Using Complicated, Wide Dynamic Range Driving to Develop Models of Single Neurons in Single Recording Sessions

Kevin H. Hobbs and Scott L. Hooper

Neuroscience Program, Department of Biological Sciences, Ohio University, Athens, Ohio

Submitted 10 January 2008; accepted in final form 31 January 2008

Hobbs KH, Hooper SL. Using complicated, wide dynamic range driving to develop models of single neurons in single recording sessions. J Neurophysiol 99: 1871–1883, 2008. First published February 6, 2008; doi:10.1152/jn.00032.2008. Neuron models are typically built by measuring individually, for each membrane conductance, its parameters (e.g., half-maximal voltages) and maximal conductance value (gmax). However, neurons have extended morphologies with nonuniform conductance distributions, whereas models generally contain error due to the electrical filtering of neurons and the differential placement of conductances on them. Model parameters (typically gmax values) are therefore generally altered by hand or brute force to match model and neuron activity. We propose an alternative method in which complicated, rapidly changing driving input is used to optimize model parameters. This method also ensures that neuron and model dynamics match across a wide dynamic range, a test not performed in most modeling. We tested this concept using leech heartbeat and generic tonically firing models and lobster stomatogastric and generic bursting models as targets and gmax values as optimized parameters. In all four cases optimization solutions exactly matched target activity. Complicated, wide dynamic range driving thus appears to be an excellent method to characterize neuron properties in detail and to build highly accurate models. In these completely defined targets, the method found each target’s 8–13 gmax values with high accuracy, and may therefore also provide an alternative, functionally based method of defining neuron gmax values. The method uses only standard experimental and computational techniques, could be easily extended to optimize conductance parameters other than gmax, and should be readily applicable to real neurons.

INTRODUCTION

Conductance-based neuron modeling plays an increasingly important role in neuroscience. The central goal of such work is to build models that accurately describe neuron activity. However, present model-building techniques do not use neuron activity as their fundamental selection criterion. Models are instead typically constructed by first isolating individual conductances by a combination of pharmacological blockade of other conductances and the use of voltage command protocols that activate only one or a few conductances and various subtractive techniques. The conductances are then characterized on the basis of ion selectivity and the dependence of their opening and closing rates and steady-state values on voltage and, for some conductances, calcium. In these experiments maximum conductance (gmax) values are also typically measured.

The resulting equations are then placed in models typically containing one or a few compartments. In this first step, model and neuron activity often do not match well. One reason for this mismatch is the extended morphology of most neurons and the nonuniform distribution of membrane conductances on this morphology. No matter where a recording electrode is placed, some parts of the neuron are therefore electrically distant. Activity arising at these distant sites thus suffers from slow temporal filtering and amplitude attenuation. Conductance time constant measurements of electrically distant conductances will consequently be slower, and gmax measurements smaller, than they actually are.

These difficulties can theoretically be overcome. With present computational resources models with thousands of compartments are possible. Patch-clamp recordings of individual channels, or sharp electrode recordings from regions where a conductance is known to be present, can accurately measure conductance parameters. Advances in immunology and molecular biology hold the promise of accurately identifying conductance placement across neuron morphology. However, it is presently experimentally impossible to perform all these measurements (full description of a neuron’s morphology, accurate description of the parameters defining all its conductances, accurate placement of the conductances across the neuron’s processes) on an individual neuron.

Neuron descriptions are therefore typically data averaged from multiple neurons of a given type. For instance, at present neuron conductance properties and amounts are typically measured by pharmacological isolation. These treatments are time consuming and, for some conductances, irreversible. Characterizing all of a neuron’s conductances during a single recording session is consequently generally impossible. Membrane conductance equations of a “neuron” are typically instead averages obtained from recordings of multiple neurons of the “same” neuron type, with one set of conductances being obtained from recordings from one group of neurons, another set of conductances from recordings from a second group of neurons, and so forth, until all the conductances of the “neuron” have been included.

The difficulty with this approach, however, is that recent work suggests that neurons of a given type can have very different maximum conductance value sets because maximum conductance values can change in a coordinated fashion that maintains neuron activity (Desai et al. 1999; Franklin et al. 1992; Goldman et al. 2001; Golowasch et al. 1999a,b, 2002; Haedo and Golowasch 2006; Hong and Lnenicka 1995; Klein...
improved model performance when $I$) noise was added; 2) when noise was added and the equations describing target and model conductances substantially differed; and 3) when noise was added, the equations describing target and model conductances substantially differed, and a conductance was omitted during the optimization. These data suggest that activity-driven optimization algorithms are a promising alternative to classic model-building methods.

The experimental procedures are rapid (minutes) and use only standard techniques. Single runs of the evolutionary algorithm required about 4 days using twenty 2-GHz processors. Several weeks were required for complete optimization, but this time would scale inversely with processor number and speed. Extending the method to optimize conductance properties as well as amounts, and to build multicompartment models, would be straightforward.

**METHODS**

**Neuron simulation**

Neuron simulation was conductance based and standard [see *Numeric values and equations* for generic and lobster burster (Prinz et al. 2003) models and Nadim et al. (1995) for the leech tonic equations]. All code was written in C using the GNU Scientific library version 1.7. Variable integration steps with Runge–Kutta stepping functions were used. Step size was continuously adjusted until all variable and derivative errors were just $<10^{-3}$ (millisecond-based models, leech and lobster) or $10^{-6}$ (second-based models, both generics). To avoid very small sizes at current injection or command voltage changes, if the change would have occurred within an otherwise acceptable step, step size was set so that step end occurred at the time of the change.

**Neuron perturbation**

The models were stimulated with a driving current in current clamp and a command voltage in voltage clamp. The stimulus sequence was generated once, randomly (from a uniform distribution), and was the same for all simulations, except for the data in Fig. 6. Current amplitude (all NA) distributions were $-400$ to $200$ for the leech tonic, $-0.4$ to $0.2$ for the lobster burster, and $-20$ to $10$ for both generic models. Command voltage distributions were $-100$ to $-30$ mV for all models. Application of tetrodotoxin, tetraethylammonium, and 4-aminopyridine was simulated by setting $g_{Na}, g_{K2}$, and $g_A$ (leech tonic), $g_{Na}, g_{K4}$, and $g_A$ (lobster burster), and $g_{Na}, g_{K3}$, $g_{Afast}$ (both generic) to zero.

**Evolutionary algorithm**

The evolutionary algorithm was largely standard (Fogel 1994; Yao et al. 1999). The algorithm began on 1,000 model neurons (“individuals”) whose maximum conductance values ($x_i$, where $i$ refers to the individual and $j$ to the conductance type) were drawn randomly (within the conductance’s range; see *Numeric values and equations*) from uniform distributions, one for each conductance type. Each individual’s maximum conductances were associated with mutation size parameters ($\eta_{ij}$) that were set in the initial population to one half of the relevant maximum conductance’s range. The error scores (see RESULTS) of each individual were then calculated.

Each individual created a child as follows. The children first inherited mutated size parameters according to

$$\eta'_{ij} = \eta_{ij} \cdot \exp [\tau \cdot N(0, 1) + \tau' \cdot N_j(0, 1)]$$

where $\eta_{ij}$ is the child’s new size parameter; $\eta_{ij}$ is the parent’s size parameter; $N(0, 1)$ is a number drawn from a standard normal
distribution once per generation and used for every maximum conductance and individual in that generation; \(N_j/(0, 1)\) is a number drawn anew from a standard normal distribution for each individual \(j\) and maximum conductance \(i\); and \(\tau' = \sqrt{2n} \) and \(\tau = (\sqrt{2}/n)^{-1}\), where \(n\) is the number of maximum conductances. The \(\eta_j\) then helped determine the child’s mutated maximum conductances according to

\[
x'_j = x_j + \eta_j \cdot \delta_j
\]

where \(\delta_j\) is a number randomly drawn for each individual \(i\) and conductance \(j\) from a Cauchy distribution with a scale parameter of 1. Cauchy distributions resemble Gaussian distributions but have longer, flatter tails and are thus more likely to generate numbers far from 0. If a mutation would cause a maximum conductance value to become negative the parameter was set to zero. Each child also inherited its parent’s error.

Recombination among the children occurred as follows. A potential mate was chosen randomly from the entire population of children (i.e., a child could mate with itself) for one child. The errors the child and its mate inherited from their parents were then compared. If the child’s error was greater than that of its mate, the child’s maximum conductance values and mutation size parameters were replaced by a random assortment of both children’s values (sexual reproduction). All values were randomly assorted; that is, a maximum conductance value could come from one child and the conductance’s mutation size parameter from the other. If the child’s error was less than or equal to that of its mate, no recombination occurred. A mate was then selected for the next child and the process repeated for all 1,000 children. Since mates were always chosen from the entire population of children, children that had undergone recombination in prior matings could serve as mates in subsequent ones.

At the end of this process the population consisted of 1,000 parents and 1,000 children. The errors of the mutated and recombined children were then calculated. Each individual was then compared with ten individuals chosen randomly from the entire (2,000) population and assigned a score equal to how many of the ten its error was less than or equal to. The 1,000 individuals with the highest scores (ties being resolved based on error) survived and became parents in the next generation. This procedure allows some individuals with relatively high errors to survive. Evolution continued until the population mean stopped moving, and the population distribution remained confined to a very small range, for about 50 generations.

The mutation size parameter and population error distribution interact as follows. If a parent’s error is large relative to the population mean error, then this parent having a large mutation size parameter is generally advantageous because large mutations increase the child’s likelihood of successfully competing against the rest of the population. If the parent’s error is small relative to the population mean error, then its having a small mutation size parameter is advantageous because it produces children similar to the parent. This interaction results in mutation size parameters being large in early generations and then becoming very small as evolution proceeds and population mean error decreases.

**Hardware**

Computations were performed on local Beowulf Linux clusters and Beowulf Linux and OSX clusters at the Ohio Supercomputer Center (OSC). Message-passing interface was used to communicate between nodes; one node served as a master node. The master node ran the evolutionary algorithm and assigned individuals of the population one at a time to the other nodes, which performed the simulations and returned errors to the master. The clusters used were the Ohio University Physics (20 Intel Pentium 3 processors) and Biology (48 AMD Athlon MP processors) clusters and the OSC Pentium 4 (112 Intel Pentium 4 processors), Itanium 2 (476 Itanium 2 64-bit processors), and Apple G5 (64 PowerPC G5 processors) clusters. We never used more than 20 processors at a time on any cluster. We used 170,994 CPU hours on the OSC clusters and a similar but unrecorded number on the local clusters, although much of this time was spent developing the method.

**Numeric values and equations**

Target conductance units were microsiemens (\(\mu\)S) for the two generic models, millisiemens (mS)/cm\(^2\) for the lobster burster model, and nanosiemens (nS) for the leech heartbeat model. Target conductance values were \(g_{\text{Na}} = 16\), \(g_{\text{Ca}} = 5\), \(g_{\text{K}} = 3\), \(g_{\text{Na}} = 350\), \(g_{\text{K}} = 100\), \(g_{\text{Ca}2} = 50\), \(g_{\text{Na}2} = 80\), \(g_h = 7\), \(g_{\text{leak}} = 10\) (leech tonic); \(g_{\text{Na}2} = 0.05\), \(g_{\text{Ca}2} = 100\), \(g_{\text{Ca}3} = 4\), \(g_{\text{K}} = 5\), \(g_{\text{Ca}4} = 15\), \(g_{\text{Kd}} = 50\), \(g_h = 0.02\) (lobster burster); \(g_{\text{Na}2} = 4.078\), \(g_{\text{Ca}2} = 2.06\), \(g_{\text{Aslow}} = 6.029\), \(g_h = 0.4539\), \(g_{k} = 0.2485\), \(g_{\text{KCa}} = 27.41\), \(g_{\text{AHP}} = 0.0303\), \(g_{\text{Caf}1} = 1.083\), \(g_{\text{Caf}2} = 0.3573\), \(g_{\text{Caf}3} = 0.0469\) (generic tonic); \(g_{leak} = 0.1\), \(g_{\text{Na}2} = 1,500\), \(g_h = 0.5\), \(g_{\text{Afast}} = 0.07\), \(g_{\text{Aslow}} = 0.6\), \(g_h = 4\), \(g_{\text{KCa}3} = 0.06\), \(g_k = 0.2\), \(g_{\text{Caf}1} = 1\), \(g_{\text{AHP}} = 0.3\), \(g_{\text{Caf}2} = 0.7\), \(g_{\text{Caf}3} = 0.002\) (generic burster).

Initial maximum conductance ranges for the evolutionary algorithm were 0–100 for all leech tonic conductances; 0–0.05 (\(g_{\text{leak}}\); \(g_h\)), 0–500 (\(g_{\text{Na}2}\), 0–12.5 (\(g_{\text{Ca}2}\), 0–10 (\(g_{\text{Ca}4}\), 0–25 (\(g_{\text{KCa}4}\), 0–125 (\(g_{\text{Kd}}\) for the lobster burster (as in Prinz et al. 2003); and 500–1,000 (\(g_{\text{Na}2}\) and 0–10 (all others) for both generic models.

The generic models used the following equations

\[
dV/dt = \frac{-\left(i_{\text{leak}} + i_{\text{Na}} + i_{\text{K}} + i_d + i_k\right)}{1.7 \text{ nF}}
\]

where \(i_{\text{leak}} = \tilde{g}_{\text{leak}}(V - E_{\text{leak}})\),

\[
i_{\text{Na}} = \tilde{g}_{\text{Na}} m^3 h (V - E_{\text{Na}})
\]

\[
m = m_0 = \frac{\alpha}{\alpha + \beta}
\]

\[
\alpha = 2.2, \text{ if } V = -13
\]

\[
\alpha = 0.11 \cdot (V + 13), \text{ otherwise}
\]

\[
\beta = 15 \cdot e^{-0.0769 \cdot (V + 34)}
\]

\[
\frac{dh}{dt} = (h_n - h) \cdot 500
\]

\[
h_n = \frac{\alpha}{\alpha + \beta}
\]

\[
\alpha = 0.08 \cdot e^{-0.125 \cdot (V + 39)}
\]

\[
\beta = \frac{1}{1 + e^{-0.2 \cdot (V + 40)}}
\]

\[
i_{\text{K}} = \tilde{g}_{\text{K}} n^4 (V - E_{\text{K}})
\]

\[
\frac{dn}{dt} = (n_n - n) k_n
\]

\[
n_n = \frac{1}{1 + e^{-0.0588 \cdot (V + 25)}}
\]

\[
k_n = \frac{180}{1 + e^{-0.0455 \cdot (V - 10)}}
\]

\[
\tilde{g}_{ij} = \tilde{g}_{ij} H_{ij}^0 (V - E_{ij})
\]

\[
k_n = \frac{1}{1 + e^{-0.0588 \cdot (V + 25)}}
\]

\[
K_n = \frac{180}{1 + e^{-0.0455 \cdot (V - 10)}}
\]
The lobster burster model equations are identical to those in Prinz et al. (2003). They are included here because this model was the one in which
equation parameters (e.g., half-maximal voltages) were altered to test whether the algorithm would continue to find improved solutions even when the equations describing the conductances being used by the algorithm and those describing the target conductances substantially differed. In these cases (Fig. 6, B and C) equation parameters were modified by sequentially going through the equations parameter by parameter and alternately multiplying the parameter by 1.1 or 0.9, except in all conductances but \( I_h \), the first term in the \( \tau_m \) equations was always multiplied by 1.1 to avoid \( \tau_m \) becoming negative at certain membrane voltages. In the following equations a parameter was multiplied by 1.1 when followed by a “\( u \)” and by 0.9 when followed by a “\( d \).”

\[
i_{Na} \]
\[
m_a = \frac{1}{1 + \exp\left(\frac{V + 25.5u}{-5.29d}\right)}
\]
\[
h_a = \frac{1}{1 + \exp\left(\frac{V + 48.9u}{5.18d}\right)}
\]
\[
\tau_m = 2.64u - \frac{2.52d}{1 + \exp\left(\frac{V + 120u}{-25d}\right)}
\]
\[
\tau_h = \frac{1.34u}{1 + \exp\left(\frac{V + 62.9d}{-10u}\right)} \left[1.5d + \frac{1}{1 + \exp\left(\frac{34.9u}{3.6d}\right)}\right]
\]

\[
i_{CaT} \]
\[
m_x = \frac{1}{1 + \exp\left(\frac{V + 27.1u}{-7.2d}\right)}
\]
\[
h_x = \frac{1}{1 + \exp\left(\frac{V + 32.1u}{5.5d}\right)}
\]
\[
\tau_m = 43.4u - \frac{42.6d}{1 + \exp\left(\frac{V + 68.1u}{-20.5d}\right)}
\]
\[
\tau_h = 210u - \frac{179.6d}{1 + \exp\left(\frac{V + 55u}{-16.9d}\right)}
\]

\[
i_{CaS} \]
\[
m_x = \frac{1}{1 + \exp\left(\frac{V + 33u}{-8.1d}\right)}
\]
\[
h_x = \frac{1}{1 + \exp\left(\frac{V + 60u}{6.2d}\right)}
\]
\[
\tau_m = 2.8u + \frac{14d}{\exp\left(\frac{V + 27u}{10d}\right) + \exp\left(\frac{V + 70u}{-13d}\right)}
\]
\[
\tau_h = 120u + \frac{300d}{\exp\left(\frac{V + 55u}{9d}\right) + \exp\left(\frac{V + 65u}{-16d}\right)}
\]

\[
i_{KCa} \]
\[
m_a = \frac{[Ca]}{[Ca] + 3} \frac{1}{1 + \exp\left(\frac{V + 28.3u}{-12.6d}\right)}
\]
\[
\tau_m = 180.6u - \frac{150.2d}{1 + \exp\left(\frac{V + 46u}{-22.7d}\right)}
\]

\[
i_{Kd} \]
\[
m_a = \frac{1}{1 + \exp\left(\frac{V + 12.3d}{-11.8u}\right)}
\]
\[
\tau_m = 14.4u - \frac{12.8d}{1 + \exp\left(\frac{V + 28.3d}{-19.2u}\right)}
\]

\[
i_h \]
\[
m_a = \frac{1}{1 + \exp\left(\frac{V + 75d}{5.5u}\right)}
\]
\[
\tau_m = \frac{2}{\exp\left(\frac{V + 169.7d}{-11.6u}\right) + \exp\left(\frac{V - 26.7d}{14.3u}\right)}
\]

[Ca\(^{2+}\)] dynamics
\[
\tau_{Ca u} \frac{d[Ca^{2+}]}{dt} = -f(I_{CaT} + I_{CaS}) - [Ca^{2+}] + [Ca^{2+}]_0
\]

RESULTS

Neuron electrical activity results from the combined activity of a neuron’s membrane conductances. We reasoned that the reverse—that a neuron’s activity can be used as the fundamental driver in building models of the neuron—should also be true. Key to this process is altering neuron activity. That is, many different models can reproduce the activity of a neuron whose rest activity is an unvarying -60 mV membrane poten-
tial. When current is injected into the neuron, its responses reduce the number of models that reproduce its activity. For instance, if the neuron fires when depolarized, models with too little \( I_{Na} \) to support action potentials are excluded. More complex perturbation further decreases the number of models that can reproduce neuron responses. The goal is to provide sufficiently complex input that only one model reproduces neuron response.

The most complex input per time is continuously varying random input. We approximated continuously varying input with a series of step changes. Shorter steps give a better approximation. However, very brief steps are difficult to implement experimentally due to membrane and electrode time constants and amplifier settling times. We chose a step short enough (50 ms) to achieve large step numbers in relatively brief recording times, but long enough to be easily implemented with almost all neurons in both current and voltage clamp, and varied only step amplitude.

Target response to input was measured in current and voltage clamp in normal physiological saline and, to provide an additional condition involving changed conductance makeup, saline containing blockers of \( I_{Na} \), \( I_K \), and the A-type potassium current, \( I_A \). Input amplitudes were chosen to induce physiologically relevant and experimentally achievable responses. Current-clamp current injections induced membrane potentials between \(-125 \text{ mV}\) and action potential threshold. To avoid experimental difficulties with action potential control, voltage-clamp commands varied from \(-125\) to \(+10 \text{ mV}\) below spike threshold.

Three error measures were used. Voltage (in current clamp) and current (in voltage clamp) response errors were the areas \((\text{mV} \times s)\) and \((\text{nA} \times s)\) between target and model activity (gray shading, Fig. 1, A–D). Spike time error was the sum of the absolute values of the times between model and nearest target spikes (Fig. 1E1) plus the sum of the absolute values of the times between target and nearest model spikes (Fig. 1E2). Measuring these times both target to model and model to target was required to account for all missing and extra spikes (note that Fig. 1E1 has two arrows and Fig. 1E2 three).

These errors determined survival in an evolutionary algorithm (Fogel 1994; Yao et al. 1999) described in detail in METHODS. An evolutionary algorithm was chosen because it does not require calculating error derivatives, is resistant to being trapped in local minima, and is easily implemented on distributed parallel computers. In brief, it worked on an initial population of 1,000 model neurons with different maximum conductances. Each individual produced one child. Mates were chosen randomly. The pair reproduced sexually if the reproducing individual’s error was greater than that of its mate; otherwise, the individual reproduced asexually. Maximum conductances were inherited with mutation. Mutation amplitude was determined by random drawing from a Cauchy distribution and an inherited mutation size parameter. One thousand individuals from the combined parent and child population were then selected to be new parents using a procedure that favored low-error individuals but allowed some high-error individuals to survive. A different error measure and experimental condition (saline current clamp, saline voltage clamp, saline spike time, drug cocktail current clamp, drug cocktail voltage clamp) were used each generation. Note that the algorithm has no knowledge of target \( g_{\text{max}} \) values; survival is determined only by activity-based error measures.

It is important to note that the method presented here thus consists of three interacting parts: complicated driving input (Fig. 1), error functions (Fig. 1), and the algorithm used to determine individual survival based on these errors. To make this presentation as clear as possible, we use the term “method” to refer to all three processes operating as a whole and “algorithm” to apply only to the evolutionary algorithm itself.

To test the method against a wide variety of conductance sets, we used multiple neuron models as targets. One (leech tonic firer) had nine conductances whose properties (e.g., half-maximal activation voltage) and maximum conductance values were measured in leech heart interneurons (Nadim et al. 1995). A second (lobster burster) had eight conductances whose properties were measured in lobster stomatogastric neurons and whose maximum conductances produce an endogenous burster (Prinz et al. 2003). A third (generic tonic firer) had ten conductances modified from measurements in a variety of neurons. A fourth (generic burster) had these ten conductances with different maximum conductance values and an additional three conductances. Because the leech, lobster, and generic models were developed from different species, somewhat different half-maximal activation voltages and time constants were used even for conductances of the same type (e.g., \( I_H \)). The four models thus differed not only in maximum conductance values, but also in which conductances were present in each model and in each conductance’s properties.

Maximum conductance values of the initial population were assigned randomly across initial ranges either arbitrarily chosen or, for the lobster burster, those used in the model’s original source (Prinz et al. 2003). These choices were not particularly important because evolutionary algorithm performance depends relatively little on initial conditions (Fogel 1994; Yao et al. 1999) and because we used the results of earlier runs to define the initial ranges of later runs (see following text).

As we subsequently show, the method finds solutions that match target activity extremely well. In practice, the \( g_{\text{max}} \) values of the neuron being modeled would be unknown, and thus a user would halt the optimization either 1) because solutions with a predetermined level of error had been obtained (at which point multiple solutions with equally small error may be present) or 2) because further optimization did not decrease solution error (at which point all solutions are presumably very near or at the global minimum of the parameter space, and thus there is little or no further improvement the algorithm can achieve). However, since we were testing method performance and had foreknowledge of target \( g_{\text{max}} \) values, we could test the method’s ability to find the global minimum by explicitly comparing method to target \( g_{\text{max}} \) values. We therefore first present the data in terms of \( g_{\text{max}} \) values (Figs. 2–4) and only then (Fig. 5) present activity data.

In all cases the populations evolved similarly. Figure 2 shows the evolution of \( g_{\text{KL}} \) values for the first five runs of the generic tonic target. The gray dots are the maximum conductance values of the entire population for one run. The black line is the population mean. The range of maximum conductance values first expanded (from \(0–10\) to \(0–10,000 \mu S\) in the first 50 generations) and then collapsed with very small variation around a single value (runs typically continued for \(300–500\) generations).
generations, and final ranges were thus smaller than that shown here). This variation was so small that all members of the population had essentially equal errors. We therefore used the population’s mean maximum conductance values as the run’s final value.

The other runs from this initial maximum conductance range (colored lines) collapsed onto final values (inset) whose range (2–3.5 μS) was smaller than the initial range (0–10 μS). This evolution to a set of final values with a smaller range than the initial range occurred simultaneously for all model conductances. These new ranges were used as initial ranges for a second set of runs, which in turn collapsed onto a more tightly grouped set of final values. Each repetition typically resulted in a 10⁶- to 10¹⁴-fold decrease in the hypervolume defined by the range of maximum conductance values. We performed two to six such repetitions, each with at least five runs.

This process found maximum conductance values that showed excellent agreement with all four targets (Fig. 3). However, the very large y-axis ranges (up to six orders of magnitude) in this figure make it difficult to see relative differences between method and target values. We therefore normalized the data to target values (Fig. 4). These again showed excellent agreement, with mean method values of 38 of the 40 conductances falling within ±1% of the target and the remaining two within ±5%. Considering the values of the individual runs from which these averages were calculated, the values of all runs were within ±1% of the target for 32 conductances, ±5% for 37, and ±10% for all 40.

Although these data show that the method finds solutions very close to target maximum conductance values, as noted earlier, in practice these values would be unknown. The truly
relevanct comparison is thus between target activity and the activities of the method solutions. These comparisons show extremely good matches, with mean voltage differences (in current-clamp saline) between method and target activity of <1 mV for all 23 solutions, 0.5 mV for 22, and 0.1 mV for 18. This extremely good match is shown in Fig. 5, which plots the worst second of the best solution for each target; the two activities are indistinguishable for all four targets.

Although the data presented earlier show the method is successful under perfect conditions, real neurons present a number of additional problems, and it was important to test that the method also functioned well under more realistic conditions. One such problem is that other researchers will use different pseudorandom stimulus patterns, and it was therefore important to ensure that the method would work with other pseudorandom patterns. A second problem is that real neuron recordings inevitably contain extrinsic noise. To test whether these difficulties destroyed the method’s ability to find solutions that well reproduce neuron conductances, we added noise to the lobster burster traces in both current and voltage clamp in control saline and the drug cocktail, and reran the evolutionary algorithm (again with added noise, the new stimulation pattern, and from intermediate initial ranges, as in Fig. 6A), with the algorithm using the altered conductance equations, to test whether it would find new conductances that produced model activity more closely matching the activity of the target (Fig. 6, B1 and B2). The black traces show target activity with the original $g_{\text{max}}$ values and original conductance equations. The red traces show the activity of a model with the target $g_{\text{max}}$ values and the modified conductance parameters.

The model with target $g_{\text{max}}$ values and modified conductance parameters (red trace, Fig. 6B1) missed some bursts (those at 5, 10.5, and 28 s), had difficulties (four misplaced bursts where there were originally three) from 11 to 14 s, and had a badly timed burst at 23 s. The evolutionary algorithm, working with the conductance equations with modified parameters, optimized $g_{\text{max}}$ values sufficiently to cause the following improvements in model performance (green traces): the missing burst at 10 s was restored, the three bursts between 11 and 14 s were restored, and the timing of the burst at 23 s became closer to the target burst time. These $g_{\text{max}}$ changes also strongly improved spiking activity within the bursts (Fig. 6B2). Using target $g_{\text{max}}$ values and modified conductance parameters (red trace) produced single spike bursts with a pronounced shoulder. The $g_{\text{max}}$ values achieved after the optimization with the modified conductances produced activity (green traces) with spike number and timing much closer to target activity, as were also the slow wave membrane potential variations between the bursts.

When using equations with parameters different from those in the target, the optimization was not able to find a set of $g_{\text{max}}$ values that exactly reproduced target activity (for instance, the bursts at 5 and 28 s are still missing, and intraburst spiking is not perfect). However, this is not surprising given that the optimization was working with incorrect conductances. What is important is that the method still improved model performance, and thus the optimization procedure continues to function even when using equations that are different from those that produced the target activity. To do so with mismatched conductance equations, of course, resulted—required—in the method finding $g_{\text{max}}$ values different from those in the target (see Discussion for further consideration of this issue). However, these are $g_{\text{max}}$ values that, with the mismatched conductances, improved model activity.

An additional potentially serious error that could occur is that the description of the neuron being modeled is missing a conductance, and thus this conductance is not included in the model and hence not available for optimization. To test whether this would cause the method to fail, we removed $I_A$ have errors. It was thus possible that the method might fail to find improved solutions if the algorithm used equations in which these parameters differed from those in the target. For instance, the actual half-maximal voltage of a neuron’s $I_{\text{Na}}$ activation could be ~28 mV, but the value measured from the cell body—the value that would be used in the conductance’s equations—would be ~25.5 mV. To test the effect of such errors on method performance we altered all equation parameters by increasing or decreasing them by 10% (see Methods) and reran the lobster burster model using target $g_{\text{max}}$ values and the new conductance equations. Model activity of course no longer matched target activity because of the changes in the conductance equations. We then reran the evolutionary algorithm (again with added noise, the new stimulation pattern, and from intermediate initial ranges, as in Fig. 6A), with the algorithm using the altered conductance equations, to test whether it would find new $g_{\text{max}}$ values that produced model activity more closely matching the activity of the target (Fig. 6, B1 and B2). The black traces show target activity with the original $g_{\text{max}}$ values and original conductance equations. The red traces show the activity of a model with the target $g_{\text{max}}$ values and the modified conductance parameters.

The model with target $g_{\text{max}}$ values and modified conductance parameters (red trace, Fig. 6B1) missed some bursts (those at 5, 10.5, and 28 s), had difficulties (four misplaced bursts where there were originally three) from 11 to 14 s, and had a badly timed burst at 23 s. The evolutionary algorithm, working with the conductance equations with modified parameters, optimized $g_{\text{max}}$ values sufficiently to cause the following improvements in model performance (green traces): the missing burst at 10 s was restored, the three bursts between 11 and 14 s were restored, and the timing of the burst at 23 s became closer to the target burst time. These $g_{\text{max}}$ changes also strongly improved spiking activity within the bursts (Fig. 6B2). Using target $g_{\text{max}}$ values and modified conductance parameters (red trace) produced single spike bursts with a pronounced shoulder. The $g_{\text{max}}$ values achieved after the optimization with the modified conductances produced activity (green traces) with spike number and timing much closer to target activity, as were also the slow wave membrane potential variations between the bursts.

When using equations with parameters different from those in the target, the optimization was not able to find a set of $g_{\text{max}}$ values that exactly reproduced target activity (for instance, the bursts at 5 and 28 s are still missing, and intraburst spiking is not perfect). However, this is not surprising given that the optimization was working with incorrect conductances. What is important is that the method still improved model performance, and thus the optimization procedure continues to function even when using equations that are different from those that produced the target activity. To do so with mismatched conductance equations, of course, resulted—required—in the method finding $g_{\text{max}}$ values different from those in the target (see Discussion for further consideration of this issue). However, these are $g_{\text{max}}$ values that, with the mismatched conductances, improved model activity.

An additional potentially serious error that could occur is that the description of the neuron being modeled is missing a conductance, and thus this conductance is not included in the model and hence not available for optimization. To test whether this would cause the method to fail, we removed $I_A$
from the conductances the algorithm had at its disposal and reperformed the steps used to produce the data in Fig. 6, B1 and B2, again using the lobster burster model with added noise, the new stimulation pattern, and mismatched conductances (Fig. 6, C1 and C2). Interestingly, removing $I_A$ improved nonoptimized (red trace; compare with red trace in Fig. 6B1) burst performance, missing only the bursts at 5 and 28 s and lacking the difficulties between 11 and 14 s. Optimization slightly improved burst performance (e.g., improved burst time at 16 s) (Fig. 6C1). However, it again strongly improved intraburst spiking performance (Fig. 6C2). Thus even when completely lacking conductances present in the target, the optimization method still improved model activity.

**DISCUSSION**

The primary goal of this work was to determine whether, using presently available computational resources, neuron response to experimentally feasible perturbation contains sufficient information, using already defined conductance types but allowing maximum conductance amounts to vary, to find models that well reproduce neuron activity. The almost perfect correspondence between method derived and target activity (Fig. 5) shows that this is true for the four targets chosen. Four examples do not prove generality. However, the models chosen include a wide variety of conductances ($I_{\text{leak}}, I_{\text{Na}}, I_{\text{Kd}}, I_P, I_A, I_{\text{Kv3}}, I_{\text{Afast}}, I_{\text{Aslow}}, I_{\text{AHP}}$).
$I_{\text{KCa}}, I_{\text{h}}, I_{\text{CaS}}, I_{\text{CaF}}, I_{\text{CaT}}$ present over a very wide ($10^5$) $g_{\text{max}}$ range, which suggests that this method may be useful for building models of a wide variety of neurons.

It is essential to stress the central importance of perturbation of neuron activity and of stringent error measures to the method’s success. In early attempts with less complicated perturbation and less stringent error measures the method found widely separated solutions with approximately equally small error. Extensive perturbation is not typically performed when describing neuron properties and testing model adequacy. Our experience suggests that, as is common in analysis of other physiological systems (Marmarelis and Marmarelis 1978), it should be.

**Application to real neurons**

The ability of the method to continue to improve model activity when faced with noise, imperfect conductance descriptions, and even a missing conductance supports the argument that the method can profitably be applied to real neurons. The fact that the method could not find perfect solutions in the latter two cases suggests that in real neurons it will also not find solutions that perfectly reproduce neuron activity because of the difficulties in conductance characterization and identification mentioned in the INTRODUCTION. However, the central role of neuron activity in the method means that models built using the maximum conductance values it does find should be near the best possible with respect to reproducing neuron activity. Furthermore, the central role of complex perturbation in the method means that these models should also well reproduce neuron activity for all inputs within the perturbation’s dynamic and amplitude limits. The ability of the method to determine presumably near-best $g_{\text{max}}$ values for a given set of conductances and conductance equations suggests that, if method solutions do not meet user standards, then model improvement likely requires changes in conductance makeup, equation parameters, or multicompartmentalization.

**FIG. 4.** Comparison of target and method final run maximum conductance values normalized to target values: 0% line marks target; other conventions as in Fig. 3.
initial ranges (e.g., that spike-generating conductances are located only in distal compartments) are often known, which would greatly reduce computation time.

What does “best” mean with respect to model parameters and neuron description?

As explained in the Introduction, all measurements of conductance properties and amounts measured from one site (often the neuron cell body), inherently and unavoidably contain errors due to morphological filtering and attenuation. It is to overcome this difficulty that hand tuning or brute force parameter optimization, or the method described here, is required. The method described here, particularly if extended to optimizing conductance parameters and placement as suggested earlier, may seem to some readers insufficiently grounded in “hard” data. That is, it may seem that descriptions of conductance time constants, half-maximal voltages, and amounts derived from direct measurements of individually isolated conductances are more nearly “correct” than those derived from a method that finds those values simply by matching neuron and model activity.

However, it cannot be overstressed that in each case the values obtained are gross oversimplifications. For instance, the very small amounts of \( I_{\text{Na}} \) that standard techniques measure for neurons in which the sodium channels are electrically distant from the cell body are certainly not “correct” for action potential generation in a single-compartment model. The method presented here is specifically designed to find maximum conductance amounts (or, if extended, other conductance properties and compartment placement) that best reproduce neuron activity, and reproducing neuron activity is the ultimate test of whether a model is correct. With respect to building neuron models, we would thus argue that the values found by the optimization method described here or similar methods are more nearly correct.

With respect to describing a neuron’s actual membrane complement, which technique gives a truer description depends on the neuron being investigated and investigator goal. For electrically compact neurons standard techniques and the activity-dependent technique described here will give similar (and true, in that little filtering is occurring) results. The ability of the activity-dependent technique to measure all neuron parameters from individual neurons would thus make it the preferred method for such neurons. For neurons with extended anatomies and unequal placement of membrane conductances across the neuron processes, standard techniques will give low-pass-filtered and reduced amplitude (relative to what is actually present where the conductance is located) measurements of distant conductances. The activity-dependent technique will give different values, finding those necessary to reproduce neuron activity. In both cases, more accurate descriptions of neuron anatomy will result in more accurate descriptions of neuron conductances. Given that each is a cluster of approximations, which is better is a matter of opinion, but the ability of the activity-dependent technique to approximate all membrane conductances simultaneously suggests that in many cases it would still be preferred. Membrane conductances are often measured to compare different neuron types (see following text for discussion of comparison within a neuron type), often with the implicit or explicit goal of understanding why the neurons have different activities. In these cases, given its fundamental dependence on activity, the activity-dependent technique would likely again be preferable.

Model performance can be further enhanced by expanding the parameters it optimizes. For instance, allowing it to adjust conductance time constants and voltage dependencies would presumably further decrease model error. Even more accurate representations could be obtained using multicompart model models. A major difficulty in such models is correctly placing the conductances in the compartments. Since this process simply represents extra parameters, modifying the method described here to identify which placings best reproduce neuron activity is straightforward. In both cases the primary difficulty is the increased parameter space that must be searched. However, experimental data to guide
FIG. 6. Effect of adding noise and using a different stimulus pattern (A), of adding noise, using a different stimulus pattern, and using imperfect conductance descriptions (B1, B2), and of adding noise, using a different stimulus pattern, using imperfect conductance descriptions, and removing $I_A$ (C1, C2) on method performance. In A the method continued to find a model with essentially perfect activity (the second shown is the worst in the entire 30 s). In B and C the method could not exactly match target activity, but optimization still improved model performance. Black traces, target activity; red traces, model activity with target maximal conductance ($g_{\text{max}}$) values but incorrect or missing conductances; green traces, model activity after $g_{\text{max}}$ optimization.
Can multiple conductance sets make the same neuron type?

Whether neurons of the “same” type—that is, neurons with similar intrinsic activity and responses to input—can be constructed with different maximum conductance amounts remains controversial (Nowotny et al. 2007; Szu ßchs and Selverston 2006). The method described here can help to resolve this issue in three ways. First, the neurons in question have never been subjected to the complicated perturbations used here. It is possible that when the neurons are so perturbed they will have several different types of response (for each of which the method would find different, statistically separate, conductance value sets as solutions). In this case this issue would be resolved in that these neurons actually constitute several separate neuron classes and have appeared to be similar only because they have not been examined in sufficient detail. Second, the neurons may instead show a graded continuum of responses centered on a mean. In this case the method would find a different best solution for each neuron, with the range of solutions found reflecting the variability across the group of neurons. Third, it is possible that when the neurons are so perturbed they will have been subjected to the complicated perturbations used here. It is important to make one final comment on this issue. Even in the case in which the neurons fall into separate classes when strongly perturbed (case 1 in the preceding paragraph), this does not necessarily mean the neurons are not functionally the same. It is unclear that real neurons would ever need to respond “correctly” to input this complicated. If real neurons need to perform only a more limited set of tasks, neurons with different conductance compositions, which our method would identify as being different, could nonetheless be functionally equivalent.

ACKNOWLEDGMENTS

We thank R. A. DiCaprio for extensive discussion and comments on the manuscript and the Ohio University Physics Department for access to its proprietary cluster.

GRANTS

This work was supported by an Ohio Supercomputer Center (OSC) Cluster Ohio Initiative grant for the Ohio University Department of Biology parallel computing facility, support from Ohio University Computer and Network Services, grants totaling 20,000 resource units on the OSC facility, a guest professorship of the University of Köln to S. L. Hooper, National Science Foundation Grant 90250, and National Institute of Mental Health Grant MH-57832.

DISCLOSURE

The authors have no potential conflicts of interest.

REFERENCES


