

Parameter estimation of the Hodgkin–Huxley model using metaheuristics: application to neuromimetic analog integrated circuits

L. Buhry, S. Saïghi, A. Giremus, E. Grivel and S. Renaud

IMS Laboratory, UMR 5218 CNRS

University of Bordeaux – ENSEIRB

351, Cours de la Libération 33405 TALENCE – FRANCE

Email: laure.buhry, sylvain.saighi, audrey.giremus, eric.grivel, sylvie.renaud@ims-bordeaux.fr

Abstract—In 1952 Hodgkin and Huxley introduced the voltage-clamp technique to extract the parameters of the ionic channel model of a neuron. Although this method is widely used today, it has a lot of disadvantages. In this paper, we propose an alternative approach to the estimation method of the voltage-clamp technique using metaheuristics such as Simulated Annealing, Genetic Algorithms and Differential Evolution. This method avoids approximations of the original technique by simultaneously estimating all the parameters of a single ionic channel with a single fitness function. To compare the different methods, we apply them on measurements from a neuromimetic integrated circuit. This circuit, due to its analog behavior, provides us noisy data like a biological system. Therefore we can validate the efficiency of our method on experimental-like data.

I. INTRODUCTION

The voltage-clamp technique introduced by Hodgkin and Huxley in 1952 is today the most used technique to estimate the parameters of the ionic channel model of a neuron. However, this method has many disadvantages linked to approximations required [1]. In particular, it is difficult to precisely estimate the time constants of the model that appear in 2nd or 3rd degree-equations. It is also difficult to recover the parameters of the sodium channel because of its so-called inactivation term and its activation term that are largely dependent on one another; thus the separate estimation of these parameters can be made impossible. A simultaneous estimation would be more efficient. Because of the strongly non linear behavior of the equations, “classical” optimization methods like gradient descent are often inefficient. Studies have been carried out concerning the parameter estimation of Hodgkin–Huxley model (HH) from biological recordings. J.L. Madden et al. used the Levenberg–Marquart method of the Matlab toolbox [2] or use a gradient–descent [3]. However, the estimations obtained with those methods can be local extrema of the cost function.

In this paper, we propose an alternative method using metaheuristics despite their computational costs. Besides avoiding the approximations of the voltage-clamp technique, it makes it possible to jointly estimate all parameters of a single ionic channel by finding the minimum of a fitness function. The latter is usually defined as the difference between an

observed datum, e.g. membrane voltage, ionic currents, and its theoretical value calculated with the parameter vector.

Metaheuristics consist in a population evolution of individuals towards a parameter vector that minimizes a beforehand defined fitness function. Those optimization techniques are inspired by natural systems like metallurgy when dealing with the Simulated Annealing (SA), biology of evolution for Genetic Algorithms (GA) or ethology with Ant colony Algorithms or Particle Swarm Optimization, which are not treated here. GA have been used by G. Orchard et al. to extract parameters of synaptic models [4]. GA (technics which seem to be costly) [5], [6], and SA [6], [7], have already been used to construct software models of multi-compartmental neurons or Hodgkin–Huxley-type models from biological recordings of membrane voltage or ionic currents. Those methods have been tested with various fitness functions that depend generally on the membrane voltage or the spike frequency of the neuron. In these references, the parameter adjustment of the SA and GA is undetailed.

In opposition to previous works, the methods that we use allow one to simultaneously estimate all the parameters of a ionic channel without requiring the usual approximations of the voltage-clamp technique. In this paper, besides the GA and the SA, we propose to use a Differential Evolution (DE) method to estimate the 15 parameters of the HH model from experimental data. The parameters are estimated channel by channel. Our research group already designed a neuromimetic analog IC [1] which operates in real-time. Thus we decided to use recordings from this circuit to test our optimization methods. Indeed we can record ionic current and membrane voltage from the circuit outputs. Moreover analog circuits have non-deterministic behavior, due to the noise and the analog dispersion in the fabrication process. Because of the uncertainty remaining on the analog parameters stored (fabrication mismatch), we need to estimate the true parameters computed by the circuit. Therefore, we can validate the efficiency of our method on voltage-clamp-like measurements.

First, we present the model and the system that we use for analog simulations. Then, we briefly describe the three optimization methods implemented to estimate the model

parameters. Finally, we compare the results obtained with the three methods for each ionic channel.

II. MODEL AND SYSTEM

The HH model establishes an analogy between electronics and biology. This formalism relies on parameters with a biophysical significance realistic by the way of a conductance-based expression of the neural activity. The electrical activity of a neuron is the consequence of the diffusion of ionic species through its membrane. The HH formalism provides a set of equations and an equivalent electrical circuit that describe these phenomena. The current flowing across the membrane is integrated on the membrane capacitance, as follows:

$$C_{\text{mem}} \frac{dV_{\text{mem}}}{dt} = I_K + I_{Na} + I_{\text{leak}} + I_S \quad (1)$$

where V_{mem} denotes the membrane potential, C_{mem} the membrane capacitance, I_S an eventual stimulation or synaptic current and I_K , I_{Na} , I_{leak} the potassium, the sodium and the leak currents respectively. These latter satisfy the following equations:

$$I_K = g_K n^4 (V_{\text{mem}} - E_K) \quad (2)$$

$$I_{Na} = g_{Na} m^3 h (V_{\text{mem}} - E_{Na}) \quad (3)$$

$$I_{\text{leak}} = g_{\text{leak}} (V_{\text{mem}} - E_{\text{leak}}) \quad (4)$$

where $g_i, i=K, Na, \text{leak}$ is the maximal conductance value. $E_i, i=K, Na, \text{leak}$ is the ion-specific reversal potential, and n , m and h respectively represent the activation term of the potassium channel, the activation term and the inactivation term of the sodium channel. Those terms are dynamic functions describing the permeability of membrane channels to the ion considered. In addition, they all satisfy the following differential equation :

$$\tau_x \frac{dx}{dt} = x_{\infty} - x \quad \text{where } x = n, m, h. \quad (5)$$

τ_x denotes the time constant for the convergence. It should be noted that, when the time t increases to ∞ , x converges towards x_{∞} which is a sigmoid function of V_{mem} .

$$x_{\infty} = \frac{1}{1 + \exp\left(\frac{\mp(V_{\text{mem}} - V_{\text{offset}_x})}{V_{\text{slope}_x}}\right)} \quad (6)$$

where V_{offset_x} is the sigmoid offset and V_{slope_x} the sigmoid slope. The sign before $(V_{\text{mem}} - V_{\text{offset}_x})$ is $-$ for the activation term and $+$ for the inactivation.

Considering these equations ((1)–(6)), the 15 parameters stored in the vectors $X_{i, i=K, Na, \text{leak}}$ must be estimated to define the HH model:

$$X_K = [g_K \tau_n E_K V_{\text{offset}_n} V_{\text{slope}_n}], \quad X_{\text{leak}} = [g_{\text{leak}} E_{\text{leak}}],$$

$$X_{Na} = [g_{Na} \tau_m \tau_h E_{Na} V_{\text{offset}_m} V_{\text{offset}_h} V_{\text{slope}_m} V_{\text{slope}_h}].$$

The IC *Pamina* designed by our group [8] reproduces in real-time the electrical activity of a neuron following the HH formalism. A computer-based system was built to run experiments using the IC *Pamina*. Using a graphical interface,

the user defines each neuron model. The specifications include the choice of the ionic channels and the parameter values for each channel. We can record individually the ionic channel response by applying different step values on the membrane voltage. In this study, we apply the estimation methods to inhibitory neurons, also called ‘‘Fast Spiking’’ (FS) neurons. This type of neurons is modeled by only three conductances (potassium, sodium and leak).

III. OPTIMIZATION METHODS

The parameter estimation is accomplished channel by channel. The leak current I_{leak} given by the affine equation (4) does not bring into play the (in)activation term; g_{leak} and E_{leak} can be estimated thanks to a linear regression. The parameter estimation of potassium and sodium channels requires more complex techniques because of the strong non-linearities of its equations. Therefore we propose to use metaheuristics.

Figure 1 depicts the way the parameters are estimated. During 50ms (biological real-time, with a sampling period $\Delta t = 0.01ms$) we record independently the currents, I_{hardware} , for each ionic channel after applying successive steps values on the membrane voltage. Then we apply a discretization¹ to the HH model equations with the estimated parameters to obtain I_{software} . Thus we find a new value of the fitness function and a new parameter vector $X_{i, i=K, Na, \text{leak}}$.

The fitness function F_{fit} that we suggest minimizing is the same for each ionic channel and is defined by :

$$F_{\text{fit}} = \sum_t \sum_{\text{stim}} (10^6 \times (I_{i, \text{hardware}}(t, \text{stim}) - I_{i, \text{software}}(t, \text{stim}))^2) \quad (7)$$

where the subscript $i = K, Na$, t is the time and stim corresponds to the different values of stimulation. Note that we add a factor 10^6 to avoid numerical limitations.

¹approximation of partial differences by : $\tau_x \frac{x_n - x_{n-1}}{\Delta t} = x_{\infty} - x_n$

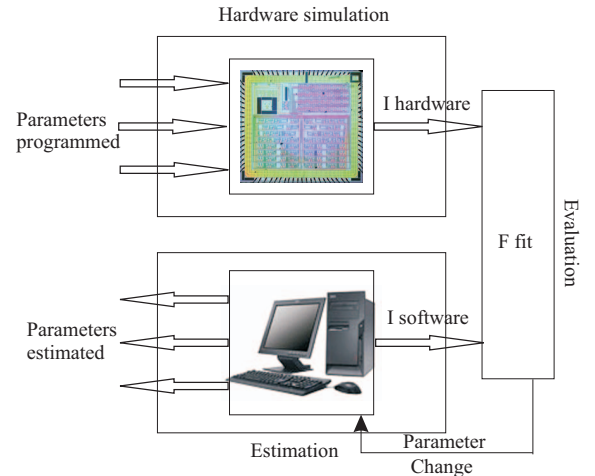


Fig. 1. Parameter estimation system.

A. Simulated Annealing

SA consists in evolving an individual (i.e. a vector storing the values of each parameter to estimated) by stage. N_{iter_stage} is the number of iterations on each stage. After the evolution, we accept consistently this individual if it minimizes the fitness function. On the contrary, if the value of F_{fit} increases, to avoid local minima, we accept the individual with a probability $exp(-D[F]_t/T_t)$ where $T_t = fact^t \times T_{init}$ is the temperature at time t , $fact < 1$ is the decrease factor of the temperature, T_{init} is the initial temperature and $D[F]_t = F_{fit}(X_t) - F_{fit}(X_{t-1})$.

B. Genetic Algorithms

For non-linear models, GAs appear to be a potentially useful approach, where locating the global optimum is a difficult task, although their computational cost is quite high, they are well suited to optimization problems, especially the search of extrema of a cost function, called fitness. GAs are based on the evolution of a population which contains a set of solutions, called individuals, encoding the values of the parameters to be optimized. The purpose of GAs is then to explore and to evaluate new regions of the solution space, by building new individuals from existing ones. In our case, the individuals are the model parameter vectors. GAs are hence based on the following four steps:

(i) Generation of the initial population of N_{ind} individuals. We denote by N_{renouv} the quantity of population replaced at each generation.

(ii) Definition of a fitness measure to rank the solutions, i.e. the set of estimated parameter values, inside the population.

(iii) Introduction of rules to generate new individuals using operators such as reproduction, crossover, mutation, etc. The crossover is made with a probability pc , and the mutation with a probability pm . Reproduction consists in copying the existing individual into the new population. When completing a crossover, two individuals of the former population referred as parents are used to create a new individual containing parent parts. Mutation consists in a random alteration of a given individual. This enables the GA to explore new regions of the solution space. However, these operators do not take into account any possible link between different solution parameters and crossover mixes these parameters in an arbitrary way. Convergence may therefore be slowed.

(iv) Definition of a criterion for stop condition and extraction of parameters defining the function.

C. Differential Evolution

As GA, DE belongs to the class of Evolutionary Algorithms. The main difference with GA lies in step (iii): indeed, for each individual of a population constituted by NP vectors, a new trial individual is built with a probability CR which is the constant of recombination. This new parameter vector is generated by adding to a randomly-chosen population member X_k^{r1} the weighted difference vector between two other population members X_k^{r2} and X_k^{r3} (eq. 8). One can also modify the population member by a second weighted

difference which takes into account the best parameter vector, X^{min} (eq. 9). Indeed, one has :

$$X_{j,k+1} = X_k^{r1} + F.(X_k^{r2} - X_k^{r3}) \quad (8)$$

$X_{j,k+1}$ is the vector j of the $k+1^{th}$ generation and F the constant of differenciation.

$$X_{j,k+1} = X_k^{r1} + F.(X_k^{r1} - X_k^{min}) + F.(X_k^{r2} - X_k^{r3}) \quad (9)$$

Note that N_{iter} , that appears in TAB.I, is the number of iterations and that we use same the notation as in [9].

IV. RESULTS–DISCUSSION

We implemented the algorithms described above with Matlab software. To compare the methods, we recorded the evolution and the final value of the fitness function. To validate the optimization process, we present also the curves obtained by the hardware measurements and software computation.

A. Comparison of the methods on potassium channel

We implemented a DE method and compared it to SA and GA for parameter estimation of potassium channel.

The parameters of the algorithms are summarized in TAB.I. Values estimated for the potassium channel, the final value of fitness function, f_{opt} , and simulation duration are found in TAB.II.

1) *Simulated Annealing*: For the Matlab simulation, parameters extracted using SA have been “re-injected” in HH equations. The final value of the fitness function is four times higher than the value obtained with the DE method.

2) *Genetic Algorithm*: The final value of the fitness function is roughly ten times greater than the value obtained with the two other methods; thus we do not show the current curves. The curves are not overlaid for each stimulation value, which shows that the estimated parameters are not the exact values computed by the IC.

TABLE I
ALGORITHM PARAMETERS.

	SA	GA	DE
Parameters	$N_{iter_stage} = 100$ $T_{init} = 10^3$ $T_{fin} = 10^{-9}$ $fact = 1 - 10^{-9}$	$N_{iter} = 300$ $N_{ind} = 300$ $N_{renouv} = 1/4$ $pm = 0, 1$ $pc = 0, 7$	$N_{iter} = 300$ $NP = 300$ $F = 0, 5$ $CR = 0, 9$

TABLE II
PARAMETERS ESTIMATED FROM POTASSIUM CHANNEL.

	SA	GA	DE
$g_K (S)$	$1, 69.10^{-5}$	$1, 33.10^{-5}$	$1, 66.10^{-5}$
$\tau_n (s)$	$4, 10.10^{-3}$	$3, 64.10^{-3}$	$3, 96.10^3$
$E_K (V)$	$-4, 49.10^{-1}$	$-6, 58.10^{-1}$	$-4, 46.10^{-1}$
$V_{offset} (V)$	$-1, 73.10^{-1}$	$-2, 18.10^{-1}$	$-1, 53.10^{-1}$
$V_{slope} (V)$	$5, 18.10^{-2}$	$9, 04.10^{-2}$	$4, 11.10^{-2}$
Final error	$f_{opt} = 5, 83.10^2$	$f_{opt} = 1, 95.10^3$	$f_{opt} = 1, 24.10^2$
Duration	6 days and 6h	1h52min	57min

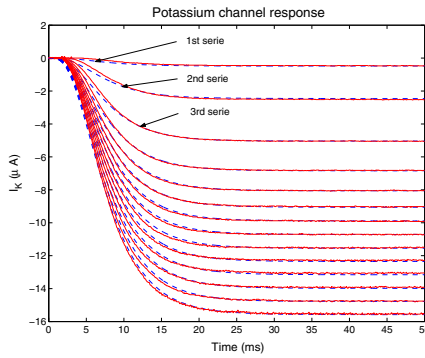


Fig. 2. Potassium channel Response – DE. Continuous line : potassium channel hardware response to different steps of stimulations. Dashed line : software response obtained with extracted parameters.

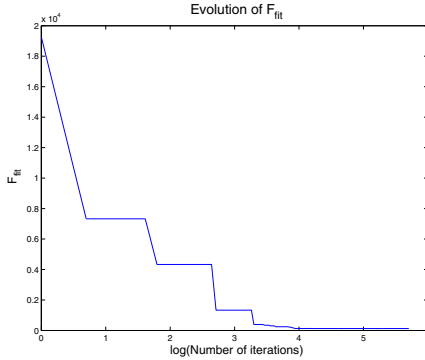


Fig. 3. Potassium channel Response – DE. $\log(\text{number of iterations})$ represents the neperian logarithm of the number of iterations.

3) *Differential Evolution*: The (hardware and software) responses of potassium channel and the evolution of the fitness function are shown in figures 2 and 3. We can see that the curves are almost overlaid for each stimulation current. The small discrepancy during the 15 first milliseconds is due to chip defects. The power raising in (2) is done by multipliers using the translinear loop. These multipliers are not really linear with weak currents [1]. Moreover, when performing random multiple starts of this algorithm in the same conditions, we obtained the same extracted parameters with a precision of six significant digits.

We conclude that the DE is the best technique to fit the curves obtained with the circuit: this method gives the minimal value of the fitness function. We also notice that the DE convergence is faster than the other methods.

B. Parameter estimation of sodium channel using DE

On figure 4, we observe the sodium channel response recorded from the IC and obtained by the software simulation with the DE method. The simulation has been run with $N_{iter} = NP = 400$. In order to see a convergence phenomenon, because of the number of parameters that we had to estimate, we had to increase the number of iterations in comparison to the potassium. Thus the effective simulation duration also increases : here, it ran 6h50min.

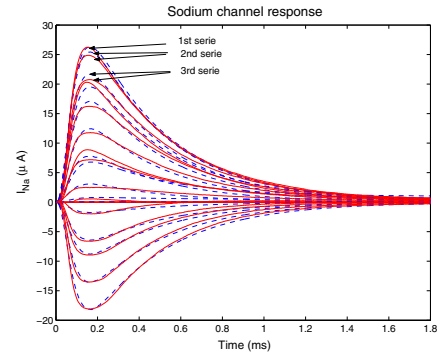


Fig. 4. Sodium channel response – DE method. Continuous line : sodium channel hardware response to different steps of stimulations. Dashed line : software response obtained with extracted parameters.

V. CONCLUSION

We presented an estimation technique adapted to conductance-based model of HH type. This method is an alternative to the estimation method of the voltage-clamp technique, but the measurement technique is the same as in voltage-clamp. Here, it is applied to artificial neurons implemented on ICs, but it could as well be used on biological cells with intra-cellular measurements. The results improve the method validness as well as the good performance of the DE algorithm (convergence quickness). The system developed will be used to estimate the parameters of more complex neuron models. It will be at the root of an automatic tuning setup of neuromimetic ICs, as computation core for a real time simulator of cortical neural networks.

REFERENCES

- [1] S. Saïghi, L. Buhry, Y. Bornat, G. N'Kaoua, J. Tomas, and S. Renaud, "Adjusting the neuron models in neuromimetic ICs using the voltage-clamp technique," in *Proceedings of ISCAS08*. IEEE, May 2008, pp. 1564–1567.
- [2] J. L. Madden, Z. B. Miled, R. C. Y. Chin, and J. Schild, "On parameter estimation for neuron models," in *BIBE '00: Proceedings of the 1st IEEE International Symposium on Bioinformatics and Biomedical Engineering*. IEEE Computer Society, 2000, p. 253.
- [3] S. Doi, Y. Onoda, and S. Kumagai, "Parameter estimation of various hodgkin-huxley-type neuronal models using a gradient-descent learning method," in *SICE 2002 : Proceedings of the 41st SICE Annual Conference*. IEEE Computer Society, August 2002, pp. 1685–1688.
- [4] G. Orchard, A. Russel, K. Mazurek, F. Tenore, and R. Etienne-Cummings, "Configuring silicon neural networks using genetic algorithms," in *Proceedings of ISCAS08*, May 2008, pp. 1048–1051.
- [5] N. Keren, N. Peled, and A. Korngreen, "Constraining compartmental models using multiple voltage-recordings and genetic algorithms," *Journal of Neurophysiol*, vol. 94, no. 4598, pp. 3730–3742, 2005.
- [6] M. C. Vanier and J. Bower, "A comparative survey of automated parameter-search methods for compartmental neural models," *Journal of Computational Neuroscience*, vol. 7, pp. 149–171(23), 10 September 1999.
- [7] M. Gurkiewicz and A. Korngreen, "A numerical approach to ion channel modelling using whole-cell voltage-clamp recordings and genetic algorithm," *PLoS Comput Biol*, vol. 3, no. 8, p. 169, August 2007.
- [8] S. Saïghi, J. Tomas, Y. Bornat, and S. Renaud, "A conductance-based silicon neuron with dynamically tunable model parameters," in *Proceedings of CNE05*. IEEE, May 2005, pp. 285–288.
- [9] R. Storn and K. Price, "Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 11, pp. 341–359, 1997.