Design of Spiking Rate Coded Logic Gates for C. elegans Inspired Contour Tracking

Shashwat Shukla(✉), Sangya Dutta, and Udayan Ganguly

Indian Institute of Technology Bombay, Mumbai, India
shashwat.shukla@iitb.ac.in

Abstract. Bio-inspired energy efficient control is a frontier for autonomous navigation and robotics. *Binary* input-output neuronal logic gates are demonstrated in literature – while *analog* input-output logic gates are needed for continuous analog real-world control. In this paper, we design logic gates such as AND, OR and XOR using networks of Leaky Integrate-and-Fire neurons with *analog* rate (frequency) coded inputs and output, where refractory period is shown to be a critical knob for neuronal design. To demonstrate our design method, we present contour tracking inspired by the chemotaxis network of the worm *C. elegans* and demonstrate for the first time an end-to-end Spiking Neural Network (SNN) solution. First, we demonstrate contour tracking with an average deviation equal to literature with non-neuronal logic gates. Second, 2x improvement in tracking accuracy is enabled by implementing latency reduction leading to state of the art performance with an average deviation of 0.55% from the set-point. Third, a new feature of local extrema escape is demonstrated with an *analog* XOR gate, which uses only 5 neurons – better than *binary* logic neuronal circuits. The XOR gate demonstrates the universality of our logic scheme. Finally, we demonstrate the hardware feasibility of our network based on experimental results on 32 nm Silicon-on-Insulator (SOI) based artificial neurons with tunable refractory periods. Thus, we present a general framework of analog neuronal control logic along with the feasibility of their implementation in mature SOI technology platform for autonomous SNN navigation controller hardware.

Keywords: Spiking Neural Network · Motor control
Neuromorphic computing

1 Introduction

Spiking Neural Networks (**SNNs**) are third generation Artificial Neural Networks that attempt to model neurons as computing units with underlying temporal dynamics that resembles the spiking nature of biological neurons. While SNNs have been used to solve a variety of problems in classification and regression, an equally intriguing aspect is the implementation of control in a natural setting that could serve the dual purpose of (i) demystifying complex biological behavior and (ii) inspiring efficient robotics applications. Chemotaxis in *Caenorhabditis elegans* (C. *elegans*) is an example of such a biological behaviour which requires control. C. *elegans* is a free living nematode,

V. Kůrková et al. (Eds.): ICANN 2018, LNCS 11139, pp. 273–283, 2018.
https://doi.org/10.1007/978-3-030-01418-6_27

which can sense a large number of chemicals including NaCl. This ability allows these worms to find and subsequently move along a set point in the chemical concentration space so as to locate food sources. Typically, the sensory neurons ASEL and ASER provide chemical gradient information [4], and this information is used by interneurons in the worm to decide the direction to subsequently move along to reach the chemical set-point. The output of this computation is fed to motor neurons, which actuate their motion. Santurkar et al. [3] proposed a SNN model for chemical contour tracking inspired by C. *elegans*. They demonstrate the superiority of spiking architectures over non-spiking models and their tolerance to noise using the biologically realistic model of sensory neurons proposed in [4]. However, the inter-neuronal operations required to drive motor neurons were computed without using neural circuits. Instead, an artificial mathematical computation was used. Hence the SNN is not performing integrated, end-to-end control of all three stages of computation i.e. (i) sensory neuron (ii) interneuron (iii) motor neuron levels. Such external control is neither biologically realistic nor energy-area efficient. Further, more sophisticated/realistic behaviour, such as escaping a local extrema, without which the worm fails to reach the desired concentration over arbitrary concentration landscapes, has not been demonstrated in neuronal circuits.

Existing SNN based logic gates [11–14] encode binary logic values using fixed spiking frequencies (low/high). But, the output spiking frequency of the gate should vary proportionately to one or more input spiking frequencies so that the worm turns in proportion to the urgency of sensory signals. This motivates the design of *analog* rate coded logic gates.

In this paper, first, we implement *analog* rate-coded logic *gates* (AND, OR, XOR) by designing neuronal responses using *refractory periods*. Second, we integrate AND and OR with the sensory and motor neurons to demonstrate end-to-end control in the chemotaxis network. Third, we incorporate an additional sub-network using the XOR gate to escape a local extrema. The XOR, being a universal gate, also enables random logic circuit implementation. Our design enables a reduced number of neurons for logic gates which leads to lower response latency (measured between sensory input and motor neuron output), critical for many control applications. Fourth, we modify the response of sensory neurons proposed in [3] to reduce response latency to enable significantly improved tracking compared to state-of-the-art. Finally, a hardware neuron with configurable refractory period is demonstrated on a highly matured 32 nm silicon-on-insulator CMOS technology.

2 Network Architecture

Figure 1 shows the proposed SNN architecture for chemotaxis in C. *elegans*. All the neurons in our network are Leaky Integrate and Fire neurons. The following sections will discuss the functional role of all the neurons used in this network.

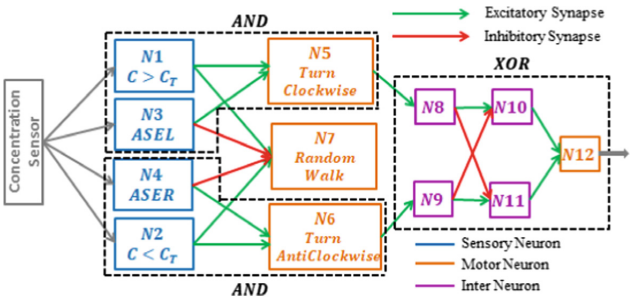

Fig. 1. Block diagram of the proposed SNN for contour tracking. N_1, N_2, N_3 and N_4 are sensory neurons, receiving input from the concentration sensor. N_5, N_6, N_7, N_{12} are motor neurons. N_8, N_9, N_{10}, N_{11} are the interneurons used to implement the XOR sub-network. The spiking frequency of N_{12} is the XOR of the spiking frequencies of N_5 and N_6.

2.1 Turning Left and Right: The AND Sub-network

2.1.1 Sensory Neurons

As shown in Fig. 1, the neurons N_1 and N_2 are threshold detectors. N_1 fires when the current concentration (C) is greater than the set-point C_T i.e. $C > C_T$, while N_2 fires when $C < C_T$. A hard-threshold (ideally a step-function) is compared to a soft-threshold based N_1 in Fig. 2. N_3 and N_4 are gradient (dC/dt) detectors, firing respectively for positive ($dC/dt > 0$) and negative ($dC/dt < 0$) changes in concentration. The input to all four sensory neurons, at each time step (t), is the concentration at the current location of the worm. The equations for the ionic currents that implement the required responses have been delegated to the appendix.

2.1.2 Motor Neurons

The target for the worm is to reach the desired concentration set-point C_T. If at a particular time instant, the worm detects $dC/dt > 0$ (i.e. N_3 spikes) and $C > C_T$ (i.e. N_1 spikes), the worm infers that it is moving away from C_T and hence tries to turn around. In this case the worm turns right by 3° and moves forward at a velocity of 0.01 mm/s with the rate of turning being proportional to dC/dt. The motor neuron N_5 encodes this command. Hence, the spiking frequency of N_5 has to be the output of an AND operation over the spiking frequencies of N_1 and N_3. i.e. $\boldsymbol{N_5 = \text{AND}(N_1, N_3)}$. The bold face is used to denote spiking frequency of the corresponding neuron. Similarly, the motor neuron N_6 spikes if $dC/dt < 0$ (i.e. N_4 spikes) and $C < C_T$ (i.e. N_2 spikes). In this case, the worm turns left by 3° at a velocity of 0.01 mm/s and the motor neuron N_6 encodes this command i.e. $\boldsymbol{N_6 = \text{AND}(N_2, N_4)}$. When $dC/dt > 0$ and $C < C_T$ or $dC/dt < 0$ and $C > C_T$, the worm infers that it is moving towards C_T and hence keeps moving forward at a constant velocity without turning.

2.1.3 Design Principles for the AND Sub-network

Under the rate-coded approximation (which implies that injected current is assumed to be proportional to the spiking frequency), a neuron fires if the sum of the input spiking

frequencies (f_i) multiplied by the corresponding synaptic weights (w_i) is greater than f_{th}. Hence the general firing condition for a neuron with k pre-synaptic neurons is:

$$\sum_{i=1}^{k} w_i f_i > f_{th} \tag{1}$$

Without a refractory period, the spiking frequency of N_3 varies linearly with the observed gradient. This gradient can be very large in some parts of the environment, leading to very high spiking frequency, which in turn can make N_5 spike by itself even if N_1 is not spiking. The saturation in the responses of N_1 and N_3 ensures that N_5 only fires when *both* N_1 and N_3 fire and hence acts as an AND gate. We choose w_1 and w_3 such that: $w_1 f_{1,\max} = w_3 f_{3,\max} = f_{th}^{-}$, where $f_{1,\max}, f_{3,\max}$ are respectively the maximum spiking frequencies of N_1 and N_3, and f_{th}^{-} is some value close to, but *smaller than* f_{th}. We choose f_{th}^{-} to be close to f_{th} so that even a small value of f_3 will lead to N_5 spiking, hence ensuring the control circuit's sensitivity to very small gradients as well.

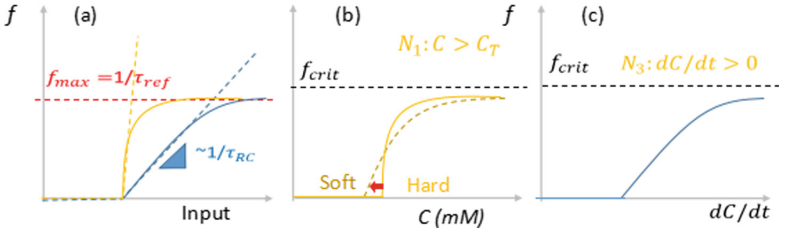

Fig. 2. (a) A typical LIF neuron (blue dashed line) has almost a rectified linear unit (ReLU) behaviour where the slope decreases with the membrane time constant τ_{RC} of the LIF neuron. Adding a refractory period (τ_{ref}) limits maximum frequency (f_{max}) to $1/\tau_{ref}$. (b) N_1 neuron $f(C)$ behaviour where it fires when C exceeds C_T. Softer threshold initiates spiking before the hard-threshold. (c) N_3 neuron has $f(dC/dt)$ behaviour which has a spike frequency proportional to the $dC/dt > 0$; Both N_1 and N_3 have an $f_{max} < f_{crit}$ such that neither can individually cause N_5 to spike but they need to fire together to cause N_5 to fire to enable the analog AND operation where N_5 fires proportionally to N_3 only if N_1 also fires. Otherwise, N_5 does not fire. (Color figure online)

The responses of the threshold detectors (N_1 and N_2) were taken as step functions in [3] with the transition at the desired set-point. This discontinuous response is softened to a sigmoid (as shown in Fig. 2(b)) by introducing a refractory period (chosen using the same logic as for N_3). The onset of the sigmoidal response is chosen to be *before* the set-point, allowing the worm to turn a little before it has reached the set-point. This enables latency reduction and closer tracking of the set-point. Identical reasoning holds for the N_2, N_4 and N_6 sub-network.

2.2 Random Walk

When the worm is on flat terrain and hence no gradient is detected ($^{dC}/_{dt} \sim 0$), the worm explores its surroundings randomly. This random search is initiated by motor neuron N_7. The spiking of N_7 causes the worm to move at an increased velocity of 0.3 mm/s and to randomly turn by an angle uniformly distributed in [−22.5°, 22.5°]. This strategy allows for rapid exploration of a local space, with N_7 continuing to fire until a gradient is detected.

2.3 Escaping Local Extrema: The XOR Sub-network

When the worm has found, and is tracking the set-point, N_5 and N_6 fire alternately as the worm keeps swerving left and right. However if only N_5 or only N_6 fires *exclusively,* then the worm is only turning left or right and hence going around in circles. Such a scenario is described in Fig. 3. If the worm starts anywhere in the valley, it will not be able to get out, as every time it moves up towards the rim of the valley (i.e. $^{dC}/_{dt} > 0$), N_3 fires with continuous firing of N_1 as $C > C_T$ at every point. As a consequence, N_5 will fire, making the worm turn back towards the basin. The worm then moves straight and now climbs up the other side and this process repeats. A second case where the worm would again be stuck is the scenario obtained when Fig. 3 is inverted on its head i.e. a small peak surrounded by a valley.

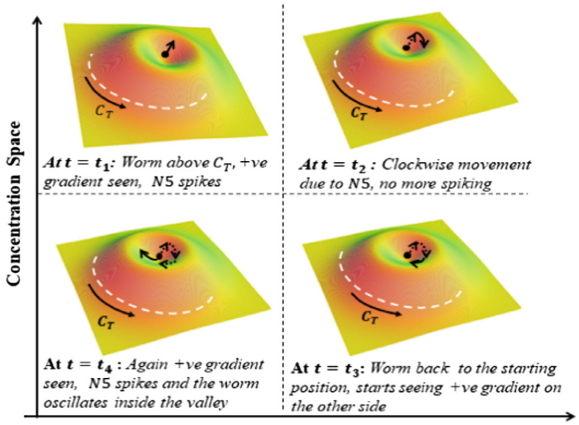

Fig. 3. Panels depicting the worm stuck in a valley at four consecutive time steps, in the absence of the XOR subnetwork.

Hence to solve this problem of getting stuck close to a local extrema, the XOR sub network is developed whose output is N_{12}. N_{12} is supposed to fire, if only N_5 or only N_6 is found to spike over some time period, i.e. $\boldsymbol{N_{12} = \mathbf{XOR}(N_5, N_6)}$. When N_{12} fires, the worm moves straight for 10 s, without turning, at a velocity of 0.5 mm/s and then resumes normal operation, having escaped the area where it was stuck. Such behavior has been observed in biology as well [5].

2.3.1 Design Principles for the XOR Sub-network

To enable XOR function, only the spiking events at N_5 or N_6 need to be detected. For such operation, we introduce a large refractory period of 10 s in the interneurons N_8 and N_9 and also set a very low voltage threshold for both these neurons, such that a single spike from N_5 is enough to make N_8 fire. Once N_8 fires, it will remain unresponsive for 10 s due to the refractory period. N_8 hence acts as a timed event detector. The same description holds for N_6 and N_9.

If we consider a long enough time period ($\sim$10 s) for our control problem, and N_8 fires once before and once after this period, without any spike from N_9, we infer that only N_5 has been firing for a significant amount of time. Hence, the worm needs to escape from this region. Similarly, if N_9 fired once before and once after a period of 10 s with N_8 not firing in between, the worm must escape this area.

We design N_{10} such that it fires once for every *two times* that N_8 fires. Note that if N_9 fires intermittently in the refractory period of N_8, then N_{10} will not fire due to the inhibitory connection linking N_9 to N_{10}. Interchanging the roles of N_8 and N_9 yields the behavior of N_{11}.

It is important to note that the current injected into N_{10} and N_{11} by N_8 and N_9 decay at a time scale much faster than the refractory period. Thus we chose very small values for the membrane conductance of N_8 and N_9 i.e. these two neurons are not very leaky and effectively function as integrators over this time-scale. Finally, N_{12} has a low spiking threshold, and functions as an OR gate. It fires when either N_{10} or N_{11} fires, i.e. $\boldsymbol{N_{12} = \mathbf{OR}(N_{10}, N_{11})}$. The firing of N_{12} causes the worm to move straight for 10 s, without turning. N_5, N_6 and N_7 are inhibited from firing during this 10 s period by injecting them with a large inhibitory EPSP current with timescale of the order of 10 s.

3 Results: Worm Dynamics

Our simulated worm is placed in a chemotaxis assay of dimensions 10 cm $\times$ 10 cm, with some arbitrary concentration distribution of the chemical NaCl. Figure 4 demonstrates the AND operation with the concentration seen by the worm and corresponding spiking patterns for N_1, N_3 and N_5. In Fig. 4, N_1 uses a **hard threshold** to fire for $C > C_T$ (Fig. 4 (b)) and N_3 fires for $C > C_T$ (Fig. 4(c)) which produces an AND behaviour at N_5 ***with significant latency*** (Fig. 4(d)). Figure 5 shows the behavior of our simulated worm for $C_T = 54$ mM. The worm moves about randomly at first, and then follows a gradient until it reaches the set-point and then continues to closely track the set-point, C_T. We observe that the worm swerves left and right, as it is slightly overshoots the tracking concentration, corrects it course and this process repeats. The corresponding concentration seen by the worm, shown in Fig. 6 shows an average 0.82% (absolute) deviation from set-point (as a fraction of the range of concentration in this space).

In Fig. 7, N_1 uses a pre-emptive soft threshold to fire earlier for $C > C_T$ (Fig. 7(b)) and N_3 fires for $C > C_T$ (Fig. 7c)). This produces an AND behaviour at N_5 with ***reduced latency*** (Fig. 7(d)). Figure 9 shows the concentration seen by the worm as it traced the trajectory in Fig. 8 to show 0.55% tracking accuracy, which is a 1.5$\times$ improvement over that in Fig. 6 due to the pre-emptive soft threshold. Figure 10 shows a simulated scenario where the worm gets stuck in a local minimum and is unable to escape. With the XOR sub-network added to our SNN, it can be seen that the worm can successfully come out of the concentration valley and starts tracking the set point as shown in Fig. 11.

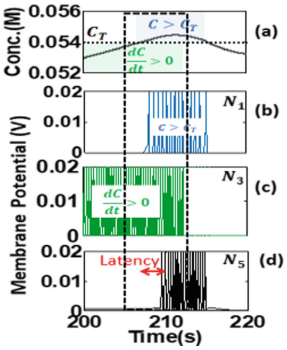

Fig. 4. (a) Concentration vs time (b) Response of N_1 for $C > C_T$ with **hard threshold** and (c) N_3 for dC/dt > 0 which produces (d) an AND function response at N_5 with a significant latency (red arrow). (Color figure online)

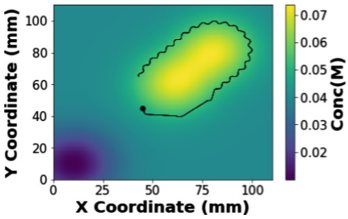

Fig. 5. Contour tracking with hard thresholding, C_T = 54 mM.

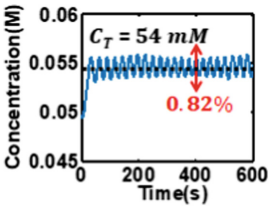

Fig. 6. Concentration tracking shows 0.82% deviation about set point due to hard-thresholding response of N_1.

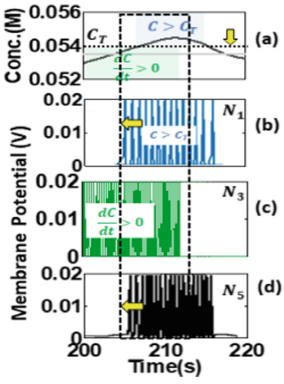

Fig. 7. (a) Concentration vs time (b) Response of N_1 with **soft threshold** for $C > C_T$ and (c) N_3 for dC/dt > 0 which produces (d) an AND function response at N_5 with a reduced latency (red arrow). (Color figure online)

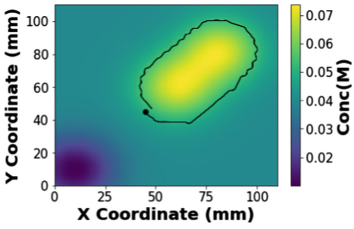

Fig. 8. Improved contour tracking with soft thresholding, C_T = 54 mM.

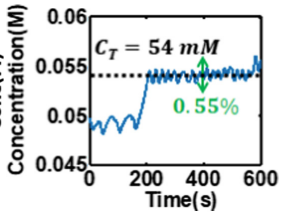

Fig. 9. Concentration tracking shows 0.55% deviation which is a 1.5X improvement due to softened response of N_1.

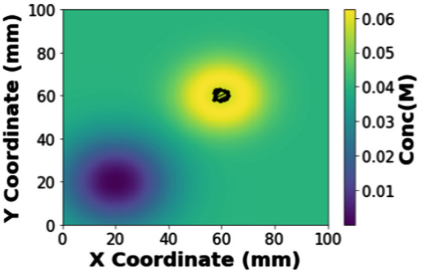

Fig. 10. Worm stuck in a valley (XOR sub-network is disabled).

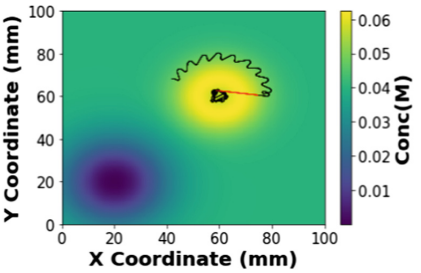

Fig. 11. The part of the trajectory marked in red is traversed when N_{12} fires, allowing the worm to escape and then resume tracking the C_T. (Color figure online)

4 Benchmarking

Table 1 benchmarks our network with previously reported contour tracking algorithms. We achieve state-of-the-art performance, with lower spiking frequencies, making our network more energy efficient. Table 2 shows the efficiency of our XOR gate implementation in terms of number of neurons used. It also works in an analog fashion unlike other reported SNN based gates, which is essential for our network.

Table 1. Benchmarks for contour tracking algorithm

Model	Max Freq (Hz)	External Bias Current	Average Deviation(%)
Santurkar et al. [3]	10	Yes	1± 0.13
Non-SNN	N.A	N.A	10± 2.9
This Work	5	No	0.55 ± 0.16

Table 2. Benchmarks for design of the XOR gate

Model	Input Values	No. of neurons for XOR
Delaney et al. [11]	Binary	6
Wade et al. [14]	Binary	12
Berger et al. [12]	Binary	7
Ferrari et al. [13]	Binary	23
This Work	Continuous	5

5 Hardware Feasibility

Hardware realization of such a SNN calls for both the feasibility as well as designability of the neuronal response. Recently our group has proposed and experimentally demonstrated a SOI MOSFET based LIF neuron [2]. The neuronal functionality has been achieved by using the SOI transistor's intrinsic carrier dynamics. The response of the SOI neuron shows high sensitivity with MHz order frequency range.

Figure 12a shows the TEM image of the fabricated SOI neuron. Figure 12b shows the response curves of such SOI neuron for different refractory periods (t_{ref}).

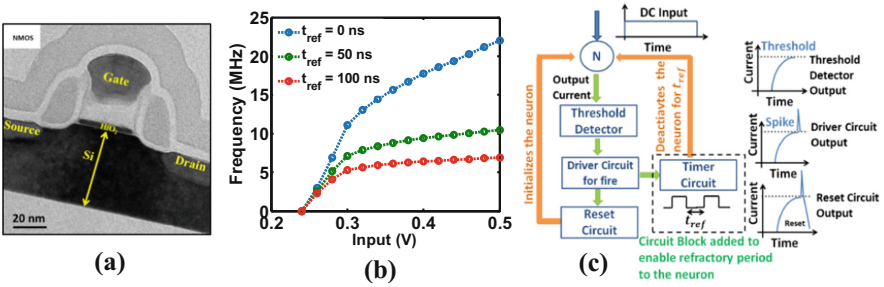

Fig. 12. (a) TEM image of the PD SOI MOSFET fabricated using 32 nm SOI technology [2]. (b) Experimental frequency vs. input curve. Without t_{ref}, the response increases sharply with input, whereas adding t_{ref} limits the frequency range. (c) Block diagram demonstrating the implementation of refractory period in SOI neuron. The neuron generates current output which is fed to the threshold detector. At threshold, the driver circuit elicits a spike enabling the timer circuit. The timer circuit deactivates the neuron during the refractory period. The reset circuit initializes the neuron. The expected transient output is shown at the output of three circuit block.

Without any refractory period, the response keeps increasing with input stimuli. Addition of t_{ref} limits the firing rate and the frequency saturates at a particular value like biological neurons. Such a tunable response provides freedom in SNN design for various applications and also aids the scope of hardware implementation. Figure 12c shows the block diagram for the implementation of refractory period in SOI neuron. The proposed electronic neuron is highly energy (35 pJ/spike) efficient and consumes lesser area ($\sim$1700 F^2 at 32 nm technology node) compared to state of the art CMOS neurons.

6 Conclusions

A complete end-to-end SNN based control circuit is proposed for chemotaxis in C. *elegans*. To implement this, analog rate-coded AND, OR, and XOR logic gates based inter-neuronal circuits are proposed. We implemented these gates using a small number of neurons, allowing for energy and area efficiency as well as reduced network latency, which is crucial for many robotics applications. The network latency was further reduced by modifying the response of the threshold detecting sensory neurons. We ensure correct operation of the network over arbitrary concentration ranges and choose parameters of the network using an analytic approach designed using the rate-coded approximation. The neuronal behaviors required to implement the neural logic gates are achieved by LIF neurons with configurable refractory periods. State-of-the-art accuracy of tracking is demonstrated (<0.6% deviation from set-point). To address the problem of being stuck around a local extrema en route to a set-point, we designed a novel XOR based sub-network that presents a biologically relevant solution. As XOR is a universal gate, this enables the implementation of any arbitrary logical functions in SNN.

Further, we show hardware implementation of such neurons on advanced 32 nm SOI platform.

Acknowledgement. The authors wish to acknowledge Nano Mission & MeitY, Government of India, for providing funding for this work.

Appendix: LIF Model and Ionic Currents

All the neurons used in our model are LIF neurons with refractory periods [17]. N_1, N_2, N_3 and N_4 have ionic channels that inject input current $I_e(t)$. The specific nature of $I_e(t)$ is what allows them to functions as threshold and gradient detectors. The ionic currents injected into N_1 and N_2 respectively are $I_{e1}(t)$ and $I_{e2}(t)$, given as:

$$I_{e,1}(t) = I_{e,0}\max(0,\ C - C_T - \delta);\ I_{e,2}(t) = I_{e,0}\max(0,\ C_T + \delta - C) \qquad (2)$$

The δ governs the degree of preemptive response of the threshold detectors. A set of equations that define $I_{e3}(t)$ and $I_{e4}(t)$ was proposed in [4] and used in [3]. We also use the same equations for the gradient detectors N_3 and N_4. These equations along with a detailed explanation can be found in Sect. II of [3].

References

1. Maas, W.: Networks of spiking neurons: the third generation of neural network models. Neural Netw. **10**(9), 1659–1671 (1997)
2. Dutta, S., et al.: Leaky integrate and fire neuron by charge-discharge dynamics in floating-body MOSFET. Sci. Rep. **7**, 8257 (2017)
3. Santurkar, S., Rajendran, B.: C. elegans chemotaxis inspired neuromorphic circuit for contour tracking and obstacle avoidance. In: Neural Networks, IJCNN (2015)
4. Appleby, P.A.: A model of chemotaxis and associative learning in C. elegans. Biol. Cybern. **106**(6–7), 373–387 (2012)
5. Gray, J.M., Hill, J.J., Bargmann, C.I.: A circuit for navigation in Caenorhabditis elegans. Proc. Natl. Acad. Sci. U. S. A. **102**(9), 3184–3191 (2005)
6. Galarreta, M., Hestrin, S.: Fast spiking cells and the balance of excitation and inhibition in the neocortex. In: Hensch, T.K., Fagiolini, M. (eds.) Excitatory-Inhibitory Balance. Springer, Boston (2003). https://doi.org/10.1007/978-1-4615-0039-1_11
7. Kato, S., et al.: Temporal responses of C. elegans chemosensory neurons are preserved in behavioral dynamics. Neuron **81**(3), 616–628 (2014)
8. Liu, Q., Hollopeter, G., Jorgensen, E.M.: Graded synaptic transmission at the Caenorhabditis elegans neuromuscular junction. Proc. Natl. Acad. Sci. U. S. A. **106**, 10823–10828 (2009)
9. Goldental, A., et al.: A computational paradigm for dynamic logic-gates in neuronal activity. Front. Comput. Neurosci. **8**, 52 (2014)
10. Yang, J., Yang, W., Wu, W.: A novel spiking perceptron that can solve XOR problem. ICS AS CR (2011)
11. Reljan-Delaney, M., Wall, J.: Solving the linearly inseparable XOR problem with spiking neural networks. https://doi.org/10.1109/sai.2017.8252173

12. Berger, D.L., de Arcangelis, L., Herrmann, H.J.: Learning by localized plastic adaptation in recurrent neural networks (2016)
13. Ferrari, S., et al.: Biologically realizable reward-modulated Hebbian training for spiking neural networks. In: Neural Networks, IJCNN (2008)
14. Wade, J., et al.: A biologically inspired training algorithm for spiking neural networks. Dissertation. University of Ulster (2010)
15. Kunitomo, H., et al.: Concentration memory-dependent synaptic plasticity of a taste circuit regulates salt concentration chemotaxis in Caenorhabditis elegans. Nat. Commun. **4**, 2210 (2013)
16. Suzuki, H., et al.: Functional asymmetry in Caenorhabditis elegans taste neurons and its computational role in chemotaxis. Nature **454**(7200), 114 (2008)
17. Naud, R., Gerstner, W.: The performance (and limits) of simple neuron models: generalizations of the leaky integrate-and-fire model. In: Le Novère, N. (ed.) Computational Systems Neurobiology. Springer, Dordrecht (2012). https://doi.org/10.1007/978-94-007-3858-4_6