THE STOCHASTIC, MARKOVIAN, HODGKIN-HUXLEY TYPE OF MATHEMATICAL MODEL OF THE NEURON

A. Świetlicka*, K. Gugała*, A. Jurkowlaniec*, P. Śniatała*, A. Rybarczyk*

Abstract: The aim of this paper is to show how the Hodgkin-Huxley model of the neuron's membrane potential can be extended to a stochastic one. This extension can be done either by adding fluctuations to the equations of the model or by using Markov kinetic schemes' formalism. We are presenting a new extension of the model. This modification simplifies computational complexity of the neuron model especially when considering a hardware implementation. The hardware implementation of the extended model as a system on a chip using a field-programmable gate array (FPGA) is demonstrated in this paper. The results confirm the reliability of the extended model presented here.

Key words: *Hodgkin-Huxley model, stochastic differential equations, Markov kinetics, kinetic formalism, field-programmable gate array (FPGA)*

1. Introduction

A developed by nature in the evolution process biological neuron provides better results than the classic artificial neuron in information processing tasks. Current research shows that classification tasks could be resolved with the use of only one living neural cell [3]. This is a significant advantage over the artificial neural networks. More accurate neuron models could result in a higher level of artificial intelligence (AI) in developed devices.

The main problem in using biological neuron models is their computational complexity requiring long simulation time and the necessity of using powerful, and expensive computing machines. In this work we present an efficient model of a neural cell optimized for hardware implementation. Our model is unique because it reduces computational complexity while maintaining full mathematical projection of the modeled processes.

---
*Aleksandra Świetlicka – Corresponding Author, Karol Gugała, Agata Jurkowlaniec, Paweł Śniatała and Andrzej Rybarczyk: Poznan University of Technology, Faculty of Computer Science, Chair of Computer Engineering, ul. Piotrowo 3A, 60-179 Poznań, POLAND, Tel: +48 61 665 2199, Fax: +48 61 665 2593, E-mail: {aleksandra.swietlicka, pawel.sniatala, andrzej.rybarczyk}@put.poznan.pl, karol.gugala@gmail.com, agata.jurkowlaniec@gmail.com

In this work we focus on the neuron's membrane potential. The first model of the neuron was created by A.L. Hodgkin and A.F. Huxley [16]. It is based on the idea that the electrical properties of the membrane can be modeled by an equivalent electrical circuit. Fig. 1 shows how the input current flows through the membrane [2, 12, 19].
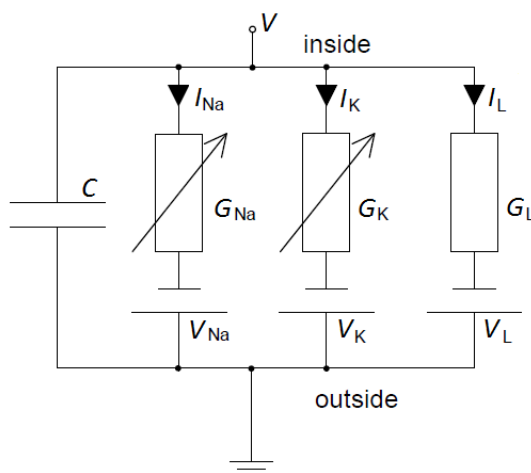

Fig. 1 *Equivalent circuit of the potential on the membrane of the neuron [2, 12, 19].*

For all models shown in this paper, the current, that flows through the membrane, is composed of two main components and can be written in this general equation [2]

$$C\frac{\mathrm{d}V}{\mathrm{d}t} + I_{\mathrm{ion}} = I_{\mathrm{ext}}, \tag{1}$$

where C is the capacitance of the membrane and V is the intracellular potential.

The first one I_{ext} expresses the input current that is applied externally to the neuron. The second current I_{ion} is related with movement of certain types of ions through the membrane. The neuron's membrane is composed of voltage-gated channels that control the movement of ions between the interior and the exterior of the neuron. In the classic approach (e.g. in the Hodgkin-Huxley model) each gate can be in one of the two states – permissive (open), and non-permissive (closed). There are three types of ions (according to the circuit shown in Fig. 1): sodium Na, potassium K, and chloride L. Type L is associated with a small leakage current (hence the "L" abbreviation). Additionally, each ion element has its own conductivity and potential. Hence, the ion current can be written in the following form [2]

$$I_{\mathrm{ion}} = \sum_i I_i = \sum_i \bar{g}_i \left(V - V_i\right), \tag{2}$$

where $\bar{g}_i$ is the ion conductivity and V_i is the ion potential, for $i \in \{\mathrm{Na}, \mathrm{K}, \mathrm{L}\}$.

In this paper we are presenting a new extension of the basic Hodgkin-Huxley model. Our approach increases the reliability of the model, thus it is making it more

accurate. The derived model, described in Section 4, is used for the FPGA-based hardware implementation as a system on a chip.

This paper is organized as follows: in Section 2, we are presenting the state of the art approaches for the neuron models by describing its action potential. A brief comparison of the different models will summarize this section. In Section 3, we describe the implementation of the kinetic models using any mathematical environment. In our case, we are using MATLAB environment. Then, in Section 4, we propose the extension of the basic Hodgkin-Huxley model, where each ion gate can be in a random number of states. In Section 5, we describe the process of FPGA-based hardware implementation of our model. Finally in Section 6, we summarize our results.

2. Models of the neuron cell

This section shows the existing model spectrum. Beginning with the basic Hodgkin-Huxley model, then followed by the simplified versions of this model (e.g. the LIF model [18] and the Izhikevich model [29]) ending with a complex amplification of the model. In the first subsection, we go over the main model, the basic Hodgkin-Huxley model. The following subsection describes the simplified models, and then the extensions with the involvement of stochasticity are presented.

2.1 Basic Hodgkin-Huxley model

The basic Hodgkin-Huxley model consists of four equations. The first one is describing the dynamics of membrane potential V [2, 16]

$$C\frac{\mathrm{d}V}{\mathrm{d}t} = I - \bar{g}_{\mathrm{Na}}\left(V - V_{\mathrm{Na}}\right) - \bar{g}_{\mathrm{K}}\left(V - V_{\mathrm{K}}\right) - \bar{g}_{\mathrm{L}}\left(V - V_{\mathrm{L}}\right), \qquad (3)$$

where elements $\bar{g}_{\mathrm{Na}} = g_{\mathrm{Na}}m^3h$, $\bar{g}_{\mathrm{K}} = g_{\mathrm{K}}n^4$ and $\bar{g}_{\mathrm{L}} = g_{\mathrm{L}}$ are ion conductivities, while m, n and h are the ion variables, describing the processes of movement of ions through the membrane. Values of conductivities, as well as of the associated potentials are shown in Tab. I. We also assume that $C = 1$ μF/cm^2 while the potential of the membrane at time $t = 0$ [s] is $V(0) = -67$ mV [12]. The I variable represents the I_{ext} in the equation (1), which express the current that is applied externally to the neuron. The values shown in Tab. I are based on the assumption that the resting potential is zero.

i	V_i [mV]	g_i [mS/cm^2]
Na	115	120
K	-12	36
L	10.6	0.3

Tab. I *The parameters of Hodgkin-Huxley model.*

The membrane is composed of small voltage-gated channels, which control the movement of ions through the membrane. A single gate can be in one of two states, namely permissive (open), or non-permissive (closed), meaning that a single

gate allows, or does not allow the movement of ions through the membrane. The potential V is related to the ion conductivity by making the assumption that the probability of a single gate being in the permissive or the non-permissive state depends upon the value of the potential on the membrane. When considering the i-th gate, the variable p_i represents the probability that this gate is in the permissive state. Transitions between these two states are governed by equation [2]

$$\frac{\mathrm{d}p_i}{\mathrm{d}t} = \alpha_i\left(V\right)\left(1 - p_i\right) - \beta_i\left(V\right)p_i, \tag{4}$$

where $\alpha_i\left(V\right)$ and $\beta_i\left(V\right)$ are functions of potential V. First part of equation (4) $\alpha_i\left(V\right)\left(1 - p_i\right)$ is the transition rate in-between the non-permissive and the permissive states, while the other part $\beta_i\left(V\right)p_i$ – in-between the permissive and the non-permissive states. The remaining three equations of the model are of the form (4), for $i \in \{m, n, h\}$. Functions $\alpha_i\left(V\right)$ and $\beta_i\left(V\right)$ are shown in Tab. II [2].

i	$\alpha_i\left(V\right)$	$\beta_i\left(V\right)$
n	$\dfrac{0.01 \cdot (10 - V)}{\exp((10 - V)/10) - 1}$	$0.125 \cdot \exp\left(-V/80\right)$
m	$\dfrac{0.1 \cdot (25 - V)}{\exp\left((25 - V)/10\right) - 1}$	$4 \cdot \exp\left(-V/18\right)$
h	$0.07 \cdot \exp\left(-V/20\right)$	$\dfrac{1}{\exp\left((30 - V)/10\right) + 1}$

Tab. II $\alpha_i\left(V\right)$ *and* $\beta_i\left(V\right)$ *rate functions [2].*

If the potential on the membrane settles on a certain value, the gates in the permissive state approach a steady state when time approaches infinity [2]

$$p_{i,t\to\infty}\left(V\right) = \frac{\alpha_i\left(V\right)}{\alpha_i\left(V\right) + \beta_i\left(V\right)}, \tag{5}$$

and the time of approaching this value is given by [2]

$$\tau_i\left(V\right) = \frac{1}{\alpha_i\left(V\right) + \beta_i\left(V\right)}. \tag{6}$$

Dynamics of the membrane's potential are shown in Fig. 2, for two different input currents. It can be seen that the number of peaks increases with higher current input.

2.2   Simplifications of the basic Hodgkin-Huxley model

Due to computational complexity of the Hodgkin-Huxley model, and significant implementation costs, various simplifications of the neural models were invented, e.g. Leaky-Integrate-Fire (LIF) model [18], or Izhikevich model [29]. Both models
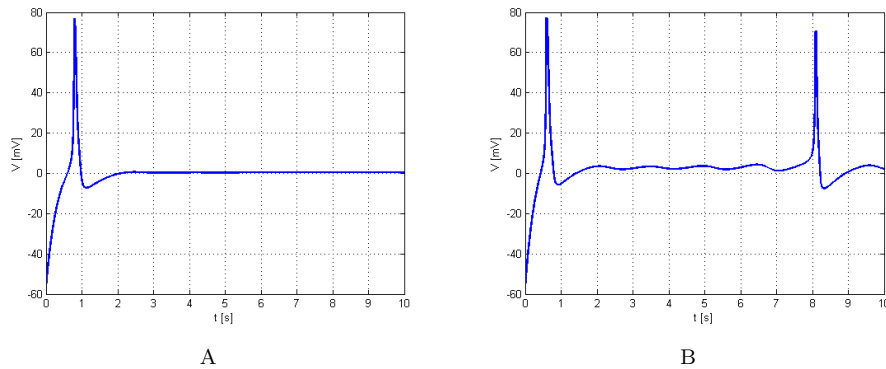
Fig. 2 *Waveforms of the potential obtained with the basic Hodgkin-Huxley model for different current inputs: A. $I = 0$ nA, B. $I = 30$ nA.*

reduce computational power requirements; however the projection accuracy of the biological neural cell behavior is reduced. As it was shown at the beginning of this paper, the classification problem could be solved using only one living neural cell [23]. However, it cannot be solved by using only one artificial neuron. Thus, in our opinion, the simplification of neural models in living cell is not the best possible way in artificial intelligence (AI) development.

2.3   Deterministic kinetic model based on Markov schemes

One of possible extensions of the Hodgkin-Huxley model is by adding some random fluctuations to the ion equations [5, 11, 25]

$$\mathrm{d}p_i = (\alpha_i\,(V)\,(1 - p_i) - \beta_i\,(V)\,p_i)\,\mathrm{d}t + \sigma dW, \tag{7}$$

where σ is a constant value, while W represents the Brownian motion [25]. This description allows for modeling a stochastic behavior of ion channels, which are activated by changes of potential that take place near these channels.

We, however, will consider other possible extension of Hodgkin-Huxley model, which use Markov kinetic formalism.
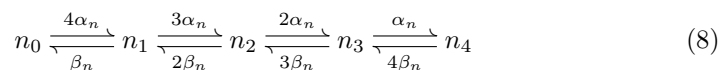
Markov's approach is commonly used in both artificial [1, 24, 31, 32], and biological [4, 26] neural networks. We want to present the approach of Markov's kinetic schemes [6] in the modeling of the neuron's membrane potential.

Single ion gate is a discrete element, so it can be described only in a stochastic way. The best way to describe the activity of the neurons is by showing the interaction between stochastic discrete elements.

At first we will show how to make the translation from the main Hodgkin-Huxley model to the kinetic one. Then we will extend the model to the stochastic version [26].
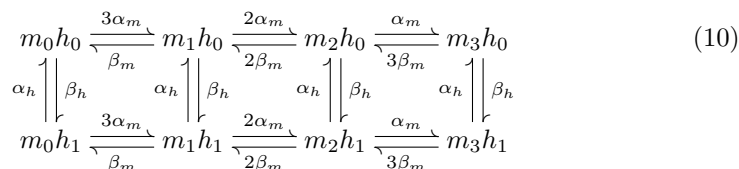
In the basic Hodgkin-Huxley model all ion gates can only exist in one of two states. In the kinetic model we will assume that the potassium gates can exist in one of five states, where the relationship between states is expressed with the Markov

kinetic scheme shown in (8). In this case only one state is permissive, while the remaining ones are non-permissive. Value $[n_i]$ is an amount of i permissive gates, so n_4 like n^4 in the basic Hodgkin-Huxley model means that four permissive gates are required, that flow of potassium ions is possible, hence the form of the equation (9) for the potassium conductivity [26].

$$n_0 \xrightarrow[\beta_n]{4\alpha_n} n_1 \xrightarrow[2\beta_n]{3\alpha_n} n_2 \xrightarrow[3\beta_n]{2\alpha_n} n_3 \xrightarrow[4\beta_n]{\alpha_n} n_4 \tag{8}$$

$$\bar{g}_K (V, t) = \varphi_K [n_4] \tag{9}$$

A sodium gate can be in one of eight states (10), with only one permissive as well. The open state is m_3h_0 which is equivalent to m^3h element in the basic model, hence the form of the equation (11) for the sodium conductivity [26].

$$
\begin{array}{ccccccc}
m_0h_0 & \xrightleftharpoons[\beta_m]{3\alpha_m} & m_1h_0 & \xrightleftharpoons[2\beta_m]{2\alpha_m} & m_2h_0 & \xrightleftharpoons[3\beta_m]{\alpha_m} & m_3h_0 \\
\alpha_h \Big\updownarrow \beta_h & & \alpha_h \Big\updownarrow \beta_h & & \alpha_h \Big\updownarrow \beta_h & & \alpha_h \Big\updownarrow \beta_h \\
m_0h_1 & \xrightleftharpoons[\beta_m]{3\alpha_m} & m_1h_1 & \xrightleftharpoons[2\beta_m]{2\alpha_m} & m_2h_1 & \xrightleftharpoons[3\beta_m]{\alpha_m} & m_3h_1
\end{array}
\tag{10}
$$

$$\bar{g}_{Na} (V, t) = \varphi_{Na} [m_3h_0] \tag{11}$$

In the equations (9) and (11), φ_K and φ_{Na} are the conductivities of a single ion gate that is in the open state.

The main equation of the kinetic model is basically the same as in the Hodgkin-Huxley model and it is of the form (3). After substituting the $\bar{g}_{Na}$, $\bar{g}_K$ and $\bar{g}_L$ values, we can write [27]

$$C\frac{\mathrm{d}V}{\mathrm{d}t} = I - \varphi_{Na} [m_3h_0] (V - V_{Na}) - \varphi_K [n_4] (V - V_K) - g_L (V - V_L). \tag{12}$$

Fig. 3 shows examples of waveforms of the potential obtained from the deterministic kinetic model. It is possible to notice that with higher input current the number of peaks is increasing. For this example, changing the current from $I = 0$ nA to $I = 30$ nA almost doubled the number of peaks.

2.4 Stochastic kinetic model

This subsection shows the translation from the deterministic kinetic model to the stochastic version [26].

Instead of accounting every gate separately, we take a more reliable scheme, where we take into account all of the gates.

Let us assume that at a time t in state A there are n_A gates, similar for state B – at the time t in the state B there are n_B gates. Value r_{AB} is a rate function that describes transition of one gate from state A to state B, similarly r_{BA} describes transition of one gate from state B to state A. Probability of transfer of one gate from state A to state B in the time interval t and $t + \Delta t$ is $p = \Delta t \cdot r_{AB}$. Hence, for every step in time it is possible to obtain a number of gates Δn_{AB}, that transfer
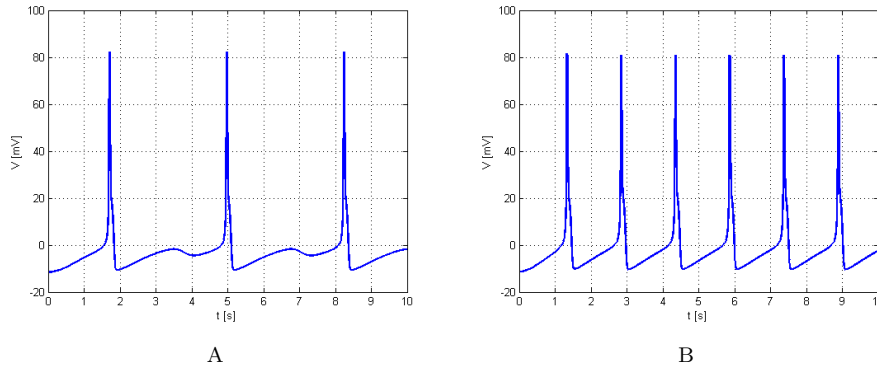
Fig. 3 *Waveforms of the potential obtained with the deterministic kinetic model for different input currents: A. $I = 0$nA, B. $I = 30$ nA. Ion gates can be in 8 and 5 states for sodium and potassium respectively.*

from state A to state B, by choosing a random number from binomial distribution, where $n = n_A$ and $k = \Delta n_{AB}$. Under certain assumptions [10] it is possible to approximate the binomial distribution with the normal one, with $\mu = \Delta n_{AB} \cdot p$ and $\sigma = \Delta n_{AB} \cdot p \cdot (1 - p)$.

The simulation results of the stochastic kinetic model are illustrated in Fig. 4. The number of gates is determined in each step in time separately. Depending on the random number generator peak irregularities appear as shown in Fig. 4A. For higher input currents it is hard to perceive the difference between the waveforms due to the high frequency of peaks.
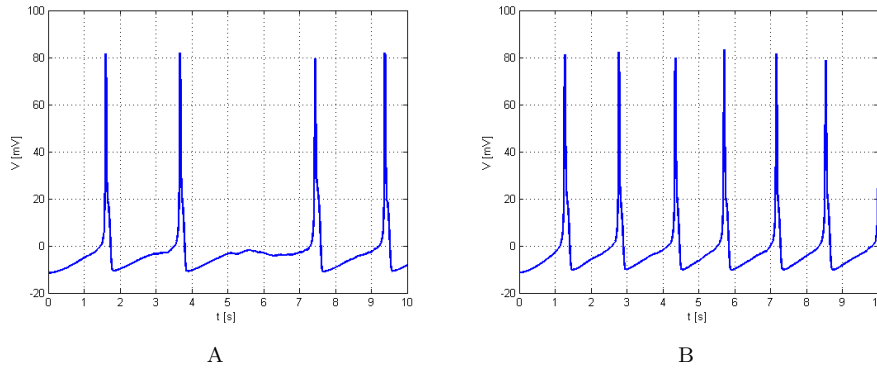

Fig. 4 *Waveforms of the potential obtained with the stochastic kinetic model for different input currents: A. $I = 0$ nA, B. $I = 30$ nA. Ion gates can be in 8 and 5 states for sodium and potassium respectively.*

The difference between deterministic and stochastic kinetic models can be illustrated by showing the difference between the waveforms of mean number of gates that change their states as a function of time. Input current is set to zero. These

waveforms are shown in Figs. 5 and 6. In both cases, top figures correspond to the deterministic kinetic model, while the bottom ones to the stochastic version of the model.

Fig. 5 shows how many gates in the current step in time are changed from being closed to being open (transfer from the state n_4 to the state n_3) and otherwise. Similarly in Fig. 6, the transfers between the state $m_3 h_0$ and two adjacent states $m_2 h_0$ and $m_3 h_1$ are illustrated.

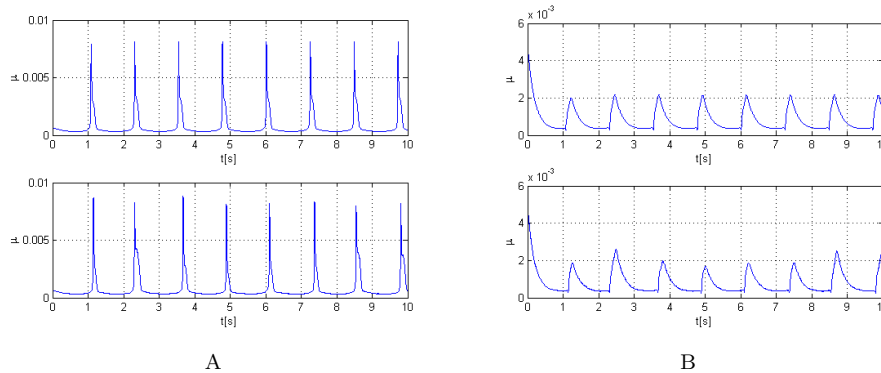
A   B

Fig. 5 *Differences in waveforms of the mean values of number of gates for scheme with five states, for input current $I = 0$ nA A. The number of gates that transfer from state n_3 to state n_4. B. The number of gates that transfer from state n_4 to state n_3.*

Changes between the waveforms of mean values in Figs. 5 and 6 are small, even unnoticeable, which means that there are small differences between the deterministic and the stochastic versions of the kinetic model. Events that take place in the nervous system are not predictable enough and processes that take place in it are not regular enough to assume that the potential on the cell's membrane appears in the form of regular spikes. That is why the stochastic version of kinetic model gives a reliable description of the cell's membrane potential.

Additionally, to strengthen the argument that the disparity between the mean values is slight, we are showing the difference between waveforms of these mean values. In Fig. 7 we are showing the difference between the mean values calculated with the deterministic and stochastic kinetic model of the neuron. The difference is in the order of 10^{-3} therefore the disparity between the two versions of the model is negligible and the stochastic model can be successfully used instead of the deterministic model.

2.5   Advantages and disadvantages of the models

Each model has advantages and disadvantages. The main advantage of the simpler models (such as the main Hodgkin-Huxley one and its stochastic extension) over their extensions presented in this paper (kinetic model and extended kinetic model) is their simpler software implementation. On the other hand, these models are not
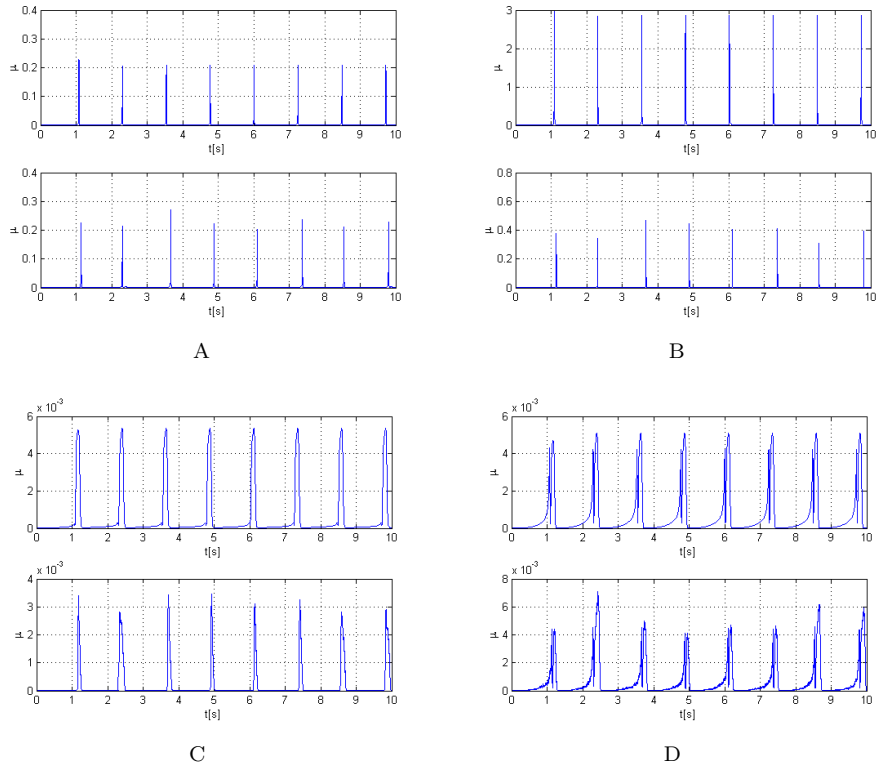
Fig. 6 *Differences in waveforms of the mean values of number of gates for scheme with eight states, for input current $I = 0$ nA. A. The number of gates that transfer from state m_2h_0 to state m_3h_0. B. The number of gates that transfer from state m_3h_0 to state m_2h_0. C. The number of gates that transfer from state m_3h_1 to state m_3h_0. D. The number of gates that transfer from state m_3h_0 to state m_3h_1.*
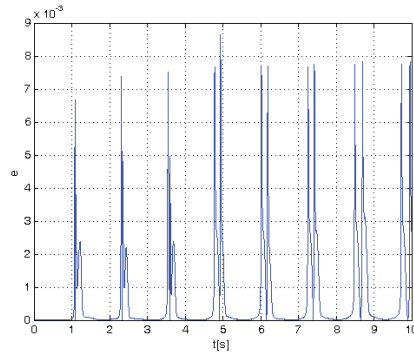

Fig. 7 *The difference between the mean values of gates (for scheme with five states) that transfer from state n_3 to state n_4 obtained with deterministic and stochastic versions of the kinetic model.*

showing every aspect of the neuron nor the processes that take place within that the kinetic model is providing.

The kinetic model is more trustworthy but requires higher computational complexity. This problem is solved with the stochastic approach, which reduces a lot of calculations by replacing them with the drawing numbers from the normal distribution.

3. Backward Euler implementation of deterministic kinetic model

Scheme (13) shows three possibilities of connections between states, that appear in kinetic schemes (8) and (10), used in the kinetic models of the neuron. Using a kinetic formalism we can write five differential equations of the form (14) that show the dynamics between these states.

$$x_0 \xrightarrow[r_{10}]{r_{01}} x_1 \xrightarrow[r_{21}]{r_{12}} x_2 \xrightarrow[r_{32}]{r_{23}} x_3 \tag{13}$$
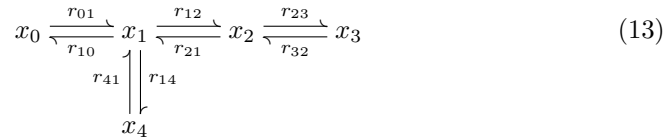
$$
\begin{aligned}
\frac{\mathrm{d}\,[x_0]}{\mathrm{d}t} &= r_{10}\,[x_1] - r_{01}\,[x_0] \\
\frac{\mathrm{d}\,[x_1]}{\mathrm{d}t} &= r_{01}\,[x_0] + r_{41}\,[x_4] + r_{21}\,[x_2] - [x_1]\,(r_{10} + r_{12} + r_{14}) \\
\frac{\mathrm{d}\,[x_2]}{\mathrm{d}t} &= r_{12}\,[x_1] + r_{32}\,[x_3] - [x_2]\,(r_{21} + r_{23}) \\
\frac{\mathrm{d}\,[x_3]}{\mathrm{d}t} &= r_{23}\,[x_2] - r_{32}\,[x_3] \\
\frac{\mathrm{d}\,[x_4]}{\mathrm{d}t} &= r_{14}\,[x_1] - r_{41}\,[x_4]
\end{aligned}
\tag{14}
$$

Using backward Euler method (which is quite simple from the implementation point of view and has advantages of the implicit methods like accuracy and stability [17]) of the form

$$x^{i+1} = x^i + \Delta t \cdot f\left(t^{i+1}, x^{i+1}\right), \tag{15}$$

where the right upper i and $i+1$ indexes denote the actual and the following steps in time, respectively and

$$f(t, x) = \frac{\mathrm{d}x}{\mathrm{d}t}, \tag{16}$$

we can rewrite the equations of the form (14) into the form

$$
\begin{aligned}
[x_0]^{i+1} &= [x_0]^i + \Delta t \left(r_{10} [x_1]^{i+1} - r_{01} [x_0]^{i+1} \right), \\
[x_1]^{i+1} &= [x_1]^i + \Delta t \left(r_{01} [x_0]^{i+1} + r_{41} [x_4]^{i+1} + r_{21} [x_2]^{i+1} \right. \\
&\qquad\qquad\qquad\qquad \left. - [x_1]^{i+1} (r_{10} + r_{12} + r_{14}) \right)' \\
[x_2]^{i+1} &= [x_2]^i + \Delta t \left(r_{12} [x_1]^{i+1} + r_{32} [x_3]^{i+1} - [x_2]^{i+1} (r_{21} + r_{23}) \right), \\
[x_3]^{i+1} &= [x_3]^i + \Delta t \left(r_{23} [x_2]^{i+1} - r_{32} [x_3]^{i+1} \right), \\
[x_4]^{i+1} &= [x_4]^i + \Delta t \left(r_{14} [x_1]^{i+1} - r_{41} [x_4]^{i+1} \right).
\end{aligned}
\tag{17}
$$

In the same way, we can rewrite the main equation of the kinetic model (12) into the following form

$$
\begin{aligned}
V^{i+1} = V^i + \frac{\Delta t}{C} \Big(I - \varphi_{\text{Na}} [m_3 h_0]^{i+1} \left(V^{i+1} - V_{\text{Na}} \right) \\
- \varphi_{\text{K}} [n_4]^{i+1} \left(V^{i+1} - V_{\text{K}} \right) - g_{\text{L}} \left(V^{i+1} - V_{\text{L}} \right) \Big).
\end{aligned}
\tag{18}
$$

Using the Markov kinetic schemes of size 8 and 5 for sodium and potassium ion gates, respectively, we can formulate the set of equations describing the waveform of potential V, of the form (19), presented on following page.

Transforming this set of equations using the backward Euler method (15) leads to the set of algebraic equations of the form (20).

In Fig. 8 we are presenting the results of simulating the Hodgkin-Huxley model, discretized with the implicit Euler method.
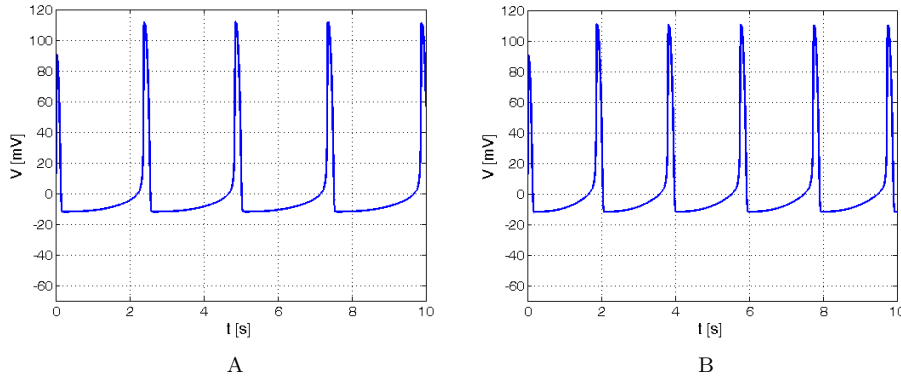

A   B

Fig. 8 *Waveforms of the potential obtained with the kinetic Hodgkin-Huxley model (discretized with the implicit Euler method) for different current inputs: A. $I = 0$ nA, B. $I = 30$ nA. Ion gates can be in 8 and 5 states for sodium and potassium respectively.*

$$C\frac{\mathrm{d}V}{\mathrm{d}t} = I - \varphi_{\mathrm{Na}}\left[m_3 h_0\right](V - V_{\mathrm{Na}}) - \varphi_{\mathrm{K}}\left[n_4\right](V - V_{\mathrm{K}})$$
$$- g_{\mathrm{L}}(V - V_{\mathrm{L}})$$

$$\frac{\mathrm{d}\left[n_0\right]}{\mathrm{d}t} = \beta_n\left[n_1\right] - 4\alpha_n\left[n_0\right]$$

$$\frac{\mathrm{d}\left[n_1\right]}{d\mathrm{d}t} = 4\alpha_n\left[n_0\right] + 2\beta_n\left[n_2\right] - \left[n_1\right](\beta_n + 3\alpha_n)$$

$$\frac{\mathrm{d}\left[n_2\right]}{\mathrm{d}t} = 3\alpha_n\left[n_1\right] + 3\beta_n\left[n_3\right] - \left[n_2\right](2\alpha_n + 2\beta_n)$$

$$\frac{\mathrm{d}\left[n_3\right]}{\mathrm{d}t} = 2\alpha_n\left[n_2\right] + 4\beta_n\left[n_4\right] - \left[n_3\right](3\beta_n + \alpha_n)$$

$$\frac{\mathrm{d}\left[n_4\right]}{\mathrm{d}t} = \alpha_n\left[n_3\right] - 4\beta_n\left[n_4\right]$$

$$\frac{\mathrm{d}\left[m_0 h_0\right]}{\mathrm{d}t} = \beta_m\left[m_1 h_0\right] + \alpha_h\left[m_0 h_1\right] - \left[m_0 h_0\right](\beta_h + 3\alpha_m)$$

$$\frac{\mathrm{d}\left[m_1 h_0\right]}{\mathrm{d}t} = 3\alpha_m\left[m_0 h_0\right] + \alpha_h\left[m_1 h_1\right] + 2\beta_m\left[m_2 h_0\right]$$
$$- \left[m_1 h_0\right](\beta_m + 2\alpha_m + \beta_h)$$

$$\frac{\mathrm{d}\left[m_2 h_0\right]}{\mathrm{d}t} = 2\alpha_m\left[m_1 h_0\right] + \alpha_h\left[m_2 h_1\right] + 3\beta_m\left[m_3 h_0\right]$$
$$- \left[m_2 h_0\right](2\beta_m + \beta_h + \alpha_m)$$

$$\frac{\mathrm{d}\left[m_3 h_0\right]}{\mathrm{d}t} = \alpha_m\left[m_2 h_0\right] + \alpha_h\left[m_3 h_1\right] - \left[m_3 h_0\right](3\beta_m + \beta_h)$$

$$\frac{\mathrm{d}\left[m_0 h_1\right]}{\mathrm{d}t} = \beta_h\left[m_0 h_0\right] + \beta_m\left[m_1 h_1\right] - \left[m_0 h_1\right](\alpha_h + 3\alpha_m)$$

$$\frac{\mathrm{d}\left[m_1 h_1\right]}{\mathrm{d}t} = 3\alpha_m\left[m_0 h_1\right] + \beta_h\left[m_1 h_0\right] + 2\beta_m\left[m_2 h_1\right]$$
$$- \left[m_1 h_1\right](\beta_m + \alpha_h + 2\alpha_m)$$

$$\frac{\mathrm{d}\left[m_2 h_1\right]}{\mathrm{d}t} = 2\alpha_m\left[m_1 h_1\right] + \beta_h\left[m_2 h_0\right] + 3\beta_m\left[m_3 h_1\right]$$
$$- \left[m_2 h_1\right](2\beta_m + \alpha_h + \alpha_m)$$

$$\frac{\mathrm{d}\left[m_3 h_1\right]}{\mathrm{d}t} = \alpha_m\left[m_2 h_1\right] + \beta_h\left[m_3 h_0\right] - \left[m_3 h_1\right](3\beta_m + \alpha_h)$$

$$(19)$$

$$V^{i+1} = V^i + \frac{\Delta t}{C} \left(I - \varphi_{\text{Na}} \left[m_3 h_0 \right]^{i+1} \left(V^{i+1} - V_{\text{Na}} \right) \right.$$
$$\left. - \varphi_{\text{K}} \left[n_4 \right]^{i+1} \left(V^{i+1} - V_{\text{K}} \right) - g_{\text{L}} \left(V^{i+1} - V_{\text{L}} \right) \right)$$

$$\left[n_0 \right]^{i+1} = \left[n_0 \right]^i + \Delta t \left(\beta_n \left[n_1 \right]^{i+1} - 4\alpha_n \left[n_0 \right]^{i+1} \right)$$

$$\left[n_1 \right]^{i+1} = \left[n_1 \right]^i + \Delta t \left(4\alpha_n \left[n_0 \right]^{i+1} + 2\beta_n \left[n_2 \right]^{i+1} \right.$$
$$\left. - \left[n_1 \right]^{i+1} \left(\beta_n + 3\alpha_n \right) \right)$$

$$\left[n_2 \right]^{i+1} = \left[n_2 \right]^i + \Delta t \left(3\alpha_n \left[n_1 \right]^{i+1} + 3\beta_n \left[n_3 \right]^{i+1} \right.$$
$$\left. - \left[n_2 \right]^{i+1} \left(2\alpha_n + 2\beta_n \right) \right)$$

$$\left[n_3 \right]^{i+1} = \left[n_3 \right]^i + \Delta t \left(2\alpha_n \left[n_2 \right]^{i+1} + 4\beta_n \left[n_4 \right]^{i+1} \right.$$
$$\left. - \left[n_3 \right]^{i+1} \left(3\beta_n + \alpha_n \right) \right)$$

$$\left[n_4 \right]^{i+1} = \left[n_4 \right]^i + \Delta t \left(\alpha_n \left[n_3 \right]^{i+1} - 4\beta_n \left[n_4 \right]^{i+1} \right)$$

$$\left[m_0 h_0 \right]^{i+1} = \left[m_0 h_0 \right]^i + \Delta t \left(\beta_m \left[m_1 h_0 \right]^{i+1} + \alpha_h \left[m_0 h_1 \right]^{i+1} \right.$$
$$\left. - \left[m_0 h_0 \right]^{i+1} \left(\beta_h + 3\alpha_m \right) \right)$$

$$\left[m_1 h_0 \right]^{i+1} = \left[m_1 h_0 \right]^i + \Delta t \left(3\alpha_m \left[m_0 h_0 \right]^{i+1} + \alpha_h \left[m_1 h_1 \right]^{i+1} \right.$$
$$\left. + 2\beta_m \left[m_2 h_0 \right]^{i+1} - \left[m_1 h_0 \right]^{i+1} \left(\beta_m + 2\alpha_m + \beta_h \right) \right)$$

$$\left[m_2 h_0 \right]^{i+1} = \left[m_2 h_0 \right]^i + \Delta t \left(2\alpha_m \left[m_1 h_0 \right]^{i+1} + \alpha_h \left[m_2 h_1 \right]^{i+1} \right.$$
$$\left. + 3\beta_m \left[m_3 h_0 \right]^{i+1} - \left[m_2 h_0 \right]^{i+1} \left(2\beta_m + \beta_h + \alpha_m \right) \right)$$

$$\left[m_3 h_0 \right]^{i+1} = \left[m_3 h_0 \right]^i + \Delta t \left(\alpha_m \left[m_2 h_0 \right]^{i+1} + \alpha_h \left[m_3 h_1 \right]^{i+1} \right.$$
$$\left. - \left[m_3 h_0 \right]^{i+1} \left(3\beta_m + \beta_h \right) \right)$$

$$\left[m_0 h_1 \right]^{i+1} = \left[m_0 h_1 \right]^i + \Delta t \left(\beta_h \left[m_0 h_0 \right]^{i+1} + \beta_m \left[m_1 h_1 \right]^{i+1} \right.$$
$$\left. - \left[m_0 h_1 \right]^{i+1} \left(\alpha_h + 3\alpha_m \right) \right)$$

$$\left[m_1 h_1 \right]^{i+1} = \left[m_1 h_1 \right]^i + \Delta t \left(3\alpha_m \left[m_0 h_1 \right]^{i+1} + \beta_h \left[m_1 h_0 \right]^{i+1} \right.$$
$$\left. + 2\beta_m \left[m_2 h_1 \right]^{i+1} - \left[m_1 h_1 \right]^{i+1} \left(\beta_m + \alpha_h + 2\alpha_m \right) \right)$$

$$\left[m_2 h_1 \right]^{i+1} = \left[m_2 h_1 \right]^i + \Delta t \left(2\alpha_m \left[m_1 h_1 \right]^{i+1} + \beta_h \left[m_2 h_0 \right]^{i+1} \right.$$
$$\left. + 3\beta_m \left[m_3 h_1 \right]^{i+1} - \left[m_2 h_1 \right]^{i+1} \left(2\beta_m + \alpha_h + \alpha_m \right) \right)$$

$$\left[m_3 h_1 \right]^{i+1} = \left[m_3 h_1 \right]^i + \Delta t \left(\alpha_m \left[m_2 h_1 \right]^{i+1} + \beta_h \left[m_3 h_0 \right]^{i+1} \right.$$
$$\left. - \left[m_3 h_1 \right]^{i+1} \left(3\beta_m + \alpha_h \right) \right)$$

$$(20)$$

4. Extended kinetic model

We decided to perform a test of simple extension of the kinetic model by enlarging kinetic schemes for potassium and sodium gates. Hence we assume that the scheme for potassium gates is of the form

$$n_0 \underset{\beta_n}{\overset{k\alpha_n}{\rightleftarrows}} n_1 \underset{2\beta_n}{\overset{(k-1)\alpha_n}{\rightleftarrows}} n_2 \underset{3\beta_n}{\overset{(k-2)\alpha_n}{\rightleftarrows}} \cdots \underset{(k-1)\beta_n}{\overset{2\alpha_n}{\rightleftarrows}} n_{k-1} \underset{k\beta_n}{\overset{\alpha_n}{\rightleftarrows}} n_k \tag{21}$$

The size of this scheme is obtained by $k \in \{0, \ldots, 31\}$, so the biggest scheme is built from 32 states. Similar for sodium gates – the size of scheme is obtained by $l \in \{0, \ldots, 15\}$, so also the biggest scheme is built from 32 states, arranged in two rows

$$
\begin{array}{c}
m_0 h_0 \underset{\beta_m}{\overset{l\alpha_m}{\rightleftarrows}} m_1 h_0 \underset{2\beta_m}{\overset{(l-1)\alpha_m}{\rightleftarrows}} m_2 h_0 \underset{3\beta_m}{\overset{(l-2)\alpha_m}{\rightleftarrows}} \cdots \underset{(l-1)\beta_m}{\overset{2\alpha_m}{\rightleftarrows}} m_{l-1} h_0 \underset{l\beta_m}{\overset{\alpha_m}{\rightleftarrows}} m_l h_0 \\[4pt]
\alpha_h \updownarrow \beta_h \qquad \alpha_h \updownarrow \beta_h \qquad \alpha_h \updownarrow \beta_h \qquad\qquad \alpha_h \updownarrow \beta_h \qquad \alpha_h \updownarrow \beta_h \\[4pt]
m_0 h_1 \underset{\beta_m}{\overset{l\alpha_m}{\rightleftarrows}} m_1 h_1 \underset{2\beta_m}{\overset{(l-1)\alpha_m}{\rightleftarrows}} m_2 h_1 \underset{3\beta_m}{\overset{(l-2)\alpha_m}{\rightleftarrows}} \cdots \underset{(l-1)\beta_m}{\overset{2\alpha_m}{\rightleftarrows}} m_{l-1} h_1 \underset{l\beta_m}{\overset{\alpha_m}{\rightleftarrows}} m_l h_1
\end{array}
\tag{22}
$$

The main equation does not change its form – it is exactly the same as for the basic kinetic model (12). The locations of the open states and the sizes of schemes are changed. We assumed that the position of the open state can be chosen randomly, and can be set at any point, so $i \in \{0, \ldots, k\}$ and $j \in \{0, \ldots, l\}$. Hence the equations (23) for $\bar{g}_K(V, t)$ and $\bar{g}_{Na}(V, t)$, as

$$
\begin{aligned}
\bar{g}_K(V, t) &= \varphi_K[n_i], \\
\bar{g}_{Na}(V, t) &= \varphi_{Na}[m_j h_0].
\end{aligned}
\tag{23}
$$

The stochastic version of this model is obtained exactly in the same way as in case of the simpler kinetic model, as shown in Section 2.4. We are not providing the numerical scheme for the extended stochastic version of the model, since it is similar to the scheme presented as (20).

We proved empirically, by simulating all possible combinations of $\{k, l, i, j\}$, that the extended model exhibits better than the basic kinetic models shown earlier in this paper. The extended kinetic model is more complicated, but it is more reliable in describing the processes on the membrane of the neuron. The main advantage of this approach can be seen by waveforms of the membrane potential. The results obtained from this model show the unpredictable nature of the neurons, which in nature are unstable and thus not able to create regular spikes (which are involved in information transfer and thus are important in eventual applications).

To confirm that conclusion in Fig. 9, and Fig. 10, we presented the examples of waveforms of the potential obtained from the extended kinetic model in the deterministic and stochastic versions, respectively. Let us consider the plots in these figures: Fig. 9A and Fig. 10A. It is assumed here that size of the scheme for potassium gate is $k = 12$, and size of the scheme for sodium gate is $l = 11$.

The open states are set for $[n_{12}]$ and $[m_7 h_0]$, hence, the equation (12) will be of the form

$$C\frac{\mathrm{d}V}{\mathrm{d}t} = I - \varphi_{\mathrm{Na}}[m_7 h_0](V - V_{\mathrm{Na}}) - \varphi_{\mathrm{K}}[n_{12}](V - V_{\mathrm{K}}) - g_{\mathrm{L}}(V - V_{\mathrm{L}}). \qquad (24)$$
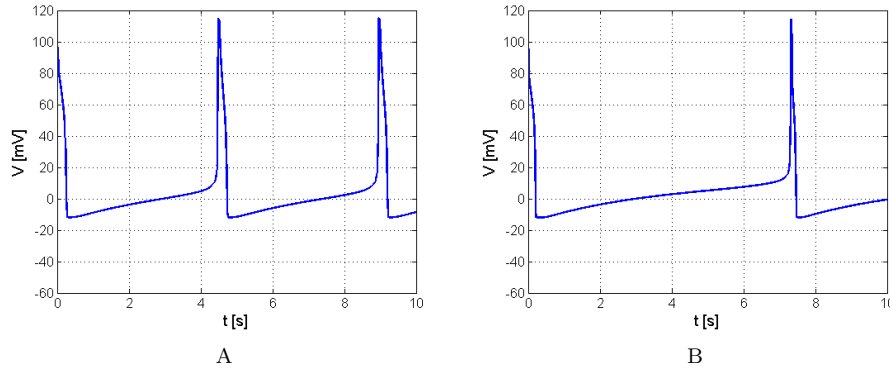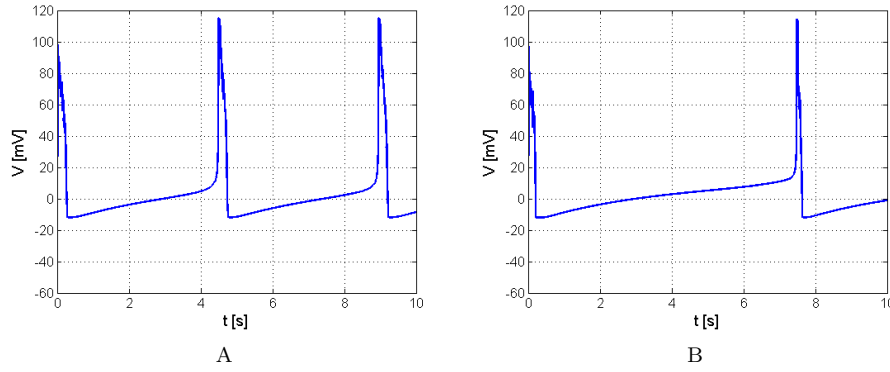

A   B

Fig. 9 *Waveforms of the potential obtained with the deterministic version of the extended kinetic model for $I = 0$ nA: A. $k = 12$, $l = 11$, open states: $[n_{12}]$ and $[m_7 h_0]$, B. $k = 31$, $l = 15$, open states: $[n_{26}]$ and $[m_9 h_0]$.*

A   B

Fig. 10 *Waveforms of the potential obtained with the stochastic version of the extended kinetic model for $I = 0$ nA: A. $k = 12$, $l = 11$, open states: $[n_{12}]$ and $[m_7 h_0]$, B. $k = 31$, $l = 15$, open states: $[n_{26}]$ and $[m_9 h_0]$.*

Finally, having the whole model described, after simulating all possible combinations, it is possible to extract a set of "proper" fours of the form $\{k, l, i, j\}$. Not all the combinations give the expected results, which means that we cannot observe spiking nature of the neuron in each case. Here for an example are shown two waveforms of the potential, where the fours are of the form $\{12, 11, 12, 7\}$ and $\{31, 15, 26, 9\}$.

Observing the plots shown in Fig. 9 and Fig. 10 it is impossible to perceive any differences between deterministic and stochastic waveforms obtained from the same set of the form $\{k, l, i, j\}$. The reason may be as simple as too small randomness which has too small influence on irregularities of appearing peaks of the potential. As we will show in Section 5, the random number generator implemented with use of FPGA platform gives more irregular peaks of the potential, thus it is more reliable.

Both, the stochastic version and deterministic version of the extended kinetic model provide very similar results. The following analysis evaluates these results. According to the absolute error of the potential

$$e\left(V\right) = \left|V_{\mathrm{ODE}} - V_{\mathrm{SDE}}\right|, \tag{25}$$

we designated the difference for the first plots shown in Fig. 9 and 10, where V_{ODE} and V_{SDE} are the vectors of potentials obtained from the deterministic and stochastic versions of the extended kinetic model, respectively. This difference shown in Fig. 11 clearly shows that the potential peaks obtained from two versions of the model are shifted with respect to each other. It is impossible to perceive it by observing the waveforms only. We are showing this difference to confirm that with properly implemented random number generator the potential peaks will appear irregularly.
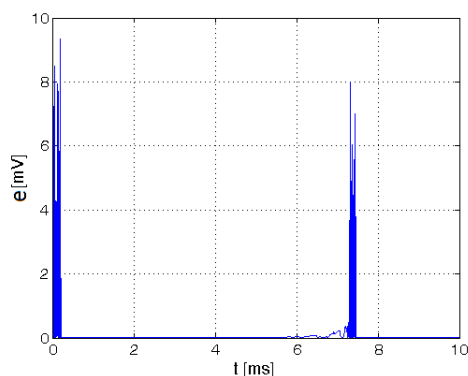

Fig. 11 *Difference between the deterministic and stochastic versions of the extended kinetic model for $k = 31$, $l = 15$, with open states $[n_{26}]$ and $[m_9 h_0]$.*

5. Hardware implementation of stochastic kinetic model

The presented equations describe a single neuron without taking into consideration its morphology. However, many studies suggest the importance of the geometrical structure of the neurons in the information processing [15, 22, 28]. The extension of the presented model to the compartmental description and its time-domain discretization generates a large system of algebraic equations leading to excessively

long simulation times. For every time step a set of the equations presented in Section 4 have to be solved. In order to achieve a reliable results time quantization has to be thick leading to nontrivial computation problem.

One of the possible solutions to the problem of computational complexity is the custom designed, problem oriented computational system that supports the most computationally demanding parts of the simulation process. Implementing additional dedicated hardware blocks accelerating some computational demanding parts of the neural networks simulations is common, widely used approach and it has proven its worth [7, 8, 20]. This section presents such an approach, resulting in FPGA-based hardware-implemented biologically realistic stochastic model of the neuron. The stochastic model requires less computational power due to replacement of differential equations by draw from normal distribution with specified parameters.

Different approaches to hardware accelerated simulating of a biological neural networks (e.g. use of GPU parallel computing) are discussed in the other authors works [13,14]. FPGA approach allows to add additional custom blocks speeding up the calculations. In the presented implementation use of the True Random Number Generator speeds up the whole calculation process while granting high accuracy.

The device was implemented as an embedded system, and it is based on Xilinx microblaze soft processor with floating point unit and uses True Random Number Generator (TRNG) based on inverter rings [30]. It was built on Virtex 5 FPGA on XUPV5-LX110T platform.

5.1   Proposed architecture

The device directly implements the presented extended stochastic Markov model presented in Section 4, i.e. computes the number of ion gates that exist in each of the states $n_0, \ldots, n_k$ and $\{m_0, m_1, m_2, \ldots, m_l\}h_0, \{m_0, m_1, m_2, \ldots, m_l\}h_1$ depicted in diagrams (21) and (22) respectively. In order to calculate the number of gates, in each of the states, the amount of gates Δn_{AB} that "transfers" from and to the adjacent states must be computed first. The number of "transferring" gates depends on the random number chosen from normal distribution. In order to compute the number of gates being in the open state (n_i and $m_j h_0$ in diagrams (23) respectively) the device should implement the following key functionality:

- fast True Random Number Generator generating the numbers governing the transfer process,
- module calculating the number of ions "transferring" between the Markov's scheme states,
- fast memory for storing temporary computation results.

The presented device consists of the following modules implementing the aforementioned functionality:

- fast hardware-based True Random Number Generator (TRNG),
- microblaze soft processor,
- 256 MB RAM memory.

The algorithm was written in ANSI C. Its flow is presented in the pseudocode listing 1:

Algorithm 1 The Microblaze algorithm

1: read the random data form TRNG
2: calculate the Δn_{AB} values for all the adjacent states in the Markov scheme basing on the numbers sampled from TRNG and current time (refer to $p = \Delta t \cdot r_{AB}$ in Section 2.4)
3: update the number of gates in every state
4: calculate the final number of gates being in the *open state*
5: calculate the g_{Na} and g_K parameters in the (12)
6: calculate the output (see Eq. (12)).

5.2 Main algorithm

The main part of the algorithm use random data to calculate number of ion gates in each state. Output potential calculation according to Eq. (12) is also performed here. Algorithm configuration, such as dimensions of Markov kinetic schemes and locations of the open states within it, is provided by a set of defines in the header file. Due to the large amount of data processed by the algorithm during calculations external RAM module is used to store partial results.

5.3 Implementation results

In Fig. 12 we are presenting two different waveforms of the potential, assuming for both that $I = 0$ nA, but different sizes of the Markov kinetic schemes.
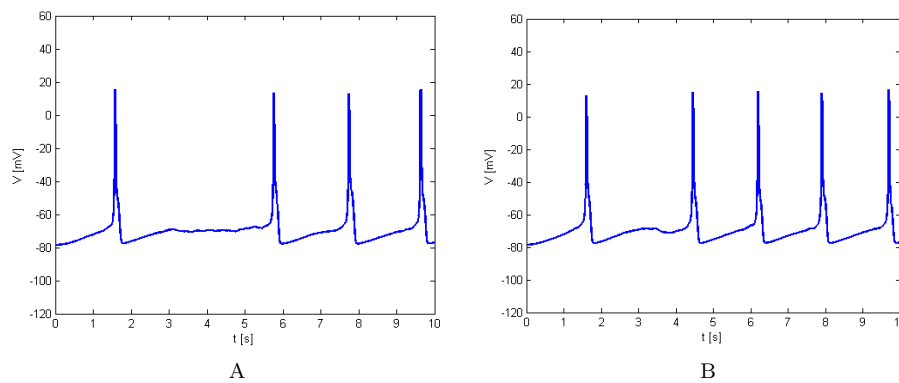

A   B

Fig. 12 *Waveforms of the potential obtained with the stochastic extended kinetic model (implemented in hardware) for $I = 0$ nA: A. $k = 12$, $l = 11$, open states: $[n_{12}]$ and $[m_7 h_0]$, B. $k = 31$, $l = 15$, open states: $[n_{26}]$ and $[m_9 h_0]$.*

6. Conclusions

The ion gate stochasticity, included in the Hodgkin-Huxley neural model, may result in the macroscopic behavior that replicates observed input-dependent reliability of the real neural cells. That model also shows other observed characteristics like the ones present in the real neurons, such as subthreshold oscillations and spontaneous spike generation [27].

The presented simulation confirms those observations. The basic Hodgkin-Huxley model and the deterministic kinetic model give periodic spikes of the potential. It has been shown that more spikes occur with increasing of input current. Because of the unpredictability of the nervous system, the stochastic version models are preferential because they enable the irregular appearance of the potential spikes. Unfortunately, with increase of input current it is difficult to differentiate between the deterministic and the stochastic kinetic models. The best results are noted for zero input current.

The authors expect that presented extension of the model providing better statistical properties, and an increased number of possible states against known solutions will enable better learning abilities, thus creating better association capability. The performed tests confirm the above thesis. This will be further verified by applying the network based on the presented models in various classification tasks.

Stochastic model using hardware random generator based on inverter rings is more similar to deterministic model than software simulation based on pseudo-random generator. High quality random number generation is crucial for stochastic simulation system due to the fact that statistical results can only be reliable when the independence of the samples can be guaranteed [21]. The true hardware random number generator presented in this paper provides the truly independent samples of the required high quality.

Additionally FPGA implementation prove of concept could first step towards full custom ASIC realization which can improve power dissipation and size of the whole system [9].

References

[1] BAO H., CAO J. Stochastic Global Exponential Stability for Neutral-Type Impulsive Neural Networks with Mixed Time-Delays and Markovian Jumping Parameters. *Communications in Nonlinear Science and Numerical Simulation,* 2011, 16(9), pp. 3786–3791, doi: 10.1016/j.cnsns.2010.12.027.

[2] BOWER J.M., BEEMAN D. *The Book of GENESIS, Exploring Realistic Neural Models with the GEneral NEural SImulation System,* Internet Edition, 2003.

[3] BRANCO T., CLARK B.A., HÄUSSER M. Dendritic Discrimination of Temporal Input Sequences in Cortical Neurons. *Science,* 2010, 329, pp. 1671–1675, doi: 10.1126/science.1189664.

[4] CARELLI P.V., REYES M.B., SARTORELLI J.C., PINTO R.D. Whole Cell Stochastic Model Reproduces the Irregularities Found in the Membrane Potential of Bursting Neurons. *Journal of Neurophysiology,* 2005, 94, pp.1169–1179, doi: 10.1152/jn.00070.2005.

[5] CASADO J.M. Synchronization of Two Hodgkin-Huxley Neurons Due to Internal Noise. *Physics Letters A,* 2003, 310, pp. 400–406, doi: 10.1016/S0375-9601(03)00387-6.

[6] DESTEXHE A., MAINEN Z.F., SEJNOWSKI T.J. Synthesis of Models for Excitable Membranes, Synaptic Transmission and Neuromodulation Using a Common Kinetic Formalism. *Journal of Computational Neuroscience,* 1994, 3(1), pp.195–230, doi: 10.1007/BF00961734.

[7] DŁUGOSZ R., KOLASA M., PEDRYCZ W., SZULC M. A Parallel Programmable Asynchronous Neighborhood Mechanism for Kohonen SOM Implemented in CMOS Technology, *IEEE Transactions on Neural Networks,* 2011, 22(12), pp. 2091–2104.

[8] DŁUGOSZ R., KOLASA M., SZULC M. An FPGA Implementation of the Asynchronous, Programmable Neighborhood Mechanism for WTM Self-Organizing Map. In: *Mixed Design of Integrated Circuits and Systems (MIXDES), 2011 Proceedings of the 18th International Conference,* Gliwice, pp. 258–263.

[9] DŁUGOSZ R., PEDRYCZ W. Łukasiewicz Fuzzy Logic Networks and Their Ultra Low Power Hardware Implementation, *Neurocomputing,* 2010, 73(7–9), pp. 1222–1234, doi:10.1016/j.neucom.2009.11.027.

[10] FELLER W. *An Introduction to Probability Theory and Its Applications.* John Wiley & Sons, 1968, ch. VII.

[11] FOX R.F. Stochastic Versions of the Hodgkin-Huxley Equations. *Biophysical Journal,* 1997, 72(5), pp. 2068–2074, doi: 10.1016/S0006-3495(97)78850-7.

[12] GERSTNER W., KISTLER W. *Spiking Neuron Models. Single Neurons, Populations, Plasticity.* Cambridge, 2000.

[13] GUGAŁA K., FIGAS A., JURKOWLANIEC A., RYBARCZYK A. Parallel Simulation of Stochastic Denritic Neurons using NVidia GPUs with CUDA C, *Elektronika – Konstrukcje, Technologie, Zastosowania,* 2011, 12, pp. 59–61.

[14] GUGAŁA K., ŚWIETLICKA A., BURDAJEWICZ M., RYBARCZYK A. Random Number Generation System Improving Simulations of Stochastic Models of Neural Cells, *Computing,* 2013, 95(1), pp. 259–275, doi: 10.1007/s00607-012-0267-z.

[15] HÄUSSER M., BARTLETT M. Dendrites: bug or feature? *Current Opinion in Neurobiology,* 2003, 13(3), pp. 372–383, doi: 10.1016/S0959-4388(03)00075-8.

[16] HODGKIN A.L., HUXLEY A.F. A Quantitative Description of Membrane Current and Its Application to Conduction and Excitation in Nerve. *Journal of Physiology,* 1952, 117, pp. 500–544, 10.1007/BF02459568.

[17] KINCAID D., CHENEY W. *Numerical Analysis: Mathematics of Scientific Computing,* The Brooks/Cole Series in Advanced Mathematics, 2002.

[18] KOCH C., SEGEV I. *Methods in Neuronal Modeling: from Ions to Networks.* Cambridge, Massachusetts: MIT Press, second edition, 1998.

[19] KOCH C. *Biophysics of Computation: Information Processing in Single Neurons.* Oxford University Press: New York, 1999.

[20] KOLASA M., DŁUGOSZ R., PEDRYCZ W., SZULC M. Programmable Triangular Neighborhood Function for Kohonen Self-Organizing Map Implemented on Chip, *Neural Networks,* 2012, 25, pp. 146–160, doi: 10.1016/j.neunet.2011.09.002.

[21] LI H., CAO Y., PETZOLD L.R., GILLESPIE D.T. Algorithms and Software for Stochastic Simulation of Biochemical Reacting Systems. *Biotechnology Progress,* 2008, 24(1), pp. 56–61, doi: 10.1021/bp070255h.

[22] MAINEN Z.F., SEJNOWSKI T.J. Influence of Dendritic Structure on Firing Pattern in Model Neocortical Neurons. *Nature,* 1996, 382, pp. 363–366, doi: 10.1038/382363a0.

[23] MEHROTRA K., MOHAN C., RANKA S. Bounds on the Number of Samples Needed for Neural Learning. *IEEE Transactions on Neural Networks,* 1991, 2(6), pp. 548–558, doi: 10.1109/72.97932.

[24] RAJA R., SAKTHIVEL R., ANTHONI S.M., KIM H. Stability of Impulsive Hopfield Neural Networks with Markovian Switching and Time–Varying Delays. *International Journal of Applied Mathematics and Computer Science,* 2011, 21(1), pp. 127–135, doi: 10.2478/v10006-011-0009-y.

[25] SAARINEN A., LINNE M.-L., YLI-HARJA O. Modeling Single Neuron Behavior Using Stochastic Differential Equations. *Neurocomputing,* 69(10–12), pp. 1091–1096, doi: 10.1016/j.neucom.2005.12.052.

[26] SCHNEIDMAN E., FREEDMAN B., SEGEV I. Ion Channel Stochasticity May Be Critical in Determining the Reliability and Precision of Spike Timing. *Neural Computation,* 1998, 10(7), pp. 1679–1703, doi: 10.1162/089976698300017089.

[27] SCHNEIDMAN E., SEGEV I., TISHBY N. Information Capacity and Robustness of Stochastic Neuron Models. *Advances in Neural Information Processing,* 2000, pp. 178–184.

[28] VETTER P., ROTH A., HÄUSSER M. Propagation of Action Potentials in Dendrites Depends on Dendritic Morphology. *Journal of Neurophysiology,* 2001, 85(2), pp. 926–937.

[29] VREEKEN J. *Spiking neural networks, an introduction.* Adaptive Intelligence Laboratory, Institute of Information and Computing Science, Utrecht University, 2003.

[30] WOLD K., TAN C.H. Analysis and Enchancement of Random Number Generator in FPGA Based on Oscillator Rings. *International Journal of Reconfigurable Computing,* 2009, doi: 10.1155/2009/501672.

[31] YANG X., CAO J., LU J. Synchronization of Randomly Coupled Neural Networks With Markovian Jumping and Time-Delay. *IEEE Transactions on Circuits and Systems I,* 2013, 60(2), pp. 363–376, doi: 10.1109/TCSI.2012.2215804.

[32] ZHU Q., CAO J., HAVAT T., ALSAADI F. Robust Stability of Markovian Jump Stochastic Neural Networks with Time Delays in the Leakage Terms. *Neural Processing Letters,* 2015, 41(1), pp. 1–27, doi: 10.1007/s11063-013-9331-8.