

BENG 260 Final Project Report

Evaluation of Memristor based models of Neurons and Neural Networks

Rajkumar Chinnakonda Kubendran

PID: A53067485

Department of Electrical and Computer Engineering

University of California San Diego

La Jolla, CA 92093

rchinnak@eng.ucsd.edu

Abstract

This project aims to explore if neurons and neural networks can be modeled and simulated using Memristors. Recent literature shows a rigorous and comprehensive nonlinear circuit-theoretic foundation for the memristive Hodgkin–Huxley Axon Circuit model [1]. Also analog hardware architecture of a memristor bridge synapse-based multilayer neural network and its learning scheme has been presented in [2-3]. In order to analyze and design memristive circuits, a laplace domain expression has been derived in [4]. This project combines these works together to analyze and derive equations for memristive Hodgkin-Huxley axons and memristive synapses. These equations can then be used for modeling and simulating simple neural networks and possibly associative learning.

1 Introduction

Memristor behaves like a ‘resistor with memory’ and has unique properties not found in the other basic circuit elements – resistor, capacitor and inductor. Unique properties of memristors could possibly be used in non-volatile memories and neuromorphic computing and drastically reduce area and power dissipation.

1.1 What is Memristance?

There are four fundamental variables in circuit theory, Charge (Q), Flux linkage (Φ), Current (I) and Voltage (V), which give rise to six relations out of which five are well known,

$$q(t) = \int_{-\infty}^t i(\tau)d\tau \quad \phi(t) = \int_{-\infty}^t v(\tau)d\tau \quad R = \frac{dv}{di} \quad L = \frac{d\phi}{di} \quad C = \frac{dq}{dv}$$

From symmetry arguments, the 6th relation between charge and flux linkage gives the fourth fundamental passive non-linear circuit element called MEMRISTOR (M).

$$M = \frac{d\phi}{dq} \quad \text{where } d\phi = vdt \text{ and } dq = idt$$

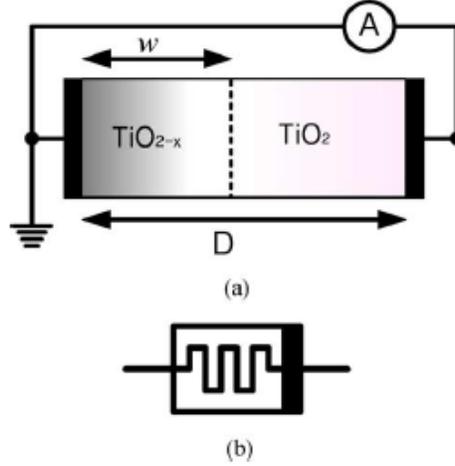
$$\therefore M(q) = \frac{v(t)dt}{i(t)dt} = \frac{v(t)}{i(t)} \equiv R$$

31 Memristor cannot be realized using any linear combination of R, L and C. Memristor is the
 32 only passive element that exhibits hysteresis behavior [5].

33

34 1.2 First Memristor Device

35 The first Memristor device was fabricated at Hewlett-Packard (HP) Labs and reported in the
 36 Journal, Nature in 2008 [6].



37

38

39

Figure 1. a) Cross section of the first TiO₂ based Memristor device
 b) Symbol for Memristor

40 The device is made of TiO₂ along with TiO_{2-x}, a slight variant in doping. The device of
 41 length D (nm) is separated into two sections, one section of width w (nm) that has TiO_{2-x}
 42 doping and another section of width D-w (nm) that has TiO₂ doping. In this device, the
 43 memristor width, w, is the state variable since it changes with time depending on the input
 44 stimulus current. μ_v is the dopant mobility. The TiO_{2-x} section has a resistance of R_{ON} and the
 45 TiO₂ section has resistance, R_{OFF}. Typical values for the device parameters are, $\mu_v = 10^{-14}$
 46 m²/Vs, R_{ON} = 116Ω, R_{OFF} = 16KΩ and D = 10nm.

47 Hence the voltage across the memristor at any time t, when a current stimulus i(t) is applied,
 48 is given by,

$$v(t) = \left[R_{ON} \frac{w(t)}{D} + R_{OFF} \left(1 - \frac{w(t)}{D} \right) \right] i(t)$$

$$\text{Since } v(t) = \frac{d\phi(q)}{dq} i(t) \equiv M(q) i(t), \quad M(q) = R_{ON} \frac{w(t)}{D} + R_{OFF} \left(1 - \frac{w(t)}{D} \right)$$

49 The state variable changes with time as follows,

$$\frac{dw}{dt} = \mu_v \frac{R_{ON}}{D} i(t) F_p(w) \quad \text{where } F_p(w) = 1 - \left(2 \frac{w}{D} - 1 \right)^{2p}$$

$$\text{Hence } w(t) = \mu_v \frac{R_{ON}}{D} \int_0^t i(\tau) d\tau + w_0 = \mu_v \frac{R_{ON}}{D} q(t) + w_0$$

50 From the above equation, we see that the state variable, w is a function of charge and thus
 51 memristance, M, is a function of charge. Memristive behavior is similar to the voltage
 52 controlled channel conductances in the Hodgkin-Huxley Neuron Model.

53

54 1.3 Project Objectives

55

56

- Develop a Memristor Model in Python
- Propose a prospective Memristor based neuron model

- 57 ➤ Simulate Memristor bridge for synaptic weight adjustment
- 58 ➤ Simulate Neural network with Memristive neuron node and bridge
- 59 ➤ Demonstrate associative learning using Memristors
- 60 ➤ Develop a HSPICE model for Memristor simulation in Cadence
- 61 ➤ Simulate circuits in Cadence with proper biasing of transistors

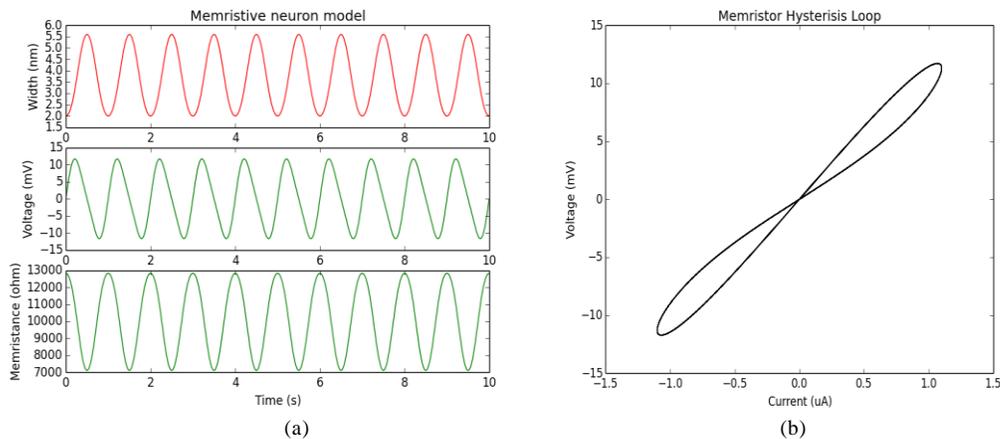
62

63 2 Modeling in Python

64

65 2.1 Memristor device model

66 Python code is used to implement a Memristor model using device equations and odeint
 67 function to calculate change in the width of Memristor – state variable. The code uses actual
 68 values for device dimensions, On/Off Resistance etc., to compute the memristance value.
 69 Some gain adjustments are required to get signals in the desired range. This proves to be a
 70 challenge when implementing circuits in Cadence.



71

72

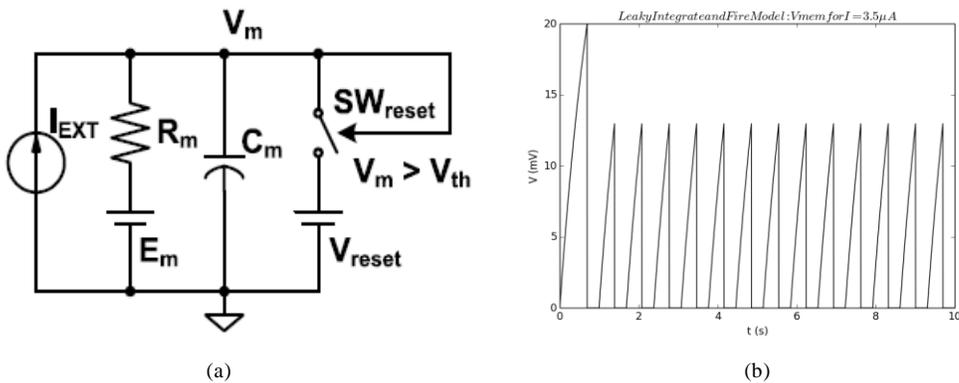
73

Figure 2. a) Output of a current controlled memristor model (b) Memristor Hysteresis Loop

74

75 2.2 Leaky Integrate and Fire Neuron Model

76 The most common model used to mimic a neuron generating action potentials in circuits is
 77 the Leaky-Integrate-and-Fire (LIF) Model.



78

79

80

81

Figure 3. a) Circuit diagram of Leaky Integrate and Fire Neuron (LIF) model
 (b) Action Potential voltage output of LIF model for input stimulus current of $3.5\mu A$.

82

83

The differential equation that describes the output membrane voltage variation with time is given by,

$$C_m \frac{dV_m}{dt} = \frac{V_m - E_m}{R_m} + I_{EXT}$$

84 The spiking frequency of the LIF neuron output is given by,

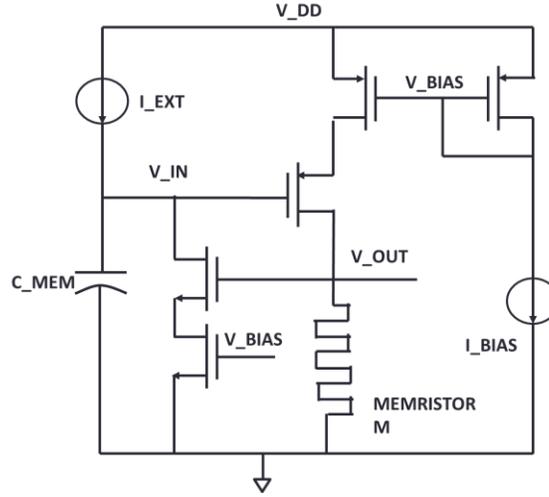
$$f_{spike} = \frac{1}{T} = \left[t_{ref} + R_m C_m \ln \left(\frac{R_m I_{EXT} - V_{reset}}{R_m I_{EXT} - V_{th}} \right) \right]^{-1}$$

85

86 2.3 Modified LIF Neuron Model using Memristors

87 A modified version of the conventional LIF model is proposed. This model uses one or
88 multiple memristors to mimic the non-linear voltage controlled channel conductance as
89 described in the Hodgkin-Huxley neuron model. Whereas in LIF model, a simple resistor is
90 used, this gives only a constant conductance.

91 Figure 4 is the circuit schematic for the new neuron model based on memristor. An external
92 DC current acts as stimulus to create a voltage ramp when dumped on the membrane
93 capacitance, initially reset to 0V. This small signal voltage ramp is converted to a small
94 signal current ramp using the g_m of an appropriately biased PMOS transistor. This small
95 signal current is then dropped on a Memristor, M. The voltage across the memristor is the
96 output voltage of the neuron. A reference bias current is used to bias the transistors in the
97 correct region of operation.



98

99

Figure 4. Circuit diagram of the modified LIF Neuron using Memristor

100 Circuit analysis of the above schematic gives the following equations,

$$i_M(t) = g_m v_{in}(t) = g_m \frac{I_{EXT}}{C_m} t + v_{in}(0), \quad \text{where } g_m = \sqrt{(2\beta I_{BIAS})}$$

$$v_{out}(t) = v_M(t) = M(t)i(t) = M(t)g_m v_{in}(t) \quad \text{where } M(t) = R_{ON} \frac{w(t)}{D} + R_{OFF} \left(1 - \frac{w(t)}{D} \right)$$

101 The rate of change of the state variable – Memristor width, will be directly proportional to
102 the time of the input current ramp. Hence, width itself will be proportional to the square of
103 the time. This in turn makes Memristance, M to decrease inverse-square with time from the
104 initial value to the minimum value, R_{ON} for the maximum width $w = D$.

$$\frac{dw(t)}{dt} = \mu_V \frac{R_{ON}}{D} i(t) F_p(w) = \mu_V \frac{R_{ON}}{D} \left(g_m \frac{I_{EXT}}{C_m} t \right) \left[1 - \left(\frac{2w}{D} - 1 \right) \right]^{2p}$$

$$w(t) = \mu_V \frac{R_{ON}}{D} \int_0^t i(\tau) d\tau + w_0 = \mu_V \frac{R_{ON}}{2D} \left(g_m \frac{I_{EXT}}{C_m} t^2 \right) \left[1 - \left(\frac{2w}{D} - 1 \right) \right]^{2p} + w_0$$

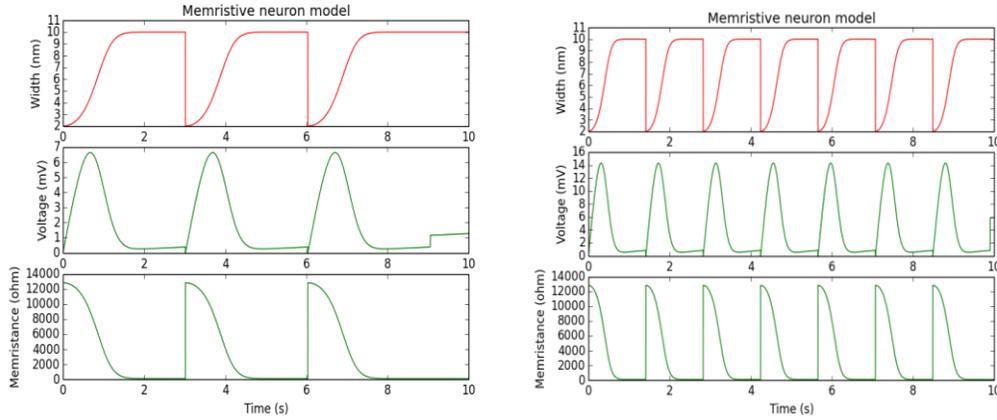
105

106

107 Finally the output voltage is,

$$v_{out}(t) = \left[R_{ON} \frac{w(t)}{D} + R_{OFF} \left(1 - \frac{w(t)}{D} \right) \right] g_m \frac{I_{EXT}}{C_m} t$$

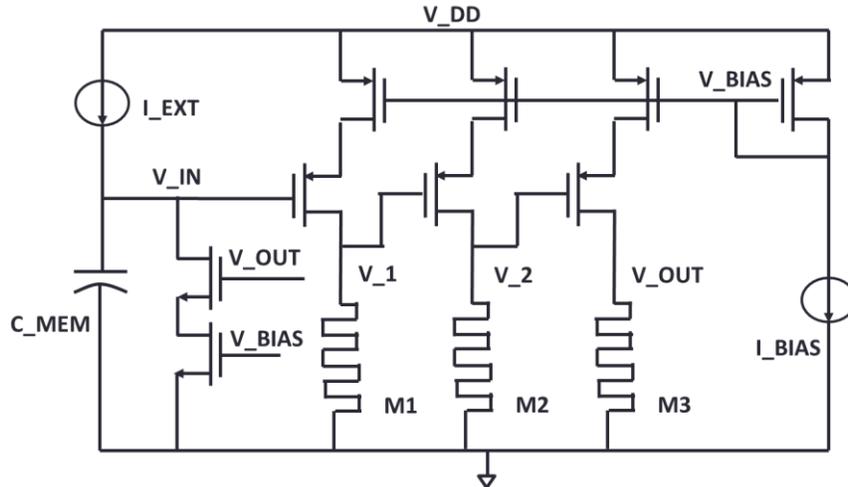
108 This equation when plotted gives a single action potential, similar to that of a biological
 109 neuron. Now, if this voltage can be reset by detecting when it crosses a threshold and then
 110 pull down the capacitor to 0V and start a new cycle again, we can get continuous action
 111 potentials fired as long as there is an external injected current. This reset mechanism is done
 112 using a NMOS pull down transistor as shown in Figure 4. Figure 5 shows the action
 113 potential waveforms generated using this model for different values of stimulus current, 1μA
 114 and 5μA.



115
 116
 117
 118

(a) (b)
 Figure 5. Action Potential voltage output of modified LIF model for input stimulus current of
 a) 1μA and b) 5μA.

119 This model can be extended further to represent higher powers of the state variable, as seen
 120 in the Hodgkin-Huxley model where the Na channel current is proportional to the 3rd
 121 power of the rate factor, m and the K channel current is proportional to the 4th
 122 power of the rate factor, n. The extended version is shown in Figure 6 below,



123
 124
 125

Figure 6. Circuit diagram of extended version of the modified LIF Neuron using Memristors

The output voltage of each stage is then given by,

$$v_1(t) = M_1(t)i_1(t) = M_1(t)g_{m_1}v_{in}(t)$$

$$v_2(t) = M_2(t)i_2(t) = M_2(t)g_{m_2}v_1(t) = M_1(t)M_2(t)g_{m_1}g_{m_2}v_{in}(t)$$

126 Extending further, for N stages, the output voltage will be,

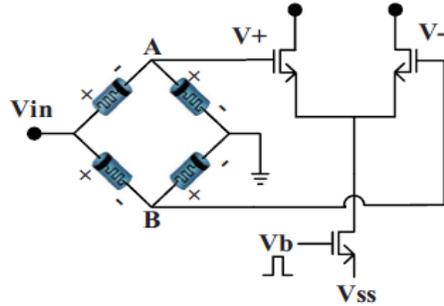
$$v_N(t) = M_N(t)i_N(t) = [M(t)g_m]^N v_{in}(t)$$

127 From above equation, we can see that the voltage is dependent on the Nth power of
 128 Memristance state variable, w. Thus this model can be designed to have state variables very
 129 similar to the channel conductance in the Hodgkin-Huxley model.

130

131 2.4 Memristive Bridge for Synaptic Weight Adjustment

132 A memristive bridge made of four memrsitors as shown in Figure 7 can be used to
 133 implement the neural synapse. V_b is the bias voltage required to set the magnitude of the
 134 output differential current using the differential pair.



135

136 Figure 7. Memristor bridge for Synapse weight adjustment

137 The weight of the synapse, W_{syn} , will then be given by,

$$V_{out} = V_A - V_B = W_{syn}V_{in} \quad \text{where } W_{syn} = \left(\frac{M2}{M1 + M2} - \frac{M4}{M3 + M4} \right)$$

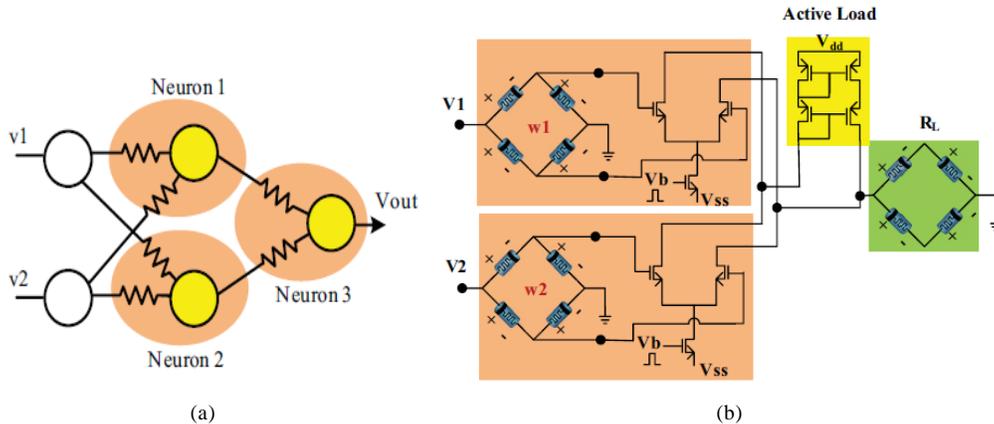
138 The differential output current, generated by the differential pair with trans-conductance g_m ,

$$i_{out} = i_{out}^+ - i_{out}^- = \frac{1}{2} g_m W_{syn} V_{in} - \left(-\frac{1}{2} g_m W_{syn} V_{in} \right) = g_m W_{syn} V_{in}$$

139

140 3 Neural Networks with Memristive Synapses

141 A neural network consists of one or more neurons connected through synapses and fire
 142 through different interaction mechanisms like feedforward, feedback inhibition or excitation.
 143 The weight of the synapses can increase or decrease depending on the strength of the activity
 144 between the two neurons which the synapse connects. Thus the output voltage of such
 145 networks is the weighted sum of the input voltages.



146

147

148 Figure 8. a) A simple neural network with 3 neurons and 2 synapses to demonstrate associative learning

149 b) Schematic showing the summation of the synaptic outputs using active load and memristor bridge load.

150 From Figure 8b), the differential output current and output voltage at the
 151 output of the N synaptic connections is given by,

$$i_{out} = \sum_{K=1}^N g_m W_{syn}^K V_{in}^K \quad V_{out} = R_L \sum_{K=1}^N g_m W_{syn}^K V_{in}^K$$

152 However, the voltage range of V_{out} is limited by the differential pair.

$$-V_{SS} + 2V_{th} \leq V_{OUT} \leq V_{DD} - 2V_{th}$$

153 Therefore, the output voltage range of the neural network is clipped to a
 154 V_{max} and V_{min} .

$$V_{out} = R_L I_{out}, \quad \text{if } \frac{(-V_{SS} + 2V_{th})}{R_{OUT}} \leq I_{out} \leq \frac{(V_{DD} - 2V_{th})}{R_{OUT}}$$

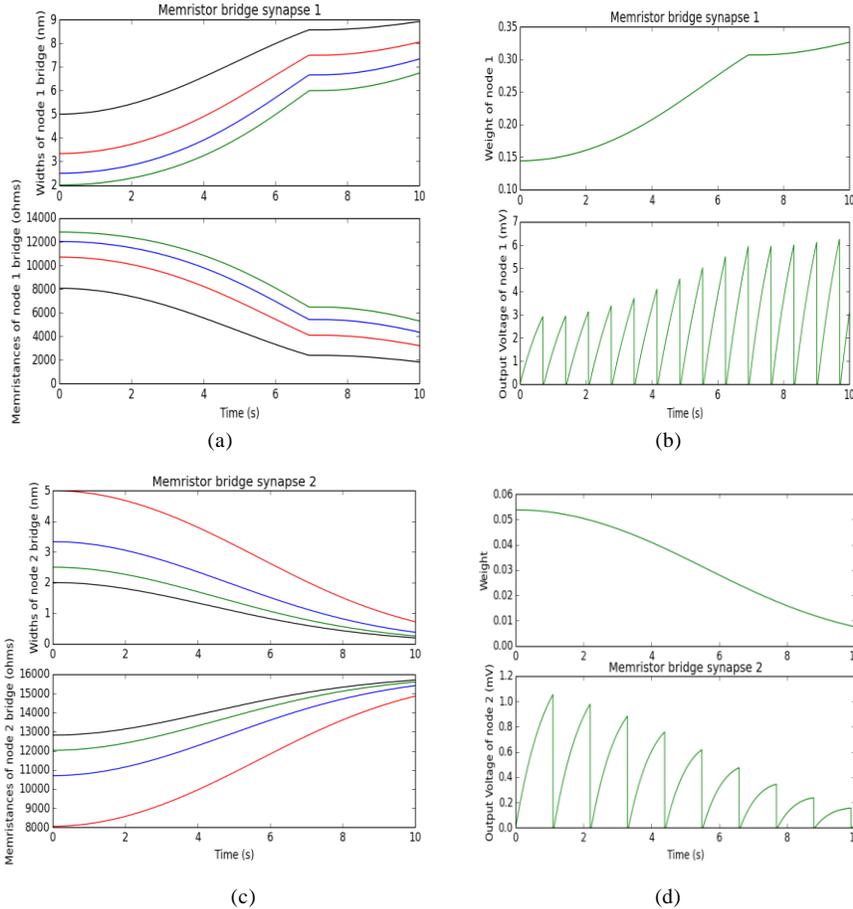
$$V_{out} = V_{max}, \quad \text{if } \frac{(V_{DD} - 2V_{th})}{R_{OUT}} \leq I_{out} \quad V_{out} = V_{min}, \quad \text{if } I_{out} \leq \frac{(-V_{SS} + 2V_{th})}{R_{OUT}}$$

155

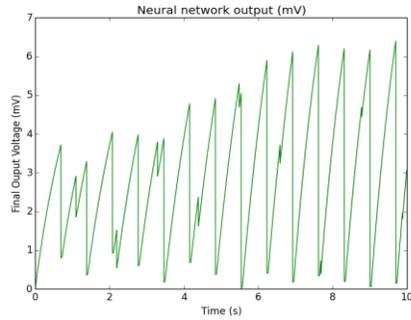
156 3.1 Neural Network simulation with LIF Neurons

157 The neural network given in Figure 8a) was simulated using LIF neurons and memristive
 158 synapses. Neuron 1 was given a positive stimulus current to increase the weight of the
 159 synapse connecting Neurons 1 and 3. In contrast, Neuron 2 was given a negative stimulus
 160 current to decrease the weight of the synapse connecting Neurons 2 and 3. Figure 9a) shows
 161 the change in the widths and Memristance of each Memristor in the bridge in Neuron1-
 162 Neuron3 synapse. Figure 9b) shows the overall weight of the synapse and the output voltage
 163 of the Neuron1-Neuron3 synapse. Figure 9c) and 9d) show the same waveforms as shown in
 164 9a) and 9b) but for the Neuron2-Neuron3 synapse. Figure 9e) shows the final output voltage
 165 of the neural network.

166
 167



168
 169



(e)

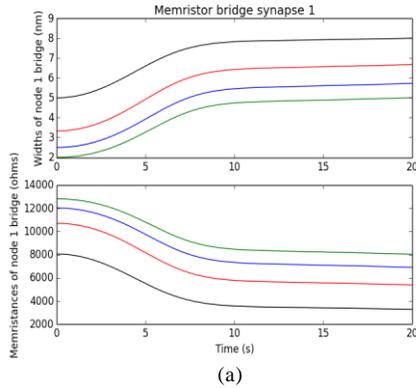
Figure 9. Neural Network simulation with LIF Neurons.

170
171
172
173
174

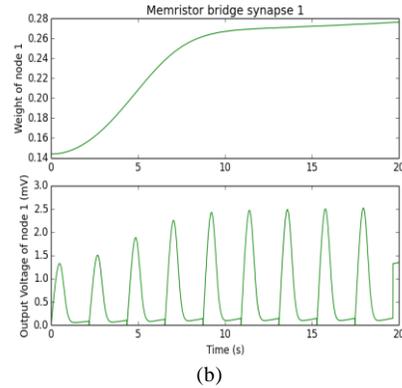
3.2 Neural Network simulation with Modified LIF Neurons

175 The neural network given in Figure 8a) was simulated using Modified LIF neurons with
176 memristors and memristive synapses. Neuron 1 was given a positive stimulus current to
177 increase the weight of the synapse connecting Neurons 1 and 3. In contrast, Neuron 2 was
178 given a negative stimulus current to decrease the weight of the synapse connecting Neurons
179 2 and 3. Figure 10a) shows the change in the widths and Memristance of each Memristor in
180 the bridge in Neuron1-Neuron3 synapse. Figure 10b) shows the overall weight of the
181 synapse and the output voltage of the Neuron1-Neuron3 synapse. Figure 10c) and 10d) show
182 the same waveforms as shown in 10a) and 10b) but for the Neuron2-Neuron3 synapse.
183 Figure 10e) shows the final output voltage of the neural network.

184
185

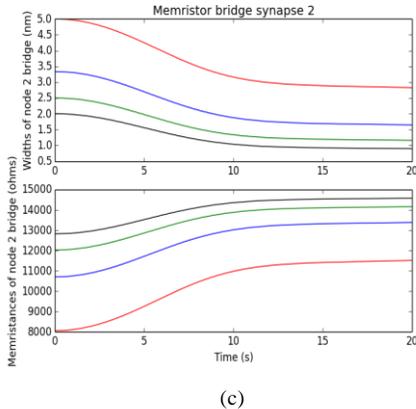


(a)

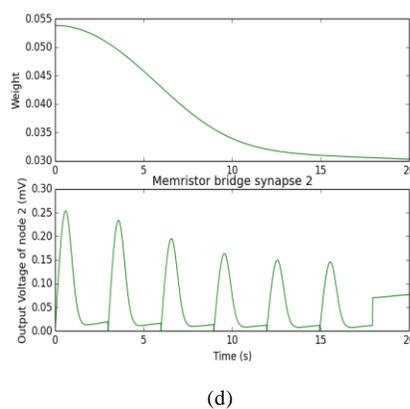


(b)

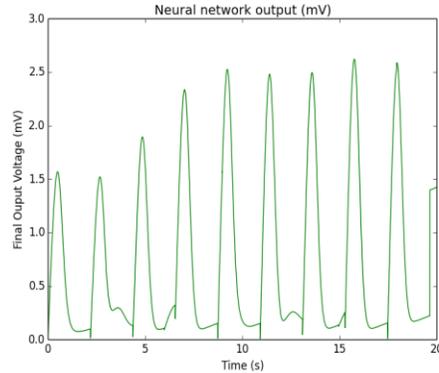
186
187



(c)



(d)



(e)
Figure 10. Neural Network simulation with LIF Neurons.

188
189
190

191 The same neural network can be used to demonstrate associative learning for a simple case
192 as shown in Figure 10 [4, 7]. Here a dog is trained to expect food right after a bell rings and
193 hence starts salivating. Eventually, after sometime the dog starts salivating even if only the
194 bell rings. In terms of the synaptic weights, the weight of the synapse connecting the neuron
195 that fires when the bell rings increases with time until it is sufficient to cause the output
196 neuron to fire and initiate salivation.

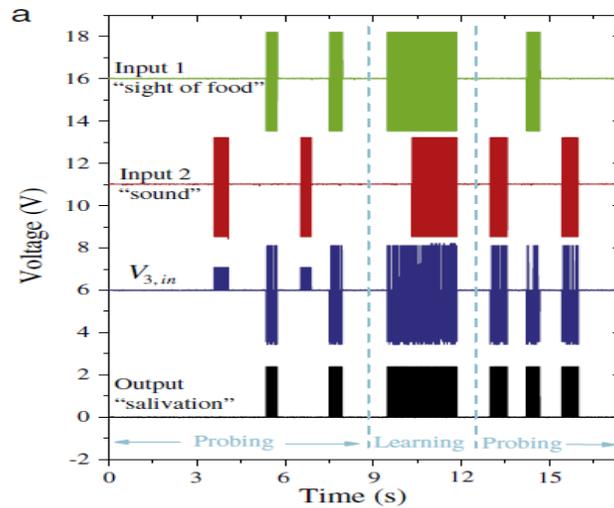


Figure 10. Neuron Firing Pattern for different excitation inputs demonstrating associative learning [7].

197
198

199

200 4 Simulations in Cadence

201 The final objective of this project is create Memristor model in SPICE language and use it
202 for simulating actual circuits using a circuit simulator like Spectre in Cadence Framework. A
203 SPICE model for the Memristor was created successfully as explained in the next section
204 [4]. However, implementing actual circuits for the synapse and neuron faced several issues
205 in convergence. Due to insufficient time to fix these issues and try different approaches, the
206 circuit simulations could not be completed. Nonetheless, I wish to pursue this as part of my
207 own interest and aim to simulate them successfully in the near future.

208

209 4.1 SPICE model for Memristor

210 A compact and configurable SPICE model, based on its charge-flux constitutive
211 relationships, considers memristance itself as the state variable. The subcircuit takes device
212 dimensions, resistance boundaries and mobility, as input CDF parameters to compute the
213 memristance value [8]. Figure 11a) shows the block diagram of the model of the Memristor

214 and 11b) shows the testbench schematic. Figure 12a) shows the simulated current and
 215 voltage waveforms across the memristor. Finally, Figure 12b) shows the hysteresis loop
 216 generated from these waveforms.

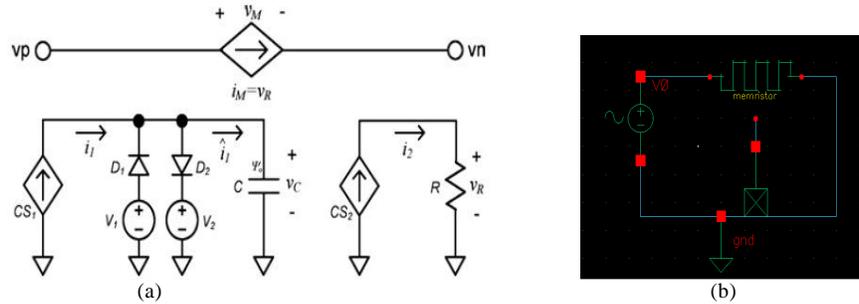


Figure 11. a) Small signal model of Memristor b) Memristor simulation testbench.

217
 218
 219
 220

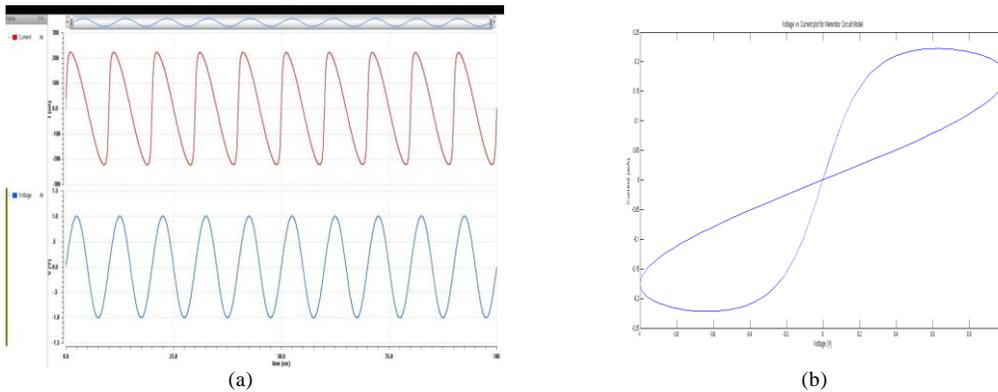


Figure 12. a) Output current of a voltage controlled memristor SPICE model b) Memristor hysteresis loop.

221
 222
 223
 224
 225

5 Summary

226 A Memristor Model was developed in Python. This model was used to simulate a Memristor
 227 bridge for synaptic weight adjustment in a neural network. A modified Leaky Integrate and
 228 Fire model was proposed which uses one Memristor as the state element. This model seems
 229 to be better than a simple LIF neuron and can be extended to implement higher powers of the
 230 state variable. Neural networks with LIF and Memristive neuron node and Memristance
 231 bridge were simulated and compared. Such neural networks can be shown to demonstrate
 232 associative learning. A HSPICE model for Memristor was implemented for simulation in
 233 Cadence. Simulation of circuits in Cadence with proper biasing of transistors is incomplete
 234 due to convergence issues and insufficient time. However, this will be pursued and
 235 completed in the near future.

236 Acknowledgments

237 I would like to thank Prof. Gert Cauwenberghs for the encouragement to pursue this project
 238 and the TAs Chul Kim, Jonathan and Bruno Pedroni for all their help.

239 References

- 240 [1] L.O. Chua, V. Sbitnev and H. Kim, "Hodgkin-Huxley Axon is made of Memristors," International
 241 Journal of Bifurcation and Chaos, vol. 22, no. 3, pp. 1230011 1–48, Jan. 2012.
- 242 [2] S. P. Adhikari, C. Yang, H. Kim and L.O. Chua, "Memristor Bridge Synapse-Based Neural
 243 Network and Its Learning," IEEE Trans. On Neural Networks and Learning Systems, vol. 23, no. 9,
 244 pp. 1426–1435, Sept. 2012.
- 245 [3] H. Kim, M. P. Sah, C. Yang, T. Roska, and L. O. Chua, "Neural synaptic weighting with a pulse-
 246 based memristor circuit," IEEE Trans. Circuit Syst. I, vol. 59, no. 1, pp. 148–158, Jan. 2012.
- 247 [4] R. Kubendran, "Electromagnetic and Laplace domain analysis of Memristance and Associative

248 learning using Memristive Synapses modeled in SPICE,” IEEE International Conference on Devices,
249 Circuits and Systems, pp. 622-626, Mar. 2012.

250 [5] Leon O. Chua and Sung Mo Kang, “Memristive Devices and Systems”, Proc. of the IEEE, Vol.64,
251 No.2, pp.209–223, Feb 1976.

252 [6] Dmitri B. Strukov, Gregory S. Snider, Duncan R. Stewart and R. Stanley Williams, “The missing
253 memristor found”, Letters to Nature, Vol.453, No.7191, pp.80–83, May 2008.

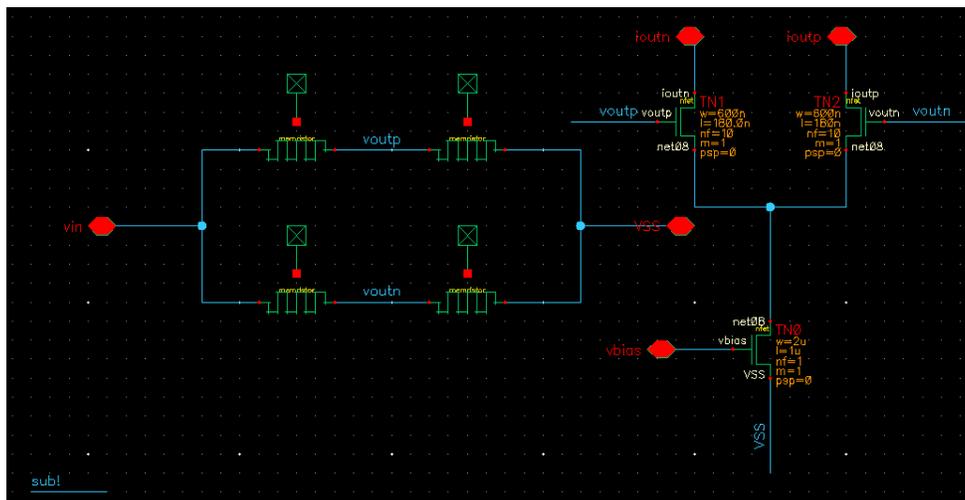
254 [7] Yuriy V. Pershin, Massimiliano Di Ventra, “Experimental Demonstration of associative memory
255 with memristive neural networks”, Neural Networks, Elsevier, May 2010.

256 [8] Sangho Shin, et.al., “Compact Models for Memristors Based on Charge–Flux Constitutive
257 Relationships”, IEEE Trans. on CAD of Integrated Circuits and Systems, Vol.29, No.4, April 2010.

258

259 Appendix

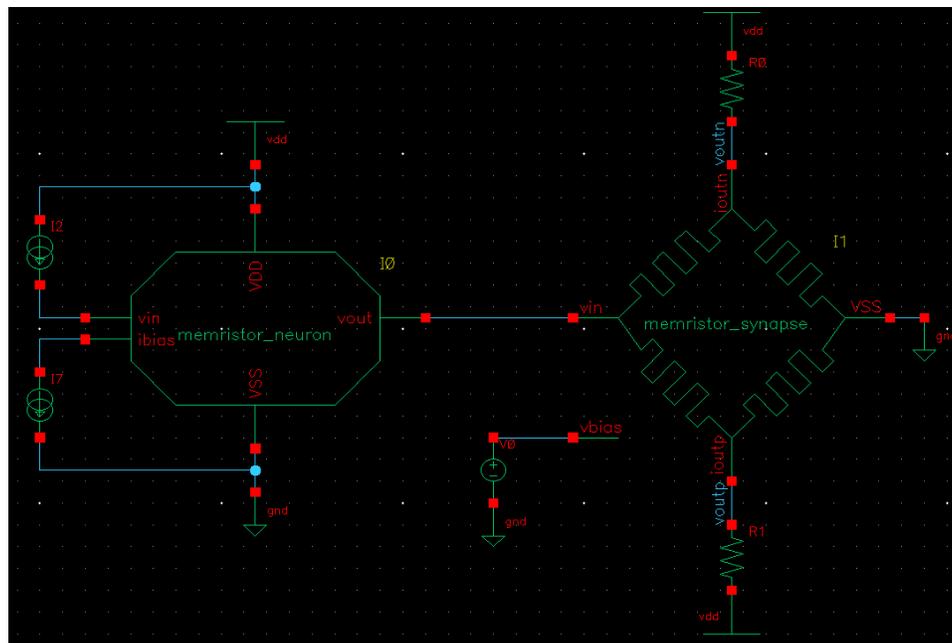
260 Schematics in Cadence for Circuit Simulation



261

262

Appendix Figure 1. Schematic of Memristor Synaptic Weight Bridge

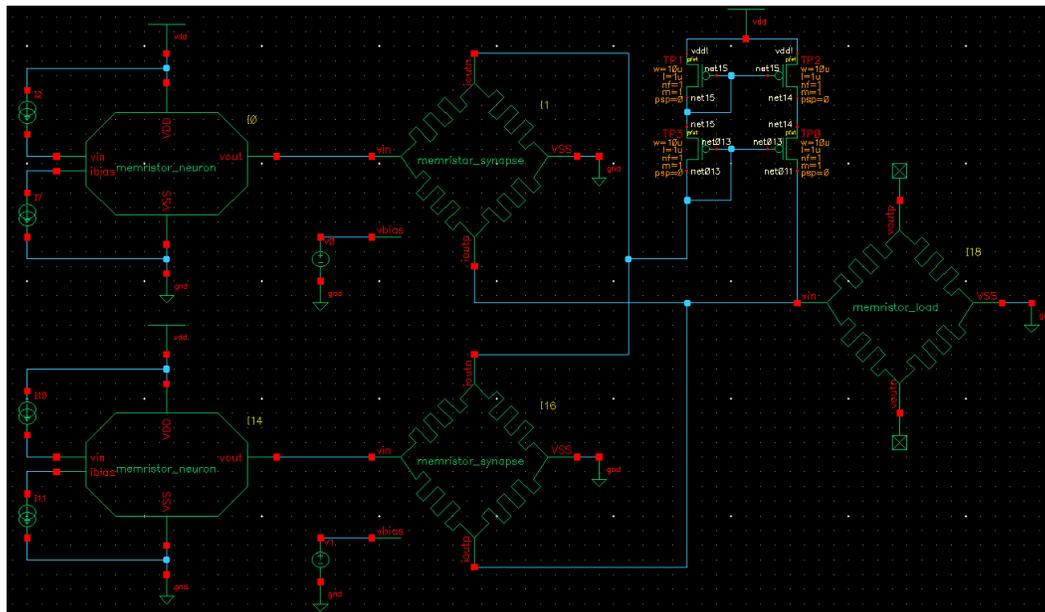


263

264

Appendix Figure 2. Schematic of Neural Network Testbench with Neuron and Synapse

265



266

267

Appendix Figure 3. Schematic of Neural Network to show Associative Learning