Neurocomputing 381 (2020) 89–104

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

A neuromorphic SLAM architecture using gated-memristive synapses

Alexander Jones[a,*], Andrew Rush[a], Cory Merkel[b], Eric Herrmann[a], Ajey P. Jacob[c], Clare Thiem[d], Rashmi Jha[a]

[a] The University of Cincinnati, 2851 Woodside Drive, Cincinnati, OH 45219, United States
[b] Rochester Institute of Technology, 1 Lomb Memorial Drive, Rochester, NY 14623, United States
[c] GLOBALFOUNDRIES, 400 Stone Break Road Extension, Malta, NY 12020, United States
[d] Air Force Research Laboratory, 525 Brooks Road, Rome, NY 13441-4505, United States

ARTICLE INFO

Article history:
Received 2 May 2019
Revised 26 September 2019
Accepted 29 September 2019
Available online 15 November 2019

Communicated by Dr. Fırat Kacar

Keywords:
SLAM
Gated-Memristors
Neuromorphic architecture
Associative learning

ABSTRACT

Navigation in GPS-denied environments is a critical challenge for autonomous mobile platforms such as drones. The concept of simultaneous localization and mapping (SLAM) addresses this challenge through real-time mapping of the platform's surroundings as it explores its environment. The computational resources required for traditional SLAM implementations (e.g. graphical processing units) require large size, weight, and power overheads; making it infeasible to employ them in resource-constrained applications. This work proposes a self-learning hardware architecture utilizing a novel gated-memristive device to address the implementation of SLAM in an energy-efficient manner. The gated-memristive devices are implemented as electronic synapses in tandem with novel low-energy spiking neurons to create a spiking neural network (SNN). This work shows how the SNN allows for navigation through an environment via landmark association without needing GPS. In the simple environment in which the network exists, it can successfully determine a direction in which to navigate while only consuming 36 μW of power and only needing to be exposed to each landmark within the environment for 1-2ms in order to remember that location.

1. Importance of tackling SLAM with neuromorphic circuits

As technology advances, mobile robots and drones are widely used in industrial, defense, and aerospace applications. Smaller robots such as micro-drones have even become more prevalent as technology has become compact and power efficient. These types of technology can either be piloted via remote control, or piloted autonomously via either onboard or off-board software. Autonomously piloting vehicles through dynamic environments is especially challenging and requires a robust navigation system. The systems that navigate these varied environments must be robust to the point of managing micro and macro-level tasks. The navigational system for a drone requires real-time control tasks such as auto-balancing with varying wind speed conditions, performing standard obstacle avoidance, and controlling the system's altitude and velocity. Simultaneously, the navigational system must complete macro-level tasks including reaching its next objective and how to navigate around obstacles efficiently.

Many mobile systems that must make complex, macro-level decisions to reach objectives require GPS or supplementary off-board sensing equipment [1–3]. While GPS can be very useful in macro-level navigation, constantly using it requires large amounts of power [4,5]. All autonomous flight systems have tremendous power requirements devoted to flight, onboard systems, etc. [6]. These power requirements leave very little space in the system's power capacity for tasks such as computation and navigation. Therefore, those systems must be made as power efficient as possible. Processing navigational decisions could be offloaded to a computer not on the drone, but this exposes additional problems. If the robotic system or drone requires remote communication for navigation, this prevents the system from moving into places where it loses the signal, or the signal does not exist (e.g. indoors, subterranean environments, or extraterrestrial bodies where GPS does not exist). The system also allows itself to be vulnerable to signal jamming.

These issues create an application space where making local, power-efficient decisions for navigation are necessary. One example of such an application space would be the rovers NASA has sent to Mars. These systems need to be extremely power efficient, as their only current methods of recharging are via solar energy

* Corresponding author.
E-mail address: jones2a5@mail.uc.edu (A. Jones).

https://doi.org/10.1016/j.neucom.2019.09.098
0925-2312/

or radioisotope power systems [7,8]. Making quick and local decisions proves useful the Mars rovers, as signals sent to and from Mars have an average round-trip latency of 14 minutes [9]. The Mars rovers *Spirit* and *Opportunity* used such local-decision navigation systems to navigate between waypoints dictated by NASA engineers back on Earth [10]. Other use-case examples for energy-efficient, local-decision making for navigation include drones performing search and rescue missions in underground or indoor environments, reconnaissance missions in war zones, and deep-sea exploration.

The concept of navigating an environment via local decisions can be solved by answering the simultaneous localization and mapping (SLAM) problem [11,12]. The problem has been studied for several decades [11], and at a basic level describes how a system navigating an unknown environment should create a map of its surroundings. In addition to creating a map, the system should also be able to localize itself within the mapped environment when given previously observed stimuli.

Historically, some of the most common methods of solving the SLAM problem include the extended Kalman filter and particle filters [11]. The extended Kalman filter is a nonlinear mathematical method to estimate a system's location within an environment. Particle filters have many derivative versions [11], but are primarily a Monte Carlo approach to state estimation within an environment. These type of classical methods of solving SLAM typically involve populating a graph (data structure) with nodes and edges or creating full 3D meshes of the environment's terrain. These processes are assisted by external sensors such as odometers, range-finders, and cameras to interpret information about how far the vehicle has traveled and what objects/obstacles can be seen [13]. Some SLAM methods even use external information such as GPS to help improve the precision of the system's location [14]. A lot of these methods of solving SLAM have specific types of environments where they excel, but they often run into similar issues of having performance constrained by the systems they serve. If all the information for the graph, terrain mesh, etc. has to be processed or stored on the robotic/drone system, extensive memory is required to store this information and the power to process the information obtained is limited. These limitations clash with the classical methods of solving SLAM to obtain accurate state estimation [15]. Recently, other methods of solving SLAM have been studied to circumvent this issue.

Over billions of years, living organisms have developed complex neural systems to navigate their environment. Recently, the SLAM community has started to use these naturally-occurring navigation systems as inspiration to develop neural networks that can solve SLAM. Many of these neural network solutions involve implementing networks that use the neuronal concepts seen in mammals such as place cells [16,17] and grid cells [17,18]. Place cells act as randomly organized networks of neurons that have unique combinations of neurons fire within the network when the organism is near specific landmarks within its environment [16]. Grid cells act as a "mesh" of the organism's environment, and can track how far an organism has traveled in space (similar to the classical SLAM models using odometer data to estimate the system's current state) [18]. These two systems can be cross-referenced to localize the organism [19]. Previous neural network systems/algorithms that use concepts such as place/grid cells to solve SLAM include RatSLAM [20] and BatSLAM [21].

In the past, most of these neural network solutions for SLAM have been implemented using artificial neural networks (ANNs) [20–25]. These are neural networks that exist purely in simulation on von Neumann hardware, such as graphics processing units (GPUs), field-programmable gate arrays (FPGAs), or application-specific integrated circuits (ASICs). These systems often consume large amounts of power and memory as previously mentioned. The data bottlenecks to/from memory when reading and updating so many weight values can also bog down processing times [26–28]. Performance in these areas could improve if neural networks could transition from traditional computing frameworks to non-von Neumann architectures that are specifically tailored to handle neural networks. This concept is the primary goal of the field of neuromorphic computing, which aims at designing hardware architectures that can efficiently implement neural networks. Implementing neuromorphic networks that specifically solve SLAM could overcome the issues that plague von Neumann designs of neural networks.

Neurons and synapses are the two core building blocks to neural networks and are represented in many ways within different architectures. Synapses act as memory devices to store information within the network as it learns over time, and neurons act as nonlinear processing elements that analyze the information passed through synapses.

Many different methods have been explored over the years to represent neurons and synapses within hardware. One of the most common methods of representing neurons typically involves designing a CMOS-based device that generates an event or voltage spike on its output node when given enough stimuli on its input node. Receiving more input at the circuit's input node causes its output node to spike more frequently. These types of neurons are the core of the research area of spiking neural networks (SNNs) within the field of neuromorphic computing, and will be the type of neuron circuits used to build the neuromorphic architecture described in this work.

For representing synapses in neuromorphic architecture, the synapse device is usually implemented in either a single device or in a CMOS circuit [29,30]. The single device or CMOS circuit will hold either a binary state of 0 or 1 to indicate if the synaptic connection is excited or not, or it can possess an analog state within a range of possible values to indicate a degree of excitation. When built using a non-CMOS implementation, the most common synaptic design is usually a memristive device such as two-terminal resistive RAM (ReRAM) (Fig. 1(a)) or phase-change memory (PCM) devices [31]. The basic concept that drives the functionality of the two-terminal (2T) device is shifting the distribution of charged defects (such as ions or dopants) within a switching oxide (SOx) channel to change the overall resistance of the device [32]. This distribution of dopants can be changed by applying biases to the device's top electrode (TE) and bottom electrode (BE). The device's resistance is then used within the neuromorphic architecture as the synapse's potentiation level. A highly potentiated two-terminal synaptic device has a low resistance, while a synaptic device that hasn't been excited has a high resistance. Despite 2T memristive devices being widely used in neuromorphic computing, they suffer from large, state-dependent power consumption, variability, state-drift during inferencing, and the issue of sneak current when implemented into crossbar arrays [33]. To address these issues, gated-memristive devices have been developed [34–38], which is the type of memristive device the architecture within this work will use for its synaptic devices.

The primary goal of this work is to show the implementation of neuromorphic SLAM architecture with gated-memristive novel memories and its performance impact in GPS-denied environments. To show this capability, this work will use three novel concepts to construct its neuromorphic architecture:

- A double gated-memristive devices; a device that inherently detects spike coincidence.
- A novel design for a CMOS-based neuron circuit.
- A novel low-power architecture specifically aimed at solving an example of SLAM.

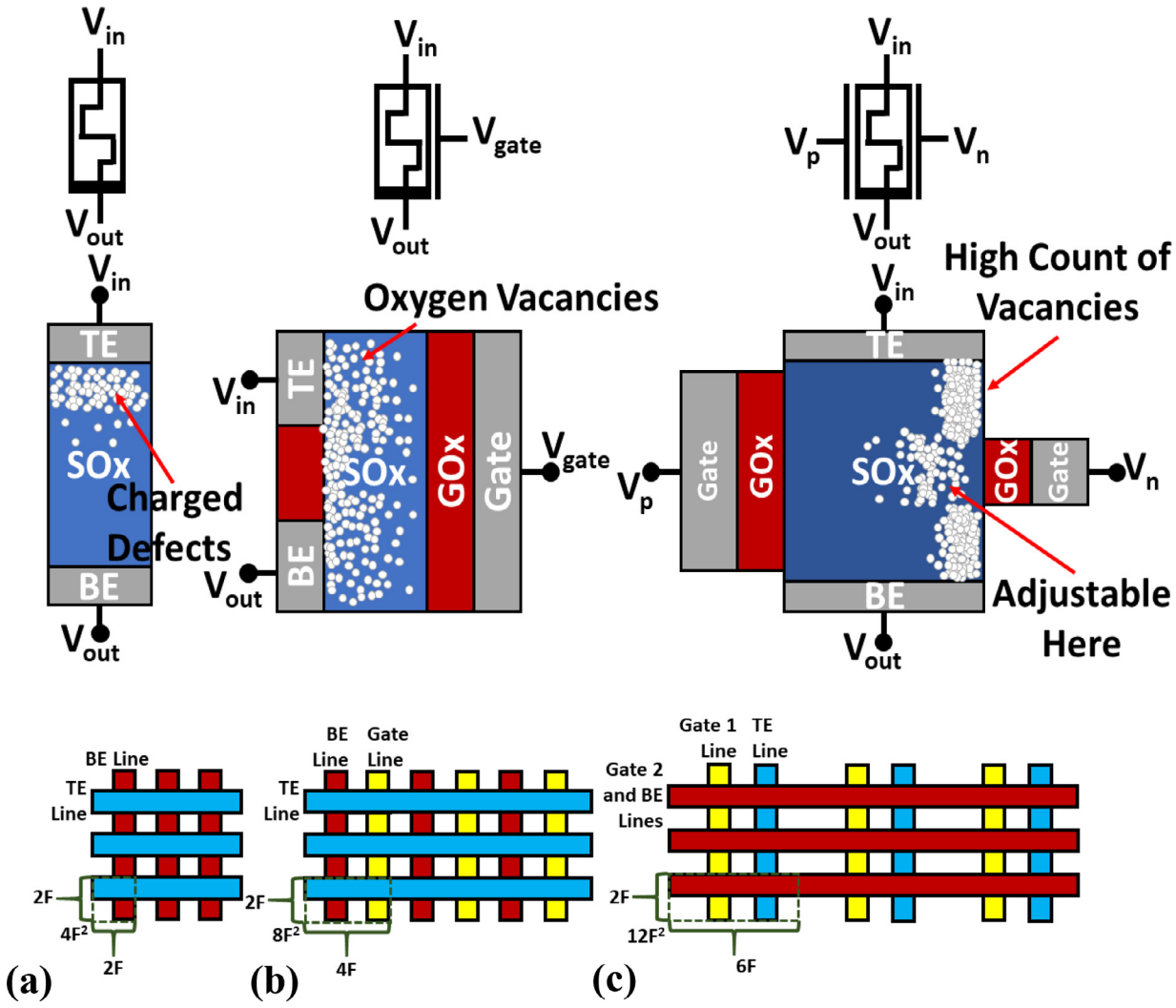


Fig. 1. (a) Diagram (sideview) depicting a classic design for a two-terminal memristive device along with a top-down diagram for the device in a 3×3 matrix architecture. Dopants within the device channel extend/retract in the SOx channel via applied potential to V_{in} and V_{out}. (b) Diagram showing design of the gated-memristive device from Herrmann et al. [34,35] along with a top-down diagram of the devices in a 3×3 matrix architecture if they were built in a planar fashion (rotated 90° from the sideview diagram). Oxygen vacancies within the SOx channel are moved away/toward the left side of the device to create a conductive path between TE and BE by applying a voltage to V_{gate}. Positive voltage pushes vacancies toward the left side of the device, while negative bias brings the vacancies toward the gate. (c) The proposed double gated-memristive device for the architecture proposed in this work along with a top-down diagram of a 3×3 matrix architecture of the devices built in planar fashion (rotated 90° from the sideview diagram). It should be noted that the Gate 1 and BE lines exist in parallel above and below the device. A high count of vacancies is placed along one side of the SOx channel and a high negative bias is applied across V_p and V_n to initially place the device in a higher resistive state. During operation, positive bias from V_p to V_n will potentiate the device, while negative bias will depress it. The bias applied from V_p to V_n must be greater than the SOx channel's threshold voltage (V_t), otherwise no change in the device's state will occur.

2. Gated-memristive devices

Gated-memristive devices operate in a slightly different fashion than their two-terminal counterparts. Previous gated-memristive devices developed by Herrmann et al. [34,35] operated by modulating the local concentration of oxygen vacancies (V_{o2+}) in SOx between TE and BE by applying appropriate bias to its extra gate terminal (Fig. 1(b)). This gate is isolated from the SOx channel via a layer of gate oxide (GOx). To lower the resistance, positive bias could be applied to V_{gate} which forces V_{o2+} towards the TE/BE side of the device. This process increased the conductivity of the path between TE and BE by increasing the concentration of charged carriers at the interface. Negative bias can be applied to V_{gate} to pull V_{o2+} towards the gate, increasing the device's resistance.

When compared to 2T ReRAM, gated-memristors offer additional features such as simultaneous read and write operations, lower write currents, state-independent write operations, and facilitating advanced learning via the additional gate [35]. Herrmann et al.'s work on strontium-titanate ($SrTiO_3$) gated-memristive devices provided measured device data along with a behavioral mathematical model for the devices to show these benefits, but offered no physics-based modeling approach in how the device operates. To develop a physics-based understanding of the original gated-memristive device, Appendix B describes a physics-based model of the devices in Verilog-A using Frenkel-Poole and ohmic conduction along with material parameters obtained from previous $SrTiO_3$ studies [39–42].

While this form of the gated-memristive device is useful, the design of the gated-memristive device could be further enhanced via the addition of a second gate. In the original gated-memristive device, resistive modulation of the device was via the gate on the device. If the circuit was required to detect simultaneous spikes from two separate neurons using only the gate, additional circuitry would be required. This circuitry would increase the size and add

additional active circuit elements to the architecture, which would go against the primary goals of the objective presented in this paper of solving SLAM in a space and power efficient manner. Adding an additional gate opposite of the original gate and placing the TE and BE on the top and bottom of the device (Fig. 1(c)) would allow for coincidence detection to be done within the device structure while still maintaining the benefits of the original gated-memristive design. Using a more conductive material for the SOx channel could also further enhance the device by allowing it to have a higher maximum output current to drive the CMOS circuitry that often is the common method for implementing spiking neuron circuits. A candidate for a more conductive channel oxide that won't change the core behavior of the gated-memristive device is Nb_2O_5 as previous work by Rush and Jha [43] and Bailey and Jha [40] has shown both Nb_2O_5 and $SrTiO_3$ to have similar analog memristive qualities.

A diagram of this double gated-memristive device can be seen in Fig. 1(c). Prior to device operation, a potential can be applied to V_p to shift defects to one side of the SOx channel. Negative voltage can then be applied to V_n to initially place the device in a high resistive state. A positive voltage bias from V_p to V_n at this point will cause the device to enter a lower resistive state (as long as the bias is above the SOx's threshold voltage for being modified, V_t), while a negative bias from V_p to V_n will shift the device to a higher resistive state. The benefit of the added gate comes in the form of enabling the device to perform coincidence detection with no additional circuitry required. For this work, the concept of coincidence detection is defined as two applied potentials (of opposite polarity) from two independent sources occurring simultaneously. For neuromorphic systems, these applied potentials come in the form of two spikes (one positive and one negative) from two separate neurons.

Top-down schematics for each of the memristive devices in a 3×3 matrix architecture can be seen at the bottom of Fig. 1. The double gated-memristive device will be the largest of the three designs with an area of $12F^2$ (where F is the minimum feature size of the device). However, the double gated-memristive device does allow for the removal of external circuits needed for detecting spiking events applied to the device. It also still brings the benefits of gated-memristive devices over two-terminal devices.

Just like the original gated-memristive device, a compact model can be created to model the new double gated-memristive device's behavior. This model considers a pair of series resistances within the device in the form of bulk-limited (relies on the properties of the dielectric [44]) Frenkel-Poole conduction and a reconfigurable Schottky/ohmic contact resistance at the device's TE and BE contacts. The simulated output current of the double gated-memristive device under different gate potentials as a function of time can be seen in Fig. 2(a). More conductance behavior of this model is shown in Appendix I.

The full double gated-memristive device model consists of a set of calculations to obtain the device's synaptic current (For a step by step process to the model described below, see the flowchart that describes the model in Fig. 2 within Appendix F.). To calculate the current through the device, the total resistance through the device is modeled as two series resistances. The first resistance is a bulk Frenkel-Poole resistance and the second is a Schottky/Ohmic contact resistance. The current is then calculated by

$$I_{syn} = \frac{V_{eff}}{R_{fp} + R_{SO}} \tag{1}$$

where R_{fp} and R_{SO} are the Frenkel-Poole and Schottky/Ohmic resistances, respectively. In this model, V_{eff} is calculated by the equation

$$V_{eff} = V_p - V_n \tag{2}$$

An addition to the double gated-memristive model is the concept of an internal threshold voltage (V_t) for the switching oxide for potentiation to occur. This is included since previous work has shown that NbOx has a threshold voltage for potentiation to occur [45]. If $V_{eff} < V_t$, V_{eff} is treated as 0V.

Frenkel-Poole resistance is given by the equation

$$R_{fp} = \frac{V_{in} - V_{out}}{q\mu N_c \frac{V_{in}}{L_{ch}} exp\left(-\frac{q}{kt}\left(\varphi_t - \sqrt{\frac{qV_{TE}}{L_{ch}\varepsilon_0\varepsilon_i\pi}}\right)\right) w_{max}^2} \tag{3}$$

This equation is the Frenkel-Poole current density equation seen from before but converted into a current by multiplying it by the square of the conductive region's maximum potential width (w_{max}). That current is then used in tandem with the voltage drop across the TE/BE of the device to calculate the resistance. All undescribed parameters are described in Appendix E (as are all other subsequently undescribed parameters for the Nb_2O_5 double gated-memristive model).

Schottky/ohmic resistance is the second resistance factored into the model. Its purpose is to model the Schottky/ohmic contact resistance at the contacts between the top and bottom electrodes and the switching oxide. It is defined by the equation [46]

$$R_{so} = R_B exp\left(\frac{\varphi_t - V_{in}}{kt}\right) exp\left(\frac{2(\varphi_t - V_{in})\sqrt{\varepsilon_i\varepsilon_0 m^*}}{\hbar\sqrt{N_c}}\right) \tag{4}$$

The N_c term represents the density of states value for the oxide channel for the double gated-memristive model. To calculate N_c, the equation is

$$N_c = \frac{w_{max}}{w_{max} + exp\left(-\left(\frac{w_c}{w_{max}} - w_{max}\right)\right)} N_{c,max} + N_{c,min} \tag{5}$$

which represents the density of states as a function of the current relative width of the conductive region (w_c) with respect to w_{max}. w_c (a modified version of the traditional drift equation) is given by the equation

$$w_c = w_{c,t-1} + \frac{t_{step}\mu_{o2vac}V_{eff}}{W_{ch}} - d, \tag{6}$$

where the added term to the drift equation, d, is a decay term that is dependent on the device's current conductance state. The term $w_{c,t-1}$ is simply the value of w_c at the previous timestep within the simulation. w_{ch} is the width of the SOx device channel.

Over time, the diffusion of V_{o2+} out of the conductive path region in the device is assumed to occur. Previous work has shown that in niobium oxide two terminal ReRAM devices, this temporal decay of the device's state appears Gaussian [43], where the peak of the device's decay occurs at some midpoint conduction value between its highest and lowest states. It can be hypothesized that this decay is due to vacancy diffusion. In lieu of this previous study, a Gaussian decay term (d) was added to the double gated-memristive model to mimic this behavior when calculating w_c. The decay term is calculated with the equation

$$d = \frac{d_{max}}{\sigma_n\sqrt{2\pi}} exp\left(\frac{-(N_c - M_n)^2}{2\sigma_n^2}\right) \tag{7}$$

where d_{max} is a scaling term to cap the maximum decay rate, σ_n is the standard deviation of defects within the SOx (N_c), and M_n is the mean value of N_c. The effect of this decay term on the output current of the device can be seen in Fig. 2(b). It should be noted that the d_{max} term for this figure was increased by several orders of magnitude to observe the influence of the decay in a reasonable timeframe and that the decay defined by d is usually much slower in the remainder of this work than what is shown in Fig. 2(b). As the figure shows, the conductive state of the device decays at varying rates depending on when the applied gate bias is released. Ensuring this decay rate matches the requirements for

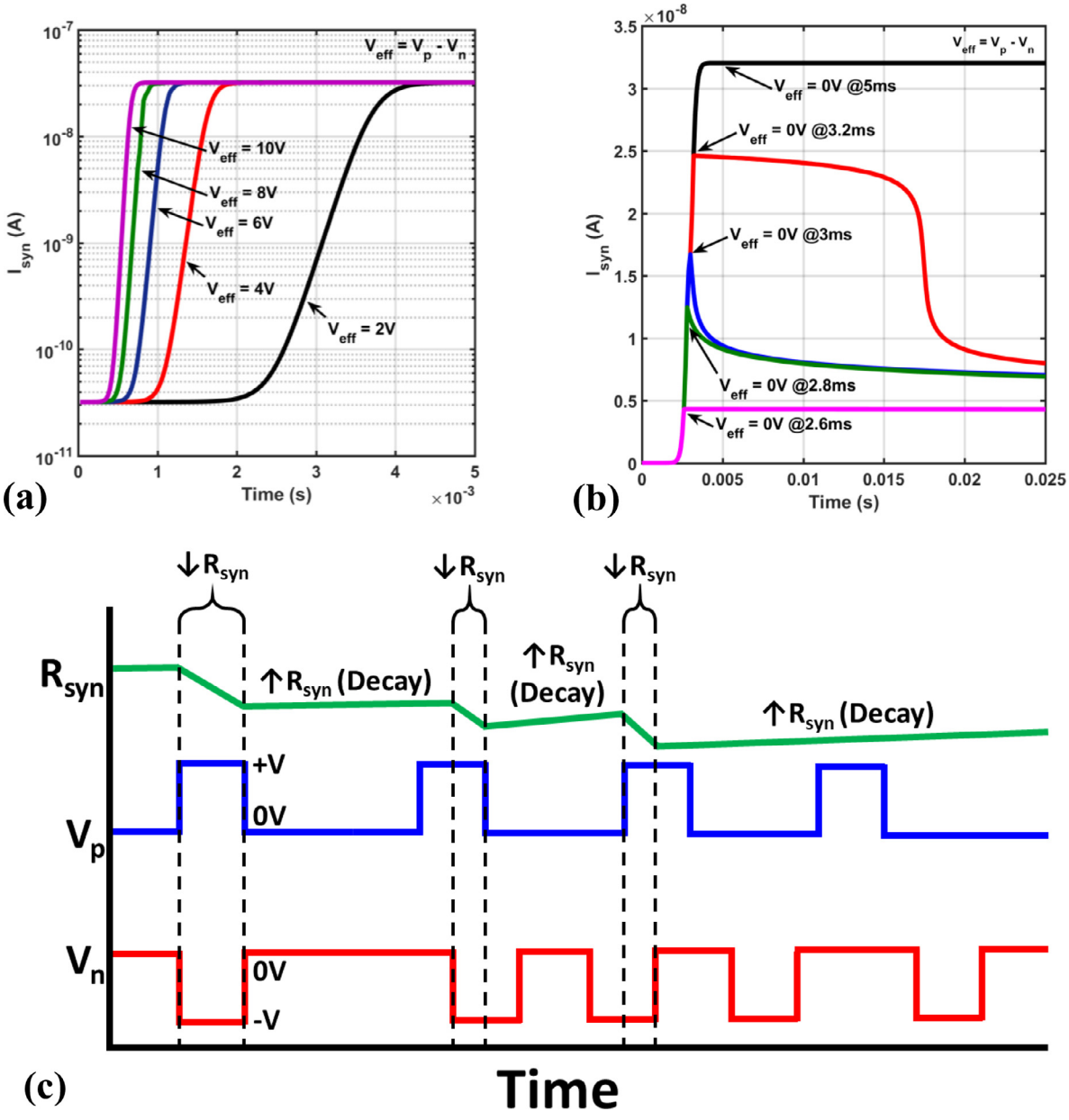


Fig. 2. (a) Output current as a function of time with a constant bias applied to V_p and V_n. The current is in the form of a sigmoid, as the equation for N_c describes. (b) Plot showing how the conductance state of the gated-memristive device affecting the decay of the device's conductance in a Gaussian fashion. (c) Conceptual description of how the device's resistance changes via spike-coincident detection and decay due to diffusion of vacancies. R_{syn} of the device will decrease whenever pulses on V_p and V_n overlap. If no overlap occurs, there is no change in the device's resistance.

the design in which they're used is a critical step in using double gated-memristive devices within a neuromorphic architecture.

The resistance (R_{syn}) of the double gated-memristive device is inversely related to the synapse's weight value (higher resistance is a lower weight, while a lower resistance is a higher weight). When the resistance of the device is changed via application of voltages to V_p and V_n (positive on V_p and negative on V_n as shown in Fig. 2(c)), the change is proportional to the time the overlap occurs and the change in vacancies within the conduction region, ΔN_c,

$$\Delta R_{syn} \sim \Delta N_c * t \tag{8}$$

This change in vacancies is proportional to two key factors depending on the stimuli applied to the device's gates (V_{eff}). If $V_{eff} > V_t$, the change in vacancies is proportional to the difference between V_{eff} and the rate of diffusion decay within the conductive region, defined as d. If $V_{eff} < V_t$, the change in vacancies is only proportional to the decay rate in the conduction region,

$$\Delta N_c \sim \begin{cases} -d, & V_{eff} < V_t \\ V_{eff} - d, & V_{eff} \geq V_t \end{cases} \tag{9}$$

Since d is a Gaussian term within the model, the decay rate is proportional to a negative exponential term where the exponential possesses N_c,

$$d \sim exp(-N_c) \tag{10}$$

On a longer scale of time for the memristive device, the overall change in resistance with respect to time can be seen as proportional to the average frequency of pulses applied to V_p and V_n (defined as f), the average duty cycle of the pulses applied to V_p and V_n (defined as D), the phase match between the pulses applied to V_p and V_n (Defined as φ, where $0 \leq \varphi \leq 1$. A value of 0 corresponds

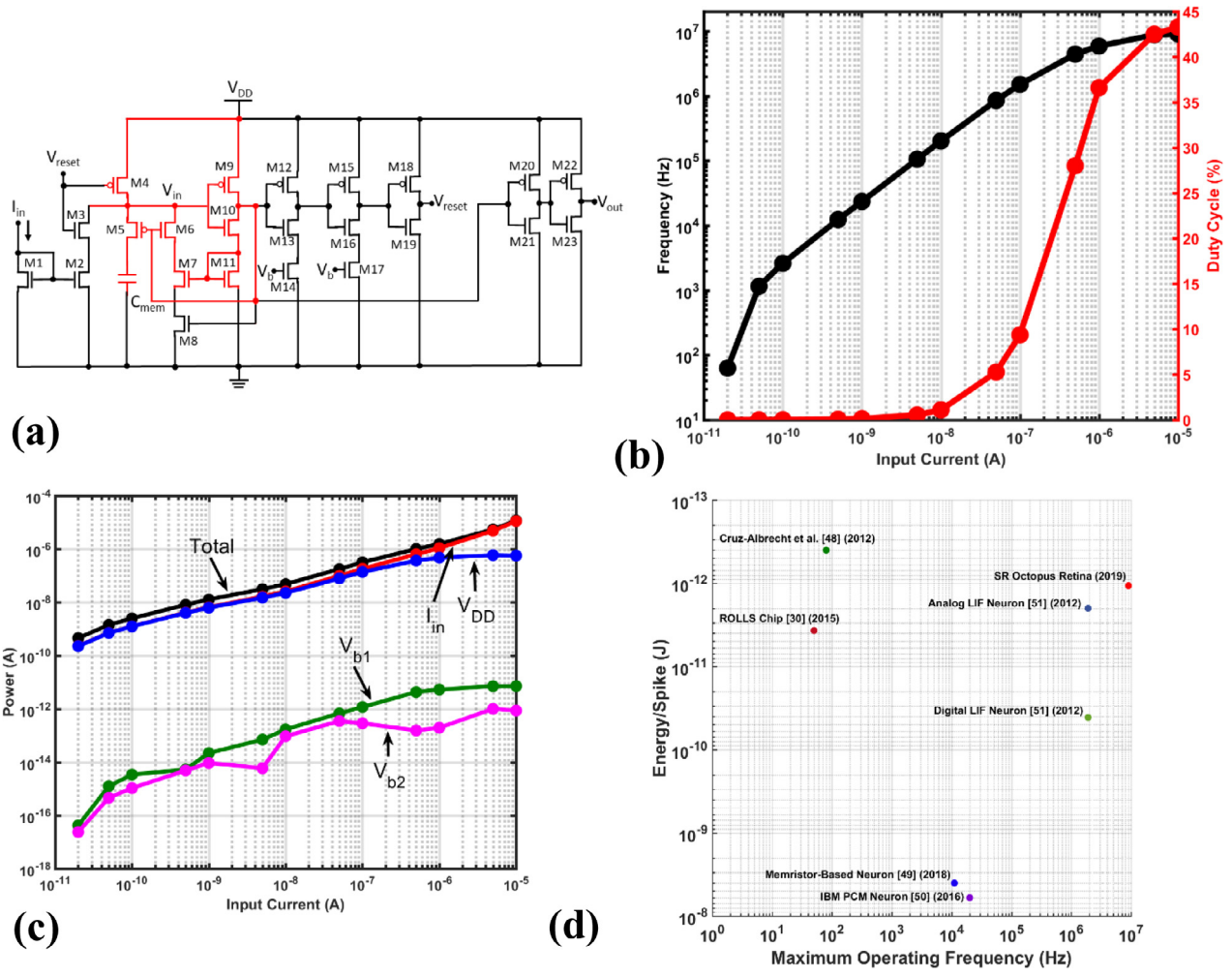


Fig. 3. (a) Schematic of the SR Octopus Retina neuron. Circuit components highlighted in red are components from the original Culurciello et al. circuit. (b) Frequency and Duty Cycle as a function of input curent (I_{in}) for the SR Octopus Retina neuron. Under 20 pA, the frequency of the neuron falls below 1 Hz. (c) Power consumption of voltage sources as a function of input current for the SR Octopus Retina neuron. Total power consumption is shown as well. (d) Comparison of the SR Octopus Retina neuron to previously published neurons and notable CMOS neuron architectures [48-51].

to no phase match and 1 corresponds to a perfect match) minus the decay rate of vacancies from before, d,

$$\frac{dR_{syn}}{dt} \sim fD\varphi - d \tag{11}$$

3. CMOS spiking neuron circuit

This work employs a modified version of the octopus retina-inspired neuron design proposed by Culurciello et al. [47]. The original neuron performs a spike on the falling edge of a simultaneous current spike on its I_{in} node and a voltage spike on its V_{reset} node. This feature limited the circuit to relying upon the externally applied V_{reset} signal to reset its V_{out} node, which meant the circuit was unable to self-reset. This feature from the original circuit was undesirable as each neuron in the architecture proposed in this work is required to be able to be continuously spiking in a dynamic fashion depending on its input current (I_{in}) value. To achieve this functionality, a modified design called the self-resetting (SR) octopus retina neuron has been designed (Fig. 3(a)).

To expand the original neuron's functionality and make it reset without external input, a series of inverters connect the neuron's original output node with the original circuit's V_{reset} node. Additionally, another pair of inverters is used as a buffer on the neuron output node to reduce loading of the reset signal. The voltage on the neuron's capacitor is set to V_{DD} upon application of an active-low reset signal to M4. Then, the input current leaks charge off of C_{mem} via the M1/M2 current mirror, reducing V_{in}. A positive feedback loop is enabled once V_{DD}-V_{in} reaches M9's threshold and the output of the M9/M10 inverter reaches M8's threshold. Reaching this threshold causes the rest of the charge on the input node to quickly be drained, minimizing any short-circuit current in the M9/M10 inverter. This feedback loops also generates a transient voltage spike at V_{out}. Sizings of all transistors in the circuit can be found in the Appendix G. The bias voltages (V_b) are used to control the time constant of the reset operation. A plot showing the circuit's input current vs. frequency/duty cycle profile is shown in Fig. 3(b) (an example output waveform for the circuit can be seen in Appendix J).

One of the primary reasons this neuron was chosen was due to the original neuron's low power consumption, which is an attractive trait for power-conscious applications such as navigation systems. The low power consumption also applies to the SR octopus retina neuron. When power consumption is measured over its operation range, it on average consumes ~1.07pJ/spike. This low energy requirement places it in a competitive spot when compared to other CMOS neurons. The power profile over the neuron's operation range can be seen in Fig. 3(c), and a comparison of the SR octopus retina neuron's operation features compared to other recent neuron circuits can be seen in Fig. 3(d).

4. Architecture concepts

One of the core concepts in neural networks and neuromorphic computing is the capability of doing pattern recognition through associative actions embedded in memories. This concept is called associative learning. When multiple inputs are shown simultaneously to a neural system, an association between those pieces of info is formed. After the association is formed, observing just one input from the set enables the neural system to recall all correlated inputs from the set. Using the previously defined synaptic devices and the SR Octopus Retina Neuron circuit, this concept of associative learning can be realized within hardware for a neuromorphic circuit.

One of the most common associative memory neural networks currently in use is the attractor network [52–54]. Attractor networks are recurrently connected sets of neurons through a matrix of synapses. A "memory" is usually formed within the network by giving direct input stimuli to two or more neurons within the attractor network. Forcing two or more neurons within the network to fire simultaneously will cause the synaptic connections that bind those neurons together to increase in conductance [53]. Most common single attractor networks use the concepts of both active excitation and inhibition to form the attractors (memories) within the network.

It is desired for the architecture in this work to store multiple memories. With the setup shown in Fig. 4(a), the network will be able to possess multiple memories in associated synaptic connections simultaneously. If inhibition was used on the synapses to create memories in the multiple memory design proposed here, old memories would be erased each time a new memory would be formed. Due to this issue, only active excitation can be relied upon to form and maintain multiple memories. The only way for them to be forgotten in this setup is for the transient decay of the synaptic devices to eventually place them back into a low conductance state if the synapses aren't potentiated for some time. If the synaptic devices are eventually potentiated to near their minimum resistance value however, the memories become effectively non-volatile. This is a design choice specific to the architecture described here and is not a general feature of attractor networks. The only way to remove the memory at that point is to apply a long, negative bias to the device to depress it to a higher resistance value. If saturation of weights in such a network becomes an issue, an occasional global depression signal could be sent to the entire network to keep it from saturating.

To place two memories into the network that have no common features/neurons between them, the stimuli for the two memories should be shown one after another. For example, the first two neurons in the network could first be fed external stimuli, followed by the third and fourth neurons being given stimuli. This training process will cause the first two neurons to be associated with one another and likewise for the third and fourth neurons. This association is created by taking the output signals from each neuron and running them through a pair of positive and negative voltage amplifiers. These buffered neuron signals are connected to the gate terminals of the synaptic devices as described in Fig. 4(a). The negative amplifiers have their signals routed to the V_n gates of each synapse while the positive amplifiers have their signals routed to the V_p gates of each synapse. For every synaptic device, it will potentiate when both neurons connected to its positive and negative gates are firing simultaneously. If only one of the two neurons is firing, the synapse will not potentiate, as the voltage applied across the gate terminals of the synaptic device is designed to not be high enough to cross the internal threshold voltage of the synaptic device in order to cause potentiation (V_t).

Once the two memories are placed into the attractor network, they can be recalled at any point by simply providing external stimuli to one of the network's neurons. If the first neuron is stimulated after training, the network can recall that the first and second neurons are associated by causing the second neuron to fire. This concept of recall also applies to the third and fourth neurons if only the third neuron is given stimuli. This entire process of training the attractor network and recalling memories as described in this section can be seen in Fig. 4(b) and (c).

In the results shown in Fig. 4(b) and (c), it can be seen that creating the memories within the network takes time (~2ms each). Despite the longer potentiation time, this feature adds robustness to the network. Since each synaptic device gradually decreases its resistance with each spike coincidence event it receives, this allows the network to be robust against spurious stimuli/noise. For example, if N1 and N2 are being shown deliberate stimuli to form an association with one another, but then N3 emits a small number of spikes from itself due to charge build-up on its input node over time, this will not affect the state of the system. It requires many spike-coincident events between two or more neurons to form an association. If a low frequency of spikes occurs over a long period, the self-decay of the synaptic devices passively depresses any incorrect memories forming in the network.

5. SLAM architecture

The concept of forming associative memories within attractor networks can be utilized to solve more complex learning problems such as SLAM. In SLAM, the navigational system must be able to create a map of its local environment during initial exploration, localize itself within that map when revisiting the environment, and be able to make localization-based decisions. The attractor network is a system that can be used to create a collection of memories that correspond to key landmarks in the system's environment to simultaneously map key environment features and localize itself within the landmark map (similar to the place and grid cells in mammals previously discussed). This architecture can be elaborated into a larger architecture that can achieve the objectives of SLAM.

The proposed architecture is designed to navigate a three-dimensional, cylindrical environment. The initial test system controlled by the architecture in this work can move up and down within its environment and rotate left and right. The system can track six unique head directions (0°, 60°, 120°, 180°, 240°, 300°) and four different z-axis altitudes (Z1, Z2, Z3, Z4). The system is also assumed to be able to identify key landmarks in its environment that it can place in its internal map. This identification process is handled by an external algorithm capable of landmark/image recognition. For the architecture proposed here, the system will be able to distinguish between four different colored landmarks: red, orange, green, and blue.

Of the landmarks the system can identify, one of them is defined as its "target." This landmark is predefined for the system to be the navigational goal within its environment. In this work, the target is predefined as the blue landmark. Once the target is seen, the system can navigate from any previously seen landmark to its target in the shortest path. A diagram of this environment can be seen in Fig. 5(a).

The neuromorphic SLAM architecture can be split into three main sections: the input layer, the associative layer, and the motor layer. The overall system layout can be seen in Fig. 5(b). The input layer will take external head direction, altitude, and recognized landmark data as input and output a spike pattern from each of its modules to the associative layer. The Z-axis/head direction neuromorphic module within this layer consists of a decoding circuit that runs to an array of SR Octopus Retina neurons that each correspond to one of the positions/head directions the system is capable of tracking (the stimuli fed to these neurons puts them into

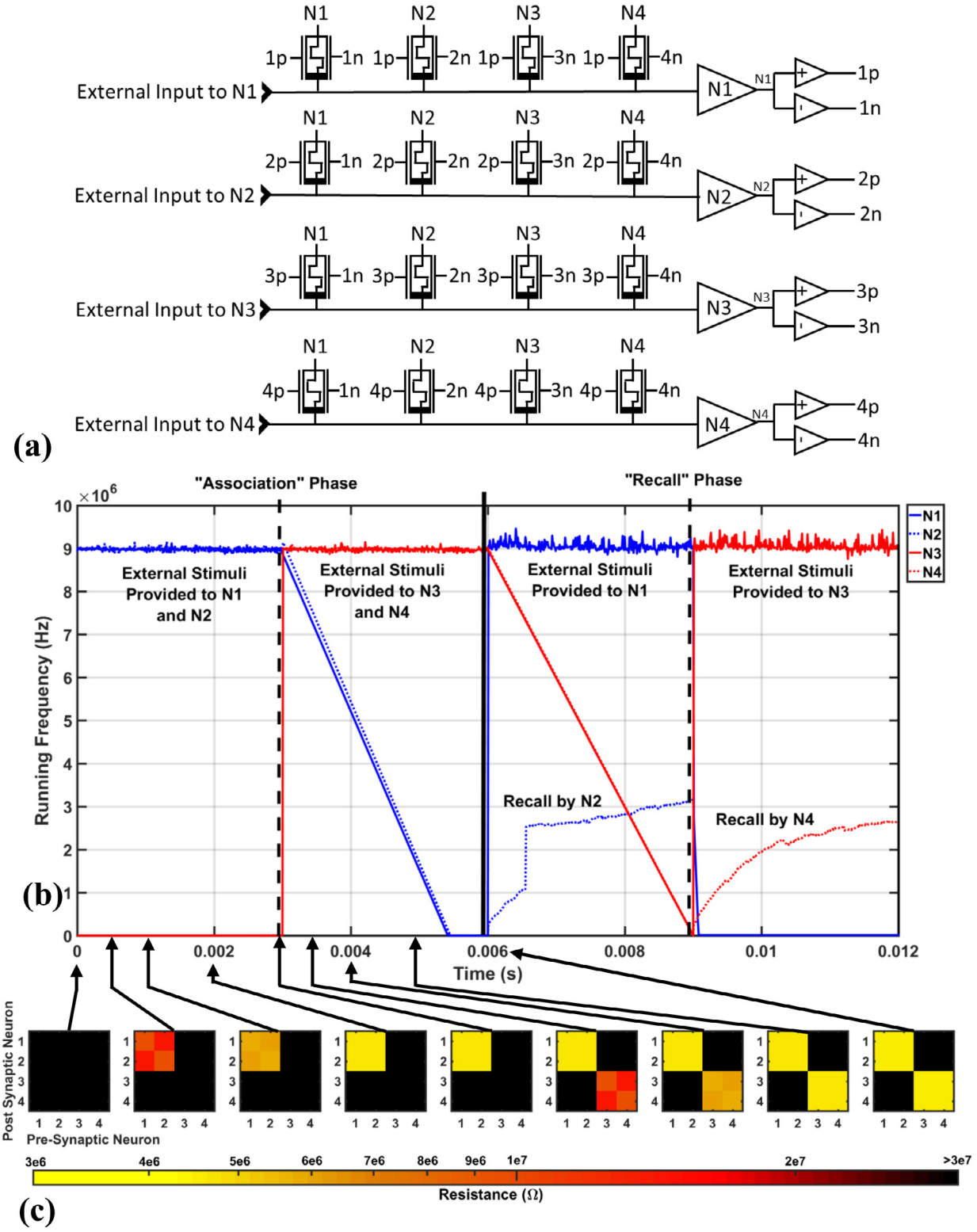


Fig. 4. (a) Schematic of the proposed four-neuron attractor network that uses the double gated-memristive devices for synapses. (b) Running frequency of the four neurons from the attractor network with respect to time during a two memory training/recall test. (c) Snapshots of the synaptic heat map for the four-neuron attractor network evolving over time as the two memories are programmed into the network. The typical off state resistance shown in these snapshots (synaptic cells shown in black) is ~3.3 GΩ.

their maximum frequency state). This allows the positioning/head direction data to be encoded into the system as a spike train to be sent to the associative layer. It's important to note that only one head direction neuron, one z-axis neuron, and one landmark input is ever firing at any given time (as defined by the setup of the decoders within the input module and the assumed incoming landmark data). Landmark data (e.g. the colored landmarks) is assumed to be coming as a spike train from an external system doing image recognition. The associative layer then takes the data given by the input layer and associates the inputs with one another

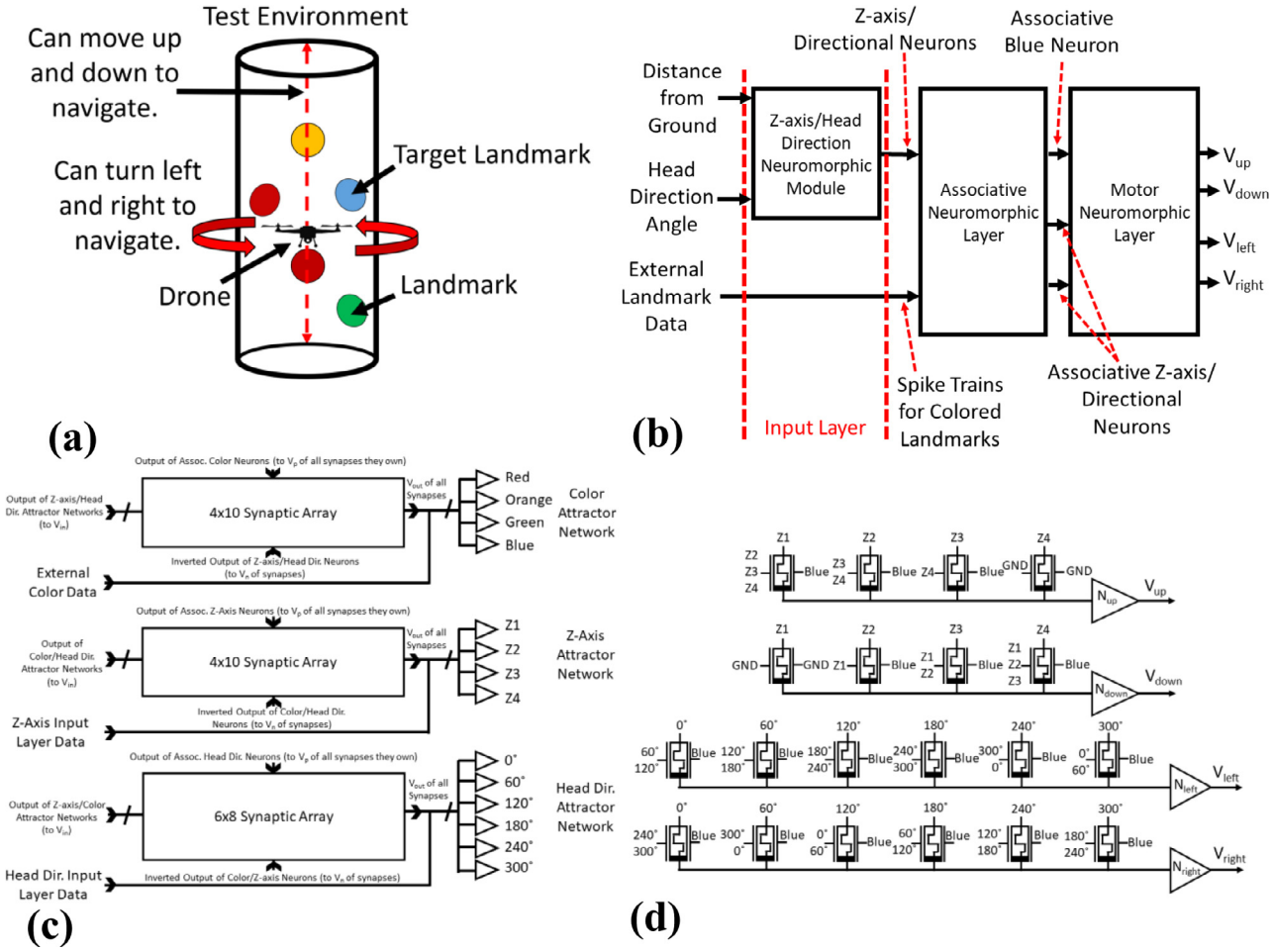


Fig. 5. (a) Diagram showing the test environment for the SLAM architecture to explore. (b) System-level diagram for the neuromorphic architecture to approach SLAM. The system is split into an input layer, associative layer, and motor layer. (c) Diagram of the layout of the associative layer and its three interconnected attractor networks to create the landmark map of the system's local environment. (d) Diagram describing the layout of the neuromorphic architecture's motor layer. All V_p terminals with multiple signals sent to them are preceded by two or three input OR gates to unify the signals. All signals attached to V_p are positively amplified prior to application, and all signals attached to V_n are negatively amplified prior to application.

using a set of attractor networks. Finally, the motor layer acts as a programmable attractor network which receives its programming signals from the associative layer (instead of itself) when a target is observed. The motor layer then takes all non-target data from the associative layer to control the system's motion.

The associative layer uses a set of attractor networks discussed in the previous section to remember the locations of landmarks within its environment. The associative layer is comprised of three interconnected attractor networks. A separate attractor network exists for each of the three parameters the system is tracking (landmark being observed, z-axis position, and head direction). The attractor networks are interconnected as described in Fig. 5(c). This setup allows the associative layer to associate each landmark to a z-axis and head direction, thereby allowing the system to localize itself when it sees this landmark later.

The motor layer contains four neurons to create the layer's output (Fig. 5(d)). These neurons correspond to the drone's four movements: move up, down, left, and right. Target (blue) outputs from the associative layer are connected to the negative gates of the synaptic devices (V_n) in the motor layer. Z-axis and head direction neurons from the associative layer have their outputs connected to one synapse per neuron per signal from the associative layer. The positively amplified versions of those signals are also sent to the positive gates (V_p) of the synapses as shown in Fig. 5(d). To send signals to the positive gates of the synaptic devices for synapses with two or more input signals to the gate, an OR gate is used to connect the signals from the associative layer to the V_p node of the synapse.

An experimental setup for a test environment is shown in Fig. 6(a). Different colored landmarks are placed throughout the environment. In simulation, the system explores its environment until its target is observed. When placed into its previously explored environment, it can identify the shortest path to its target when only shown the color of a previously seen landmark (5ms-6ms in Fig. 6(b)). Heat maps for the associative and motor layers' synapses are in Fig. 6(c) and (d), respectively.

The architecture was simulated in two halves to obtain the results shown in Fig. 6. The first half involved simulating the input layer and the associative layers of the architecture. Output patterns of the associative layer were then recorded and replicated for use in the second half of the simulation. Figs. 6(c) and (d) were obtained from the first simulation. The second half of the simulation then took inputs from a small array of SR Octopus Retina neurons designed to replicate the output of the associative layer from the first simulation and places them into the motor layer.

One final thought for an architecture focused on a task such as navigation is power. Most navigational systems must rely on very limited power. In the simulation conducted in Fig. 6, the system consumed an average of 36μW of power during operation. The small array of neurons designed to replicate the behavior of the

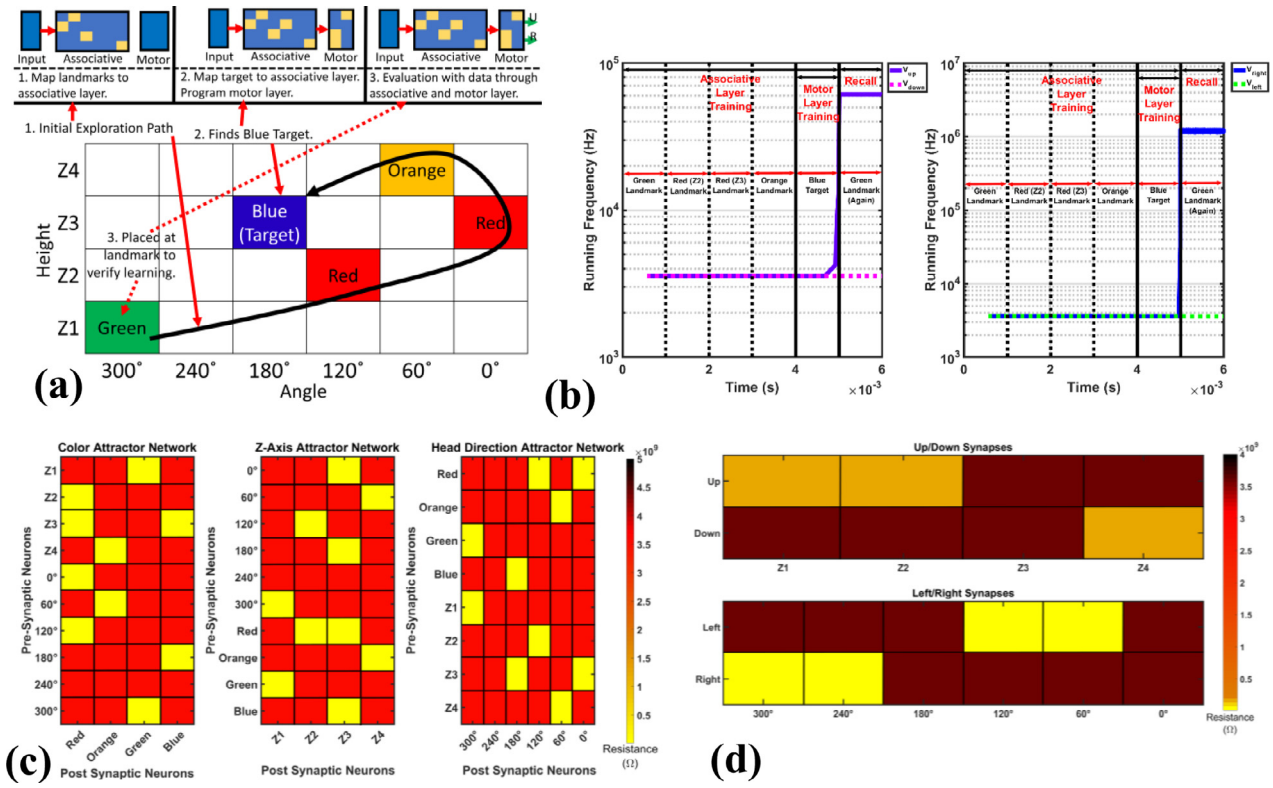


Fig. 6. (a) Diagram showing the test pattern that will be given to the SLAM architecture for analysis. (b) Running Frequency plot showing how the system explores the environment as shown in Fig. 20. After the system explores its environment, it revisits a previously seen landmark to verify it knows the shortest path to its target. (c) Synaptic heat map for the associative layer in the SLAM architecture after the system fully explored its environment and found its target. (d) Synaptic heat map for the motor layer of the SLAM architecture after the system found its target. The cause of the up/down synapses being not set as far as the left/right synapses is due to the Z3 signal from the associative layer was more out of phase with the blue signal from the associative layer than the 180° signal.

associative layer in the second simulation was not factored into the measured power consumption of the circuit. For a simple comparison, the STA8089FGA GPS module from STMicroelectronics consumes an average of 25-90mW during normal operation [5]. The comparison isn't exactly on par in terms of functionality (as the architecture shown here hasn't been scaled up to something that would be in an implemented system), but it shows the promise a system such as the architecture shown in this paper has for tackling navigational tasks such as SLAM.

6. Future design direction

The architecture proposed in this work is a proof of concept. The architecture does not include more advanced features such as moving in the X-Y plane, environments with operational ceilings that vary in height, etc. These types of features are planned to be further implemented in a future work focusing more on the system-level functionality and abstraction of the SLAM architecture. Evaluating the architecture at a system level is critical to understanding how the network would respond in a closer-to-reality environment. One of the limitations in the area of SLAM is a lack of standard training datasets. Many papers have their own methods of showing how they can evaluate a SLAM problem. Some implementations give test environments [24,55,56] while others will take images of a real world location such as an office or bedroom in an attempt for the system to identify its location within that environment [19,57]. Other test environment data also isn't normally set up for direct compatibility with hardware-based simulators such as SPICE. Observing how the network behaves in a long term versus short term operation time span is planned to see how the network navigates to landmarks more efficiently as times passes. This functionality can be tested in a real world or simulated environment.

In addition to the extra functionality, fabrication of the devices that create the architecture such as the CMOS neurons is planned for verification of a physically realizable architecture. This testing process will include testing on the fabricated CMOS neurons and gated-memristive devices along with simulations to test the architecture's robustness to process variation that will naturally exist in a physically realized network.

7. Conclusion

This work has demonstrated a neuromorphic architecture using novel device models capable of localization within space and solving for the shortest path to a target within a cylindrical environment. To our knowledge, this is the first neuromorphic architecture using gated-memristive devices to tackle the navigational task of SLAM. The architecture uses an adapted SR octopus retina neuron that outperforms other existing neuron circuits by consuming only 1.07pJ per spike and having a maximum output frequency of ~9 MHz. The neurons are used in tandem with gated-memristive devices that act as synaptic memory for the system to remember the locations of landmarks within the navigational system's environment. The system can recognize the location of targets within its environment and use its localization state to navigate to a previously seen target while only consuming an average of 36μW of power during operation.

Funding sources

This work is supported by the National Science Foundation under award numbers: ECCS 1156294 and SHF-1718428.

Declaration of Competing Interest

The authors of this work declare no competing financial interests.

Acknowledgments

The authors would like to thank Thomas Schultz for his assistance in ensuring proper layout scalability of the double gated-memristive devices.

The collaborative research activity documented in this paper between the Air Force Research Laboratory and the University of Cincinnati along with the Rochester Institute of Technology was facilitated through Educational Partnership Agreements, 18-RI-EP-08 and 15-RI-EP-09, respectively. The viewpoints represented in this paper are those of the authors' and not the U. S. Government, the Department of Defense or its agencies, or the Air Force Research Laboratory. This paper has been cleared for public release (Case Number: 88ABW-2019-1937).

Data availability

The data that supports this work is available from the corresponding author upon reasonable request.

Competing interests

The authors of this work declare no competing financial interests.

Appendix A – Simulation environment

All simulations were conducted in HSPICE N-2017.12 on a Windows 10 operating system. All MOSFETs used utilized Arizona State's Predictive Technology Models [58] 180nm bulk technology file (180nm_bulk.pm). All circuit components used excluding the synaptic devices were constructed in HSPICE spice netlist files (file extension *.sp). The models for synaptic devices were written in Verilog-a (file extension *.va) and then had the models instantiated as circuit components within the HSPICE spice netlist files. Any output data needed to be read was read with the Synopsys tool Custom WaveView Version O-2018.09, which is a standard waveform viewer tool for HSPICE simulations.

One important thing to note within HSPICE is when simulating oscillating circuits such as neurons during transient analysis, ensure that the simulator is using initial conditions. This is done by appending the *"uic"* token to the end of any transient analysis command as shown in the following example.

.tran 1ns 1ms uic

Without the added token, the HSPICE simulator won't interpret the neuron circuit as oscillating if the neuron circuit is being given a DC input voltage or current as its external stimuli. Therefore, it will appear as if the neuron isn't firing at all.

Appendix B – $SrTiO_3$-memristive device theoretical model

The Verilog-A model for the $SrTiO_3$ gated-memristive device is based off a parallel current calculation through the switching oxide channel within the device. Although this model uses similar device parameters as the device used in [34], the model differs from the one in [34] as it more closely follows device physics instead of a behavioral model. Having this physics-based model of the previously designed gated synaptic device is key to show the step by step process in improving the device for future use. For a step by step process to the model described below, see the flowchart that describes the model in Fig D1. To calculate the current through the gated-memristive device, the equation

$$I_{syn} = \left(J_{ohm} + J_{fp}\right) w_{max}^2 \tag{B.1}$$

is used where J_{ohm} is the ohmic current density through the switching oxide, J_{fp} is the Frenkel-Poole current density through the switching oxide, and w_{max} is the maximum width of the conductive path that can form within the oxide. Both current density terms can be calculated using Eqs. (2) and (3), which are taken from Chiu [44].

$$J_{ohm} = \frac{q\mu N_c V_{in}}{L_{ch}} exp\left(\frac{-\left(E_c - E_f\right)}{kT}\right) \tag{B.2}$$

$$J_{fp} = \frac{q\mu N_c V_{in}}{L_{ch}} exp\left(-\frac{q}{kT}\left(\varphi_t - \sqrt{\frac{qV_{in}}{L_{ch}\varepsilon_0\varepsilon_i\pi}}\right)\right) \tag{B.3}$$

In the current density calculations, E_f is the Fermi level within the oxide, and N_c is the density of states value for the oxide channel that describes the amount of oxygen vacancies along the conductive path from the device's top electrode to its bottom electrode. All other parameters are described in Table C1 in Appendix C (as are all other subsequently undescribed parameters for the $SrTiO_3$ gated-memristive model).

To obtain the values for the Fermi level and N_c, the following pair of adapted equations from Chiu [44] are used.

$$N_c = \frac{w_c}{L_{ch}} N_{c,max} + \left(1 - \frac{w_c}{L_{ch}}\right) N_{c,min} \tag{B.4}$$

$$E_f = \frac{E_c}{2} + \frac{N_c E_c}{2N_{c,max}} \tag{B.5}$$

To calculate V_{eff}, which is the effective voltage applied to the gate of the device, the equation [34]

$$V_{eff} = V_{gate} - \frac{V_{in} + V_{out}}{2} \tag{B6}$$

is used. The w_c and d terms are calculated in the exact same manner as the model for the Nb_2O_5 device.

Appendix C – Gated-memristive model ($SrTiO_3$) parameters

Below is a list of all parameters and constants needed to model the gated-memristive device in addition to a description and the variable's value. Derivations for any values that require derivation are shown following the list.

To obtain the value of M_n, M_n was defined as

$$M_n = \frac{N_{c,min} + N_{c,max}}{2} = \frac{1e22 + 1e25}{2} = 5e24$$

where M_n was then used to calculate σ_n

$$\sigma_n = \frac{M_n}{6} = \frac{5e24}{6} = 8.33e23.$$

It should be noted that the equation for d in this model acted as a fitting parameter for ensuring the hysteresis matched the measured $SrTiO_3$ gated-memristive device. Therefore, all parameters within the equation for d should be treated as fitting parameters.

Table C1
Parameters used in simulation of $SrTiO_3$ gated-memristive device.

Parameter	Description	Value
q	Charge of an electron	1.6e-19 C
μ	Electron mobility of $SrTiO_3$	0.001 $m^2/(V^*s)$ [39]
L_{ch}	Length of oxide channel in gated-memristive device	1e-8 m
φ_t	Trap height in oxide	1.35 [40]
ε_0	Vacuum permittivity	8.85e-12 F/m
ε_i	Dielectric constant of $SrTiO_3$	130 [41]
k	Boltzmann constant	8.62e-5 eV/K
T	Temperature	300 K
m_0	Mass of an electron	9.109e-31 kg
E_c	Bandgap of $SrTiO_3$	3.3 eV [42]
w_{max}	Maximum width of filament in SOx	1e-9 m
$N_{c,max}$	Defined maximum density of states in SOx	1e25
$N_{c,min}$	Defined minimum density of states in SOx	1e22
t_{step}	Time step in drift calculation	1e-6 s
μ_{o2vac}	Mobility of oxygen in vacuum	4e-17 $m^2/(V^*s)$
W_{ch}	Width of SOx	2e-8 m
d_{max}	Fitting parameter to control decay	1.5e10
σ_n	Standard deviation of N_c	8.33e23
M_n	Mean value of N_c	5e24

Appendix D – Gated-memristive model ($SrTiO_3$)

To model the gated-memristive device within Verilog-a, the equations shown in the discussion section are executed in simulation as shown in Fig. D1. At the very start of the simulation, every gated-memristive device has its parameters put into their initial state. At every time step, each gated-memristive device executes one pass through the Fig. D1 (ending with the calculation of I_{syn}).

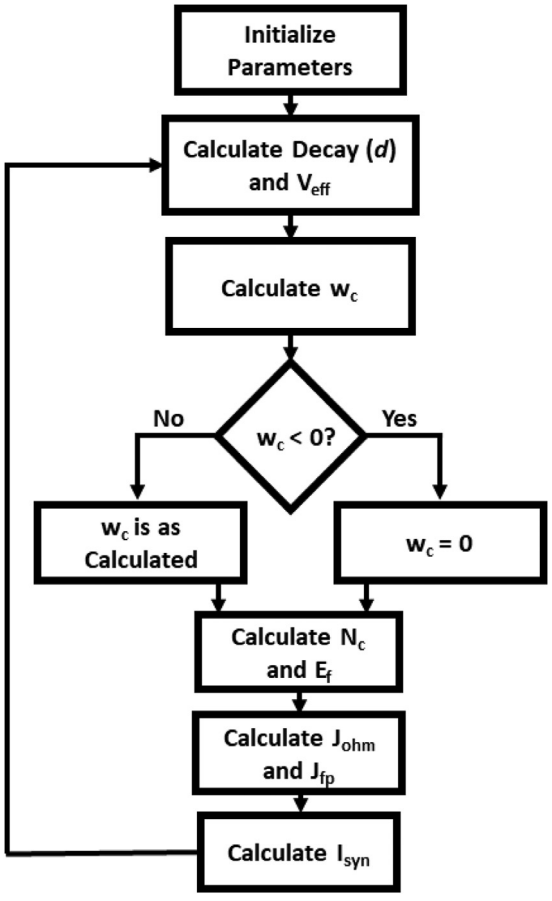


Fig. D1. Flow of model for gated-memristive device implemented in Verilog-a.

Table E1
Parameters used in simulation of the Nb_2O_5 double gated-memristive device.

Parameter	Description	Value
q	Charge of an electron	1.6e-19 C
μ	Electron mobility of Nb_2O_5	2e-5 $m^2/(V^*s)$ [59]
L_{ch}	Length of oxide channel in gated-memristive device	2e-8 m
φ_t	Trap height in oxide	0.62 eV [60]
ε_0	Vacuum permittivity	8.85e-12 F/m
ε_i	Dielectric constant of Nb_2O_5	28 [59]
k	Boltzmann constant	8.62e-5 eV/K
T	Temperature	300 K
m*	Relative mass of Nb_2O_5	$4m_0$ [59]
m_0	Mass of an electron	9.109e-31 kg
w_{max}	Maximum width of filament in SOx	1e-9 m
$N_{c,max}$	Defined maximum density of states in SOx	1e27
$N_{c,min}$	Defined minimum density of states in SOx	1e24
t_{step}	Time step in drift calculation	1e-6 s
μ_{o2vac}	Mobility of oxygen in vacuum	4e-17 $m^2/(V^*s)$
W_{ch}	Width of SOx	1e-8 m
d_{max}	Fitting parameter to control decay	1.6e8
σ_n	Standard deviation of N_c	8.33e25
M_n	Mean value of N_c	5e26
V_t	Threshold Voltage for Potentiation in SOx	6.5V*

* The value of V_t was adjusted to a higher value to allow for faster acceleration time above the usual value of 1.8V [45].

Appendix E – Double gated-memristive model (Nb_2O_5) – Parameters

Below is a list of all constants and parameters used in the model for the double gated-memristive device in addition to a description and its value. Derivations for any values that require derivation are shown following the list.

The values for M_n and σ_n for this model were obtained in the same manner as they were in the $SrTiO_3$ model. It should also be noted that some of the simulations shown in the main paper exaggerated the value of d_{max} to 1.4e13 to easily show the pattern of decay in the device when not being potentiated in a viewable amount of time.

Appendix F – Double gated-memristive model (Nb_2O_5)

The model for the double gated-memristive model is ran in an almost exact manner as the original model. The only addition to the model is the extra step in checking the value of V_{eff}. If that value doesn't exceed the value of V_t, V_{eff} is made zero for the w_c calculation so no potentiation is caused within the oxide channel. The chart for the double gated-memristive model can be seen in Fig. F1.

Appendix G – SR octopus retina neuron transistor sizings

To obtain proper behavior from the neuron circuit used, transistors must be of appropriate size. Any resizings beyond the minimum feature size are done in multiples of the minimum features size (e.g. 360 nm, 540 nm, 720 nm, etc.). Table G1 shows a full list of dimensions used for every transistor in the neuron circuit. In addition to proper trasnsistor sizings, C_{mem} within the circuit was defined as 120fF. All transistors within the circuit possess a minimum feature of size of 180 nm.

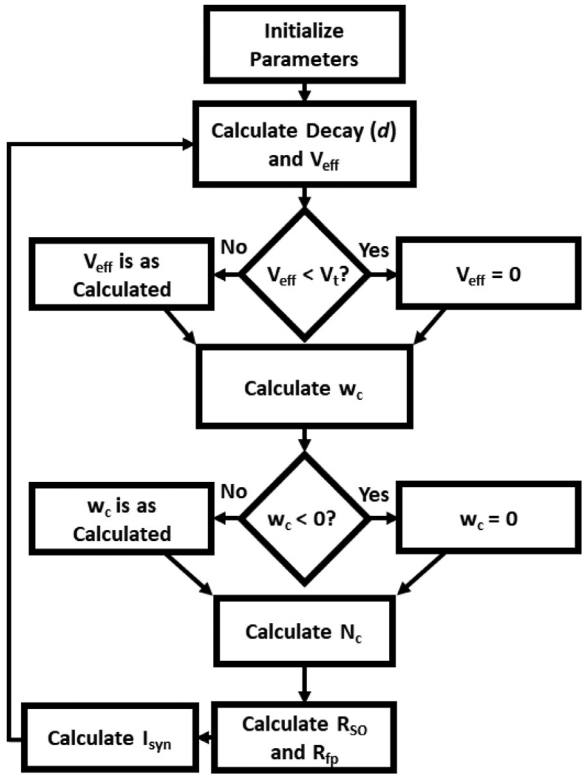


Fig. F1. Flow of model for double gated-memristive device implemented in Verilog-a.

Table G1
Sizings of all transistors used in the implementation of the SR octopus retina neuron.

Transistor	Length (nm)	Width (nm)
M1	720	180
M2	720	180
M3	180	180
M4	180	180
M5	180	180
M6	180	180
M7	180	180
M8	180	180
M9	720	180
M10	720	180
M11	180	180
M12	720	180
M13	720	180
M14	720	180
M15	720	180
M16	720	180
M17	720	180
M18	720	180
M19	720	180
M20	180	180
M21	180	180
M22	180	180
M23	180	180

Appendix H - Comparing the $SrTiO_3$ model to measured data

In Herrmann et al., a test was performed to show the hysteretic behavior within the $SrTiO_3$ gated-memristive device [34]. This test swept a voltage applied to V_{gate} from -5V to 5V and back over the course of 60 seconds. This test showed that the gated-memristive device could be used as a synaptic device, as it confirmed it was indeed resistive RAM. A similar test can be performed on the gated-memristive model discussed in this work to show its similarity to the original device. The same exact test was performed on the gated-memristive model, and the comparison can be seen in Fig. H1. The hysteresis curve between the measured and modeled device are close in behavior. The shifting of the hysteresis curve between the two curves is noticeable but could be attributed to device fabrication variance for the measured device.

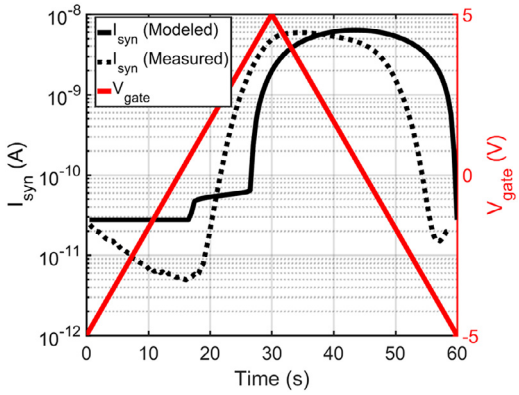


Fig. H1. Output current (I_{syn}) as a function of gate bias (V_{gate}) and time of a $SrTiO_3$ gated-memristive device from Herrmann et al. [21].

Appendix I – Sweeping the Nb_2O_5 model

Taking the adapted double gated-memristive model and performing a similar test on it to the test performed in Herrmann et al. does indeed show hysteresis in the model. This result can

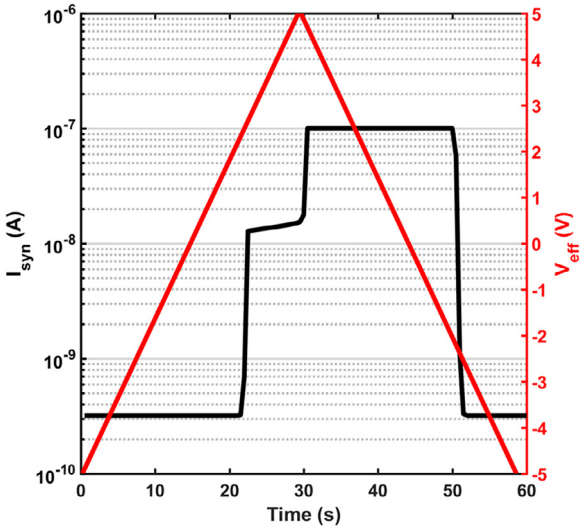


Fig. I1. Output current (I_{syn}) as a function of gate bias (V_{eff}) and time of a Nb_2O_5 double gated-memristive device proposed in this work.

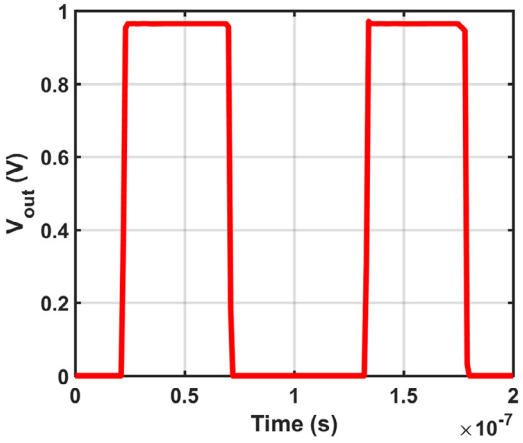


Fig. J1. Example output waveform of the SR Octopus Retina neuron.

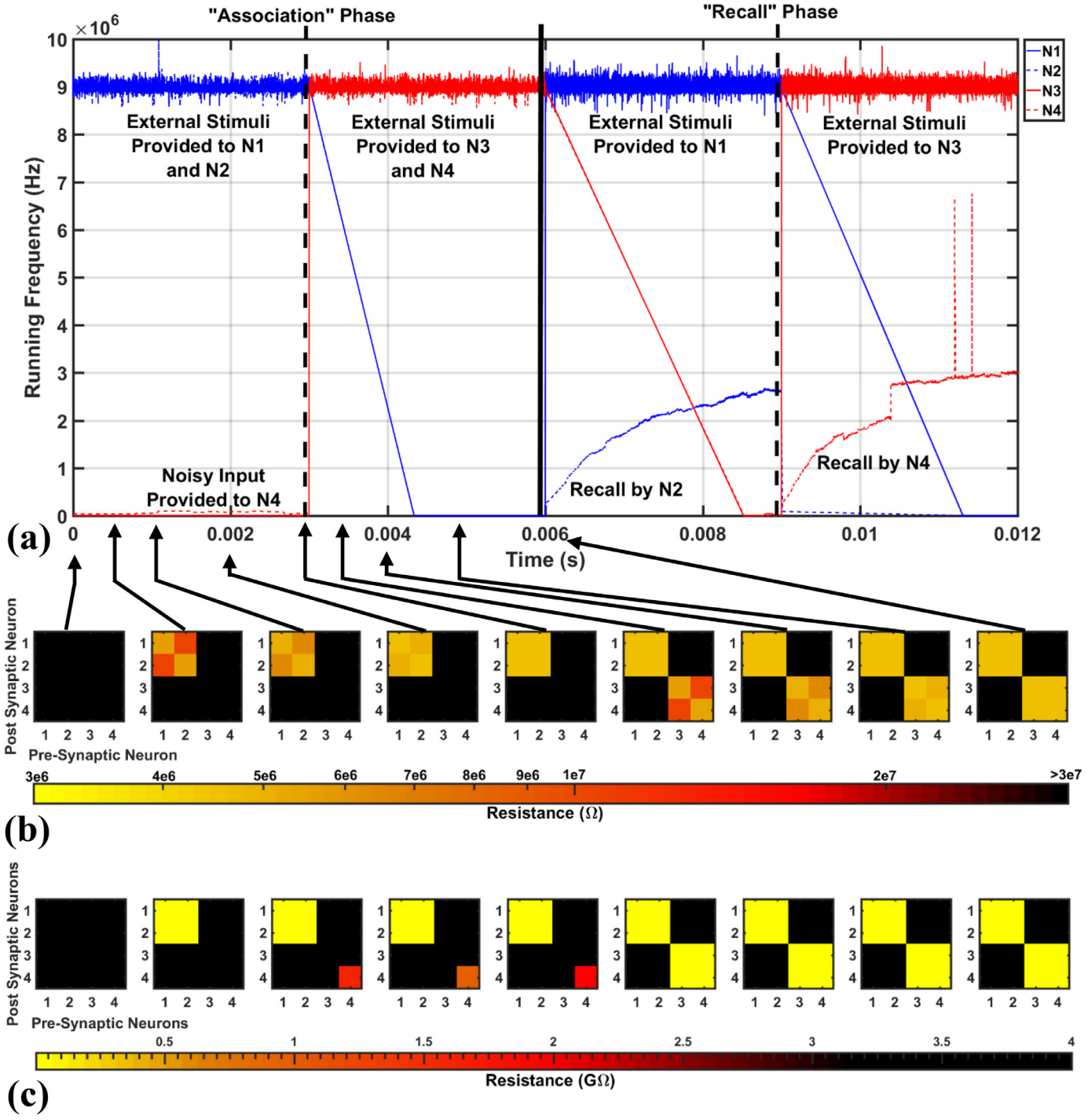


Fig. L1. (a) Running frequency of the four neurons from the attractor network with respect to time during a two-memory training/recall test when 100 pA is provided to N4 to simulate noise given to the attractor network during the first 3ms of the simulation. (b) Snapshots of the synaptic heat map for the four-neuron attractor network evolving over time as the two memories are programmed into the network in log scale. (c) Snapshots of the synaptic heat map for the four-neuron attractor network evolving over time as the two memories are programmed into the network in linear scale. The noise given to N4 can be seen to begin to potentiate the synapse wired to itself, but then watch it depress from 2ms to 3ms.

be seen in Fig. I1. The curve does not look the same in comparison to the original model, however that is expected. The internal structure and functionality of the device has changed from one model to the next. The device now spends must less time transitioning between its lowest and highest conductance states.

Appendix J – Example output of SR octopus retina neuron

If the SR octopus retina neuron is given external stimuli such as a DC current on its input node, the neuron will begin to spike. The neuron's output waveform is in the form of a square wave instead of the abrupt spike seen in other CMOS circuits that have been designed (including the original version of this neuron design by Culurciello et al.) [47]. The output waveform this neuron would be that original spike waveform if it were not for the final buffer before the output node of the circuit. This buffer is needed in this design to separate the implemented self-reset mechanism part of the circuit with any external load applied the output node of the neuron. Without this buffer, any external load would eliminate the self-reset mechanism, and prevent the neuron from

oscillating. Both V_b terminals were set to 0.4V for all simulations in this work. A sample square waveform output from the neuron circuit can be seen in Fig. J1.

Appendix K – Matching the synaptic devices and neuron circuits

Before performing associative learning in hardware, the neuron and synaptic devices must be paired properly to work with one another. This pairing process primarily involves matching the operation range of the neuron being used to the output current range of the synaptic device being used to ensure the operation ranges overlap.

For example, if the desire was to create a very simple associative learning architecture that consisted of four neurons that were connected to form a four-neuron attractor network (Fig. 4(a)), the neuron would have to be properly paired to a set of four synaptic devices. In this case, this pairing means that the moderate to maximum conductance value of a single synaptic device is enough to excite its post-synaptic neuron. It also means that the output current of an array of synaptic devices when at a low conductance value is not sufficient to place the neuron above a firing rate that would not cause any measurable change in the network's state.

Appendix L – Behavior of attractor network when given noise input

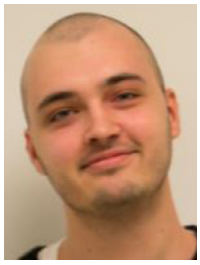
When the four-neuron attractor network is given input that is a small non-zero value (100 pA in this case) to neurons not intended to fire, it will cause the neuron(s) given the noisy input to output spurious sequences of spikes. By design, the attractor network is built to handle such unintended noise due to the robust nature of the double gated-memristive devices. When the experiment from Fig. 4 is repeated with 100 pA provided to N4 while N1 and N2 are being associated with one another, the spurious spikes generated by N4 are not enough to cause any real association with N1 and N2. This behavior can be seen in the following figure where the network has a weak response to the spikes from N4, but they begin to decay over time. The memory recall is still successful for both sets of associated neurons despite the noise interference during the first memory association.

References

[1] B. Kim, M. Jung, Y. Chang, Proceedings of the International Conference on Embedded Systems, Cyber-Physical Systems, and Applications (Aug. 2018) 37–40.
[2] P.M. Kornatowski, A. Bhaskaran, G.M. Heitz, D. Floreano, Last-centimeter personal drone delivery: field deployment and user interaction, IEEE Robot. Autom. Lett. 3 (4) (Oct. 2018) 3813–3820, doi:10.1109/LRA.2018.2856282.
[3] M. Gowda, J. Manweiler, A. Dhekne, R.R. Choudhury, J.D. Weisz, Tracking drone orientation with multiple GPS receivers, in: Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking, ACM, 2016, pp. 280–293, doi:10.1145/2973750.2973768.
[4] L.A. Tawalbeh, A. Basalamah, R. Mehmood, H. Tawalbeh, Greener and smarter phones for future cities: characterizing the impact of GPS signal strength on power consumption, IEEE Access 4 (2016) 858–868, doi:10.1109/ACCESS.2016.2532745.
[5] STMicroelectronics, STA8089FGA GPS Receiver, [Online], https://www.st.com/resource/en/datasheet/sta8089fga.pdf. Last Access: 3/11/2019.
[6] R. Arning, S. Sassen, Flight control of micro aerial vehicles, in: Proceedings of the AIAA Guidance, Navigation, and Control Conference and Exhibit, Providence, RI, Aug. 2004, doi:10.2514/6.2004-4911.
[7] NASA, "Technologies of Broad Benefit: Power," [Online], https://mars.nasa.gov/mer/technology/bb_power.html. Last Access: 3/11/2019.
[8] NASA, Multi-Mission Radioisotope Thermoelectric Generator (MMRTG), NASA Facts, 2013.
[9] X. Ning, M. Gui, J. Fang, G. Liu, Differential X-ray pulsar aided celestial navigation for mars exploration, Aerosp. Sci. Technol. 62 (2017) 36–45, doi:10.1016/j.ast.2016.10.032.
[10] M. Maimone, Y. Cheng, L. Matthies, Two years of visual odometry on the mars exploration rovers,, J. Field Robot. 24 (3) (2007) 169–186, doi:10.1002/rob.20184.
[11] H. Durrant-Whyte, T. Bailey, Simultaneous localization and mapping: part I, IEEE Robot. Autom. Mag. 13 (2) (2006) 99–110, doi:10.1109/MRA.2006.1638022.
[12] T. Bailey, H. Durrant-Whyte, Simultaneous localization and mapping (SLAM): part II, IEEE Robot. Autom. Mag. 13 (3) (2006) 108–117, doi:10.1109/MRA.2006.1678144.
[13] N.L. Doh, W.K. Chung, A systematic representation method of the odometry uncertainty of mobile robots, Intell. Autom. Soft Comput. 12 (4) (Mar. 2013) 397–409, doi:10.1080/10798587.2006.10642941.
[14] D. Schleicher, L.M. Bergasa, M. Ocana, R. Barea, E. Lopez, Real-time hierarchical GPS aided visual SLAM on urban environments, in: Proceedings of the IEEE International Conference on Robotics and Automation, Kobe, Japan, May 2009, pp. 4381–4386, doi:10.1109/ROBOT.2009.5152370.
[15] L. Nardi, et al., Introducing SLAMBench, a performance and accuracy benchmarking methodology for SLAM, in: Proceedings of the IEEE International Conference on Robotics and Automation, May 2015, pp. 5783–5790, doi:10.1109/ICRA.2015.7140009.
[16] L.T. Thomspon, P.J. Best, Place cells and silent cells in the hippocampus of freely-behaving rats, J. Neurosci. 9 (7) (Jan. 1989) 2382–2390, doi:10.1523/JNEUROSCI.09-07-02382.1989.
[17] M.J. Milford, G.F. Wyeth, Mapping a suburb with a single camera using a biologically inspired SLAM system, IEEE Trans. Robot. 24 (5) (Oct. 2008) 1038–1053, doi:10.1109/TRO.2008.2004520.
[18] C.N. Boccara, et al., Grid cells in pre-and parasubiculum, Nat. Neurosci. 13 (2010) 987–994, doi:10.1038/nn.2602.
[19] S-C. Zhou, R. Yan, J-X. Li, Y-K, H. Tang, A brain-inspired SLAM system based on ORB features, Int. J. Autom. Comput. 14 (5) (Oct. 2017) 564–575, doi:10.1007/s11633-017-1090-y.
[20] M.J. Milford, G.F. Wyeth, D. Prasser, RatSLAM: a hippocampal model for simultaneous localization and mapping, in: Proceedings of the IEEE International Conference on Robotics & Automation, New Orleans, LA, Apr. 2004, pp. 403–408. doi: ROBOT.2004.1307183.
[21] J. Steckel, H. Peremans, BatSLAM: simultaneous localization and mapping using biomimetic sonar, PloS One 8 (1) (2013), doi:10.1371/journal.pone.0054076.
[22] M. Turan, Y. Almalioglu, H. Araujo, E. Konukoglu, M. Sitti, Deep EndoVO: a recurrent convolutional neural network (RCNN) based on visual odometry approach for endoscopic capsule robots, Neurocomputing 275 (2018) 1861–1870, doi:10.1016/j.neucom.2017.10.014.
[23] G.B. Zaffari, M.M. dos Santos, P.L.J. Drews-Jr, S.S.C. Botelho, Effects of water currents in a continuous attractor neural network for SLAM applications, in: Proceedings of the XIII Latin American Robotics Symposium and IV Brazilian Robotics Symposium, 2016, doi:10.1109/LARS-SBR.2016.62.
[24] Q-L. Li, Y. Song, Z.-G. Hou, Neural network based FastSLAM for autonomous robots in unknown environments, Neurocomputing 165 (Oct. 2015) 99–110, doi:10.1016/j.neucom.2014.06.095.
[25] M. Milford, G. Wyeth, Persistent navigation and mapping using a biologically inspired SLAM system, Int. J. Robot. Res. 29 (9) (Aug. 2010) 1131–1153, doi:10.1177/0278364909340592.
[26] V. Vanhoucke, A. Senior, M. Z.Mao, Improving the speed of neural networks on CPUs, in: Proceedings of the Deep Learning and Unsupervised Feature Learning Workshop, NIPS, 2011 Dec. 2011.
[27] A. Yazdanbakhsh, J. Park, H. Sharma, P. Lotfi-Kamran, H. Esmaeilzadeh, Neural Acceleration for GPU Throughput Processors, in: Proceedings of the Fourty-Eighth International Symposium on Microarchitecture, 2015, pp. 482–493, doi:10.1145/2830772.2820810.
[28] Y-H. Chen, J. Emer and V. Sze, Eyeriss v2: A Flexible and High-Performance Accelerator for Emerging Deep Neural Networks, arXiv:1807.07928, 2018.
[29] C. Li, et al., Efficient and self-adaptive in-situ learning in multilayer memristor neural networks, Nat. Commun. 9 (Jun. 2018), doi:10.1038/s41467-018-04484-2.
[30] G. Indiveri, F. Corradi, N. Qiao, Neuromorphic architectures for spiking deep neural networks, in: Proceedings of the IEEE International Electron Devices Meeting (IEDM), Washington, DC, 2015, pp. 4.2.1–4.2.4, doi:10.1109/IEDM.2015.7409623.
[31] G.W. Burr, et al., Experimental demonstration and tolerancing of a large-scale neural network (165 000 synapses) Using phase-change memory as the synaptic weight element, IEEE Trans. Electron Devices 62 (11) (2015) 3498–3507, doi:10.1109/TED.2015.2439635.
[32] D.B. Strukov, G.S. Snider, D.R. Stewart, R.S. Williams, The missing memristor found, Nature 453 (7191) (2008) 80–83, doi:10.1038/nature06932.
[33] M.A. Zidan, A.M. Eltawil, F. Kurdahi, H.A.H. Fahmy, K.N. Salama, Memristor multiport readout: a closed-form solution for sneak paths, IEEE Trans. Nanotechnol. 13 (2) (2014) 274–282, doi:10.1109/TNANO.2014.2299558.
[34] E. Herrmann, A. Rush, T. Bailey, R. Jha, Gate Controlled Three-Terminal Metal Oxide Memristor, IEEE Electron Device Lett. 39 (4) (2018), doi:10.1109/LED.2018.2806188.
[35] E. Herrmann, R. Jha, Gate-controlled memristors and their applications in neuromorphic architectures, in: Proceedings of the Great Lakes Symposium on VLSI (GLSVLSI '18), May 2018, pp. 385–390, doi:10.1145/3194554.3194613.
[36] KW. Brew, TS. Gershon, S. Kim, DM. Newns, TK. Todorov, Three-terminal metastable symmetric zero-volt battery memristive device, U.S. Patent No. 10/186,657, Dec. 2018.
[37] V.K. Sangwan, H-S. Lee, M.C. Hersam, Gate-tunable memristors from monolayer MoS_2, in: Proceedings of the IEEE International Electron Devices Meeting, San Francisco, CA, Dec. 2017, doi:10.1109/IEDM.2017.8268330.

[38] E.J. Fuller, et al., Li-ion synaptic transistor for low power analog computing, Adv. Mater. 29 (4) (Jan. 2017), doi:10.1002/adma.201604310.
[39] J. Son, et al., Epitaxial $SrTiO_3$ films with electron mobilities exceeding 30,000 $cm^2V^{-1}s^{-1}$, Nat. Mater. 9 (Jun. 2010) 482–484, doi:10.1038/NMAT2750.
[40] T.J. Bailey, R. Jha, Understanding synaptic mechanisms in $SrTiO_3$RRAM devices, IEEE Trans. Electron Devices 65 (8) (Aug. 2018) 3514–3520, doi:10.1109/TED.2018.2847413.
[41] F.M. Pontes, E.J.H. Lee, E.R. Leite, E. Longo, J.A. Varela, High dielectric constant of $SrTiO_3$ thin films prepared by chemical process, J. Mater. Sci. 35 (2000) 4783–4797, doi:10.1023/A1004816611050.
[42] J. Robertson, Band offsets of wide-band-gap oxides and implications for future electronic devices, J. Vac. Sci. Technol. B 18 (3) (Jun. 2000) 1785–1791, doi:10.1116/1.591472.
[43] A.J. Rush, R. Jha, NbOx synaptic devices for spike frequency dependent plasticity learning, in: Proceedings of the Seventy-Fifth Annual Device Research Conference (DRC), South Bend, IN, 2017, pp. 1–2, doi:10.1109/DRC.2017.7999430.
[44] F-C. Chiu, A review on conduction mechanisms in dielectric films, Adv. Mater. Sci. Eng. (2014), doi:10.1155/2014/578168.
[45] X. Liu, et al., Co-occurrence of threshold switching and memory switching in Pt/NbOx/Pt cells for crosspoint memory applications, IEEE Electron Device Lett. 33 (2) (Feb. 2012), doi:10.1109/LED.2011.2174452.
[46] KC. Saraswat, Metal/Semiconductor Ohmic Contacts, Stanford Online Lecture Notes. [Online], https://web.stanford.edu/class/ee311/NOTES/Contacts.pdf. Last Access: 3/11/2019.
[47] E. Culurciello, R. Etienne-Cummings, K.A. Boahen, A biomorphic digital image sensor, IEEE J. Solid-State Circuits 38 (2) (Feb. 2003) 281–294, doi:10.1109/JSSC.2002.807412.
[48] J.M. Cruz-Albrecht, M.W. Yung, N. Srinivasa, Energy-efficient neuron, synapse and STDP integrated circuits, IEEE Trans. Biomed. Circuits Syst. 6 (3) (Jun. 2012) 246–256, doi:10.1109/TBCAS.2011.2174152.
[49] L. Zhao, Q. Hong, X. Wang, Novel designs of spiking neuron circuit and STDP learning circuit based on memristor, Neurocomputing 314 (Nov. 2018) 207–214, doi:10.1016/j.neucom.2018.06.062.
[50] T. Tuma, A. Pantazi, M. Le Gallo, A. Sebastian, E. Eleftheriou, Stochastic phase-change neurons, Nat. Nanotechnol. 11 (May 2016) 693–699, doi:10.1038/NNANO.2016.70.
[51] A. Joubert, B. Belhadj, O. Temam, R. Héliot, Hardware spiking neurons design: analog or digital? in: Proceedings of the International Joint Conference on Neural Networks, Brisbane, Australia, Jun. 2012, doi:10.1109/IJCNN.2012.6252600.
[52] P.V. Thanh, J. Sajad, V. Sundarapandian, V. Christos, W. Xiong, A novel memristive neural network with hidden attractors and its circuitry implementation, Sci. China Technol. Sci. 59 (3) (Mar. 2016) 358–363, doi:10.1007/s11431-015-5981-2.
[53] V. Milo, D. Ielmini, E. Chicca, Attractor networks and associative memories with STDP learning in RRAM synapses, in: Proceedings of the IEEE International Electron Devices Meeting, San Francisco, CA, Dec. 2017, doi:10.1109/IEDM.2017.8268369.
[54] L. Susman, N. Brenner and O. Barak, Stable memory with unstable synapses, arXiv:1808.00756, 2018.
[55] M.J. Milford, J. Wiles, G.F. Wyeth, Solving Navigational Uncertainty Using Grid Cells on Robots, PLoS Comput. Biol. 6 (11) (2010) e1000995, doi:10.1371/journal.pcbi.1000995.
[56] G. Tang, K.P. Michmizos, Gridbot: an autonomous robot controlled by a spiking neural network mimicking the brain's navigational system, in: Proceedings of the International Conference on Neuromorphic Systems (ICONS '18), Knoxville, TN, USA, ACM, July 23-26, 2018 Article No. 4, doi:10.1145/3229884.3229888.
[57] K. Tateno, et al., CNN-SLAM: real-time dense monocular SLAM with learned depth prediction, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 6243–6252, doi:10.1109/CVPR.2017.695.
[58] Y. Cao et al., Arizona State University's Predictive Technology Models. [Online], http://ptm.asu.edu/. Last Access: 3/11/2019.
[59] R.F. Janninck, D.H. Whitmore, Electrical Conduction in Nonstoichiometric α-Nb_2O_5, J. Chem. Phys. 37 (1962) 2750–2754, doi:10.1063/1.1733100.
[60] N. Alimardani, J.M. McGlone, J.F. Wager, J.F. Conley Jr., Conduction processes in metal-insulator-metal diodes with Ta2O5 and Nb_2O_5 insulators deposited by atomic layer deposition, J. Vac. Sci. Technol. A 32 (Feb. 2014) pp. 01A122-1–01A122-6, doi:10.1116/1.4843555.

Alexander Jones is currently a doctoral student in EECS department at the University of Cincinnati. He completed his Bachelors of Science in Computer Engineering from the University of Cincinnati in 2015 and Masters of Science in Computer Engineering at the University of Cincinnati in 2016. In 2017, Alexander attended the Neuromorphic Computing Symposium in Knoxville, TN to give a poster presentation on the modeling of neurons within HSPICE. The current focus of his doctoral research is the design and simulation of neuromorphic architectures that use novel memory technologies. He also works as a part time instructor to teach the topics of digital logic design and C++ programming at the University of Cincinnati.



Andrew Rush is currently a doctoral student in the EECS department at the University of Cincinnati. He completed his Bachelors of Science in Mechanical Engineering from Miami University in 2012 and Masters of Science in Computational Engineering and Computing from Miami University in 2016. Andrew presented work at the *2017 Device Research Conference* at the University of Notre Dame on the behavior of two terminal NbOx memristive devices. His current work involves the design and fabrication of novel memristive technologies.

Cory Merkel is an assistant professor with the Department of Computer Engineering, Rochester Institute of Technology. He earned his BS and MS degrees in computer engineering (2011) and a Ph.D. in microsystems engineering (2015), all from RIT. From 2016 to 2018, Dr. Merkel was a research electronics engineer with the Information Directorate, Air Force Research Lab. His current research focuses on mapping of AI algorithms, primarily artificial neural networks, to mixed-signal hardware and the design of brain-inspired computing systems using emerging technologies such as memristors. Cory has published his work in a number of peer-reviewed conferences, journals, and books, has been awarded one U.S. patent, and has two pending. He is also engaged in several mentorship and STEM outreach activities.

Eric Herrmann earned both his Bachelors and Masters of Science in Electrical Engineering from the University of Cincinnati in the Spring of 2018. His work primarily focused on the fabrication of gated-ReRAM devices and designing neuromorphic architectures adapted for gated-ReRAM devices. His work on gated-ReRAM was published in *IEEE Electron Device Letters* in April 2018, and his work on neuromorphic architectures for gated-ReRAM devices was published at the *2019 Great Lakes Symposium on VLSI* in May 2019.

Ajey Jacob leads the differentiating technology research group for GLOBALFOUNDRIES (GF) TD Research division. He has also lead the exploratory device and integration research group at GF, and the pathfinding integration research program for the IBM Alliance in Albany, NY. Until 2011, Ajey Jacob worked as a senior research scientist at Intel Corporation. Ajey Jacob received his Ph.D. in Physics from the Chalmers University of Technology/Gothenburg University, Gothenburg, Sweden in 2002. Ajey Jacob has more than 115 USPTO patents and more than 50 patents pending USPTO approval. He has also published three book chapters, more than 75 journal and conference papers in a wide area of scaled and differentiating technologies.

Clare Thiem is a Senior Electronics Engineer in the Air Force Research Laboratory's (AFRL's) Information Directorate's High Performance Systems Branch (RITB). He holds a Bachelor of Science in Engineering (Aerospace Engineering) from the University of Michigan and a Master of Science in Mechanical Engineering from the National Technological University. Currently he serves as the Lead Engineer and Project Lead for activities pursuing advanced computing concepts for reasoning and perception for size, weight, and power constrained environments. His current research interest seeks to develop practical, hardware-based, biologically inspired computing technology for Air Force systems.

Dr. Rashmi Jha is an Associate Professor in Electrical Engineering and Computing Systems Department at the University of Cincinnati. She worked as an Assistant Professor and then Associate Professor in Electrical Engineering and Computer Science Department at the University of Toledo from 2008 to 2015. Before this, she worked as a Process Integration Engineer for 45 nm/32 nm High-k/Metal Gates based Advanced CMOS technologies at Semiconductor Research and Development Center, IBM, East Fishkill New York between 2006-2008. She finished her Ph.D. and M.S. in Electrical Engineering from North Carolina State University in 2006 and 2003, respectively, and B.Tech. in Electrical Engineering from Indian Institute of Technology (IIT) Kharagpur, India in 2000.