

Original Research Paper

Quantization of Map-Based Neuronal Model for Embedded Simulations of Neurobiological Networks in Real-Time

^{1,2}Nikolai F. Rulkov, ³Ariel Mark Hunt, ⁴Peter N. Rulkov and ⁵Andrey G. Maksimov

¹BioCircuits Institute, University of California, San Diego, La Jolla, USA

²DYNABIS, San Diego, USA

³Ariel Systems, Poway, USA

⁴California State University San Marcos, San Marcos, USA

⁵National Research University Higher School of Economics, Nizhny Novgorod, Russia

Article history

Received: 21-10-2016

Revised: 05-11-2016

Accepted: 07-11-2016

Corresponding Author:

Nikolai F. Rulkov

BioCircuits Institute, University of California, San Diego, La Jolla, USA

Email: nrulkov@uscd.edu

Abstract: The discrete-time (map-based) approach to modeling nonlinear dynamics of spiking and spiking-bursting activity of neurons has demonstrated its very high efficiency in simulations of neuro-biologically realistic behavior both in large-scale network models for brain activity studies and in real-time operation of Central Pattern Generator network models for biomimetic robotics. This paper studies the next step in improving the model computational efficiency that includes quantization of model variables and makes the network models suitable for embedded solutions. We modify a map-based neuron model to enable simulations using only integer arithmetic and demonstrate a significant reduction of computation time in an embedded system using readily available, inexpensive ARM Cortex L4 microprocessors.

Keywords: Map-Based Neuron Models, Quantization, Spiking-Bursting Activity, Embedded Solutions, Biomimetic Robotics, Neurobiological Networks

Introduction

The functionality of a brain and other neuronal systems of living organisms rely on the synaptic coupling organization among populations of neurons forming a function specific network of neurons with proper pathways of electrical activity in forms of short electrical spikes known as action potentials. The spikes generated in a neuron propagate along its axon towards synaptic inputs of the other neurons making the network achieve its functional goals.

Dynamical properties of spiking electrical activity of neurons, conditioned by the intrinsic properties of neurons or evoked in the response to external stimuli or other synaptic inputs, provide critical elements in shaping proper neuronal activity and, therefore, the functionality of neurobiological networks. This spiking activity propagating along the synaptic pathways is a key element of information processing in the brain, classification of sensory inputs, decision-making and motor control (Shepherd, 2004). Studies of the mechanisms behind the formation of network spatio-temporal patterns of spiking activity rely on neuronal models that can capture essential

characteristics of spiking activity for specific types of neurons (Dayan and Abbott, 2001).

Numerous recent studies focus on the development of artificial (engineering) systems that use elements of spiking activity to create a network that could perform some elements of brain functionality, sensory processing, motor control (Dura-Bernal *et al.*, 2014) and other functions. They mainly rely on the development and implementation of a spiking neuron model and synapses that replicate the dynamics of neurons as close to real biological neurons as possible that fits in computational hardware. Two main directions of these studies are the use of analog computing simulating the network with specially designed VLSI integrated circuits (Indiveri *et al.*, 2006; Silver *et al.*, 2007) and numerical simulation of the model with specially designed microchips, GPUs FPGAs or DSPs (Yavuz *et al.*, 2016; Zbrzeski *et al.*, 2016; Cheung *et al.*, 2016; Brette *et al.*, 2007 and references therein).

This paper focuses on the design, implementation and validation of a neuronal model that could be used in real-time simulations of relatively large networks using off-the-shelf 32-bit microprocessors. We base our design on the map-based approach to modeling spiking

neuronal activity in discrete times. Such models are very computationally efficient, realistically replicate the dynamics of spiking activity and have been successfully used in many studies of neuronal activity in large-scale cortical networks, olfactory systems and Central Pattern Generator (CPG) networks. The map-based model of swimming motor control of the CPG network of a Lamprey was implemented using a TI DSP and operated in real-time on a Lamprey-based robot (Westphal *et al.*, 2011).

Map-Based Modeling of Neuronal Activity

Consider a model of spiking and spiking-bursting activity designed in the form of a two-dimensional map and generating neuronal activity states at discrete moments of time. This model is very efficient in numerical simulation since its time step of consecutive iterations can be made to be on the order of spike duration and the dynamics of the spiking does not depend on the step size, which is usually fixed at 0.5 or 1.0 ms, depending on the network model. The equation of such a map can be written in the form proposed in (Rulkov, 2002):

$$\begin{aligned} x_{n+1} &= f_{\alpha}(x_n, y_n + \beta_n), \\ y_{n+1} &= y_n - \mu(1 + x_n - \sigma - \sigma_n) \end{aligned} \quad (1)$$

where, variables x_n and y_n are fast and slow variables describing the state of the neuron at discrete moments of time indicated by n . β_n and σ_n are input variables that model synaptic inputs and injected currents and other stimuli. The intrinsic dynamics of the map, Equation 1 is controlled by parameters μ , σ and α . The spiking in the model is due to the specific shape of the nonlinear function which can be written in the following form:

$$f_{\alpha}(x_n, u) = \begin{cases} \frac{\alpha}{(1-x_n)} + u, & \text{if } x_n < -0.5, \\ 1, & \text{if } -0.5 \leq x_n < 1, \\ -1, & \text{if } x_n \geq 1 \end{cases} \quad (2)$$

Note, Equation 2 slightly differs from the function proposed originally in (Rulkov, 2002). As shown in (Komarov *et al.*, 2016) it is more efficient for large-scale network simulations, simplifies the equation and improves the spike shape consistency.

Nonlinear Map of Spiking

Spike generation in model Equation 1 is provided by the nonlinear dynamics of fast variable x_n and can be

considered independently from the rest of the model using the following one-dimensional map:

$$x_{n+1} = f_{\alpha}(x_n, u) \quad (3)$$

where, u is treated as a control parameter. Figure 1 illustrates the mechanism of spike formation and positions of stable and unstable fixed points x_{sp} and x_{up} , respectively, indicated by green and blue filled dots.

By definition, fixed points, in a one-dimensional map, are located at the intersection of the first segment of the function Equation 2 with the diagonal ($x_{n+1} = x_n$), Fig. 1. Therefore, when the parameter u increases the fixed points move closer together, merge and then disappear when u crosses bifurcation value $u_{cr} = 1 - 2\sqrt{\alpha}$. This value is found using the fixed point equation $x^* = \alpha(1-x^*)^{-1} + u$, when it has a single solution (merged solution) in domain $x^* < -0.5$. Such bifurcation scenario of the fast subsystem fixed points is a key element of many simple models of spiking neurons, such as nonlinear integrate-and-fire and the Izhikevich model. It is important for our study to emphasize that near the bifurcation state the dynamics of the system is very sensitive to the resolution set for variable x_n because changes of the variable in that region of phase space can be arbitrarily small. Simulations with insufficient resolution can treat such small changes as zero and artificially stall the trajectory.

The one-dimensional map, Equation 3, demonstrates only the solutions in the form of steady state or tonic spiking and are typically used within the range of parameters $3 < \alpha < 6$ and $u_{cr}-0.3 < u < u_{cr}+0.2$. This is sufficient to mimic the dynamics of active sodium (Na+) and potassium (K+) pumps in generating action potentials (spikes). To expand the dynamics of this model to describe behavior of various types of neurons the parameter u is substituted with a variable or a set of variables that capture dynamics of relatively slow ionic channels of the neuronal membrane. An example of such a model is Equation 1, which has been used in many studies as a reduced model of a Regular Spiking Cell in a cortical network (Bazhenov *et al.*, 2008) and a Spiking-Bursting neuron in modeling CPG networks (Ayers and Rulkov, 2008).

The detailed parametric analysis of nonlinear dynamics of Equation 1 with $\beta_n = 0$ and $\sigma_n = 0$ can be found elsewhere (Shilnikov and Rulkov, 2003). An example of the spiking-bursting oscillations that occurs in the model without any external inputs is shown in Fig. 2. The phase portrait in Fig. 2a shows the trajectory (black dots connected with a line) and manifolds of slow motions (blue curves). The corresponding waveforms of fast x_n and slow y_n variables are shown in Fig. 2b and 2c.

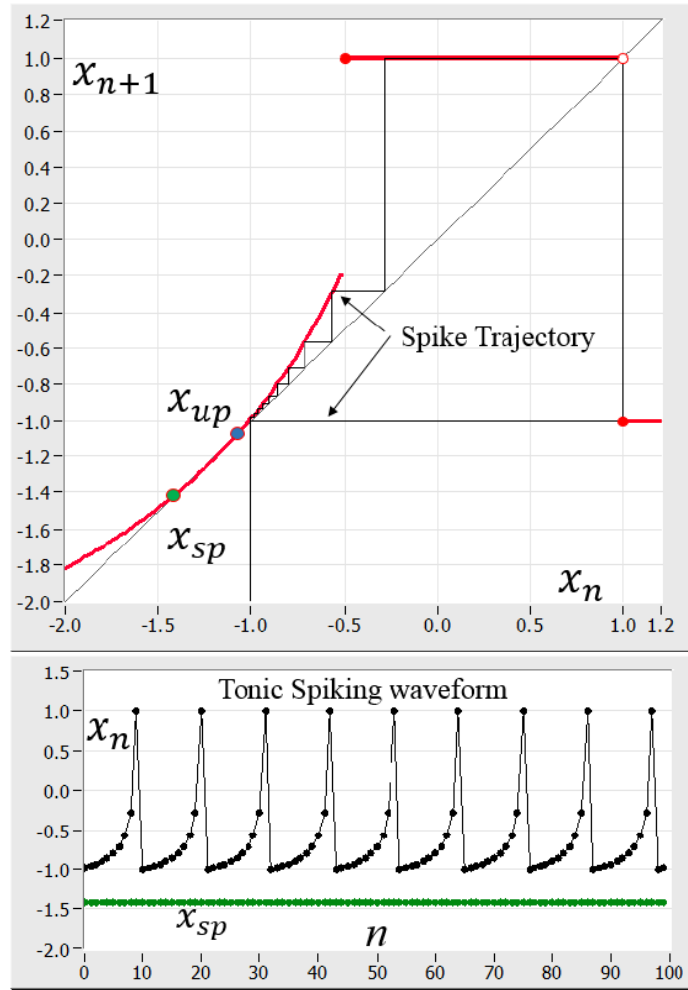


Fig. 1. Shape of nonlinear function, Equation 2 and an example of typical trajectories of the map Equation 3 computed with $\alpha = 5.0$ and $u = -3.484$ (Top panel). The waveforms of attracting sets in this case show spiking oscillations and fixed point x_{sp} (Bottom Panel)

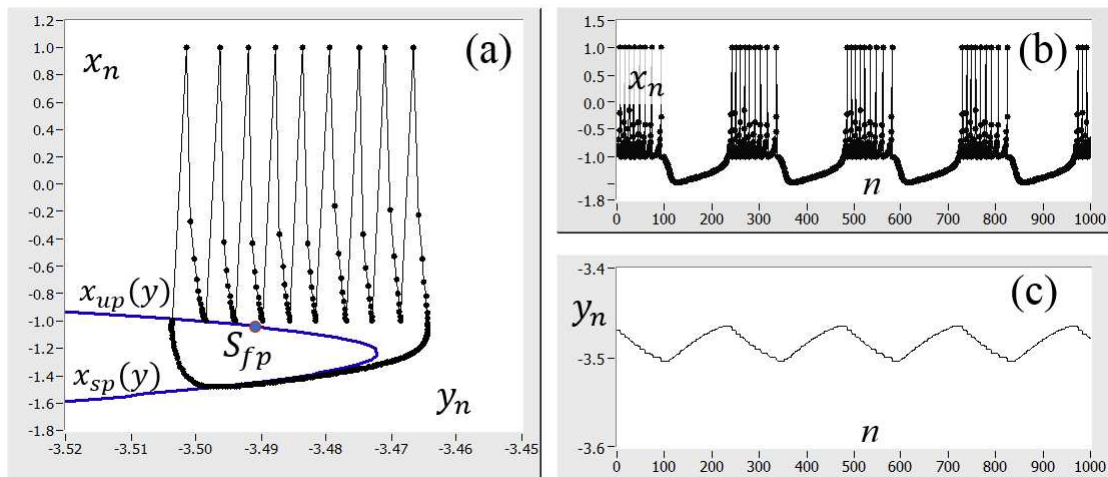


Fig. 2. An example of intrinsic spiking-bursting activity modeled with Equation 1 with parameter values $\alpha = 5.0$, $\mu = 0.001$, $\sigma = -0.04$. Trajectory in the phase plane (a) and corresponding waveforms of x_n (b) and y_n (c)

The response dynamics of the model is controlled with two input variables β_n and σ_n . The use of two input variables significantly expands the possibilities in shaping the desired response behavior of the models without adding new dynamical variables (Rulkov and Bazhenov, 2008).

Modeling of Response Spiking Behavior

In an experimental study, the type of the measured neuron is usually identified by analyzing its response characteristics to an electrical stimulus, such as a rectangular pulse of current, injected into the neuron through a microelectrode. In the model Equation 1 any such received stimulus is described by input variables β_n and σ_n , which can be written as a sum of all inputs:

$$\beta_n = \beta^D I_n^D + \beta^{syn} \sum_{N_{syn}} I_n^{syn},$$

$$\sigma_n = \sigma^D I_n^D + \sigma^{syn} \sum_{N_{syn}} I_n^{syn}$$

where, $\sum_{N_{syn}} I_n^{syn}$ is accumulated current corresponding to all synaptic inputs from the other neurons and N_{syn} is the

number of different types of synapses, I_n^D is any stimuli directly injected in to the soma and $\beta^D, \beta^{syn}, \sigma^D$ and σ^{syn} are weights used to tune the model to produce a desired response behavior to fit a particular type of neuron.

Examples of tuning this model to regular spiking cortical cells can be found in (Bazhenov *et al.*, 2005; Rulkov and Bazhenov, 2008). Effects of spike deceleration and after-hyperpolarization rebound bursts controlled by parameters β^D and σ^D are shown in Fig. 3, where $\alpha = 3.9, \mu = 0.001, \sigma = 0.0, \beta^D = 0.2, \sigma^D = 1.0$ and duration of the I_n^D pulse of 400 iterations that starts from initial state $I_n^D = 0$.

The mechanism of shaping response behavior can be examined by following the map trajectory in the phase portraits, Fig. 3, where blue curves $x_{sp}(y)$ and $x_{up}(y)$ correspond to the stable and unstable manifolds of slow motions, respectively. The green fixed point indicates the baseline state of the neuron while the red fixed point corresponds to the perturbed state during pulse action. The red dotted arrow indicates fast motion at pulse onset and solid red—a slow drift during pulse action. Similar behavior at the termination of the pulse is indicated with green arrows.

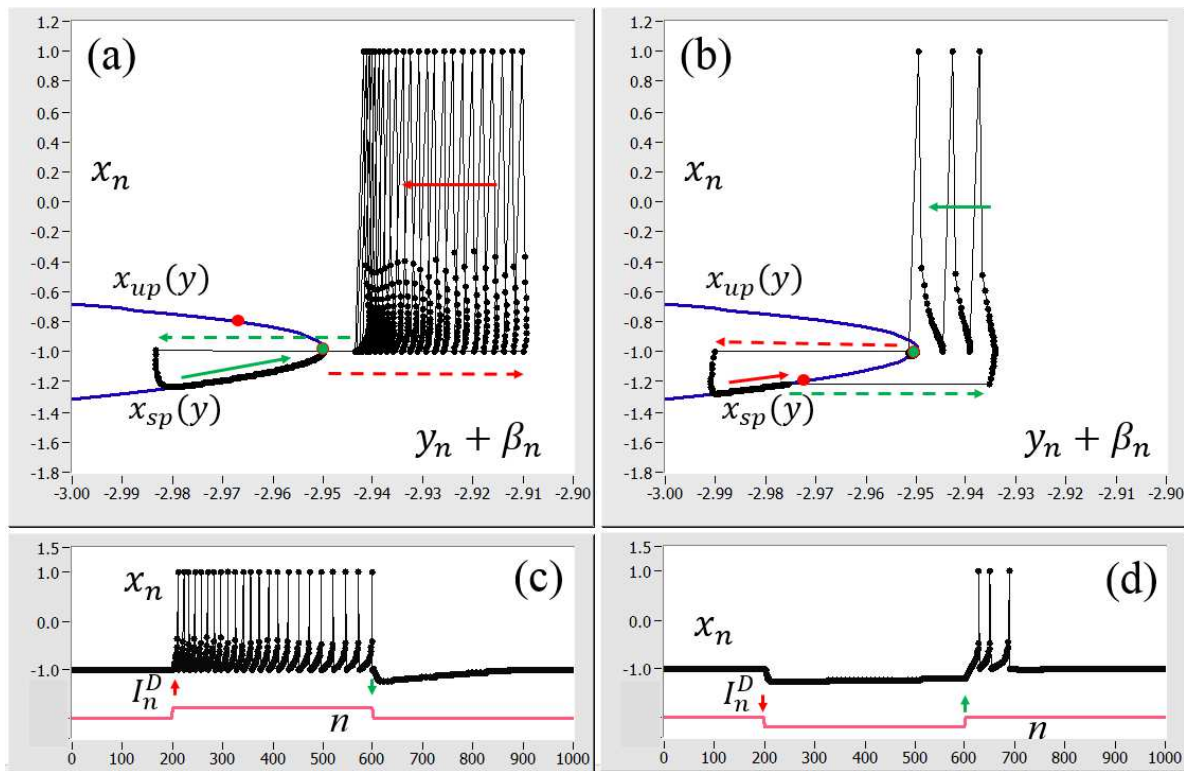


Fig. 3. Phase portraits (a), (b) and waveforms of spiking (c), (d) activity in response to the rectangular current pulse. The amplitude of I_n^D pulses in both cases is 0.2. Responses to the case of positive (depolarizing) and negative (hyperpolarizing) pulses are shown in the left and right panels, respectively

Dynamics of Synaptic Inputs

A spike generated by a neuron propagates towards the other neurons that receive and respond to it due to the synaptic connections located in the dendrite of the receiving (postsynaptic) neuron. There are electrical synapses, working as a conductor between neighboring neurons and chemical synapses where a spike propagating over an axon can reach dendrites of rather remote neurons forming a complex network structure. The dynamics and network organization of chemical synapses play a key role in the formation of neurobiological network functions. The basic elements of synapse dynamics is the modulation of membrane conductance at synapse contact evoked by a received presynaptic spike, which is characterized by elevation and relaxation times (Destexhe *et al.*, 1994). There are known several distinct types of synapses characterized by different timing rates and conductance. We adopt a model that captures the response to all presynaptic spikes of the same type of synaptic connections in the postsynaptic neuron using the following simple set of equations. The synaptic current I_n^{syn} computed using synapse conductance g_n and variable x_n of the postsynaptic neuron:

$$I_n^{syn} = g_n (x_n^{post} - x_{RP})$$

where, parameter x_{RP} defines reverse potential and, therefore, the synapse type (excitatory or inhibitory). The temporal modulation of conductance is modeled as the combination of two exponential curves:

$$g_n = e_n^{Relax} - e_n^{Rise}$$

which can be computed using a linear high pass filter of all received spikes trains:

$$\begin{aligned} e_{n+1}^{Rise} &= (1 - \delta_u^{syn}) e_n^{Rise} + \sum_{pre} W_{post_n}^{pre} S_n^{pre}, \\ e_{n+1}^{Relax} &= (1 - \delta_d^{syn}) e_n^{Relax} + \sum_{pre} W_{post_n}^{pre} S_n^{pre} \end{aligned} \quad (4)$$

Here parameters $0 < \delta_u^{syn}, \delta_d^{syn} < 1$ define the rise time and relaxation rate of the synapse, $W_{post_n}^{pre}$ defines the weight of the synaptic strength between two neurons and S_n^{pre} (the spike variable of the presynaptic neurons):

$$S_n^{pre} = \begin{cases} 1, & \text{if spike,} \\ 0, & \text{otherwise} \end{cases}$$

The weights $W_{post_n}^{pre}$ can change dynamically in time to model various types of synaptic plasticity. The models of synaptic plasticity are beyond of the scope of this paper.

Quantization of Model for Integer Computation

The main goal of this study is to modify the models of neurons and synapses, considered above, for computation using readily available inexpensive off-the-shelf hardware utilizing only integer operations and verify model performance. To allow such computation we need to rescale all variables and parameters to provide sufficient resolution that supports proper dynamical behavior of the model. The method of such scaling in the case of the Izhikevich model (Izhikevich, 2003) has been discussed in detail in (Jin *et al.*, 2008). In our case the map-based model can be rewritten as:

$$\begin{aligned} \bar{x}_{n+1} &= \bar{f}_{\bar{\alpha}} (\bar{x}_n, \bar{y}_n + \bar{\beta}_n + \bar{y}_{Sh}), \\ \bar{y}_{n+1} &= \bar{y}_u - \bar{\mu} (p_x + \bar{x}_n - \bar{\sigma} - \bar{\sigma}_n) \end{aligned} \quad (5)$$

where all variables and parameters are scaled integer numbers: $\bar{x}_n = p_x x_n$, $\bar{y}_n = p_y y_n - \bar{y}_{Sh}$, $\bar{y}_{Sh} = p_y y_{Sh}$, $\bar{\sigma} = p_x \sigma$, $\bar{\mu} = p_y \mu$, $\bar{\alpha} = p_x \alpha$ and the nonlinear function computed in integers as:

$$\bar{f}_{\bar{\alpha}} (\bar{x}_n, \bar{\mu}) = \begin{cases} \frac{\bar{\alpha} p_x}{(p_x - \bar{x}_n)} + \frac{p_x \bar{\mu}}{p_y}, & \text{if } \bar{x}_n < -\frac{p_x}{2}, \\ p_x & \text{if } -\frac{p_x}{2} \leq \bar{x}_n < p_x \\ -p_x & \text{if } \bar{x}_n \geq p_x \end{cases} \quad (6)$$

The values of scaling parameters p_x and p_y are selected to be as large as possible assuming that all computation still fits within 32-bit operations of the microprocessor. Note that in Equation 4 we shifted slow variable, y_n by the value $y_{Sh} = u_{cr} = 1 - 2\sqrt{\alpha}$ to remove its bias and enable a better scaling to deal with small parameter μ . The value y_{Sh} needs to be computed only at initialization time.

The scaled input variables $\bar{\sigma}_n$ and $\bar{\beta}_n$ are computed directly from the scaled values of currents. In the case of synaptic currents, change to integer values is done using scaled weights $\bar{W}_{post_n}^{pre} = p_{syn} W_{post_n}^{pre}$ and $\bar{W}_{post_n}^{pre} = p_{syn} W_{post_n}^{pre}$ in Equation 4 and representing timing parameters δ_u^{syn} and δ_d^{syn} as fractions $0 < \frac{p_u}{p_s} < 1$ and $0 < \frac{p_d}{p_s} < 1$, respectively. Spike variable S_n^{pre} takes only two integer values 0 or 1 and remains unchanged.

The equation for synaptic timing exponents Equation 4 can be rewritten in integer form as:

$$\bar{e}_{n+1}^{Rise} = \bar{e}_n^{Rise} - p_u \bar{e}_n^{Rise} / p_s + \sum_{pre} \bar{W}_{post_n}^{pre} S_n^{pre}$$

$$\bar{e}_{n+1}^{Relax} = \bar{e}_n^{Relax} - p_d \bar{e}_n^{Relax} / p_s + \sum_{pre} W_{post_n}^{pre} S_n^{pre}$$

where, in the case, when the second and the third terms of the equation are zero and the values of the corresponding exponent is above zero (for example $\bar{e}_n^{Rise} > 0$), the new value of the exponent should be reduced by 1 (e.g., $\bar{e}_{n+1}^{Rise} = \bar{e}_n^{Rise} - 1$). This will make sure that both exponents do not stick at some positive value due to finite resolution and always relax down to zero when presynaptic spikes are absent.

The scaled synaptic current \bar{I}_n^{syn} will be of the form $\bar{I}_n^{syn} = \bar{g}_n (\bar{x}_n^{post} - \bar{x}_{RP})$, where $\bar{g}_n = \bar{e}_n^{Relax} - \bar{e}_n^{Rise}$. Along with the other stimulus currents it forms the input variables of the model, Equation 5, as follows:

$$\bar{\beta}_n = \beta^D \bar{I}_n^D + \beta^{syn} \bar{I}_n^{syn}$$

$$\bar{\sigma}_n = \sigma^D \bar{I}_n^D + \sigma^{syn} \bar{I}_n^{syn}$$

where, parameters β^D , β^{syn} , σ^D and σ^{syn} are of the order of one or less and should be computed as a fraction.

Experimental Studies of Model Performance

To study the efficiency of models and demonstrate their computation in real time we built a simple computer using an STM32F427 microprocessor. We utilized an old empty Printed Circuit Board (PCB), which we had designed before for another project and

rewired it to fit our study of neuron simulations. The populated STM computer PCB with an array of 4 12-bit DACs, serial DATA link and power control we built for this study is shown in Fig. 4.

The STM32F427 was set to run with a 180MHz clock. It also supports 8 Analog Input Channels with internal 12-bit ADCs. The microprocessor workflow is designed to operate in one of two modes selected by a command switch in the code. In the first mode, we call it real-time continuous operation, the microprocessor runs all computations for each iteration within controlled 500µs time intervals taking input signals from Analog Inputs and showing the neuronal responses on the Analog Outputs. In this case only 4 selected variables are available at the Analog Outputs. The other “two state” outputs, such as spikes, are made available at digital output pins that can be used to control external actuators, generate motor activity controlled by a simulated CPG model or other tasks. The use of serial DATA in this case is very limited.

Figure 5 and 6 present such a study for three implementations of the same neuronal model using *double* Equation 1, *int32_t* Equation 5 and *float* Equation 1 types for variables and parameters.

We designated a Test Pin to measure time intervals used by the microprocessor to complete computation of the model codes, see Fig. 7. Since the computation can be affected by the implementation of the code for the model we present here the main elements of the test code for measuring timing for individual autonomous behavior of neurons.

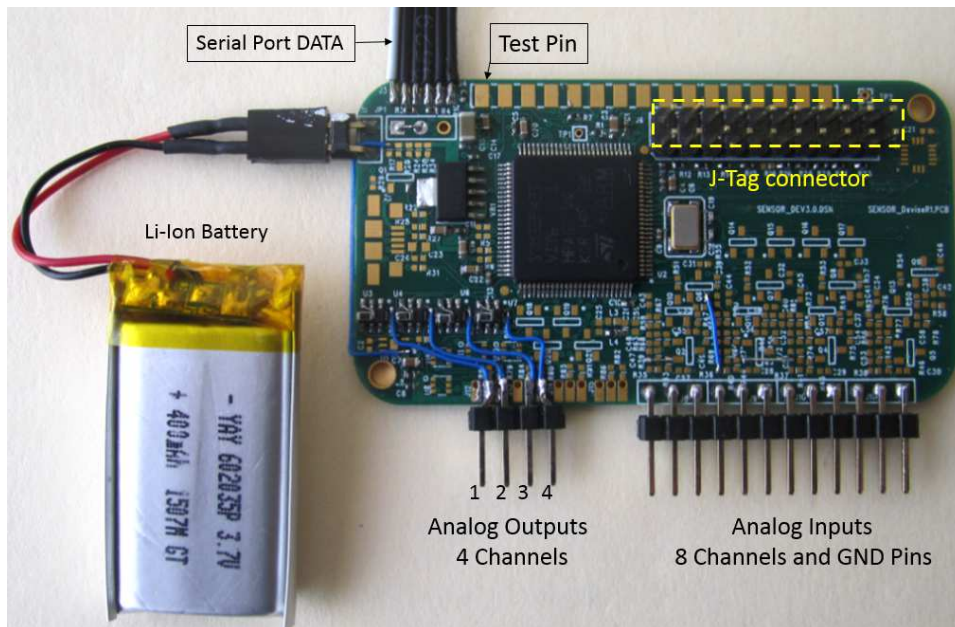


Fig. 4. Microprocessor PCB

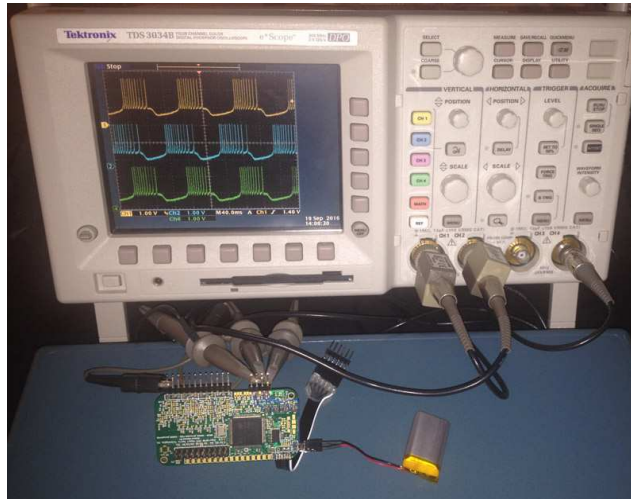


Fig. 5. Experimental setup for real-time operation analysis

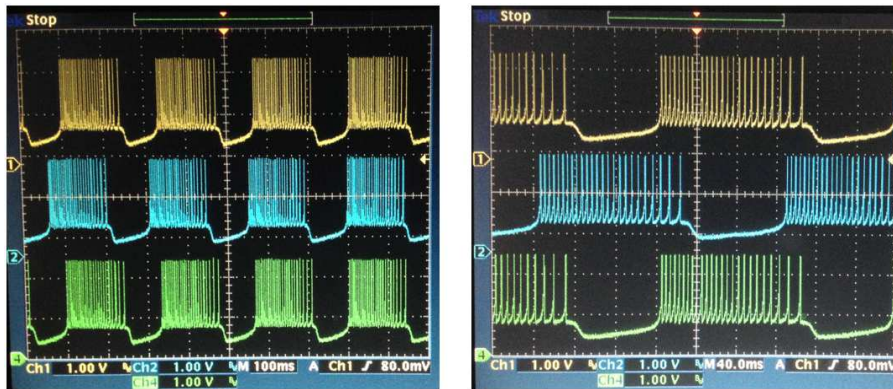


Fig. 6. Waveforms of spiking-bursting neuronal activity computed using STM32F427 with map-model implemented with *double* (yellow), *int32* (blue) and *float* (green) running in real-time mode and measured at the Analog Outputs, see elements of C code below. Spiking-Bursting oscillations shown at resolution 100ms per Dev. (Left panel) and a fragment at 40ms per Dev. (Right panel)

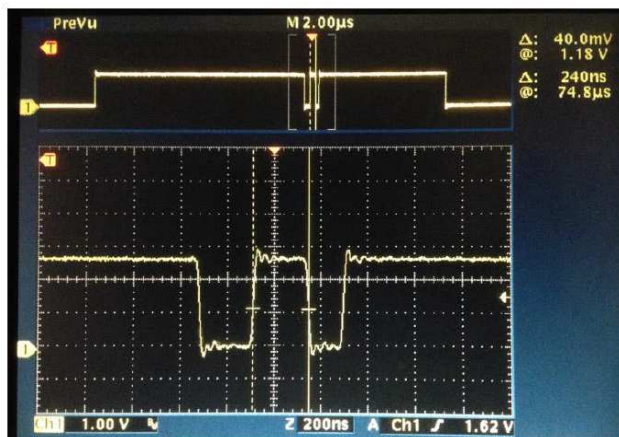


Fig. 7. Test Pin waveform indicating computation time (during High State, see Top Panel) used for one iteration of each model and a zoomed fragment showing the measurement of time for the case of the *int32* model (Bottom panel)

```
-----Time Test Code #1-----
double alpha= 5.0, sigma= 0.08, mu= 0.0005;
double xp1= -1, yp1= -3.4, x1= -1, y1= -3.4;
```

```
int32_t scl32 = 16384,
int32_t alpha2= alpha*scl32;
int32_t mu2= scl32/2000;
```

```

int32_t sigma2= sigma*scl32;
int32_t xp2= xp1*scl32;
int32_t yp2= yp1*scl32;
int32_t x2= x1*scl32;
int32_t y2= y1*scl32;

float alpha3= alpha;
float sigma3= sigma;
float mu3= mu;
float xp3= xp1;
float yp3= yp1;
float x3= x1;
float y3= y1;

while(1) { // Run simulations in continuous loop
    while(getTimer1Count() != 0); // wait for next interval
    SET_D00; // drive Pin(D00) HIGH
    dacWrite4(); //Send scaled data xp1,xp2,xp3 to DAC
    CLR_D00; // drive Pin(D00) LOW
    sampleADC(); // convert Analog Inputs to Data
    SET_D00; //drive Pin(D00) HIGH

    // do one step computation with double
    if(xp1 <= -0.5) x1= alpha / (1.0 - xp1) + yp1;
    if(xp1 > -0.5) x1= 1.0;
    if(xp1 >= 1.0) x1= -1.0;
    y1= yp1 - mu * (xp1 + 1 - sigma);
    xp1= x1;
    yp1= y1;

    CLR_D00; // drive Pin(D00) LOW
    delay();
    delay();
    SET_D00; //drive Pin(D00) HIGH

    // do one step computation with int32
    if(xp2 <= -scl32/2) x2= alpha2*scl32 / (scl32 - xp2) +
    yp2;
    if(xp2 > -scl32/2) x2= scl32;
    if(xp2 >= scl32) x2= -scl32;
    y2= yp2 - mu2 * (xp2 + scl32 - sigma2)/scl32;
    xp2= x2;
    yp2= y2;

    CLR_D00; // drive Pin(D00) LOW
    delay();
    delay();
    SET_D00; //drive Pin(D00) HIGH

    // do one step computation with float
    if(xp3 <= -0.5) x3= alpha3 / (1.0 - xp3) + yp3;
    if(xp3 > -0.5) x3= 1.0;
    if(xp3 >= 1.0) x3= -1.0;
    y3= yp3 - mu3 * (xp3 + 1 - sigma3);
    xp3= x3;
    yp3= y3;

    CLR_D00; // drive Pin(D00) LOW
} //-----

```

The results of this study of isolated models of neurons are summarized in Table 1.

Table 1. Computation time of isolated map-model of neuron

	Double	int32	Float
Time [μ S]	8.9	0.24	5.45

The second mode of operation supports detailed analysis of neuronal and synaptic dynamics in the network operations and allows one to complete the simulation of each iteration of the network and save all states of the model in memory. This may exceed the 500 μ s allowed for each iteration in real-time simulation. After the completion of the specified number of iterations all data can be uploaded to the PC via the Serial Port for detailed analysis of network dynamics.

Simulations of Map-Based Networks

To deal with simulations of large networks with various topologies of synaptic coupling we developed code that can be compiled for simulations on a PC or for real-time operation in the microprocessor using only a minor change in the parameter definitions. In the PC case, the network topology and all parameters are defined in a text file, which is read by the simulation program. In the case of embedded implementations, an additional program uses the same text file to generate a header file that is included in the compilation of the microprocessor based code. In both cases the network description is translated into data structures in the PC or microprocessor memory. The network simulation code represents models for neurons, synapses and interconnection topology and delays as structures. This provides convenient flexibility with minor computational overhead.

The synapse model includes code for spike propagation delays between pre and post-synaptic neurons. Each spike delay line is implemented with an *int32_t* variable and propagation is accomplished by a binary shift applied to the variable every iteration of the network. Time delay is achieved by reading the proper bit in the presynaptic delay variable as it is defined in the synapse connectivity of the postsynaptic neuron. A block diagram illustrating the structure of synapse simulation code in the case of two types of synapses, Excitatory (E) and Inhibitory (I), is presented in Fig. 8. To simplify the network diagram description the argument *n* of variable $x_i(n)$ represents the iteration step (time) and subscript *i*, the neuron index. The strengths of synaptic connections between neurons are defined using weights $W_{post_pre}^{synType}$ applied to received spikes in the summation block (in the text below we call it Presynaptic Inputs). Note that weights can be made to vary in time during simulations to model various mechanisms of synaptic plasticity.

Map-Based Network Model

As an example, that is useful for validating the dynamics and testing the performance of integer computation in an embedded microprocessor environment, we consider a simple network consisting of two neurons demonstrating the effects of spike adaptation and rebound bursts coupled with chemical inhibitory synapses as shown in Fig. 9.

The baseline state of each neuron is set to be in the silent condition. The injected rectangular current excites the first neuron, which starts spiking and inhibits the second neuron. The second neuron demonstrates the effect of a rebound burst after the inhibition from the first neuron is terminated. Then a pulse of injected current is applied to the second neuron and the process repeats in the opposite sequence.

An example of such simulations computed using floating point operations is presented in Fig. 10a and b. The simulations are done for the following set of parameters for both neuron models $\alpha = 3.7$, $\mu = 0.01$, $\sigma = -0.05$, $\beta^D = 0.4$, $\sigma^D = 1.0$ and $\beta^{syn} = 0.4$, $\sigma^{syn} = 1.0$, the inhibitory synapses are modeled with constant weights $w_{1,2} = 0.3$, $w_{2,1} = 0.2$, delays $d_{1,2} = 5$, $d_{2,1} = 3$ and both synapses have $\delta_u^{syn} = 0.1$, $\delta_d^{syn} = 0.2$ and $x_{RP} = -2.9$. The same network computed using scaled equations where all variables and parameters are 32-bit integers is presented in Fig. 10c and 10d. It uses scaling multipliers $p_x = p_y = 16384$ and $p_s = 1000$.

Panels Fig. 10a and 10c show response waveforms of the neuronal activity and panels b and d waveforms of the synaptic conductance activated by the presynaptic spikes. One can see that both simulations exhibit very similar dynamical behavior including very fine elements of neuronal excitability. Note that in both cases the first rebound burst spikes evoke rebound spikes in the opposite neuron. This indicates that the subthreshold dynamics of both neuronal models are the same, despite the difference in the computation method. At the same time in the case of the scaled model code for integer operations the computation time is about ten times more efficient than in the case of floating point including all the overhead components associated with the structure architecture of the code.

Measurements of computation time in the microprocessor for various sizes of network and connection topology performed for the cases of *float* and *int32_t* were used to derive approximate estimations of time needed for computation of each element of the simulated network when it contains 5 or more neurons. The results of this study are summarized in Table 2.

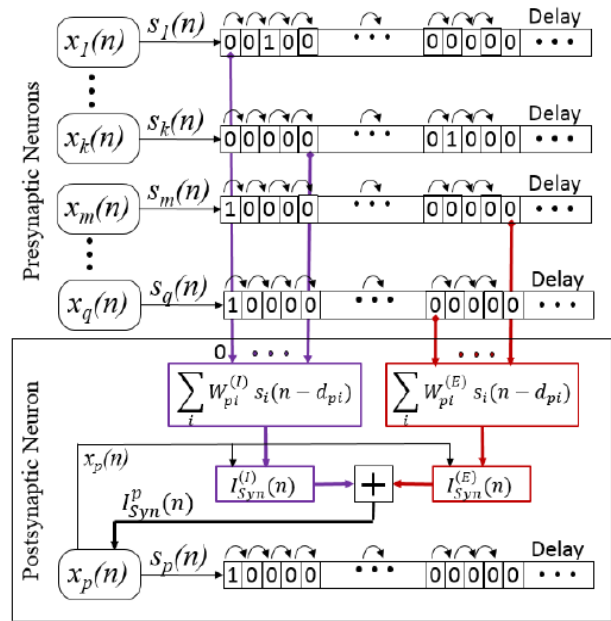


Fig. 8. Block diagram of synaptic model as implemented in the simulation code

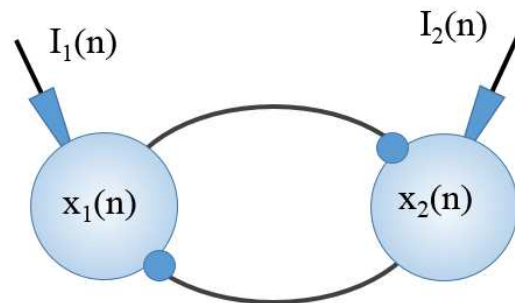


Fig. 9. Structure of reciprocally-inhibitory network excited with external pulse inputs $I_1(n)$ and $I_2(n)$

For example if the network has 50 neurons and each neuron has two types of synaptic inputs (e.g., excitatory and inhibitory) with 20 presynaptic inputs of each type, each time iteration of the network in the microprocessor will require $50 \cdot (1\mu s + 2 \cdot 0.7\mu s + 2 \cdot 20 \cdot 0.076\mu s) = 272\mu s$ for *int32_t* and $50 \cdot (10\mu s + 2 \cdot 4.4\mu s + 2 \cdot 20 \cdot 0.095\mu s) = 1130\mu s$ for *float*.

The results of computational time analysis demonstrate that quantization of a map-based model for dynamics of neurons and synaptic conductance give a significant improvement in simulation speed. The use of a linear model of synaptic conductance, Equation 4, enables one to sum presynaptic inputs with corresponding weights very efficiently in both methods of simulation, because the floating-point Multiply-And-Accumulate (MAC) loop is optimized in STM32F4xx microprocessors.

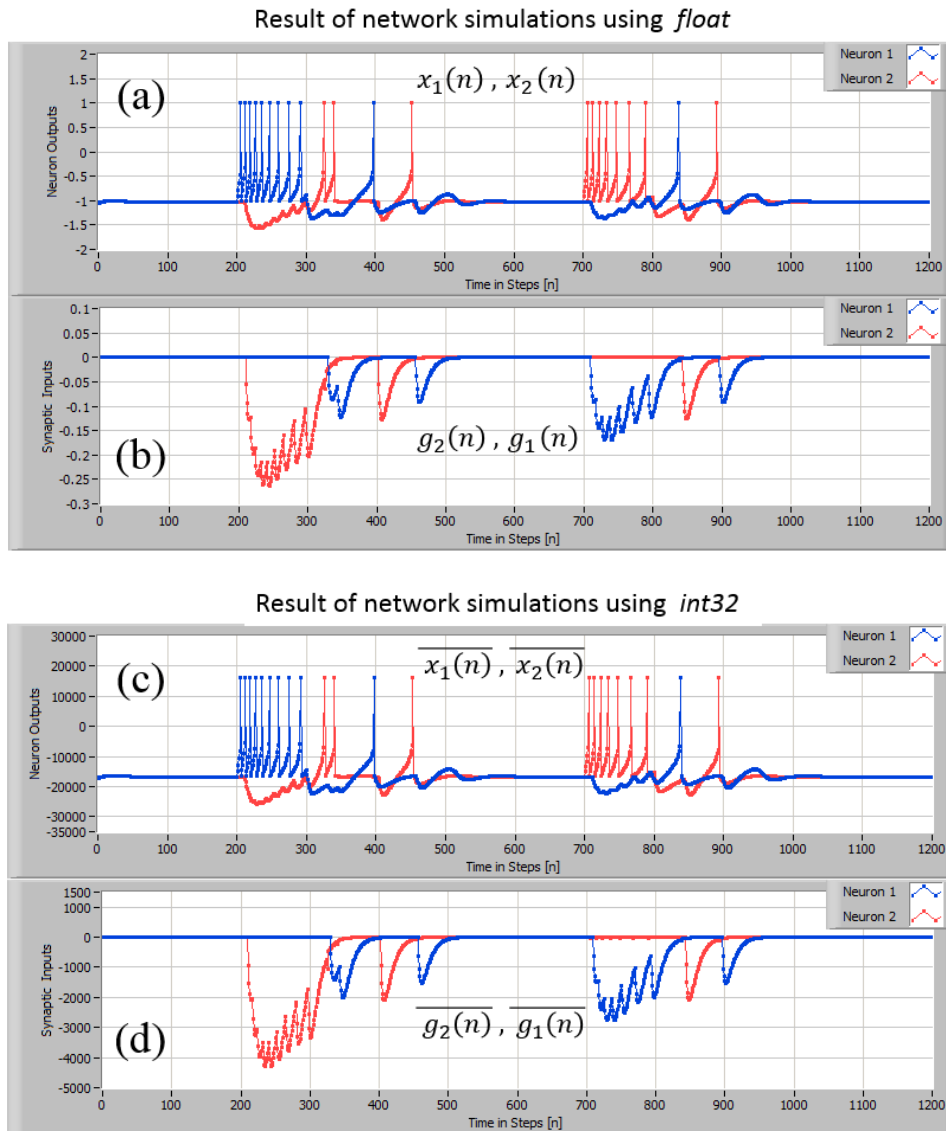


Fig. 10. Results of network simulations of response activity of two neurons coupled as shown in Fig. 9 and excited by injected currents I_n^D of duration 100 iterations (50ms) and amplitude $A_1 = 0.3$ initiated at $n = 200$ into the first neuron and $A_2 = 0.25$ at $n = 700$ into the second one

Table 2. Averaged computation time if each iteration for each component of the network in μs

	<i>int32_t</i>	<i>float</i>
Neuron	1.0	10
Synapse Type	0.7	4.4
Presynaptic Input	0.076	0.095

Conclusion

The presented results demonstrate that a transition from floating point to integer implementation of the map-based models can significantly speed up computation of large networks of neurons. In the

reported study we used neuronal and synaptic models scaled and coded using 32-bit integer variables and parameters. This increased the computational efficiency of the model Equation 1 and 2 about ten times and the overall network simulation, about five times. This improvement is very important and beneficial for real time implementation of neurobiological network models using low power off-the-shelf microprocessors. All computations presented in this study were done using an STM32 F427 running at a clock rate of 180MHz and consuming about 60mA at 4.0V. The size of the simulated network did not have a significant effect on the power consumption. This power consumption figure

can be reduced by either reducing the processor clock rate and/or placing the processor in low power mode after it completes an iteration and subsequently resuming full power at the start of the next step interval.

The presented results of quantization are focused on the use of 32-bit integers, where the map-model shows a very good behavioral match with the original model Equation 1 and 2. Further decrease of resolution may be possible and has been reported in similar types of models, see for example (Jin *et al.*, 2008). However, in our studies we observed that use of quantized models with resolution of 16-bits and below sometimes cause the onset of artificial instability and unexpected oscillations in sub threshold dynamics. The dynamical mechanisms and conditions of such distortions needs to be studied in more detail as the use of lower resolution could be beneficial for FPGA and other similar implementations.

The considered neuronal map-based models and the developed software and hardware implementation reveal a power efficient, readily available and inexpensive real-time engine for medium size neurobiological network simulations. These network models are good candidates for use in the design of wearable electronics for controllers in bioinspired prosthetic, sensors and robotic devices.

Acknowledgement

The theoretical studies of the map-based models and methods for improving computational efficiency are supported by ONR grant #N00014-16-1-2252. The authors are grateful to Donald Brady for permission to use Impact Sports Technology Inc. facilities and for providing access to R&D lab, equipment and the use of spare electronic parts for fabrication of the microprocessor PCB. Collaboration of NR with the MURI team working on large-scale neuronal network models for memory consolidation sponsored by ONR grant N000141310672 were very beneficial for inspiration of this study.

Author's Contributions

Nikolai F. Rulkov: Proposed a way to develop computationally efficient models of spiking-bursting activity using discrete phase space and time in generating nonlinear dynamics of neurons. He was involved in the development of the custom STM based computer, design of PCB with peripherals and prepared initial draft of the paper.

Ariel Mark Hunt: Developed firmware for the STM microprocessor, peripheral controls and data communication links. He developed the architecture for encoding the code responsible for loading the network into RAM data structures and their subsequent processing. Worked on the manuscript revisions.

Peter N. Rulkov: Assisted in fabrication of custom STM computer and development of embedded code for model simulations. He carried out the experiments with the STM microprocessor, performed measurements of computational time in real-time simulations. Collected data for analysis of network dynamics and worked on the manuscript revisions.

Andrey G. Maksimov: Worked on the analysis of scaling for variables and parameters using Matlab simulations. Carried out comparative analysis of bifurcations in integer and float cases on various stages of model verification. Revised the draft of the manuscript.

Ethics

This paper is original and includes unpublished materials. The corresponding author confirms that all other authors have read and approved the manuscript.

References

- Ayers, J. and N. Rulkov, 2008 Controlling Biomimetic Underwater Robots with Electronic Nervous Systems. In: Bio-Mechanisms of Swimming and Flying: Fluid Dynamics, Biomimetic Robots and Sports Science, Kato, N. and S. Kamimura (Eds.), Springer Science and Business Media, Berlin, ISBN-10: 4431733809, pp: 295-307.
- Bazhenov, M., N.F. Rulkov, J.M. Fellous and I. Timofeev, 2005. Role of network dynamics in shaping spike timing reliability. *Phys. Rev. E*, 72: 041903-041903. DOI: 10.1103/PhysRevE.72.041903
- Bazhenov, M., N.F. Rulkov and I. Timofeev, 2008. Effect of synaptic connectivity on long-range synchronization of fast cortical oscillations. *J. Neurophysiol.*, 100: 1562-1575. DOI: 10.1152/jn.90613.2008
- Brette, R., M. Rudolph, T. Carnevale, M. Hines and D. Beeman *et al.*, 2007. Simulation of networks of spiking neurons: A review of tools and strategies. *J. Comput. Neurosci.*, 23: 349-398. DOI: 10.1007/s10827-007-0038-6
- Cheung, K., S.R. Schultz and W. Luk, 2016. NeuroFlow: A general purpose spiking neural network simulation platform using customizable processors. *Front. Neurosci.*, 9: 516-516. DOI: 10.3389/fnins.2015.00516
- Dayan, P. and L.F. Abbott, 2001. *Theoretical Neuroscience: Computational and Mathematical Modeling of Neural Systems*. 1st Edn., MIT Press, Cambridge, ISBN-10: 0262041995, pp: 460.
- Destexhe, A., Z.E. Mainen and T.J. Sejnowski, 1994. An efficient method for computing synaptic conductances based on a kinetic model of receptor binding. *Neuronal Comput.*, 6: 14-18. DOI: 10.1162/neco.1994.6.1.14
- Dura-Bernal, S., G.L. Chadderton, S.A. Neymotin, J.T. Francis and W.W. Lytton, 2014. Towards a real-time interface between a biomimetic model of sensorimotor cortex and a robotic arm. *Patt. Recognit. Lett.*, 36: 204-212. DOI: 10.1016/j.patrec.2013.05.019

- Indiveri, G., E. Chicca and R. Douglas, 2006. A VLSI array of low-power spiking neurons and bistable synapses with spike-timing dependent plasticity. *IEEE Trans. Neural Netw.*, 17: 211-221.
DOI: 10.1109/TNN.2005.860850
- Izhikevich, E.M., 2003. Simple model of spiking neurons. *IEEE Trans. Neural Netw.*, 14: 1569-1572.
DOI: 10.1109/TNN.2003.820440
- Jin, X., S.B. Furber and J.V. Woods, 2008. Efficient modelling of spiking neural networks on a scalable chip multiprocessor. *Proceedings of the IEEE International Joint Conference on Neural Networks*, Jun. 1-8, IEEE Xplore Press, pp: 2812-2819.
DOI: 10.1109/IJCNN.2008.4634194
- Komarov, M., G. Krishnan, S. Chauvette, N. Rulkov and I. Timofeev *et al.*, 2016. Large-scale cortical network dynamics determine sequence replay during slow-wave sleep.
- Rulkov, N.F., 2002. Modeling of spiking-bursting neural behavior using two-dimensional map. *Phys. Rev. E*, 65: 041922-041922.
DOI: 10.1103/PhysRevE.65.041922
- Rulkov, N.F. and M. Bazhenov, 2008. Oscillations and synchrony in large-scale cortical network models. *J. Biol. Phys.*, 34: 279-299.
DOI: 10.1007/s10867-008-9079-y
- Shepherd, M.G., 2004. *The Synaptic Organization of the Brain*. 5th Edn., Oxford University Press, USA, New York, ISBN-10: 019515956X, pp: 719.
- Silver, R., K. Boahen, S. Grillner, N. Kopell and K.L. Olsen, 2007. Neurotech for neuroscience: Unifying concepts, organizing principles and emerging tools. *J. Neurosci.*, 27: 11807-11819.
DOI: 10.1523/JNEUROSCI.3575-07.2007
- Shilnikov, A.L. and N.F. Rulkov, 2003. Origin of chaos in a two-dimensional map modeling spiking-bursting neural activity. *Int. J. Bifurcat. Chaos*, 13: 3325-3340. DOI: 10.1142/S0218127403008521
- Westphal, A., N.F. Rulkov, J. Ayers, D. Brady and M. Hunt, 2011. Controlling a lamprey-based robot with an electronic nervous system. *Smart Structures Syst.*, 8: 39-52.
- Yavuz, E., J. Turner and T. Nowotny, 2016. GeNN: A code generation framework for accelerated brain simulations. *Sci. Reports*, 6: 18854-18854.
DOI: 10.1038/srep18854
- Zbrzeski, A., Y. Bornat, B. Hillen, R. Siu and J. Abbas *et al.*, 2016. Bio-inspired controller on an FPGA applied to closed-loop diaphragmatic stimulation. *Front. Neurosci.*, 10: 275-275.
DOI: 10.3389/fnins.2016.00275