On Parameter Estimation for Neuron Models *

Jeffrey L. Madden [1] Zina Ben Miled[1] Raymond C.Y. Chin [2] John Schild [1]
[1] *ECE Dept., Purdue School of Eng. & Tech.* [2] *CSCI Dept., Purdue School of Science*
Indiana University Purdue University Indianapolis

Abstract

Membrane bound ion channels give rise to many of the electrical signal characteristics exhibited by neurons. Ion channel models of neural function such as that proposed by Hodgkin-Huxley can be represented as a set of differential equations. Solving these differential equations for a given neuron involves finding optimal values for the parameters that define the Hodgkin-Huxley equations. Most often, these parameters are evaluated using an optimization algorithm that takes as input ion channel current data recorded from a neuron using the voltage clamp technique.

Real-valued optimization algorithms often fail to find a global optimum for the parameters of the Hodgkin-Huxley differential equations. In this paper, we show that interval analysis based optimization algorithm, a branch and bound algorithm, provides an accurate solution for the Hodgkin-Huxley model.

1 Introduction

A neuron can be described by an equivalent electrical circuit with both active and passive components. This circuit can be modeled as a parallel combination of a capacitor and conductance (Figure 1). The shunt capacitor represents the inherent capacitance of the neural membrane. The conductor is a lumped representation of the thousands of ion channels forming selective pores through the neural membrane. These pores open and close in a voltage and time dependent matter and allow selected ions to flow along its concentration gradient, as represented by the series battery.

A neuron membrane includes various families of ion channels each controlling the flow of a specific ionic current in or out of the cell. Hodgkin and Huxley formulated a model for ionic channel currents in giant squid axons that provides a generalized description of their electrical properties [1]. A *gating variable* was added to the equation describing the circuit of Figure 1. This gating variable represents the observed behavior of a channel as it opens or closes, thus controlling the flow of ions through the cell membrane. A channel may have more than one gating variable associated with it, where some gating variables control the *activation* (channel opening) of the ionic current, and others control the *inactivation* (channel closing) of the same current [1]. These gating variables are time and voltage dependent and they can range in value from 0 to 1. They modify the cell conductance for the corresponding ion, thus acting as a scale factor on the maximum cell conductance, $\overline{g}$. A Hodgkin-Huxley equation for a given ionic current equation is as follows:

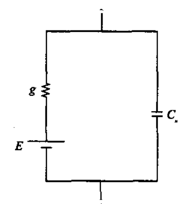

Figure 1: **Circuit Representation of a Neuron.**

$$I(t,V) = \overline{g} m^x(t,V) h^y(t,V)(V-E) \qquad (1)$$

where $\overline{g}$ is the maximum whole-cell conductance for that particular current. The parameters m and h are activation and inactivation gating variables raised to the power x and y, respectively. The voltage V is the membrane potential and E is the Nernst potential for the given ion.

The gating variables are described by relaxation equations of the form

$$\dot{z} = \alpha_z(1-z) - \beta_z z \qquad (2)$$

where $\dot{z}$ represents the first time derivative of an arbitrary gating variable z. A more common form of the relaxation equation describing the gating variable is

$$\dot{z} = \frac{z_\infty - z}{\tau_z} \qquad (3)$$

*This research was supported in part Sun Microsystems & Co and Purdue School of Engineering & Technology, Indianapolis.

0-7695-0862-6/00 $10.00 © 2000 IEEE

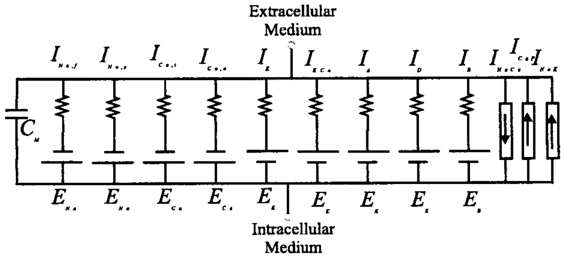

Figure 2: **Parallel Circuit Representation of a Neuron.**

where z_∞ is the steady-state value of z (the value of z as time goes to infinity), and τ_z is the time constant associated with z as it approaches z_∞. The parameters z_∞ and τ_z can be expressed in terms of α and β as

$$z_\infty = \frac{\alpha(V)}{\alpha(V)+\beta(V)} \tag{4}$$

$$\tau_z = \frac{1}{\alpha(V)+\beta(V)} \tag{5}$$

A Hodgkin-Huxley model for the action potential discharge in rat sensory neurons was introduced in [2]. The essential formalisms for this equivalent circuit model are consistent with those originally proposed by Hodgkin-Huxley (Figure 2).

The voltage across this circuit can be described by the following differential equation:

$$\dot{V} = -\frac{\sum I_{ion} - I_{stim}}{C_m} \tag{6}$$

where I_{stim} is a stimulus current applied to the cell to evoke its natural response, and I_{ion} represents the individual electric currents produced when sodium, calcium and potassium ions flow in or out of the cell membrane.

In [2], the steady-state value for the gating variable was described using a standard Boltzmann function, while the time constant was given in a simple Gaussian form. However, in order to more realistically describe the channel operation of the cell, these two variables are reexpressed using a forward (α) and reverse (β) constants which are given by

$$\alpha = \frac{x_1(x_2 - V)}{\exp(\frac{x_2 - V}{x_3}) - 1} \tag{7}$$

$$\beta = x_4 \exp(-\frac{V}{x_5}) \tag{8}$$

The parameters x_1 through x_5 are dependent on the specific ion channel current and vary from cell to cell. In this form, α and β are rate constants of the transition of the gating variable from a nonpermissive form to a permissive form (α) and from a permissive form to a nonpermissive form (β). This $\alpha - \beta$ format is consistent with the Hodgkin-Huxley 2-state model of channel gating [1].

As shown in Equations 7 and 8, each gating variable has five parameters that are dependent on the properties of the cell. These parameters must be evaluated for each cell individually in order to create a model that accurately describes the behavior of the cell. In this paper we show that the interval analysis optimization method can provide a solution for these parameters that is more accurate, in the least square sense, than the traditional optimization algorithms.

2 Parameter Estimation

A cell has several ion channel currents. In order to develop a method of estimating the gating variable parameters, we selected a single, representative current : the delayed rectifier current (I_K) which is given by

$$I_K = \overline{g}_K n(V - E_K) \tag{9}$$

where the gating variable, n, is given by:

$$\dot{n} = \frac{n_\infty - n}{\tau_n} \tag{10}$$

The variable n_∞ is the steady-state value of n (the value of n as time goes to infinity), and τ_n is the time constant associated with n as it approaches n_∞. Under a constant voltage condition, this differential equation has a closed form solution given by

$$n = n_\infty(1 - e^{-t/\tau_n}) \tag{11}$$

Substituting this back into Equation 9 yields the following:

$$I_K = \overline{g}_K(V - E_K)[n_\infty(1 - e^{-t/\tau_n})] \tag{12}$$

Under normal condition, the membrane voltage is not constant which makes solving Equation 9 more complex. In this case, the solution can be determined based on several set of current data where each set of current data is experimentally recorded when the cell membrane is held constant at a given voltage value. This process is called voltage clamping and is described next.

2.1 Voltage Clamp

The cell membrane can be held at an approximately constant voltage in a technique known as voltage clamping. While the voltage is held constant, current data is recorded from the cell.

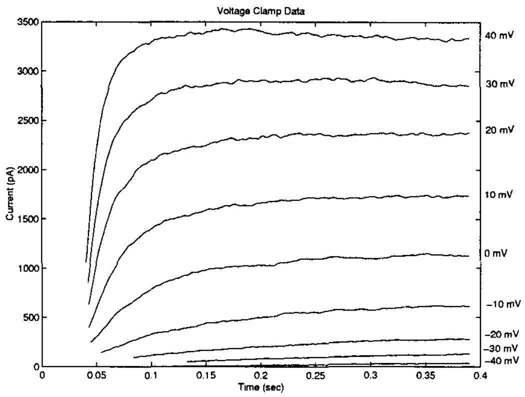

Figure 3: **Current versus Time in Voltage Clamp Conditions (capacitor artifact has been removed).**

The cell membrane can be held at an approximately constant voltage in a technique known as voltage clamping. While the voltage is held constant, current data is recorded from the cell. This current data represents the voltage and time-dependent gating characteristics for a given channel. The membrane voltage is then varied and a new set of current data is collected. The number of sets collected depends on how thoroughly the activation characteristics need to be described. Generally, voltage clamp data is gathered at multiple voltages ranging from -100 millivolts to 60 millivolts which spans the natural operating range of the cell. Figure 3 shows the data gathered using voltage clamp for the I_K current that will be studied in this paper. As the voltage clamp is applied to the cell, the membrane potential is changed from equilibrium to the desired voltage as if it were a simple RC circuit that is being charged. The membrane current being measured in the voltage clamp technique is described as follows:

$$I_M = I_i + I_c = I_i + C_M \frac{dE}{dt} \qquad (13)$$

where I_i is the current due to the movement of ions across the membrane in response to the changing membrane voltage and I_c is the current due to the charging or discharging of the membrane capacitance. Equation 13 shows that I_c is dependent on $\frac{dE}{dt}$, which goes to zero after several time constants. As $\frac{dE}{dt}$ goes to zero, I_i dominates the current data. The voltage clamp data is truncated to delete the RC remnants at the beginning of the data since the current data under study is I_i.

2.2 Objective Function

In order to find the best estimates to current parameters, an objective function is defined and minimized in the mean-square sense. The objective function for the potassium current is as follows:

$$F = \sum_{i=1}^{k} \sum_{j=1}^{l} (y_j - \overline{g}_K(V_i - E_K)n_\infty(V_i)(1 - e^{-t_j/\tau_n(V_i)}))^2 \qquad (14)$$

where k is the number of data sets obtained when performing the voltage clamp at k different voltages, y_j is the value of the current at time t_j and l is the number of data points in each data set. The parameters x_1 through x_5 of Equations 7 and 8 are then manipulated to find the values that result in the minimum value of F.

An alternative approach to the objective function in Equation 14 is to fit each data set individually using the following objective function:

$$F = \sum_{i=1}^{j} (y_i - \overline{g}_K(V - E_K)n_\infty(1 - e^{-t_i/\tau_n}))^2 \qquad (15)$$

where y_i is the measured current at time t_i and j is the number of data points in the voltage clamp data gathered at a single value of V. Here, the objective function can be minimized by direct manipulation of the parameters x_1 through x_5, or simply by varying α and β. In either case, values for α and β are found for each value of V. These values are then used to find final values for x_1 through x_5 by minimizing the following two functions in the mean-square sense:

$$F_\alpha = \sum_{i=1}^{k} (\alpha_k - \frac{x_1(x_2 - V)}{\exp(\frac{x_2 - V}{x_3}) - 1})^2 \qquad (16)$$

$$F_\beta = \sum_{i=1}^{k} (\beta_k - x_4 \exp(-\frac{V}{x_5}))^2 \qquad (17)$$

In the remainder of this paper, we will refer to the approach of Equation 14 as the global optimization approach. Similarly, the approach defined by equations 15, 16 and 17 as the individual optimization approach.

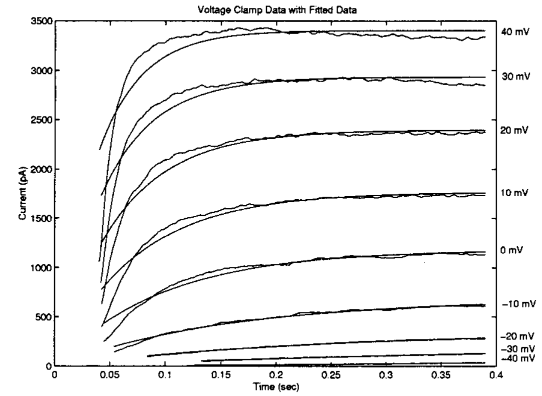

Figure 4: **Current versus time in voltage clamp conditions with fitted Data. The smooth curves represent the fitted data. The other curves are the original voltage clamp data. The current data sets have been fit individually using the Levenberg-Marquardt method.**

2.3 Traditional Methods

The first algorithm used to minimize Equation 15 was the Levenberg-Marquardt method [4] [5], implemented in the Optimization Toolbox of MATLAB, using the function call "curvefit" [6]. Using this toolbox, only one set of data may be fit at a time. Therefore, the individual optimization approach (equations 15, 16 and 17) must be used. When only fitting the current data corresponding to one voltage value, the algorithm was successful. Figure 4 shows the result of fitting each current data set individually. This figure shows that the calculated values agree with the recorded data. The computational time needed to fit one data set was 0.738 seconds on average when run on a dual processor Ultra Sparc 60. The average number of data points in the data sets was 540.

The next step was to use the values for α and β obtained from each individual data set to determine unique values for the five parameters of equations 7 and 8. In this case the voltage is the independent variable. It is during this step that the method failed. Figure 5 shows the result of substituting the new values for the parameters into Equations 7 and 8, and then substituting the resulting values of α and β into Equation 12. The calculated currents no longer adequately modeled the data. Furthermore, as shown in Figures 6 and 7, α and β do not behave as expected. The average CPU time needed to solve the complete problem was 4.6 seconds.

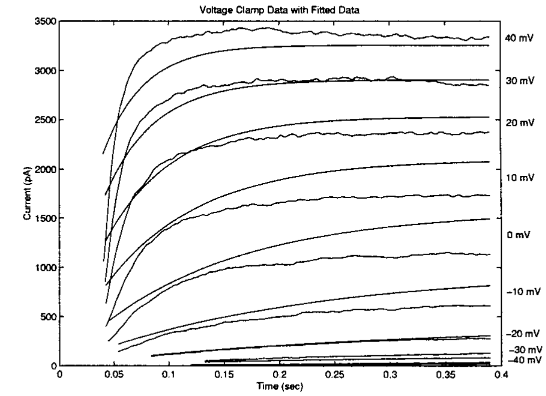

Figure 5: **Current versus time in voltage clamp conditions with fitted data. The smooth curves represent the fitted data. The other curves are the original voltage clamp data. The current data sets have been calculated by first using the values of α and β from individual current sets (Figure 4) to derive unique values for the parameters x_1 through x_5. These values are then used to calculate the current through a series of substitution into equations 7, 8 and 12.**

In an attempt to remove some of the noise from the original current data, the data was down sampled and filtered using a Discrete Cosine Transform. This process was performed in an attempt to obtain better parameter values in a shorter amount of computational time. First each data set was down sampled by a factor of four, then every eight data points were put into the DCT, only the DC coefficient was kept, and this coefficient was processed through an Inverse DCT. The filtered input current data kept the basic characteristics of the original, but did not have as much noise. When applying the Levenberg-Marquardt method to this filtered data, the final values were within an average of 6% of the values obtained with the non-filtered data, and the reduction in CPU was 10% on the average.

These previous attempts indicated that a different method for solving this optimization problem is needed.

3 Interval Analysis

A branch and bound technique called interval analysis [7] was selected. Although this method has not been previously used to solve this problem, it was suc-

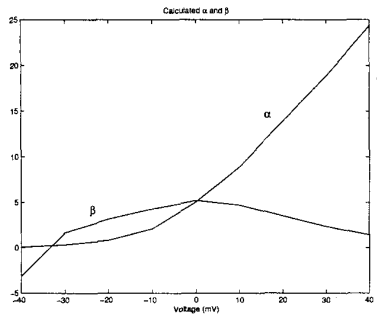

Figure 6: α **and** β **as calculated.**

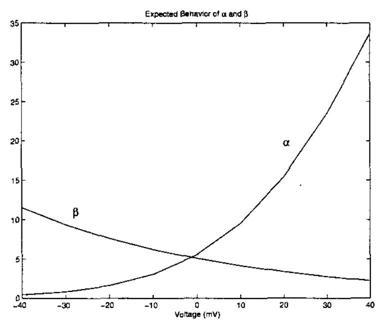

Figure 7: **Expected behavior of** α **and** β**.**

cessful in solving problems with similar characteristics [7, 8, 9, 10, 11, 12, 13]. The following section describes the interval analysis method.

Interval Analysis represents a real number by an interval that contains the real number. It was originally developed as a way to bound computational error and since has grown into a global optimization tool, as well as a means for verifying other types of mathematical computations such as finding the zeros of a function and Gaussian elimination [14]. The rules of Interval Arithmetic, which are summarized in [8], are as follows: Given intervals $X = [\underline{x}, \overline{x}]$ and $Y = [\underline{y}, \overline{y}]$, where $\underline{x}, \overline{x}, \underline{y}, \overline{y}$ are real numbers such that $\underline{x} \leq \overline{x}$ and $\underline{y} \leq \overline{y}$, and a binary operator $\circ$, the following is true:

$$X \circ Y = [\min x \circ y, \max x \circ y] \text{ for } x \in X \text{ and } y \in Y. \tag{18}$$

Also, given a unary function $\phi(x)$,

$$\phi(X) = [\min \phi(x), \max \phi(x)] \text{ for } x \in X \tag{19}$$

A comprehensive introduction to interval analysis can be found in [15] with applications in [16]. The basic rules of interval arithmetic are presented in [9]. In addition, [9] includes a simple algorithm for solving global optimization problems. Branch and bound example algorithms using interval analysis are included in [10]. A brief survey of interval based optimization algorithms is given in [17]. The following section describes interval analysis in order to show how it was used to obtain an accurate model for the ionic current I_k.

3.1 Notation

Table 1 includes the notation used in the rest of this section. Some of this notation applies to either real-valued variables or vector-valued variables. For example, the objective function of Equation 15 is a function of x_1 through x_5. These parameters can be grouped in a five-dimensional vector x which can be considered a vector-valued variable where X is the corresponding interval vector.

3.2 Optimization Algorithm

Given an n-dimensional cube X (where n is the number of parameters to be evaluated), the branch and bound interval algorithm introduced in [9] is as follows:

1. Place X into an ordered list L.
2. Bisect the first cube of L into two sub cubes V_1 and V_2.
3. Delete V_i if it can be proven that V_i contains no solution or diminish V_i if it can be proven that a part of V_i contains no solution (where $i = 1, 2$).
4. Place V_i (as a whole or diminished) into list L if V_i has not been deleted in Step 3 (where $i = 1, 2$).
5. Stop if termination criteria hold.
6. Goto Step 2.

The performance of the algorithm depends on how well it can delete or diminish cubes in the list L. As stated in [11], there are different methods for deleting or diminishing cubes from the list L, depending on whether or not the objective function and any constraints are differentiable. The constraint functions are of the form $g_i(x) \leq 0$, where $i = 1, \ldots, k$, and k is the number of constraints.

The two simplest tests on a given cube do not depend on any derivative of the objective or constraint functions. The first is a simple feasibility test. Using interval arithmetic, each constraint function is evaluated on the cube X. If $g_i(X) \leq 0$ for all $i = 1, \ldots, k$,

Symbol	Definition
X	An interval valued variable
$\underline{x}$	*Lower bound* of X, a real number or real vector in X such that $x \geq \underline{x}$ for all $x \in X$
$\overline{x}$	*Upper bound* of X, a real number or real vector in X such that $x \leq \overline{x}$ for all $x \in X$
$F(X)$	*Interval extension* of the real-valued or vector-valued function $F(x)$, $F(x)$ evaluated on the interval X
$m(X)$	*Midpoint* of X, a real number or real vector in X such that $m(X) = \underline{x} + \frac{\overline{x}-\underline{x}}{2}$
$g(x)$	Constraint function
$G(x)$	Gradient of $F(x)$
$H(x)$	Hessian of $F(x)$

Table 1: **Interval Analysis Notation.**

then X is labeled *certainly feasible*. If, for any $i = 1, \ldots, k$, $g_i(X) > 0$, then X is *certainly infeasible* and can be eliminated from further consideration. The second test is the midpoint test. Let $m(X)$ be the midpoint of the cube X, or $m(X) = \underline{x} + \frac{\overline{x}-\underline{x}}{2}$. If X has passed the feasibility test, then the objective function is evaluated at $m(X)$, obtaining the interval $F(m(X)) = [\underline{m(X)}, \overline{m(X)}]$ where $\underline{m(X)}$ is the lower bound of $F(m(X))$ and $\overline{m(X)}$ is the upper bound. It can be shown that $\overline{m(X)} \geq F^*(X)$ where $F^*(X)$ is the minimum value of the objective function on X. The lowest $\overline{m(X)}$ found is labeled $\overline{m(X)}^*$. If F is then evaluated over another cube Y to find $F(Y) = [\underline{y}, \overline{y}]$, and $\underline{y} > \overline{m(X)}^*$, then Y cannot contain a *global* minimum of F and can be eliminated from further consideration.

The objective function used here, Equation 15, is twice continuously differentiable everywhere that the function exists, and therefore tests that depend on the first two derivatives of the function can be applied. The first is a monotonicity test. For a global minimum to occur in a cube X, the gradient of the objective function must be zero somewhere in X. The gradient on X is evaluated using interval analysis, to obtain $G_i(X) = [\underline{G_i}, \overline{G_i}]$, where $i = 1, \ldots, j$ and j is the number of parameters to be estimated. Then, if $\underline{G_i} > 0$ or $\overline{G_i} < 0$, X can be eliminated from further consideration.

Next, X is tested for nonconvexity. For a global minimum to be contained in X, the Hessian, $H(X)$, must be non-negative definite at the solution x^*. A necessary condition for this to occur is that the diagonal elements of $H(X)$ must be non-negative. The diagonal elements of $H(X)$ can be evaluated as $H_{ii}(X) = \delta^2 f(X)/\delta x_i^2$ for $i = 1, \ldots, n$. If $H_{ii}(X) < 0$ for any i, then X can be eliminated from further consideration.

Two more tests that make use of the differentiability of the objective function can be applied to this problem. One is an interval Newton method [18], and the other is a method that uses the upper bound of F^* as found in the midpoint test to eliminate portions of X when all of X cannot be eliminated. The details of these tests are very complex and can be found in [19] and [20], respectively.

Another important aspect of the algorithm is the subdivision of each cube. There are many schemes for subdividing cubes that depend on the geometry of the problem. However, most interval analysis based algorithms use rectangular partitioning, since the solution space consists of an n-dimensional rectangle [8]. In rectangular partitioning, each cube is bisected using various criteria. The simplest criterion is to bisect the cube in the direction of the longest edge. This is the method used by VerGO [8]. VerGO is the tool that we modified in this paper and applied to solving the ionic current optimization problem. Other schemes make use of the objective function and its derivatives [12].

Finally, the algorithm must provide a stopping criteria. Ideally, an optimization algorithm would stop when it is known that the given answer is within an acceptable error of the correct result. This is easily accomplished using interval analysis by setting the algorithm to continue until every cube left is smaller than or equal to the acceptable error in every dimension. If the acceptable error is some $\epsilon > 0$, and a cube X that contains a feasible solution is less than ϵ in its greatest dimension, then the midpoint of the cube, $m(X)$, is guaranteed to be within $\epsilon/2$ of the global minimum F^*.

4 Evaluation

A branch and bound algorithm will split the solution space into regions (branch) and find the least upper bound of the objective function for all of the regions (bound). Any part of the solution space whose lower bound is greater than the least upper bound can then be eliminated. Interval analysis branch and bound techniques have the advantage of being able to find (if one exists) a verifiable global solution to optimization problems. In this paper, the Verified Global Optimization [8] (VerGO) tool was modified to solve the optimization problem of the potassium ionic current.

After being modified to solve a data fitting problem, VerGO was first employed to solve the parameter estimation of one current data set. The average minimum objective function value found by VerGO was 7.2538×10^6, which is twice that of the MATLAB solution. With a priori knowledge of the optimum values for each data set, the simultaneous solutions can be completed in under 1,200 seconds, the longest solution taking 1,152 seconds, the shortest 55.07 seconds, as shown in Table 2.

Voltage (mV)	CPU Time (sec.)	x_1	x_2	x_3	x_4	x_5
40	1,096	1.00	23.02	14.47	24.98	14.93
30	1,100	1.50	24.87	9.69	29.37	11.87
20	1,152	1.50	23.75	11.06	23.12	10.62
10	1,109	1.50	22.62	13.47	20.31	10.31
0	742	1.50	21.87	18.28	20.31	14.37
-10	419	1.50	21.87	18.28	20.31	19.69
-20	182	1.50	24.87	19.66	20.31	19.69
-30	55	1.50	20.37	19.66	20.31	19.69
-40	179	1.50	20.37	18.28	25.31	19.37

Table 2: **CPU Times and Parameter Values for Nine Voltage Values Fitted Individually using Interval Analysis (Dual Processor Ultra 60, 360Mhz).**

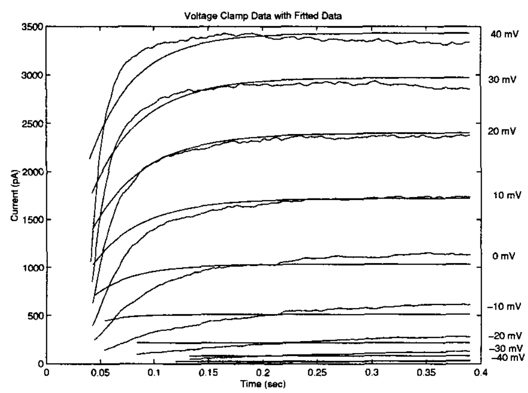

Figure 8: **Voltage clamp data and fitted Data. The smooth curves represent the the fitted data. The other curves are the original voltage clamp data. The current data sets have been fit individually using an interval analysis branch and bound technique.**

Figure 8 shows the currents plotted with these parameter values along with the original data. After these values are found, again with a priori knowledge of the optimum solutions, α and β can be calculated in under one minute, α taking 40.94 seconds and β 0.69 seconds. The results of these calculations are shown in Figure 9, along with the original data.

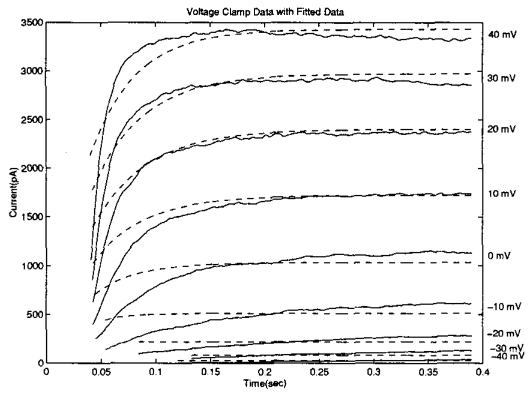

Figure 9: **Voltage Clamp Data and Fitted Data. The solid curves are the original voltage clamp data and the dashed curves are the fitted data. The current data sets have been calculated by first using the values of α and β from individual current sets (Figure 8) to derive unique values for the parameters x_1 through x_5. These values are then used to calculate the current through a series of substitution into equations 7, 8 and 12.**

Based on the value of the objective functions and a comparison of Figure 5 and Figure 9, the interval analysis based approach produces acceptable results. With no a priori knowledge of the correct solution, a small initial search area was given to VerGO that was based on the solution found using the Levenberg-Marquardt method of Section 2.3. There is no guarantee that the global minimum solution is contained in this area. Therefore, the initial search area should be as wide as possible. Using one voltage clamp data set and a large initial search area, VerGO can take as long as nine hours. Several different schemes were used to preprocess the data in order to reduce the computational time of this approach. The data was down sampled by factors of 4, 16, and 32 and then filtered with the DCT. The data was also filtered first, and then down sampled. Also, down sampling was used with no DCT and the DCT filter was used with no down sampling. The minimum objective function values and the CPU times for these different cases are shown in Table 3.

Preprocessing Method	CPU Time (sec.)	F_{min} ($\times 10^3$)
None	28,640	43.527
DCT, Down sample by 32	3,745	1.1326
DCT, Down sample by 16	4,614	0.5687
DCT, Down sample by 4	13,926	1.9924
DCT, No Down sample	34,379	10.753
DCT, Down sample by 32	1,291	0.7038
DCT, Down sample by 16	3,125	2.6942
DCT, Down sample by 4	14,767	3.503
No DCT, Down sample by 32	1,073	1.657
No DCT, Down sample by 16	2,713	2.9249
No DCT, Down sample by 4	9,078	2.025

Table 3: **CPU Times and Minimum Objective Function Values for Various Data Preprocessing Methods.**

The longest computational time was associated with the no down sampling case, because this case used the highest number of data points. The shortest computational time was when the data was down sampled by a factor of 32, without using the DCT filter. In most cases, when the down sampling factor was higher than 4, unacceptable results were produced. An example of unacceptable results can be seen in Figure 10, where the input current data is first filtered and then down sampled by a factor of 32.

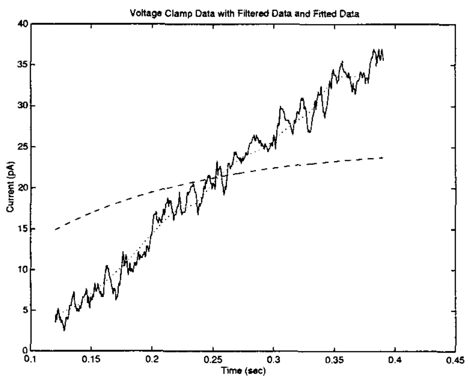

Figure 10: **Voltage Clamp Data with Filtered and Fitted Data. The solid curve is the original data, the dotted curve is the data after is has been filtered with an eight-point Discrete Cosine Transform and then down sampled by a factor of 32, and the dashed curve is the fitted data.**

Preprocessing Method	Average CPU Time (sec)	Average F_{min} ($\times 10^6$)	F_{min} at $V =$ -40mV ($\times 10^3$)
None	11.400	3.6659	40.73
DCT, Down sample by 32	1.100	13.743	1.350
DCT, Down sample by 16	0.861	3.8812	1.635
DCT, Down sample by 4	2.890	3.6617	10.01
DCT, No Down sample	11.100	3.6787	39.98
DCT, Down sample by 32	0.720	3.9465	1.207
DCT, Down sample by 16	0.738	3.6710	2.540
DCT, Down sample by 4	2.930	3.6909	9.930
No DCT, Down sample by 32	0.980	4.2785	1.542
No DCT, Down sample by 16	0.731	3.8876	2.898
No DCT, Down sample by 4	2.850	3.6601	10.41

Table 4: CPU Times and Minimum Objective Function Values for Various Data Preprocessing Methods

The best result in terms of objective function came when the data was down sampled by 4 and with no filter applied. This result took 2 hours and 31 minutes to calculate. The data used in this analysis was gathered at a low membrane voltage of -40 millivolts. This current is more linear than that at other voltages. For this reason, this data set is more difficult to fit using Equations 7 and 8. This difficulty is precisely why this data set was chosen for the analysis. If the technique developed in this analysis can find acceptable parameter values to fit this data set, then other data sets should also be successfully fit using this technique. Also, this case shows that when the data is not preprocessed either through down sampling or filtering, an unacceptable fit is obtained. Therefore, the data must be preprocessed.

Based on these high computational times, a different approach was adopted. In this new approach, the interval analysis method was used to estimate α and β directly from the data. This amounts to Equation 15 being represented as

$$F = \sum_{i=1}^{j} (y_i - \overline{g}_K(V - E_K)[\frac{\alpha}{\alpha+\beta}](1 - e^{-t_i / \frac{1}{\alpha+\beta}}))^2 \quad (20)$$

Because this form of the objective function reduces the number of parameters from 5 to 2, the complexity of the algorithm is also greatly reduced. This is reflected in the average CPU times shown in Table 4. The data in this table is averaged over all nine data sets ($V = -40, -30, -20, -10, 0, 10, 20, 30, 40$ millivolts) except for the last column, which shows F_{min} for $V = -40$ millivolts. This value is shown separetly in order to compare this method directly with the first interval based method used. This method results in minimum objective function values on the same order as that obtained in the first method. However, the computational time are at least two order of magnitude lower than in the first attempt at using the interval analysis based approach to solve the ionic current equation. This indicates that for the ionic current optimization problem, it is very important to express the problem in a mathematical formulation that is appropriate for a given optimization algorithm.

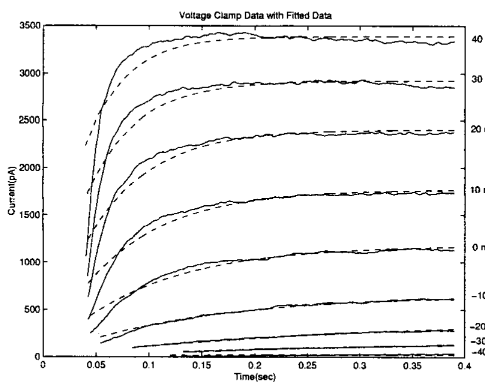

Figure 11: **Voltage clamp data with fitted data obtained using the individual optimization approach and the objective function of Equation 20.**

Figure 11 shows the original data plotted with the fitted data obtained from individually fitting α and β, while Figure 12 shows the result of using the α and β values to fit x_1 through x_5. These results are more accurate than the one presented in [2].

The interval analysis is an efficient optimization algorithm for solving ionic current equations. However, the formulation of the interval based objective function could lead to problems when the function is evaluated in a cube that is in the neighborhood of the global minimum [13]. Different forms of the objective function may lead to better performance. This is due to slight differences in interval arithmetic when compared to real arithmetic. For example, under interval arithmetic, multiplication is not distributive [14]. Given the intervals $A = [-1, 2]$, $B = [-3, 4]$ and $C = [5, 6]$, the function $D = A(B + C)$ can be evaluated in two ways. First the multiplication is distributed resulting in $D = [-6, 8] + [-6, 12]$. After the addition, the result is $D = [-12, 20]$. However, if the addition is done first, $D = [-1, 2] * [2, 10]$ which results in $D = [-10, 20]$. This latter result is the correct result. While the first result is not wrong in that it does *contain* the correct answer, it does not produce the tightest bounds on the answer. In [7] this is labeled the "dependency problem". The dependency problem does not change the correctness of an algorithm. However, it will result in an over-estimation of the bounds of the problem.

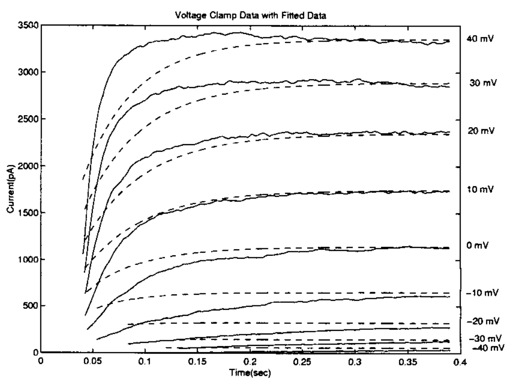

Figure 12: **Voltage clamp data with fitted data btained using the global optimization approach and the objective function of equations 16 and 17.**

5 Conclusion

The problem of finding an accurate model for ionic current is difficult because of the non-linear nature of the parameter estimation. Methods based on real analysis suffer from the tendency to find local minimums to the objective function. In this paper we showed that interval analysis based methods can find verified global solutions for the parameters of the potassium current. The drawback of this approach is that it is not as computationally fast as more traditional methods. While the computational time is within the lifetime (30 minutes) of a cell, it is desirable to reduce this time further in order to allow for more experiments to be conducted on the same cell. Additionally, there are several currents in the complete model. In order to reduce the computational time of the interval analysis method, we are investigating alternative preprocessing techniques as well as the use of parallel processing. Parallel processing can be used to speed-up computation by performing the parameter estimation in parallel and by performing the parameter estimation for different currents concurrently. The interval analysis method is particularly suitable for parallel processing having been shown to achieve superlinear speedup [11].

References

[1] Hodgkin, A.L. and Huxley, A.F. A quantitative description of membrane current and its application to conduction and excitation in nerve. *J. Physiol. Lond.*, 117:500–544, 1952.

[2] Schild, J.H., Clark, J.W., Hay, M., Mendelowitz, D., Andresen, M.C., and Kunze, D.L. A- and C-type rat nodose sensory neurons: Model interpretations ofM dynamic discharge characteristics. *Journal of Neurophysiology*, 71(6):2338–2358, June 1994.

[3] Bertil Hille. Ionic Channels of Excitable Membranes, Second Edition. *Sinauer Associates, Inc., Sunderland, MA*, 1992.

[4] Levenberg, K. A method for the solution of certain problems in least squares. *Quart. Appl. Math. 2*, pages 164–168, 1944.

[5] Marquardt, D. An algorithm for least-squares estimation of nonlinear parameters. *SIAM J. Appl. Math.*, 11:431–441, 1963.

[6] Mary Ann Branch and Andrew Grace. Matlab Optimization Toolbox: The User's Guide. *he Math Works, Inc., Natick, MA*, 1996.

[7] Hansen, E.R. lobal optimization using interval analysis. *Pure and Applied Mathematics*, page 165, 1992.

[8] Van Iwaarden, R.J. An improved unconstrained global optimization algorithm. *PhD Thesis, University of Colorado at Denver*, 1996.

[9] Holmqvist, K. and Migdalas, A. A C++ class library for interval arithmetic in global optimization. *State of the Art in Global Optimization*, pages 213–226, 1996.

[10] Kearfott, R.B. est results for an interval branch and bound algorithm for equality-constrained optimization. *State of the Art in Global Optimization*, pages 181–199, 1996.

[11] Moore, R., Hansen, E., and Leclerc, A. Rigorous methods for global optimization. *Recent Advances in Global Optimization*, 1992.

[12] Patil, R.B. Efficient Verified Global Optimization Using Interval Arithmetic. *New Mexico State University, Las Cruces, N.M.*, 1996.

[13] Kaisheng Du. Cluster problem in global optimization using interval arithmetic. *PhD Thesis, University of Southern Louisiana*, 1994.

[14] Burford, L.J. Interval analysis and applications. *MA Thesis, University of Texas at Austin*, 1968.

[15] Moore, R.E. Interval Analysis. *Prentice-Hall, Englewood Cliffs, N.J.*, 1996.

[16] Moore, R.E. Methods and Applications of Interval Analysis. *Society for Industrial and Applied Mathematics, Philadelphia, PA*, 1979.

[17] Epperly, T.G.W. Global Optimization of Nonconvex Nonlinear Programs Using Parallel Branch and Bound. *University of Wisconsin, Madison, WI*, 1995.

[18] Nickel, K. On the Newton Method in Interval Analysis. *The University of Wisconsin Mathematics Research Center, Madison, WI*, 1971.

[19] Hansen, E.R. and Greenberg, R.I. An interval newton method. *Appl. Math. Comp.*, 12:89–98, 1983.

[20] Hansen, E.R. Global optimization using interval analysis - the multi-dimensional case. *Numer. Math.*, 34:247–270, 1980.