ReScience

[Re] A Generalized Linear Integrate-and-Fire Neural Model Produces Diverse Spiking Behaviors

Georgios Detorakis[1]

1 Department of Cognitive Sciences, UC Irvine, Irvine, CA, USA

gdetorak@uci.edu
gdetor@protonmail.com

Editor
Name Surname

Reviewers
Name Surname
Name Surname

Received Sep, 1, 2015
Accepted Sep, 1, 2015
Published Sep, 1, 2015

Competing Interests:
The author has declared that no competing interests exist.

Article repository

Code repository

A reference implementation of

→ A Generalized Linear Integrate-and-Fire Neural Model Produces Diverse Spiking Behaviors, Stefan Mihalas and Ernst Niebur, Neural Computation 21, 704–718, 2009.

Introduction

Integrate-and-fire neurons are being used extensively in the field of neuroscience for modeling spiking behaviors [1]. In this work we provide a reference implementation of [3], where the authors have introduced a generalization of the leaky integrate-and-fire neuron model. The Mihalas-Niebur Neuron (MNN) model is a linear integrate-and-fire neuron model capable of expressing a rich spiking behavior based on a set of parameters.

An MNN model expresses tonic and phasic spiking, class 1 and 2, spike frequency adaptation, accommodation, threshold variability, rebound spike, integrator, input bistability, hyperpolarizing spiking and bursting, tonic, phasic and rebound bursting, mixed mode, afterpotentials, basal bistability, preferred frequency and spike latency. Due to its simplicity, the MNN model has been used in neuromorphic implementations such as [2].

The model consists of linear differential equations, which describe the membrane and threshold potentials and internal currents. All the results provided in [3] have been obtained by using only two internal currents and thus we use the exact same number of internal currents in this work. The differential equations describe the subthreshold dynamics, while the instantaneous threshold potential controls when the neuron firing an action potential (spike) in a dynamic way. The ability of the MNN model to generate such a diverse spiking behavior is due to the complex update rules. In this work the MNN model has been implemented in Python (version 3.6.1) using Numpy (version 1.13.1) and Matplotlib (version 2.0.2) packages.

Methods

In order to implement the model described in [3], we discretized the dynamical system using the Euler's integration scheme. The time step is fixed to 0.1 ms for all the simulations, and the simulation time varies according to figure 1 of the original paper. We provide all equations and parameters of the model in tables as it has been suggested by [4]. Table 1 provides the summary of the model. Tables 2 and 3 give the subthreshold dynamics (differential equations) describing the membrane and the threshold potentials as well as the two internal currents and the update rules. The parameters for all the simulations are given in table 4. Table 5 provides the external current intensities (and pulse durations), and initial conditions are described in table 6.

Model Summary	
Populations	No population – single neuron model
Topology	–
Connectivity	–
Neuron Model	Linear Integrate-and-Fire Neuron
Channel Models	Linear, first order ODEs
Synapse Model	–
Plasticity	–
Input	Constant current or rectangular pulses
Measurements	Membrane potential, phase plane

Table 1: Summary of the model

Neuron Model	
Name	Mihalas-Niebur Neuron (MNN)
Type	Linear Leaky Integrate-and-Fire Neuron
Membrane Potential	$\frac{\mathrm{d}V(t)}{\mathrm{d}t} = \frac{1}{C}\Big(I_e + I_1 + I_2 - G(V(t) - E_L)\Big)$
Instantaneous Threshold Potential	$\frac{\mathrm{d}\Theta(t)}{\mathrm{d}t} = a(V(t) - E_L) - b(\Theta(t) - \Theta_\infty)$
Internal Currents	$\frac{\mathrm{d}I_1(t)}{\mathrm{d}t} = -k_1 I_1(t)$ $\frac{\mathrm{d}I_2(t)}{\mathrm{d}t} = -k_2 I_2(t)$

Table 2: Description of the subthreshold dynamics of Mihalas-Niebur neuron model. $V(t)$ and $\Theta(t)$ are the membrane and threshold potentials, respectively. E_L and Θ_∞ are the reversal potentials for the membrane and the threshold variables, respectively. a, b, k_1, k_2 and G are constant parameters. I_e is the external current applied on the neuron model.

Update rules	
Variable	Rule
$V(t)$	V_r
$\Theta(t)$	$\max\{\Theta_r, \Theta(t)\}$
$I_1(t)$	$R_1 \times I_1(t) + A_1$
$I_2(t)$	$R_2 \times I_2(t) + A_2$

Table 3: Update rules. V_r and Θ_r are the reset values for the membrane and threshold potentials, respectively. R_1, R_2, A_1 and A_2 are constants.

All simulations ran on a Dell OptiPlex 7040, equipped with a sixth generation i7 processor, 16 GB of physical memory and running Arch Linux. The total execution time of all simulations was 2.41 seconds and the peak consumed memory was 162 MB[1].

[1] Python memory profiler used (https://pypi.python.org/pypi/memory_profiler).

Model Parameters				
Figure	a (s^{-1})	A_1/C (V/s)	A_2/C (V/s)	t_f (s)
1A	0	0	0	0.2
1B	0	0	0	0.5
1C	5	0	0	0.2
1D	5	0	0	0.5
1E	5	0	0	1.0
1F	5	0	0	0.4
1G	5	0	0	1.0
1H	5	0	0	0.3
1I	5	0	0	0.4
1J	5	0	0	1.0
1K	30	0	0	0.4
1L	30	10	-0.6	0.4
1M	5	10	-0.6	0.5
1N	5	10	-0.6	0.5
1O	5	10	-0.6	1.0
1P	5	5	-0.3	0.5
1Q	5	5	-0.3	0.2
1R	0	8	-0.1	0.2
1S	5	-3	0.5	0.8
1T	-80	0	0	0.05
Common Parameters				
$b = 10\,s^{-1}$, $G/C = 50\,s^{-1}$, $k_1 = 200\,s^{-1}$, $k_2 = 20\,s^{-1}$, $\Theta_\infty = -0.05\,V$, $R_1 = 0$, $R_2 = 1$, $E_l = -0.07\,V$, $V_r = -0.07\,V$, $\Theta_r = -0.06\,V$.				

Table 4: Simulation Parameters

Model Parameters		
Figure	Type	I_e/C(V/s)
1A		1.5
1B		$1 + 10^{-6}$
1C		2
1D		1.5
1E		1.5(0.1s), 0(0.5s), 0.5(0.1s), 1(0.1s), 1.5(0.1s), 0(0.1s)
1F		1.5(0.02s), 0(0.18s), −1.5(0.025s), 0(0.025s), 1.5(0.025s), 0(0.125s)
1G		0(0.05s), −3.5(0.756s), 0(0.194s)
1H		$2(1 + 10^{-6})$
1I		1.5(0.02s), 0(0.01s), 1.5(0.02s), 0(0.25s), 1.5(0.02s), 0(0.02s) 1.5(0.02s), 0(0.04s)
1J		1.5(0.1s), 1.7(0.4s), 1.5(0.1s), 1.7(0.4s)
1K		−1
1L		−1
1M		2
1N		1.5
1O		0(0.1s), −3.5(0.5s), 0(0.4s)
1P		2
1Q		2(0.015s), 0(0.185s)
1R		5(0.01s), 0(0.09s), 5(0.01s), 0(0.09s)
1S		5(0.005s), 0(0.005s), 4(0.005s), 0(0.385s), 5(0.005s), 0(0.045s) 4(0.005s), 0(0.345s)
1T		8(0.002s), 0(0.048s)

Table 5: External current. This table provides the external current for each panel in Figure 1. There are two types of external currents, constants and pulses. In the case of pulses the duration of each pulse is given in seconds along with its intensity.

Initial Conditions	
Variable	Initial Value
$V(t)$	−0.07 V / −0.03 V (Figure 1H)
$\Theta(t)$	−0.05 V / −0.03 V (Figure 1H)
$I_1(t)$	0.01 V
$I_2(t)$	0.001 V

Table 6: Initial conditions. In all simulations have been used the same initial conditions, except from the one illustrated in figure 1H.

Results

All three figures from the original article have been successfully replicated. All the different spiking behaviors of the model are illustrated in Figure 1, where the black solid line indicates the membrane potential ($V(t)$), the red dashed line illustrates the instantaneous threshold potentials ($\Theta(t)$), and the gray line shows the input to the neuron (I_e/C). Figures 2 and 3 depict the phase space of the phasic spiking ($V(t)$ and $\Theta(t)$) and phasic bursting ($V(t)$, $I_1(t)$, and $I_2(t)$). All the figures express the same qualitative behavior as the original figures in [3].

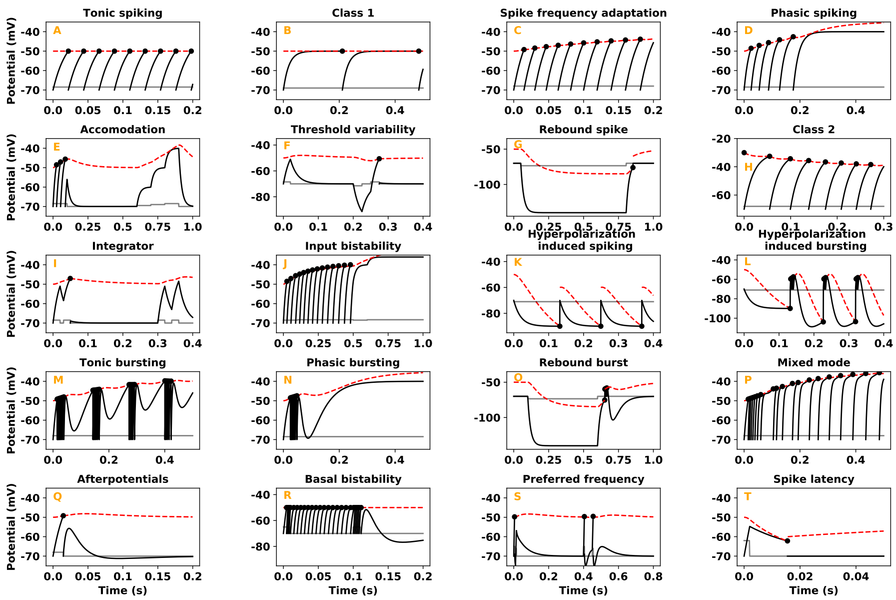

Figure 1: Neural responses of MNN. Black solid lines indicate the membrane potential ($V(t)$), the red dashed lines show the threshold potentials ($\Theta(t)$), and the gray lines the external currents applied on each case. **A** tonic spiking, **B** class 1, **C** spike frequency adaptation, **D** phasic spiking, **E** accommodation, **F** threshold variability, **G** rebound spike, **H** class 2, **I** integrator, **J** input bistability, **K** hyperpolarization induced spiking, **L** hyperpolarization induced bursting, **M** tonic bursting, **N** phasic bursting, **O** rebound burst, **P** mixed mode, **Q** afterpotentials, **R** basal bistability, **S** preferred frequency, **T** spike latency.

Conclusion

All figures in Mihalaş and Niebur [3] have been successfully replicated with high fidelity. Overall, the whole reproducing process was smooth and without obscure points since most of the parameters are provided in the original article. Only the external current time intervals and the initial conditions are not provided explicitly and thus we had to extract that information from figure 1 of the original article. To conclude, the article [3] has been successfully reproduced without any discrepancy.

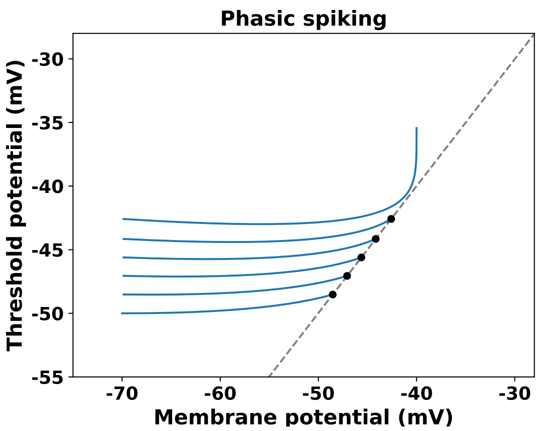

Figure 2: Phase space of phasic spiking. Blue solid lines indicate the trajectories of the model in the phase spiking behavior (Figure 1D). The dashed line corresponds to $V(t) = \Theta(t)$, and the black dots represent spike events. The parameters for this simulation are the same as in Figure 1D.

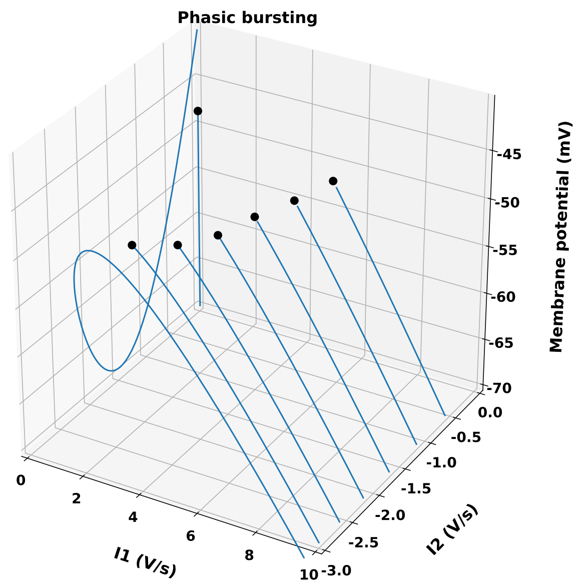

Figure 3: Phase space of phasic bursting. Blue solid lines represent the trajectories of the system and the black dots indicate spiking events. The parameters for this simulation are the same as in Figure 1N.

References

[1] Peter Dayan and Laurence F Abbott. *Theoretical neuroscience*. Vol. 806. Cambridge, MA: MIT Press, 2001.

[2] Fopefolu Folowosele, Tara Julia Hamilton, and Ralph Etienne-Cummings. "Silicon Modeling of the Mihalaș–Niebur Neuron". In: *IEEE transactions on neural networks* 22.12 (2011), pp. 1915–1927.

[3] Ștefan Mihalaș and Ernst Niebur. "A generalized linear integrate-and-fire neural model produces diverse spiking behaviors". In: *Neural computation* 21.3 (2009), pp. 704–718.

[4] Eilen Nordlie, Marc-Oliver Gewaltig, and Hans Ekkehard Plesser. "Towards reproducible descriptions of neuronal network models". In: *PLoS computational biology* 5.8 (2009), e1000456.