frontiers | Frontiers in Neuroscience

TYPE Original Research
PUBLISHED 09 January 2023
DOI 10.3389/fnins.2022.1069133

OPEN ACCESS

EDITED BY
Yiran Chen,
Duke University, United States

REVIEWED BY
Lei Deng,
Tsinghua University, China
Ruben A. Tikidji-Hamburyan,
George Washington University,
United States

*CORRESPONDENCE
Takuya Nanami
nanami@iis.u-tokyo.ac.jp

SPECIALTY SECTION
This article was submitted to Neuromorphic Engineering, a section of the journal Frontiers in Neuroscience

RECEIVED 13 October 2022
ACCEPTED 17 November 2022
PUBLISHED 09 January 2023

CITATION
Nanami T and Kohno T (2023) Piecewise quadratic neuron model: A tool for close-to-biology spiking neuronal network simulation on dedicated hardware. *Front. Neurosci.* 16:1069133. doi: 10.3389/fnins.2022.1069133

Piecewise quadratic neuron model: A tool for close-to-biology spiking neuronal network simulation on dedicated hardware

Takuya Nanami* and Takashi Kohno

Institute of Industrial Science, The University of Tokyo, Tokyo, Japan

Spiking neuron models simulate neuronal activities and allow us to analyze and reproduce the information processing of the nervous system. However, ionic-conductance models, which can faithfully reproduce neuronal activities, require a huge computational cost, while integral-firing models, which are computationally inexpensive, have some difficulties in reproducing neuronal activities. Here we propose a Piecewise Quadratic Neuron (PQN) model based on a qualitative modeling approach that aims to reproduce only the key dynamics behind neuronal activities. We demonstrate that PQN models can accurately reproduce the responses of ionic-conductance models of major neuronal classes to stimulus inputs of various magnitudes. In addition, the PQN model is designed to support the efficient implementation on digital arithmetic circuits for use as silicon neurons, and we confirm that the PQN model consumes much fewer circuit resources than the ionic-conductance models. This model intends to serve as a tool for building a large-scale closer-to-biology spiking neural network.

KEYWORDS

silicon neuronal network, silicon neuron, spiking neuron model, FPGA, PQN model

1. Introduction

Spiking neural network (SNN) models can reproduce the information processing of the nervous system at the cellular and synaptic level, enabling us to analyze and understand the brain's information processing and to realize the brain's intelligence for engineering applications. While large-scale simulations of SNN models using general-purpose computers require extensive computational facilities and are slow in simulation speed, the silicon neuronal networks (SiNNs), which are electronic circuit systems specifically designed to simulate SNN models, enable highly power-efficient and high-speed simulation. A main element of SiNNs is the silicon neuron that simulates neuronal activities by using the spiking neuron model. Silicon neurons using analog circuits (Schemmel et al., 2010; Arthur and Boahen, 2011; Grassia et al., 2011; Brink et al., 2013; Qiao et al., 2015; Kohno and Aihara, 2016; Moradi et al., 2018; Rubino et al., 2021; Schemmel et al., 2021) show very high power efficiency, but have technical hurdles including

fabrication mismatch and temperature dependence. In contrast, silicon neurons using digital circuits are far less sensitive to these factors. Although they tend to consume higher power than analog silicon neurons, they are more portable, easy-to-operate, and highly integratable. Large-scale networks have been built using digital application-specific integrated circuits (ASICs) (Merolla et al., 2014; Davies et al., 2018). In addition, field-programmable gate arrays (FPGAs) have also been used (Thomas and Luk, 2009; Cassidy et al., 2011, 2013; Li et al., 2012; Khoyratee et al., 2019) due to their ease of implementation and low cost.

In both SNN models and SiNNs, a variety of spiking neuron models have been used due to the trade-off between the reproducibility of neuronal activities and computational efficiency. The ionic-conductance neuron models (Hodgkin and Huxley, 1952; Plant, 1981; Wang, 1993; SCHUTTER and BOWER, 1994; Pospischil et al., 2008) describe neuronal dynamics at the level of each ion channel and can replicate arbitrary electrophysiological properties. The nervous system contains a wide variety of neuronal classes, each with different electrophysiological properties, which are reported (Cunningham et al., 2004; Benda et al., 2005; Grillner, 2007) to play an essential role in information processing. Therefore ionic-conductance models are used in SNN models (Markram et al., 2015; Bezaire et al., 2016; Ecker et al., 2020) that attempt to fully replicate certain brain regions. However, they consume enormous resources in circuit implementation, making their application to large-scale SiNNs difficult. In contrast, the Leaky integrate-and-fire (LIF) model is one of the simplest spiking neuron models and reproduces only simplified features such as the perturbation by stimulus inputs over time and convergence to the resting potential by the leak current. The LIF model approximates the neuronal spiking process by resetting the state variables and is therefore very computationally inexpensive. It is used for SiNNs (Cassidy et al., 2011; Merolla et al., 2014; Davies et al., 2018) that intend to realize large-scale networks with limited power consumption. In addition, the Izhikevich (IZH) model (Izhikevich, 2003) and the adaptive exponential I&F (AdEx) model (Brette and Gerstner, 2005), which are extensions of the LIF model, similarly approximate the spiking process by resetting the state variables. They have also been used in many SNN models (Thomas and Luk, 2009; Schemmel et al., 2010; Cassidy et al., 2013; Qiao et al., 2015; Moradi et al., 2018; Rubino et al., 2021; Schemmel et al., 2021) because they cover several neuronal classes, although they require a little more computational cost than the LIF model due to their squared and exponential term. In these integrate-and-fire-based (I&F-based) models, resetting the state variables reduces their computational cost but also compromises the reproducibility of neuronal activities. For example, it is known that the amplitude of spikes is gradedly dependent on the stimulus in some regions in the nervous system (Alle and Geiger, 2006), which is referred to as graded response. It was

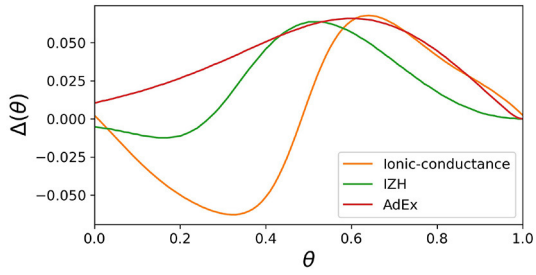

FIGURE 1
Phase-resetting curve of spiking neuron models. The horizontal axis represents the phase at which the pulse stimulus was given, and the vertical axis represents how much the phase was shifted by the pulse stimulus.

shown theoretically (Rinzel and Ermentrout, 1998) that the Class II in the Hodgkin's classification (Hodgkin, 1948) can exhibit the graded response. However, the I&F-based models, even in their Class II modes, cannot realize the graded response because it assumes the spike as a stereotyped event. Another example is the phase-resetting curve (PRC) (Rinzel and Ermentrout, 1998), which represents how the phase of a neuron in an oscillatory spiking state changes depending on the timing of the pulse stimulus. The Class II modes of the IZH and AdEx models reproduce the discontinuous frequency-current curve, but the shape of their PRC is not the standard Type II, which is typical for the Class II (Figure 1). Because neurons with the Type II PRC tend to promote synchronous firing (Hansel et al., 1995), it might be important to model the synchronous activities in the brain. As shown in Figure 1, the Class II mode of the ionic-conductance model has a biphasic (Type II) PRC, while the PRC of the AdEx model is monophasic. The Class II mode of the IZH model has a very small negative part and $\Delta(\theta)$ is less than 0 when $\theta = 0$. Furthermore, it was reported (Nanami and Kohno, 2016) that the responses of the intrinsically bursting (IB) class of the IZH model to step stimuli are different from those of a typical ionic-conductance model.

A qualitative modeling approach has a long history that starts with the FitzHugh–Nagumo model (FitzHugh, 1961; Nagumo et al., 1962) and was theoretically established by the theory of nonlinear dynamics (Kepler et al., 1992; Rinzel and Ermentrout, 1998; Izhikevich, 2007). The qualitative neuron models (FitzHugh, 1961; Nagumo et al., 1962; Hindmarsh and Rose, 1984; Nanami and Kohno, 2016) reproduce only the key mathematical structures behind neuronal activities. Although they have much less computational cost than the ionic-conductance models, they can faithfully reproduce the dynamics of the spiking process without resetting the state

variables. In the FitzHugh–Nagumo model and Hindmarsh–Rose model (Hindmarsh and Rose, 1984), the right-hand sides of the differential equations for each state variable are represented by polynomials of degree 3 or less. The digital spiking silicon neuron (DSSN) model (Nanami and Kohno, 2016) employs quadratic terms with piecewise and step functions to achieve qualitatively equivalent dynamics, and this approximation enables efficient implementation in software and digital arithmetic circuits. In addition, while preceding qualitative models represent a single neuron class, the DSSN model is designed to represent a wide variety of classes by tuning its parameters. The DSSN model covers the regular spiking (RS), fast spiking (FS), low-threshold spiking (LTS), IB, elliptic bursting (EB), and parabolic bursting (PB) classes. In the FS mode, both the Class I and Class II in Hodgkin's classification are supported. Recently, a general framework of qualitative models described by piece-wise polynomial formula has been proposed (Tikidji-Hamburyan and Colonnese, 2021). This framework, polynomial, piecewise-linear, step (PLS)-framework, provides a generalized method of constructing, describing, and discussing these models. In this framework, the right-hand side of the differential equation is expressed as any combination of polynomial, piecewise-linear, and step functions.

In this study, we propose a Piecewise Quadratic Neuron (PQN) model, a refined version of the DSSN model. Though the DSSN model supports various neuronal classes, its ability to reproduce their dynamical activity for a wide range of stimulus intensity is limited. In addition, a number of parameters have to be fitted to reproduce a wide variety of neuronal dynamics, which makes the fitting difficult. Thus, a scaling parameter for the stimulus input is inserted and six parameters are omitted in the PQN model by assuming that the nullcline of each state variable is continuous and smooth. Since the cellular dynamics corresponding to these nullclines are expected to be continuous and smooth, this assumption is biologically natural. The reduction in the number of parameters leads to a drastic reduction in the explore space in the parameter tuning, facilitating better parameter sets. First, it is demonstrated that the PQN model can reproduce the graded response and the Type II PRC typically seen in the Class II neurons in the Hodgkin's classification. Next, it is shown that the PQN model reproduces six major neuronal classes, RS, FS, LTS, IB, EP, and PB. Here, we confirm that the PQN model can reproduce the activity of ionic-conductance models of six neuronal classes in response to step inputs of various magnitudes. Although these neuron classes were also supported by the DSSN model, their responses to step inputs with a wide range of intensity are not completely reproduced. In addition, we show that the PQN model consumes much fewer circuit resources than the ionic-conductance model in the FPGA implementation due to its simple equations fitted to the digital arithmetic circuits. Moreover, we provide the python codes for the software simulation and VHDL codes for hardware implementation and they are available at the following links (PQN_pya[1]; PQN_vhdl[2]). Note that the model must be computed using the fixed-point operation with the specified bit-width, 28 bits for Class II and 18 bits otherwise, and the Euler's method with the specified time-step width, 1 ms for PB and 0.1 ms otherwise. In Methods section, in addition to the original form of the PQN model, equations using the PLS-framework are also presented for a more comprehensive perspective. This study intends to provide an easy-to-use tool for constructing large-scale and close-to-biology SNN models and SiNNs.

The remainder of this paper is organized as follows: Section 2 describes the details of the model, Section 3 shows the simulation results and comparison, and Section 4 provides a summary and future perspectives.

2. Methods

2.1. Equations of the PQN model

In order to efficiently reproduce a variety of neuronal classes, each with different dynamics, the PQN model has four variations. The first one is the two-variable PQN model, which is the basic form of the other variations. It supports the Class II mode in the Hodgkin's classification. Its equations are as follows:

$$\frac{dv}{dt} = \frac{\phi}{\tau}(f(v) - n + I_0 + k\,I_{\text{stim}}), \tag{1}$$

$$\frac{dn}{dt} = \frac{1}{\tau}(g(v) - n), \tag{2}$$

$$f(v) = \begin{cases} a_{fn}(v - b_{fn})^2 + c_{fn} & (v < 0) \\ a_{fp}(v - b_{fp})^2 + c_{fp} & (v \geq 0), \end{cases} \tag{3}$$

$$g(v) = \begin{cases} a_{gn}(v - b_{gn})^2 + c_{gn} & (v < r_g) \\ a_{gp}(v - b_{gp})^2 + c_{gp} & (v \geq r_g), \end{cases} \tag{4}$$

$$b_{fp} = \frac{a_{fn}b_{fn}}{a_{fp}}, \tag{5}$$

$$c_{fp} = a_{fn}b_{fn}^2 + c_{fn} - a_{fp}b_{fp}^2, \tag{6}$$

$$b_{gp} = r_g - \frac{a_{gn}(r_g - b_{gn})}{a_{gp}}, \tag{7}$$

$$c_{gp} = a_{gn}(r_g - b_{gn})^2 + c_{gn} - a_{gp}(r_g - b_{gp})^2, \tag{8}$$

1 https://github.com/tnanami/PQN_py

2 https://github.com/tnanami/PQN_vhdl

where v and n represent the membrane potential and recovery variable, respectively. Parameter I_0 is a bias constant. I_{stim} is the stimulus input and k is its scaling parameter. The parameters τ and ϕ determine the time constants of the variables. The parameters r_g, a_x, b_x, and c_x, where x is fn, fp, gn, or gp, are constants that determine the nullclines of the variables. Constants b_{fp}, c_{fp}, b_{gp}, and c_{gp}, are determined by other parameters such that the nullclines are continuous and smooth. All of the variables and parameters are purely abstract with no physical units. The differences from the DSSN model are that the four constants, b_{fp}, c_{fp}, b_{gp}, and c_{gp}, are determined by other parameters to ensure continuity and smoothness, and that the parameter k, which scales the stimulus input, is inserted. The ionic-conductance models generally require three or more state variables for the calculation of the spiking process, and the differential equations include a quartic term and exponential functions (Hodgkin and Huxley, 1952; Pospischil et al., 2008). On the other hand, the PQN model requires only two variables with quadratic terms and can be implemented with less computational cost in software and hardware implementations.

The second one is the three-variable PQN model. A new variable q is introduced to support the RS, FS, and EB classes. Its equations are given by

$$\frac{dv}{dt} = \frac{\phi}{\tau}(f(v) - n - q + I_0 + k\,I_{\text{stim}}), \tag{9}$$

$$\frac{dn}{dt} = \frac{1}{\tau}(g(v) - n), \tag{10}$$

$$\frac{dq}{dt} = \frac{\epsilon_q}{\tau}(h(v) - q), \tag{11}$$

$$h(v) = \begin{cases} a_{hn}(v - b_{hn})^2 + c_{hn} & (v < r_h) \\ a_{hp}(v - b_{hp})^2 + c_{hp} & (v \geq r_h), \end{cases} \tag{12}$$

$$b_{hp} = r_h - \frac{a_{hn}(r_h - b_{hn})}{a_{hp}}, \tag{13}$$

$$c_{hp} = a_{hn}(r_h - b_{hn})^2 + c_{hn} - a_{hp}(r_h - b_{hp})^2, \tag{14}$$

where q represents the slow variable. The parameter ϵ_q determine its time constants and parameters r_h, a_x, b_x, and c_x, where x is hn or hp, determine its nullcline. Constants b_{hp} and c_{hp} are determined by other parameters such that the nullcline is continuous and smooth. Note that Equation (9) is the same as Equation (1) except that the new variable q is inserted. Equation (10) is the same as Equation (2). The difference from the DSSN model is that the constants b_{fp}, c_{fp}, b_{gp}, c_{gp}, b_{hp}, and c_{hp} are defined by other parameters and k is inserted.

The third one is the four-variable PQN model. Another variable u is introduced to support the PB class. Its equations are

$$\frac{dv}{dt} = \frac{\phi}{\tau}(f(v) - n - q + u + I_0 + k\,I_{\text{stim}}), \tag{15}$$

$$\frac{dn}{dt} = \frac{1}{\tau}(g(v) - n), \tag{16}$$

$$\frac{dq}{dt} = \frac{\epsilon_q}{\tau}(h(v) - q), \tag{17}$$

$$\frac{du}{dt} = \frac{\epsilon_u}{\tau}(v + v_0 - \alpha_u u), \tag{18}$$

where u is the slow variable and the slow subsystem consisting of u and q is capable of producing oscillations. The parameter ϵ_u determines the time constant of u. The parameters v_0 and α_u determine the nullcline of u, respectively. Note that Equation (15) is the same as Equation (9) except that variable u is incorporated. Equations (16) and (17) are shared with the three-variable PQN model. The differences from the DSSN model are exactly the same as those in the three-variable mode.

The final variation is the extended four-variable PQN model, which supports the LTS and IB classes. The only difference to the four-variable PQN model is that the time constant of n is dependent on u. Its equations are given by

$$\frac{dv}{dt} = \frac{\phi}{\tau}(f(v) - n - q + I_0 + k\,I_{\text{stim}}), \tag{19}$$

$$\frac{dn}{dt} = \frac{\eta(u)}{\tau}(g(v) - n), \tag{20}$$

$$\frac{dq}{dt} = \frac{\epsilon_q}{\tau}(h(v) - q), \tag{21}$$

$$\frac{du}{dt} = \frac{\epsilon_u}{\tau}(v + v_0 - \alpha_u u), \tag{22}$$

$$\eta(u) = \begin{cases} \eta_0 & (u < r_u) \\ \eta_1 & (u \geq r_u), \end{cases} \tag{23}$$

where u is a state variable with a large time constant that varies the structure of the fast subsystem consisting of v and n. The η controls the time constant of n, whose value is determined to be η_0 or η_1 depending on the value of u and a threshold r_u. In the DSSN model, the LTS class belongs to the three-variable mode. In the PQN model, it is included in the extended four-variable model to reproduce the rebound bursting more faithfully. In the extended four-variable model, the structure of the fast subsystem consisting of v and n is shifted depending on the change in the time constant of the state variable. In the DSSN model, the time constant of v is

varied by u, whereas in the PQN model, the time constant of n is varied instead to avoid increasing the calculation process for v.

These differential equations are numerically integrated with Euler's method, whose time steps are 1 ms for the PB class and 0.1 ms for the other classes. The neuronal parameters are fitted so that the model reproduces the properties of each class as accurately as possible and consumes as few circuit resources as possible in the digital arithmetic circuit implementation. The parameters are determined by a semi-automatic optimization method developed previously, the detailed protocol of which is described in Nanami et al. (2017, 2018). Since the synaptic current is not fitted in this study, the synaptic parameters α and β are the same values used in the previous study (Li et al., 2012). The neuronal parameters of all classes are listed in the Supplementary material.

In the PLS framework, neuron models are described by a combination of core functions P, L, and S, which represent polynomial, piecewise-linear, and step functions, respectively. For example, core functions P_1, L_1, and S_1 are defined as follows:

$$P_1(x, x_0) = x_0 - x, \tag{24}$$

$$L_1(x, x_0, x_1, x_2) = x_1 + x_2(x - x_0), \tag{25}$$

$$S_1(x, x_0, x_1, x_2) = \begin{cases} x_1 & (x < x_0) \\ x_2 & (x \geq x_0). \end{cases} \tag{26}$$

The Class II mode of the PQN model in the PLS-framework is given by

$$\begin{aligned} \frac{\tau}{\theta}\frac{dv}{dt} &= S_1\Big(v, 0, a_{fn}(P_1(v, b_{fn}))^2 \\ &\quad + c_{fn}, a_{fp}(P_1(v, b_{fp}))^2 + c_{fp}\Big) - n + k\, I_{stim}, \quad (27) \\ \tau\frac{dn}{dt} &= S_1\Big(v, r_g, a_{gn}(P_1(v, b_{gn}))^2 \\ &\quad + c_{gn}, a_{gp}(P_1(v, b_{gp}))^2 + c_{gp}\Big) - n. \quad (28) \end{aligned}$$

Equations of the RS, FS, and EB classes are given by

$$\begin{aligned} \frac{\tau}{\theta}\frac{dv}{dt} &= S_1\Big(v, 0, a_{fn}(P_1(v, b_{fn}))^2 + c_{fn}, a_{fp}(P_1(v, b_{fp}))^2 + c_{fp}\Big) \\ &\quad -n - q + k\, I_{stim}, \quad (29) \\ \tau\frac{dn}{dt} &= S_1\Big(v, r_g, a_{gn}(P_1(v, b_{gn}))^2 \\ &\quad + c_{gn}, a_{gp}(P_1(v, b_{gp}))^2 + c_{gp}\Big) - n, \quad (30) \\ \frac{\tau}{\epsilon_q}\frac{dq}{dt} &= S_1\Big(v, r_h, a_{hn}(P_1(v, b_{hn}))^2 \\ &\quad + c_{hn}, a_{hp}(P_1(v, b_{hp}))^2 + c_{hp}\Big) - q, \quad (31) \end{aligned}$$

and equations of the PB classes are given by

$$\begin{aligned} \frac{\tau}{\theta}\frac{dv}{dt} &= S_1\Big(v, 0, a_{fn}(P_1(v, b_{fn}))^2 + c_{fn}, a_{fp}(P_1(v, b_{fp}))^2 \\ &\quad + c_{fp}\Big) - n - q + u + k\, I_{stim}, \quad (32) \\ \tau\frac{dn}{dt} &= S_1\Big(v, r_g, a_{gn}(P_1(v, b_{gn}))^2 \\ &\quad + c_{gn}, a_{gp}(P_1(v, b_{gp}))^2 + c_{gp}\Big) - n, \quad (33) \\ \frac{\tau}{\epsilon_q}\frac{dq}{dt} &= S_1\Big(v, r_h, a_{hn}(P_1(v, b_{hn}))^2 \\ &\quad + c_{hn}, a_{hp}(P_1(v, b_{hp}))^2 + c_{hp}\Big) - q, \quad (34) \\ \frac{\tau}{\epsilon_u\alpha_u}\frac{du}{dt} &= L_1\left(v,\ 0, \frac{v_0}{\alpha_u}, \frac{1}{\alpha_u}\right) - u. \quad (35) \end{aligned}$$

Equations of the LTS and IB classes are written using the step function as follows:

$$\begin{aligned} \frac{\tau}{\theta}\frac{dv}{dt} &= S_1\Big(v, 0, a_{fn}(P_1(v, b_{fn}))^2 \\ &\quad + c_{fn}, a_{fp}(P_1(v, b_{fp}))^2 + c_{fp}\Big) \\ &\quad - n - q + k\, I_{stim}, \quad (36) \\ S1\left(u, r_u, \frac{\tau}{\eta_0}, \frac{\tau}{\eta_1}\right)\frac{dn}{dt} &= S_1\Big(v, r_g, a_{gn}(P_1(v, b_{gn}))^2 + c_{gn}, \\ &\quad a_{gp}(P_1(v, b_{gp}))^2 + c_{gp}\Big) - n, \quad (37) \\ \frac{\tau}{\epsilon_q}\frac{dq}{dt} &= S_1\Big(v, r_h, a_{hn}(P_1(v, b_{hn}))^2 + c_{hn}, \\ &\quad a_{hp}(P_1(v, b_{hp}))^2 + c_{hp}\Big) - q, \quad (38) \\ \frac{\tau}{\epsilon_u\alpha_u}\frac{du}{dt} &= L_1\left(v, 0, \frac{v_0}{\alpha_u}, \frac{1}{\alpha_u}\right) - u. \quad (39) \end{aligned}$$

2.2. Implementations of the PQN model

Here, the implementation of the extended four-variable model, the most complex variant, is explained. Figure 2 shows the block diagram of the PQN engine, which calculates the state variables at the next time step. The symbols ×, +, M, and C in the figure represent multipliers, adders, multiplexers, and comparators, respectively. The vv is a square of v and the n_0 is an intermediate result of calculation for n_{next}. The v_c, n_c, q_c, u_c, and s_c_L, are constants, and they are calculated in advance and stored. The signals v_vv_S, v_vv_L, v_v_S, v_v_L, v_n, v_q, v_I, n_uS, n_uL, n_vv_S, n_vv_L, n_v_S, n_v_L, n_n, q_vv_S, q_vv_L, q_v_S, q_v_L, q_q, u_v, and u_u are the intermediate results of the calculation. They are calculated by the multiplication of a signal and a coefficient, which is implemented by shifters and adders. For example, Figure 3 shows the calculation of the signal v_vv_S. If $Y_{v\_vv\_S}$ is 0.037109375, and its binary representation is 0.000010011. Therefore, calculating the sum of the sixth, fifth, and ninth right-shift operations on the square of v is performed. The

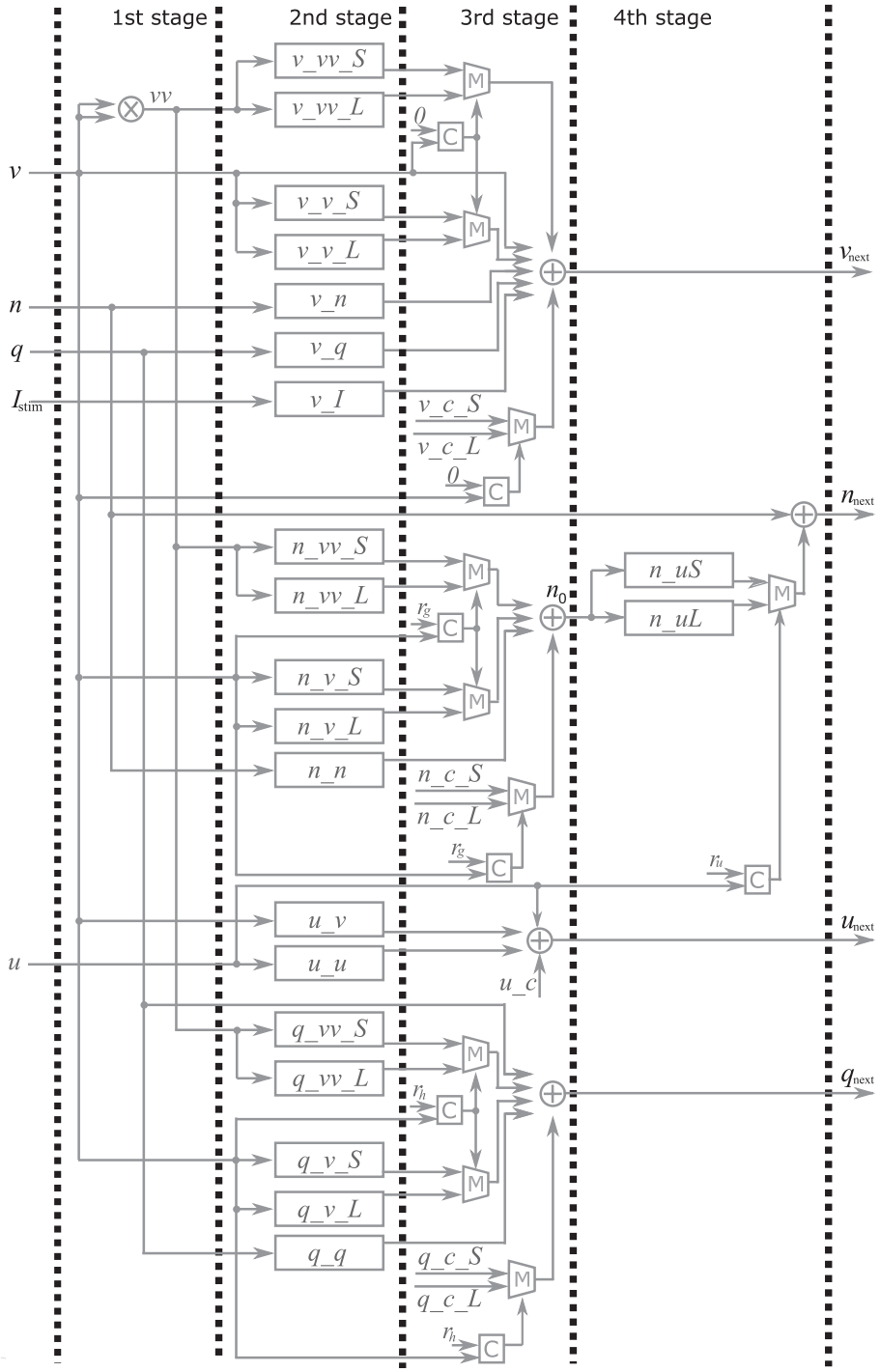

FIGURE 2
Block diagram of the PQN engine for the IB and LTS mode. The symbols ×, +, *M* and *C* represent multipliers, adders, multiplexers, and comparators, respectively. The *v*, *n*, *q*, *u* and I_{stim} are input for the PQN engine. The output v_{next}, n_{next}, q_{next}, and u_{next} are values of the state variables at the next time step.

results of the calculations are stored in registers for pipelined design. Details of the signals, coefficients, and constants are shown in Supplementary material.

In the first stage, the square of *v* is calculated using the multiplier. The second stage involves the multiplication of the variables and coefficients. In the third stage, the result

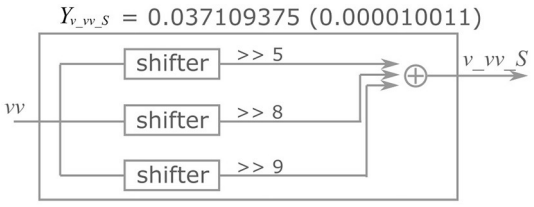

FIGURE 3
An example of the multiplication of a state variable and a coefficient.

of the calculation of v_{next}, q_{next}, $u_{\text{next}}$is obtained, In the fourth stage, the value of s_{next} is determined based on the s and v_{next} the value of n_{next} is determined based on the u and n_0.

The PQN unit is a complex of a PQN engine and registers for the state variables (Figure 4A). The registers are configured as FIFO memory and the values are in turn passed to the PQN engine. Here, N represents the number of neurons that one PQN unit computes. The values of the state variables for the next time step are returned to the FIFO memory and stored. Figure 4B shows the pipeline usage of the PQN unit. In IB and LTS modes, the time step is 0.1 ms. To perform the real-time simulation at 100 MHz clock, 10,000 cycles can be consumed to update all state variables. Since the PQN engine takes 4 cycles and reads and writes of the FIFO require 3 clocks, the N becomes 9993.

Since multiplications between a coefficient and state variable are realized by shifters and adders, the values of the parameters affect the circuit resources. Each additional number of ones in the binary representation of the coefficients consumes an additional pair of shifters and adders.

Except for the Class II mode, all state variables are expressed in an 18-bit fixed-point with an 8-bit integer part. In the Class II mode with the 18-bit fixed-point, the PRC is biphasic but not smooth, so we used a 28-bit fixed-point with an 8-bit integer part representation.

3. Results

3.1. Simulation results of the class II mode

Class II mode in the Hodgkin's classification is known (Rinzel and Ermentrout, 1998) to show the graded response to pulsed stimuli and has the biphasic PRC. Figure 5A shows simulation results of the PQN model in response to five pulse stimuli of different magnitudes. The maximum amplitude of the spike increases as the magnitude of the pulse stimulus increases, which is referred to as the graded response.

The PRC was evaluated by measuring phase shift while varying the phase of the pulse stimulus. The phase of the pulse stimulus θ and phase shift $\Delta(\theta)$ are defined as follows:

$$\theta = \frac{t_i - t_s}{T}, \tag{40}$$

$$\Delta(\theta) = \frac{T - T_i}{T}, \tag{41}$$

where T represents the time length of the oscillatory spiking cycle, and t_s and t_i are the time of the beginning of the cycle and the time when the pulse stimulus was given, respectively (Figure 5B). T_i represents the length of the resulting cycle shifted by the pulse stimulus. Figure 5C shows the resulting PRC, and it exhibits the standard Type II as well as the ionic-conductance model.

3.2. Simulation results of the RS, FS, LTS, IB, EB, and PB modes

The target models for the RS, FS, LTS, and IB classes are the ionic-conductance models given in Pospischil et al. (2008). For the EB and PB classes, the ionic-conductance models shown in Plant (1981) and Wang (1993) are chosen, respectively. For the RS class, we reproduced an excitatory cell and an inhibitory cell shown in Pospischil et al. (2008), and thus we prepared seven PQN model parameter sets in total. All the results were obtained using a Xilinx Artix-7 XC7A35T FPGA on a Digilent cmod-a7 board. The circuit was developed using Xilinx Vivado 2018.3 software. The clock signal of the FPGA is 100 MHz. The input current was given from the PC to the PQN unit on the FPGA via serial communication, and the value of the membrane potential v of the PQN unit was sent back to the PC. For comparison, we also prepared simulation results of DSSN models for each class; the DSSN model presented in Nanami et al. (2017, 2018) for the RS, FS, LTS, and IB classes, and Nanami et al. (2016) for the EB and PB classes. The DSSN models and ionic-conductance models were simulated on a PC using the python software.

Figures 6–12A show the responses of the membrane potential in response to three different amplitudes of stimulus for excitatory RS, inhibitory RS, FS, LTS, IB, EB, and PB classes, respectively. The orange and blue plots represent the membrane potentials of the ionic-conductance and PQN models, respectively. The gray plots are the stimulus input currents, whose unit in the ionic-conductance model is pA. In the PQN model, they have no physical unit. In the excitatory RS mode, both models exhibit spike-frequency adaptation. Both models in the inhibitory RS mode show more intense spike-frequency adaptation than those in the excitatory RS mode. In the FS mode, weak spike frequency adaptation is only seen immediately after the step input is given. In the LTS mode, both models show a strong spike-frequency adaptation in response to positive step stimuli. Immediately after the long inhibitory input is removed, they show transient burst firing, which is called rebound bursting. Both models

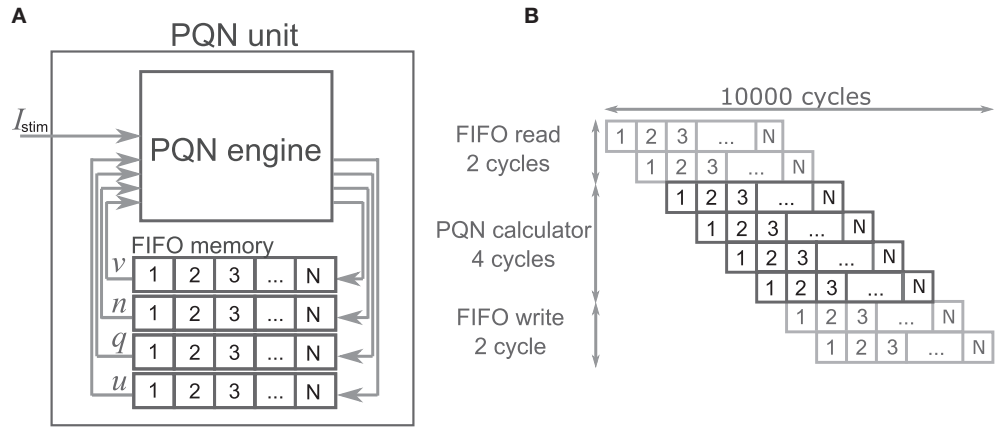

FIGURE 4
The structure and clock cycle of the PQN unit. **(A)** The internal structure of the PQN unit. The state variables are stored in the FIFO memory and sent to the PQN engine in turn. **(B)** The clock cycle of the PQN unit.

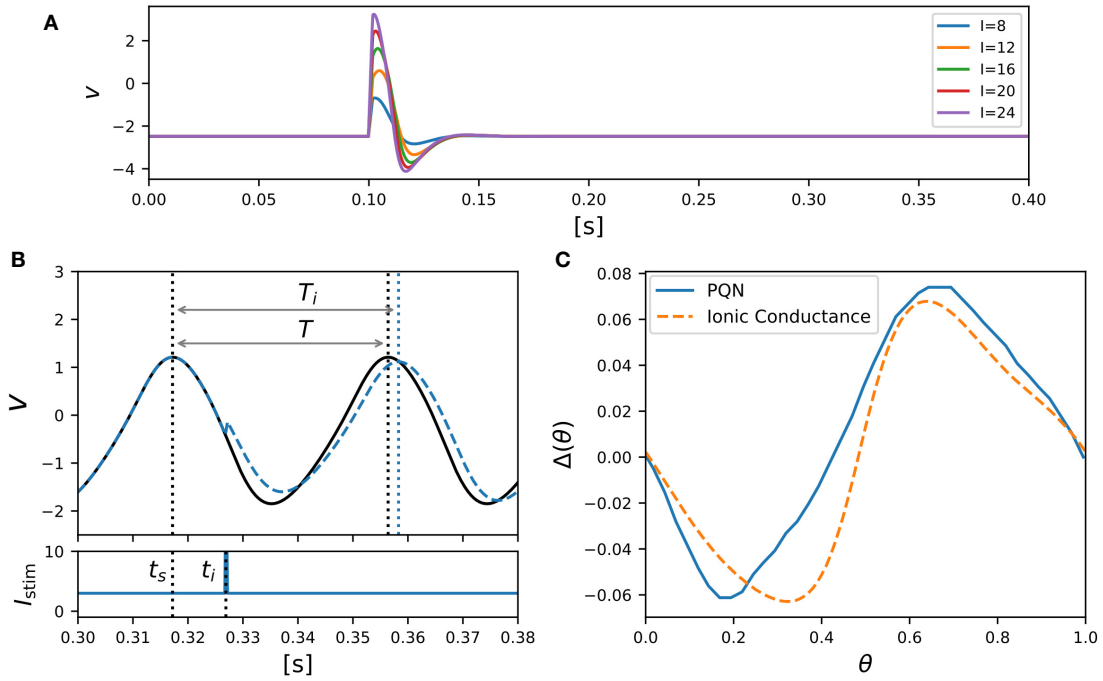

FIGURE 5
PQN model of Class II mode in the Hodgkin's classification. **(A)** Graded responses to pulse stimuli. Pulse stimuli of various magnitudes were given between 0.1 and 0.102 [s]. **(B)** Phase changes induced by the pulse stimulus. **(C)** PRCs of the PQN model and the ionic-conductance model. The horizontal axis represents the phase at which the pulse stimulus was given, and the vertical axis represents how much the phase was shifted by the pulse stimulus.

in the IB mode exhibit burst firing immediately after step stimuli are given. In response to more intense inputs, they show the periodic firings of the low frequency following the bursting. In the EB mode, both models repeat burst firings at constant intervals in response to constant stimulus inputs. As the stimuli become more intense, the number of spikes in the bursts becomes larger, and the intervals between bursts become shorter. Both models in the PB mode show repeated burst firings at constant intervals to the constant stimulus inputs. During the intervals between burst firings, the

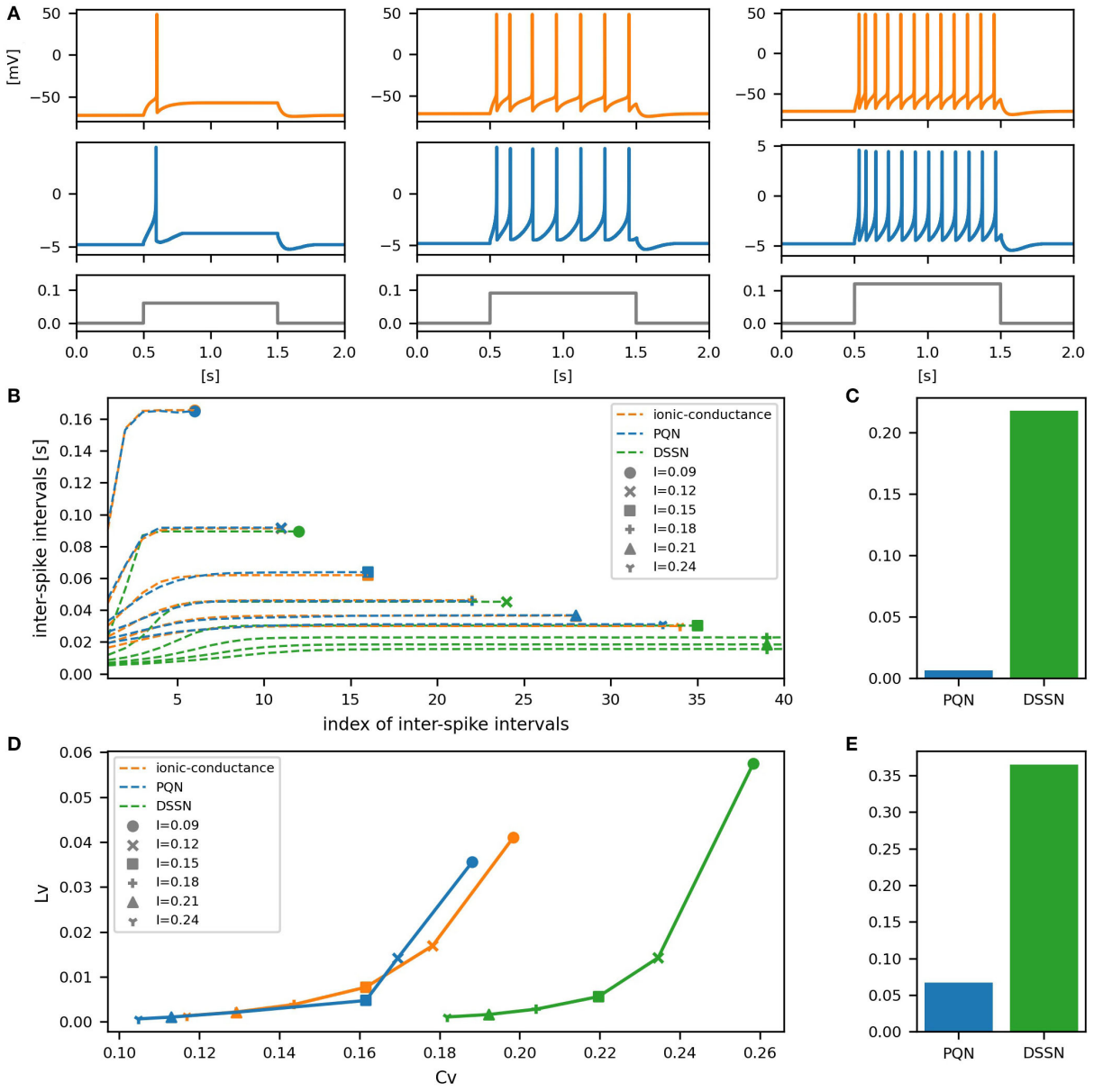

FIGURE 6
Waveforms and spiking properties in excitatory RS mode. **(A)** Waveforms of the ionic-conductance model (orange) and the PQN model (blue). Gray lines represent stimulus inputs, whose unit in the ionic-conductance model is nA. In the PQN model, they have no physical unit. **(B)** The transitions of ISIs are shown while increasing the magnitude of the stimulus input. The horizontal axis represents the index of the ISIs, and the vertical axis represents the magnitude of the ISIs. Orange, blue, and green lines represent the ionic-conductance, PQN, and DSSN models, respectively. Markers were plotted to show the corresponding magnitude of the stimulus input. **(C)** The mean square errors of the PQN model and the DSSN model with the ionic-conductance model over the data points shown in **(B)**. **(D)** Values of the C_V and L_V were calculated from the waveforms and plotted while varying the magnitude of the stimulus inputs. The waveforms used here are the same as that used in **(B)**. **(E)** The mean square errors of the PQN model and the DSSN model with the ionic-conductance model over the data points shown in **(D)**.

subthreshold oscillation is seen in the EB class but not in the PB class.

To quantitatively assess the differences in activities between the ionic-conductance model and the PQN and DSSN models, we employed several statistics. For the RS, FS, LTS, and IB classes we initially measured and plotted the transition of inter-spike intervals (ISIs) for each spike sequence (Figures 6–10B). The RS, FS, and LTS classes exhibit spike frequency adaptation, and the ISI slowly increases in response to constant input. We measured the transition of ISIs because it can capture the characteristics of the change in ISIs. In the IB class, the ISIs are small for the first few spikes due to the initial bursting, and then ISIs

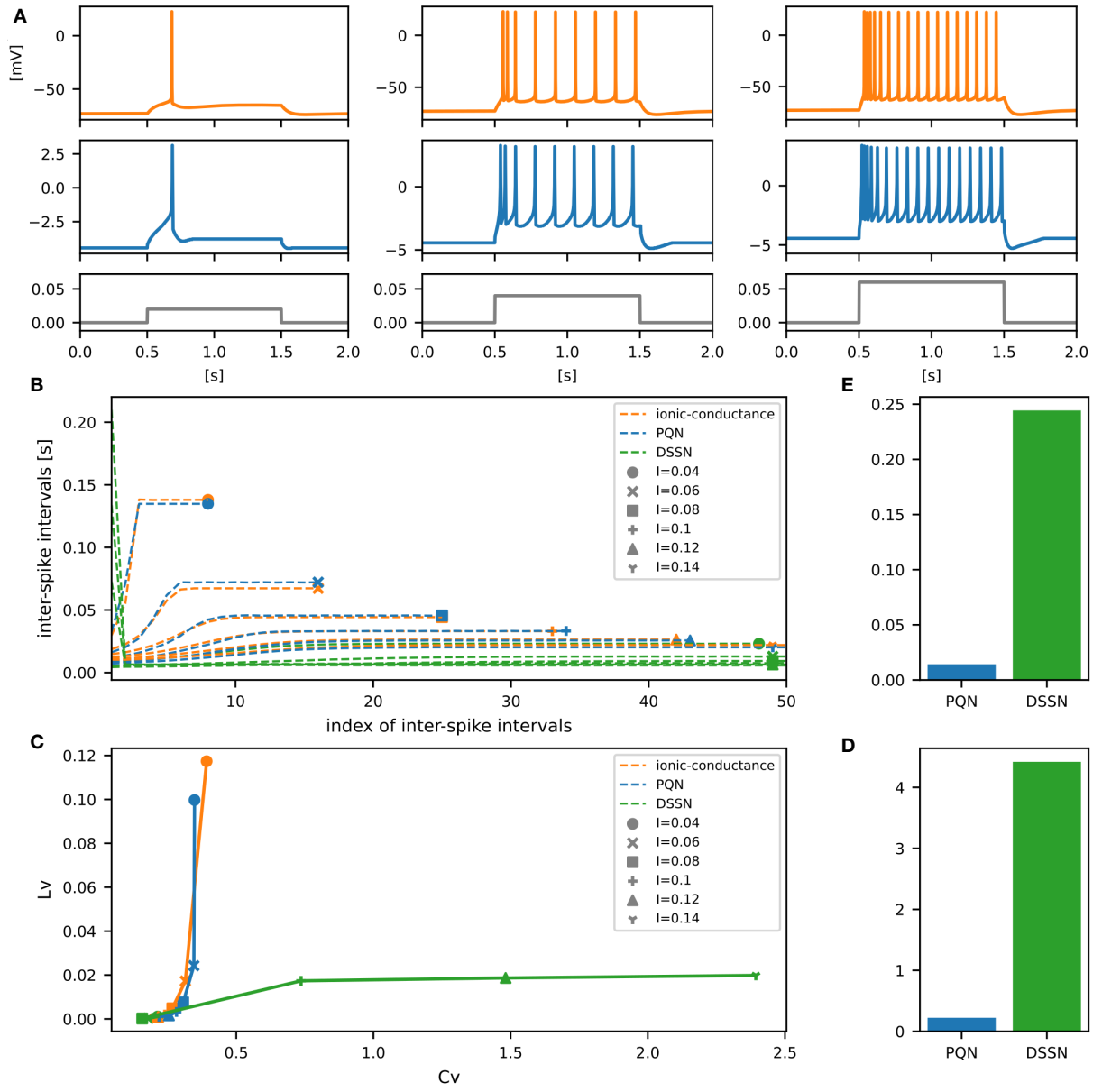

FIGURE 7
Waveforms and spiking properties in inhibitory RS mode. **(A)** Waveforms of the ionic-conductance model (orange) and the PQN model (blue). Gray lines represent stimulus inputs. **(B)** The transitions of ISIs are shown while increasing the magnitude of the stimulus input. **(C)** The mean square errors of the PQN model and the DSSN model with the ionic-conductance model over the data points shown in **(B)**. **(D)** Values of the C_V and L_V were calculated from the waveforms and plotted while varying the magnitude of the stimulus inputs. The waveforms used here are the same as that used in **(B)**. **(E)** The mean square errors of the PQN model and the DSSN model with the ionic-conductance model over the data points shown in **(D)**.

become larger. ISIs were also measured and plotted in the IB class to visualize these characteristics. The EB and PB classes repeat burst firing in response to a constant stimulus. The characteristics of repeated burst firing were quantified by the number of spikes in a single bursting and the intervals between bursting (Figures 11, 12B).

In addition, for all classes, we plotted the coefficient of variation (C_V) and local variation (L_V) (Shinomoto et al., 2003). The C_V and L_V are calculated as follows:

$$C_V = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(T_i - \overline{T})^2}/\overline{T}, \tag{42}$$

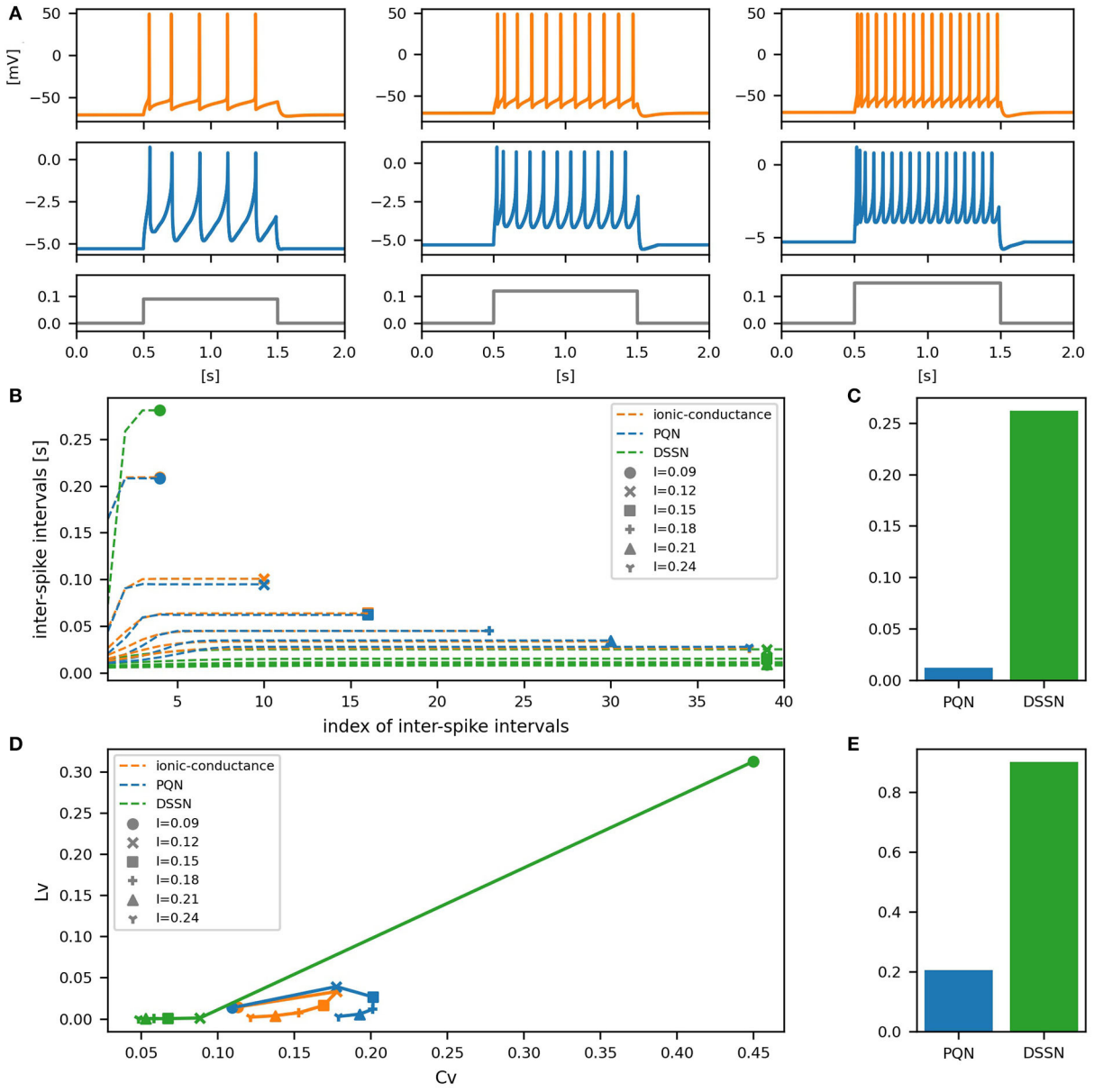

FIGURE 8
Waveforms and spiking properties in FS mode. **(A)** Waveforms of the ionic-conductance model (orange) and the PQN model (blue). Gray lines represent stimulus inputs. **(B)** The transitions of ISIs are shown while increasing the magnitude of the stimulus input. **(C)** The mean square errors of the PQN model and the DSSN model with the ionic-conductance model over the data points shown in **(B)**. **(D)** Values of the C_V and L_V were calculated from the waveforms and plotted while varying the magnitude of the stimulus inputs. The waveforms used here are the same as that used in **(B)**. **(E)** The mean square errors of the PQN model and the DSSN model with the ionic-conductance model over the data points shown in **(D)**.

$$L_V = \frac{1}{n-1}\sum_{i=1}^{n-1}\frac{3(T_i - T_{i+1})^2}{(T_i + T_{i+1})^2}, \tag{43}$$

where T_i represents the ith ISI, and $\overline{T}$ is the average of T_i. n corresponds to the number of spikes. The coefficient 3 in L_V is determined so that the expectation value of L_V in the Poisson spike sequence becomes one. The C_V represents the standard deviation divided by the mean ISI, and L_V measures a local fluctuation of the ISIs. Both C_V and L_V become zero for a spike sequence whose ISIs are constant. The C_V and L_V are widely used (Shinomoto et al., 2005a, 2009; Miura et al., 2006) to assess the firing properties of neurons in the nervous system.

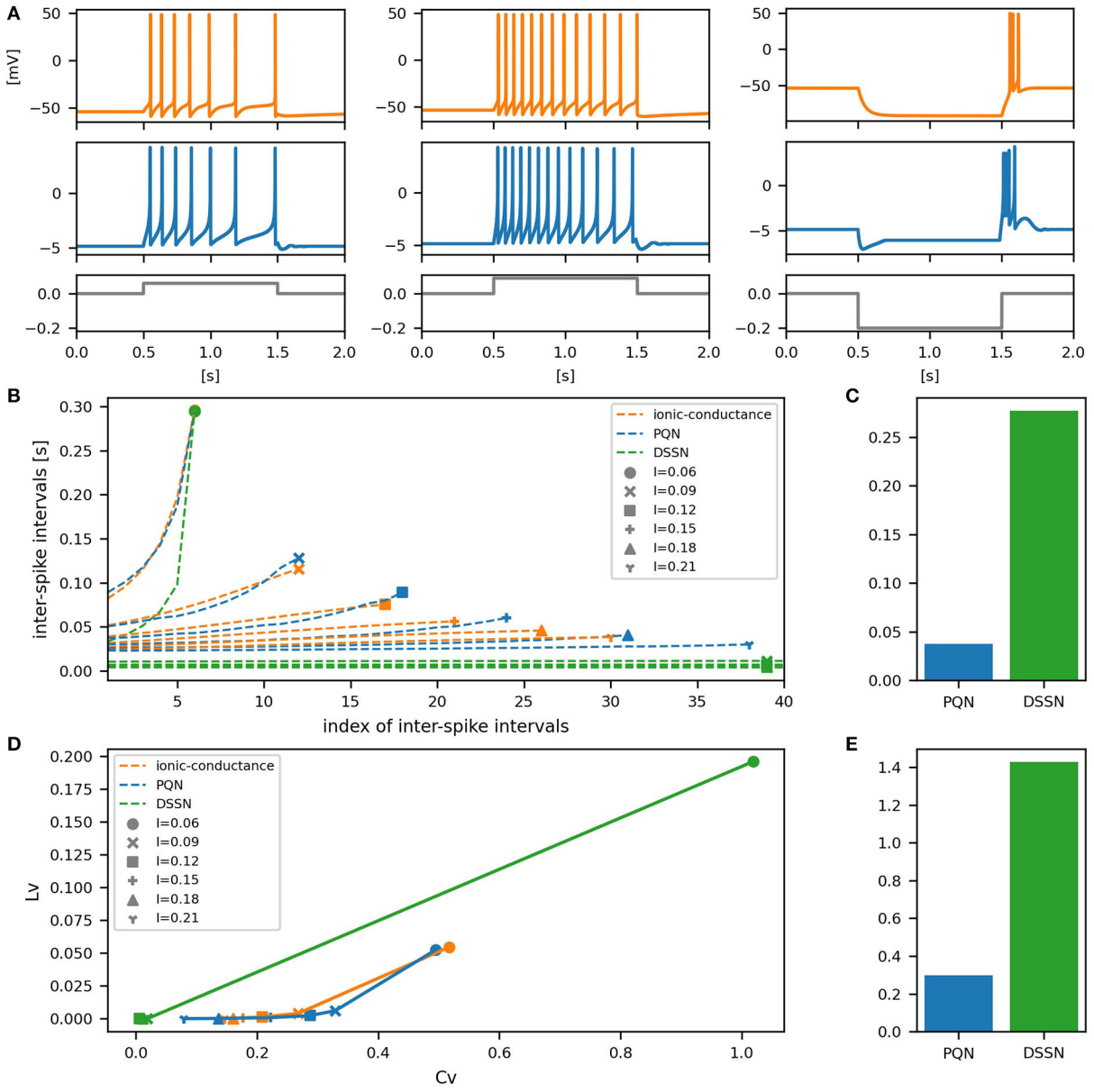

FIGURE 9
Waveforms and spiking properties in LTS mode. **(A)** Waveforms of the ionic-conductance model (orange) and the PQN model (blue). Gray lines represent stimulus inputs. **(B)** The transitions of ISIs are shown while increasing the magnitude of the stimulus input. **(C)** The mean square errors of the PQN model and the DSSN model with the ionic-conductance model over the data points shown in **(B)**. **(D)** Values of the C_V and L_V were calculated from the waveforms and plotted while varying the magnitude of the stimulus inputs. The waveforms used here are the same as that used in **(B)**. **(E)** The mean square errors of the PQN model and the DSSN model with the ionic-conductance model over the data points shown in **(D)**.

In Figures 6–10B, the horizontal axes represent the index of the ISIs, and the vertical axes represent the length of the ISIs. Solid, dashed, and dotted curves represent the ionic-conductance model, PQN model, and DSSN model, respectively. Six different magnitudes of step inputs are given. Although the DSSN model reproduces the mathematical structure of each class, it is not intended to reproduce the transition of spiking activities in response to the change in stimulus magnitude. In these figures, the input to the DSSN model is linearly transformed so that the number of spikes in the DSSN model matches those in the ionic-conductance model when the weakest input is applied in each figure. The response of the DSSN model differs significantly from that of the ionic-conductance model, while the response of the PQN

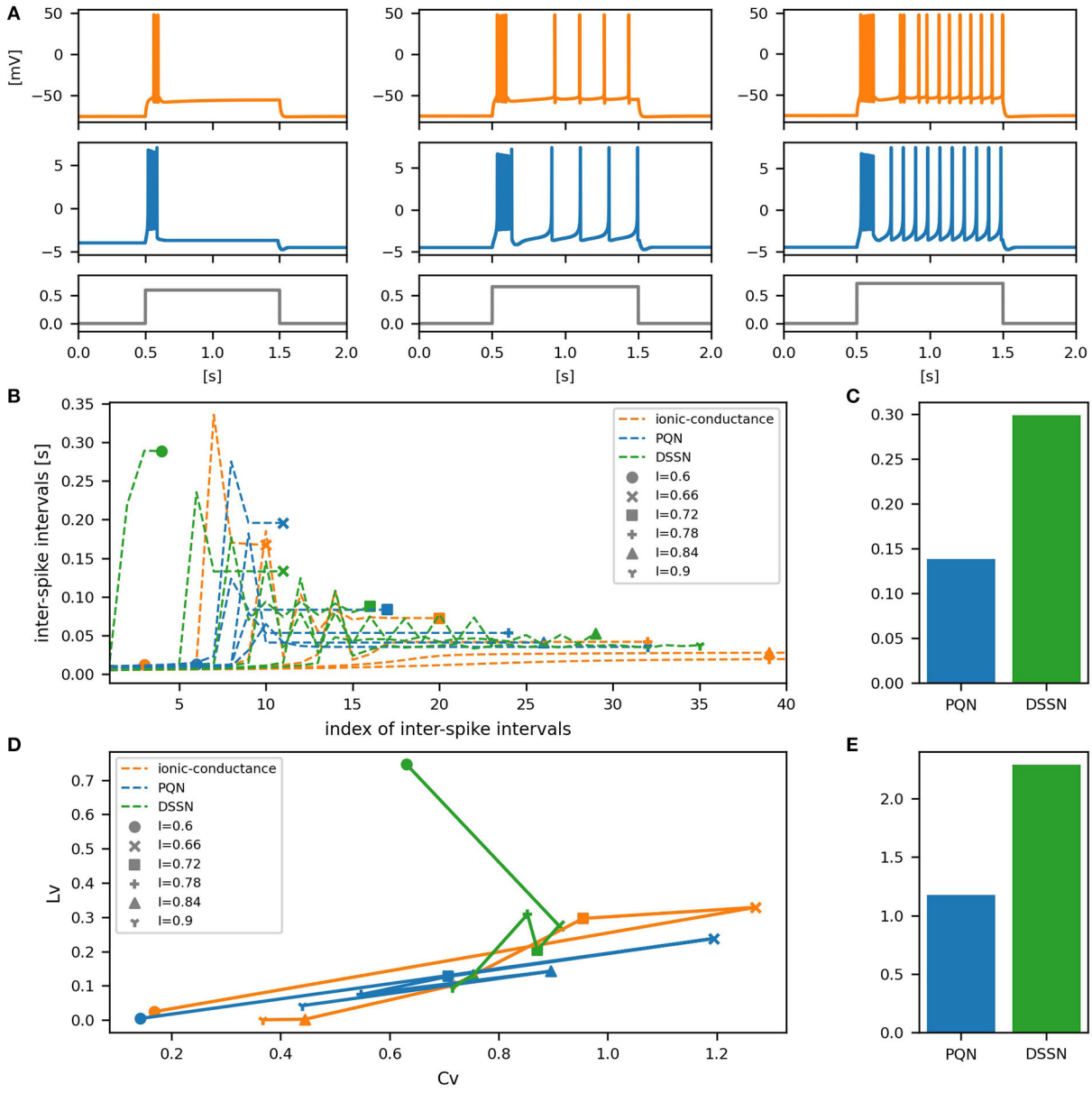

FIGURE 10
Waveforms and spiking properties in IB mode. **(A)** Waveforms of the ionic-conductance model (orange) and the PQN model (blue). Gray lines represent stimulus inputs. **(B)** The transitions of ISIs are shown while increasing the magnitude of the stimulus input. **(C)** The mean square errors of the PQN model and the DSSN model with the ionic-conductance model over the data points shown in **(B)**. **(D)** Values of the C_V and L_V were calculated from the waveforms and plotted while varying the magnitude of the stimulus inputs. The waveforms used here are the same as that used in **(B)**. **(E)** The mean square errors of the PQN model and the DSSN model with the ionic-conductance model over the data points shown in **(D)**.

model is closer to the ionic-conductance model. Figures 6–10C show the mean square errors of the PQN model and the DSSN model with the ionic-conductance model over the data points shown in Figures 6–10B. The value of the mean square errors of the PQN model was about half of that of the DSSN model for the IB class and less than one-fifth for the other classes.

In Figures 11, 12B, the timing properties of bursting are plotted while increasing the magnitude of the stimulus input. The horizontal axes represent the intervals between the bursting phases, and the vertical axes represent the number of spikes in a burst. Six different magnitudes of constant inputs are given. The inputs to the DSSN model were linearly transformed so that the number of spikes in the DSSN model matches

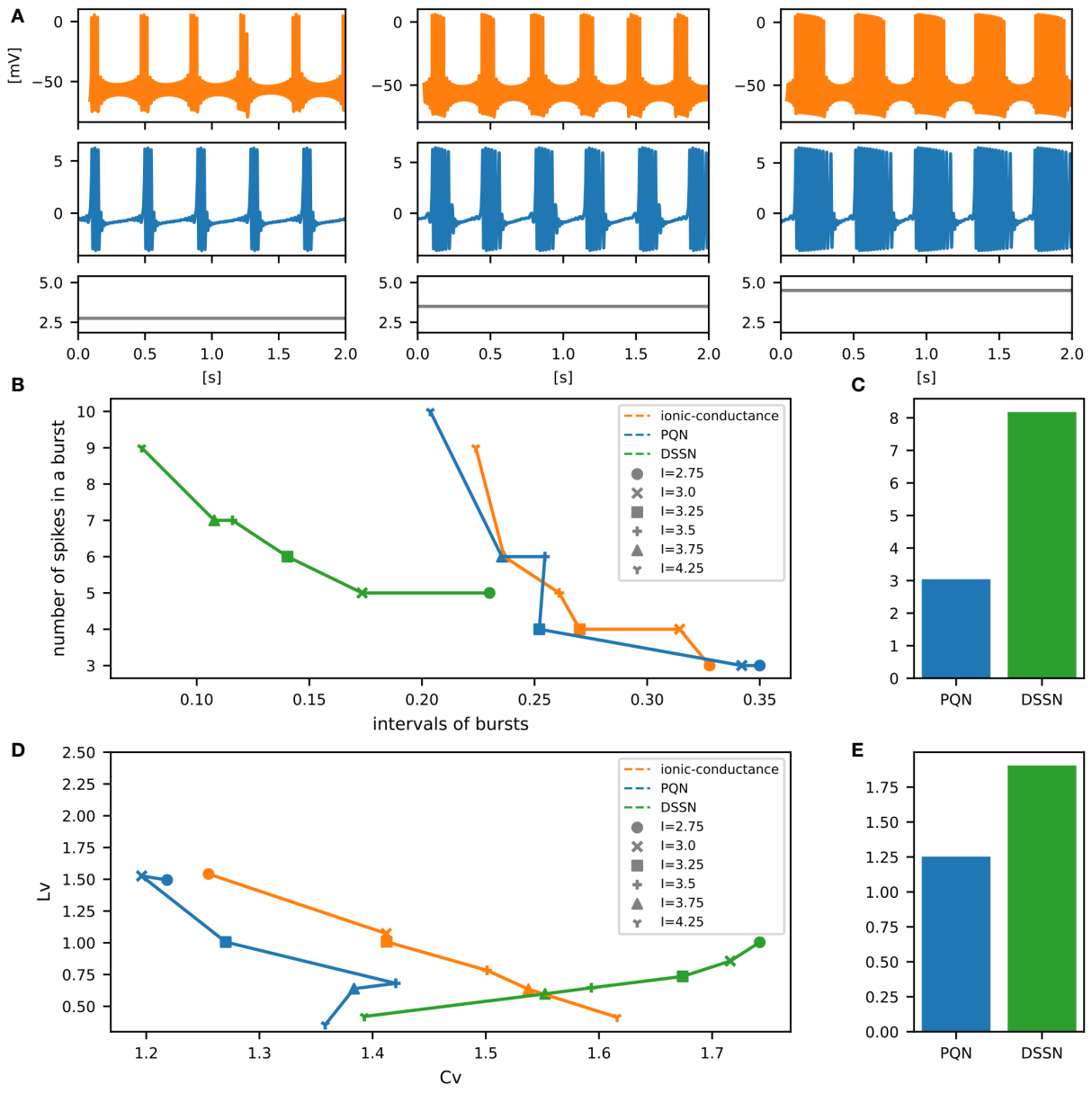

FIGURE 11
Waveforms and spiking properties in EB mode. **(A)** Waveforms of the ionic-conductance model (orange) and the PQN model (blue). Gray lines represent stimulus inputs. **(B)** Properties of bursting. The horizontal axis represents the interval between bursts, and the vertical axis represents the number of spikes in a burst. Stimulus inputs were increased by 0.25 from 2.75 to 4.25. **(C)** The mean square errors of the PQN model and the DSSN model with the ionic-conductance model over the data points shown in **(B)**. **(D)** Values of the C_V and L_V were calculated from the waveforms and plotted while varying the magnitude of the stimulus inputs. **(E)** The mean square errors of the PQN model and the DSSN model with the ionic-conductance model over the data points shown in **(D)**.

those in the ionic-conductance model when the weakest input shown in each figure is given. Figures 11, 12C show the mean square errors of the PQN model and the DSSN model with the ionic-conductance model over the data points shown in Figures 11, 12B. The value of the mean square errors of the PQN model was about a quarter of that of the DSSN model.

In Figures 6–12D, the characteristics of the three models are plotted on the C_V-L_V plane. The response data used in Figures 6–12B were used here. Figures 6–12E show the mean square errors of the PQN model and the DSSN model with the ionic-conductance model over the data points shown in Figures 6–12D. The value of the mean square errors of the PQN model was about half of that of the DSSN model for

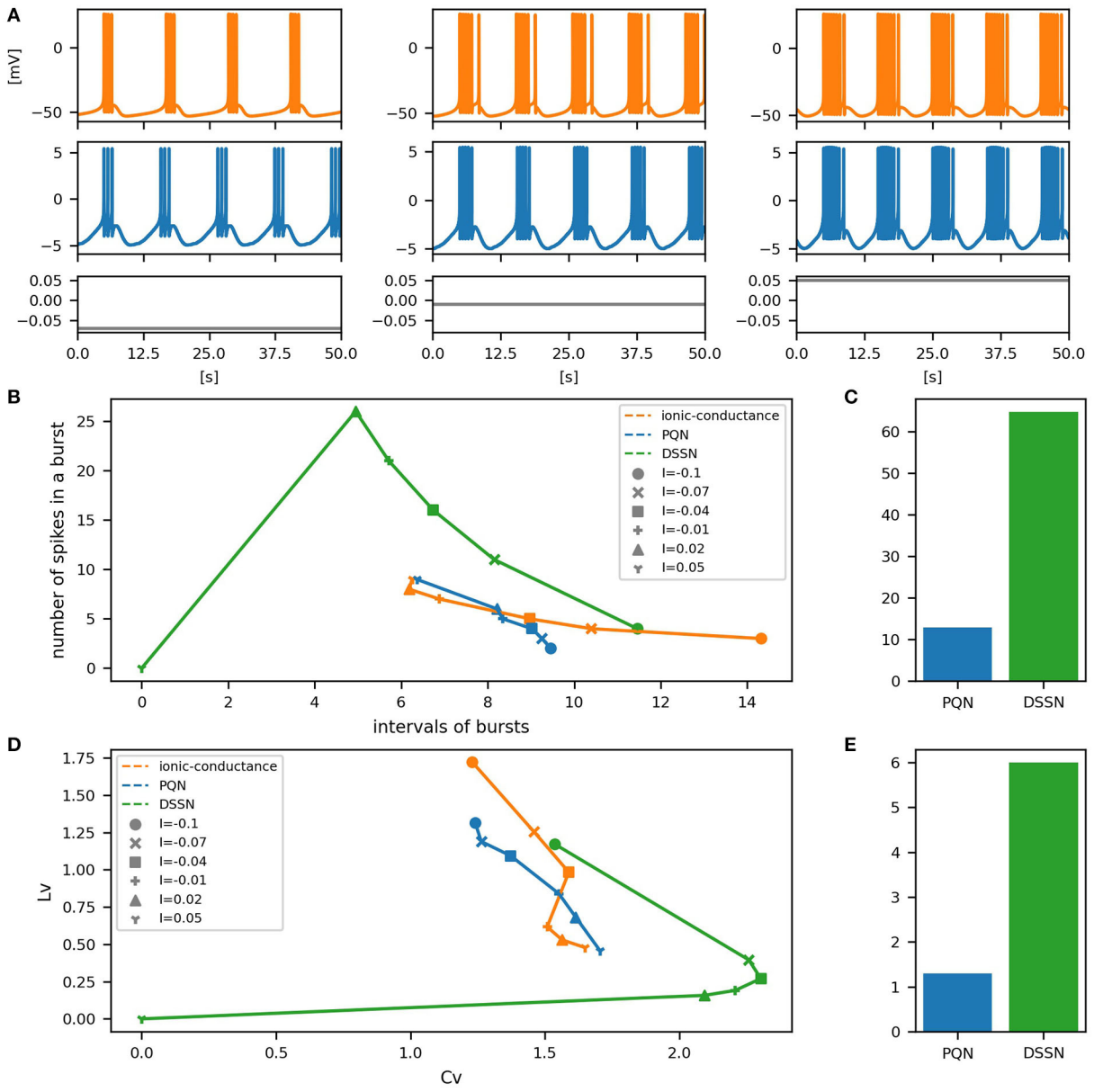

FIGURE 12
Waveforms and spiking properties in PB mode. **(A)** Waveforms of the ionic-conductance model (orange) and the PQN model (blue). Gray lines represent stimulus inputs. **(B)** Properties of bursting. The horizontal axis represents the interval between bursts, and the vertical axis represents the number of spikes in a burst. Stimulus inputs were increased by 0.25 from 2.75 to 4.25. **(C)** The mean square errors of the PQN model and the DSSN model with the ionic-conductance model over the data points shown in **(B)**. **(D)** Values of the C_V and L_V were calculated from the waveforms and plotted while varying the magnitude of the stimulus inputs. **(E)** The mean square errors of the PQN model and the DSSN model with the ionic-conductance model over the data points shown in **(D)**.

the IB and EB classes and less than one-quarter for the other classes.

3.3. Resource utilization on the FPGA

Table 1 shows resource consumption of the PQN, DSSN, and ionic-conductance models in the FPGA implementation. Previous studies (Nanami et al., 2016, 2017, 2018) of the DSSN model showed only the resources of the circuit of the DSSN engine, which computes the values of the state variables in the next step from the current values of the state variables. Therefore, we show the resources of both the PQN engine and the PQN unit for comparison. Here, look-up tables (LUTs) are truth tables and digital signal processors (DSPs) are blocks for complex calculations. Flip-flops (FFs) are memory elements

and are mainly used for the registers. In the PQN and DSSN models, LUTs are mainly used for addition, and the DSP is used to calculate square v. The consumption of LUTs in the PQN engines is larger than that of the DSSN engines, except for the PB mode. This is because we focused more on reproducing spiking properties than on reducing resources in parameter tuning. Both the PQN and DSSN models consume only one DSP. The PQN unit consists of the PQN engine, FIFO memories, and the controller part responsible for data transfer between the PQN engine and FIFO memories, requiring more LUTs and FFs than the PQN engine. The PQN unit operates at 100 MHz clock at maximum and is capable of simulating 9993 neurons in real-time. Since the time step of the PB mode is 1 ms, it can be estimated that a single PQN unit of the PB mode can compute 99,993 neurons, but due to the lack of capacity of the block RAM, only 16,384 neurons are realized here. The FPGA implementation (Khoyratee et al., 2019) of the ionic-conductance model is programmable and can cover RS, FS, LTS, and IB classes by varying parameters. However, it can only simulate 500 neurons and requires a large amount of DSPs, which can be a bottleneck in building large-scale networks.

4. Conclusion

In this study, we proposed the PQN model, which is a refined version of the DSSN model. The PQN model has fewer parameters than the DSSN model, and its parameter sets for each class were renewed.

It was shown that the two-variable PQN model can properly reproduce the Class II in Hodgkin's classification. Both the graded response to pulse stimulus and the standard Type II PRC, which are typical in the Class II neurons, were confirmed. Note that these properties cannot be reproduced by the I&F-based models. For the other variants of the PQN model, parameters were fitted to reproduce the spiking activities in the ionic-conductance models of the RS, FS, LTS, IB, EB, and PB classes. With the RS, FS, LTS, and IB classes, we measured the transition of the ISIs of the spike sequences to evaluate characteristics of the spike-frequency adaptation and initial bursting with subsequent sparse firing. With the EB and PB classes, the bursting properties were evaluated by the interval between bursts and the number of spikes in a burst. In addition, statistical indicators for spike sequence classification in theoretical neuroscience (the C_V and L_V) were evaluated for all classes. It is reported (Shinomoto et al., 2003, 2005b) that spike trains recorded from different areas of the cortex exhibit different values of C_V and L_V. C_V focuses on variations in the length of ISIs across entire spike trains, while L_V captures the change in adjacent ISIs. For example, in the bursting neuron class including the EB and PB, L_V shows larger values than that in other classes because ISIs change extremely at the beginning and end of bursts. These statistics showed that the PQN model reproduces the activities of the ionic-conductance models with dramatically higher accuracy than the DSSN model.

In Table 1, the circuit resources required for updating the state variables for a single time step were compared between the PQN and DSSN models. Since a single DSP supports the multiplication of 25-bit by 18-bit, the Class II mode with 28-bit state variables requires four DSPs. The other modes require only one DSP as well as the DSSN model. The PQN model consumes a larger number of LUTs and FFs than the DSSN model. This is because the PQN model consumes more shifters and adders to accurately reproduce the activities of the ionic-conductance models. In addition, buffers in the pipeline process consume more FF in the PQN model. Please note that the implementation of the DSSN model was not pipelined in the previous studies (Nanami et al., 2017, 2018) because the main goal of these studies was to build parameter fitting methods of the model and the implementation was done only to verify that it worked correctly. Therefore, the maximum clock frequency and the number of neurons that can be run in the real-time simulation were not measured. The ionic-conductance unit consumes much more FFs than the PQN unit. Their implementation uses the coordinate rotation digital computer (CORDIC) algorithm to calculate hyperbolic functions, which is an iterative algorithm and requires long pipeline stages, thus consuming many FFs for buffers. In addition, the ionic-conductance model generally has more state variables, which also leads to an increase in FF consumption.

Table 2 shows resource utilization of the FPGA implementations of the IZH and AdEx models (Soleimani et al., 2012; Heidarpour et al., 2016; Heidarpur et al., 2019; Haghiri and Ahmadi, 2020). In Heidarpour et al. (2016) and Heidarpur et al. (2019), they used the coordinate rotational digital computer (CORDIC) algorithm to implement the quadratic and exponential terms. In Soleimani et al. (2012) and Haghiri and Ahmadi (2020), the differential equations are approximated by piecewise linear functions and power of 2-based functions to reduce circuit resources. All of these implementations are pipelined and can operate at clocks above 100 MHz. They do not require a DSP and consume much fewer LUTs than the PQN model. However, these IZH and AdEx models do not reproduce all the classes that the PQN models reproduced. They cover the RS, FS, and IB classes and also exhibit the rebound bursting seen in the LTS class. However, as far as we know, they do not cover the EB and PB classes. In addition, there is a significant gap between covering a certain class and faithfully reproducing the responses of the ionic-conductance model or an actual cell of that class.

In order for the IZH and AdEx models to replicate the responses of the ionic-conductance models with inputs of various magnitudes, as the PQN model did here, extensive modification of parameters and linear or nonlinear transformations of the input stimuli are thought to be necessary. Furthermore, as reported in the previous study (Nanami

TABLE 1 FPGA implementation results of the PQN, DSSN, and ionic-conductance models.

	Mode	LUT	FF	DSP		
PQN engine	Class II	526	485	4		
	RS exci	1,407	480	1		
	RS inhi	1,630	474	1		
	FS	1,500	463	1		
	LTS	2,060	590	1		
	IB	1,779	571	1		
	EB	1,843	486	1		
	PB	1,919	596	1		
DSSN engine (Nanami et al., 2016, 2017, 2018)	RS exci	697	109	1		
	RS inhi	690	109	1		
	FS	681	109	1		
	LTS	938	120	1		
	IB	1,265	136	1		
	EB	1,320	110	1		
	PB	3,318	228	1		
	Mode	**LUT**	**FF**	**DSP**	**N**	**F (MHz)**
PQN unit	Class II	783	819	4	9,993	100
	RS exci	1,771	963	1	9,993	100
	RS inhi	1,995	957	1	9,993	100
	FS	1,866	946	1	9,993	100
	LTS	2,458	1,111	1	9,993	100
	IB	2,177	1,092	1	9,993	100
	EB	2,210	969	1	9,993	100
	PB	2,089	1,094	1	16,384	100
ionic-conductance unit (Khoyratee et al., 2019)	RS, FS, LTS, and IB	2,360	5,551	28	500	100
Available (Xilinx Artix-7 XC7A35T)		20,800	41,600	90		

TABLE 2 FPGA resource utilization of the IZH and AdEx models.

	LUT	FF	DSP	F (MHz)
IZH (Piecewise Linear-based) (Soleimani et al., 2012)	493	617	0	241.937
IZH (CORDIC-based) (Heidarpur et al., 2019)	229	410	0	183.4
AdEx (CORDIC-Based) (Heidarpour et al., 2016)	864	1,246	0	128.7
AdEx (Power-2 Based) (Haghiri and Ahmadi, 2020)	270	643	0	196

and Kohno, 2016), the responses of the IB class of the IZH model differ significantly from those of the ionic-conductance model, whose dynamical structure of the fast subsystem varies depending on the value of the membrane potential. Therefore, in order to reproduce the IB class of the ionic-conductance model, the equation of the IZH model needs considerable modification. A previous study (Jolivet et al., 2004) demonstrated that an I&F-based model that has a quadratic term accurately reproduces the ionic-conductance model of the FS class. However, the FS is the class with a simplest dynamics. It is left unknown whether their approach can reproduce the activities of the other classes with more complex dynamical structures.

With the recent availability of a vast amount of anatomical and physiological data, data-driven SNN models (Markram et al., 2015; Bezaire et al., 2016; Ecker et al., 2020), which aim to create a complete copy of the mammalian brain, have been built. These data-driven SNN models consist of ionic-conductance models and are designed to faithfully reproduce target brain regions as precisely as possible. However, the enormous computational cost of the models precludes the casual

use of the models and the construction of larger-scale networks. For example, the model (Bezaire et al., 2016) that reproduces a CA1 in a rat hippocampus runs on a supercomputer consisting of 3,000 processors and requires 1 h of execution time for a 1-s simulation. Since SiNNs using PQN models can dramatically reduce power consumption and downsize computational systems, without severely compromising the reproducibility of neuronal activities, the PQN model is expected to be a promising modeling tool for building data-driven SNNs, and it will also expand the applicability of data-driven SNNs in both scientific and engineering fields.

The ionic-conductance models of neurons in the Pre-Bötzinger complex has been studied (Butera et al., 1999a,b; Del Negro et al., 2001). These models exhibit activities of the square-wave bursting class. We will analyze the mathematical structure behind square-wave bursting and extend the PQN model to reproduce these models. In addition, we will examine efficient implementations of PQN models in the digital ASIC. The ASIC versions of PQN units can be expected to achieve higher power efficiency. Furthermore, we will study fully automatic parameter fitting methods applicable to arbitrary neuronal cells. In the PQN unit, all the parameters are configured as circuits of adders and shifters rather than data, thus a single PQN unit can simulate only a homogeneous population of neurons. We plan to modify the PQN unit to store a number of parameters as data so that it can simulate heterogeneous populations that belong to the same neuron class but have slightly different firing properties.

Data availability statement

The original contributions presented in the study are included in the article/Supplementary material, further inquiries can be directed to the corresponding author.

Author contributions

TN performed the study. TK supervised it. Both authors contributed to the article and approved the submitted version.

Funding

This study was partially supported by the JSPS KAKENHI Grant Number 21K17849 and JST SICORP Grant Number JPMJSC22H1, Japan.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

Supplementary material

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fnins.2022.1069133/full#supplementary-material

References

Alle, H., and Geiger, J. R. P. (2006). Combined analog and action potential coding in hippocampal mossy fibers. *Science* 311, 1290–1293. doi: 10.1126/science.1119055

Arthur, J. V., and Boahen, K. A. (2011). Silicon-neuron design: a dynamical systems approach. *IEEE Trans. Circ. Syst. I* 58, 1034–1043. doi: 10.1109/TCSI.2010.2089556

Benda, J., Longtin, A., and Maler, L. (2005). Spike-frequency adaptation separates transient communication signals from background oscillations. *J. Neurosci.* 25, 2312–2321. doi: 10.1523/JNEUROSCI.4795-04.2005

Bezaire, M. J., Raikov, I., Burk, K., Vyas, D., and Soltesz, I. (2016). Interneuronal mechanisms of hippocampal theta oscillations in a full-scale model of the rodent ca1 circuit. *Elife* 5, 1–106. doi: 10.7554/eLife.18566

Brette, R., and Gerstner, W. (2005). Adaptive exponential integrate-and-fire model as an effective description of neuronal activity. *J. Neurophysiol.* 94, 3637–3642. doi: 10.1152/jn.00686.2005

Brink, S., Nease, S., and Hasler, P. E. (2013). Computing with networks of spiking neurons on a biophysically motivated floating-gate based neuromorphic integrated circuit. *Neural Netw.* 45, 39–49. doi: 10.1016/j.neunet.2013.02.011

Butera, R. J., Rinzel, J., and Smith, J. C. (1999a). Models of respiratory rhythm generation in the pre-botzinger complex. i. bursting pacemaker neurons. *J. Neurophysiol.* 82, 382–397. doi: 10.1152/jn.1999.82.1.382

Butera, R. J., Rinzel, J., and Smith, J. C. (1999b). Models of respiratory rhythm generation in the pre-botzinger complex. ii. populations of coupled pacemaker neurons. *J. Neurophysiol.* 82, 398–415. doi: 10.1152/jn.1999.82.1.398

Cassidy, A., Andreou, A. G., and Georgiou, J. (2011). "Design of a one million neuron single fpga neuromorphic system for real-time multimodal scene analysis," in *2011 45th Annual Conference on Information Sciences and Systems* (Baltimore, MD), 1–6.

Cassidy, A. S., Georgiou, J., and Andreou, A. G. (2013). Design of silicon brains in the nano-cmos era: spiking neurons, learning synapses and neural architecture optimization. *Neural Netw.* 45, 4–26. doi: 10.1016/j.neunet.2013.05.011

Cunningham, M. O., Whittington, M. A., Bibbig, A., Roopun, A., LeBeau, F. E. N., Vogt, A., et al. (2004). A role for fast rhythmic bursting neurons in cortical gamma oscillations *in vitro*. *Proc. Natl. Acad. Sci. U.S.A*. 101, 7152–7157. doi: 10.1073/pnas.0402060101

Davies, M., Srinivasa, N., Lin, T., Chinya, G., Cao, Y., Choday, S. H., et al. (2018). Loihi: a neuromorphic manycore processor with on-chip learning. *IEEE Micro* 38, 82–99. doi: 10.1109/MM.2018.112130359

Del Negro, C. A., Johnson, S. M., Butera, R. J., and Smith, J. C. (2001). Models of respiratory rhythm generation in the pre-botzinger complex. iii. experimental tests of model predictions. *J. Neurophysiol.* 86, 59–74. doi: 10.1152/jn.2001.86.1.59

Ecker, A., Romani, A., Saray, S., Kali, S., Migliore, M., Falck, J., et al. (2020). Data-driven integration of hippocampal ca1 synaptic physiology in silico. *Hippocampus* 30, 1129–1145. doi: 10.1002/hipo.23220

FitzHugh, R. (1961). Impulses and physiological states in theoretical models of nerve membrane. *Biophys. J.* 1, 445–466. doi: 10.1016/S0006-3495(61)86902-6

Grassia, F., Buhry, L., Levi, T., Tomas, J., Destexhe, A., and Saighi, S. (2011). Tunable neuromimetic integrated system for emulating cortical neuron models. *Front. Neurosci.* 5, 134. doi: 10.3389/fnins.2011.00134

Grillner, S. (2007). Biological pattern generation: the cellular and computational logic of networks in motion. *Neuron* 52, 751–766. doi: 10.1016/j.neuron.2006.11.008

Haghiri, S., and Ahmadi, A. (2020). A novel digital realization of adex neuron model. *IEEE Trans. Circ. Syst. II* 67, 1444–1448. doi: 10.1109/TCSII.2019.2938180

Hansel, D., Mato, G., and Meunier, C. (1995). Synchrony in excitatory neural networks. *Neural Comput*. 7, 307–337. doi: 10.1162/neco.1995.7.2.307

Heidarpour, M., Ahmadi, A., and Rashidzadeh, R. (2016). A cordic based digital hardware for adaptive exponential integrate and fire neuron. *IEEE Trans. Circ. Syst. I* 63, 1986–1996. doi: 10.1109/TCSI.2016.2598161

Heidarpur, M., Ahmadi, A., Ahmadi, M., and Rahimi Azghadi, M. (2019). Cordic-snn: on-fpga stdp learning with izhikevich neurons. *IEEE Trans. Circ. Syst. I* 66, 2651–2661. doi: 10.1109/TCSI.2019.2899356

Hindmarsh, J. L., and Rose, R. M. (1984). A model of neuronal bursting using tree coupled first order differential equations. *Philos. Trans. R. Soc. Lond. B* 221, 87–102. doi: 10.1098/rspb.1984.0024

Hodgkin, A. L. (1948). The local electric changes associated with repetitive action in a non-medullated axon. *J. Physiol.* 107, 165–181. doi: 10.1113/jphysiol.1948.sp004260

Hodgkin, A. L., and Huxley, A. F. (1952). A quantitative description of membrane current and its application to conduction and excitation in nerve. *J. Physiol.* 117, 500–544. doi: 10.1113/jphysiol.1952.sp004764

Izhikevich, E. (2007). *Dynamical Systems in Neuroscience*. Cambridge, MA: MIT Press.

Izhikevich, E. M. (2003). Simple model of spiking neurons. *IEEE Trans. Neural Netw*. 14, 1569–1572. doi: 10.1109/TNN.2003.820440

Jolivet, R., Lewis, T. J., and Gerstner, W. (2004). Generalized integrate-and-fire models of neuronal activity approximate spike trains of a detailed model to a high degree of accuracy. *J. Neurophysiol.* 92, 959–976. doi: 10.1152/jn.00190.2004

Kepler, T., Abbott, L., and Marder, E. (1992). Reduction of conductance-based neuron models. *Biol. Cybern*. 66, 381–387. doi: 10.1007/BF00197717

Khoyratee, F., Grassia, F., Saighi, S., and Levi, T. (2019). Optimized real-time biomimetic neural network on fpga for bio-hybridization. *Front. Neurosci.* 13, 377. doi: 10.3389/fnins.2019.00377

Kohno, T., and Aihara, K. (2016). "A three-variable ultralow-power analog silicon neuron circuit," in *2016 International Symposium on Nonlinear Theory and its Applications* (Yugawara), 190–193.

Li, J., Katori, Y., and Kohno, T. (2012). An fpga-based silicon neuronal network with selectable excitability silicon neurons. *Front. Neurosci.* 6, 183. doi: 10.3389/fnins.2012.00183

Markram, H., Muller, E., Ramaswamy, S., Reimann, M., Abdellah, M., Aguado, C., et al. (2015). Reconstruction and simulation of neocortical microcircuitry. *Cell* 163, 456–492. doi: 10.1016/j.cell.2015.09.029

Merolla, P., Arthur, J., Alvarez-Icaza, R., Cassidy, A., Sawada, J., Akopyan, F., et al. (2014). A million spiking-neuron integrated circuit with a scalable communication network and interface. *Science* 345, 668–673. doi: 10.1126/science.1254642

Miura, K., Okada, M., and Amari, S.-,i. (2006). Estimating spiking irregularities under changing environments. *Neural Comput*. 18, 2359–2386. doi: 10.1162/neco.2006.18.10.2359

Moradi, S., Qiao, N., Stefanini, F., and Indiveri, G. (2018). A scalable multicore architecture with heterogeneous memory structures for dynamic neuromorphic asynchronous processors (dynaps). *IEEE Trans. Biomed. Circ. Syst*. 12, 106–122. doi: 10.1109/TBCAS.2017.2759700

Nagumo, J. S., Arimoto, S., and Yoshizawa, S. (1962). An active pulse transmission line simulating nerve axon. *Proc. IRE* 50, 2061–2071. doi: 10.1109/JRPROC.1962.288235

Nanami, T., Grassia, F., and Kohno, T. (2017). A parameter optimization method for digital spiking silicon neuron model. *J. Rob. Network. Artif. Life* 4, 97–101. doi: 10.2991/jrnal.2017.4.1.21

Nanami, T., Grassia, F., and Kohno, T. (2018). A metaheuristic approach for parameter fitting in digital spiking silicon neuron model. *J. Rob. Network. Artif. Life* 5, 32–36. doi: 10.2991/jrnal.2018.5.1.8

Nanami, T., and Kohno, T. (2016). Simple cortical and thalamic neuron models for digital arithmetic circuit implementation. *Front. Neurosci*. 10, 181. doi: 10.3389/fnins.2016.00181

Nanami, T., Aihara, K., and Kohno, T. (2016). "Elliptic and parabolic bursting in a digital silicon neuron model," in *2016 International Symposium on Nonlinear Theory and Its Applications* (Yugawara), 198–201.

Plant, R. E. (1981). Bifurcation and resonance in a model for bursting nerve cells. *J. Math. Biol*. 67, 15–32. doi: 10.1007/BF00275821

Pospischil, M., Toledo-Rodriguez, M., Monier, C., Piwkowska, Z., Bal, T., Fregnac, Y., et al. (2008). Minimal hodgkin-huxley type models for different classes of cortical and thalamic neurons. *Biol. Cybern*. 99, 427–441. doi: 10.1007/s00422-008-0263-8

Qiao, N., Mostafa, H., Corradi, F., Osswald, M., Stefanini, F., Sumislawska, D., et al. (2015). A reconfigurable on-line learning spiking neuromorphic processor comprising 256 neurons and 128k synapses. *Front. Neurosci.* 9, 141. doi: 10.3389/fnins.2015.00141

Rinzel, J., and Ermentrout, B. (1998). "Analysis of neural excitability and oscillations," in *Methods in Neuronal Modeling, 2nd Edn*, chapter 7, eds C. Koch and I. Segev (Cambridge, MA: MIT Press), 251–292.

Rubino, A., Livanelioglu, C., Qiao, N., Payvand, M., and Indiveri, G. (2021). Ultra-low-power fdsoi neural circuits for extreme-edge neuromorphic intelligence. *IEEE Trans. Circ. Syst*. 68, 45–56. doi: 10.1109/TCSI.2020.3035575

Schemmel, J., Bruderle, D., Grubl, A., Hock, M., Meier, K., and Millner, S. (2010). "A wafer-scale neuromorphic hardware system for large-scale neural modeling," in *2010 IEEE International Symposium on Circuits and Systems (ISCAS)* (Paris: IEEE), 1947–1950.

Schemmel, J., Billaudelle, S., Dauer, P., and Weis, J. (2021). *Accelerated Analog Neuromorphic Computing*. Berlin; Heidelberg: Splinger.

Schutter, E. D., and Bower, J. M. (1994). An active membrane model of the cerebellar purkinje cell ii. simulation of synaptic responses. *J. Neurophysiol.* 71, 375–400. doi: 10.1152/jn.1994.71.1.375

Shinomoto, S., Kim, H., Shimokawa, T., Matsuno, N., Funahashi, S., Shima, K., et al. (2009). Relating neuronal firing patterns to functional differentiation of cerebral cortex. *PLoS Comput. Biol.* 5, e1000433. doi: 10.1371/journal.pcbi.1000433

Shinomoto, S., Miura, K., and Koyama, S. (2005a). A measure of local variation of inter-spike intervals. *Biosystems* 79, 67–72. doi: 10.1016/j.biosystems.2004.09.023

Shinomoto, S., Miyazaki, Y., Tamura, H., and Fujita, I. (2005b). Regional and laminar differences in *in vivo* firing patterns of primate cortical neurons. *J. Neurophysiol.* 94, 567–575. doi: 10.1152/jn.00896.2004

Shinomoto, S., Shima, K., and Tanji, J. (2003). Differences in spiking patterns among cortical neurons. *Neural Comput*. 15, 2823–2842. doi: 10.1162/089976603322518759

Soleimani, H., Ahmadi, A., and Bavandpour, M. (2012). Biologically inspired spiking neurons: piecewise linear models and digital implementation. *IEEE Trans. Circ. Syst*. 59, 2991–3004. doi: 10.1109/TCSI.2012.2206463

Thomas, D. B., and Luk, W. (2009). "FPGA accelerated simulation of biologically plausible spiking neural networks," in *2009 17th IEEE Symposium on Field Programmable Custom Computing Machines*, eds K. L. Pocek and D. A. Buell (Napa, CA: IEEE Computer Society), 45–52.

Tikidji-Hamburyan, R. A., and Colonnese, M. T. (2021). Polynomial, piecewise-linear, step (pls): a simple, scalable, and efficient framework for modeling neurons. *Front. Neuroinform*. 15, 642933. doi: 10.3389/fninf.2021.642933

Wang, X.-J. (1993). Ionic basis for intrinsic 40 hz neuronal oscillations. *Neuroreport* 5, 221–224. doi: 10.1097/00001756-199312000-00008