A parameter optimization method for Digital Spiking Silicon Neuron model

Takuya Nanami
Institute of Industrial Science, University of Tokyo
Tokyo, Japan

Filippo Grassia
LTI Lab., University of Picardie Jules Verne
Saint-Quentin, France

Takashi Kohno
Institute of Industrial Science, University of Tokyo
Tokyo, Japan
nanami@sat.t.u-tokyo.ac.jp, kohno@sat.t.u-tokyo.ac.jp

Abstract

DSSN model is a qualitative neuronal model designed for efficient implementation in a digital arithmetic circuit. In our previous studies, we extended this model to support a wide variety of neuronal classes. Parameters of the DSSN model were hand-fitted to reproduce neuronal activity precisely. In this work, we studied automatic parameter fitting procedure for the DSSN model. We optimized parameters of the model by the differential evolution algorithm in order to reproduce waveforms of the ionic-conductance models and reduce necessary circuit resources for the implementation.

Keywords: Silicon neuronal network, Spiking neuron model, Differential evolution, FPGA

1. Introduction

A wide variety of neuronal models have been used in silicon neuronal networks because of the trade-off between reproducibility of neuronal activities and computational efficiency. For example, ionic-conductance models can reproduce a neuronal activity accurately but demands enormous computational resources in large-scale implementations. In contrast, integrate-and-fire based models such as the LIF and Izhikevich models [1] can be implemented with less resource, because they approximate a spiking process by the resetting of the state variables. However, they have reduced reproducibility of complex neuronal activities. For example, these models assume fixed maximum membrane potentials during the spike process, whereas the spike intensity is nonuniform in the nervous system [2]. The DSSN model [3] is a qualitative neuronal model designed for efficient implementation in a digital arithmetic circuit. In our previous studies [4][5], we extended the DSSN models to support various neuronal activities; regular spiking, fast spiking, intrinsically bursting, low-threshold spiking, elliptic bursting, and parabolic bursting.

To reproduce a variety of spiking properties in neurons, appropriate parameter sets for neuronal models have to be found. The relationship between the values of the parameters and the behavior of the model is generally complex, and it is difficult and time-consuming to find an appropriate parameter set. Automatic parameter fitting procedures have been studied as a solution to this problem. Parameters for Hodgkin-Huxley-type models have been tuned automatically in [6] and [7]. Pospischil et al [8] reproduced regular and fast spiking neuron classes by simulated annealing based optimization method using a cost function composed of the firing rate and the adaptation time. Buhry et al. [9][10] compared several heuristic algorithms for automatic parameter estimation on a Hodgkin-Huxley-type model, where the differential evolution algorithm had a best performance. They also applied the algorithm to a neuromimetic analog integrated circuit.

In this work, we applied the differential evolution algorithm to parameter fitting of the DSSN model in regular and fast spiking settings, because it shares its mathematical structures with Hodgkin-Huxley-type models. Regular-spiking (RS) is a most typical classes of cortical neurons and is characterized by spike-frequency adaptation; the spike frequency decreases over time in response to a constant stimulus input. Conversely, Fast-spiking (FS) neurons maintain firing at a constant frequency. Parameters were optimized to reproduce waveforms of the ionic-conductance models in [8] and reduce the circuit resource requirements for implementation.

The remainder of this paper is organized as follows. Section 2 introduces our neuron model, differential evolutional algorithm, details of our parameter optimization procedure, and GPU-based implementation. The result is shown in Section 3. Section 4 summarizes the work and suggests ideas for future.

2. Method

2.1. *Digital Spiking Silicon Neuron model*

The 3-variable DSSN model is a qualitative neuron model that can simulate several classes of neuronal activities by Euler's method with fixed point operation including RS and FS [4]. Equations for regular and fast spiking are given by

$$\frac{dv}{dt} = \frac{\emptyset}{\tau}(f(v) - n - q + I_0 + I_{stim}),$$

$$\frac{dn}{dt} = \frac{1}{\tau}(g(v) - n),$$

$$\frac{dq}{dt} = \frac{\varepsilon}{\tau}(h(v) - q),$$

$$f(v) \equiv \begin{cases} a_{fn}(v - b_{fn})^2 + c_{fn} & (v < 0) \\ a_{fp}(v - b_{fp})^2 + c_{fp} & (v \geq 0), \end{cases}$$

$$g(v) \equiv \begin{cases} a_{gn}(v - b_{gn})^2 + \mathrm{c_{gn}} & (v < r_g) \\ a_{gp}(v - b_{gp})^2 + c_{gp} & (v \geq r_g), \end{cases}$$

$$h(v) \equiv \begin{cases} a_{hn}(v - b_{hn})^2 + \mathrm{c_{hn}} & (v < r_h) \\ a_{hp}(v - b_{hp})^2 + c_{hp} & (v \geq r_h), \end{cases}$$

where v corresponds to the membrane potential, and n and q are the fast and slow variables, respectively, that abstractly describe the activity of the ion channels. Parameter I_0 is a bias constant and I_{stim} represents the input stimulus. Parameters ∅, ε, and τ control the time constants of the variables. Parameters, a_x, b_x, and c_x, where x is $fn, fp, gn, gp, hn,$ or hp, are constants that adjust the nullclines of the variables. All of the variables and constants in this qualitative model are purely with no physical units. A cubic shaped v-nullcline is a key to replicate the spiking dynamics in qualitative models. Because multipliers are resource-consuming in a digital arithmetic circuit, the DSSN model adopts a piecewise quadratic function for the v -nullcline so that its numerical integration step requires only one multiplication between variables. The architecture of the DSSN circuit was explained in our previous study [11].

2.2. *Differential evolution algorithm*

Differential evolution (DE) algorithm [12] is a popular heuristic method to solve optimization problems using a real number function. It is characterized by its mutation process that uses geometrical location of whole population. In the following, we explain each step in the DE algorithm. We define $x_k^i(j)$ as the gene j of the i th individual of the kth generation.

- **Initialization**

We generate M-dimensional parameter vectors x_0^i $(i = 0, \ldots, \mathrm{N} - 1)$, where N and M are the number of the individuals and parameters of the DSSN model. Parameter vectors are chosen randomly within the boundary constraints of the DSSN model.

- **Mutation**

We generate new parameter vector x_{mut}^i as follows:

$$\forall i = 1, \ldots, \mathrm{N}, \qquad x_{mut}^i = x_k^{r1} + F \cdot (x_k^{r2} - x_k^{r3}),$$

where r1, r2, and r3 are integer randomly chosen from $[0, \mathrm{N} - 1]$. $F$is usually set to 0.5.

- **Crossover**

We generate new parameter vectors x_{cross}^i that inherit genes of x_{mut}^i with a probability CR.

$$\forall i = 1, \ldots, \mathrm{N}, \qquad \forall j = 1, \ldots, \mathrm{M},$$

$$x_{cross}^i(j) = \begin{cases} x_{mut}^i(j) & if(rand \leq CR) \\ x_k^i(j) & otherwise, \end{cases}$$

where CR is usually set to 0.5.

- **Selection**

A selection is conducted by comparing the cost function values of x_k^i and x_{mut}^i, respectively, as follows:

$$\forall i = 1, \ldots, \mathrm{N},$$

$$x_{k+1}^i = \begin{cases} x_{cross}^i(j) & if\ f(x_{cross}^i) \leq f(x_k^i) \\ x_k^i & otherwise, \end{cases}$$

where $f(x)$ returns the value of the cost for parameter vector x.

A set of procedures including mutation, crossover, and selection is repeated for 10000 times, and the parameter vector that produces the lowest value in the cost function is finally adopted.

2.3. *Parameter optimization procedure*

The DSSN model for RS and FS classes has 24 parameters, exploring whose spanning space is not realistic with average computing power available to those who are not specialized in high performance computing (HPC). To reproduce the dimension of the parameter space to be explored by the DE algorithm, we first explored the parameters that control the dynamics of the fast state variables v and n to fit the spike generation mechanism. Then, the parameters related to the slow state variable q were determined to fit the adaptation of the spike-frequency.

In the DSSN model's equation, some parameters are a coefficient of a variable. In its circuit implementation, a multiplication between a coefficient and a state variable was realized by shifters and adders, and we need a larger number of couples of an adder and a shifter as the number of digits with value 1 in the fixed point representation of the coefficient increases. Parameters were optimized for not only reproducing waveforms of the ionic-conductance model but also reducing the circuit size.

In the procedure to determine the parameters that control the dynamics of the two fast variables, we fixed v-nullcline to facilitate parameter exploration while considering the dynamics of the spike generation process. It does not severely restrict the dynamical property of the model, because the relation between the v- and n- nullclines is one of the major factors that rule the dynamical structures in our neuronal model. We defined the cost function as follows:

$$f_1(x) = f_{\mathrm{w}}(x) + k_1 f_{\mathrm{b1}}(x),$$

where $f_{\mathrm{w}}(x)$ is a square error between the waveforms of the DSSN model with parameter vector x and a target waveform. The target waveform was calculated by a fast subsystem of the DSSN model in our previous study [4], which is tuned to reproduce mathematical structure of the ionic conductance model in [8]. Note that minimum and maximum values of waveforms are normalized to 0 and 1 for comparison. Function f_{b1} is a number of digits with value 1 in the binary fixed-point expression of the coefficients in x. Constant k_1 is to balance this couple of cost functions. Parameters that control the dynamics of the fast variables were determined to minimize the cost function f_1 by the DE algorithm.

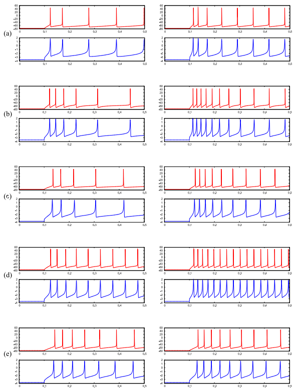

Figure 1: Waveforms of the ionic-conductance model (red) and the DSSN model (blue) in response to a weak (left) and strong (right) step stimulus input. (a)-(c) corresponds to regular spiking neuron cells. (d) and (e) corresponds to fast spiking neuron cells.

The remaining parameters related to slow variable q were determined to reproduce the spike frequency adaptation. The cost function is

$$f_2(x) = f_{\mathrm{s}}(x) + k_2 f_{\mathrm{b2}}(x),$$

where $f_{\mathrm{s}}(x)$ is a square error between spike timings of the DSSN model with a parameter vector x and that of the target ionic-conductance model in response to a step stimulus. Function f_{b2} is a number of digits with value 1 in the binary fixed-point expression of the coefficients in x, and k_2 is a constant to balance this couple of cost functions.

The DE algorithm is known to be compatible with parallel computation schemes because of its intrinsic parallel structure. We accelerated the calculation by the Graphic processing unit (GPU) to obtain better parameter solution by executing a larger number of DE steps in a time period. The calculation of the cost function, in which the differential equations are solved by the Euler's method for thousands of times, was parallelly processed on a GPU. GPU (NVIDIA Tesla K40) –based implementation for calculating cost functions of 2^{14} parameter vectors consumed 1.578[s], and it was about six times faster than a single-thread program on Xeon E5-2670 CPU, which consumed 9.51[s].

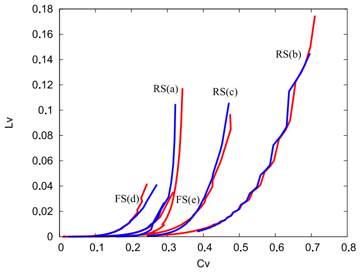

Figure 2: $C_V - L_V$ characteristics of the ionic-conductance model (red) and the DSSN model (blue).

Table. 1 : Device Utilization. The labels (a)-(e) correspond to those in Fig.1.

Name	(a)	(b)	(c)	(d)	(e)	(f)	(g)
FF	109	109	109	109	109	107	107
LUTs	697	660	624	681	631	1410	1357
DSPs	1	1	1	1	1	1	1

3. Result

We found five parameter sets with which the DSSN model reproduces the activities of the ionic-conductance models for three RS and two FS cells whose characteristics are slightly different from each other (Fig. 1).

We evaluated the similarity of the spiking patterns between the ionic-conductance model and the DSSN model by calculating C_V and L_v [13] that are statistics for the spike sequence while changing the stimulus intensity (Fig. 2). They are defined as follows:

$$C_V = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(T-\bar{T})^2/\bar{T}}$$

$$L_V = \frac{1}{n-1}\sum_{i=1}^{n-1}\frac{3(T_i - T_{i+1})^2}{(T_i + T_{i+1})^2}$$

where T_i denotes the ith interspike interval, $\bar{T}_i$ is the average of T_i, and n is the number of spikes in the sequence. Both models' simulation results draw similar curves on the $C_V - L_V$ plane for each cell. The similarity of these characteristics will be estimated based on the requirements specified by the circuits' applications.. The results in Figs. 1 and 2 were obtained by numerical integration with the Euler's method (dt $= 2^{-13}$ s) where each variable was expressed by 18-bit fixed point with 14-bit fraction part. The same results are expected to be observed in the following FPGA implementation, because this condition is the same as our VHDL codes.

We compiled the DSSN models for Virtex-7 XC7VX690T FPGA using Xilinx Vivado Design Suite. Device utilization is listed in Table 1. In the column labeled (f) and (g), the resource usage for RS and FS settings in [4] is shown. The requirement for LUTs was reduced down to less than half in all the settings.

4. Conclusion

In this work, we developed a parameter optimization method for the DSSN model on the basis of the DE algorithm. By splitting the optimization process in 24-dimentional parameter space into two steps, we could find parameter sets with which the DSSN model reproduces the characteristics activities of RS and FS cells without using special HPC systems. Cost functions f_{bx} $(x = 1, 2)$ were introduced so that the circuit size is reduced. We also confirmed qualitative similarity between the ionic-conductance model and the DSSN model by measuring statistics for the spike timing. In our future work, we will improve this optimization method to cover above other neuron classes.

Acknowledgement

This work was partially supported by JSPS SAKURA Program and JST PRESTO and CREST.

References

1. Eugene M. Izhikevich, Simple model of spiking neurons, *IEEE Transactions on neural networks*, 14(6), pp. 1569-1572, 2003
2. Andrew A.V. et al., A Model of a Segmental Oscillator in the Leech Heartbeat Neuronal Network., *Journal of Computational Neuroscience,* Vol. 10, pp. 281-302, 2001.
3. T. Kohno and K. Aihara. Digital spiking silicon neuron: Concept and behaviors in gj-coupled network. *Proceedings of International Symposium on Artificial Life and Robotics* 2007, 2007.
4. T. Nanami and T. Kohno. Simple cortical and thalamic neuron models for digital arithmetic circuit implementation, *Frontiers in Neuroscience*, Vol. 10, No.181, pp. 1—12, 2016
5. T. Nanami, K. Aihara, and T. Kohno, Elliptic and parabolic bursting in a digital silicon neuron model, *2016 International Symposium on NOLTA*, Japan, November 27th–30th, 2016
6. Tawfik B, Durand DM, Nonlinear parameter-estimation by linear association : Application to a 5-parameter

passive neuron model. *IEEE Trans Biomed Eng* 41:461–469, 1994.

7. Haufler D, Morinc F, Lacaille JC, Skinner FK, Parameter estimation in single-compartment neuron models using asynchronization-based method. *Neurocomputing* 70:1605–1610, 2007.
8. M. Pospischil et al.. Minimal Hodgkin-Huxley type models for different classes of cortical and thalamic neurons. *Biological Cybernetics*, 99(4-5):427–441, 2008.
9. Buhry, L., Sa¨ıghi, S., Giremus, A., Grivel, E., and Renaud, S. (2008). Parameter estimation of the Hodgkin–Huxley model using metaheuristics: Application to neuromimetic analog integrated circuits. *In Proceedings of IEEE International Conference on Biomedical Circuits and Systems* (pp. 173–176). Piscataway, NJ: IEEE.
10. L. Buhry et al., Automated parameter estimation of the Hodgkin-Huxley model using the differential evolution algorithm: application to neuromimetic analog integrated circuits, *Neural computation* 23 (10), 2599-2625, 2011.
11. Takuya Nanami and Takashi Kohno, "An FPGA-based cortical and thalamic silicon neuronal network," *Journal of Robotics, Networking and Artificial Life*, Vol. 2, No. 4, pp. 238-242, Mar., 2016.
12. R. Storn, K. Price, A Simple and Efficient Heuristic for global Optimization over Continuous Spaces, *Journal of Global Optimization*, Vol. 11, pp. 341-359, 1997.
13. Shinomoto, S., Shima, K., and Tanji, J. Differences in spiking patterns among cortical neurons. *Neural Comput.* 15, 2823–2842. 2003