Biomimetic, Soft-Material Synapse for Neuromorphic Computing: from Device to Network

Md Sakib Hasan[*]✉, Catherine D. Schuman[‡], Joseph S. Najem[†‖], Ryan Weiss[*],Nicholas D. Skuda[*],
Alex Belianinov[§], C. Patrick Collier[§], Stephen A. Sarles[‖], and Garrett S. Rose[*]
[*]Department of Electrical Engineering & Computer Science, University of Tennessee, Knoxville, TN 37996 USA
[†]Joint Institute for Biological Sciences, Oak Ridge National Laboratory, Oak Ridge, TN 37831 USA
[‡]Computer Science and Mathematics Division, Oak Ridge National Laboratory, Oak Ridge, TN 37831 USA
[§]Center for Nanophase Materials Sciences, Oak Ridge National Laboratory, Oak Ridge, TN 37831 USA
[‖]Deparment of Mechanical, Aerospace and Biomedical Engineering, University of Tennessee, Knoxville, TN 37916 USA
✉Corresponding Author, mhasan4@vols.utk.edu

***Abstract*—Neuromorphic computing refers to a variety of brain-inspired computers, devices, and models inspired by the interconnectivity, performance, and energy efficiency of the human brain. Unlike the ubiquitous von Neumann computer architectures with complex processor cores and sequential computation, biological neurons and synapses operate by storing and processing information simultaneously with the capacity of flexible adaptation resulting in massive computational capability with much less power consumption. The search for a synaptic material which can closely imitate bio-synapse has led to an alamethicin-doped, synthetic biomembrane which can emulate key synaptic functions due to generic memristive property enabling learning and computation. This two-terminal, biomolecular memristor, in contrast to its solid-state counterparts, features similar structure, switching mechanism, and ionic transport modality as biological synapses while consuming considerably lower power. In this paper, we outline a methodology for using this biomolecular synapse to build neural networks capable of solving real-world problems. The physical mechanism underlying its volatile memristance is explored followed by the development of a model of this device for circuit simulation. We outline a circuit design technique to integrate this synapse with solid-state neuron circuit for hardware implementation. Based on these results, we develop a high level simulation framework and use a training scheme called Evolutionary Optimization for Neuromorphic System (EONS) to generate networks for solving two problems, namely iris dataset classification and EEG classification task. The small network size and comparable to state-of-the-art accuracy of these preliminary networks show its potential to enhance synaptic functionality in next generation neuromorphic hardware.**

I. Introduction

The impending end of Moore's Law, the breakdown of Dennard Scaling, the low bandwidth between CPU and memory known as the von Neumann bottleneck coupled with ever increasing computing demands is driving researchers to explore alternative computing paradigm [1]. Neuromorphic computing has emerged in recent years as a complementary architecture to von Neumann systems. Compared to traditional architecture, neuromorphic computers comprising artificial neurons and synapses provide a better platform for a more efficient implementation of neural network algorithms. However, even after significant advances in very-large-scale-integration (VLSI) circuits [2], neuromorphic networks are still far from achieving the complexity, neuronal density and power efficiency of the human brain. The brain uses sophisticated molecular mechanisms to continually reconfigure connectivity between neurons and the resulting synaptic plasticity [3] enables the brain to remember patterns and adapt to incoming information, as well as perform massive amounts of parallel operations with significantly low power consumption [4]. In contrast, VLSI networks emulate synaptic activities using transistors, bearing little resemblance to bio-counterparts at the mechanism level, and require significantly large, power-hungry complementary metal-oxide-semiconductor (CMOS) circuitry.

Most state-of-the-art solid state emerging devices used for neuromorphic circuits have been developed with the primary target of high integration density and computational power-efficiency, and for the most part, have disregarded biological realism on both structural and functional levels. A possible alternative approach is to design more biologically faithful systems that are energy-efficient, soft, stochastic, fault-tolerant, and preferably biological. To this end, a biomolecular memristor(memory resistor) with composition, structure, switching mechanism, and ionic transport similar to bio-synapses has been recently reported by Najem et al. [5]. This device can emulate key synaptic functions including paired-pulse facilitation and depression due to a generic memristive property, enabling learning and computation while consuming considerably lower power. Moreover, the synapse-like dynamic properties of the device enables simplified learning circuit implementations. In this paper, we outline a methodology for using this bio-synapse to build neural networks for solving real-world problems. The small network size and comparable to state-of-the-art accuracy of these preliminary networks show its potential to enhance synaptic functionality in next generation neuromorphic hardware.

The remainder of the paper is organized as follows: In Section II, we explore the physical mechanism resulting in volatile memristance. The device model and SPICE implementation for circuit level simulation is described in Section III. In Section IV, we outline a circuit design technique to integrate this synapse with solid-state neuron circuit for hardware implementation. Based on these results, a high level simulation framework is developed in Section V where we use a training scheme called Evolutionary Optimization for Neuromorphic System (EONS) to generate networks for solving two classification problems, namely the IRIS dataset classification task and EEG classification task. Finally, we discuss our results and conclude the paper in Section VI.

II. Biomimetic Synaptic Device

Biological neurons and synapses operate by storing and processing information simultaneously, while maintaining the capacity to adapt. These attributes, co-location of computing and memory, as well as plasticity are unique to the brain, allowing it to perform massive computational operations while consuming as little as 20 watts [4]. Neuromorphic circuits based on traditional silicon-based circuitry fail to mimic basic transport properties of biological synapses and neural networks, and, as a result, require far more complex neural networks and power to achieve similar computational capability. Meaningful insight into how the brain processes and acts on information in complex environments requires developing easily configurable devices that emulate nature's biological neural design and the complex biomolecular processes responsible for brain memory and computing. To this end, a soft, two-terminal biomolecular memristor device that mimics the physical structure, switching mechanism, and ion transport of bio-synapses has been recently demonstrated [5]. This device consists of an alamethicin-doped synthetic biomembrane that is 3-5 nm in thickness (Fig. 1). In brief, the highly insulating ($\sim$10 GΩ) lipid membrane self assembles at the interface of two in-contact, lipid-encased aqueous droplets placed in hexadecane oil. In the presence of alm peptides and sufficient transmembrane voltage, conductive and memristive ionic pathways are created through volatile, voltage-driven insertion of alamethicin peptides (alm) into the insulating lipid membrane (Fig. 1). At low voltages, where alm peptides are surface-bound, the device is considered to be in the resting state. However, the device abruptly switches into a voltage-dependent conductive state at voltages exceeding a certain potential, $V_{threshold}$. This response closely resembles the voltage-modulated variable conductance in biosynapses. Current-voltage relationship of the device in response to harmonic voltage input exhibits pinched hysteresis which demonstrates the memrisitive nature of the two-terminal system.

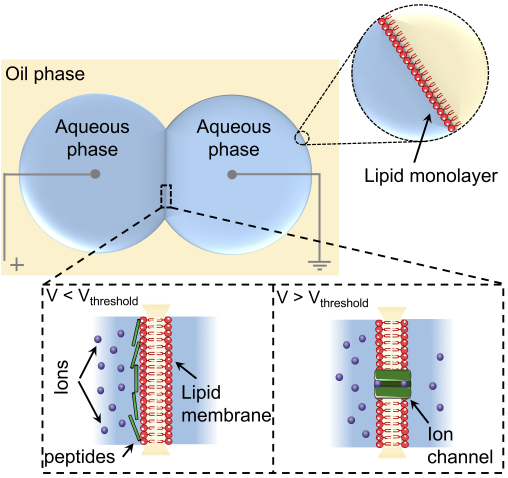


Fig. 1: Biomolecular memristors with alamethicin ion channels for voltage-controlled signal transmission

Experiments and simulation illustrated voltage-dependent threshold switching and volatile memristive behavior governed by two voltage-dependent state variables: the areal density of alamethicin channels, and the increase in membrane area due to electrowetting, which in turn dictate the total number of ion channels, and, thus, the net conductance of the device. As a result, the two-terminal device exhibited switching dynamics that are comparable to depolarizing pulses in actual nerve cells, with short- and long-term plasticity such as paired-pulse facilitation (PPF), paired-pulse depression (PPD), and, when paired with a non-volatile memristor, spike timing dependent plasticity (STDP) [5]. Compared to prior memristive devices, this biomolecular memristor consumes significantly less power (0.1-10 nW) and is easier to fabricate.

III. Device Model and SPICE Implementation

The current-voltage relationship of a generic voltage-controlled memristor can be written as

$$I = G(x)V \tag{1}$$

$$\frac{dx}{dt} = f(x; V) \tag{2}$$

Here, G is the nominal memory conductance and x represents one or more voltage-controlled state variables that control the conductance. The conductance of our biomolecular memristor is determined by the total number of alm pores creating ion-channels across the insulating membrane, which in turn depends on the areal density of alm pores and the nominal size of the bilayer. Both these factors are controlled by voltage giving the device its voltage-controlled memristive nature. By choosing the number of open alm pores per unit area, N_a, and the fractional increase in bilayer area, A_m, as two state variables, Eq. 1 can be rewritten as,

$$I = G(N_a, A_m)V \tag{3}$$

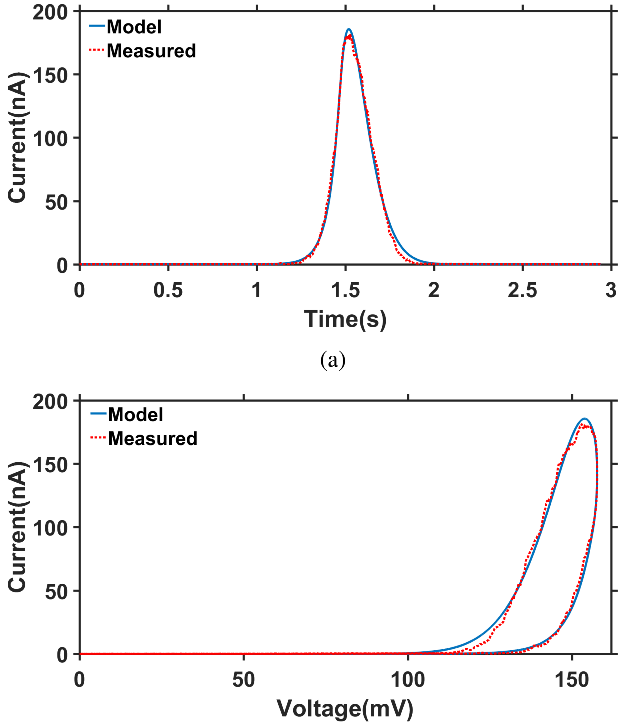
For an applied voltage V, the state equations for N_a and A_m, as derived in [5], are

$$\frac{dN_a}{dt} = a_1 + a_2 N_a \tag{4}$$

$$\frac{dA_m}{dt} = b_1 + b_2 A_m \tag{5}$$

where,

$$a_1 = \frac{N_0 exp(|V|/V_e)}{\tau_0 exp(|V|/V_\tau)}, \quad a_2 = \frac{-1}{\tau_0 exp(|V|/V_\tau)}$$
$$b_1 = \frac{\alpha V^2}{\tau_{ew}}, \quad b_2 = \frac{-1}{\tau_{ew}}$$



(a)

(b)

Fig. 2: Model Validation, (a) Current response over time and (b) Pinced hysteresis loop for 0.17 Hz 160 mV triangular wave input.

Here, V_e , N_0, V_τ and τ_0 are the voltage required to cause an e-fold increase in the number of alm pore, a proportionality constant that represents the number of alm pores at zero volts, the voltage required to induce an e-fold increase in τ, and the time constant for pore closure at zero volts, respectively. α and τ_{ew} are voltage-sensitivity constant, and characteristic time constant describing electrowetting process, respectively. Combining these results, the overall conductance can be written as,

$$G(t) = G_u N_a(t) A_0 (1 + A_m(t)) \tag{6}$$

where A_0 and G_u are bilayer area at zero volts and average unit conductance of a single alm pore determined by both the structure of the alm pore and the conductivity of the electrolyte solution within the droplets. τ_{ew} and α were determined by fitting numerical solutions of Eq. 5 to the measured change in membrane area during voltage sweeps and these, along with measured relaxation time constants (τ_0), were used in another fitting routine to estimate the parameters for alm insertion by fitting numerical solutions of Eq. 4 to measured *I*-*V* responses. The current response over time and *I*-*V* hysteresis curve of DPhPC memristor for 0.17 Hz triangular wave of amplitude 160 mV is shown in Fig. 2. As seen in this figure, the model fits reasonably well with the experimental results. More details on the analytical modeling for pulse and sinusoidal inputs can be found in [6].

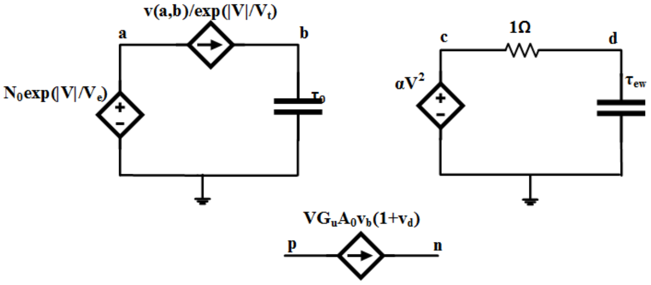


Fig. 3: SPICE Implementation

A subcircuit, shown in Fig. 3, was developed to implement the model in SPICE. Here, p and n represent the positive and negative terminals which are arbitrary since this is a bidirectional symmetric device where the terminals can always be interchanged.The node voltage v_b and v_d are not physical voltages. They represent N_a and A_m, respectively. The key underlying idea is to reformulate equation 1 and 2 in terms of RC circuit. For a series RC circuit driven by a voltage source V_{in}, the equation describing the voltage across the capacitor is

$$\frac{dv_c}{dt} = \frac{V_{in}}{RC} - \frac{v_c}{RC} \tag{7}$$

If we replace R, C and V_{in} with $exp(\frac{V}{V_\tau})$, τ_0 and $N_0 exp(\frac{V}{V_e})$, we get back Eq. 4 (top left in Fig. 3) and if we replace those with 1, τ_{ew} and αV^2 we get back Eq. 5 (top right in Fig. 3). We use voltage controlled voltage source for V_{in} and behavioral current source to implement voltage dependent resistance. Finally, we combine Eq. 6 and Eq. 3 using a behavioral current source to get the current through the device.

IV. Neuromorphic Hardware Implementation

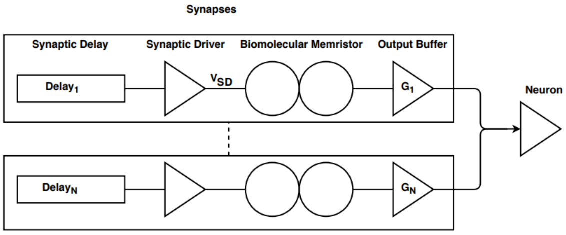


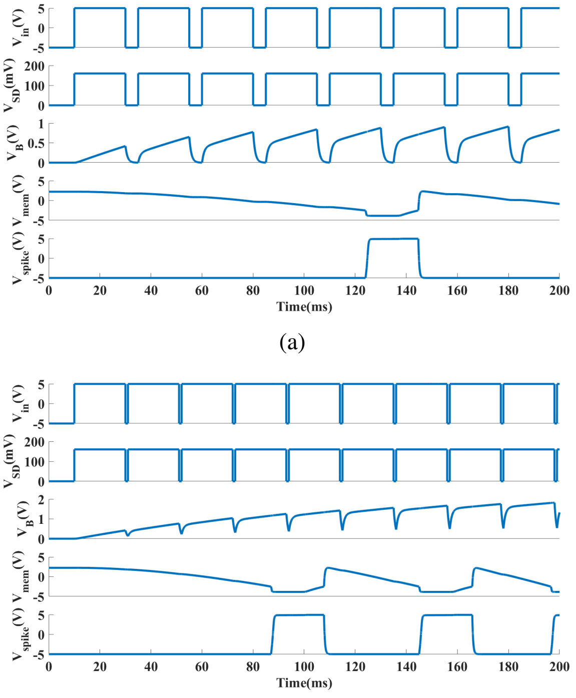
Fig. 4: Block Diagram of Hardware Components for Synapses to Neuron

The hardware design for this spiking neural network architecture takes advantage of the unique characteristics of the synaptic device. The neuromorphic architecture uses a leaky integrate and fire neuron model that is emulated with solid state electronics. The synaptic connections in the system are given signed weights and delays and connect neurons to build neural networks to solve tasks. A high level simulation uses the hardware characteristics for each component to train networks built from the components. The circuits for these networks can then be physically built and tested. With consistency across network simulation and the internal components, these circuits verify the effects of a soft material synapse as an online learning feature for spiking neural networks.

The networks seen in Section V, emulate two basic building blocks for spiking neural networks, the neuron and the synapse. There are four components used in each synapse, the delay block, the synapse driver, the synapse, and the output amplifier. Fig. 4 represents all the synapses connected to a single neuron. The inputs into each synapse are generated from neurons. The depicted neuron's output feeds into synapses connected to subsequent neurons. The delay block takes the output of a neuron as input and recreates that output with a specified time delay. There are many realizations for the delay block including a simple inverter chain. The biomolecular memristor is activated by a synaptic driver circuit. The synaptic driver circuit takes the output of a neuron and converts it to the appropriate voltage level for the input spike(V_{SD}) for the biomolecular memristor, 160 mV in Fig. 5. Careful consideration must go into the circuitry surrounding the biomolecular memristor because a voltage drop across the biomolecular memristor greater than a few hundred millivolts can destroy the device. The synaptic weight is defined by the impedance of the biomolecular memristor. The impedance changes of the memristor, as described in Section II, give the system short term online learning capabilities. The biomolecular memristor output is buffered before entering its connected neuron. The output buffer has a controllable gain factor, G_i. The output node is held at a virtual ground by the output buffer giving a weighted output current from the biomolecular memristor.

$$I_{neuron}(t) = \sum_{i=1}^{N} G_i \cdot I_{BM_i}(t) \tag{8}$$

The input current into a neuron, I_{neuron}, can be written as Eq. 8. The current is the sum of the output current of all the connected synapses, I_{BM_i}. This is multiplied by the gain, G_i, of the output buffer. In Fig. 5 V_B is the output current from the biomolecular memristor times the gain G_i. The neuron stores the summed current and discharges a spike when crossing its threshold. The circuit implementation used for an integrate and fire neuron consists of two stages, first an integrator followed by a comparator. The neuron takes the summed current, I_{Neuron}, from the output buffers of every connected synapse and integrates it on the capacitor in the integrator, C_{Mem}. For every synaptic event, the voltage, V_{Mem}, is the integration of the input current, as seen in Eq. 9. The capacitance of the feedback network of the integrating operational amplifier determines the accumulation rate of the neuron. The voltage on the output of this op-amp, V_{Mem}, is compared to the threshold voltage, $V_{Threshold}$ by a comparator. A threshold of -2.5V is used in simulations seen in Fig. 5. When the accumulated voltage drops below the threshold voltage the output of the second op-amp goes from negative to positive rail creating an output spike, V_{spike} in Fig. 5. The output spike width matches the input spike width. At the end of a spike V_{Mem} resets and begins accumulating inputs shortly after. The time taken between resetting V_{Mem} and integrating input current is the neuron's refractory period.



(a)

(b)

Fig. 5: Simulation result of the neuromorphic circuit with 160 mV pulse train input of 20 ms pulse width for two different off times; (a) off time= 5 ms, (b) off time = 1 ms.

$$V_{mem}(t) = \frac{1}{C_{Mem}} \int_{0}^{t_{spike}} I_{neuron}(t)dt \tag{9}$$

Network design parameters such as spike width, voltage, and refractory period must be adjusted according to the device parameters to maximize short term effects. Fig. 5 illustrates how the PPF capability of the synapse results in more output spikes as we reduce the off time. Spike width, voltage and quantity of spikes needed to solve a given task determine the time and power efficiency of the network. Optimizing these parameters can improve the energy efficiency for solving a particular task. This circuit implementation has all the necessary components to implement the neural network architecture and has been experimentally verified [7].

V. Training: Evolutionary Optimization

It is not immediately clear how to utilize the unique characteristics that this device will offer, nor is it clear how to hand-construct networks of these components that are capable of performing real-world tasks. There are multiple characteristics of the device that make it unsuitable for traditional training approaches such as back-propagation. For example, the device's short-term plasticity updates the synaptic weights based on activity in the network and returns the weights to an equilibrium state if no activity is present. This prevents the use of a traditional training algorithm that relies on being able to customize weights. However, it adds a more complex functionality that can be leveraged in real applications.

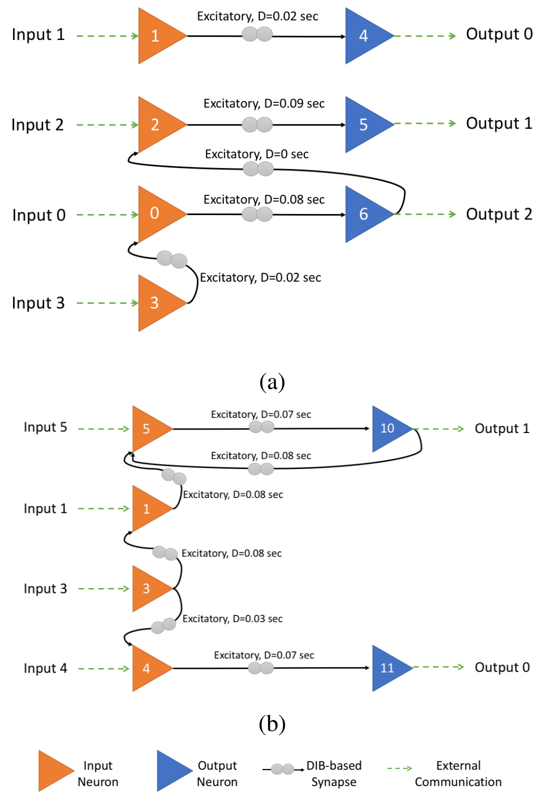
In this work, we utilize an existing neuromorphic training scheme based on evolutionary optimization or genetic algorithms called Evolutionary Optimization for Neuromorphic System (EONS) [8]. Unlike other neuromorphic training algorithms, EONS will attempt to optimize within the characteristics and constraints of the system. It determines the number of neurons and synapses required, as well as how those components are connected together. In the case of these networks, EONS also determines just a few parameters of each synapse: the delay value on each synapse and whether the synapse should be excitatory or inhibitory. In order to utilize the EONS framework, we have implemented a high-level simulation of the full neuromorphic circuit, simulating the biomolecular synapses based on the model described in Section III.

A. Training Results

We used EONS to build the appropriate network for this neuromorphic implementation for two tasks: the iris dataset classification task [9] and an EEG classification task [10]. In the iris task, there are four inputs (petal length and width and sepal length and width) and three outputs, which correspond to the three classes of iris flowers. Since the iris task is a static classification task (i.e., there is no temporal component to the data), we encode the input values temporally, by using multiple pulses per input over time to encode the input values. In particular, we normalize the data inputs to be integer values between 0 and 10; then, the normalized value is used to determine how many input pulses to apply to the corresponding input neuron. For example, if the normalized value is 2, then 2 pulses are applied to the associated input neuron.

For the EEG task, there is one input, which is the EEG signal over time, and two outputs, which correspond to the two output classes (healthy and epileptic). Because the EEG task is temporal in nature and there is a single input signal, we encode the signal over ten input neurons (one for 0-0.1, one for 0.1-0.2, and so on). During each time step, the input neuron stimulated depends on the magnitude of the signal. We frame the EEG task as a classification (healthy vs. epileptic) rather than on-set detection (detecting when a seizure is about to occur). For both iris and EEG, we count the number of output spikes on each of the output neurons, and whichever output neuron fires the most corresponds to the assigned class.

Since the goal is to demonstrate that these devices are capable of performing tasks in a neuromorphic system, we used EONS to construct several networks for each task and illustrate the best performing networks in this work. The best performing network for the iris task is shown in Fig. 6a. This network achieves 96 percent accuracy on the training set and 97.33 percent accuracy on the testing set, which is comparable with other neuromorphic implementation results in [8]. Fig. 6b shows the resulting network for the EEG task. Though there are ten total input neurons for this task, we do not show six of them, as they are not connected to other input or output neurons. This network achieves 98.25 percent accuracy on both the training and the testing set, which is comparable with other results on the classification version of this task [11].



(a)

(b)

Fig. 6: Biomolecular memristive synapse based network built using EONS for two classification tasks, (a)iris, (b)EEG.

For both of these networks, there are no hidden neurons required to get satisfactory results. Moreover, the number of synapses required to solve this task is much smaller than we would expect to see from other neuromorphic implementations, especially for the EEG task. We expect that this can be attributed to the more complex behavior enabled by the biomolecular synapses (e.g., PPF and PPD), though we plan to explore this in more detail in future work.

VI. Discussion and Conclusion

Neuromorphic computing architectures attempt to emulate the energy efficiency and adaptability of the human brain. To achieve similar power dissipation and adaptability, we use a biomimetic soft-material for synaptic connections. The synaptic device has memristive properties that directly emulate biological synaptic plasticity. We built a model for the device that captures the dynamic characteristics and used it in circuit simulation. We surrounded the device with the necessary circuitry for implementing our spiking neural network architecture. From these simulations, we know the device works as a synapse, and we see the impact of the device on the spike rates of synapses to neurons. The architecture can be improved by building devices and the circuitry for energy efficiency that does not degrade computational performance. It has been shown that using a software tool called EONS (based on evolutionary optimization), and a simulation of the biomimetic soft-material synapses and solid-state neurons, we were able to construct networks of those components that can solve problems. In particular, we have shown networks for iris and EEG classification tasks, that are composed of biomimetic soft-material synapses and solid-state neurons and have achieved comparable to state-of-the-art results. We plan to continue using EONS to build networks for other applications and to use what we learn from those networks to drive the co-design of a programmable device using these components.

Acknowledgment

This material is based in part upon research sponsored by the National Science Foundation under Grant No. NCS-FO-1631472. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon.

References

[1] C. Li, M. Hu, Y. Li, H. Jiang, N. Ge, E. Montgomery, J. Zhang, W. Song, N. Dávila, C. E. Graves *et al.*, “Analogue signal and image processing with large memristor crossbars,” *Nature Electronics*, vol. 1, no. 1, p. 52, 2018.

[2] P. A. Merolla, J. V. Arthur, R. Alvarez-Icaza, A. S. Cassidy, J. Sawada, F. Akopyan, B. L. Jackson, N. Imam, C. Guo, Y. Nakamura *et al.*, “A million spiking-neuron integrated circuit with a scalable communication network and interface,” *Science*, vol. 345, no. 6197, pp. 668–673, 2014.

[3] M. Mayford, S. A. Siegelbaum, and E. R. Kandel, “Synapses and memory storage,” *Cold Spring Harbor perspectives in biology*, vol. 4, no. 6, p. a005751, 2012.

[4] M. P. van den Heuvel, C. J. Stam, R. S. Kahn, and H. E. H. Pol, “Efficiency of functional brain networks and intellectual performance,” *Journal of Neuroscience*, vol. 29, no. 23, pp. 7619–7624, 2009.

[5] J. S. Najem, G. J. Taylor, R. J. Weiss, M. S. Hasan, G. Rose, C. D. Schuman, A. Belianinov, C. P. Collier, and S. A. Sarles, “Memristive ion channel-doped biomembranes as synaptic mimics,” *ACS nano*.

[6] M. S. Hasan, J. S. Najem, R. J. Weiss, C. D. Schuman, A. Belianinov, C. P. Collier, S. A. Sarles, and G. Rose, “Response of a memristive biomembrane and demonstration of potential use in online learning,” in *2018 IEEE Nanotechnology Materials and Devices Conference (NMDC)*, Portland, OR, 2018.

[7] R. Weiss, J. S. Najem, M. S. Hasan, C. D. Schuman, A. Belianinov, C. P. Collier, S. A. Sarles, and G. S. Rose, “A soft-matter biomolecular memristor synapse for neuromorphic systems,” in *2018 IEEE Biomedical Circuits and Systems Conference (BioCAS)*, Cleveland, OH, 2018.

[8] C. D. Schuman, J. S. Plank, A. Disney, and J. Reynolds, “An evolutionary optimization framework for neural networks and neuromorphic architectures,” in *Neural Networks (IJCNN), 2016 International Joint Conference on*. IEEE, 2016, pp. 145–154.

[9] D. Dheeru and E. Karra Taniskidou, “UCI machine learning repository,” 2017. [Online]. Available: http://archive.ics.uci.edu/ml

[10] R. G. Andrzejak, K. Lehnertz, F. Mormann, C. Rieke, P. David, and C. E. Elger, “Indications of nonlinear deterministic and finite-dimensional structures in time series of brain electrical activity: Dependence on recording region and brain state,” *Physical Review E*, vol. 64, no. 6, p. 061907, 2001.

[11] J. J. M. Reynolds, J. S. Plank, C. D. Schuman, G. Bruer, A. W. Disney, M. Dean, and G. S. Rose, “A comparison of neuromorphic classification tasks,” in *International Conference on Neuromorphic Computing Systems*. Knoxville, TN: ACM, July 2018.