m3 "Alpha-base value for Na 3 m at V1/2" :number)
      (*valence-m3 "Valence for Na 3 m" :number)
      (*gamma-m3 "Gamma for Na 3 m" :number)
      (*base-tm3 "Minimum value of time constant [msec]" :number)
      ""
      " H Variable Kinetics "
      ""
      (*v-half-h3 "V/12 for Na 3 h" :number)
      (*base-rate-h3 "Alpha-base value for Na 3 h at V1/2" :number)
      (*valence-h3 "Valence for Na 3 h" :number)
      (*gamma-h3 "Gamma for Na 3 h" :number)
      (*base-th3 "Minimum value of time constant [msec]" :number))
    ':label "Standard-spike A; Na1=trigger, Na2=slow tail, Na3=rep."))



;;; A-M-NA
(defun a-m-na (flag voltage)
  (cond ((and (= flag 1)(= *na-choose 3))       (a-m-na1-hippo  voltage ))
        ((and (= flag 2)(= *na-choose 3))       (a-m-na2-hippo  voltage ))
        ((and (= flag 3)(= *na-choose 3))       (a-m-na3-hippo  voltage ))))

;;; B-M-NA
(defun b-m-na (flag voltage)
  (cond ((and (= flag 1)(= *na-choose 3))       (b-m-na1-hippo  voltage ))
        ((and (= flag 2)(= *na-choose 3))       (b-m-na2-hippo  voltage ))
        ((and (= flag 3)(= *na-choose 3))       (b-m-na3-hippo  voltage ))))

;;; A-H-NA
```

```
(defun a-h-na (flag voltage)
  (cond ((and (= flag 1)(= *na-choose 3))       (a-h-na1-hippo  voltage ))
        ((and (= flag 2)(= *na-choose 3))       (a-h-na2-hippo  voltage ))
        ((and (= flag 3)(= *na-choose 3))       (a-h-na3-hippo  voltage ))))

;;; B-H-NA
(defun b-h-na (flag voltage)
  (cond ((and (= flag 1)(= *na-choose 3))       (b-h-na1-hippo  voltage ))
        ((and (= flag 2)(= *na-choose 3))       (b-h-na2-hippo  voltage ))
        ((and (= flag 3)(= *na-choose 3))       (b-h-na3-hippo  voltage ))))

;;; M-NA-INF
(defun m-na-inf (flag)
  (cond ((= 1 flag)      (// *a-m-na1  (+ *a-m-na1 *b-m-na1)))
        ((= 2 flag)      (// *a-m-na2  (+ *a-m-na2 *b-m-na2)))
        ((= 3 flag)      (// *a-m-na3  (+ *a-m-na3 *b-m-na3)))))

;;; T-M-NA
(defun t-m-na (flag)
  (let ((tm
         (cond ((= 1 flag) (//   1.0    (+ *a-m-na1 *b-m-na1)))
               ((= 2 flag) (//   1.0    (+ *a-m-na2 *b-m-na2)))
               ((= 3 flag) (//   1.0    (+ *a-m-na3 *b-m-na3))))))
    (* *qten-factor-at-24  (cond ((= 1 flag) (if (< tm *base-tm1) *base-tm1 tm))
                                 ((= 2 flag) (if (< tm *base-tm2) *base-tm2 tm))
                                 ((= 3 flag) (if (< tm *base-tm3) *base-tm3 tm))))))

;;; H-NA-INF
(defun h-na-inf (flag)
  (cond ((= 1 flag) (// *a-h-na1 (+ *a-h-na1 *b-h-na1)))
        ((= 2 flag) (// *a-h-na2 (+ *a-h-na2 *b-h-na2)))
        ((= 3 flag) (// *a-h-na3 (+ *a-h-na3 *b-h-na3)))))

;;; T-H-NA
(defun t-h-na (flag)
  (let ((th
         (cond ((= 1 flag) (//   1.0  (+ *a-h-na1 *b-h-na1)))
               ((= 2 flag) (//   1.0  (+ *a-h-na2 *b-h-na2)))
               ((= 3 flag) (//   1.0  (+ *a-h-na3 *b-h-na3))))))
    (* *qten-factor-at-24  (cond ((= 1 flag) (if (< th *base-th1) *base-th1 th))
                                ((= 2 flag) (if (< th *base-th2) *base-th2 th))
                                ((= 3 flag) (if (< th *base-th3) *base-th3 th))))))

;;; GBAR-NA
(defun gbar-na (flag area)                              ;total na-channel conductance (microS)
  (* *qten-g-24 (cond ((= 1 flag) (* *gbar-na1-dens area 1.0e3))
                      ((= 2 flag) (* *gbar-na2-dens area 1.0e3))
                      ((= 3 flag) (* *gbar-na3-dens area 1.0e3)))))


;;; GBAR-NAD
(defun gbar-nad (area)                          ;total dendrite na-channel conductance (microS)
  (* *gbar-nad-dens area 1.0e3))

;;;  WHEN EDITING POWERS OF GATING PARTICLES, ALSO EDIT APPROPRIATE M-EFF AND H-EFF FUNCTIONS

;;; NA1-CURRENT
(defun na1-current (gbar-na m-na h-na v)  (* (g-na1 gbar-na m-na h-na) (- v *e-na)))

;;; G-NA1
(defun g-na1 (gbar-na m-na h-na)  (* gbar-na   m-na  h-na h-na))

;;; M-EFF-NA1, H-EFF-NA1
(defun m-eff-na1 (m-na)  (^ m-na 1.0))
(defun h-eff-na1 (h-na)  (^ h-na 2.0))


;;; NA2-CURRENT
(defun na2-current (gbar-na m-na h-na v)  (* (g-na2 gbar-na m-na  h-na) (- v *e-na)))

;;; G-NA2
(defun g-na2 (gbar-na m-na h-na )  (* gbar-na   m-na  h-na))

;;; M-EFF-NA2, H-EFF-NA2
(defun m-eff-na2 (m-na)  (^ m-na 1.0))
(defun h-eff-na2 (h-na)  (^ h-na  1.0))


;;; NA3-CURRENT
(defun na3-current (gbar-na m-na h-na v)  (* (g-na3 gbar-na m-na h-na) (- v *e-na)))

;;; G-NA3
```

```
(defun g-na3 (gbar-na m-na h-na)  (* gbar-na   m-na m-na h-na h-na h-na ))

;;; M-EFF-NA3, H-EFF-NA3
(defun m-eff-na3 (m-na)  (^ m-na 2.0))
(defun h-eff-na3 (h-na)  (^ h-na 3.0))


;;; Generic Na Kinetics ******* Basic functions are from Traub et al.
(defvar *e-ref-na 70.0)
(defvar faraday 9.65e4)                          ;C/mol
(defvar temperature! 298.0)                      ;degrees Kelvin
(defvar R 8.31)                                  ;J/mol*K


;;; A-M-NA1-HIPPO
(defun a-m-na1-hippo (voltage)
  (alpha voltage *v-half-m1 *base-rate-m1 *valence-m1 *gamma-m1))

;;; B-M-NA1-HIPPO
(defun b-m-na1-hippo (voltage)
  (beta voltage *v-half-m1 *base-rate-m1 *valence-m1 *gamma-m1))

;;; A-H-NA1-HIPPO
(defun a-h-na1-hippo (voltage)
  (beta voltage *v-half-h1 *base-rate-h1 *valence-h1 *gamma-h1))

;;; B-H-NA1-HIPPO
(defun b-h-na1-hippo (voltage)
  (alpha voltage *v-half-h1 *base-rate-h1 *valence-h1 *gamma-h1))

;;; A-M-NA2-HIPPO
(defun a-m-na2-hippo (voltage)
  (alpha voltage *v-half-m2 *base-rate-m2 *valence-m2 *gamma-m2))

;;; B-M-NA2-HIPPO
(defun b-m-na2-hippo (voltage)
  (beta voltage *v-half-m2 *base-rate-m2 *valence-m2 *gamma-m2))

;;; A-H-NA2-HIPPO
(defun a-h-na2-hippo (voltage)
  (beta voltage *v-half-h2 *base-rate-h2 *valence-h2 *gamma-h2))

;;; B-H-NA2-HIPPO
(defun b-h-na2-hippo (voltage)
  (alpha voltage *v-half-h2 *base-rate-h2 *valence-h2 *gamma-h2))

;;; A-M-NA3-HIPPO
(defun a-m-na3-hippo (voltage)
  (alpha voltage *v-half-m3 *base-rate-m3 *valence-m3 *gamma-m3))

;;; B-M-NA3-HIPPO
(defun b-m-na3-hippo (voltage)
  (beta voltage *v-half-m3 *base-rate-m3 *valence-m3 *gamma-m3))

;;; A-H-NA3-HIPPO
(defun a-h-na3-hippo (voltage)
  (beta voltage *v-half-h3 *base-rate-h3 *valence-h3 *gamma-h3))

;;; B-H-NA3-HIPPO
(defun b-h-na3-hippo (voltage)
  (alpha voltage *v-half-h3 *base-rate-h3 *valence-h3 *gamma-h3))


;;; NA1-PLOT, NA2-PLOT, NA3-PLOT
(defvars *m-inf1* *t-h1* *h-inf1* *t-m1*
         *m-inf2* *t-h2* *h-inf2* *t-m2*
         *m-inf3* *t-h3* *h-inf3* *t-m3*
         *volts* *g-na1-inf* *g-na2-inf* *g-na3-inf*
         *m-eff1*  *h-eff1* *m-eff2*  *h-eff2* *m-eff3*  *h-eff3*)

(defun na1-plot ()
  (menu-for-na-current)
  (setq *m-inf1* nil *t-m1* nil *h-inf1* nil *t-h1* nil *g-na1-inf* nil *volts* nil *h-eff1* nil *m-eff1* nil)
  (do* ((v -100.0 (+ v .50))
        (m)(h))
       ((>  v  50.0))
    (setq *a-m-na1 (a-m-na 1 v) *b-m-na1 (b-m-na 1 v)
          *a-h-na1 (a-h-na 1 v) *b-h-na1 (b-h-na 1 v)
          m (m-na-inf 1) h (h-na-inf 1)
          *volts* (nconc *volts* (list v))
          *m-inf1* (nconc *m-inf1* (list m)) *h-inf1* (nconc *h-inf1* (list h))
```

```
        *m-eff1* (nconc *m-eff1* (list (m-eff-na1 m))) *h-eff1* (nconc *h-eff1* (list (h-eff-na1 h)))
        *g-na1-inf* (nconc *g-na1-inf* (list (g-na1 1.0 (m-na-inf 1)(h-na-inf 1))))
        *t-m1* (nconc *t-m1* (list (t-m-na 1))) *t-h1* (nconc *t-h1* (list (t-h-na 1))))))


(defun na2-plot ()
  (menu-for-na-current)
  (setq *m-inf2* nil *t-m2* nil *h-inf2* nil *t-h2* nil *g-na2-inf* nil *volts* nil *h-eff2* nil *m-eff2* nil)
  (do* ((v -100.0 (+ v .50))
        (m)(h))
       ((> v 50.0))
    (setq *a-m-na2 (a-m-na 2 v) *b-m-na2 (b-m-na 2 v)
          *a-h-na2 (a-h-na 2 v) *b-h-na2 (b-h-na 2 v)
          m (m-na-inf 2) h (h-na-inf 2)
          *volts* (nconc *volts* (list v))
          *m-inf2* (nconc *m-inf2* (list m)) *h-inf2* (nconc *h-inf2* (list h))
          *m-eff2* (nconc *m-eff2* (list (m-eff-na2 m))) *h-eff2* (nconc *h-eff2* (list (h-eff-na2 h)))
          *g-na2-inf* (nconc *g-na2-inf* (list (g-na2 2.0 (m-na-inf 2)(h-na-inf 2))))
          *t-m2* (nconc *t-m2* (list (t-m-na 2))) *t-h2* (nconc *t-h2* (list (t-h-na 2))))))


(defun na3-plot ()
  (menu-for-na-current)
  (setq *m-inf3* nil *t-m3* nil *h-inf3* nil *t-h3* nil *g-na3-inf* nil *volts* nil *h-eff3* nil *m-eff3* nil)
  (do* ((v -100.0 (+ v .50))
        (m)(h))
       ((> v 50.0))
    (setq *a-m-na3 (a-m-na 3 v) *b-m-na3 (b-m-na 3 v)
          *a-h-na3 (a-h-na 3 v) *b-h-na3 (b-h-na 3 v)
          m (m-na-inf 3) h (h-na-inf 3)
          *volts* (nconc *volts* (list v))
          *m-inf3* (nconc *m-inf3* (list m)) *h-inf3* (nconc *h-inf3* (list h))
          *m-eff3* (nconc *m-eff3* (list (m-eff-na3 m))) *h-eff3* (nconc *h-eff3* (list (h-eff-na3 h)))
          *g-na3-inf* (nconc *g-na3-inf* (list (g-na3 3.0 (m-na-inf 3)(h-na-inf 3))))
          *t-m3* (nconc *t-m3* (list (t-m-na 3))) *t-h3* (nconc *t-h3* (list (t-h-na 3))))))


;;; SOMATIC AND DENDRITIC Ca-CURRENT ***********

(defvar *gbar-Ca-dens 50.0)                    ;conductance density, mS/(cm-squared)
(defvar *gbar-Cad-dens 20.0)                   ;dendrite conductance density, mS/(cm-squared)
(defvar *base-tsca 2.0)
(defvar *base-twca 5.0)
(defvars-w-value (*v-half-s -24.0) (*base-rate-s .10)  (*valence-s 4.0) (*gamma-s 0.5)
                 (*v-half-w -35.0) (*base-rate-w 0.001)  (*valence-w 12.0) (*gamma-w 0.2))

(defvar *gbar-ca)


;;; MENU-FOR-CA-CURRENT
(defun menu-for-Ca-current ()
  (tv:choose-variable-values
    '((*gbar-Ca-dens "Ca current conductance density [mS/sq-cm]" :number)
      ""
      " S Variable Kinetics "
      ""
      (*v-half-s "V/12 for Ca  s" :number)
      (*base-rate-s "Alpha-base value for Ca  s at V1/2" :number)
      (*valence-s "Valence for Ca  s" :number)
      (*gamma-s "Gamma for Ca  s" :number)
      (*base-tsca "Minimum value of activation time constant [msec]" :number)
      ""
      " W Variable Kinetics "
      ""
      (*v-half-w "V/12 for Ca  w" :number)
      (*base-rate-w "Alpha-base value for Ca  w at V1/2" :number
                    :documentation "Increase makes gating particle faster")
      (*valence-w "Valence for Ca  w" :number)
      (*gamma-w "Gamma for Ca  w" :number)
      (*base-twca "Minimum value of inactivation time constant [msec]" :number)))
  (setq *gbar-ca (gbar-ca (surf-area *soma-radius))))


;;; K1-S-CA
(defun k1-s-ca (voltage)
  (alpha voltage *v-half-s *base-rate-s *valence-s *gamma-s))

;;; K2-S-CA
(defun k2-s-ca (voltage)
```

```
  (beta voltage *v-half-s *base-rate-s *valence-s *gamma-s))

;;; A-W-CA
(defun a-w-ca (voltage)
  (beta voltage *v-half-w *base-rate-w *valence-w *gamma-w))

;;; B-W-CA
(defun b-w-ca (voltage)
  (alpha voltage *v-half-w *base-rate-w *valence-w *gamma-w))


;;; MENU-FOR-CAD-CURRENT
(defun menu-for-Cad-current (DENDRITE-ARRAY segment)
  (let ((label (format nil "Ca current conductance density in ~A segment ~2d [mS//sq-cm]"
                       (aref DENDRITE-ARRAY 0 label$) (+ 1 segment))))
    (tv:choose-variable-values
      '((*gbar-cad-dens ,label :number)))
    (aset (* (aref DENDRITE-ARRAY segment length$) 3.14159 (aref DENDRITE-ARRAY segment diameter$)
             1.0e-8 *gbar-cad-dens)
          DENDRITE-ARRAY segment gbar-ca$)))

;;; GBAR-CA
(defun gbar-Ca (area)                          ;total Ca-channel conductance (microS)
  (* *gbar-Ca-dens area 1.0e3))

;;; GBAR-CAD
(defun gbar-Cad (area)                         ;total dendrite Ca-channel conductance (microS)
  (* *gbar-Cad-dens area 1.0e3))

;;; S-CA-INF
(defun s-ca-inf (v)
    (// (k1-s-ca v) (+ (k1-s-ca v)(k2-s-ca v))))

;;; T-S-CA
(defun t-s-ca (v)
  (let   ((tau (//   1.0 (+ (k1-s-ca v)(k2-s-ca v)))))
    (* *qten-factor-at-32 (if (< tau *base-tsca) *base-tsca tau))))

;;; W-CA-INF
(defun w-ca-inf (v)
    (// (a-w-ca v) (+ (a-w-ca v)(b-w-ca v))))

;;; T-W-CA
(defun t-w-ca (v)
  (let  ((tau (//  1.0 (+ (a-w-ca v)(b-w-ca v)))))
    (* *qten-factor-at-32 (if (< tau *base-twca) *base-twca tau))))

;;; CA-CURRENT
(defun ca-current (gbar-ca s-ca w-ca v)
  (* (g-ca gbar-ca s-ca w-ca)
     (- v (e-ca))))

;;; G-CA
(defun g-ca (gbar-ca s-ca w-ca)
  (if (< w-ca 0.001) (setq w-ca 0.0))
  (* gbar-Ca  s-ca s-ca w-ca w-ca w-ca w-ca))

;;;S-CA-EFF, W-CA-EFF
(defun s-ca-eff (s-ca)
  (if (< s-ca 0.001) 0.0
      (^ s-ca 2.0)))
(defun w-ca-eff (w-ca)
  (if (< w-ca 0.001) 0.0
  (^ w-ca 4.0)))

;;; CA-PLOT
(defvars *s-ca-inf* *w-ca-inf* *s-ca-eff* *w-ca-eff*
  *t-s-ca* *t-w-ca* *g-ca-inf*)
(defun ca-plot ()
  (menu-for-ca-current)
  (setq *volts* nil *w-ca-inf* nil *s-ca-inf* nil  *w-ca-eff* nil *s-ca-eff* nil
        *t-s-ca* nil *t-w-ca* nil *g-ca-inf* nil)
  (do   ((v -100.0 (+ v 0.5)))
        ((> v 50))
    (setq
      *w-ca-inf*  (nconc *w-ca-inf* (list (w-ca-inf v)))
      *s-ca-inf* (nconc *s-ca-inf* (list (s-ca-inf v)))
      *w-ca-eff*  (nconc *w-ca-eff* (list (w-ca-eff (w-ca-inf v))))
      *s-ca-eff* (nconc *s-ca-eff* (list (s-ca-eff (s-ca-inf v))))
      *t-s-ca* (nconc *t-s-ca* (list (t-s-ca v)))
      *t-w-ca* (nconc *t-w-ca* (list (t-w-ca v)))
      *volts* (nconc *volts* (list v))
```

```
      *g-ca-inf* (nconc *g-ca-inf* (list (g-ca 1.0 (s-ca-inf v)(w-ca-inf v)))))))


;;;  Persistant Calcium Current
;;;
;;; Ca,slow current as reported by Johnston, Hablitz, and Wilson.
;;;
;;; From deriving the IV curves of JH&W, it is determined that this current is due to a non-inactivating inward
;;; current with a reversal potential around 0mV. Thus it is unclear as to what species are actually comprising this
;;; current.

;;; E-CAS Empirically-derived reversal potential for the so-called slow "Ca" current.
(defvar *e-cas 0)

;;; conductance in micro-siemens
(defvar *gbar-cas .080)

;;; T-X-CAS Time constant for activation - ranges between 50 and 100 ms.
(defun t-x-cas (voltage)
  (ignore voltage)
  75.0)

;;; X-CAS-INF Steady state value for the activation variable.
(defun x-cas-inf (voltage)
  (let ((midpoint -30.0)(steepness 3.60))
    (// 1.0 (+ 1.0 (exp (// (- midpoint voltage) steepness))))))


;;; CAS-CURRENT
(defun cas-current (x-cas voltage)
  (* *gbar-cas x-cas (- voltage *e-cas)))


;;; CAS-PLOT
(defvars *x-cas-inf* *t-x-cas*)
(defun cas-plot ()
  (setq *x-cas-inf* nil *volts* nil *t-x-cas* nil)
  (do   ((v -100.0 (+ v 0.5)))
        ((> v 50.0))
    (setq *x-cas-inf* (nconc *x-cas-inf* (list (x-cas-inf v)))
;          *t-x-cas* (nconc *t-x-cas* (list (t-x-cas v)))
          *volts* (nconc *volts* (list v)) ) ))



;;; Ca++SHELL CONCENTRATION AND Ca++REVERSAL POTENTIAL SYSTEM
;;;
;;; Initially we will use a simple description of Ca++accumulation in a thin shell just underneath the cell
;;; membrane. This model will include the contribution of the total Ca++currents (transient and slow), and the first
;;; order removal of Ca++via some combination of diffusion and/or binding that is expressed with a single rate
;;; constant. This treatment is modelled after Traub and Llinas.
;;;
;;; New version with two concentric shells -
;;;
;;; [Ca++]sh-dot = {K * (sum of Ca++currents)} -  { ([Ca++]sh - [Ca++]sh2) / t-ca-conc }
;;;
;;; [Ca++]sh2-dot =   { ([Ca++]sh2 - [Ca++]sh) / t-ca-conc }- { [Ca++]sh2 / t2-ca-conc }

(defvar *faraday 9.648e4)                       ;Coulombs/mole
(defvar *R 8.314)                               ;Gas constant -- (Volts*Coulombs)/(DegreesKelvin*mole)
(defvar *core-conc 50.0e-6)                     ;mM
(defvar *ca-conc-shell1-rest 50.0e-6)
(defvar *ca-conc-shell2-rest 50.0e-6)
(defvars-w-value (*shell-depth 0.25)                        ;microns
                 (*core-volume (* 4.0 3.1415 *soma-radius *soma-radius *soma-radius 1.0e-12))    ;volume of core in ml
  (*Ficks-shell-shell 2.00e-11)         ;Modified Fick's constant between shells, cm^3/msec
  (*Ficks-shell-core 4.0e-7)    ;Fick's constant between shells and core, cm/msec
  (*alpha-shell 0.001))                         ;Fraction of soma shell assigned to shell 1

;(defvars-w-value (*shell1-vol
;                  (* *shell-depth 1.33333 3.1415 1.0e-12
;                     *soma-radius *soma-radius *shell1-shell2-ratio))  ;shell 1 volume in ml
;                 (*shell2-vol
;                   (* *shell-depth 1.33333 3.1415 1.0e-12
;                      *soma-radius *soma-radius (- 1.0 *shell1-shell2-ratio))))  ;shell 2 voulme in ml

(defun ca-conc-shell-dot (total-ca-current shell1-conc shell2-conc)     ;New version for two shells.
  (* *dt (+ (// (* -1.0 total-ca-current 1.0e-6)
                (* 2.0  *faraday
```

```
                *shell-depth 1.0e-4
                *soma-radius *soma-radius 1.3333 3.14159 1.0e-8 *alpha-shell))
          (// (* *Ficks-shell-shell (- shell2-conc shell1-conc ))
              (* *shell-depth 1.0e-4
                *soma-radius *soma-radius 1.3333 3.14159 1.0e-8 *alpha-shell))
          (// (* *Ficks-shell-core (- *core-conc shell1-conc))
              (* *shell-depth 1.0e-4)))))

(defun ca-conc-shell2-dot (shell1-conc shell2-conc)
  (* *dt (+ (// (* *Ficks-shell-shell (- shell1-conc shell2-conc ))
                (* *shell-depth 1.0e-4
                  *soma-radius *soma-radius 1.3333 3.14159 1.0e-8 (- 1.0 *alpha-shell)))
            (// (* *Ficks-shell-core (- *core-conc shell2-conc))
                (* *shell-depth 1.0e-4)))))

(defun e-ca ()
  (* .04299 (+ *temperature 273.0) (log (// *ca-conc-extra (aref SOMA ca-conc-shell$)))))


;;; PLOT-IV

(defun plot-iv ()
  (setq *iv-current* nil *iv-voltage* nil)
  (menu-for-soma-geometry-and-passive-components)
  (menu-for-soma-currents)
  (do   ((voltage -90.0 (+ voltage 0.5)))((> voltage 50.0))
    (let ((voltage-index (+ 1000 (fixr (* 10 voltage)))))
      (setq *iv-current*
            (nconc *iv-current*
                   (list (+ (if *include-dr (dr-current (aref x-dr-inf-array voltage-index) 1.0
                                                        ;(aref y-dr-inf-array voltage-index)
                                                        voltage) 0)
                            (if *include-c (c-current (aref x-c-inf-array voltage-index)
                                                      (aref y-c-inf-array voltage-index) voltage) 0)
                            (if *include-q (* (aref SOMA gbar-q$)(aref x-q-inf-array voltage-index)(- voltage *e-k)
) 0.0)
                            (if *include-m (m-current (aref x-m-inf-array voltage-index) voltage) 0 )
                            (if *include-a (a-current (aref x-a-inf-array voltage-index)
                                                      (aref y-a-inf-array voltage-index) voltage) 0)
                            (if *include-na1 (na1-current (aref SOMA gbar-na1$)(aref m-na1-inf-array voltage-index)
                                                          (aref h-na1-inf-array voltage-index)
                                                          voltage) 0)
                            (if *include-na2 (na2-current (aref SOMA gbar-na2$)(aref m-na2-inf-array voltage-index)
                                                          (aref h-na2-inf-array voltage-index)
                                                          voltage) 0)
                            (if *include-na3 (na3-current (aref SOMA gbar-na3$)(aref m-na3-inf-array voltage-index)
                                                          (aref h-na3-inf-array voltage-index)
                                                          voltage) 0)
                            (if *include-nap (nap-current (aref SOMA gbar-nap$) (aref x-nap-inf-array voltage-index
)
                                                          voltage) 0 )
                            (if *include-ca (ca-current (aref SOMA gbar-ca$)(aref s-ca-inf-array voltage-index)
                                                        (aref w-ca-inf-array voltage-index) voltage) 0 )
                            (if *include-cas (cas-current (aref x-cas-inf-array voltage-index) voltage) 0)
                            (l-current voltage)
                            (if *include-shunt (* *g-electrode voltage) 0)
                            (- *i-constant-injection))))
            *iv-voltage* (nconc *iv-voltage* (list voltage)) ) )))
```

Bibliography

[1] P. Adams, D. Brown, and A. Constanti. M-currents and other potassium currents in bullfrog sympathetic neurones. *Journal of Physiology*, 330:537–572. 1982.

[2] P Adams. C. Koch. and et al. Modelling of bullfrog sympathetic ganglion cells. 1986. In preparation.

[3] R. Aldrich. Voltage dependent gating of sodium channels: towards an integrated approach. *Trends In Neuroscience*. Feb 1986.

[4] R. W. Aldrich. D. P. Corey. and C. F. Stevens. A reinperpretation of mammalian sodium channel gating based on single channel recording. *Nature*. 306:436–441. Dec 1983.

[5] T.J. Blaxter. P.L. Carlen. M.L. Davies. and P.W Kujtan. Gamma-aminobutryic acid hyperpolarizes rat hippocampal pyramidal cells through a calcium-dependent potassium conductance. *Journal of Physiology*. 373:191–194, 1986.

[6] D. Brown and W. Griffith. Persistant slow inward calcium current in voltage clamped hippocampal neurones of the guinea pig. *Journal of Physiology*. 337:303–320. 1983.

[7] T. H. Brown. R. A. Fricke. and D. H. Perkel. Passive electrical constants in three classes of hippocampal neurons. *Journal of Neurophysiology*. 46(4):812–827. Oct 1981.

[8] W. Catterall. Voltage-dependent gating of sodium channels: correlating structure and function. *Trends in Neuroscience*. Jan 1986.

[9] S. Chiu. Inactivation of sodium channels: second order kinetics in myelinated nerve. *Journal of Physiology*. 1977.

[10] S.Y. Chiu, J.M. Ritchie, R.B. Rogart, and D. Stagg. A quantitative description of membrane currents in rabbit myelinated nerve. *Journal of Physiology*, 292:149–166, 1979.

[11] J. Cooley and F. Dodge. Digital computer solutions for excitation and propagation of the nerve impulse. *Biophysics Journal*, 6:, 1966.

[12] C. French and P. Gage. A threshold sodium current in pyramidal cells in rat hippocampus. *Neuroscience Letters*, 56:289–293, 1985.

[13] R. J. Gould, K. M. M. Murphy, and S. H. Snyder. Autoradiographic localization of calcium channel antagonist receptors in rat brain with [3-h]nitrendipine. *Brain Research*, 330:217–223, 1985.

[14] B. Gustafsson, M. Galvan, H. Grafe, and H. Wigstrom. A transient outward current in a mammalian central neurone blocked by 4-aminopyridine. *Nature*, 299:252–254, Sep 1982.

[15] J. Halliwell. Caesium loading reveals two distinct ca-currents in voltage clamped guinea pig hippocampal neurones in vitro. *Journal of Physiology*, 10P–11P, Mar 1983.

[16] J. Halliwell and P. Adams. Voltage clamp analysis of muscarinic excitation in hippocampal neurons. *Brain Research*, 250:71–92, 1982.

[17] J. V. Halliwell, I. B. Othman, A. Pelchen-Matthews, and J. O. Dolly. Central action of dedrotoxin: selective reduction of a fast transient potassium conductance in hippocampus and binding to localized acceptors. *Proc. Natl. Acad. Sci. USA*, 83:493–497, Jan 1986.

[18] O. P. Hamill, J. R. Huguenard, and D. A. Prince. Selective calcium channel expression in pyramidal and nonpyramidal cells of the rat neocortex. *Biophysical Journal*, 51:223a, 1987.

[19] B. Hille. *Ionic Channels of Excitable Membranes*. Sinauer Associates, Sunderland, Massachusetts, 1984.

[20] A. L. Hodgkin and A. F. Huxley. The components of membrane conductance in the giant axon of *Loligo*. *Journal of Physiology*, 116:473–96, 1952.

[21] A. L. Hodgkin and A. F. Huxley. Currents carried by sodium and potassium ions through the membrane of the giant axon of *Loligo*. *Journal of Physiology*, 116:449–72, 1952.

[22] A. L. Hodgkin and A. F. Huxley. The dual effect of membrane potential on sodium conductance in the giant axon of *Loligo*. *Journal of Physiology*, 116:497–506, 1952.

[23] A. L. Hodgkin and A. F. Huxley. A quantitative description of membrane current and its application to conduction and excitation in nerve. *Journal of Physiology*. 117:500–44, 1952.

[24] R. Isaacson and K. Pribram, editors. *The Hippocampus*. Volume 1, Plenum Press, 1975.

[25] R. Isaacson and K. Pribram, editors. *The Hippocampus*. Volume 2, Plenum Press, 1975.

[26] J. J. B. Jack, D. Noble, and R. W. Tsien. *Electric Current Flow In Excitable Cells*. Clarendon Press, Oxford, 1983.

[27] J. J. B. Jack and S. J. Redman. An electrical description of the motoneurone, and its application to the analysis of synaptic potentials. *Journal of Physiology*, 215:321–352, 1971.

[28] D. Johnston, J. Hablitz, and W. Wilson. Voltage clamp discloses slow inward current in hippocampal burst-firing neurones. *Nature*, (286):, 1980.

[29] K. Krnjevic, M. E. Morris, and N. Ropert. Changes in free calcium ion concentration recorded inside hippocampal pyramidal cells in situ. *Brain Research*, 374:1–11. 1986.

[30] B. Lancaster and P. Adams. Calcium dependent current generating the afterhyperpolarization of hippocampal neurons. *Journal of Neurophysiology*. 1986.

[31] D. Madison, R. Malenka, and R. Nicoll. A voltage dependent chloride current in hippocampal pyramidal cells is blocked by phorbol esters. *Biophysical Journal*, 49. 1986.

[32] D. Madison and R. Nicoll. Noradrenaline blocks accommodation of pyramidal cell discharge in the hippocampus. *Nature*. 299:, Oct 1982.

[33] D. V. Madison, A. P. Fox, and R. W. Tsien. Adenosine reduces an inactivating component of calcium current in hippocampal ca3 neurons. *Biophysical Journal*. 51(30a):, 1987.

[34] L. Masukawa and D. Prince. Synaptic control of excitability in isolated dendrites of hippocampal neurons. *Journal of Neuroscience*, 4(1):217–227, Jan 1984.

[35] Y. Nakajima, S. Nakajima, R. J. Leonard, and K. Yamaguchi. Acetylcholine raises excitability by inhibiting the fast transient potassium current in cultured hippocampal neurons. *Proc. Natl. Acad. Sci. USA*, 83:3022–3026, May 1986.

[36] Y. Nakajuma, S. Nakajima, R. Leonard, and K. Yamaguchi. Actetylcholine inhibits a-current in dissociated cultured hippocampal neurons. *Biophysical Journal*, 49:575a, 1986.

[37] A. Newell. Unified theories of cognition. *Williams James Lecture Series, Harvard University*, 1987.

[38] M. Nowycky, A. Fox, and R. Tsien. Three types of neuronal calcium channel with different calcium agonist sensitivity. *Nature*, 316:, Jan 1985.

[39] P. Sah, C. French, and P. Gage. Effects of noradrenaline on some potassium currents in cal neurones in rat hippocampal slices. *Neuroscience Letters*, 60:295–300, 1985.

[40] P. A. Schwartzkroin and M. Slawsky. Probable calcium spikes in hippocampal neurons. *Brain Research*, 135:157–161, 1977.

[41] M. Segal and J. Barker. Rat hippocampal neurons in culture: calcium and calcium-dependent potassium conductances. *Journal of Neurophysiology*, 55(4):751–766, Apr 1986.

[42] M. Segal and J. Barker. Rat hippocampal neurons in culture: potassium conductances. *Journal of Neurophysiology*, 51(6):, Jun 1984.

[43] M. Segal, M. Rogawski, and J. Barker. A transient potassium conductance regulates the excitability of cultured hippocampal and spinal neurons. *Journal of Neuroscience*, 4(2):604–609, Feb 1984.

[44] D. P. Shelton. Membrane resistivity estimated for the purkinje neuron by means of a passive computer model. *Neuroscience*, 14(1):111–131, 1985.

[45] T. Shimahara. Presynaptic modulation of transmitter release by the early outward potassium current in aplysia. *Brain Research*, 263:51–56, 1983.

[46] J. Storm. A-current and ca-dependent transient outward current control the initial repetitive firing in hippocampal neurons. *Biophysical Journal*, 49:369a, 1986.

[47] J. Storm. Mechanisms of action potential repolarization and a fast after-hyperpolarization in rat hippocampal pyramidal cells. ***Journal of Physiology***, 1986.

[48] R. Traub and R. Llinas. Hippocampal pyramidal cells : significance of dendritic ionic conductances for neuronal function and epileptogenesis. *Journal of Neurophysiology*, 42(2):. Mar 1979.

[49] R. Traub and R. Llinas. The spatial distribution of ionic conductance in normal and axotomized motorneurons. *Neuroscience*, 2:829–849. 1977.

[50] D. Turner. Conductance transients onto dendritic spines in a segmental cable model of hippocampal neurons. *Biophysics Journal*. 46:85–96, Jul 1984.

[51] D. Turner and Schwartzkroin P. *Brain Slices*. chapter 4. Plenum Publishing Corporation. 1984. Passive electrotonic structure and dendritic properties of hippocampal neurons.

[52] D. Turner and P. A. Schwartzkroin. Steady-state electrotonic analysis of intracellularly stained hippocampal neurons. *Journal of Neurophysiology*. 44(1):184–199. Jul 1980.

[53] R. K. S. Wong and D. A. Prince. Participation of calcium spikes during intrinsic burst firing in hippocampal neurons. *Brain Research*. 159:385–390. 1978.

[54] K. Zbicz and F. Weight. Transient voltage and calcium dependent outward currents in hippocampal ca3 pyramidal neurons. *Journal of Neurophysiology*. 53(4):. Apr 1985.

[55] K. Zbicz and F. Weight. Voltage-clamp analysis of inward calcium current in hippocampal ca3 pyramidal neurons. *Neuroscience Abstracts*, 11:520. 1986.