

Realistic Single Cell Modeling – from Experiment to Simulation

Dieter Jaeger

Department of Biology, Emory University, Atlanta, GA, USA, djaeger@emory.edu

urn:nbn:de:0009-3-2228

Abstract. This tutorial gives a step by step explanation of how one uses experimental data to construct a biologically realistic multicompartmental model. Special emphasis is given on the many ways that this process can be imprecise. The tutorial is intended for both experimentalists who want to get into computer modeling and for computer scientists who use abstract neural network models but are curious about biological realistic modeling. The tutorial is not dependent on the use of a specific simulation engine, but rather covers the kind of data needed for constructing a model, how they are used, and potential pitfalls in the process.

Keywords: Realistic modeling, compartmental modeling, Purkinje cell, computational neuroscience, GENESIS, tutorial

Citation: Jaeger D (2005). Realistic Single Cell Modeling - from Experiment to Simulation. Brains, Minds and Media, Vol. 1, bmm222 (urn:nbn:de:0009-3-2228).

Licence: Any party may pass on this Work by electronic means and make it available for download under the terms and conditions of the Digital Peer Publishing Licence. The text of the licence may be accessed and retrieved via Internet at http://www.dipp.nrw.de/lizenzen/dppl/dppl/DPPL_v2_en_06-2004.html.

Introduction

The objective of realistic single cell modeling is to approximate the electrophysiological behavior of biological neurons as closely as possible. However, a perfect match is not possible for several reasons, and each model should be viewed as a working hypothesis as to the integrative properties of the cell type under study. This article points out some of the basic tenets of realistic modeling, as well as possible pitfalls. These objectives are clarified in the course of a step by step discussion of how a realistic model is constructed. The talk is not focused on a particular simulator (the most popular ones being NEURON and GENESIS), but some specific examples of scripts are shown for GENESIS simulations in the Appendix.

Brain structures are made of microcircuits with specific cell types and connections. Each type of neuron in the brain is characterized by distinct morphological features, as well as distinct electrophysiological features (Fig.1). Realistic modeling therefore needs to take these individual properties into account, and a 'standard model neuron' (one shoe fits all) such as an integrate and fire model can not be used to adequately describe the dynamics of signal processing in biological neurons.

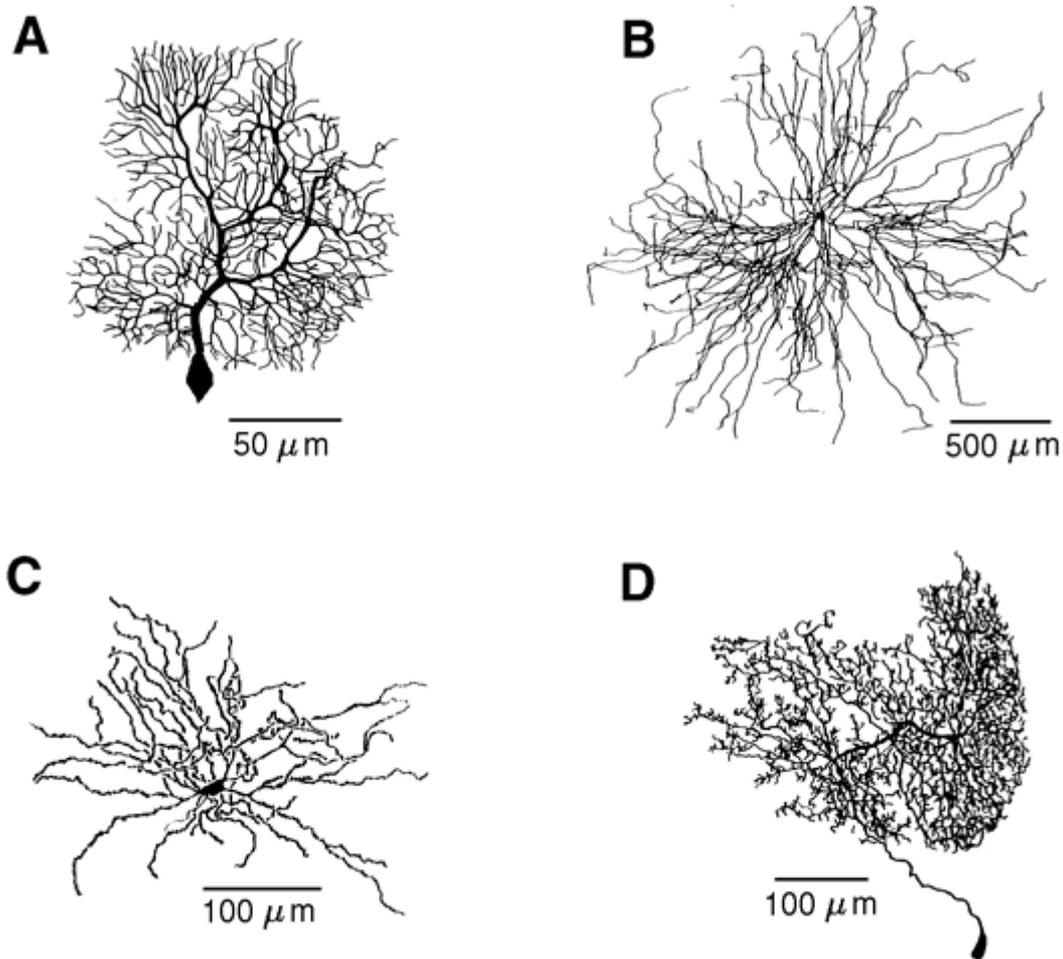


Figure 1: Different types of neurons. (A) Cerebellar Purkinje cell of the guinea pig (reconstructed by Moshe Rapp); (B) alpha-motoneuron from the cat spinal cord (reconstructed by Robert Burke); (C) Neostriatal spiny neuron from the rat (from M. A. Wilson); (D) Axonless interneurons of the locust (reconstructed by Giles Laurent). From [Segev \(1998\)](#).

The morphology of single neurons can be analyzed as to their branching structure, surface area, process diameters, and synaptic connectivity. These features have profound influences on the processing of synaptic input and the control of action potential generation. The examples shown here (Fig. 2) highlight that individual neurons receive thousands of synapses and the expected barrage of inputs can be processed in dendrite-specific ways.

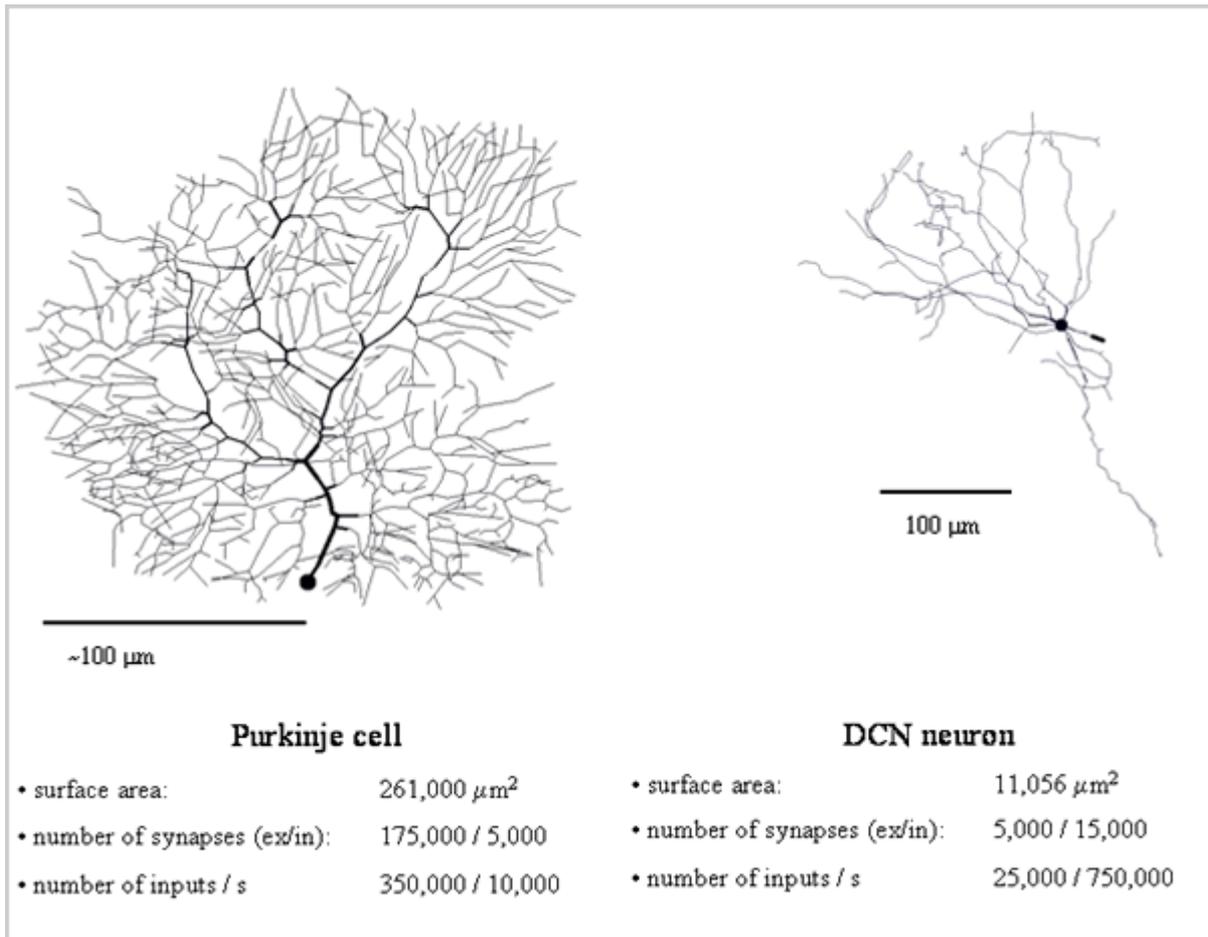
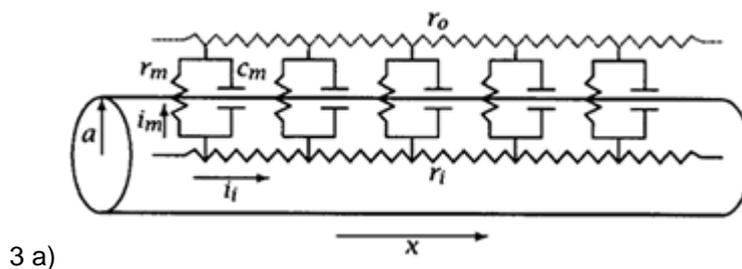
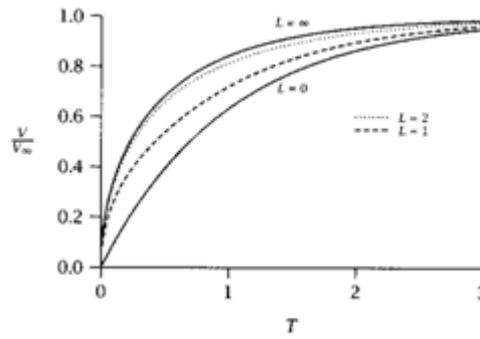


Figure 2: a) Purkinje Cell, b) DCN neuron; (ex/ in) = excitatory / inhibitory

The theoretical basis of compartmental modeling is given by cable theory, a set of well understood concepts originally introduced by Wilfried Rall (Fig. 3). A basic dictum of cable theory is that in passive cylinders (i.e. neuronal processes without voltage-gated conductances) signals decay over space and time in a mathematically understood fashion. The concepts of time constant (τ) and space constant (λ) simplify the temporal and spatial description of signal decay.





3 b)

Figure 3: a) “Diagram for current flow in a uniform cylinder such as an axon or segment of dendrite.” b) “Nonisopotential cell (cylinder). Comparison of normalized charging curves for finite cables of different electronic lengths. A step of current is injected and the voltage is measured, at $x = 0$. (After Lacj et al. 1975)”. Reprinted and cited from Johnston/Wu (1995), Fig. 4.6 and 4.16, with kind permission from MIT Press.

$$V_m(t, x) = C_0 e^{-t/\tau_0} + C_1 e^{-t/\tau_1} + C_2 e^{-t/\tau_2} + \dots + C_n e^{-t/\tau_n}$$

$$\lambda = \sqrt{\frac{r_m}{r_i}} = \sqrt{\frac{\hat{a}R_m}{2R_i}} \tag{1}$$

In practice, neurons have finite length processes with closed ends. Steady state voltage decays significantly when processes are of length > 1 space constant or so, which is typically the case in large neurons.

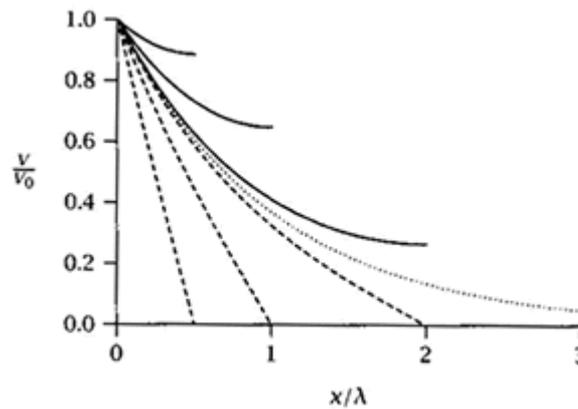


Figure 4: “Nonisopotential cell (cylinder). Comparison of voltage decays along finite cables of different electronic lengths and with end terminations. Current is injected at $x=0$. The solid lines are for finite cables with sealed ends, the dashed lines are for finite cables with open ends, and the dotted line is for dsemi-infinite cable (After Rall 1959.)”. Reprinted and cited from Johnston/Wu (1995), Fig. 4.15, with kind permission from MIT Press.

Importantly, signal decay in cylinders is frequency dependent, and signals with fast rise and fall time (such as action potentials) decay over much shorter distances than steady state potentials.

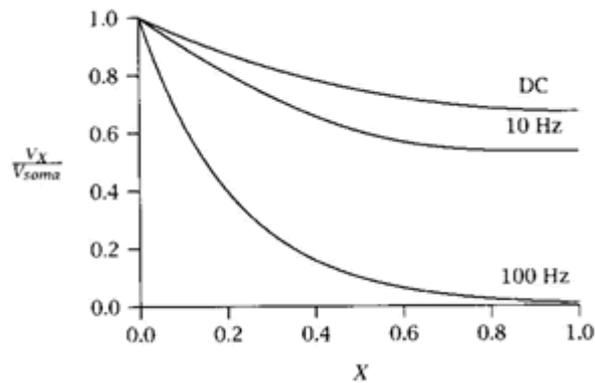


Figure 5: Voltage attenuation along a finite-length cable ($L=1$) for current injections (DC to 100 Hz) at $X=0$ (i.e., soma) ($R_m=50,000 \Omega\text{-cm}^2$). Reprinted and cited from Johnston/Wu (1995), Fig. 4.18. With kind permission from MIT Press.

Signal decay can be depicted graphically by 'morphoelectronic transforms'. Here it is demonstrated that signal decay is non-symmetrical and that voltage transients generated in the dendrites decay much more than voltage transients from the soma to the tip of the dendrites. This effect is due to the branching structure of the neuron and the increasing diameter of processes toward the soma.

Another important feature of signal decay is its dependence on input resistance, which itself decreases dramatically during barrages with synaptic input. Thus the shown morphoelectronic transforms (Fig. 6) would show much more pronounced signal decay yet if the neuron was subjected to a synaptic input barrage.

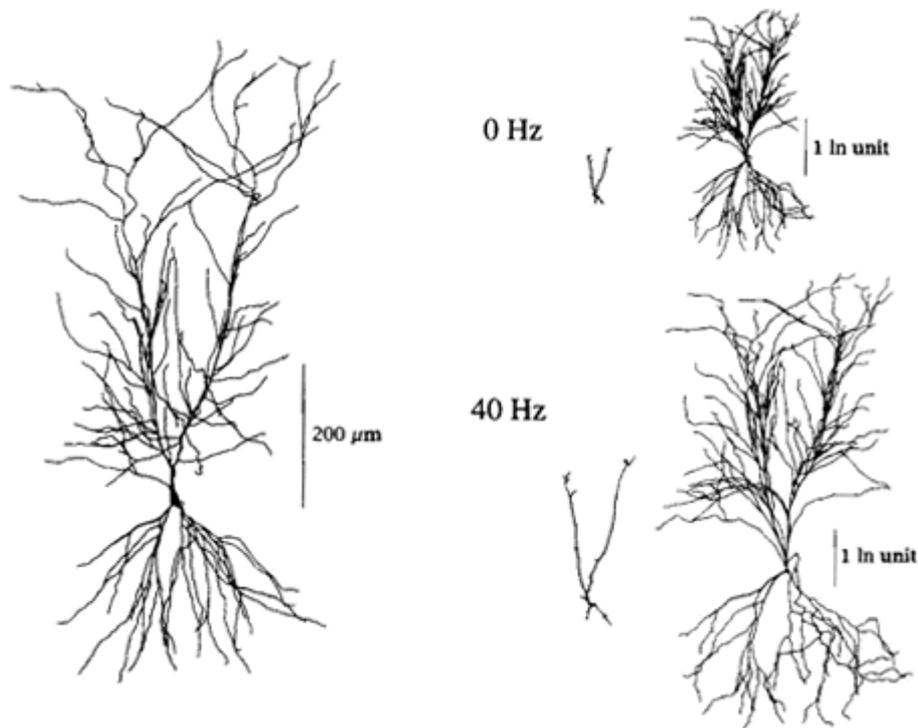


Figure 6: "Morfoelectronic transforms. A reconstructed CA1 neuron is shown on the left, and the morfoelectronic transforms for the decay of potential from the soma out the dendrites and from the dendrites toward the soma are illustrated on the right. The neuron diagrams on the right are drawn according to a scale that represents the \ln of the voltage attenuation ratios (1 \ln unit represents voltage attenuation of $1/e$ from the site of current injection). The top middle transform represents the $\ln(\text{attenuation})$ of a DC signal applied to the soma as it decays to different sites of dendrites. The top right transform represents the $\ln(\text{attenuation})$ of a DC signal applied to different sites in the dendrites as it decays toward the soma. The two transforms at the bottom right represent the same measurement as those above but resulting from the application of a 40 Hz sine wave signal instead of 0 Hz. (Kindly provided by Tsai, Caneval, Claiborne, and Brown.)" Reprinted and cited from Johnston/Wu (1995), Fig. 4.30, with kind permission from MIT Press.

Beyond the passive structure of a neuron (process diameters, lengths, passive input resistance, membrane capacitance, and axial resistance) neural dynamics are crucially dependent on the presence of voltage-gated ion channels. The modeling of individual ion channels would be too computationally costly, however, and the Hodgkin-Huxley formalism of macroscopic currents (one for each species of ion channel) is typically employed.

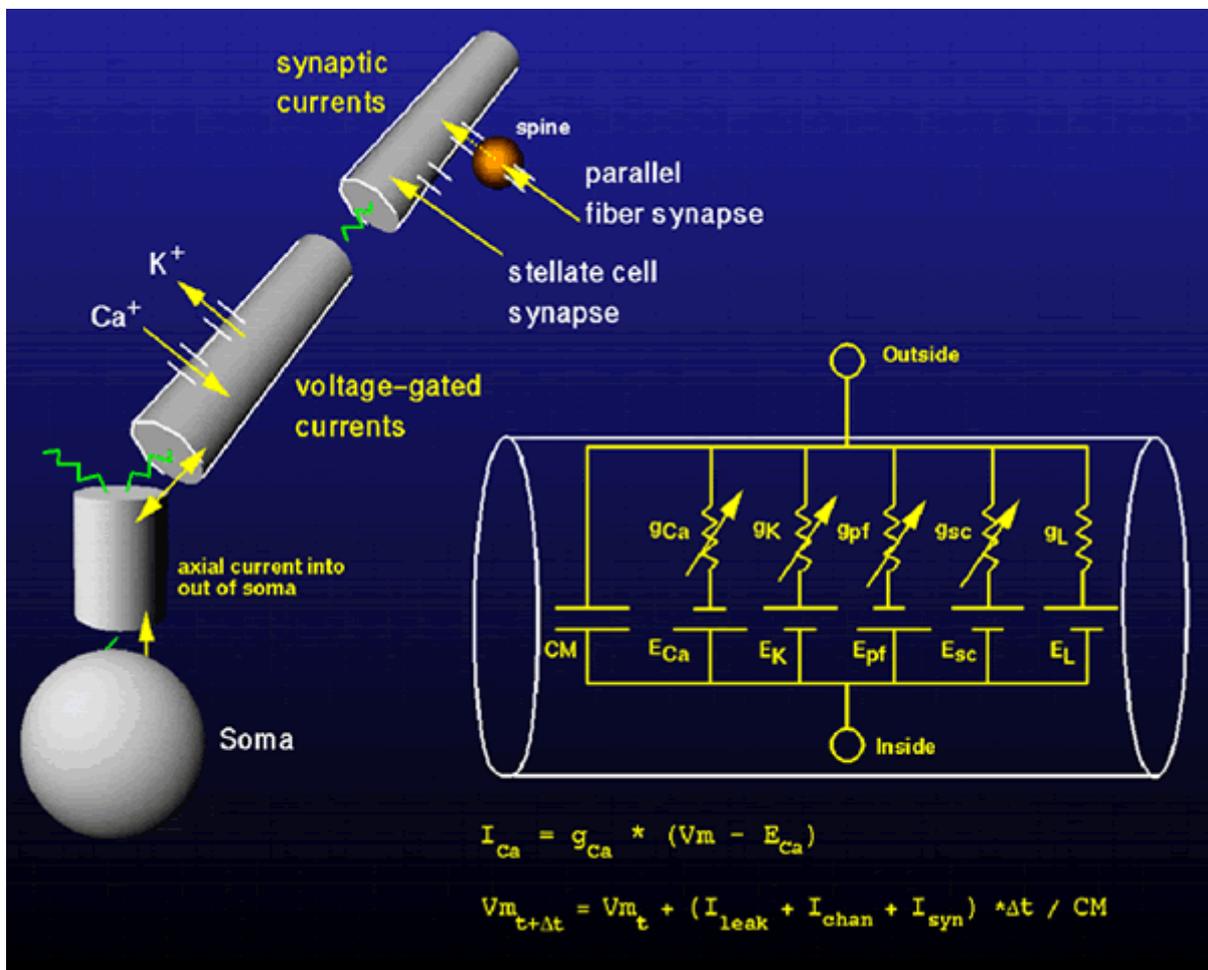


Figure 7: Structure of a compartmental model.

The overall structure of a compartmental model is depicted (Fig. 7). The neuron is divided into isopotential compartments (these need to be chosen small enough such that signal decay within the

length of one compartment is expected to be negligible under all conditions), and each compartment is given passive properties: membrane capacitance (C_M), membrane resistance (described by leak conductance and leak reversal potential). Lastly, active conductance properties are inserted, and are described by maximal conductance values (e.g. g_K), voltage-dependence (denoted by the arrow), and reversal potential.

These curves (Fig. 8) depict a set of Hodgkin-Huxley descriptions for voltage-gated conductance properties of a published Purkinje cell model. This model is available as tutorial on Dr. DeSchutter's website (<http://www.tnb.ua.ac.be>). The books by Hille (2001) and Johnston and Wu (1995) are excellent resources to learn more about the Hodgkin-Huxley formalism.

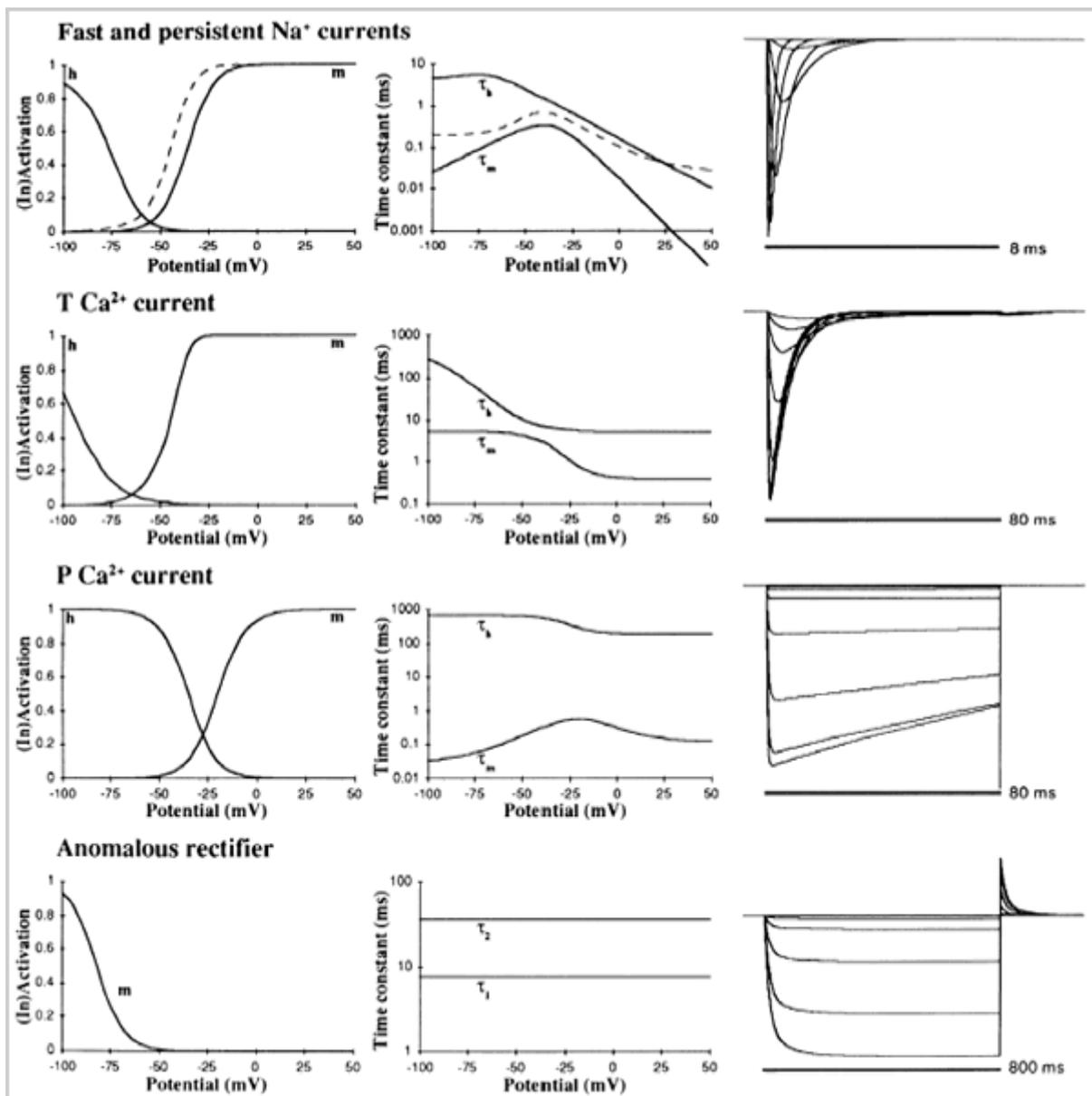


Figure 8: Set of Hodgkin-Huxley descriptions for voltage-gated conductance properties. DeSchutter and Bower (1994), Fig 2a-d. With kind permission from the American Physiological Society.

Potential Features of Single Cell Processing

The following highlights some important properties exhibited by biological neurons and realistic simulations, which are typically not present in simplified neural models, such as the integrate-and-fire formalism. These added properties are likely to be fundamental to signal processing in the brain.

Spatially inhomogenous processing

- signal decay
- signal delay
- shunting
- local amplification
- filtering

Memory

- Calcium concentration changes integrative properties
- Channel phosphorylation as well
- Time scales: multiple from ms to min to hours/years

Signal Generation

- plateau potentials
- oscillations
- bursting

Three steps to make a single neuron model

The bulk of this talk will now address how a realistic model is constructed, and what the common pitfalls are that prevent a model from truly matching the desired biological dynamics.

1. Create an Accurate Morphological Reconstruction
2. Create an Accurate Passive Model
3. Match active properties with physiological data

Create an Accurate Morphological Reconstruction

Creating a model starts with a morphological reconstruction of dye-filled neurons. Typically biocytin fills from slice recordings are used. The initial histological processing can already lead to distortions of the true neural morphology due to fixation shrinkage as well as mechanical stresses. Here, the same

Purkinje cell is shown before and after cover-slipping (Fig. 9). A little bit of squeeze on the cover slip significantly distorts the relative branch lengths. A certain amount of error in morphological reconstructions due to such distortions as well as incomplete fills of spines and distal processes should always be expected.

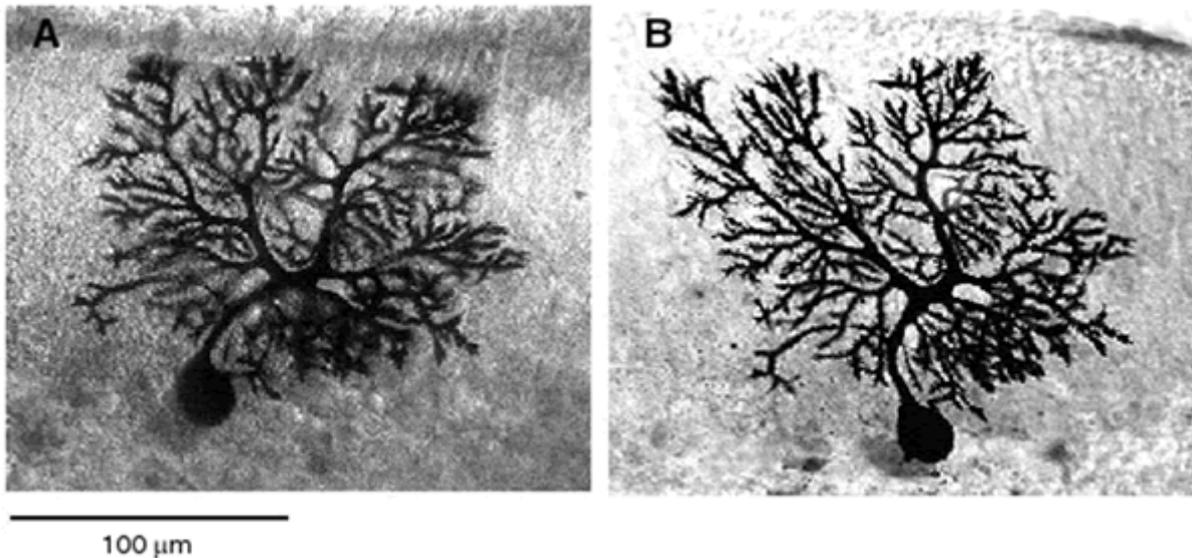


Figure 9: Biocytin fills from slice recordings.

Figure 10 shows another problem with morphological reconstructions. They are typically performed using the light -microscope, which has a resolution limit around 1 micron. Thus, the reconstruction of sub-micron processes is partly in the eye of the beholder, and different users of the reconstruction software can come up with quite divergent reconstructions of the same cell. Further details on cell reconstruction methods are described in [Jaeger \(2001\)](#).

A morphological reconstruction from Neurolucida can be imported into GENESIS or NEURON using Robert Cannon's CVAPP Java program (see <http://www.compneuro.org/>).

In GENESIS, a morphological description is given in .p files with an ASCII format highlighted below. Each compartment is listed as a line with its coordinates, parent compartment, and diameter. Prototype compartments are defined to tell the simulator which set of active properties to include in each compartment.

```
// Cell morphology file for GENESIS.
// Written by cvapp
// http://www.neuro.soton.ac.uk/cells/#software).

*absolute
*asymmetric
*cartesian

// End of cvapp-generated header file.

*origin 0.706 0.756 0
```

```

*compt /library/soma
soma none      0.706  0.756  0  6.067

*compt /library/dendrite
p0[1] soma  1.28  9.45  0  1.4
p0[2] p0[1]  2.09  12.56  0  1.4
p0[3] p0[2]  2.67  15.9  0  1.4
p0b1[0] p0[3]  10  21.77  0  1.16
p0b1b1[0] p0b1[0]  20.35  22.12  0  0.93
p0b1b2[0] p0b1[0]  11.39  29.37  0  0.7
p0b2[0] p0[3]  -5.93  27.42  0  1.05

p1[1] soma  22.09  -0.35  0  1.63
p1b1[0] p1[1]  27.91  9.68  0  1.16
p1b2[0] p1[1]  35.35  -7.83  0  0.93
    
```

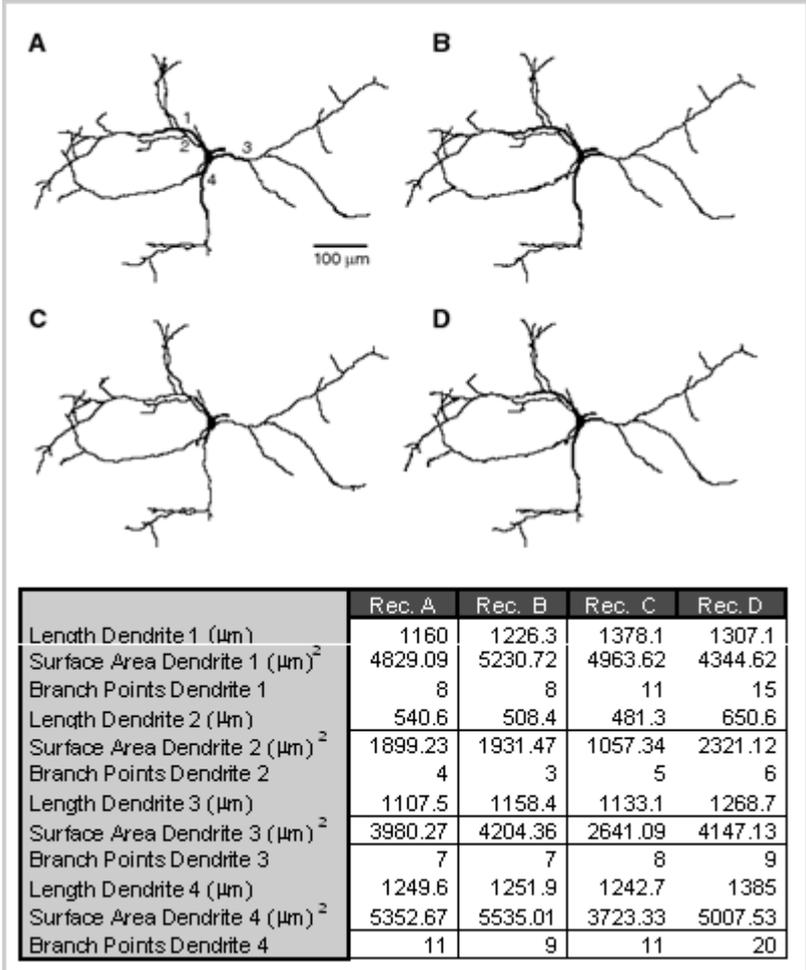


Figure 10: NeuroLucida (MicroBrightField, Inc.), Reconstructions of GP neuron

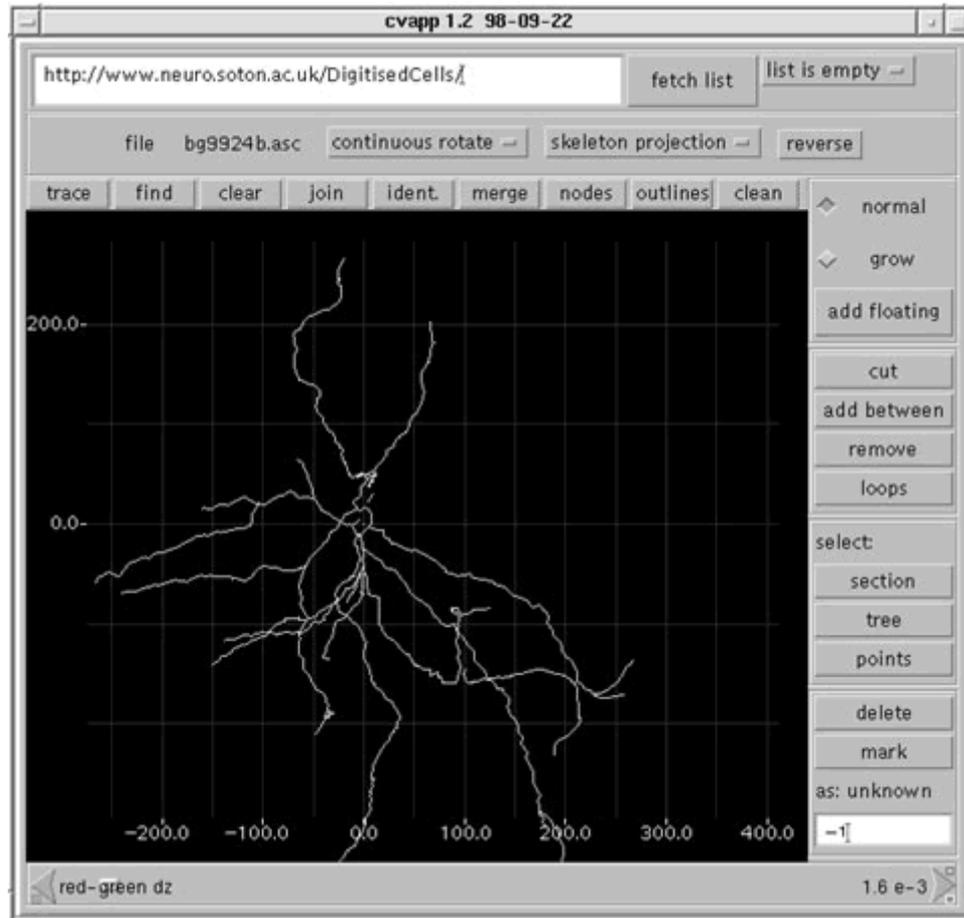


Figure 11: Screenshot CVAPP by Robert Cannon, available at <http://www.compneuro.org/>.

Create an Accurate Passive Model

After the morphology description has been imported into the simulation software, the passive parameters of the cell need to be fit to data. If the morphological reconstruction is less than ideal, the fit parameters may show a shift with respect to the true values to offset erroneous measurements on cell surface area and process diameters. For example, if surface area is missing, the specific membrane capacitance may come out higher than the standard value of 1 microfarad per square cm.

- Task: Set R_m , C_m , and R_i correctly for each compartment
- If assumption that parameters are uniform holds, need to fit 3 values.
- Strategy: Obtain data from recordings, then optimize R_m , C_m , R_i to fit data.

To fit passive parameters, recorded neural responses of the same cell type (preferably the same cell!!) two current injection pulses need to be available. These recordings need to be undertaken after voltage-gated conductances are blocked. A complete block is often difficult, and the recorded passive responses may deviate from the true passive response due to several other potential problems as well. **Message: optimize recordings to get linear (passive) response!**

Potential Problems

- Incomplete Block of Active Properties
- Partial Block of Leak Channels
- Electrode Serial Resistance
- Electrode Shunt

A somewhat improved method to fit passive responses has been described by [Guy Major \(1994\)](#) and should be used when possible. Here, the offset voltage decay after short current injection pulses is used (Fig. 12). See the source paper for details.

Message: Short current injection pulses have several advantages, obtain such measurements!

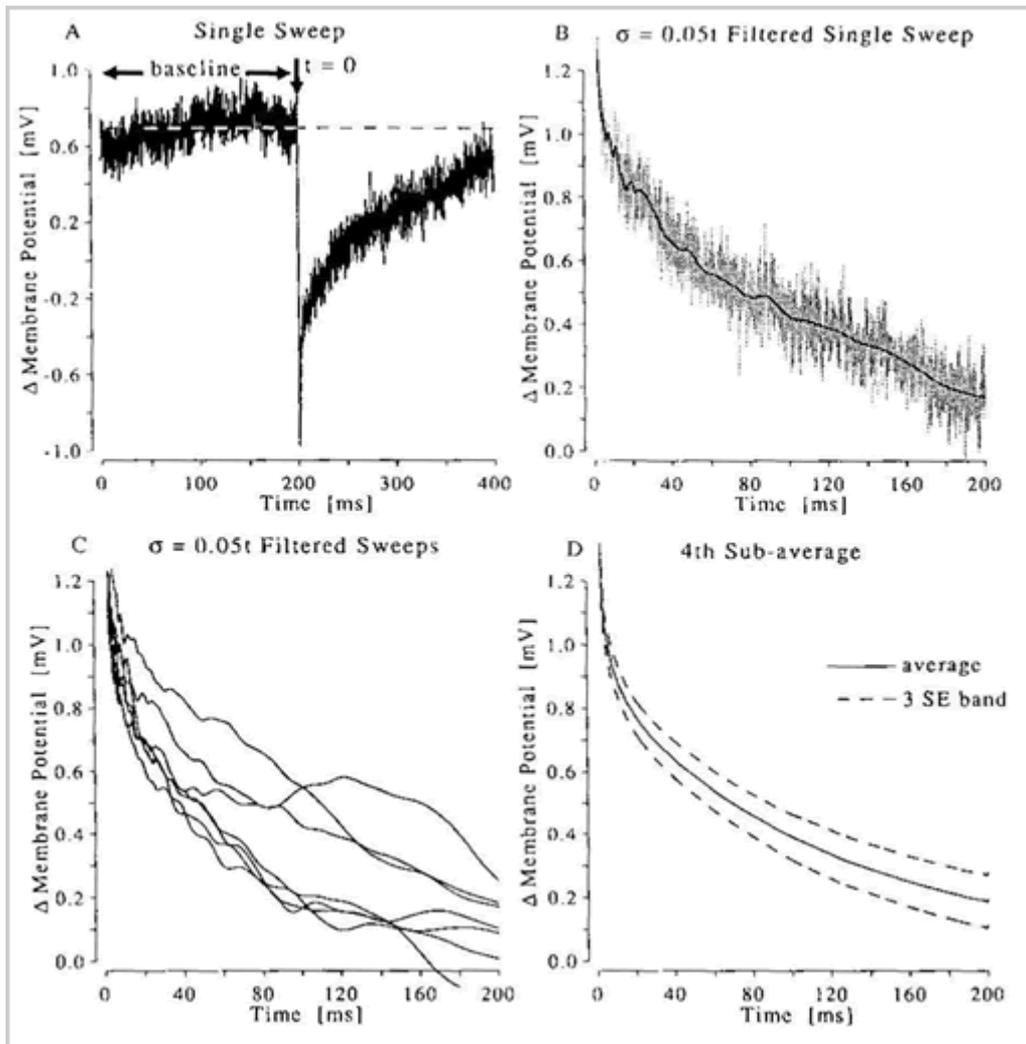


Figure 12: Reprint from [Major et al. \(1994\)](#). Kindly permitted by Guy Major and the Journal of Neuroscience.

Fit Data

Start with reasonable default parameters, and provide upper and lower bounds. Use automated fitting routine, e.g. with Matlab

R_m : 1 ohm * m² (limits: 0.5 10)

R_i : 1 ohm * m (limits: 0.5 .. 3)

C_m : 0.01 F / m² (limits: 0.005 ... 0.02)

If fitting of uniform parameters fails, try non-uniform ones. (1) Special case for soma (2) Special case for thin dendrites (3) Possibly try corrections to morphology

Fitting passive responses is done by running current injections into the soma of the model cell and optimizing the match of the voltage response with that of recordings. The bad news is that a perfect fit is often not possible due to the accumulation of errors of the sort already mentioned. The good news is that the fit is usually quite good after some fiddling, and that in the end the active properties will dominate over the influence of the passive ones in any case. Thus, in the final analysis, obtaining a decent passive model is not usually the bottleneck in creating a good active model.

Match active properties with physiological data

Step 3 is one of a kind. It takes about 85% of the total time of making a good model. The construction of a biologically realistic model is a more hopeful prospect if the availability of voltage-clamp data for the various voltage-gated conductances of the cell type under study is good. If such data are not available, one can proceed with default channel kinetics determined in other cell types and fit current-clamp data as well as possible. However, this latter process is unlikely to generate an accurate model in the end.

Tasks:

1. Construct set voltage-gated conductances with correct kinetics.
2. Incorporate intracellular calcium handling
3. Find needed densities of channels in each compartment
4. Add realistic synaptic input conductances

One of the most careful constructions of an active model I am aware of was performed by [Alain Destexhe et al \(1998\)](#) for the thalamocortical projection neuron. The paper should be read. The match of channel kinetics between data and model is exemplary.

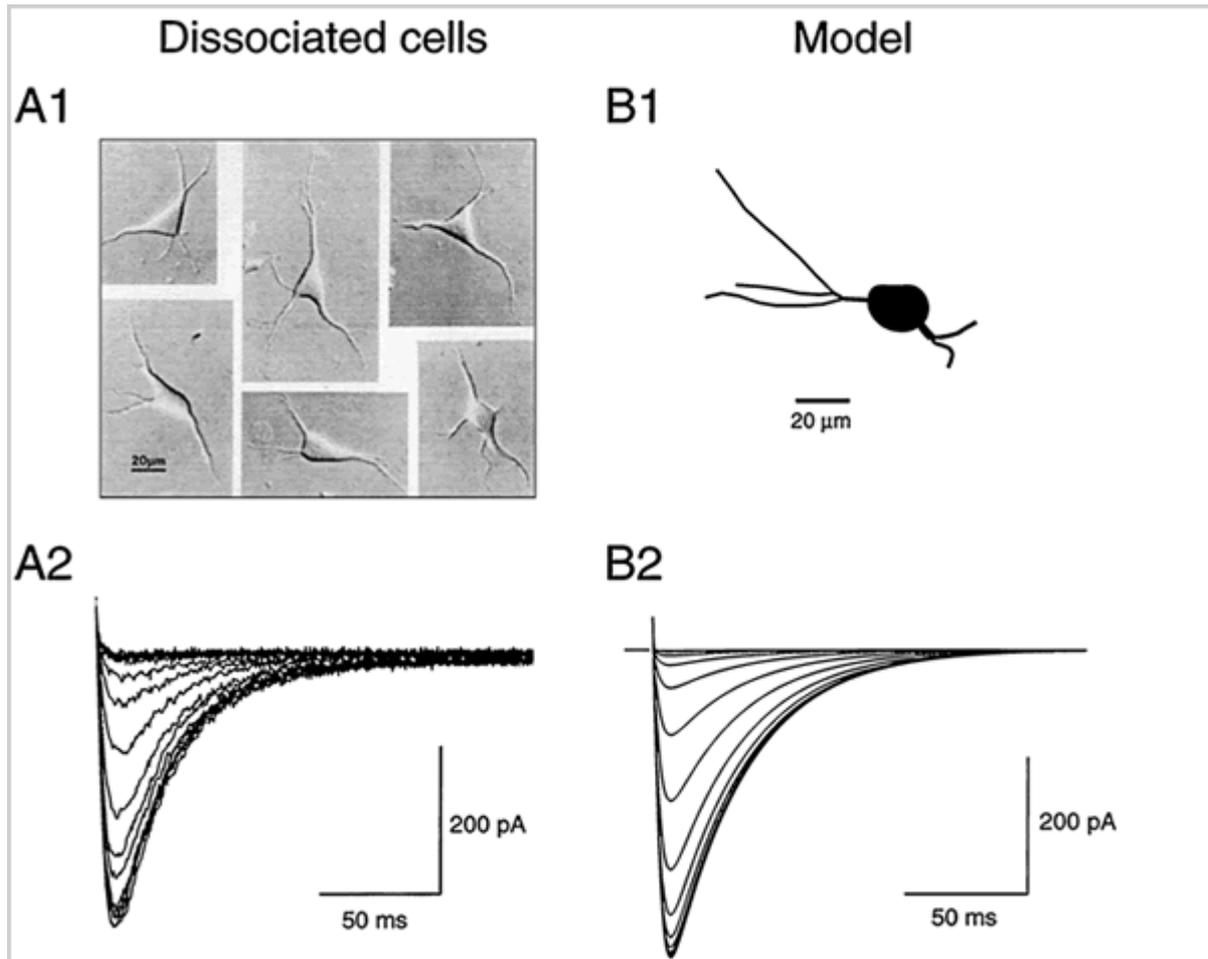


Figure 13: Redrawn from [Destexhe et al \(1998\)](#). With kind permission from Alain Destexhe and the Journal of Neuroscience.

The Hodgkin-Huxley formalism for the active conductances is spelled out. The method by which voltage-dependence of steady state activation and inactivation as well as time constants is described, can differ between simulators and investigators. Usually a mathematically closed form is used to match the observed voltage dependence, but a lookup table can also be used to describe the voltage-dependence (*Eq. 2 and 3 as well as the explanatory text are cited from [Destexhe et al \(1998\)](#). With kind permission from Alain Destexhe and the Journal of Neuroscience*):

$$\begin{aligned}
 I_{\tau} &= \bar{P}_{Ca} m^2 h G(V, Ca_o, Ca_i) \\
 \dot{m} &= -\frac{1}{\tau_m(V)} (m - m_{\infty}(V)) \\
 \dot{h} &= -\frac{1}{\tau_h(V)} (h - h_{\infty}(V))
 \end{aligned} \tag{2}$$

where P_{Ca} (in centimeter per second) is the maximum permeability of the membrane to Ca^{2+} ions and m and h are, respectively, the activation and inactivation variables. $G(V, Ca_o, Ca_i)$ is a nonlinear function of voltage and ionic concentrations:

$$G(V, Ca_o, Ca_i) = Z^2 F^2 V / RT \frac{Ca_i - Ca_o \exp(-ZFV/RT)}{1 - \exp(-ZFV/RT)} \quad (3)$$

where $Z = 2$ is the valence of the calcium ions, F is the Faraday constant, R is the gas constant, and T is the temperature in Kelvins. Ca_i and Ca_o are the intracellular and extracellular Ca^{2+} molar concentrations, respectively.

After some curve-fitting, the functions for voltage-dependence are fully parameterized. The conditions under which these conductances were recorded need to be carefully taken into account. In particular, the temperature dependence and calcium dependence of conductances can have a dramatic impact on the measured voltage-dependence (*Eq. 2 and 3 as well as the explanatory text are cited from Destexhe et al (1998). With kind permission from Alain Destexhe and the Journal of Neuroscience*):

The optimal functions that accounted for both voltage-clamp and current-clamp data on TC cells were:

$$m_{\infty}(V) = 1 / (1 + \exp[-(V + 56)/6.2]) \quad (4a)$$

$$h_{\infty}(V) = 1 / (1 + \exp[(V + 80)/4]) \quad (4b)$$

The voltage dependent time-constant for activation was:

$$\tau_m(V) = 0.204 + 0.333 / (\exp[-(V + 131)/16.7] + \exp[(V + 15.8)/18.2]) \quad (4c)$$

and for inactivation:

$$\tau_h(V) = \begin{cases} 0.333 \exp[(V + 466)/66.6] & \text{for } V < -81 \text{ mV} \\ 9.32 + 0.333 \exp[-(V + 21)/10.5] & \text{for } V > -81 \text{ mV} \end{cases} \quad (4d)$$

These functions correspond to an external Ca^{2+} concentration of 2 nM and a temperature of 36°C. All voltage-clamp simulations were done at 24°C assuming Q_{10} values of 2.5 for both m and h , whereas current-clamp behavior was simulated at 34°C.

It should be recognized that voltage-clamp recordings from large neurons are fraught with space-clamp errors. Thus when the type of neuron under study is large (such as the thalamocortical projection cell) good voltage-clamp records are best obtained from dissociated cells. A good model can actually match the shift in measured voltage-dependence of a conductance between a dissociated soma and a full dendritic morphology.

The conductance density of a given ion channel species across different areas of the neuron has dramatic influences on observed neural dynamics and needs to be carefully optimized to match recorded current clamp data. This match of conductance densities should be performed after the individual channel kinetics are well constrained to avoid the accumulation of errors.

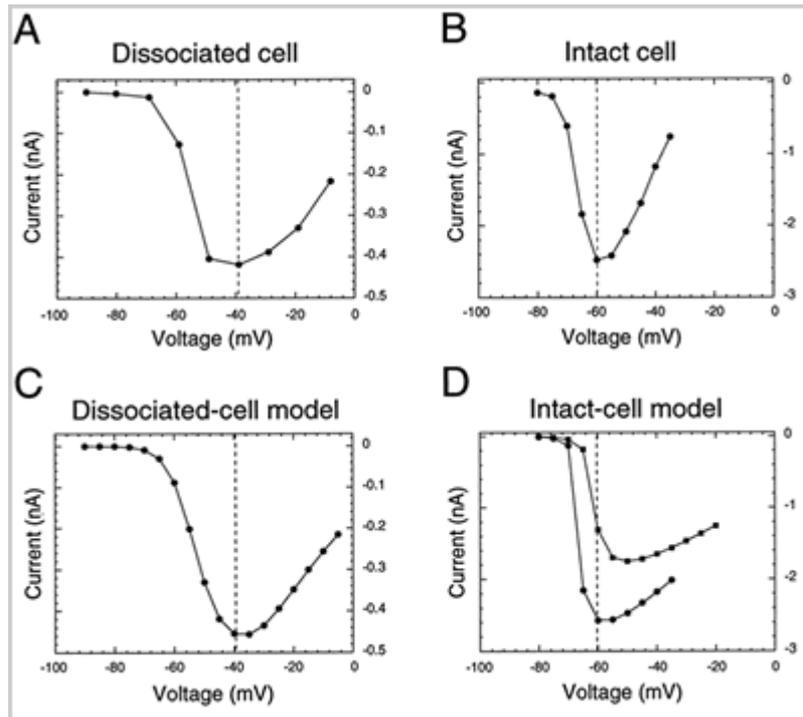


Figure 14a-d: Redrawn from [Destexhe et al \(1998\)](#). With kind permission from Alain Destexhe and the Journal of Neuroscience.

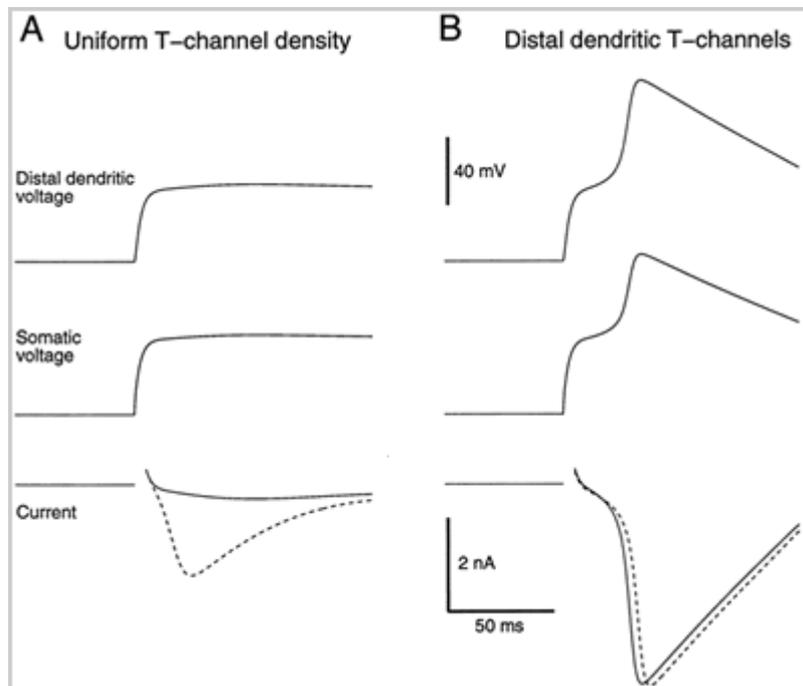


Figure 15: Redrawn from [Destexhe et al \(1998\)](#). With kind permission from Alain Destexhe and the Journal of Neuroscience.

Nevertheless, solutions for current densities that match physiological recordings may not be unique and need to be carefully examined. The determination of such densities can be done either with a process known as ‘manual tuning’ (trial and error augmented by investigator’s understanding of what each ion channel accomplishes) or with an automated parameter search. The latter is usually blind as to the properties of individual channel species, and will take a considerable amount of computer resources and simulation runs. Genetic algorithms have been used in simple cell models (less than 20 compartments) to find optimal channel density combinations, but this method is largely unexplored for full morphological models due to the high computational load.

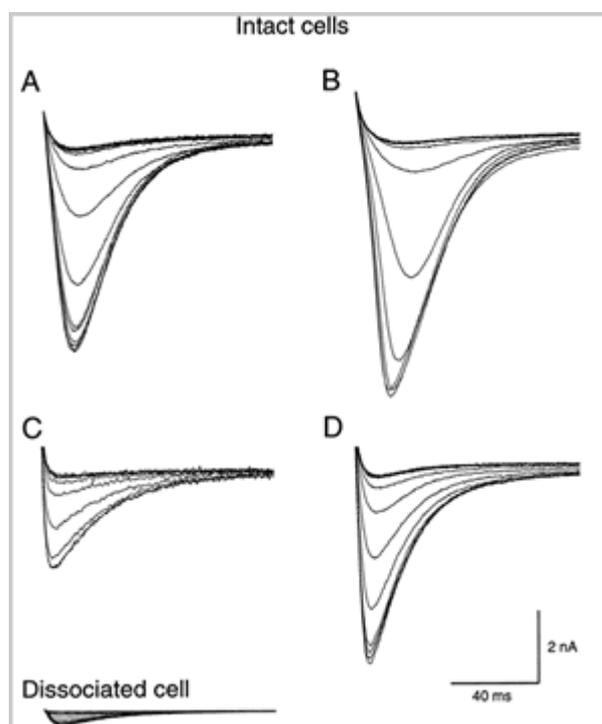


Figure 16: Redrawn from [Destexhe et al \(1998\)](#). With kind permission from Alain Destexhe and the Journal of Neuroscience.

After a lot of work, the model finally behaves like a real neuron (Fig. 17). This can be quantified by matches of individual spike waveforms, as well as spike patterns with various regimes of current injection. However, one still needs to be concerned about the state-dependence of this match. It can easily break down under new recording conditions, such as massive synaptic input or partial channel block. Thus the realistic model can be used for ‘in computo’ experiments that are not achievable in the biological preparation, but model predictions need to be always checked with further experiments. The model is thus a work in progress that functions as a ‘dynamical working hypothesis’ and also points out deficiencies in our understanding of neural properties under conditions when the model fails.

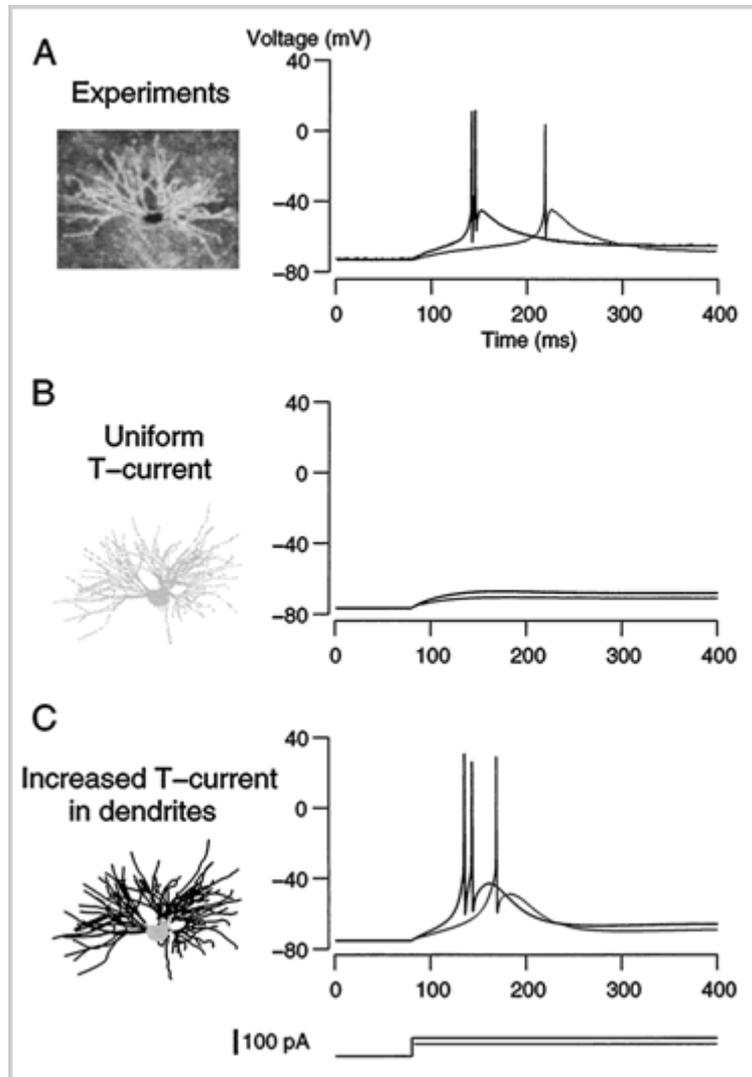


Figure 17: Redrawn from [Destexhe et al \(1998\)](#). With kind permission from Alain Destexhe and the Journal of Neuroscience.

These plots (Fig. 18) show in a compact format how the steady state activation and time constants of active conductances in a Purkinje cell model are a function of voltage and intracellular calcium. Note that this model is of a generation where some conductances are derived from Purkinje cell data, while for others more generic forms are used. Because of the strong calcium dependence of some potassium currents, a realistic model needs to include the simulation of intracellular calcium dynamics. This can be done in various ways (see below).

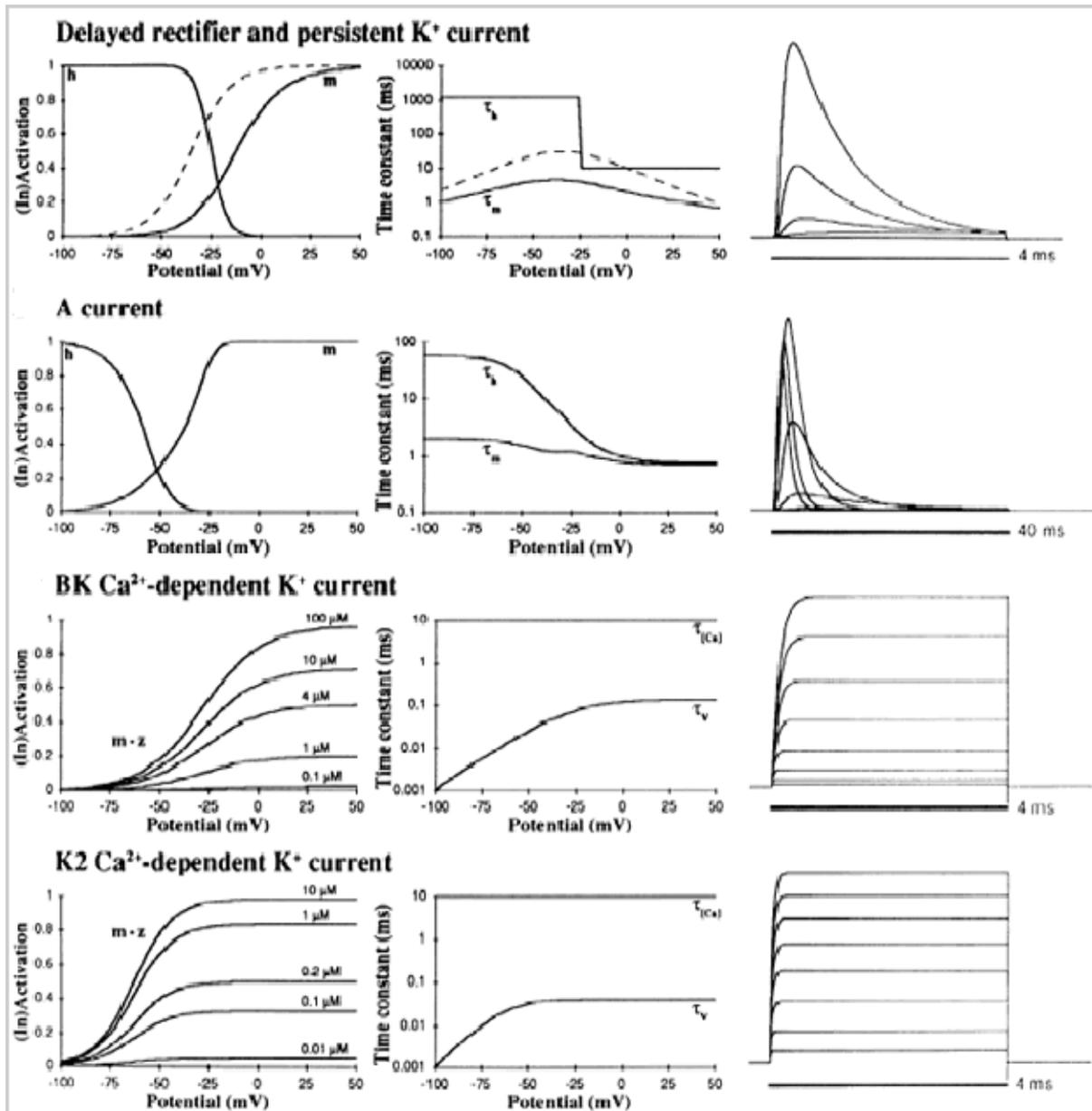


Figure 18: Reprint from DeSchutter and Bower (1994), Fig.2e-h. With kind permission from the American Physiological Society.

Calcium Handling

A *single diffusion shell* is a simple solution that works reasonably well. In this method, a thin (e.g. 0.2 micron) shell of volume below the membrane is taken as a reservoir for calcium with an influx proportional to membrane calcium currents, and an exponential decay to baseline. The baseline concentration is taken from physiological data (e.g. 40 nM). More complex solutions include binding to buffers, and intracellular stores. However, it is nearly impossible to get correct kinetic data for these processes.

Synapses

Finally, we can address synaptic integration after a model is tuned to include a set of voltage-gated conductances such that it reproduces voltage traces with current injections. Somewhat analogous to simulating macroscopic voltage-gated conductances with the Hodgkin-Huxley formalism, synapses are also abstracted in our simulations. Although synaptic physiologists can choose to model detailed processes such as calcium diffusion and calcium inflow in the presynaptic terminal, this level of detail is not required for more macroscopic questions about the integration of postsynaptic currents. The typical synapse in our neural models is simulated as a biexponential postsynaptic conductance change at some delay of the spiking from a presynaptic neuron. The presynaptic neuron can either be modeled explicitly, or the times of synaptic activation can be driven by other processes, such as random number generators, or saved time series from actual neural recordings.

To fit simulated biexponential synaptic conductance waveforms to realistic synaptic EPSCs and IPSCs, voltage clamp recordings from the cell type under study should again be available. Both the amplitude and time course of data can usually be fit quite well with the formalism shown. (This particular formula is taken from the `synchan` element in GENESIS).

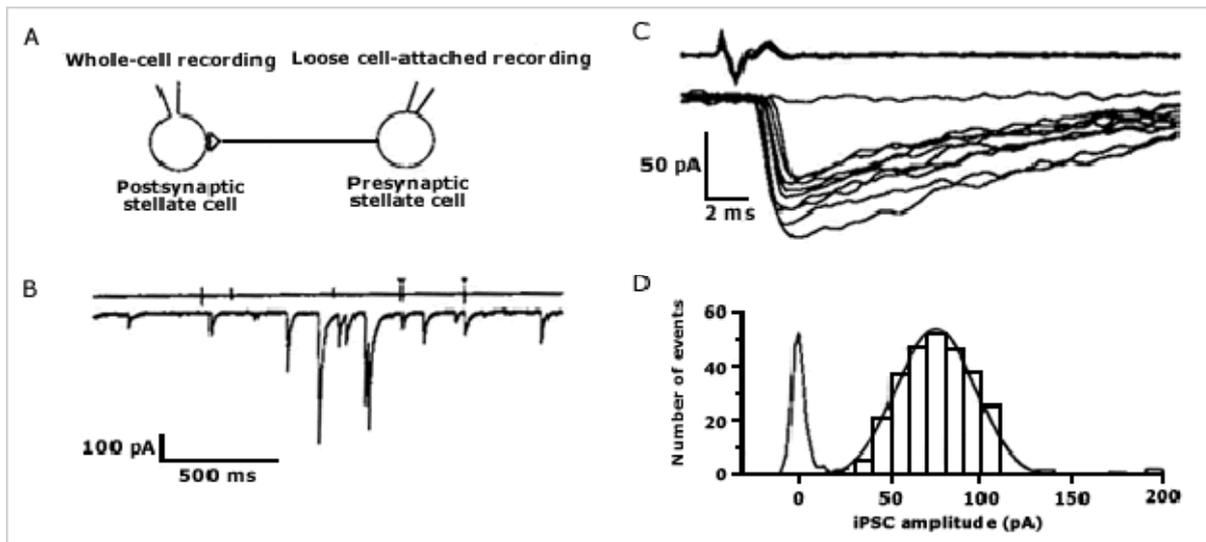


Figure 19a-d: adapted from [Kondo and Marty \(1998\)](#). Reprinted with kind permission from Alain Marty and the Journal of Physiology (London), Blackwell Publisher.

$$g_{syn}(t) = \frac{Ag_{max}}{\tau_1 - \tau_2} (e^{-t/\tau_1} - e^{-t/\tau_2}), \text{ for } \tau_1 > \tau_2 \quad (6)$$

Equation 6: adapted from [Kondo and Marty \(1998\)](#).

Even though highly simplified, the modeling of synapses as postsynaptic conductance waveforms retains highly important biological properties of synaptic input, such as voltage dependence and shunt currents. This is quite different from the typical synapse in simplified neural network models, which do not have a reversal potential.

Potential Shortcomings of simplified synaptic modeling:

- Missing all or none release probability
- Missing minis
- Missing short-term plasticity
 - change in release probability, e.g. due to residual Ca
 - extrasynaptic receptor activation
 - presynaptic receptor activation

Once a model is complete, it can help the scientist to appreciate the 3-dimensional distribution of voltage and currents throughout different activity regimes (Fig. 20). Movie files are one helpful tool to obtain this information. Here, the membrane potential distribution of the Purkinje cell model is depicted shortly after the onset of current injection into the soma.

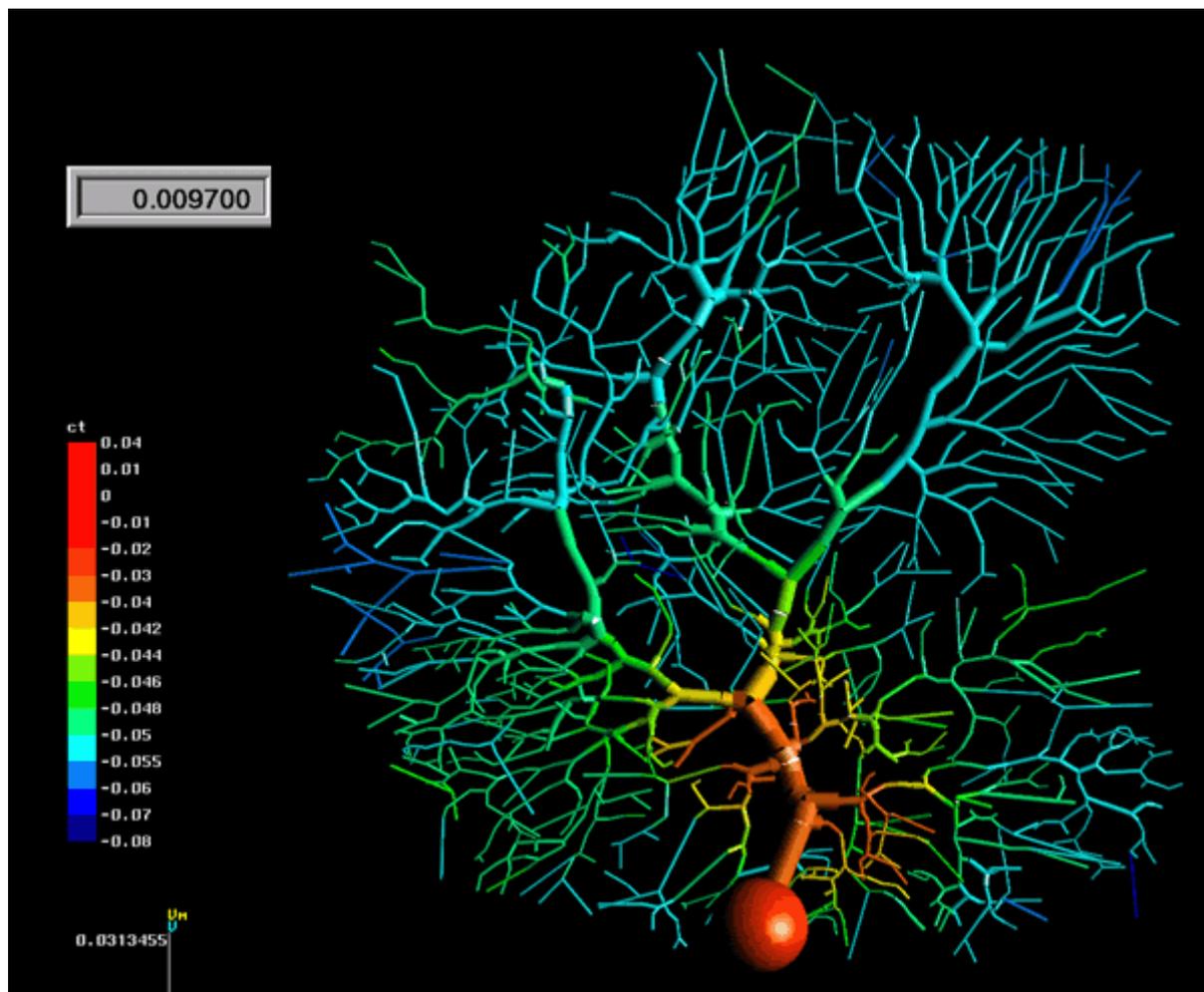


Figure 20: The membrane potential distribution of the Purkinje cell model shortly after the onset of current injection into the soma.

Analyzing and understanding the behavior of a complex realistic model is very much an additional task beyond the construction of the model. Because of many interacting non-linear dynamics, a complex model becomes subject to experimentation much like real recordings. A great advantage of the model vs. experimental data, however, is given by the complete availability of all parameters and variables for manipulation and recording. Figure 21 is taken from a study I performed in examining the dendritic and somatic balances of inward and outward currents during spontaneous and synaptically driven spiking. Interestingly, the model showed a near perfect balance of inward and outward currents, such that very small synaptic input fluctuations could modulate spiking. In fact, due to the intrinsic pacemaking properties of the model, a small net outward synaptic current was the result of a barrage of excitatory and inhibitory synaptic inputs at a mix that resulted in biologically realistic spiking frequencies.

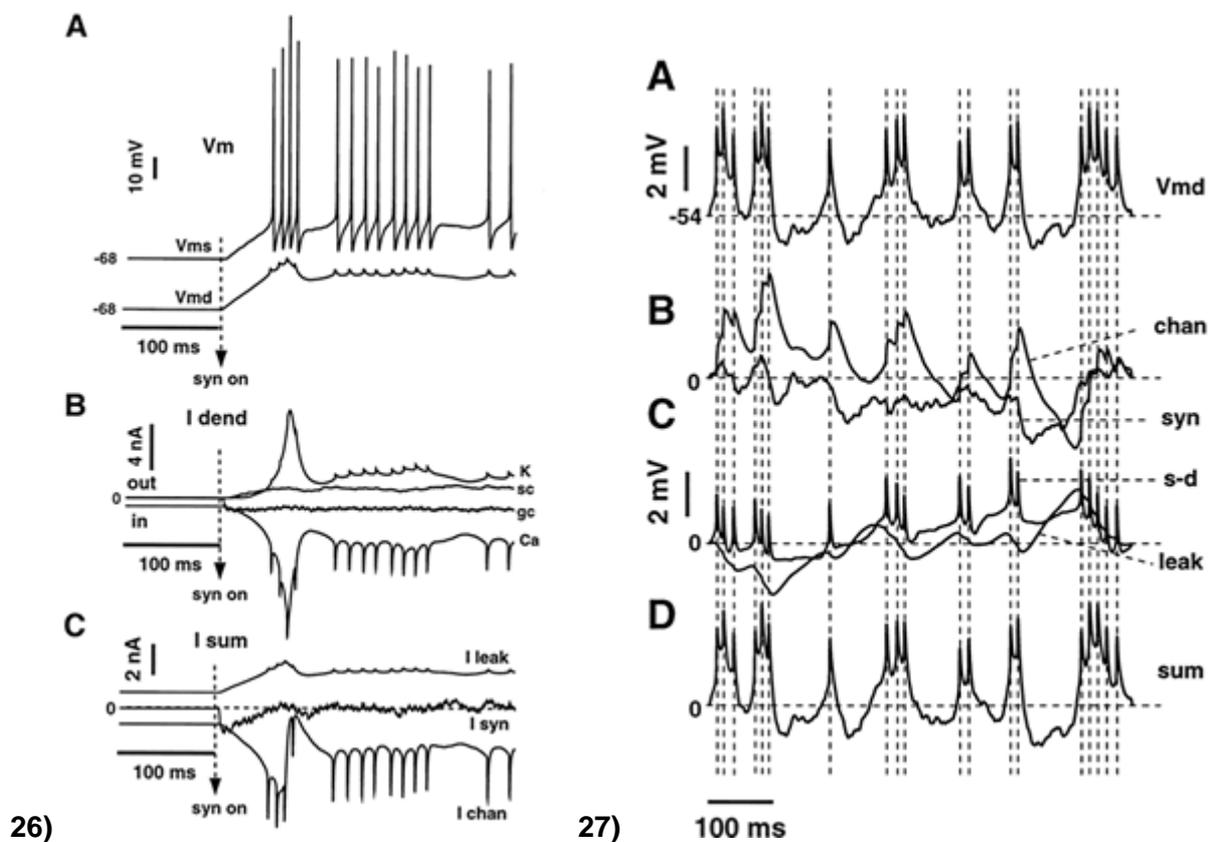


Figure 21 a-c and Figure 22 a-d. Details see text.

By plotting the contribution of each conductance type to membrane voltage fluctuations (Fig. 22; here the average of the dendrite of the Purkinje cell model is shown), one can delineate the contribution of synaptic and intrinsic conductances to the observed firing behavior. The observed spike pattern is the result of both synaptic input fluctuations and the intrinsic dynamics of voltage-gated conductances.

The modeling results can be tested for example by using the technique of dynamic clamping, in which arbitrary synaptic conductance patterns can be applied to the soma of recorded neurons in brain slices. In the case of the Purkinje cell, this test verified that realistic spike frequencies were obtained, when the synaptic current had a net outward direction (Fig. 23). Individual spikes were primarily caused by fluctuations in inhibitory inputs.

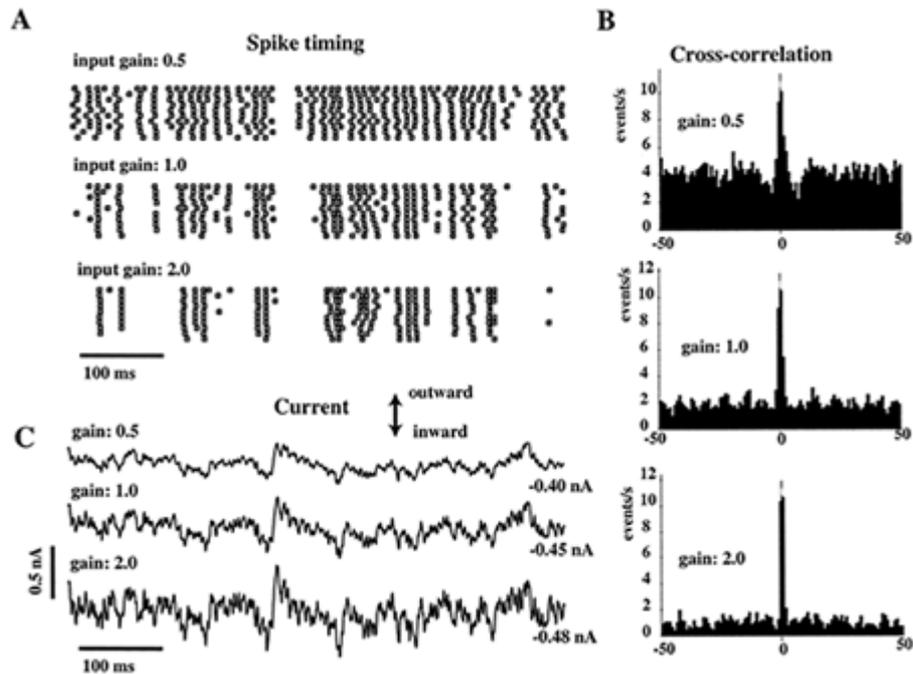


Figure 23 a-c: Details see text.

In vivo intracellular recordings (Fig. 24) generally confirm a constant barrage of synaptic inputs, which lead to rich subthreshold dynamics. Here the up- and down states of a striatal medium spiny neuron under ketamine-anesthesia are shown in relation to the slow-wave activity of the cortical EEG.

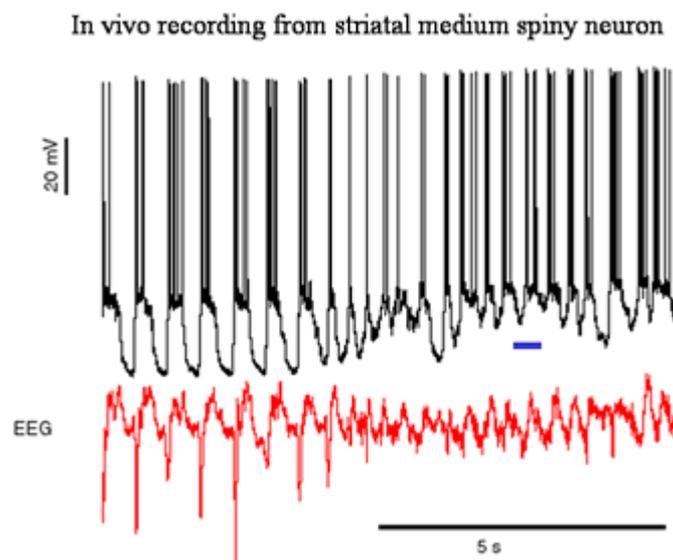


Figure 24: Details see text.

An expanded segment of figure 24 highlights the constant barrage with synaptic inputs in vivo (Fig. 25). Asterisks denote action potentials, which are cut off in this figure.

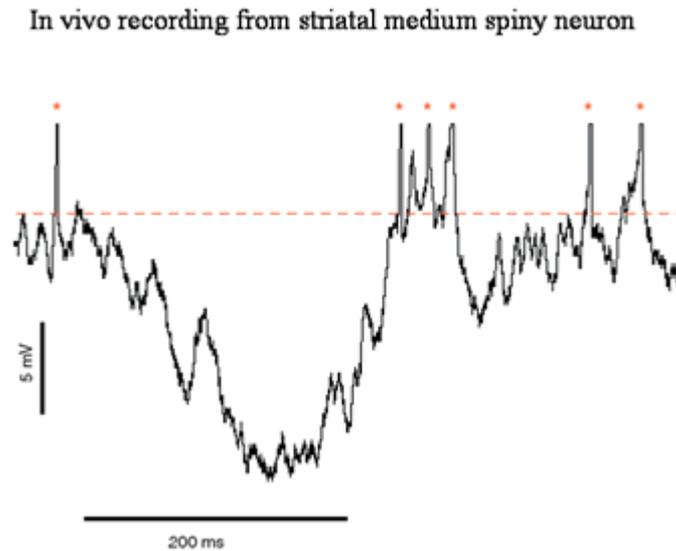


Figure 25: Details see text.

The big picture

- Realistic single cell models can be constructed, but it is a lot of work.
- Even the best model is still not going to completely replicate biological single cell dynamics
- Each type of neuron is a separate modeling job
- Insight into the processing capabilities of neurons can be gained from realistic single cell modeling
- Experimental predictions can be made and tested
- Insights gained are invaluable for realistic network modeling
- Simplified single cell models are likely required for large network models

A brief primer on how to model complex single cells with GENESIS

1. Create morphology file
2. Create channel and compartment prototypes
3. Create main simulation script
4. Create parameter search scheme to tune model
5. Include synapses
6. Enjoy

Appendix: Purkinje cell model in GENESIS

The following 7 steps show some implementation details of the *Purkinje cell model in GENESIS*. Note that the complete set of simulation scripts and a tutorial interface can be downloaded from Dr. DeSchutter's web site in Antwerp.

Step 1

The script segment shown here depicts function declarations used to set up Hodgkin-Huxley voltage dependencies. In GENESIS the tabchannel object is usually taken to construct these conductances.

```
<genhome>/Scripts/Purkinje/Purk_chansave.g
/* Make tabchannels */
function make_chan(chan, Ec, Gc, Xp, XAA, XAB, XAC, XAD, XAF, \
    XBA, XBB, XBC, XBD, XBF, Yp, YAA, YAB, YAC, YAD, YAF, YBA, \
    YBB, YBC, YBD, YBF, Zp, ZAA, ZAB, ZAC, ZAD, ZAF, ZBA, ZBB, \
    ZBC, ZBD, ZBF)
    str chan
    int Xp, Yp, Zp
    float Ec, Gc
    float XAA, XAB, XAC, XAD, XAF, XBA, XBB, XBC, XBD, XBF
    float YAA, YAB, YAC, YAD, YAF, YBA, YBB, YBC, YBD, YBF
    float ZAA, ZAB, ZAC, ZAD, ZAF, ZBA, ZBB, ZBC, ZBD, ZBF
    if ({exists {chan}})
        return
    end
    create tabchannel {chan}
    setfield {chan} Ek {Ec} Gbar {Gc} Ik 0 Gk 0 Xpower {Xp} \
        Ypower {Yp} Zpower {Zp}
    if (Xp > 0)
        setupalpha {chan} X {XAA} {XAB} {XAC} {XAD} {XAF} \
            {XBA} {XBB} {XBC} {XBD} {XBF} -size {tab_xfills+1} \
            -range {tab_xmin} {tab_xmax}
    end
    if (Yp > 0)
        setupalpha {chan} Y {YAA} {YAB} {YAC} {YAD} {YAF} {YBA} \
            {YBB} {YBC} {YBD} {YBF} -size {tab_xfills+1} -range \
            {tab_xmin} {tab_xmax}
    end
    if (Zp != 0)
        setupalpha {chan} Z {ZAA} {ZAB} {ZAC} {ZAD} {ZAF} {ZBA} \
            {ZBB} {ZBC} {ZBD} {ZBF} -size {tab_xfills+1} -range \
            {tab_xmin} {tab_xmax}
    end
end
end
```

Step 2

Here, a customized way of filling the voltage-tables for a tabchannel is shown. Note the comments on top of the script segment. The use of extensive comments is vital in making portable and understandable simulations.

```
<genhome>/Scripts/Purkinje/Purk_const.g

/* non-inactivating (muscarinic) type K current, eq#2
** eq#2: corrected typo in Yamada equation for tau: 20 instead
of 40
** Refs: Yamada (his equations are also for 22C) */
if (!{exists Purk_KM})
    create tabchannel Purk_KM
    setfield Purk_KM Ek {EK} Gbar {GK} Ik 0 Gk 0 \
        Xpower 1 Ypower 0 Zpower 0
    call Purk_KM TABCREATE X {tab_xdivs} {tab_xmin} \
        {tab_xmax}
    x = {tab_xmin}
    dx = ({tab_xmax} - {tab_xmin})/{tab_xdivs}

    for (i = 0; i <= ({tab_xdivs}); i = i + 1)
        y = 0.2/(3.3*({exp {(x + 0.035)/0.02}}) \
            + {exp {-(x + 0.035)/0.02}})
        setfield Purk_KM X_A->table[{i}] {y}

        y = 1.0/(1.0 + {exp {-(x + 0.035)/0.01}})
        setfield Purk_KM X_B->table[{i}] {y}
        x = x + dx
    end
    tau_tweak_tabchan Purk_KM X
    setfield Purk_KM X_A->calc_mode 0 X_B->calc_mode 0
    call Purk_KM TABFILL X {tab_xfills + 1} 0
end
call Purk_KM TABSAVE Purk_KM.tab
```

Step 3

The parameters of the simulation are conveniently declared in a central location.

```
<genhome>/Scripts/Purkinje/Purk_const.g
* cable parameters */
float CM = 0.0164 // *9 relative to passive
float RMs = 1.000 // /3.7 relative to passive comp
float RMd = 3.0
float RA = 2.50
/* preset constants */
float ELEAK = -0.0800 // Ek value used for the leak
conductance
```

```

float EREST_ACT = -0.0680      // Vm value used for the RESET
/* concentrations */
float CCaO = 2.4000           //external Ca as in normal slice
Ringer
float CCaI = 0.000040         //internal Ca in mM
//diameter of Ca_shells
float Shell_thick = 0.20e-6
float CaTau = 0.00010        // Ca_concen tau

/* Currents: Reversal potentials in V and max conductances S/m^2
*/
/* Codes: s=soma, m=main dendrite, t=thick dendrite, d=spiny
dendrite */
float ENa = 0.045
float GNaFs = 75000.0
float GNaPs = 10.0
float ECa = 0.0125*{log {CCaO/CCaI}} // 0.135 V
float GCaTs = 5.00
float GCaTm = 5.00
float GCaTt = 5.00
float GCaTd = 5.00
float GCaPm = 45.0
float GCaPt = 45.0
float GCaPd = 45.0
float EK = -0.085
float GKAs = 150.0
float GKAm = 20.0
float GKdrs = 6000.0
float GKdrm = 600.0
float GKMs = 0.400
float GKmm = 0.100

```

Step 4

Here the conductances are added to 'prototype' compartments, which are later used in reading the morphology file. In GENESIS, we often construct 'prototypes' in a 'library' hierarchy, which is later used for building blocks in making the actual simulated cell.

```

<genhome>/Scripts/Purkinje/Purk_cicomp.g
if (!(exists Purk_soma))
    create compartment Purk_soma
end
setfield Purk_soma Cm {{CM}*surf} Ra {8.0*{RA}/(dia*{PI})} \
    Em {ELEAK} initVm {EREST_ACT} Rm {{RMs}/surf} inject 0.0 \
    dia {dia} len {len}

copy Purk_NaF Purk_soma/NaF
addmsg Purk_soma Purk_soma/NaF VOLTAGE Vm
addmsg Purk_soma/NaF Purk_soma CHANNEL Gk Ek

```



```

setfield Purk_soma/NaF Gbar {{GNaFs}*surf}
copy Purk_NaP Purk_soma/NaP
addmsg Purk_soma Purk_soma/NaP VOLTAGE Vm
addmsg Purk_soma/NaP Purk_soma CHANNEL Gk Ek
setfield Purk_soma/NaP Gbar {{GNaPs}*surf}
copy Purk_CaT Purk_soma/CaT
addmsg Purk_soma Purk_soma/CaT VOLTAGE Vm
addmsg Purk_soma/CaT Purk_soma CHANNEL Gk Ek
setfield Purk_soma/CaT Gbar {{GCaTs}*surf}
copy Purk_KA Purk_soma/KA
addmsg Purk_soma Purk_soma/KA VOLTAGE Vm
addmsg Purk_soma/KA Purk_soma CHANNEL Gk Ek
setfield Purk_soma/KA Gbar {{GKAs}*surf}
copy Purk_Kdr Purk_soma/Kdr
addmsg Purk_soma Purk_soma/Kdr VOLTAGE Vm
addmsg Purk_soma/Kdr Purk_soma CHANNEL Gk Ek
setfield Purk_soma/Kdr Gbar {{GKdrs}*surf}
copy Purk_KM Purk_soma/KM
addmsg Purk_soma Purk_soma/KM VOLTAGE Vm
addmsg Purk_soma/KM Purk_soma CHANNEL Gk Ek
setfield Purk_soma/KM Gbar {{GKM}s}*surf}

copy Purk_h1 Purk_soma/h1
addmsg Purk_soma Purk_soma/h1 VOLTAGE Vm
addmsg Purk_soma/h1 Purk_soma CHANNEL Gk Ek
setfield Purk_soma/h1 Gbar {{Ghs}*surf}
copy Purk_h2 Purk_soma/h2
addmsg Purk_soma Purk_soma/h2 VOLTAGE Vm
addmsg Purk_soma/h2 Purk_soma CHANNEL Gk Ek
setfield Purk_soma/h2 Gbar {{Ghs}*surf}
create Ca_concen Purk_soma/Ca_pool
setfield Purk_soma/Ca_pool tau {CaTau} \
    B {1.0/(2.0*96494*shell_vol)} Ca_base {CCaI} \
    thick {Shell_thick}
addmsg Purk_soma/CaT Purk_soma/Ca_pool I_Ca Ik
copy GABA2 Purk_soma/basket
setfield Purk_soma/basket gmax {{GB_GABA}*surf}
addmsg Purk_soma/basket Purk_soma CHANNEL Gk Ek
addmsg Purk_soma Purk_soma/basket VOLTAGE Vm

```

Step 5

Here, the different pieces of the simulation are included, the functions building the simulation are run, the morphology file is read with the readcell command, and the Hines solve structure for the fastest simulation mode in GENESIS is set up.

```
<genhome>/Scripts/Purkinje/Purk_cicomp.g

//- include scripts to create the prototypes

include Purk_chanload
include Purk_cicomp
include Purk_syn

include info.g
include bounds.g
include config.g
include control.g
include xgraph.g
include xcell.g

//- ensure that all elements are made in the library

ce /library

//- make prototypes of channels and synapses

make_Purkinje_chans

readcell {cellfile} {cellpath}

echo preparing hines solver {getdate}
ce {cellpath}
create hsolve solve

//- We change to current element solve and then set the fields of
the parent
//- (solve) to get around a bug in the "." parsing of GENESIS

ce solve

setfield . \
    path "../##[ ][TYPE=compartment]" \
    comptmode 1 \
    chanmode {iChanMode} \
    calcmode 0

call /Purkinje/solve SETUP

//- Use method to Crank-Nicolson
setmethod 11
```

Step 6

Simulation results can be stored in ascii or binary data files. For automated processing of the data files (e.g. with Matlab) it is useful to concatenate parameter settings into the file name.

```

File Saving System:  from DCN simulation by vsteuber

include cn_fileout16

outfilev = "data/cn0106c_" @ {simnum} @ "_v_"
outfilei = "data/cn0106c_" @ {simnum} @ "_i_"
outfilevc = "data/cn0106c_" @ {simnum} @ "_vc_"
outfilechan = "data/cn0106c_" @ {simnum} @ "_chan_"

// set up messages to the output files
write_voltage soma 0
write_currents_c soma 0
write_chan_activations soma 0
//write_voltage p0b2 0
//write_voltage p0b2b1b2 0
//write_voltage p0b2b1b2 4
write_voltage p0 1
write_voltage p1 1

function write_voltage(compt, comptnum)
str compt, comptnum
str hinesloc, outfile, outelement, clampstr

outfile = {outfilev @ compt @ comptnum @ "_" @ clampstr @ ".dat"}
outelement = "/output/v_" @ {compt} @ {comptnum}

if (!{exists {outelement}})
    create asc_file {outelement}
    echo
    echo creating output element {outelement}
    useclock {outelement} 8
    hinesloc = {findsolvefield {cellpath} \
                {cellpath}/{compt} \ Vm}
    addmsg {cellpath} {outelement} SAVE {hinesloc}
end

setfield {outelement}      filename {outfile} \
                             initialize 1 \
                             leave_open 1 \
                             append 1

echo
echo sending {cellpath}/{compt} Vm to {outfile}.

end

```

Step 7

Start of Matlab function to read GENESIS output.

```
function p_cn6c_sc13(cip,n,fignum1,fignum2,eleak)

addpath('../data')

if (cip < 0.0)
    ifile = sprintf('cn0106c_%s_chan_soma0_m%spA.dat', \
        num2str(n),num2str(-cip));
else ifile = sprintf('cn0106c_%s_chan_soma0_%spA.dat', \
    num2str(n), num2str(cip));
end
temp = load(ifile);
...
```

Step 8

Matlab is a powerful off-line analysis tool for simulation results. Here, a set of plots is constructed from a simulation in the process of tuning conductance density parameters to obtain realistic voltage traces.

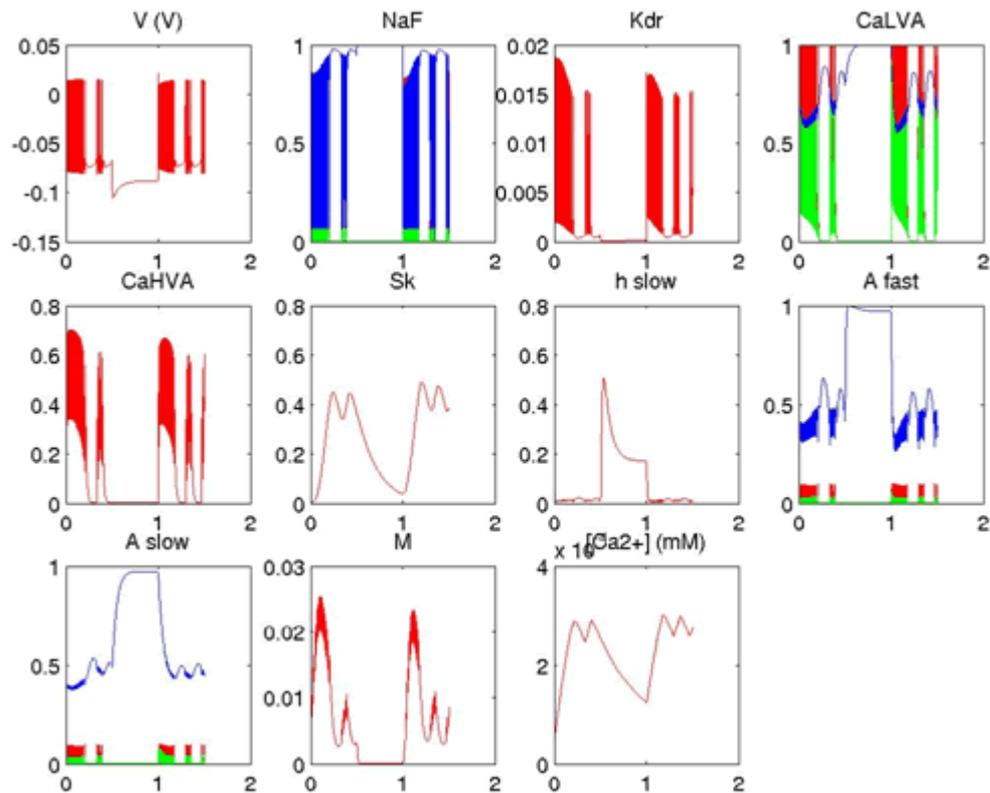


Figure A1: Matlab output to scan results of parameter combination.

References

DeSchutter, E, and Bower, JM (1994). An Active Membrane Model of the Cerebellar Purkinje Cell: I. Simulation of current clamps in slice J. Neurophysiol. 71:375-400.

Destexhe, A, Neubig, M, Ulrich, D, Huguenard, J. (1998). Dendritic Low-Threshold Calcium Currents in Thalamic Relay Cells, The Journal of Neuroscience, May 15, 18(10):3574-3588.

Hille, B (2001). Ion Channels of Excitable Membranes. Third Edition. Sinauer.

Jaeger, D. (2001). Accurate Reconstruction of Neuronal Morphology. In: De Schutter, E (ed.) (2001). Computational Neuroscience: Realistic Modeling for Experimentalists. CRC Press.

Johnston, D and Wu, SM (1995). Foundations of Cellular Neurophysiology, MIT Press.

Kondo, S and Marty, A (1998) Synaptic currents at individual connections among stellate cells in rat cerebellar slices. J.Physiol.(Lond.). 1998 May 15; 509(1)221-232.

Major, G., Larkman, AU, Jonas, P, Sakmann, B. Jack, JJB. (1994). Detailed Passive Cable Models of Whole-Cell Recorded CA3 Pyramidal Neurons in Rat Hippocampal Slices, J. Neurosci., August 1994, 14(8): 4613-4638.

Segev, I. (1998) Cable and Compartmental Models of Dendritic Trees in "The Book of GENESIS Internet Edition, Chapter 5" (Bower, J. and Beeman, D., eds.).

For general introductions to Neurosciences and neuronal modeling see:

Kandel, ER, Schwarz, JH and Jessell, TM (2000). Principles of Neural Science, Fourth Edition, McGraw Hill.

Koch, C and Segev, I (1998). Methods in Neuronal Modeling. From Ions to Networks. Second Edition. MIT Press.