This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 828841

Project Acronym:	ChipAI
Project Title:	Energy-efficient and high-bandwidth neuromorphic nanophotonic Chips for Artificial Intelligence systems
Project Number:	828841
Topic:	FETOPEN-01-2018-2019-2020
Type of Action:	RIA

Tutorial

Neuromorphic system simulator free-software

Version and date	V1.0, 27/01/2022
Author name(s):	Ignacio Ortega Piwonka, João Lourenço, José Figueiredo, Artur Andrishak, Bruno Romeira, Julien Javaloyes

Table of Contents

Executive Summary

In this document, the details of a collection of three apps that simulate the dynamics of neuromorphic nanophotonic devices are presented. In particular, we are interested in reproducing the generation and transmission of excitable optical pulses in artificial nano-optoelectronic spiking neurons. The structure of the latter is based upon nanoscale resonant tunnelling diode (RTD) elements coupled to either laser diodes (LD) and photodetector (PD) components forming respectively spiking transmitter (TX) and receiver (RX) modules. The three apps developed account respectively for a single RTD-LD node acting as a spiking optoelectronic neuron, a network made of two RTD-LD nodes connected via a photodetector (PD) and at last another two-node network with an additional (recurrent) connection in the reverse direction with a delay line. These apps are available in the ChipAI Project website (https://www.chipai.eu/neuromorphic-simulator) to be used by other researchers/stakeholders for them to have an understanding on how optical pulses (spikes) are generated in these systems, and how these are affected by the systems' parameters and specifications. These apps will therefore contribute to disseminate in a practical, interactive way the project's research concepts and outputs, as well as be a tool for other scientists to use them in their own research.

The document is organized as follows. First, the theoretical models that reproduce the dynamics of the nano-optoelectronic devices are explained, as well as the proper configuration to elicit excitability and pulse generation in them. Next, the numerical methods that integrate those models and that are internally implemented in the apps are discussed. Finally, the graphical interface and functions of the apps are detailed, and some examples of performance are provided.

1. Introduction

In the last few years, the fields of Artificial Intelligence (AI) and Machine Learning (ML) have experienced a significant growth. New revolutionary algorithms are being developed which are rapidly finding applications in both industry and academic research. A main challenge these methodologies have met, however, is the fact that they have been traditionally implemented using conventional computers based on the Von Neumann architecture [1] (and built upon CMOS-based electronic circuitry technology [2]). This was not purposely-designed to work with the neural network algorithms AI and ML systems rely on, resulting on high-energy consumption, long waiting times and bandwidth limitations.

Several efforts are being carried out nowadays to develop new neuromorphic computing processors whose fundamental units –like biological neurons of the brain– process and transmit bits of information in the form of spikes that are generated at arbitrary times. The term '*neuromorphic'* is used here as these processors aim at mimicking the way biological neural circuits operate in the brain. In addition to a more continuous nature, and therefore, more expressiveness of information than binary coding, spike signalling is robust in the presence of noise and ensures low power consumption since energy is spent only at very specific and short moments of time when spikes are triggered.

Examples of neuromorphic processors include the IBM TrueNorth™ [3], the Intel Loihi 2™ [4], the BRAIN Initiative™ [5] and the Innatera™ chip [6]. This generation of neuromorphic processors have been developed all using electronic-based technologies. The ChipAI Project intends to contribute to the field of neuromorphic computing with novel nano-optoelectronic implementations based on double barrier quantum well resonant tunnelling diodes (DBQW-RTD), laser diodes (LD) and Photodetectors (PD). ChipAI therefore will permit to capitalise on the inherent benefits of using light signals (instead of electronic currents) for operation, including high-speed, reduced energy consumption, high bandwidths, etc. In ChipAI's vision the RTD acts as the *excitable* part of the neuron which, very much like their biological counterparts, fires a spike when perturbed with an external stimulus, provided that such stimulus is above a certain energy threshold. Otherwise, the RTD (neuron) does not fire [7]. The ChipAI team has carried out extensive research on the dynamics on both RTD and LD-based circuits, and contributed with theoretical and numerical results [8,9,10] in order to understand the suitable configurations for nano-optoelectronic neurons to behave as an excitable generator of optical pulses (spikes).

In this document, three computer applications developed as part of the ChipAI Project that simulate spike generation and transmission in RTD-LD-based optoelectronic devices, are presented. These apps respectively account for a single optoelectronic node, a network made of two optoelectronic nodes connected via a photodetector (PD), and the same network with an additional connection in the reverse direction in the form of a delay line. Delayed dynamics allow using a single neuron in neuromorphic algorithms as multiple virtual nodes at different intervals of time [6]. The apps are intended to be available on the ChipAI Project website (https://www.chipai.eu/neuromorphic-simulator) and allow users to specify physical and numerical parameters, run simulations and export generated data as spread sheet files.

The document is structured as follows. In chapter 2, the theoretical models that reproduce the dynamics of the devices emulated by the apps are presented and their configuration and behaviour as excitable optoelectronic spike generators are discussed. In chapter 3, the numerical methods implemented in the apps to simulate the theoretical models are explained. In chapter 4, the graphical user interfaces (GUI) and functionalities of the apps are discussed, together with some performance examples. Finally, Chapter 5 presents the conclusions and prospective steps in the app development.

2. Neuromorphic circuits modelling

In this section, the neuromorphic circuits to be simulated by the apps are discussed, their mathematical models are described and the proper configurations for spike generation and propagation are demonstrated.

2.1 Neuromorphic transmitter node

An optoelectronic transmitter is made of a double barrier quantum well resonant tunnelling diode (DBQW-RTD or RTD for short) connected to a nanolaser diode (nanoLD or LD). The RTD functions as an excitable system that responds to an external (above-threshold) perturbation (e.g., a square voltage pulse) with a single ultrafast current spike. This current spike is injected into the LD which translates it into an optical spike. This permits to use light spikes for signalling as well as to avoid the power dissipation typical of electrical interconnections. In our design, the RTD and the LD are monolithically integrated in a nanopillar structure with highly reduced footprint, as illustrated in Fig. 1.

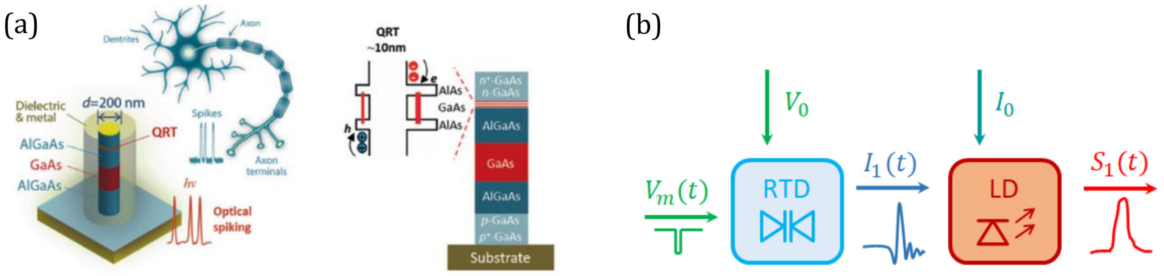


Fig. 1. (a) Schematics of the neuromorphic RTD-LD nanopillar (figure taken from [11]). **(b)** Excitable optical spike generation in the RTD-LD optoelectronic circuit.

The RTD-LD system is described by Eqs. (1-4) below, where Eqs. (1,2) and (3,4) describe respectively the dynamics of the RTD and the LD,

$$C\frac{dV}{dt} = I - f(V) + \sigma\xi_V(t), \tag{1}$$

$$L\frac{dI}{dt} = V_0 + V_m(t) - V - RI, \tag{2}$$

$$\frac{dS}{dt} = \left(\gamma_m(N - N_0) - \frac{1}{\tau_p}\right)S + \gamma_m N + \sqrt{\gamma_m N S}\,\xi_S(t), \tag{3}$$

$$\frac{dN}{dt} = \frac{I_0 + I}{q_e} - (\gamma_m + \gamma_l + \gamma_{nr})N - \gamma_m(N - N_0)S. \tag{4}$$

2.1.1 *Electrical part*

Eqs. (1,2) account for the electrical part of the circuit and are derived from Kirchhoff laws. Here, V and I are the voltage and current across the DBQW. R, L, C are the circuit's intrinsic resistance, inductance and capacitance, respectively. For the rest of this document, these values will be set at $R = 10\ \Omega$, $C = 2$ fF and $L = 126$ nH (unless stated otherwise). V_0 is the input bias voltage and $V_m(t)$ is an input modulation voltage (e.g., a train of square voltage pulses). The fluctuations in the systems are represented by additive noise, with $\xi_V(t)$ being a time-uncorrelated white noise function and σ being the noise intensity. $f(V)$ is the nonlinear, N-shaped current-voltage characteristic of the RTD. Schulman [12] provides an analytical expression for this curve,

$$f(V) = a\ln\left(\frac{1+e^{(b-c+n_1V)q/kT}}{1+e^{(b-c-n_1V)q/kT}}\right)\left[\frac{\pi}{2}+\tan^{-1}\left(\frac{c-n_1V}{d}\right)\right]+h\left(e^{n_2qV/kT}-1\right) \tag{5}$$

The current-voltage characteristic $f(V)$ defined in Eq. 5 has a single region of negative differential conductance embedded between two regions of positive differential conductance. These regions are delimited by peak and valley points. The shape of the curve is tuned by the parameters a, b, c, d, n_1, n_2, h as well as the electron charge, $q = 1.602 \times 10^{-19}$ C, the Boltzmann's constant, $k = 1.38 \times 10^{-23}$ J/°K, and the room temperature, $T = 300$°K.

Two current-voltage characteristics will be considered in this document; an overall smooth curve and a sharp curve with a pointed peak followed by a sudden drop in the current and a wide valley. The curves are illustrated in Fig. 2, together with the voltages at the peak and valley points. The parameters that reproduce the curves are shown in Table 1.

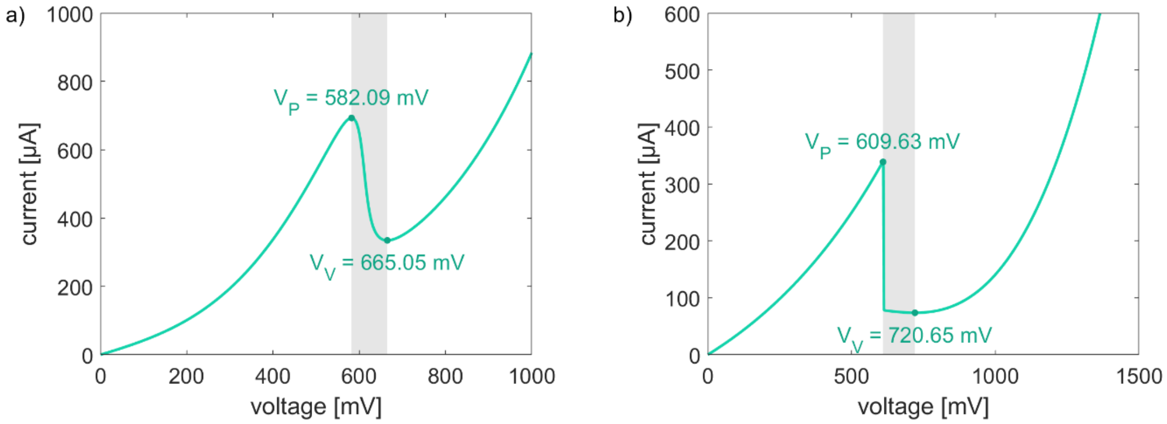


Fig. 2. Examples of RTD current-voltage characteristic curves defined according to Eq. 5. **(a)** Smooth curve. **(b)** Sharp curve. The parameter values for each curve are summarized in Table 1. The region of negative differential conductance is coloured in grey. Voltages at the peak and valley point are also included.

	a [µA]	b [mV]	c [mV]	d [mV]	n_1	n_2	h [µA]
a) smooth curve	137.5	33	113	2.8	0.185	0.00845	34.2
b) sharp curve	–55	33	113	–2800	0.185	0.045	180

Table 1. Parameter values of the RTD current-voltage characteristic curves illustrated in Fig. 1.1.

It is well known [13,14,15,16] that an RTD biased in the region of negative differential conductance exhibits self-oscillations (i.e., stable limit cycle). Otherwise, the RTD responds with a fixed output given by the intersection of the RTD current-voltage characteristic and the circuit's load line, $V_0 - V - RI = 0$. The details of the transition between these two responses are explained in the reference [8]. These works also state that if the stiffness coefficient $\mu = \sqrt{C/L}$ is sufficiently small, the self-oscillation response exhibits then slow-fast dynamics and a stiff limit cycle orbit resulting in periodical spike firing (Fig. 2).

The RTD is configured as an excitable spike generator by setting its bias operation point in the proximity to the region of negative conductance, yet slightly outside, and injecting a positive or negative square voltage pulse, depending on whether the bias is set close to the peak or the valley points in the I-V characteristic of the device. This is equivalent to displacing briefly the load line into the region of negative conductance, where spike firing is produced. Fig. 4 summarizes the responses of a valley-biased RTD to a variety of negative square voltage pulses with 50 ps in temporal width and amplitudes ranging from 10 to 100 mV. There is a threshold for the voltage pulse amplitude at about 50 mV beyond which the RTD responds with a single spike firing event before returning to the stable fixed point. On the other hand, responses to sub-threshold voltage pulses are negligible. This confers to the RTD response the all-or-none characteristic typical of excitable systems [7]. Similarly, a peak-biased RTD responds to a positive square voltage pulse with a negative current spike, provided that such voltage pulse has a large enough amplitude. For both biases, the current spike has an amplitude of about 280 μA, determined by the peak and valley of the I-V characteristic. The duration of the excitable spike is roughly estimated at 250 ps.

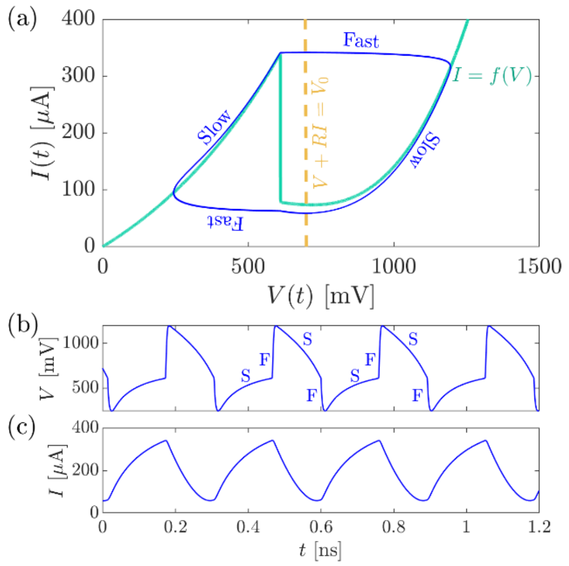


Fig. 3. Simulation of Eqs. (1,2) with $V_0 = 700$ mV, $\sigma = 0$, sharp current-voltage characteristic and no modulation voltage (i.e., $V_m = 0$). The system exhibits self-oscillations and stages of slow and fast dynamics. **(a)** Phase space. **(b)** Output voltage over time. **(c)** Output current over time. The system's response, the I-V characteristic and the circuit's load line are coloured in blue, cyan and yellow, respectively.

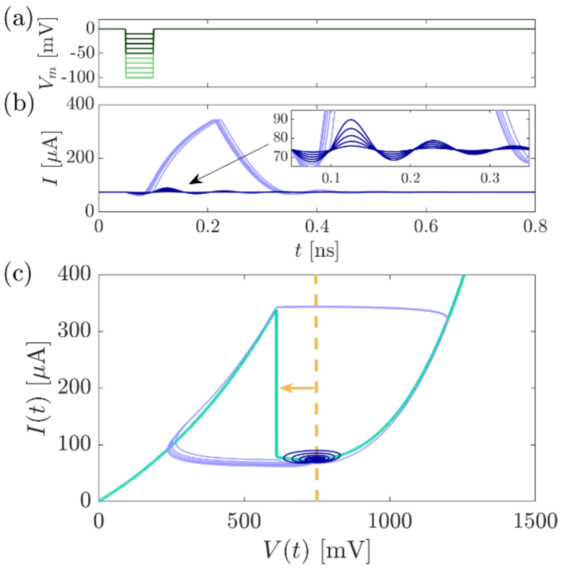


Fig. 4. Simulations of Eqs. (1,2) with $V_0 = 750$ mV, $\sigma = 0$, sharp current-voltage characteristic and modulated with 50 ps long negative square voltage pulses of different amplitudes. **(a)** Pulse profiles. **(b)** System's current response to each input pulse. **(c)** System's response to input pulses on the phase space. Sub (supra) threshold pulses and their responses are coloured in dark (light) shades of green and blue, respectively.

2.1.2 Optical part

Eqs. (3,4) account for the optical part of the circuit and are the standard rate equations that describe the dynamics for the photon number and carrier population in an LD [17]. S is the photon number and N is the carrier number. The parameter values are set to describe a low-Q metal cavity nanolaser and will be fixed for the rest of this document (unless stated otherwise). $N_0 = 5\times 10^5$ is the transparency carrier number. $\tau_p = 5\times 10^{-3}$ s is the photon lifetime. $\gamma_m = 10^7$ s^{-1} and $\gamma_l = 10^9$ s^{-1} are the Purcell enhanced spontaneous emission coefficients into the lasing mode and leaky mode, respectively. $\gamma_{nr} = 2\times 10^9$ s^{-1} is the non-radiative spontaneous emission coefficient. The fluctuations in the system are represented by a nonlinear multiplicative noise term, with $\xi_S(t)$ being a time-uncorrelated white noise function. I_0 is an additional input bias current injected into the nanolaser.

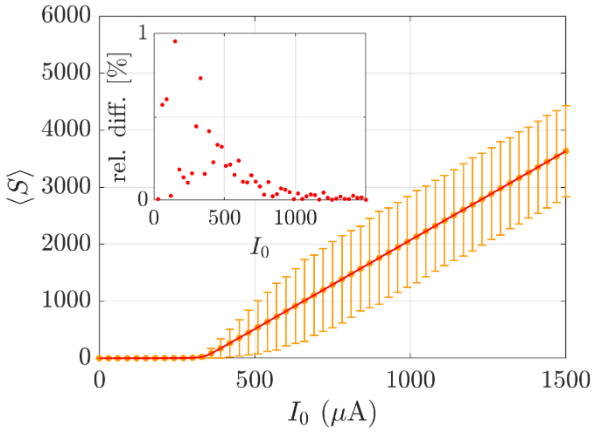


Fig. 5. Mean photon number in the LD with no RTD modulation (i.e., Eqs. (3,4) with $I = 0$) as a function of input bias current. Red line: stable equilibrium in the deterministic system (i.e., $\xi_S = 0$). Orange dots: time-average estimated from numerical simulations of the stochastic system, with error bars accounting for standard deviation. Inset: relative difference between both estimations.

The LD connected to a DC current source I_0 (i.e., no RTD current $I(t)$) emits light provided that I_0 is above a threshold value which, for the parameters chosen above, is at $I_{th} = 337.54$ µA. As illustrated in Fig. 5, the emission intensity is directly proportional to the current difference (i.e., $\langle S \rangle \sim I_0 - I_{th}$) while there is no emission if $I_0 < I_{th}$. This type of transition is known as *transcritical bifurcation* [18].

2.1.3 Integrated optoelectronic neuron

An LD injected with current spikes generated by a valley-biased RTD will respond with optical pulses (spikes) provided that the input bias current I_0 is properly tuned. The nature of the output optical spikes will depend on the characteristics of both the RTD and the LD. The idea is to displace the RTD output current $I(t)$ in such a way that its rest value remains under the LD threshold current I_{th} while the current spike surpasses it. As a result, the LD is expected to emit only when the current spike takes place.

Figs. 6 and 7 show the process' steps within the circuit with the sharp RTD current-voltage characteristic. 500 simulations are run in each figure. As a consequence of the noise in the system, the responses are non-identical. The moments of time when a spike is triggered in each response, as well as their profiles over time, differ slightly. On that regard, we refer to the responses as *incoherent*. A supra-threshold negative square voltage pulse is injected into the valley-biased RTD, which responds with a positive current spike. As shown in Fig. 4 (b), the rest value and maximal value of $I(t)$ are approx. 75 and 340 mV, respectively, given by the peak and valley of the I-V characteristic. These values are under or slightly above the LD threshold, $I_{th} = 337.54$. Therefore, this spike will not elicit an emission from the LD. Hence, an additional input bias current is required. In Fig. 6, a bias $I_0 = 214$ µA is added to $I(t)$, increasing the rest and maximal values of the total current injected up to 290 and 570 µA, respectively. Expectedly, the LD responds to this configuration with an optical pulse (spike) at its output. This pulse is very short, with a FWHM of about 50 ps. Its response delay (relative to the square voltage pulse) is approximately 150 ps. These features are changed when a different bias current is chosen. Fig. 7 shows the LD response to the same RTD configuration and $I_0 = 250$ µA. With an increased bias, the rest value of the total current injected into the LD is now closer to the threshold. The LD response pulse has now a shorter delay (~100 ps) but longer FWHM, higher amplitude and hence higher energy. The dramatic change in the delay response is consequence of a phenomenon called *critical slowing down*, typical in transcritical bifurcations [19,20,21]. In simple terms, the farther the input current is initially under the threshold, the longer the LD takes to emit once the input current surpasses the threshold. All the above findings are similar when the smooth I-V characteristic is chosen.

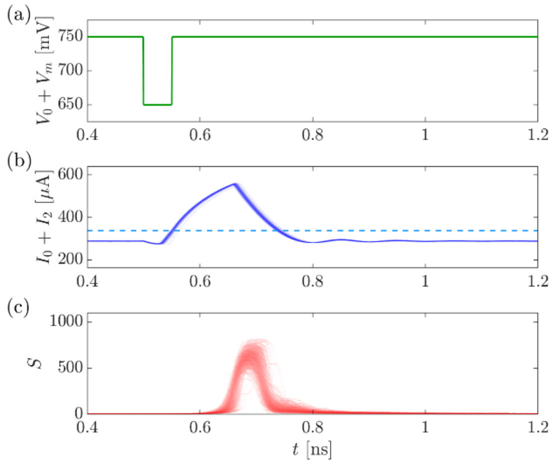


Fig. 6. 500 Simulations of Eqs. (1-4) with $I_0 = 214$ μA, $V_0 = 750$ mV, $\sigma = 2.5 \times 10^{-12}$ [SI], sharp current-voltage characteristic and modulated with a single 50 ps long, 100 mV deep negative voltage pulse. **(a)** RTD total input voltage. **(b)** LD total input current. The dashed line marks the LD current threshold. **(c)** LD optical output.

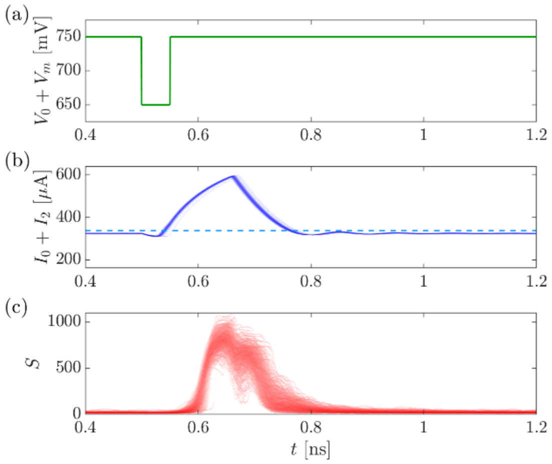


Fig. 7. 500 Simulations of Eqs. (1-4) with $I_0 = 250$ μA, $V_0 = 750$ mV, $\sigma = 2.5 \times 10^{-12}$ [SI], sharp current-voltage characteristic and modulated with a single 50 ps long, 100 mV deep negative voltage pulse. **(a)** RTD total input voltage. **(b)** LD total input current. The dashed line marks the LD current threshold. **(c)** LD optical output.

2.2 Neuromorphic transmitter-receiver (Tx-Rx) network

In this section, the propagation of optical pulses (spikes) along two neuromorphic RTD-LD nodes connected via a photodetector (PD) is studied. The full system is described by the following equations,

$$C\frac{dV_1}{dt} = I_1 - f(V_1) + \sigma\xi_V^1(t), \tag{6}$$

$$L\frac{dI_1}{dt} = V_0 + V_m(t) - V_1 - RI_1, \tag{7}$$

$$\frac{dS_1}{dt} = \left(\gamma_m(N_1 - N_0) - \frac{1}{\tau_p}\right)S_1 + \gamma_m N_1 + \sqrt{\gamma_m N_1 S_1}\xi_S^1(t), \tag{8}$$

$$\frac{dN_1}{dt} = \frac{I_0 + I_1}{q_e} - (\gamma_m + \gamma_l + \gamma_{nr})N_1 - \gamma_m(N_1 - N_0)S_1, \tag{9}$$

$$C\frac{dV_2}{dt} = I_2 - f(V_2) + \kappa S_1(t) + \sigma\xi_V^2(t), \tag{10}$$

$$L\frac{dI_2}{dt} = V_0 - V_2 - RI_2, \tag{11}$$

$$\frac{dS_2}{dt} = \left(\gamma_m(N_2 - N_0) - \frac{1}{\tau_p}\right)S_2 + \gamma_m N_2 + \sqrt{\gamma_m N_2 S_2}\xi_S^2(t), \tag{12}$$

$$\frac{dN_2}{dt} = \frac{I_0 + I_2}{q_e} - (\gamma_m + \gamma_l + \gamma_{nr})N_2 - \gamma_m(N_2 - N_0)S_2. \tag{13}$$

Eqs. (6-9) account for the first node, referred to as *transmitter* (Tx), while Eqs. (10-13) account for the second node, called *receiver* (Rx). Fig. 8 illustrates the transmission of optical pulses in the network. The optical output from the Tx LD ($S_1(t)$) is injected into the Rx PD, which in response returns a current $\kappa S_1(t)$ injected into the Rx RTD. In order to generate and transmit excitable optical pulses, both RTDs are biased close to the valley of the current-voltage characteristic (V_0) and both LDs are biased under the emission threshold (I_0). The Tx RTD is perturbed with a negative square voltage pulse, like in subsection 2.1.3. Under this configuration, the Tx RTD responds with a current spike, to which the Tx LD responds with an optical pulse. The duration and length of the optical pulse are tuned by the value of I_0 (see Figs. 6 and 7).

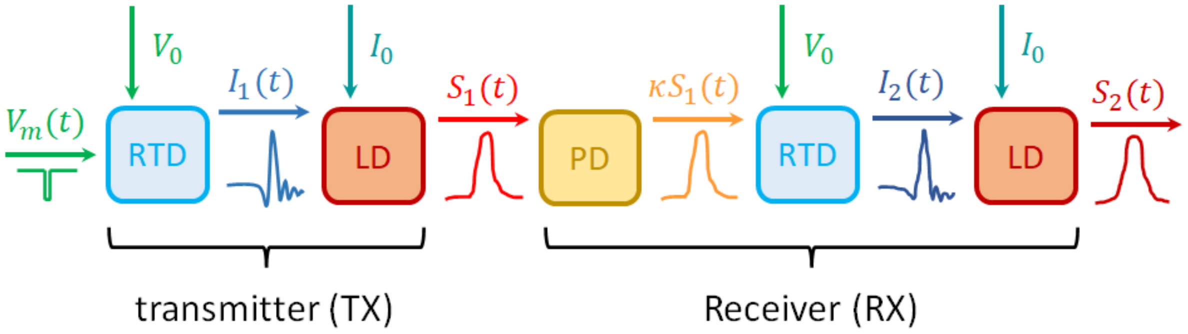


Fig. 8. Generation and transmission of optical spikes in the two-node RTD-LD optoelectronic network.

The PD conversion coefficient κ must be chosen properly to elicit an excitable response from the Rx RTD. If κ is too small, the Rx PD current pulse will be sub-threshold. If κ is too big, the fluctuations in the rest state may elicit excitable responses randomly at undesired moments. The proper value of κ depends of the shape of the current-voltage characteristic. For the smooth and sharp curves defined in subsection 2.1.1, proper values for pulse transmission are κ = 0.02 μA and κ = 0.05 μA, respectively.

Figs. 9 (10) shows that for I_0 = 214 μA (250 μA), short (long) optical pulses are transmitted. The optical pulse from the Tx is injected into the Rx PD, which responds with a suprathreshold current pulse. Since the Rx RTD is also biased close to the valley, it also responds with a spike. These spikes are almost identical to those from the Tx RTD (Figs. 6 and 7), given the all-or-nothing nature of the excitable response. These current spikes however, are less coherent because the

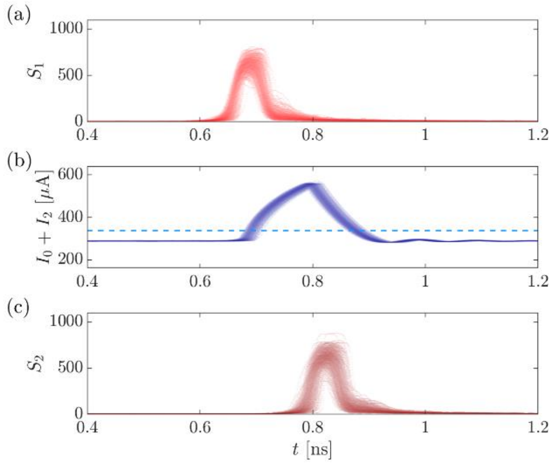


Fig. 9. 500 Simulations of Eqs. (6-13) with I_0 = 214 μA, V_0 = 750 mV, κ = 0.05 μA, $\sigma = 2.5 \times 10^{-12}$ [SI], sharp current-voltage characteristic and modulated with a single 50 ps long, 100 mV deep negative voltage pulse. **(a)** transmitter LD optical output. **(b)** Receiver LD total input current. The dashed line marks the LD current threshold. **(c)** Receiver LD optical output.

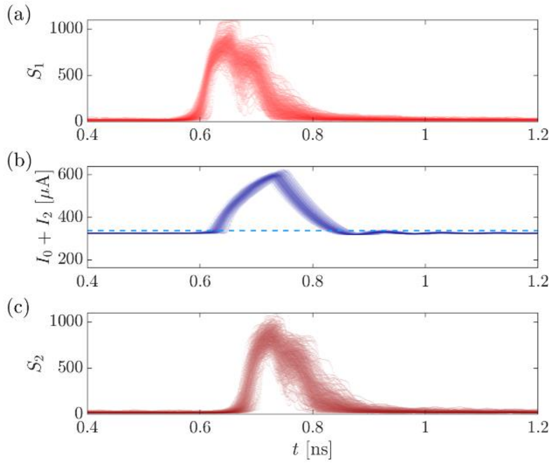


Fig. 10. 500 Simulations of Eqs. (6-13) with I_0 = 250 μA, V_0 = 750 mV, κ = 0.05 μA, $\sigma = 2.5 \times 10^{-12}$ [SI], sharp current-voltage characteristic and modulated with a single 50 ps long, 100 mV deep negative voltage pulse. **(a)** transmitter LD optical output. **(b)** Receiver LD total input current. The dashed line marks the LD current threshold. **(c)** Receiver LD optical output.

optical input form the Tx is incoherent itself (in contrast with the noiseless voltage pulse used to drive the Tx, identical in all simulations). The Rx LD responds to these current spikes with the same optical pulses that left the Tx LD (since both LDs are biased at the same current I_0). The Tx optical response seems to be slightly less coherent than that from the Rx. More notoriously, the response delay in the two-node network is longer than that in the single node (since the transmission of pulses involves more steps and more delays are accumulated) and it is longer for short pulses.

2.3 Neuromorphic transmitter node with bidirectional communication

In this section, a delay line is added to the RTD-PD two-node network. This delay line injects the optical output from the Rx LD back into the Tx RTD via a photodetector (PD). This is modelled by adding a term $\kappa S_2(t-\tau)$ in Eq. 6,

$$C\frac{dV_1}{dt} = I_1 - f(V_1) + \kappa S_2(t-\tau) + \sigma\xi_V^1(t), \tag{14}$$

$$L\frac{dI_1}{dt} = V_0 + V_m(t) - V_1 - RI_1, \tag{15}$$

$$\frac{dS_1}{dt} = \left(\gamma_m(N_1 - N_0) - \frac{1}{\tau_p}\right)S_1 + \gamma_m N_1 + \sqrt{\gamma_m N_1 S_1}\xi_S^1(t), \tag{16}$$

$$\frac{dN_1}{dt} = \frac{I_0 + I_1}{q_e} - (\gamma_m + \gamma_l + \gamma_{nr})N_1 - \gamma_m(N_1 - N_0)S_1, \tag{17}$$

$$C\frac{dV_2}{dt} = I_2 - f(V_2) + \kappa S_1(t) + \sigma\xi_V^2(t), \tag{18}$$

$$L\frac{dI_2}{dt} = V_0 - V_2 - RI_2, \tag{19}$$

$$\frac{dS_2}{dt} = \left(\gamma_m(N_2 - N_0) - \frac{1}{\tau_p}\right)S_2 + \gamma_m N_2 + \sqrt{\gamma_m N_2 S_2}\xi_S^2(t), \tag{20}$$

$$\frac{dN_2}{dt} = \frac{I_0 + I_2}{q_e} - (\gamma_m + \gamma_l + \gamma_{nr})N_2 - \gamma_m(N_2 - N_0)S_2. \tag{21}$$

Here, τ is the time associated to the delay line. Fig. 11 illustrates the cyclical propagation of pulses along the circuit. As explained in section 2.2, a two-node optoelectronic network with both RTDs biased close to the valley of the current-voltage characteristic and both LDs biased under the emission threshold responds to a single negative square voltage pulse with an optical pulse. This pulse is now sent back to the Tx RTD through the delay line and Tx PD, and arrives with a delay time of approximately τ. Assuming that τ is significantly longer than the refractory time of the excitable response, the delayed pulse will elicit a second excitable response from the Tx RTD, which will ultimately lead to a second optical pulse from the Rx. This new pulse is also sent to the Tx with a delay, which leads to the emission of a third pulse by the Rx. In other words, the injection of a single suprathreshold voltage pulse injected into the Rx RTD produces the emission of optical pulses periodically. This delayed synapsis has potential applications in memory storage, as discussed by Romeira et al [10]. Indeed, it is valid to say that the same bit of information is circulating periodically around the RTD-PD two-node network. On that regard, the propagation of such a bit is equivalent to that in an indefinitely long sequence of nodes.

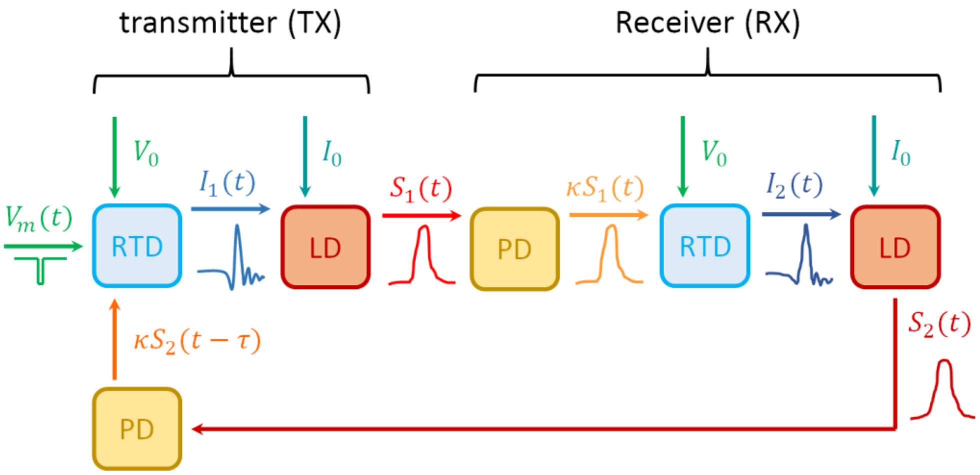


Fig. 11. Generation and transmission of optical spikes in the optoelectronic network with a delay line.

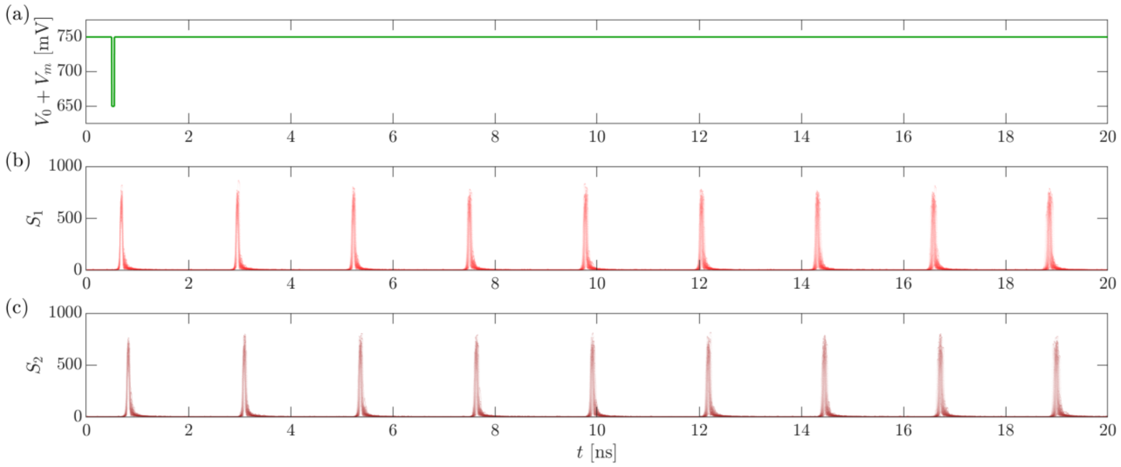


Fig. 12. 100 Simulations of Eqs. (14-21) with $I_0 = 214$ μA, $V_0 = 750$ mV, $\kappa = 0.05$ μA, $\sigma = 2.5 \times 10^{-12}$ [SI], $\tau = 2$ ns, sharp current-voltage characteristic and injected with a single 50 ps long, 100 mV deep negative voltage pulse. **(a)** transmitter RTD total input voltage. **(b)** Transmitter LD optical output. **(c)** Receiver LD optical output.

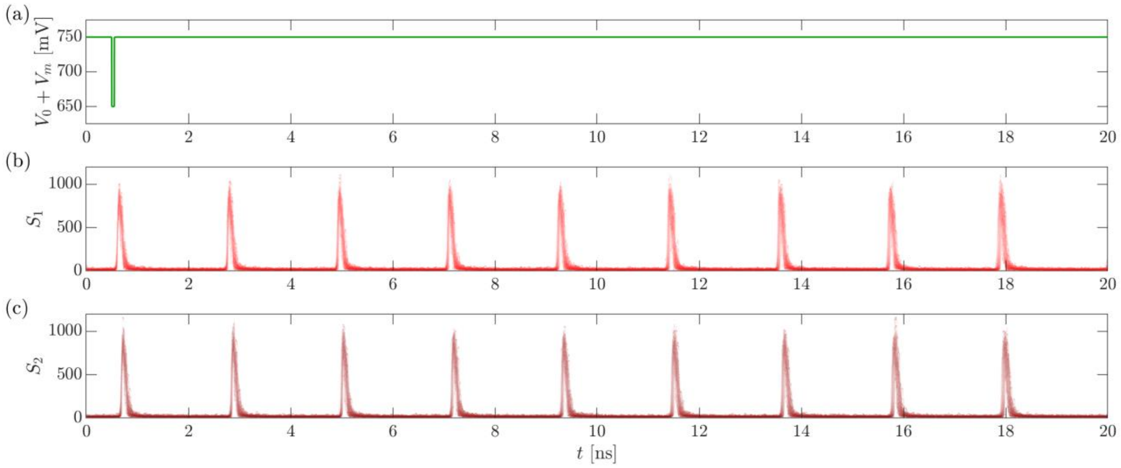


Fig. 13. 100 Simulations of Eqs. (14-21) with $I_0 = 250$ μA, $V_0 = 750$ mV, $\kappa = 0.05$ μA, $\sigma = 2.5 \times 10^{-12}$ [SI], $\tau = 2$ ns, sharp current-voltage characteristic and injected with a single 50 ps long, 100 mV deep negative voltage pulse. **(a)** transmitter RTD total input voltage. **(b)** Transmitter LD optical output. **(c)** Receiver LD optical output.

This is the principle behind another application of delayed synapsis in spiking neural networks, where a single node in the neuromorphic processor serves as multiple *virtual nodes* in a network algorithm [22]. Furthermore, the same bit propagation can also be emulated with a single optoelectronic node where the LD is connected to the PD and then the RTD with a delayed line (i.e., the node is connected to itself in *autapsis*). Such a system is modelled with four equations instead of eight, with the subsequent decrease of time and energy consumption in numerical simulations, in comparison with the two-node network discussed here.

The pulse propagation process is however, not flawless, as illustrated in Figs. 12 and 13, where short and long pulses are generated respectively. In both cases, the delay time is set at τ = 2 ns. Nonetheless, pulses are emitted with a longer separation period (around 2.2 ns in the case of short pulses and 2.1 ns in the case of short pulses). This is attributed to the intrinsically delayed response of each node to supra-threshold perturbations, first discussed in subsection 2.1.3. Also, the pulse loses coherence as it is emitted again and again, in the sense that, when multiple simulations with same parameters and initial conditions are run, the time where the pulse is emitted is not exactly the same. This is notorious when all the responses are plotted together. The loss of coherence is attributed to the intrinsic noise in both the RTD and LD components.

3. Numerical simulation methods

In this section, the numerical methods used in the apps are detailed. These methods have been coded on MATLAB™ and are based on semi-implicit integration. A uniform time partition $t_n = n \times \Delta t, n \in \mathbb{N}_0$, is considered. A reliable level of accuracy is ensured by choosing a time step dt much shorter than the inverse of the RTD tank frequency ($\omega_{tk} = 1/\sqrt{LC}$, an estimation of the order of the self-oscillation frequency) and the LD photon lifetime,

$$\Delta t \ll \sqrt{LC}, \quad \Delta t \ll \tau_p. \tag{22}$$

For the rest of this report, the time step will be set at $\Delta t = 0.01 \times \min\left(\sqrt{LC}, \tau_p\right)$. However, variables are stored only every 0.5 ps in order to avoid excessive memory consumption.

3.1 Integration method in the neuromorphic transmitter node

3.1.1 Electrical part

In order to recursively find the RTD output variables (V, I) at times t_n, Eqs. (1,2) are integrated over the time interval $[t_n, t_{n+1}]$,

$$C(V_{n+1} - V_n) = \int_{t_n}^{t_{n+1}} \big(I - f(V)\big)\, dt + \sigma \Delta W_V^n, \tag{23}$$

$$L(I_{n+1} - I_n) = \int_{t_n}^{t_{n+1}} (V_0 + V_m(t) - V - RI)\, dt, \tag{24}$$

where $W(t)$ is a normalized Wiener process (i.e., $\Delta W \sim \mathrm{Norm}(0, \Delta t)$). Integration over I in Eq. 23 is approximated by a trapezoidal estimation,

$$\int_{t_n}^{t_{n+1}} I\, dt \approx \tfrac{1}{2}(I_n + I_{n+1})\Delta t. \tag{25}$$

To approximate the integration over $f(V)$ as a linear expression with respect to V_{n+1}, V is approximated as the secant line connecting V_n and V_{n+1}. Then, $f(V)$ is expanded to the first order in t,

$$\begin{aligned}\int_{t_n}^{t_{n+1}} f(V)\, dt &\approx \int_{t_n}^{t_{n+1}} f(V_n + \tfrac{t}{\Delta t}(V_{n+1} - V_n))\, dt,\\ &\approx \int_{t_n}^{t_{n+1}} \Big(f(V_n) + f'(V_n)\tfrac{t}{\Delta t}(V_{n+1} - V_n)\Big)\, dt,\\ &\approx \Big(f(V_n) + \tfrac{1}{2} f'(V_n)(V_{n+1} - V_n)\Big)\Delta t.\end{aligned} \tag{26}$$

After trapezoidal approximation of the integral in Eq. 24, Eqs. (23,24) now read,

$$\begin{aligned}C(V_{n+1} - V_n) &= \Big(\tfrac{1}{2}(I_n + I_{n+1}) - \Big(f(V_n) + \tfrac{1}{2} f'(V_n)(V_{n+1} - V_n)\Big)\Big)\Delta t + \sigma \Delta W_V^n,\\ L(I_{n+1} - I_n) &= \Big(V_0 + \tfrac{1}{2}\big((V_m^n + V_m^{n+1}) - (V_n + V_{n+1}) - R(I_n + I_{n+1})\big)\Big)\Delta t.\end{aligned} \tag{27}$$

Eqs. 27 constitute a 2×2 system of linear equations from which V_{n+1} and I_{n+1} are found,

$$V_{n+1} = \frac{d_n \varepsilon_n - b_n \varphi_n}{a_n c_n - b_n d_n}, \qquad I_{n+1} = \frac{-c_n \varepsilon_n + a_n \varphi_n}{a_n c_n - b_n d_n}. \tag{28}$$

Where $a_n, b_n, c_n, d_n, \varepsilon_n, \varphi_n$ are given by,

$$\begin{gathered} a_n = -C - \tfrac{1}{2} f'(V_n)\Delta t, \qquad b_n = \tfrac{1}{2}\Delta t, \qquad c_n = -\tfrac{1}{2}\Delta t, \qquad d_n = -\tfrac{1}{2} R \Delta t, \\ \varepsilon_n = -C V_n + \left(f(V_n) - \tfrac{1}{2}(f'(V_n)V_n + I_n) \right) \Delta t - \sigma \Delta W_V^n, \\ \varphi_n = -L I_n + \left(\tfrac{1}{2}(V_n + R I_n - (V_m^n + V_m^{n+1})) - V_0 \right) \Delta t. \end{gathered} \tag{29}$$

The initial conditions (n = 0) in the app integration algorithm are set at the equilibrium fixed point, which is found by substitution of all time-derivatives, noise and modulation in Eqs. (1,2) with zero,

$$\begin{gathered} I - f(V) = 0, \\ V_0 - V - RI = 0. \end{gathered} \tag{30}$$

Eqs. 30 correspond to the intersection of the nullclines (see Figs. 3 and 4) and ultimately lead to a single equation, $V_0 - V - Rf(V) = 0$, which can be solved numerically. The stability of the fixed point (or points) is determined by the associated eigenvalues, also calculated numerically. The stable fixed point with smallest V will be chosen as initial condition. If there is no stable fixed point, the unstable fixed point with smallest V will be chosen.

3.1.2 Optical part

The dynamics of the optical part in the optoelectronic node is accounted for simulations of the following equations,

$$\frac{dE}{dt} = \frac{1-i}{2}\left(\gamma_m (N - N_0) - \frac{1}{\tau_p} \right) E + \sqrt{\tfrac{1}{2}\gamma_m N}\left(\xi_x(t) + i \xi_y(t) \right), \tag{31}$$

$$\frac{dN}{dt} = \frac{I_0 + I}{q_e} - (\gamma_m + \gamma_l + \gamma_{nr}) N - \gamma_m (N - N_0) |E(t)|^2. \tag{32}$$

Here, $\xi_x(t), \xi_y(t)$ are white noise functions uncorrelated from each other and over time. $E(t)$ is a complex field that defines the photon number as $S(t) = |E(t)|^2$. With this in mind, Eq. 3 corresponds to the square norm of Eq. 31, and Eq. 4 is identical to Eq. 32. Integration over of Eqs. (31,32) over the time interval $[t_n, t_{n+1}]$ leads to,

$$\begin{aligned} E_{n+1} - E_n = {} & \frac{1-i}{2} \int_{t_n}^{t_{n+1}} \left(\gamma_m (N_n - N_0) - \frac{1}{\tau_p} \right) E \, dt \\ & + \int_{t_n}^{t_{n+1}} \sqrt{\tfrac{1}{2}\gamma_m N} \left(dW_x + i dW_y \right), \end{aligned} \tag{33}$$

$$\begin{aligned} N_{n+1} - N_n = {} & \int_{t_n}^{t_{n+1}} \Bigg(\frac{I_0 + I}{q_e} - (\gamma_m + \gamma_l + \gamma_{nr}) N \\ & - \gamma_m (N - N_0) S \Bigg) dt. \end{aligned} \tag{34}$$

where $W_x(t), W_y(t)$ are uncorrelated Wiener processes. In order to approximate the integrals in Eq. 33, $N(t)$ is approximated as N_n and $E(t)$ is approximated as the secant line connecting E_n and E_{n+1},

$$E_{n+1} - E_n \approx \frac{1-i}{2}\left(\gamma_m(N_n - N_0) - \frac{1}{\tau_p}\right)\frac{E_n + E_{n+1}}{2}\Delta t + \sqrt{\tfrac{1}{2}\gamma_m N_n}\left(\Delta W_x^n + i\Delta W_y^n\right). \quad (35)$$

where $\Delta W_x^n, \Delta W_y^n \sim \mathrm{Norm}(0, \Delta t)$. This allows for an approachable, recursive derivation of E_{n+1},

$$E_{n+1} = \frac{1}{1 - M_n}\left((1 + M_n)E_n + \sqrt{\tfrac{1}{2}\gamma_m N_n}\left(\Delta W_x^n + i\Delta W_y^n\right)\right), \quad (36)$$

where $M_n = \dfrac{1-i}{4}\left(\gamma_m(N_n - N_0) - \dfrac{1}{\tau_p}\right)\Delta t$. Finally, $S_n = |E_n|^2$ for all n.

In order to calculate the integral in Eq. 34, $I(t)$ and $S(t)$ are approximated as I_{n+1} (already calculated in subsection 3.1.1) and S_{n+1} (already calculated above), while $N(t)$ is approximated as the secant line connecting N_n and N_{n+1},

$$N_{n+1} - N_n = \left(\frac{I_0 + I_{n+1}}{q_e} + (\gamma_m + \gamma_l + \gamma_{nr})\tfrac{1}{2}(N_n + N_{n+1}) - \gamma_m\left(\tfrac{1}{2}(N_n + N_{n+1}) - N_0\right)S_{n+1}\right)\Delta t.$$

This allows for an approachable, recursive derivation of N_{n+1},

$$N_{n+1} = \frac{1}{1 + \widetilde{M}_n}\left((1 - \widetilde{M}_n)N_n + \left(\frac{I_0 + I_{n+1}}{q_e} + \gamma_m N_0 S_{n+1}\right)\Delta t\right), \quad (37)$$

where $\widetilde{M}_n = \frac{1}{2}\left(\gamma_m + \gamma_l + \gamma_{nr}(1 + S_{n+1})\right)\Delta t$. The initial conditions (n = 0) in the app integration algorithm are set at the equilibrium fixed point, which is found by substitution of all time-derivatives and noise in Eqs. (3,4) with zero, and substitution of I with the starting point chosen at subsection 3.1.1,

$$\begin{aligned} &\left(\gamma_m(N - N_0) - \frac{1}{\tau_p}\right)S + \gamma_m N = 0, \\ &\frac{I_0 + I}{q_e} - (\gamma_m + \gamma_l + \gamma_{nr})N - \gamma_m(N - N_0)S = 0. \end{aligned} \quad (38)$$

For parameters $N_0, \tau_p, \gamma_m, \gamma_l, \gamma_{nr} > 0$, the nonlinear system of Eqs. 38 has always two solutions accounting for one stable and one unstable fixed point. The stable fixed point is chosen as the starting point for the numerical integration algorithm.

3.1.3 Integrated optoelectronic neuron (algorithm outline)

The integrator algorithm in the Neuromorphic Transmitter Node App can run simulations on Eqs. (1-4) concurrently (it should be repeated here that simulations with identical parameters and initial conditions will return non-identical outputs given the intrinsic noise in the system). The user can choose the specifications of the RTD and LD, as well as the voltage and current bias. The modulation voltage $V_m(t)$ is a train of square voltage pulses and the user can also choose its features (amplitude, period, number of cycles, phase and pulse length). Finally, the user can also choose the number of realizations to be simulated in parallel. The algorithm returns the output variables V, I, S, N as matrices, where the rows represent evolution over time and columns represent multiple realizations. The initial conditions are set at equilibrium, according to Eqs. (30,38). The time step Δt is set as $\Delta t = 0.01 \times \min\left(\sqrt{LC}, \tau_p\right)$.

The integrator algorithm consists of a loop where new rows $V_{n+1}, I_{n+1}, S_{n+1}, N_{n+1}$ are computed in terms of the previous rows V_n, I_n, S_n, N_n by applying Eqs. (28,29,36,37) in element-wise row operations. At each step of the loop, operations are carried out in the following order:

- Compute V_{n+1}, I_{n+1} in terms of V_n, I_n according to Eqs. (28,29).
- Compute S_{n+1} in terms of E_n, N_n according to Eq. 36.
- Compute N_{n+1} in terms of I_{n+1}, S_{n+1}, N_n according to Eq. 37.
- If the total time $(n+1)\Delta t$ is multiple of 0.5 ps (more specifically, if *n*+1 is multiple of round(0.5 ps/Δt)), add the rows $V_{n+1}, I_{n+1}, S_{n+1}, N_{n+1}$ at the bottom of the output matrices V, I, S, N.

3.2 Integration method in the neuromorphic Tx-Rx network

The two-node network is modelled by Eqs. (6-13). Eqs. (6-9) account for the Tx node (TX) and are identical to Eqs. (1-4) accounting for the single node. Consequently, the integration methods are identical as well. Eqs. (10-13) account for the Rx node and the only differences with Eqs. (1-4) are in that the modulation voltage term $V_m(t)$ is not present in the *I*-equation and that there is an additional $\kappa S_1(t)$ current term in the *V*-equation.

Just like the integration algorithm in the single node app (subsection 3.1.3), the integration algorithm here receives the user's specifications on the RTD and LD (voltage and current bias), and on the train of square voltage pulses $V_m(t)$ that drives the Tx as well as the number of realizations to be simulated in parallel. The algorithm returns matrices accounting for the variables $V_1, I_1, S_1, N_1, V_2, I_2, S_2, N_2$. The rows correspond to the evolution of the variables over time and the columns account for multiple realizations of the simulation. The initial conditions (*n* = 0) are set at equilibrium, given by Eqs. (30,38). The simulation time step Δt is set as $\Delta t = 0.01 \times \min\left(\sqrt{LC}, \tau_p\right)$.

The algorithm consists of a loop where a new row for the output matrices is computed in terms of the previous rows. These computations are carried out via element-wise row operations, like in the single node app (subsection 3.1.3). Each iteration of the loop includes the following operations:

- Compute V_1^{n+1}, I_1^{n+1} in terms of V_1^n, I_1^n. First, define,

$$a_1^n = -C - \tfrac{1}{2}f'(V_1^n)\Delta t, \qquad b_1^n = \tfrac{1}{2}\Delta t, \qquad c_1^n = -\tfrac{1}{2}\Delta t, \qquad d_1^n = -\tfrac{1}{2}R\Delta t,$$

$$\varepsilon_1^n = -CV_1^n + \left(f(V_n) - \tfrac{1}{2}(f'(V_1^n)V_1^n + I_1^n)\right)\Delta t - \sigma\Delta W_{V1}^n,$$

$$\varphi_2^n = -LI_1^n + \left(\tfrac{1}{2}\left(V_1^n + RI_1^n - (V_m^n + V_m^{n+1})\right) - V_0\right)\Delta t.$$

Then, compute,

$$V_1^{n+1} = \frac{d_1^n \varepsilon_1^n - b_1^n \varphi_1^n}{a_1^n b_1^n - c_1^n d_1^n}, \qquad I_1^{n+1} = \frac{-c_1^n \varepsilon_1^n + a_1^n \varphi_1^n}{a_1^n b_1^n - c_1^n d_1^n}.$$

- Compute S_1^{n+1} in terms of E_1^n, N_1^n.

First, define $M_1^n = \dfrac{1-i}{4}\left(\gamma_m(N_1^{n+1} - N_0) - \dfrac{1}{\tau_p}\right)\Delta t.$

Next, compute $E_1^{n+1} = \dfrac{1}{1-M_1^n}\left((1+M_1^n)E_1^n + \sqrt{\tfrac{1}{2}\gamma_m N_1^n}\left(\Delta W_{x1}^n + i\Delta W_{y1}^n\right)\right).$

Finally, compute $S_1^{n+1} = \left|E_1^{n+1}\right|^2$.

- Compute N_1^{n+1} in terms of $I_1^{n+1}, S_1^{n+1}, N_1^n$.

First define $\widetilde{M}_1^n = \frac{1}{2}\left(\gamma_m + \gamma_l + \gamma_{nr}(1 + S_1^{n+1})\right)\Delta t$.

Next, compute $N_1^{n+1} = \dfrac{1}{1 + \widetilde{M}_1^n}\left((1 - \widetilde{M}_1^n)N_1^n + \left(\dfrac{I_0 + I_1^{n+1}}{q_e} + \gamma_m N_0 S_1^{n+1}\right)\Delta t\right)$.

- Compute V_2^{n+1}, I_2^{n+1} in terms of S_1^{n+1}, V_2^n, I_2^n. First, define,

$$a_2^n = -C - \tfrac{1}{2}f'(V_2^n)\Delta t, \qquad b_2^n = \tfrac{1}{2}\Delta t, \qquad c_2^n = -\tfrac{1}{2}\Delta t, \qquad d_2^n = -\tfrac{1}{2}R\Delta t,$$

$$\varepsilon_2^n = -CV_2^n + \left(f(V_2) - \tfrac{1}{2}(f'(V_2^n)V_2^n + I_2^n) + \kappa S_1^{n+1}\right)\Delta t - \sigma \Delta W_{V2}^n,$$

$$\varphi_2^n = -LI_2^n + \left(\tfrac{1}{2}(V_2^n + RI_2^n) - V_0\right)\Delta t.$$

Then, compute,

$$V_2^{n+1} = \frac{d_2^n \varepsilon_2^n - b_2^n \varphi_2^n}{a_2^n b_2^n - c_2^n d_2^n}, \qquad I_2^{n+1} = \frac{-c_2^n \varepsilon_2^n + a_2^n \varphi_2^n}{a_2^n b_2^n - c_2^n d_2^n}.$$

- Compute S_2^{n+1} in terms of E_2^n, N_2^n.

First, define $M_2^n = \dfrac{1-i}{4}\left(\gamma_m(N_2^{n+1} - N_0) - \dfrac{1}{\tau_p}\right)\Delta t$.

Next, compute $E_2^{n+1} = \dfrac{1}{1 - M_2^n}\left((1 + M_2^n)E_2^n + \sqrt{\tfrac{1}{2}\gamma_m N_2^n}\left(\Delta W_{x2}^n + i\Delta W_{y2}^n\right)\right)$.

Finally, compute $S_2^{n+1} = \left|E_2^{n+1}\right|^2$.

- Compute N_2^{n+1} in terms of $I_2^{n+1}, S_2^{n+1}, N_2^n$.

First define $\widetilde{M}_2^n = \frac{1}{2}\left(\gamma_m + \gamma_l + \gamma_{nr}(1 + S_2^{n+1})\right)\Delta t$.

Next, compute $N_2^{n+1} = \dfrac{1}{1 + \widetilde{M}_2^n}\left((1 - \widetilde{M}_2^n)N_2^n + \left(\dfrac{I_0 + I_2^{n+1}}{q_e} + \gamma_m N_0 S_2^{n+1}\right)\Delta t\right)$.

- If the total time $(n+1)\Delta t$ is multiple of 0.5 ps (more specifically, if n+1 is multiple of round(0.5 ps/Δt)), add the rows $V_1^{n+1}, I_1^{n+1}, S_1^{n+1}, N_1^{n+1}, V_2^{n+1}, I_2^{n+1}, S_2^{n+1}, N_2^{n+1}$ at the bottom of the output matrices.

It is worth emphasizing that the Wiener processes are represented by uncorrelated random variables with normal distribution of mean value zero and variance Δt, i.e., $\Delta W_k^n \sim \mathrm{Norm}(0, \Delta t)$.

3.3 Integration in the Tx-Rx network with bidirectional communication

The two-node network with a delay line is covered in section 2.3 and its dynamics are described by Eqs. (14-21). These are almost identical to Eqs. (6-13) except for the delay term $\kappa S_2(t-\tau)$ in Eq. 14. In the integration algorithm for this app, the external modulation voltage term $V_m(t)$ is a single square pulse instead of a train of pulses, given that a single pulse together with the delay line ensure periodical emission of pulses (spikes) in the system. There are no other differences with the integration algorithm covered in section 3.2. Multiple realizations of the simulation can be run simultaneously as specified by the user. The algorithm returns matrices accounting for the variables $V_1, I_1, S_1, N_1, V_2, I_2, S_2, N_2$. The rows correspond to the evolution of the variables over time and the columns account for multiple realizations of the simulation. The initial conditions (n = 0) are set at equilibrium, and the time step is set as $\Delta t = 0.01 \times \min\left(\sqrt{LC}, \tau_p\right)$.

The delay term is only present in the computation of V_1^{n+1}, I_1^{n+1}. At every step of the algorithm loop, a delay coefficient is defined as $n_{del} = \max(0, n - \mathrm{round}(\tau/\Delta t))$. Then, the delayed optical output from the receiver LD is computed as $\kappa S_2^{n_{del}}$ and added into the computation of ε_1^n,

$$a_1^n = -C - \tfrac{1}{2} f'(V_1^n)\Delta t, \qquad b_1^n = \tfrac{1}{2}\Delta t, \qquad c_1^n = -\tfrac{1}{2}\Delta t, \qquad d_1^n = -\tfrac{1}{2} R\Delta t,$$

$$\varepsilon_1^n = -CV_1^n + \left(f(V_n) - \tfrac{1}{2}(f'(V_1^n)V_1^n + I_1^n) + \kappa S_2^{n_{del}}\right)\Delta t - \sigma \Delta W_{V1}^n,$$

$$\varphi_2^n = -LI_1^n + \left(\tfrac{1}{2}\left(V_1^n + RI_1^n - (V_m^n + V_m^{n+1})\right) - V_0\right)\Delta t.$$

Next, V_1^{n+1}, I_1^{n+1} are computed like in section 3.2,

$$V_1^{n+1} = \frac{d_1^n \varepsilon_1^n - b_1^n \varphi_1^n}{a_1^n b_1^n - c_1^n d_1^n}, \qquad I_1^{n+1} = \frac{-c_1^n \varepsilon_1^n + a_1^n \varphi_1^n}{a_1^n b_1^n - c_1^n d_1^n}.$$

The rest of the computations are exactly like in section 3.2. If the total time $(n+1)\Delta t$ is multiple of 0.5 ps (more specifically, if n+1 is multiple of round(0.5 ps/Δt)), the rows $V_1^{n+1}, I_1^{n+1}, S_1^{n+1}, N_1^{n+1}, V_2^{n+1}, I_2^{n+1}, S_2^{n+1}, N_2^{n+1}$ are added at the bottom of the output matrices.

4. Neuromorphic system simulator apps

Three apps with graphical user interfaces (GUI) have been programmed intending to reproduce the dynamics of a single optoelectronic node, a two-node integrated network and a two-node integrated network with bidirectional communication, respectively. The three aps have been coded on MATLAB™ and their GUIs have been designed on MATLAB GUIDE™. The standalone versions of the apps have been compiled with MATLAB Compiler™. The standalone apps require MATLAB™ and MATLAB Runtime™ installed on a computer to run (although MATLAB™ is not required to be running).

In 2016, MathWorks® released MATLAB AppDesigner™, a more modern platform to code and design GUIs. Apps coded on AppDesigner™ can be compiled as web apps with MATLAB Web App compiler™. Web apps are run directly on a browser and can be included on a website as long as the server has MATLAB Web App Server™ installed. MathWorks® is intending to discontinue MATLAB GUIDE™ in favour of MATLAB App Designer™. To this day, however, apps coded on MATLAB AppDesigner™ have a tendency to glitch when executed on some versions of Linux™. Migration strategies have already been developed in order to continue the work on the neuromorphic simulator apps with MATLAB AppDesigner™ and compile standalone and web apps.

4.1 Neuromorphic transmitter node simulator app

Fig. 14 shows a screenshot of the single RTD-LD node simulator app. The GUI dimensions are 1250 × 818 pixels. The user can tune the specifications for the simulations by filling the edit fields and choosing options from the pop-up menus at the left side of the GUI.

The "I-V characteristic" pop-up menu (Fig. 14a) allows to choose the RTD current-voltage characteristic, $f(V)$. At the moment, there are 3 characteristics available: "Smooth curve" to choose the smooth curve defined in subsection 2.1.1 (see Fig. 1a and Table 1a), "Sharp curve" to choose the sharp curve defined in subsection 2.1.1 (see Fig. 1b and Table 1b), and "Sharp curve (R)" to choose a rescaled sharp curve where the distance between the peak and valley points (i.e, $V_v - V_p$ and $I_p - I_v$) is the same as in the smooth curve.

Under the "I-V characteristic" pop-up menu there are four edit fields to choose the parameters in the electrical part of the system (Fig. 14b); the circuit's intrinsic resistance (R), capacitance (C) and inductance (L) as well as the noise intensity in voltage units, which is related to the actual noise intensity as $\sigma = C^{3/4}L^{-1/4}V_{\text{noise}}$.

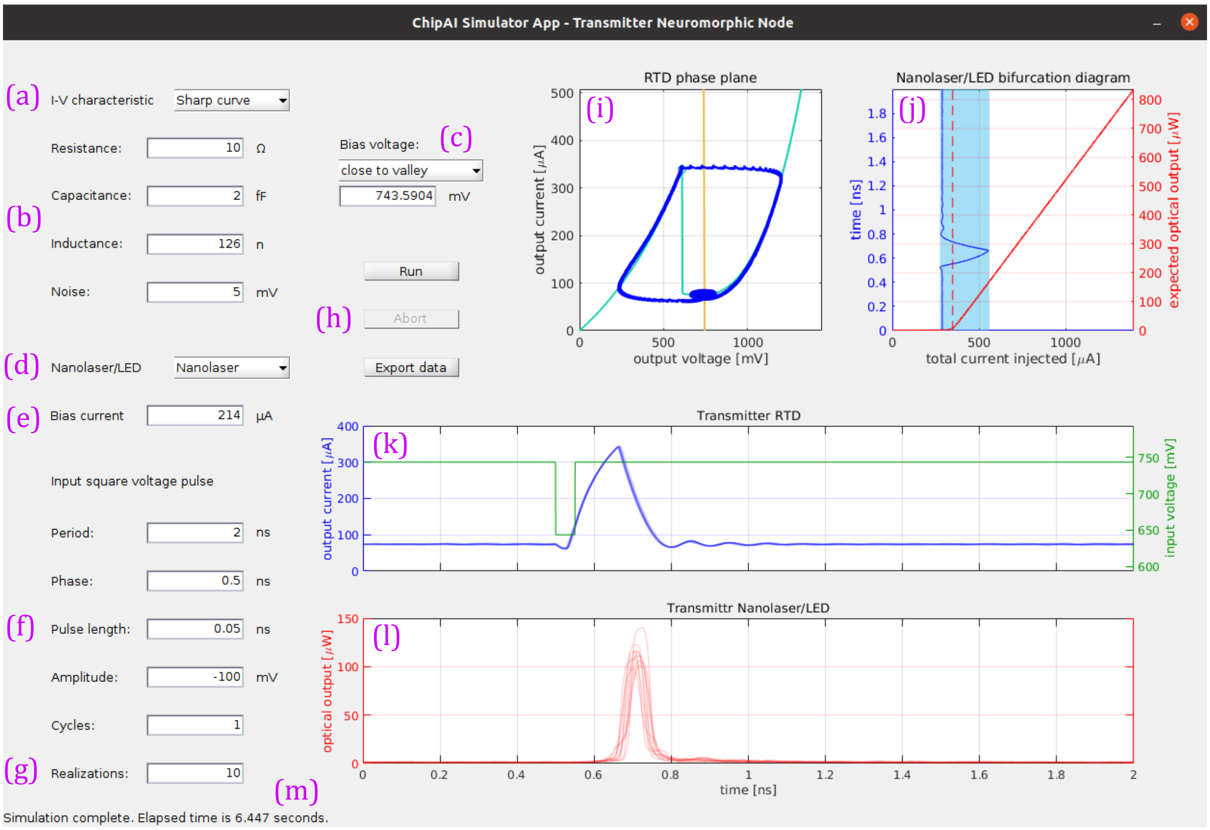
At the right side of these fields, there are the options to choose the input bias voltage, V_0 (Fig. 14c). A pop-up menu allows to set the bias either manually or outside the NDC region, close to the peak or close to the valley of the I-V characteristic. If the "set by user" choice is selected, the user can set V_0 manually in the edit field under the pop-up menu. If either the "close to peak" or "close to valley" choice is selected, V_0 is computed in order to have the intersection between the nullclines outside the NDC region, at a distance $0.2(V_v - V_p)$ from it, and the edit field is automatically filled.

Below the RTD parameters edit fields, a pop-menu (Fig. 14d) allows to set the LD specifications, i.e., the parameters $N_0, \tau_p, \gamma_l, \gamma_m, \gamma_{nr}$. Right now there is only one option available, "Nanolaser", which assigns the values detailed in subsection 2.1.2. We plan to add more choices in the future. Under the pop-up menu, there is an edit field to choose the LD bias current, I_0 (Fig. 14e).

In the app simulation, the input modulation voltage $V_m(t)$ is a train of voltage pulses. Under the bias current edit field, there are more edit fields (Fig. 14f) to customize this train in terms of amplitude (which can be positive or negative, accounting for positive or negative pulses), period, phase, pulse length and number of cycles (and hence the total time of simulation). The "Realizations" edit field (Fig. 14g) permits to choose the number of simulations to be run in parallel (it should be noted that due to noise in the system, simulations with identical parameters and initial conditions will return non-identical outputs). To avoid RAM memory saturation and/or OS crashing, the number of cycles and repetitions are limited to 50 and 500, respectively.

The GUI includes three action buttons (Fig. 14h); "Run", "Abort" and "Export data". The "Run" button runs a simulation of Eqs. (1-4) with the specifications given by the user (see section 3.1 for a detailed explanation on the numerical calculations). Once the simulation is finished, the figures in the GUI (Fig. 14i-l) are painted and the static text at the lower left corner (Fig. 14m) reports the elapsed time. A simulation may take between a few seconds up to several minutes, depending on the parameters, total time, number of repetitions and the machine the app is being run on. It is recommended to run some 1-realization simulations before running multiple simulation realizations to have an understanding on the system dynamics and how it is affected by the parameters.

The "Abort" button is enabled to push only when a simulation is in progress, in order to interrupt it. After pushing the button, the static text at the lower left corner (Fig. 14m) reports that the simulation has been aborted.



ChipAI Simulator App - Transmitter Neuromorphic Node

I-V characteristic: Sharp curve

Resistance: 10 Ω

Capacitance: 2 fF

Inductance: 126 n

Noise: 5 mV

Nanolaser/LED: Nanolaser

Bias current: 214 µA

Input square voltage pulse

Period: 2 ns

Phase: 0.5 ns

Pulse length: 0.05 ns

Amplitude: -100 mV

Cycles: 1

Realizations: 10

Bias voltage: close to valley, 743.5904 mV

Simulation complete. Elapsed time is 6.447 seconds.

Fig. 14. Screenshot of the ChipAI Tx Neuromorphic Node simulator app. **(a)** I-V characteristic pop-up menu. **(b)** RTD parameters edit fields. **(c)** Input bias voltage pop-up menu and edit field. **(d)** LD specifications pop-up menu. **(e)** LD Input bias current edit field. **(f)** Input square voltage pulse edit fields. **(g)** Number of realizations edit field. **(h)** Action buttons. **(i)** RTD phase plane and nullclines. **(j)** LD bifurcation diagram, current threshold and total injected current. **(k)** Tx RTD output. **(l)** Tx LD output. **(m)** Informative static text.

The "Export data" button opens a dialog box to save the output data; specifically, the RTD output current, $I(t)$, and the LD optical output power, $P_0 S(t)$, where $P_0 = hc/\tau_p \lambda_0$. Here, h is the Planck constant, c is the speed of light τ_p is the photon lifetime and $\lambda_0 = 1550$ nm is the second window of opportunity for single-mode transmission. The data is saved as an XLS or XLSX file (at the user's discretion) with three sheets accounting for time and the two output variables. The time sheet has a single column. Columns in the other two sheets account for different realizations of the simulations while rows account for the evolution of the variables over time. The data export process may take between a few seconds up to several minutes depending on the data size. The data export process cannot be aborted. Also, it is important to keep in mind that most OSs do not allow data to be saved if the number of rows or columns exceeds a certain limit.

The figure at the top and centre of the GUI (Fig. 14i) illustrates the RTD phase plane. It is entitled "RTD phase plane". This figure shows the RTD I-V characteristic (cyan line), the circuit's load line (yellow line) and the variables $\big(V(t), I(t)\big)$ returned from the simulation (blue dots).

The figure at the upper right corner of the GUI (Fig. 14j) is entitled "Nanolaser/LED bifurcation diagram". It includes a bifurcation diagram of the LD dynamics; specifically, the mean value of the optical output power as a function of the total current injected into the LD, $I + I_0$ (solid red line). The current emission threshold is also included (dashed red line). On the same figure, the total current injected is plotted over time (solid blue line). If multiple realizations are simulated, the ensemble average is plotted. The idea is to give the user an intuition of where the rest value of the total current injected is relative to the current emission threshold and at what moments the threshold is crossed.

The two figures at the lower half of the GUI (Fig. 14k,l) are entitled "Transmitter RTD" and "Transmitter Nanolaser/LED". They show the RTD output current, $I(t)$, and the LD output power, $P_0S(t)$ over time as blue and red curves, respectively. If multiple realizations are simulated, the curves will have a degree of transparency (alpha channel). The "Transmitter RTD" figure also includes the input voltage, $V_0 + V_m(t)$, as a green line, with its axis drawn at the right side of the figure. This allows the user to have a notion on how long the RTD and the LD take to respond to the external stimulus.

Finally, the static text at the lower left corner of the GUI (Fig. 14m) informs the user when a simulation is running, data is being exported and the outcome and elapsed time of both actions once they finish. This text also informs when the user aborts a simulation, the user mistakenly inputs prohibited parameters (e.g., negative resistance, non-natural number of repetitions, infinity, NaN, etc.) or when an error occurs. In the latter cases, the static text is coloured in red.

Examples of simulations performed by the app are shown in Figs. 15 and 16. This time, the smooth curve is chosen. Fig. 15 shows the Tx response with the RTD biased close to the valley of the I-V characteristic, the LD biased below its threshold current, and the RTD perturbed with a train of suprathreshold negative voltage pulses. The pulses elicit excitable current spikes from the RTD. The rest value of the RTD output current is slightly under the LD emission threshold ($I_{th} = 337.54$ μA). When a spike is injected into the LD, the current increases up to 700 μA and the LD emits a light pulse. Fig. 16 shows the transmitter's response with the RTD biased close to the peak of the I-V characteristic, the LD biased over the current emission threshold, and the RTD perturbed with a train of suprathreshold positive voltage pulses. The rest value of the total current injected into the LD ($I - I_0$) is about 500 μA. Since this current is above the emission threshold, the LD is permanently emitting light. The voltage pulses elicit excitable current negative spikes from the RTD. When these spikes are injected into the LD, $I - I_0$ descends from its rest value down to 150 μA, thus crossing the threshold and momentarily stopping the LD emission. This is a valid way to transmit bits of information in the form of negative optical pulses, albeit at a high energy cost. In both configurations, the period of the voltage pulse train is set at 0.5 ns, significantly longer than the optical pulse FWHM as well as the excitable response refractory time (about 0.5 ns), in order to ensure that the system's response to each perturbation will not be affected by prior ones.

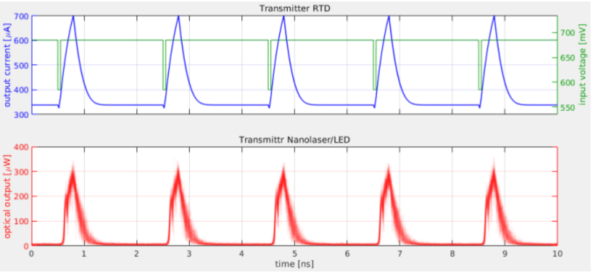


Fig. 15. 100 Simulations of Eqs. (1-4) run by the single node app, with $I_0 = 0$ μA, $V_0 = 685$ mV, $V_{\text{noise}} = 5$ mV, smooth current-voltage characteristic and modulated with a train of 50 ps long, 100 mV deep and 2 ns period negative voltage pulses.

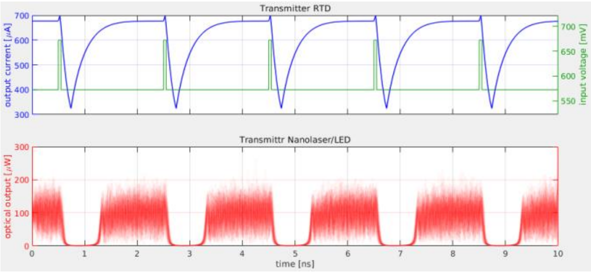


Fig. 16. 100 Simulations of Eqs. (1-4) run by the single node app, with $I_0 = -200$ μA, $V_0 = 572$ mV, $V_{\text{noise}} = 5$ mV, smooth current-voltage characteristic and modulated with a train of 50 ps long, 100 mV high and 2 ns period positive voltage pulses.

4.2 Neuromorphic transmitter-receiver network simulator app

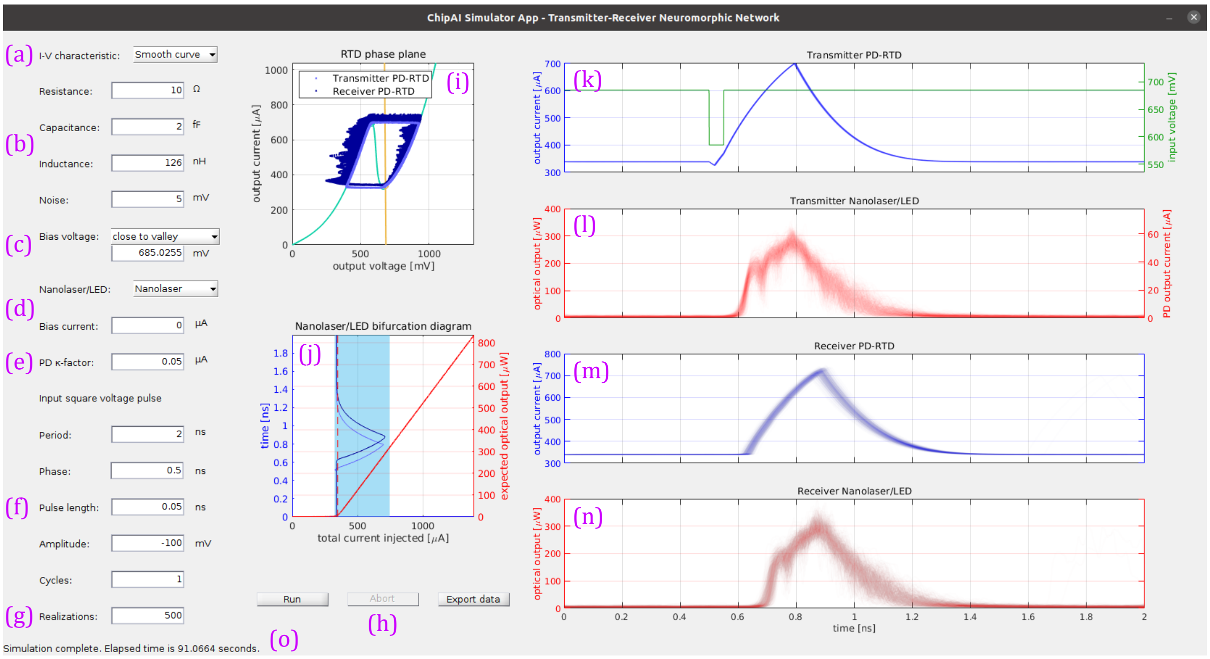
Fig. 17 shows a screenshot of the two-node network app. It is quite similar to the single node app detailed in section 4.1, but it has additional figures accounting for the Rx output variables. The GUI dimensions are 1650 × 868 pixels.

The left side of the GUI has some edit fields and pop-up menus for the user to set the specifications and parameters for simulation. The pop-up menu at the upper left corner of the GUI allows to choose the RTD I-V characteristic (Fig. 17a). There are currently three options available: smooth curve, sharp curve and rescaled sharp curve just like with the single node app. Under this menu there are four edit fields to enter the values of the RTD circuit parameters, both at the Tx and Rx: resistance, inductance, capacitance and noise intensity in voltage units (Fig. 17b). Under these fields, there is a pop-up menu to choose the input bias voltage either close to the peak, close to the valley or specified manually in the edit field right below (Fig. 17c). Next, there is a pop-up menu to choose the specifications for the Tx LD and Rx LD, (Fig. 17d). There is a single option available at the moment: "Nanolaser", which assigns values to the LD parameters as given in subsection 2.1.2. Under this menu, there is an edit field to enter the input bias current injected into both LDs, I_0 (Fig. 17d), and another edit field to enter the Tx PD conversion factor, κ (Fig. 17e). At the lower left part of the GUI, there are the edit fields to enter the specifications for the input train of square voltage pulses, $V_m(t)$, including amplitude (which can be positive or negative), period, phase, pulse length and number of cycles (Fig. 17f), as well as an additional edit field to enter the number of realizations (i.e., simulations to be run in parallel, Fig. 17g). To avoid RAM memory saturation and/or OS crashing, the number of cycles and number of repetitions are limited to 50 and 500, respectively.

At the right side of the "Realizations" edit field, there are three action buttons to run a simulation, abort a simulation in progress or export data from an already complete simulation (Fig. 17h). The time a simulation might take will depend on the specifications given by the user as well as the machine the app is running on. In general, it will take twice as long a simulation in the single node app under similar specifications, since the app detailed here accounts for two nodes in sequence, with twice the number of equations. The same applies for the amount of time exporting data may take, which may ascend up to several minutes. Data exportation in progress cannot be aborted.

Above the three action buttons, there are two figures providing information about the RTDs and LDs, respectively. The upper figure (Fig. 17i) is entitled "RTD phase plane" and shows the RTD I-V characteristic, the load line and the electrical output from the Tx RTD, $(V_1(t), I_1(t))$, as well as the Rx RTD ($V_2(t), I_2(t)$), shown as light and dark blue dots, respectively. The lower figure (Fig. 17j) is entitled “Nanolaser/LED bifurcation diagram” and includes a plot of the mean optical power as a function of the total current injected into the Tx LD or Rx LD ($I_1 + I_0$ and $I_2 + I_0$, respectively), together with the current emission threshold. The figure also includes a plot of the aforementioned total currents over time (light and dark blue lines accounting for the Rx LD and Tx LD), in order to provide some intuition on where the rest values of the total currents are relative to the lasing threshold, and when the threshold is crossed to trigger or supress light emission.

At the right side of the GUI there are four figures summarizing the evolution of the output variables over time. The top figure (Fig. 17k) shows the Tx RTD output current ($I_1(t)$) together with its total input voltage ($V_0 + V_m(t)$), with the corresponding vertical axes traced at the left and right side of the figure, respectively. In the second figure, the Tx LD output power ($P_0 S_1(t)$) and the Rx PD input current ($\kappa S_1(t)$) are plotted together, with the corresponding axes shown at the left and right side of the figure. Note that these two variables are actually the same, measured in power and current units. The third figure (Fig. 17m) shows the Tx RTD output current ($I_2(t)$). The bottom figure (Fig. 17n) shows the Rx LD optical output in power units ($P_0 S_2(t)$). The power coefficient is given by $P_0 = hc/\tau_p \lambda_0$. Here, h is the Planck constant, c is the speed of light τ_p is the photon lifetime and $\lambda_0 = 1550$ nm is the second window of opportunity for single-mode transmission.



I-V characteristic: Smooth curve

Resistance: 10 Ω

Capacitance: 2 fF

Inductance: 126 nH

Noise: 5 mV

Bias voltage: close to valley
685.0255 mV

Nanolaser/LED: Nanolaser

Bias current: 0 μA

PD κ-factor: 0.05 μA

Input square voltage pulse

Period: 2 ns

Phase: 0.5 ns

Pulse length: 0.05 ns

Amplitude: -100 mV

Cycles: 1

Realizations: 500

Simulation complete. Elapsed time is 91.0664 seconds.

Fig. 17. Screenshot of the ChipAI Tx-Rx Neuromorphic Network simulator app. **(a)** I-V characteristic pop-up menu. **(b)** RTD parameters edit fields. **(c)** Input bias voltage pop-up menu and edit field. **(d)** LD specifications pop-up menu and LD Input bias current edit field. **(e)** Tx PD conversion factor edit field. **(f)** Input square voltage pulse edit fields. **(g)** Number of realizations edit field. **(h)** Action buttons. **(i)** RTD phase plane and nullclines. **(j)** LD bifurcation diagram, threshold and total injected currents. **(k)** Tx RTD output. **(l)** Tx LD output. **(m)** Rx RTD and **(n)** LD outputs. **(o)** Informative static text.

Finally, the static text at the lower left corner of the GUI (Fig. 17o) informs the user when a simulation is running, data is being exported and the outcome and elapsed time of both actions once they finish. This text also informs when the user aborts a simulation, the user mistakenly inputs prohibited parameters (e.g., negative resistance, non-natural number of repetitions, infinity, NaN, etc.) or when an error occurs. In the latter cases, the static text is coloured in red.

An example of the performance of the two-node app is visualized in Fig. 18, where a simulation is carried out with the same specifications as in Fig. 15. This shows that the optical pulses generated by the Tx node are successfully propagated by the Rx node when the smooth RTD I-V characteristic is selected.

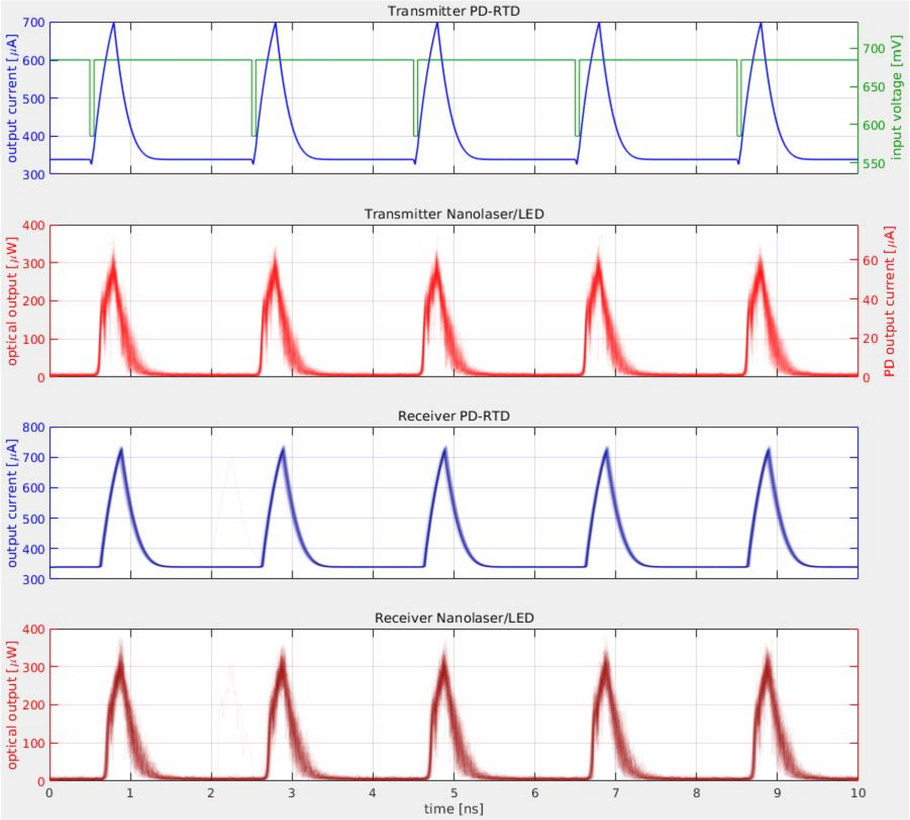
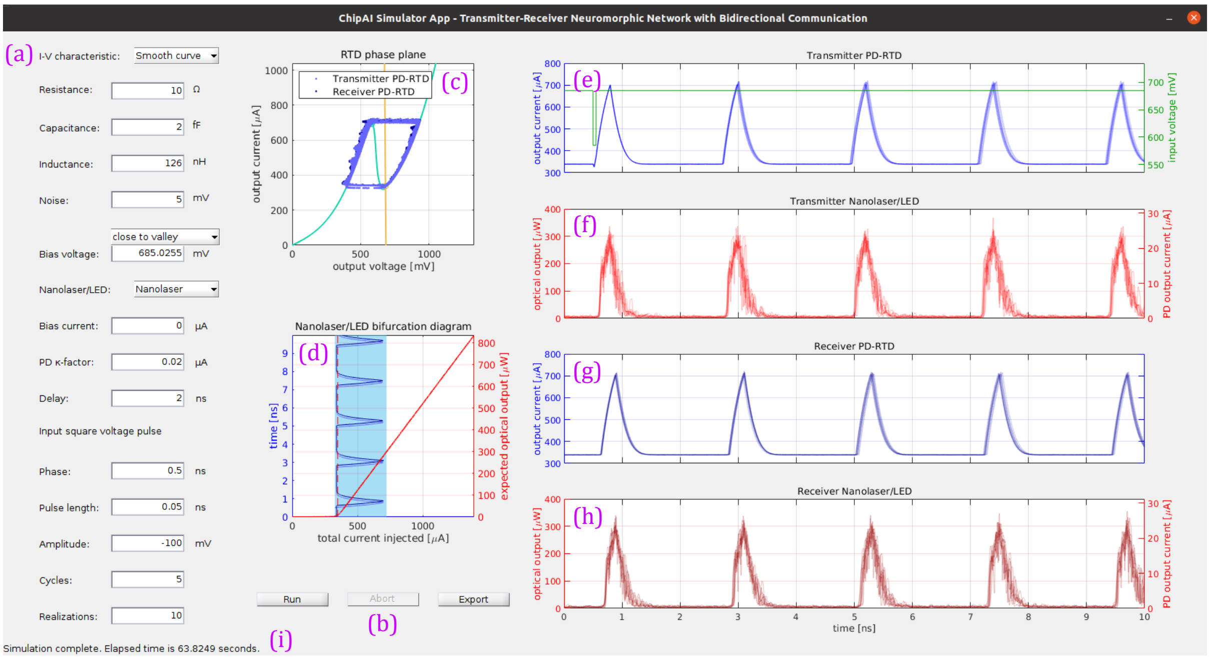


Fig. 18. 100 Simulations of Eqs. (6-13) run by the two-node app, with $I_0 = 0$ μA, $V_0 = 685$ mV, $V_{\text{noise}} = 5$ mV, $\kappa = 0.05$ μA, smooth current-voltage characteristic and modulated with a train of 50 ps long, 100 mV deep and 2 ns period negative voltage pulses

4.3 Transmitter-receiver with bidirectional communication simulator app

Since the equations that model the two-node network with and without delay line are almost identical, so are the corresponding simulator apps. The GUI for the app with delay line is shown in Fig. 19. It has the same size (1605 × 868 pixels), figures and action buttons as the app without delay line. At the left side of the GUI, there are the pop-up menus to enter the simulation parameters (Fig. 19a). In the app with delay line, the input voltage $V_m(t)$ is a single square pulse instead of a train of pulses, since the delay line ensures that optical pulses will be generated periodically. Therefore, the “Period” edit field is not present anymore. Instead, the app has a “Delay” edit field to enter the delay time, τ. The “Cycles” edit field accounts for the number of times the delay time is repeated (i.e., the simulation total time will be $N_{\text{cycles}}\tau$). Like in the previous apps, multiple simulations can be run in parallel, specified by the “Realizations” edit field.



ChipAI Simulator App - Transmitter-Receiver Neuromorphic Network with Bidirectional Communication

I-V characteristic: Smooth curve

Resistance: 10 Ω

Capacitance: 2 fF

Inductance: 126 nH

Noise: 5 mV

close to valley

Bias voltage: 685.0255 mV

Nanolaser/LED: Nanolaser

Bias current: 0 μA

PD κ-factor: 0.02 μA

Delay: 2 ns

Input square voltage pulse

Phase: 0.5 ns

Pulse length: 0.05 ns

Amplitude: -100 mV

Cycles: 5

Realizations: 10

Simulation complete. Elapsed time is 63.8249 seconds.

Fig. 19. Screenshot of the ChipAI simulator app accounting for the Tx-Rx Neuromorphic Network with bidirectional communication. **(a)** Pop-up menus and edit fields to specify the simulation parameters. **(b)** Action buttons. **(c)** RTD phase plane and nullclines. **(d)** LD bifurcation diagram, threshold and total injected currents. **(e)** Tx RTD and **(f)** LD outputs. **(g)** Rx RTD and **(h)** LD outputs. **(i)** Informative static text.

As in the previous apps covered in sections 4.1 and 4.2, the two-node app with delay line has three action buttons to run a simulation under the specifications provided by the user, abort a simulation in progress and export data from the last simulation (Fig. 19b). The time a simulation might take will depend on the simulation parameters, but in general it takes longer than a simulation run by the two-node app without delay line with similar specifications (about twice longer), since the delay term $\kappa S_1(t-\tau)$ requires saving S_1^n at every step of the simulation and not only every 0.5 ps like the other variables and like in the other apps (see chapter 3).

The two-node app includes the same figures as its counterpart without delay line, illustrating the RTD phase plane (Fig. 19c), LD bifurcation diagram (Fig. 19d) and electrical and optical, Tx and Rx output variables over time (Fig. 19e-h). The optical output variables are plotted in both power and current units (the latter referring to the Tx and Rx output PD currents).

Finally, the static text at the lower left corner of the GUI (Fig. 17i) informs the user when a simulation is running, data is being exported and the outcome and elapsed time of both actions once they finish. This text also informs when the user aborts a simulation, the user mistakenly inputs prohibited parameters (e.g., negative resistance, non-natural number of repetitions, infinity, NaN, etc.) or when an error occurs. In the latter cases, the static text is coloured in red.

An example of the performance of the app is visualized in Fig. 20, where a simulation is carried out with the same specifications as in Fig. 18 and a delay time τ = 2 ns. A single negative voltage pulse successfully elicits periodic emission of optical pulses, as already explained in section 2.3 (see Figs. 12 and 13), although the smooth curve is used here instead of the sharp curve. Note that the pulse emission period is slightly longer than 2 ns, given the intrinsic delays in the responses of every component of the circuit. Also, as new pulses are generated, the response loses coherence, in the sense that, as multiple simulations are run under the same parameters, the exact times when pulses are emitted are not identical, as a consequence of the system's intrinsic noise.

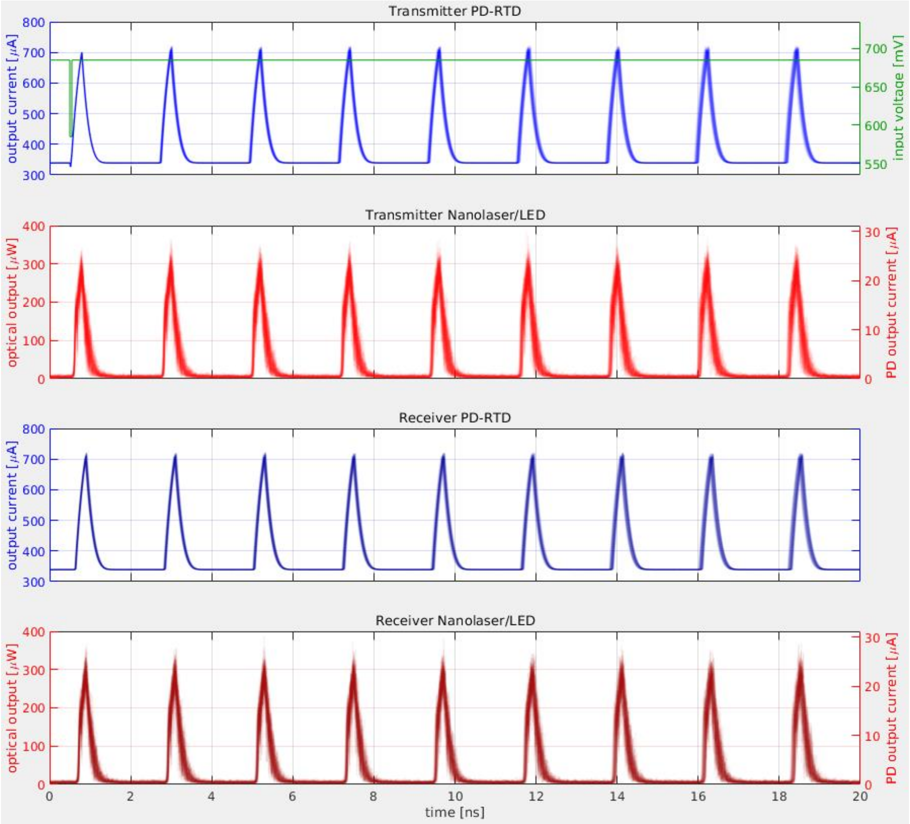


Fig. 20. 100 Simulations of Eqs. (14-21) run by the two-node app, with $I_0 = 0$ μA, $V_0 = 685$ mV, $V_{\text{noise}} = 5$ mV, $\kappa = 0.05$ μA, $\tau = 2$ ns, smooth current-voltage characteristic and injected with a single 50 ps long, 100 mV deep negative voltage pulse.

5. Summary and conclusions

In this document, three graphical applications under development as part of the ChipAI Project have been presented. These apps simulate excitable spike generation and propagation as well as other responses in optoelectronic, neuromorphic, nanoscale devices based on resonant tunnelling diodes (RTD) and laser diodes (LD). The three apps account for a single neuromorphic node consisting of an RTD and an LD, a network made of two neuromorphic nodes connected via a photodetector (PD) and the same two-node network additionally connected with a delay line in the reverse direction. The apps are programmed in MATLAB GUIDE™.

The theoretical models of the simulations are based on are discussed in detail, including the parameters involved. It is explained that a single node behaves as an excitable optical pulse generator when the RTD is biased in the proximity to the valley of its current-voltage (I-V) characteristic, the LD is biased under its threshold current and a negative square voltage pulse is injected. Likewise, the two-node network successfully propagates the aforementioned pulses when both nodes follow the same configuration. If a delay line is connected in the reverse direction, a single voltage pulse triggers optical pulses periodically, with a period slightly longer than the delay time, since every component in the device has its own intrinsic response delay. The numerical methods internally implemented in the apps are based on semi-implicit integration, with the possibility of multiple simulations being run in parallel via element-wise row operations. This is relevant because simulations under identical parameters and initial conditions will return non-identical outputs, as a consequence of the intrinsic noise in the models.

The apps' graphical interfaces have a number of pop-up menus and edit fields where the user can enter the parameters and specifications for a simulation (e.g., RTD current-voltage characteristic, LD parameters, resistance, inductance, capacitance, delay time, specifications for the external modulation/perturbation, etc.), three action buttons to run a simulation, abort a simulation or export data as an XLS or XLSX spread sheet, and figures to summarize the simulation outcome, including evolution of the output variables over time, RTD phase plan and LD bifurcation diagram. Examples of generation and propagation of positive and negative optical spikes simulated by the apps are included as well.

The apps intend to serve as a didactic tool for casual or expert users visiting the ChipAI Project website (https://www.chipai.eu/neuromorphic-simulator) who might be interested in getting to know more in depth about the dynamics in the optoelectronic neuromorphic devices under design and fabrication in the project. However, it may also be a valuable contribution for the project theoretical and experimental team members in their research. The loss of coherence (i.e., the capacity to respond to the same input square pulse with identical spikes triggered at identical times) in the excitable pulses generated in the two-node network with delay line is of significant interest. The loss of coherence is also a consequence of the intrinsic noise. This phenomenon can be studied with an equivalent model accounting for diffusion of spatiotemporal localized structures in a one-dimensional periodic layer [10]. In that regard, the third app can provide valuable feedback for the theoretical modelling.

This work is still in progress and there are some courses of action which could be included in the apps to be followed in the nearby future. These include additional RTD I-V characteristics and LD specifications are to be included in the apps. There is currently only one LD model and only two RTD characteristics; a smooth curve and a sharp curve, following Schulman's formula [12]. More choices to include may be the third-degree polynomial used in the FitzHugh-Nagumo model [7,23] and the Van Der Pol Oscillator [24], a simplified curve made of a linear function minus a sigmoid (which has already been studied by the project team, as demonstrated by one paper published in the Chaos Journal [25]) and a piecewise linear function. Studies with PWL functions relevant to RTD dynamics have been carried out in the past [26].

Bibliography

[1] John Von Neumann, "First Draft of a Report on the EDVAC," Philadelphia, 1945.

[2] Sorin Voinigescu, *High-frequency integrated circuits*. New York: Cambridge University Press, 2013.

[3] Paul Merolla et al, "A million spiking-neuron integrated circuit with a scalable communication network and interface," *Science*, vol. 345, pp. 68-673, 2014.

[4] (2021, September) Intel Advances Neuromorphic with Loihi 2, New Lava Software Framework and New Partners. [Online]. A million spiking-neuron integrated circuit with a scalable communication network and interface

[5] National Institute of Health. The BRAIN Initiative. [Online]. https://braininitiative.nih.gov/

[6] Sally Ward-Foxton. (2021, July) Innatera Unveils Neuromorphic AI Chip to Accelerate Spiking Networks. [Online]. https://www.eetimes.com/innatera-unveils-neuromorphic-ai-chip-to-accelerate-spiking-networks/#

[7] Eugene Izhikevich, *Dynamical Systems in Neuroscience: The Geometry of Excitability and Bursting*. Cambridge, Masaschusetts, London, England: The MIT Press, 2007.

[8] Ignacio Ortega-Piwonka, Oreste Piro, Bruno Romeira, and Julien Javaloyes, "Bursting and excitability in neuromorphic resonant tunneling diodes," *Physical Review Applied*, vol. 15, no. 034037, 2021.

[9] Bruno Romeira et al, "Excitability and optical pulse generation in semiconductor lasers driven by resonant tunneling diode photo-detectors," *Optics Express*, vol. 12, pp. 20931-20940, 2013.

[10] Bruno Romeira, Jose Figueiredo, and Julien Javaloyes, "Delay dynamics of neuromorphic optoelectronic nanoscale resonators: Perspectives and applications," *Chaos: An Interdisciplinary Journal of Nonlinear Sciecne*, vol. 27, no. 114323, 2017.

[11] Bruno Romeira, Jose Figueiredo, and Julien Javaloyes, "NanoLEDs for energy-efficient and gigahertz-speed spike-based sub-λ neuromorphic nanophotonic computing," *Nanophotonics*, vol. 9, no. 3, 2020.

[12] Joel N. Schulman, "Physics-based RTD current-voltage equation," *IEEE Electron Device Letters*, vol. 17, pp. 220-222, 1996.

[13] Sebastian Diebold et al, "High-speed error-free wireless data transmission using a terahertz resonant tunnelling diode transmitter and receiver," *Electronic Letters*, vol. 52, no. 24, 2016.

[14] Rinnosuke Izumi, Safumi Suzuki, and Masahiro Asada, "1.98 THz Resonant-Tunneling-Diode Oscillator with Reduced Conduction Loss by Thick Antenna Electrode," in *2017 42nd International Conference on Infrared, Millimeter, and Terahertz Waves (IRMMW-THz)*, 2017, pp. 1-2.

[15] Jue Wang et al, "High Performance Resonant Tunneling Diode Oscillators for THz Applications," in *2015 IEEE Compound Semiconductor Integrated Circuit Symposium (CSICS)*, 2015, pp. 1-4.

[16] Jue Wang et al, "15 Gbps 50 cm Wireless Link Using a High Power Compact III-V 84 GHz Transmitter," *IEEE Transactions on Microwave Theory and Techniques*, vol. 66, no. 11, pp. 4698-4705, November 2018.

[17] Bruno Romeira and Andrea Fiore, "Purcell effect in the stimulated and spontaneous emission rates of nanoscale semiconductor lasers.," *IEEE Journal of Quantum Electronics*, vol. 54, no. 2, pp. 1-12, 2018.

[18] Steven H. Strogatz, *Nonlinear Dynamics and Chaos: With Applications to Physics, Biology, Chemistry, and Engineering*. Boca Raton, London, New York: CRC Press, 2015.

[19] Paul Mandel, "Laser Lorenz Equations with a Time-Dependent Parameter," *Physical Review Letters*, vol. 53, no. 1818, 1984.

[20] Mathias Marconi et al, "Testing Critical Slowing Down as a Bifurcation Indicator in a Low-dissipation Dynamical System," *Physical Review Letters*, vol. 125, no. 134102, 2020.

[21] Jorge R. Tredicce et al, "Critical slowing down at a bifurcation," *American Journal of Physics*, vol. 72, no. 799, 2004.

[22] Michael Nielsen. (2019, December) Neural Networks and Deep Learning. [Online]. http://neuralnetworksanddeeplearning.com/

[23] Ricard FitzHugh, "Impulses and Physiological States in Theoretical Models of Nerve Membrane," *Biophysical Journal*, vol. 1, no. 6, pp. 445-466, 1961.

[24] Balthazar Van Der Pol, "A theory of the amplitude of free and forced triode vibrations," *Radio Review (later Wireless World)*, vol. 1, pp. 701-710, 1920.

[25] Ignacio Ortega-Piwonka et al, "Simplified description of dynamics in neuromorphic resonant tunneling diodes," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 31, no. 113128, 2021.

[26] Victoriano Carmona, Soledad Fernández-García, and Antonio Teruel, "Saddle-node of limit cycles in planar piecewise linear systems and applications," *Discrete & Continuous Dynamical Systems*, vol. 39, no. 9, 2019.