University of South-Eastern Norway

Faculty of Technology, Natural Sciences, and Maritime Sciences

—

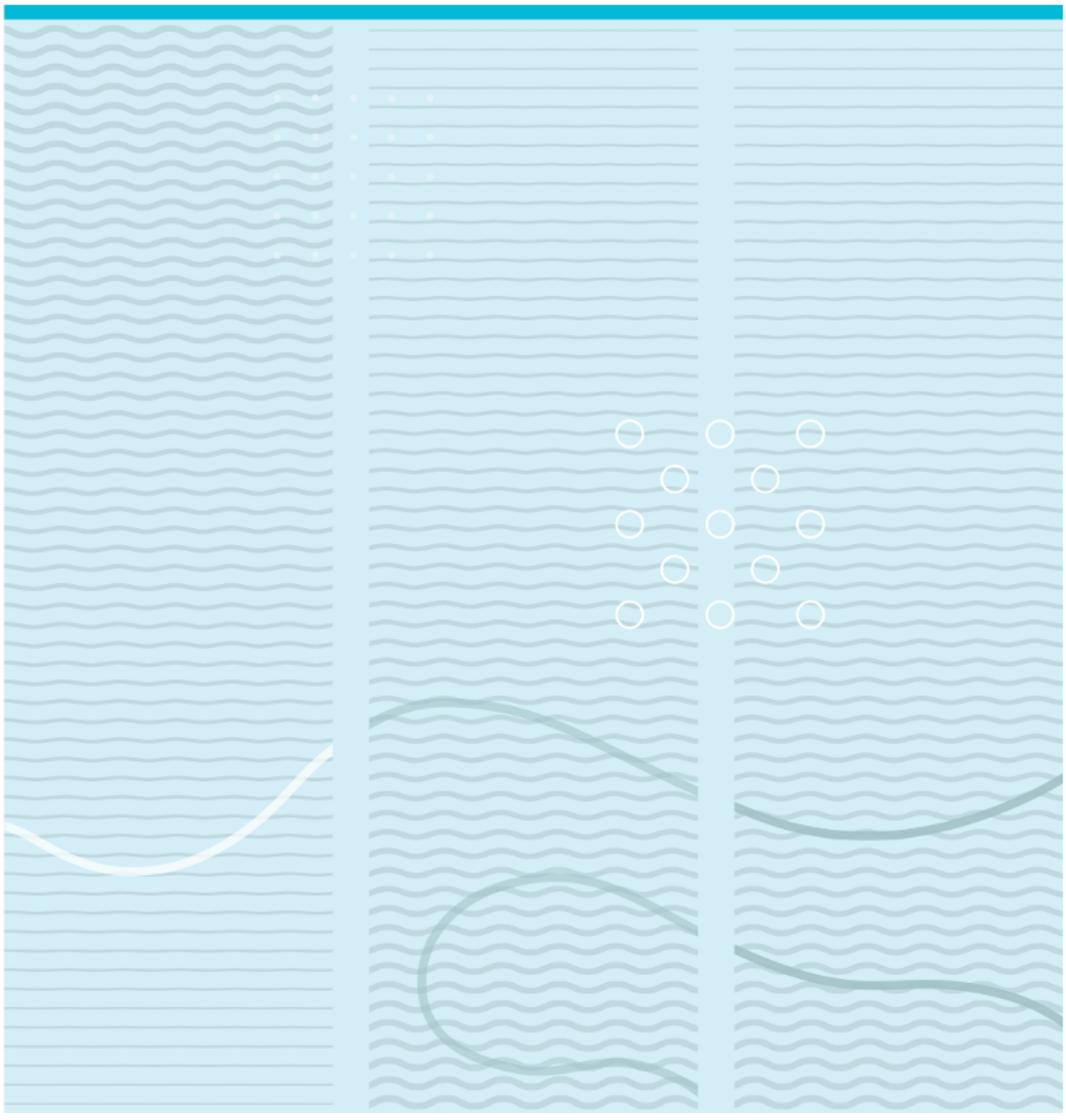
Master's Thesis in Systems Engineering with Embedded Systems

Department of Science and Industry Systems

November 27, 2020

Julian Breivold Nilsen

Memristor Implementation of a Ternary Storage Circuit



University of South-Eastern Norway
Faculty of Technology, Natural Sciences, and Maritime Sciences
Department of Science and Industry Systems
PO Box 235
NO-3603 Kongsberg, Norway

`http://www.usn.no`

This thesis is worth 30 study points

The undersigned have examined the thesis entitled ***Memristor Implementation of a Ternary Storage Circuit*** presented by *Julian Breivold Nilsen*, a candidate for the degree of *Master of Science in Systems Engineering with Embedded Systems* and hereby certify that it is worthy of acceptance.

Date	Head of Department
Date	Supervisor's name
Date	Committee member's name
Date	Committee member's name

ACKNOWLEDGEMENTS

I want to thank the Ternary Research Group at USN containing my wonderful supervisor Henning Gundersen, PhD Candidate Steven Bos who has functioned as my 2nd supervisor, brainstorm buddy, and first author of the paper we wrote together with me; Halvor Nybø Risto who has been a good classmate and fellow Master student in the research group for many hours of idea sharing, interesting cooperation, and friendship; as well as the rest of the research team. I also want to give a huge thank you to Sylwia Winiecka for moral support, encouragement, and for rebuking my awful grammar; Henrik Wallumrød for amazing writing advice, proofreading, and great adventures; Mikael Lindberg and Joshua Bower for companionship and proofreading; Eric Johnson at Knowm Inc for providing data and answering questions on memristor behaviour; and the USN institution for being my home away from home throughout my Bachelor and Masters degree.

I also want to thank Blaise Pascal, not only for his contribution to science, but also for the quote "Je n'ai fait celle-ci plus longue que parce que je n'ai pas eu le loisir de la faire plus courte.", loosely translated too: "I only made this one longer because I did not have time to make it shorter.", a sentiment I feel is all too relevant in this work.

ABSTRACT

This thesis presents two circuits capable of writing ternary data to a memristor, and reading that data without corrupting it. It also investigated how the mean metastable switch memristor model with modified parameters, simulated in LTspice can recreate the behaviours observed in Knowm SDC W/tungsten memristors. How the model can simulate ternary memristor memory, and what limitations the model presents. Initial experimentation on the effect of changes to the parameters of various models were analyzed before the MMSS model was selected, and the parameters were tuned until a behavioral match was validated. Using the model as a tool, development of the first ternary circuit resulted in a circuit capable of writing trits to a memristor and read them without the need of a waveform generator. Knowledge from the design was then applied to the second circuit and simulation results from this circuit was part of the work published in a paper. Analysis of the simulation and breadboard results was then used to show that the MMSS model was good at matching the behaviour of imouts to the Knowm SDC memristor , bot that with the selected parameters it was unsieted for time analysis due to drift.

Keywords

LTspice, memristor, memristor storage, memristor simulation, ternary storage, multi-valued logic, MVL, Knowm, SDC, self directed channel

Contents

1 INTRODUCTION **1**

1 Motivation 1

2 Choosing the Area of Research 2

3 MOSFET & CMOS 3

4 Carbon Nanotube Field Effect Transistor 3

4.1 Carbon nanotube field effect transistor and ternary logic 4

5 Ternary Logic and Computing 5

5.1 How ternary works 5

5.2 The Seturn 7

6 Research question 9

2 Memristors **10**

1 A Glance Into the History of Memristor 10

2 How the Memristor Functions 13

3 Opposition to the memristor 16

3.1 The Missing Memristor has Not been Found 16

3.2 The case for rejecting the memristor as a fundamental circuit element 16

3.3 Final remarks on the opposition 17

4 Real Memristors 17

4.1 HP memristor 17

4.2 Campbells Self-directed channel Memristor 18

4.3 The Knowm SDC Memristors 19

5 Memristors Models 20

5.1 The Joglekar and Biolek Models 20
5.2 Generalized metastable switch memristor model 21
5.3 Knowm mean metastable switch model 21

3 RELATED WORKS 23

1 The Elusive Memristor: Properties of Basic Electrical Circuits 23
2 SPICE Model of Memristor with Nonlinear Dopant Drift 24
3 Self-Controlled Writing and Erasing in a Memristor Crossbar Memory . 24
4 Robust hybrid memristor-CMOS Memory: Modeling and design 25
5 Memristors' Potential for Multi-bit Storage and Pattern Learning . . . 26
6 Self-Directed Channel Memristor for High Temperature Operation . . . 27
7 Implementation and Characterization of a Memristive Memory System 27

4 METHODOLOGY 29

1 Data Sources & Type 29
2 Data Gathering & Analysis 30
3 Justification for Methodological Choices 31
3.1 Choosing Simulation software 32
3.2 The path to selecting a memristor model 32
3.3 The Transition From Sine to Square 33

5 RESULTS 35

1 Parameter Tweaking 35
2 Ternary Storage 42

6 DISCUSSION 48

1 Matching Model Behaviour to the Real Memristors 49
1.1 Hysteresis response 49
1.2 Memristance drift 52
1.3 Reevaluation of hysteresis response 53
1.4 Section summary and last remarks 54
2 Developing a Ternary Memristor Storage Circuit 55
2.1 The first ternary memory circuit 55
2.2 Post Binary: The second ternary memory circuit 56

7 CONCLUSION **57**

1 Future Work 59

BIBLIOGRAPHY **62**

APPENDICES **70**

A Post Binary Robotics Using Memristors With Ternary States for Robotics Control **70**

B SIMULATION AND CIRCUIT DATA **77**

1 Memristor Model 77

1.1 Knowm Mean Metastable Switch Memristor Model 77

1.2 University of Michigan Memristor Model 104

1.3 The Yakopcic Model 109

2 Making the Circuit 118

Chapter 1

INTRODUCTION

This chapter will show the journey to the research question of this thesis, where the journey started, and what lead the thesis in the direction it has developed. The chapter starts by looking at a recent paper written by the thesis supervisor and head of the research group Henning Gundersen. The chapter highlights some questions that needed to be answered to understand the work. The rest of this chapter tries to give a brief answer to these questions and others that might arise without going into too much detail before stating the research goal of the thesis.

The second chapter looks more deeply into the theory of memristors and memristor modeling. Chapter three proceeds to give an overview of the related works found in the body of knowledge, whilst the following chapter highlights the methods used in order to attempt to answer the research question. The fifth chapter presents the most relevant results of the gathered data, before being discussed in chapter six. The final chapter attempts to draw conclusions based on the data and its discussion, as well as propose potential future work.

1 Motivation

The journey started with the supervisor of this thesis and the ternary research group. The research group is part of the university of South-East Norway (USN) and consists

of Assoc. Prof. Henning Gundersen, PhD Candidate Steven Bos, Prof. Nils-Olav Skeie, Assoc. Prof. Radmila Juric, as well as master students Halvor Nybø Risto, Mehtab Singh Virk, and Julian Breivold Nilsen as of 2020. USN has actively taught and researched ternary computing to some degree for over a decade, and the Ternary research group is now spearheading this effort [1].

This has made both past and present work done by the research group [2–4] as well as the previous master thesis [5] a logical starting point for the research.

2 Choosing the Area of Research

The research group is primarily focused on ternary computing. It has gathered resources on the history of ternary, what is going on in the field, and produce new research that will drive ternary in the belief that it will change in the future.

The author of this thesis had to answer the following questions to participate in the research:

- What is a CMOS?
- How does a CNFET work, and how does it enable ternary?
- What is balanced and unbalanced Ternary, and how are they useful?
- What is Radix, and radix economy?
- Is it possible to take advantage of the 1.58 informational overhead?
- Is development with 19683 functions viable?
- What are NTI, PTI, and STI inverters?
- How did the Seturn work, and why was it replaced by a binary solution?
- What are memristors?

Because memristors are the main focus of the thesis, a deeper exploration into memristors and memristor modeling can be found in chapter 2.

3 MOSFET & CMOS

A complementary metal oxide semiconductor (CMOS) is a technology used to make modern integrated circuits. The CMOS consists of complementary metal oxide semiconductor field-effect transistors (MOSFETs) working together, namely the P-type that is open when the input level is low, and the N-type that is open when the input level is high.

The most basic CMOS gate is the inverter and is made up of a single P-type and a N-type MOSFET in series with the output of the gate between the two. A buffer could be made using a reverse setup with an N-type and P-type in series, but due to how MOSFETs work, two inverters are instead used with the input of the second being the output of the first [6, 7].

Complementary metal oxide semiconductors (CMOS) has for the past two decades been the predominant technology in large scale circuits. It is an essential component to make most sectors such as telecommunication, finance, medicine, and several more possible as we know them in modern society.

As technology improves, and the demand for more powerful and efficient computing increase, the limit for scaling down conventional CMOS is starting to hit a wall as we are approaching the nano-scale. Various problems, including high leakage currents, decreased gate control, and anomalous behaviour start to emerge and create obstacles for further down-scaling of the CMOS technology [8].

New technologies are therefore needed to overcome these obstacles. One promising technology to achieve this might be Carbon nanotube field effect transistors (CNFET) [9, 10].

4 Carbon Nanotube Field Effect Transistor

There are several CNFET alternatives that are being, or have been developed. Three general alternatives of the CNFET are:
The back-gated CNFET: This was the first technique of CNFET fabrication. In this

method several carbon-nanotubes (CNs) are deposited randomly over the source and drain [11].
Top-gated CNFETs: This version is somewhat newer, here Arrays of CNFETs can be fabricated on the same wafer, the gate contacts are electrically isolated, but the fabrication of them are complex [12].
Gate-all-around CNFETs: This kind is an improvement to the top-gated CNFET, is has less leakage current and a superior on/off ratio [13].

There has been major development in the last decade, both in manufacturing techniques and the technology itself. A result of this is that in 2019, a modern microprocessor was shown and characterized that is made up of CNFETs [10].

4.1 Carbon nanotube field effect transistor and ternary logic

The voltage threshold of a CNFET is directly linked to the diameter of the CN [2, 9]. This means that ternary can be achieved with a combination of CNFETs with different diameters. [2, 3, 14].

Early designs of ternary circuits utilized large ohm resistors, or suffered from large static power consumption; but newer iterations show novel solutions to overcome that and other issues [9, 15, 16]. One is an inverter, another show one design that use 6 CNFETs to build a circuit where you can access all three ternary inverter functions in one structure. One paper that is of particular interest is the one by S. Kim et al. published in 2018 that proposes a design method for ternary logic gates [14].

There are 19683 possible two input one output ternary logic gates alone [2]. With more inputs, the possible gates make manually designing the circuits an impractical endeavor. S. Lee, S. Kim and S. Kang continued this work in in their 2019 paper [17], automating the design process, although this work was not made open source.

H. Risto in the USN ternary research group published a paper in 2020 ([18]) where he developed an open-source implementation of any N-array ternary-valued CNFET circuit. He shows simulation results of a novel 3 operand, classical 2 operand and a hybrid 1-trit balanced full adder circuit.

5 Ternary Logic and Computing

Ternary logic computing as a base 3 multi-valued logic (MVL) system is an alternative to binary computing. Where binary logic consists of two states of 1s and 0s, true, and false; ternary and MLV systems have more states, which can potentially save storage elements and wiring as more data can pass over a single wire.

The main drawback, and one of the reasons binary has dominated, is that with conventional CMOS transistors, the increased complexity to a system introduced by adding a third state or other MLV radix does introduce more complexity to the hardware and can cause it to takes up more die area and be less efficient.

The drive and push behind ternary and MVL research is the belief that the benefits can, and will outperform the added complexity. If CNFETs or another technology can reduce that complexity, there is a potential 58% informational overhead with ternary over binary [2]. The calculation behind the overhead is shown in figure 1.1.

$$\frac{log_3}{log_2} = 1.58496250072 \tag{1.1}$$

The argument for ternary logic systems over higher radix MVL alternatives, is that ternary is the closest discrete number system to the mathematically optimal e. One thing worth mentioning is D. Etiembles paper from 2019, where they presented a counter-claim, and attempted to prove why R=3 is not the optimal radix for computing [19]. The paper concludes that ternary circuits have been, and probably will be a small niche. Citing his own article published in 2003, H. Gundersen made a counter argument in their paper [2], that since it is possible to build a modern microprocessor with complementary CNFETs (as shown in Hills paper [10]), it is only a matter of time and research before it is possible to build a ternary microprocessor.

5.1 How ternary works

At its core any system that uses base 3 radix is ternary. In ternary logic however, there are mainly three dominant forms that is used; a truth value, balanced ternary (and it's

short hand form), and unsigned; these can be seen in table 1.1.

Truth value	Balanced	Short form	Unsigned
True	+1	+	2
Unknown	0	0	1
False	-1 (T)	-	0

Table 1.1: Ternary values in truth value, full numeric form, in short hand form, and in unsigned notation.

Unbalanced ternary simply use 0,1, and 2. unbalanced is arguably easier to read, but it needs a sign bit for negative numbers and addition is more complex when compared to balanced ternary.

Balanced ternary utilizes +1 0 and -1 ((1, 0, T and -, 0, + is also used) [20]. Counting in balanced ternary (-5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5...) using the short hand is done as follows: -11, 0–, 0-0, 0-1 00-, 000, 001, 01-, 010, 011, 1– ,and so on; this system makes it possible to add both positive and negative numbers without having to use a sign bit (-2+4 would be 0-1 + 011 = 01- with is indeed 2) [2].

In ternary there are 27 (3 trivial) monadic, and 19683 diadic operators countra 4 (2 trivial) monadic and 16 diadic operators in binary [20]. When the inputs are increased to three this increase to over 7 trillion (7.6e12) operators in ternary. This does give trenary a lot of special operators that can be incredibly useful in special circumstances, but the design of these can be tedious and impractical, as well and introducing a need for a system to refer to each function without causing confusion. H. Ristos paper from 2020 proposed a indexing system that makes it possible to reference any ternary-valued logic function unambiguously, as well as the already mentioned generation of design for any N-array ternary-valued CNFET circuit ([source] Halvor!). If we look at the most used and useful monadic operator used today, the inverter, it is quite simple in a binary as is simply turns 0 into 1 and vise versa; but when a third state is introduced things become more complex.

The ternary high state can be set low and vice versa, but the middle state has no

a) STI		b) NTI		c) PTI	
A	**Y**	**A**	**Y**	**A**	**Y**
-	1	-	1	-	1
0	0	0	-	0	1
1	-	1	-	1	-

Table 1.2: This table shows what output each balanced ternary inverter will output (Y) based on input (A): a) Simple Ternary Inverter (STI). b) Negative Ternary Inverter (NTI). c) Positive Ternary Inverter (PTI).

obvious state to be set after the operator, it can be left where it is in the mid state, it can be set low, or it can be set high. This means that there is three different monadic operators that are considered inverters, the simple ternary inverter (STI), the negative ternary inverter (NTI), and the positive ternary inverter (PTI) [3]. The truth table for each inverter can be seen in table 1.2.

5.2 The Seturn

The Seturn was a ternary computer developed in 1956 and spearheaded by S. L. Sobolev and N. P. Brousentsov in Russia. Although it was developed by a small team of young men, the first Seturn was built in 1958, after only two years. Brousentsov credits this partly due to the superiority of working with a ternary system [21, 22].

Two years after this, in 1960, they had developed enough programs for the Setun to present the machine for official testing. Brousentsov reports that the results of the testing was very successful, and that the computer performed incredibly reliably and stably in a wide range of temperatures and voltages, when compared to other systems of the time [21].

The system used 4 state switches to store 3 states, and although a new modern ternary system need to be able to store trits in a more optimal manner to become a real contender with its binary counterpart, there is a lot that can be learned from the Seturn project [4].

The Seturn was a sequential computer using balanced ternary with a fast multiplier

thanks to parallel operations. It has a small 3 page 54 word ferrite ram that had page exchange with a magnetic drum cach memory. It has a one address architecture with one index register, and it uses the polish inverse postfix notation system [21].

It had 24 instructions (addition, multiplication...). as a result of balanced ternary, it had no need to use a sign bit, this resulted in a decreased need of conditional instructions and of the 24, 3 was in reserve and ended up not being needed [21].

Ternary also allowed free variations of the operand length, allowing them to be different lengths and still be combined; the ideal rounding when applicable was achieved simply by truncation

The content of the index register could, dependent on the value (+, 0, -) of the modification trit, be added or subtracted from the address part of the current instruction [21].

The ferrite core semiconductor diodes where designed due to an inaccessibility of transistors at the time, and due to vacuum tubes having a too low reliability. The cores functioned as controlled current transformers, making them effective for ternary threshold logic [21].

The reasons Brousentsov gives for making the Seturn ternary was that ternary threshold logic elements was that, compared to its binary counterparts they; required less power, were faster, more reliable, required less equipment, [21].

The Setun was intended to create a small, inexpensive computer that was simple to use; for schools, research laboratories, and such. [21]

In 1960 Sobolev left the team, he was the member with influence in the Russian academic circle. Brousentsov attempted to keep the project going, but he did not have enough influence in the soviet system and it started to fall apart [22]. The USSR had a negative a attitude to "non-planed" projects in general, especially projects they deemed "fruits

of university fantasy" [21].

There where many orders of the Seturn, including international export orders, but due to the politics, and the attempts to annihilate the project from the USSR, only 10-15 where produced annually, and none was exported. There was also plans to increase the production in Czechoslovakia, but the plans where foiled before they bore fruits [21].

Before being finally in 1965 when the manufacturing was completely shut down by the USSR 50 units was produced. It was shut down despite multiple unsatisfied orders. The binary computer that replaced the Seturn had comparable performance but cost 2.5 times as much to manufacture [21].

6 Research question

After this brief look into the topics researched by the ternary research-group. It was decided early in the process that this thesis would focus on reading and writing ternary data to a memristor. The main motivations for this was the possibility to use the ternary memory in robotics, the fact that real memristors was available to perform test on, and the fact that another member of the research group was writing a thesis focusing on CNFET implementations for ternary logic and radix conversions.

The research questions this thesis attempts to answers therefor evolved to be:

- How will the memristor simulation compare to real memristor behaviour?
- What are the limitations and strengths of the memristor model explored in this work?
- Can current memristor models be used to simulate ternary, memristor based memory?

Chapter 2

Memristors

1 A Glance Into the History of Memristor

The memristor was envisioned to be the forth fundamental circuit element as shown in figure 2.1. Just as a circuit element where the there is a direct link between voltage (v) and current (i) where a change to one cause a corresponding change in the other is a resistor; An element where the voltage (v) and charge (q) is linked is a capacitor; And an element where the relationship is between flux (ϕ) and current (i) is an inductor; L. Chua inferred that there had to be a missing link between that had a direct relationship between flux (ϕ) and charge (q) [24, 25].

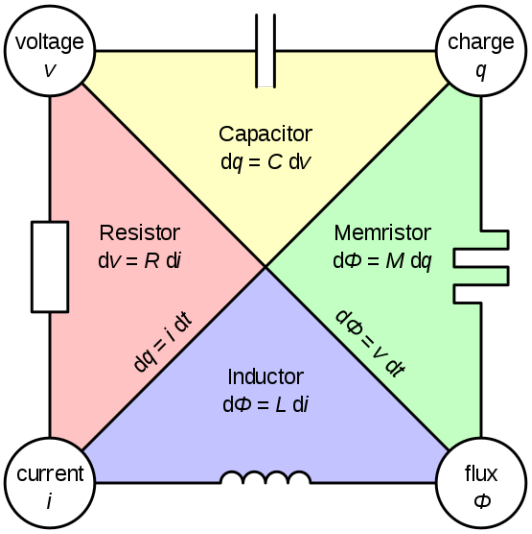


Figure 2.1: Figure owned by Parcly Taxel 2013 [23], it shows the relations between the four fundamental electronic variables and the devices that implement these relationships. This file is licensed under the Creative Commons Attribution-Share Alike 3.0 Unported license, and no changes was made to it.

According to Chua the relationships mentioned does not need to be a linear relationship, although a simple ohmic resistor is linear there are non linear resistors. An illustration

of the various kinds can be seen in 2.2, a varistor is such an example and its IV-curve can be seen in figure 2.3. Other examples include vacuum diodes, semi conductor diodes and a host of others that have this IV relationship [25], some of which can be seen in [25] from 14:25.

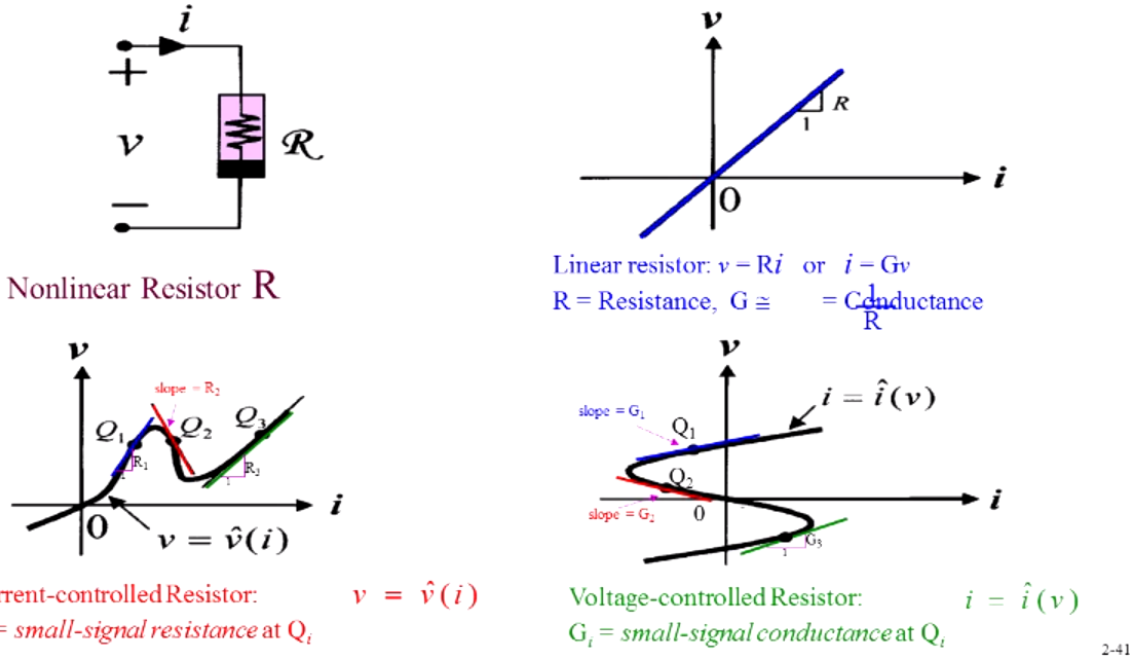


Figure 2.2: The figure shows the circuit symbol for a non-linear resistor, the IV-curve of a linear resistor, a current-controlled resistor, and a voltage-controlled resistor. This image is taken from [25] at timestamp 14:02

Chua stated this realisation came to him when after he spent two years studying the vacuum tube. Then after he graduated, and he he went to the Massachusetts Institute of Technology (MIT) and there was 20 new devices.

L. Chua asked himself: "how will I have time for all of these devises all though I spent two years learning vacuum tube" and he came to realise that it is not needed to learn them all individually because they are all various non-linear resistors, and there is a common procedure to understand and analyze them [25].

The same is true for capacitors and inductors as well. Both have linear and non linear versions, but for a capacitor it would be a voltage-charge(VQ)-curve, and for the inductor it would be a flux-current(ϕI)-curve [25].

An extension of this logic and the realisation that there was a missing connection in figure 2.1 between charge and flux; Chua realised that, like how the first periodic table from 1869 had missing elements like Gallium and Scandium that needed to exist even if they where yet to be discovered, there was a need for a two terminal passive element with this flux charge relationship for the symmetry to be complete [24, 25].

All every real-world capacitor, inductor, and battery have some trace resistance, all resistors have some inductive part, and text books state that all devices can be represented by only using the three ideal representations of R C and L. But with the postulation that the ideal memristor is the missing link between flux and charge, the trio might not always be sufficient to model all real-world circuits because some may (and will) have memristive components as well. Chua goes as far as state that this is proven several well known circuits such as the Hodgkin- Huxley cells, Josephson junctions, and ionic transport in neurons [26–28].

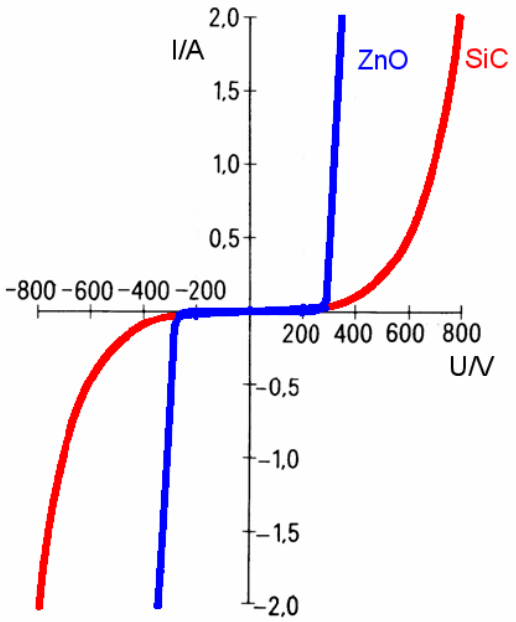


Figure 2.3: The figure shows the IV-graph of a zinc oxide and a silicone carbide varistor, also known as a voltage-dependent resistor (VDR). The image is released into the public domain by Journey234 in 2007.

Although, at the time of Chuas paper in 1971 a passive physical device had yet to be found, he did present a laboratory realisation of a memristor using active elements and an internal power source. diagrams of this can be seen in [24].

In the 3 chua lecture [28], Chua states that in the paper released the same year he released [24], a Japanese paper by M. Kikuchi, M. Saito, H. Okushi, and A. Matsuda was published [29]; and that he did not learn about this particular paper until after 2008, and that in this paper they describes a memristor, but that the authors did not understand what their discovery ment and the device was not identified as a memristor [24, 29].

It was not until 2008 that a paper was published by D. B. Strukov et. al. [8] a device was found and identified as a memristor.

2 How the Memristor Functions

This section will give a quick look into the general equations that govern memristors, what types of memristors are expected to exist, and present the genealogy of memristors.

The definition of memristance and what memristance is, is postulated by the following equations for a charge controlled, it starts with the flux charge plane:

$$\phi = f(q) \tag{2.1}$$

Then both sides are derived with respect to time, this give:

$$\begin{aligned} v &= \frac{d\phi}{dt} \\ &\equiv \frac{df(q)}{dq} \cdot \frac{dq}{dt} \end{aligned} \tag{2.2}$$

Where:

$$\frac{df(q)}{dq} \equiv M(q) \tag{2.3}$$

And:

$$\frac{dq}{dt} \equiv i \tag{2.4}$$

Which leaves:

$$v = M(q) \cdot i \tag{2.5}$$

Where "v" is voltage, "M(q)" is memristance, and "i" is current.

What can be taken from this is that to calculate the memristance the charge (q) and the past history of the device needs to be known. In other words, the memristance is defined by a state-dependent ohms' law. It also shows that the voltage is the derivative with respect to time of the flux (and therefor the flux is the integral of the voltage with respect to time), and the charge has the same mathematical relationship with the current [8, 24].

The same breakdown of a flux controlled memristor can be done as in equation 2.2-2.5 where the the charge flux plane:

$$q = f(\phi) \tag{2.6}$$

And the following is left:

$$i = G(\phi) \cdot v \tag{2.7}$$

Where "G(ϕ)" is memductance.

The Equations above do however describe ideal memristors only, but Chua has described a genealogy of memristors, where each tier is part of the larger one. All memristors fall under what is dubbed "extended memristors", equations for the expanded memristor class can be seen in equations 2.8 and 2.9; inside the extended memristor set the "generic memristor" set can be found in equations 2.10 and 2.11; again, the "ideal generic memristor" set can be found within the generic memristor set, equations for these are 2.12 and 2.13; and the final subset found in the center is the "ideal memristor", there equations are 2.7 and 2.5; an illustration of this can be seen in figure 2.4.

Extended charge controlled memristor:

$$\begin{aligned} v &= M(x,i) \cdot i \\ & \quad M(x,i) \neq \inf \\ \frac{dx}{dt} &= f(x,i) \end{aligned} \tag{2.8}$$

Extended flux controlled memristor:

$$\begin{aligned} i &= G(x,v) \cdot v \\ & \quad G(x,v) \neq \inf \\ \frac{dx}{dt} &= g(x,v) \end{aligned} \tag{2.9}$$

Generic charge controlled memristor:

$$\begin{aligned} v &= M(x) \cdot i \\ \frac{dx}{dt} &= f(x,i) \end{aligned} \tag{2.10}$$

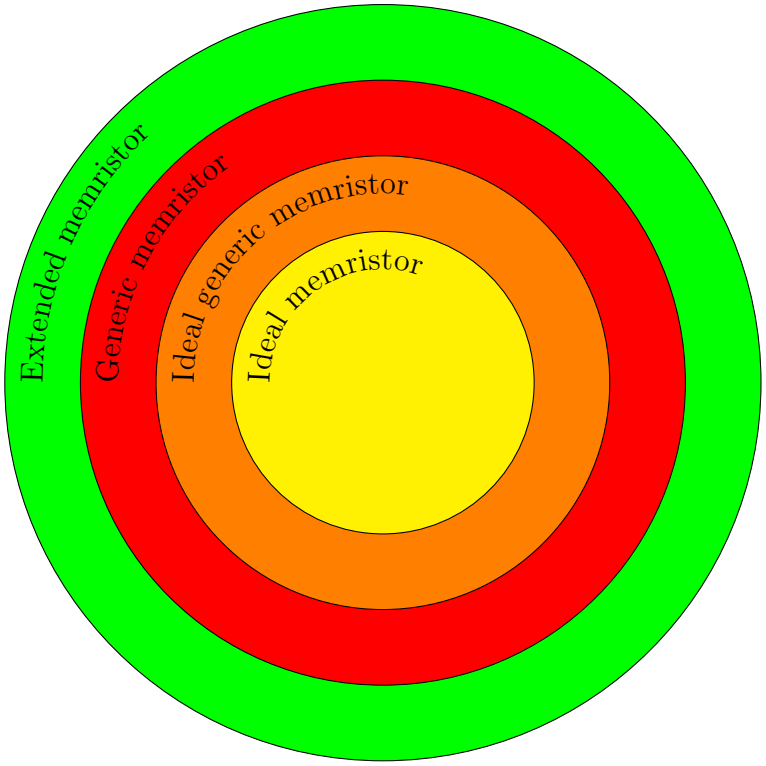


Figure 2.4: The diagram visualizes the memristor genealogy as it was dubbed by L. Chua in [30].

Generic flux controlled memristor:

$$\begin{aligned} i &= G(x) \cdot v \\ \frac{dx}{dt} &= g(x, v) \end{aligned} \tag{2.11}$$

Ideal generic charge controlled memristor:

$$\begin{aligned} v &= M(x) \cdot i \\ \frac{dx}{dt} &= \hat{f}(x) i \end{aligned} \tag{2.12}$$

Ideal generic flux controlled memristor:

$$\begin{aligned} i &= G(x) \cdot v \\ \frac{dx}{dt} &= \hat{g}(x) v \end{aligned} \tag{2.13}$$

Unlike a capacitor or an inductor who stores energy, a memristor stores information. If

an inductors magnetic field or capacitors chard field is charged up and a switch with a light-bulb is switched, the bulb will light up until the energy is discharged, this is not the case with a memristor where no energy is stored [28].

3 Opposition to the memristor

There are oppositions to the memristor for various reasons. This Subsection will look at two of these and what arguments they have with the memristor.

3.1 The Missing Memristor has Not been Found

The first is presented in the paper published in 2015 by S. Vongehr and X. Mengs [31] an argue that the device presented in 2008 [8] is not the the hypothesised device from 1971, claiming that such a device requires magnetic flux which is lacking in the claimed memristors; and that such a device is still missing and in fact most likely impossible.

The paper draws a parallel to gravitational waves and their claimed detection by Joseph Weber in the late 1960s that was discredited in the mid 1970s, siting a calculation by Garwin that if Webers detections where real, the universe would have converted all its energy into gravitational energy in 50 million years.

Coincidentally enough, although Weber's detection was not correct, gravitational waves were in fact detected in 2016, one year after Vongehrs paper was published [31, 32].

Vongehr does admit in the paper that the device discovered might well be a breakthrough in building artificial brains, but that this hype hinders the discussion on whether the device is in fact the missing circuit element and that this justifies the papers narrow focus [31].

3.2 The case for rejecting the memristor as a fundamental circuit element

The second paper was published in 2018 and written by I. Abraham [33]. The paper states it investigates the claim that the memristor is the forth passive circuit element with the fresh perspective that: electrical engineering is the science of charge management. And states that the ideal memristor as describes is an nonphysical, active device; and that any "real" memristor is simply a non-linear composition of resistors with active

hysteresis.

The paper goes quite deep into the mathematics of memristors as postulated by Chua [24] and its own interpretation of the concepts and the physical device presented in [8]. It also states that it proves there is only three fundamental passive circuit elements.

The paper methodical, advanced, and comprehension off it is well beyond the scope of this thesis. But even Abraham in this paper states: "The memristor's potential for phenomenal computing is in no way diminished by this negative assessment of its qualifications as a fundamental device." [33].

3.3 Final remarks on the opposition

Chua does address points made in both these papers in his lecture series [25, 27, 28, 34], and he does inspire confidence with his font of knowledge on the subject. And no doubt the debate will carry on in the community. But regardless the utility and potential of the "memristor" can not be denied!

4 Real Memristors

Many new memristors have been introduced in the last decade and a half, and there are more still documented before 2008 that have later been identified. Most solutions approach the memristor as a pure circuit element, and suggest that the memristor can solve many modern problems, then there are others like the one described in described by K. Liu et.al. that mechanically propels objects and might be crucial in future artificial muscle devices.

In this section however, the exploration will be limited to the three considered most relevant to this thesis. The HP memristor for its historical importance, the self directed channel (SDC) memristors described in [35] by the inventor of the SDC, and the SDC memristors sold by Knowm because it is the memristor used in this thesis and because it is commercially available.

4.1 HP memristor

The HP memristor was presented in 2008 [8]. Although there where devices described before this that in later years have been proven to be memristors it was the first

memristor, The HP memristor was the first to be discovered and identified as a memristor.

L. Chua defends this device as a memristor in his lectures [25, 27, 28, 34] as well as in some of his papers [36].

The device described consists of two thin layered TiO_2 films roughly 10nm in size sandwitched between platinum contacts. One of the layers is oxygen doped vacancies and behave as a semi conductor, the other is undoped and has insulating properties. The boundary between the two move depending on electric current that causes a change in memristance [8].

The final equation given in [8] that gives its memristance is:

$$M(q) = R_{OFF}\left(1 - \frac{\mu_V R_{ON}}{D^2} q(t)\right) \tag{2.14}$$

A simpler equations from [37] is presented as:

$$\begin{aligned} M(x) =& R_{ON}x + R_{OFF}(1-x) \\ Where\, x =& \frac{w}{D} \in (0,1) \end{aligned} \tag{2.15}$$

4.2 Campbells Self-directed channel Memristor

The memristor Campbell described in her paper [35] is a physical SDC memristor with $Ge_2Se_3/SnSe/Ag$ layer composition, the strongest advantages this class of memristors have according to Campbells paper over over other ion-conducting devises is that they offer continuous operations up to $150C^o$, and that no photo doping is required during fabrication.

The device changes memristance by the movement of AG+ ionn, the ammount of Ag within channels determine memristance, and can go as high as 1MΩ, cut high currents will irrevocably lower this level [35].

The paper states that the device is simple to fabricate, and only commercial sputter tools are required for all device layers including the top electrode. It also states that the device can withstand at least 1 billion cycles both at room temperature, and $140C^o$.

The device described falls in the "Generic memristor" class in the memristor genealogy seen in 2.4

4.3 The Knowm SDC Memristors

Knowm inc. sells various memristor solutions, chips and tools to use, experiment, and develop with them. The memristors are all SDC type memristors invented by Kris Campbell [38]. They offer chips with resin as well as options encased in ceramic. The memristors can handle the same temperatures and offer comparable cycle durability to that of Campbells device, but states that the resin options are not suited for high temperature.

Knowm offers 4 variations for sale, the compositions of each can be seen in 2.5, each variation offers distinct strengths and limitations with respect to each-other, these are:

- W/Tungsten: Analog state retention with modest/fast switching response [38, 39].
- C/Carbon: Reduced analog state retention. Very low switching energy. Fastest switching response [38, 39].
- Sn/Tin: Analog state retention. Medium switching response [38, 39].
- Cr/Chromium: Analog state retention. Highest thresholds. Slowest switching response [38, 39].

The W/tungsten variant is the memristor used in this thesis to recreate the I-V plots, as well as the breadboard implementation used in the included paper [40]. The memristor was chosen because of its modest retention quality while still presenting fast switching speeds [38, 39].

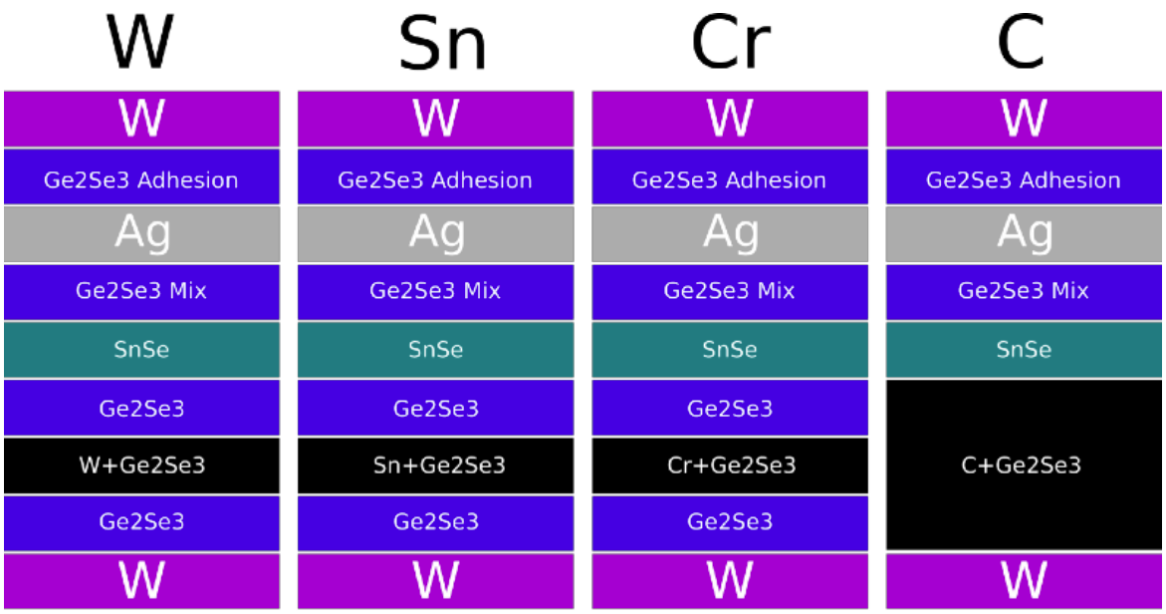


Figure 2.5: The figure shows the layer composition of each memristor dopant variation they produce and sell. The figure is taken from [38] page 3.

5 Memristors Models

This section discusses various memristor models, if and were they have simulation implementations, and will provide a general explanation of their behaviour. [37, 40–45]

5.1 The Joglekar and Biolek Models

Three LTspice model based on the Joglekar paper from 2009 [26] is described in [46] and can be found at [44]. The first, referred to as Joglekar was created as part of writing [37], with the Joglekar 2 and Biolek models being modifications and improvements of the first.

They are basic memristor models, and though it is not ideal at recreating real memristor behaviour [46], and described as a "hacked together" model [45, 47]; it is good first step when getting familiarized with memristor modeling in SPICE, and understanding the mathematics of the models is useful, or even necessary to develop better models.

The models have a tutorial on how to implement them in LTspice on the Knowm.org blogs [48, 49].

5.2 Generalized metastable switch memristor model

This model was created as a tool as part of a paper by M.A. Nugent and T.W. Molter, it was published in 2014 [50].

The method models memristors by simulating a collection om idealized two-state switched that change as a function of applied voltage and temperature. The conductance of the memristor is then given as the sum over all switches [49, 50].

No simulation tool implementation of the model has been found during the this work, but it is the basis of the model used in this paper. That model is described in the following section.

5.3 Knowm mean metastable switch model

The Knowm mean meta-stable switch model, is a continuation of the generalized model described above. It was first first adopted to the Xyse simulation software, were it was adapted in 4 ways [41]:

- Define on and off conductances as resistances
- Define the two main resistance levels as On and Off rather than A and B
- Define the main state of the device as X [0,1], rather than N_{off}. It's the same but it's scaled to one and is continuous.
- Define the state, X, in terms of dX/dt.

Because of complications regarding Gaussian distribution sampling and constraints of the Xyce simulation engines, the model is reformulated using the mean of the Gaussian sample. This makes it possible to define the state in terms of dX/dt [42].

The mathematics of the model is desctibed in the blog post, in in short changes states as a probabilistic function of the applied voltage.

The current through the device is then given through Ohms law $I = VG$, where $G = \frac{X}{R_{ON}} \frac{1-X}{R_{OFF}}$ [41].

A last note on the model is that the blog post states that GMSS model uses both a memory-dependent current component (MSS), I_m , and a Schottky diode current, I_s in parallel. The blog post shows how this is added to the Xyce model, though there seem to be no such function in the LTspice implementation of the MMSS model.

Chapter 3

RELATED WORKS

This chapter will give a brief overview of the most relevant related works to this paper. The papers discussed here will be investigated in the ascending order of publication.

The papers discussed in this chapter is in no way an exhaustive list of relevant work. Far to many papers deserve a spot in this chapter, so only most relevant papers, those that has given special insights, or given rise to interesting or innovative future work can therefor be included so the chapter remains at a reasonable length in respects to the other chapters.

1 The Elusive Memristor: Properties of Basic Electrical Circuits

One of the pioneer papers on memristor modeling was published in 2009 by Y. N. Joglekar and S. J. Wolf. [26]. The paper starts by stating it presents a tutorial on the properties of memristors.

In it's introduction it goes on to give a brief explanation basic circuit theory, memristor history, and why memristors are needed before moving on to the discovery of Strukov et. al. [8] discovery at HP labs of a real memristor device.

The paper goes on to discuss the device presented in [8] before attempting to analytically derive its I-V relationship. Next it explore memristor-capacitor, and memristor-inductor circuits using the linear drift models presented in [8].

In section 4 they present models that characterize memristors as dependant on dopant drift, and implement more complex models they characterized as more realistic then the ones previously shown. The fifth section discuss how a memristor in place of a resistor in a RCL circuit far more tenability, and that the circuit can be over-dampened or under-dampened depending on the polarity of the memristor.

2 SPICE Model of Memristor with Nonlinear Dopant Drift

Later that year Z. Biolek et.al. published a paper [37] that built on the work of Joglekar [26] and Strukov [8]. The paper starts by summarizing the device of Strukov and the model Joglekar presented. It then goes on to introduce a SPICE model that is based on the summery. It demonstrates the presented SPICE model with simulation results. Finally, it discusses general problems and boundary effect issues.

3 Self-Controlled Writing and Erasing in a Memristor Crossbar Memory

The paper by I. E Ebong and P. Mazumder [51] from 2011 introduce a method that is advantageous to memristor memory systems. The data is simulated both by hand with a lumped model, and with nano-wire modeling with a distributed pi-model.

The paper lays out some problems with resistive memory, such as resistance drift, non linearity across devices, and more.

The paper states that the problems are addressed by using correcting pulses to mitigate resistive drift, using a temperature-compensating circuit to counter issued caused by temperature variation, writing with pulses and adaptive reads to counter non-uniform memristance profiles, and using diodes to reduce leakage in the crossbar array.

Due to memristor non-uniformity, the solutions read may be destructive to the data stored, therefor it implements a data refresh after every read cycle that is built into the read function when necessary.

For the same reason, due to device variation, the write schema writes in several pulses, nudging the memristor until it is in the correct state. This causes devises with a large δ between high and low memristance to require more pulses then the one with a lower δ.

The proposed system also handled defective devices, classifying them as either: stuck open, stuck closed, and the lower or upper bound memristance targets are not met (on account of the system not working with memristive values directly but the ratio between upper and lower bound levels of each device). The first two are easy to identify as a pulse in each direction will result in the same logic level, causing the system to not waste more cycles on that device.

The papers simulation results show that the system is vulnerable to leakage currents in the crossbar array, and states that one solution might be adjustable reference resistors.

The biggest drawback of the method for ternary applications is that it depends on writing the memristors to extremes, therefor it can not be directly used for multi-bit (and by extension trit/multi-trit) memory. But modifications to the method can be devised to allow this.

4 Robust hybrid memristor-CMOS Memory: Modeling and design

The paper was written by B. Mohammad et. al. and published in 2013. The aim of the paper is to study the reported device behaviour of mathematical memristor models in depth using matlab to compare, analyze stability impact of various device parameters, create a SPICE model to match existing methematical models, and propose a CMOS-memristor memory cell with minimal impact on device stability.

One finding of the paper of particular interest to this work is that the in the models

explored there are trade-offs that need to be considered, the paper reports that models with stable read behavior tend to have large delays. Finally it highlights the issue of current (at the time) memristor models did not agree with at the time limited experimental data, and that they intend to continue work on a future model to bring it more in line with reality.

5 Memristors' Potential for Multi-bit Storage and Pattern Learning

In 2015 N. Taherinejad et. al. published a paper called "Memristors' potential for multi-bit storage and pattern learning" [52].

The paper unique property of the memristor that allows it to store multi-bit data with a focus on the advantages this brings to pattern learning applications.

The paper makes a hypothesis that: "Not only the resistance of memristor depends on its current state, but also the changes of resistance depends on its current state", and provides a simplified proof of the claim concludes that the memristance change due to the same pulse at two different states is proportional to the ratio of the memristance at the respective points.

An experiment to test this is to take a memristor and apply a positive pulse (representing logical 1), then a negative pulse (representing logical 0) will yield a different change in memristance then if pulses of the same size, but this time apply the negative pulse first and the positive second.

The paper states that this behaviour can then be used to store multi-bit values because every sequence of inputs would lead to a unique end state in the memristor, in other words none of the input patterns "00", "01", "10", or "11" would not leave the memristor in the same state as any of the other input patterns, thereby making it possible to store multiple bits in one memristor, and decode the state when read.

It is worth noting that some of the final states achieved in the simulation are very close,

such as most significant bit (MSB) "010" and "100" with 0.5V and -0.25V pulses of 0.1s gives states on 4975Ω and 4974Ω respectively.

In the least significant bit (LSB) case, these close states are instead adjacent, the same end states (4974Ω and 4975Ω) would result from the patterns "001" and "010" respectively. In some applications such as pattern learning, the eventual error might not be a big deal and small errors can be tolerated.

6 Self-Directed Channel Memristor for High Temperature Operation

In 2017 K. A. Campbell published a paper describing the characteristics of the physical SDC $Ge_2Se_3/SnSe/Ag$ device, states strengths of the device and discuss advantages this devise have over other ion-conducting devises.

The paper goes on to discuss how the memristor operates, how the memristor is physically structured, and compare fabrication to that of CBRAM devices.

The paper then goes on to discuss experimentation, detail some experiments, show experimental results, and discuss these results. This part of the paper is particularly relevant to this master, as some of these experiments are recreated with the memristor model in the thesis, particularly the pulsed response, and write endurance experiment. Though endurance is not relevant to the research in this thesis, the memristor response and the data presented is of most interest.

7 Implementation and Characterization of a Memristive Memory System

In 2019, D. Radakovits and N. Taherinejad published two papers together, the first in May with Radakovits as the first author with the title "Implementation and Characterization of a Memristive Memory System" [53].

The paper presents a method of writing and reading data to and from memristors. It states that the method used to read eliminates the need for read compensation or

refresh due to corruption of the stored data from read reported in earlier work. It goes on to detail experiments used to evaluate the method, before going on to discuss the results [53].

To read the circuit sends a pulse over two parallel resistors were one is in series with the memristor and the other in series with a reference resistor. A operational amplifier then compares the levels and reports if the memristor is larger or smaller in resistance then the reference resistor [53].

This approach will not affect the state of the memristor if the memristor threshold level is higher then the amplitude of the pulse, this is device reliant however and needs to be accounted for [53].

Chapter 4

METHODOLOGY

This chapter will highlight the sources and types of data, provide methods used to acquire the data and explain the approach taken to analyze the data; as well as give a justification for the selected approach.

1 Data Sources & Type

The data used in this thesis was gathered from simulations and experiments on real circuits, this data is displayed mostly in chapter 5: Result of this thesis, appendix B, the github project found at [54], and A. Another source of the data is other work in the body of knowledge, data received from other researchers, and data generated by the ternary research group.

The simulation data is gathered from LTspice in the from or plot figures.

The breadboard data was created using the Knowm M+SDC memristor 16 discrete 32 DIP W/Tungsten chip, the Analogue Discovery (1) with the Waveform software was used to generate sine waves, collect data from voltage readings, perform calculations, and export the data in .CSV format (this data can be found at [54]). The data was processed and plotted using Matlab R2019b.

The data analyzed comprises graphs, plots, and other figures, as well as descriptions collected from the body of knowledge and data generated by the ternary research group; this data will be referred to as "external" data; and, finally, figures generated through simulation and experimentation which will be referred to as "internal" data.

Although the data collected with the Analogue Discovery can be said to be quantitative and is available to any interested party, it was collected and used in order to generate figures for comparison with simulation plots and other qualitative data; therefore all data presented in this thesis is considered qualitative.

The external data, containing mostly figures and descriptions of results and memristor behaviour, as well as the internal data from the breadboard experiment were analyzed by attempting to recreate the experiments in LTspice, then compared the simulation graph with the corresponding breadboard or external graph.

2 Data Gathering & Analysis

The first step of the data analysis involved comparing real memristor I-V graphs to graphs of the corresponding LTspice memristor models from [44] which were simulated with a recreation of the real circuits. After the graphs from each model were compared to the external data, the parameters controlling the model behavior were tweaked in an attempt to match the behaviour, this was repeated with each of the LTspice model described in 5 until an acceptably close match was achieved, or enough attempts were made to justify moving on from that model.

When a model was chosen and an approximate match was found, the model with the new parameter setting was used in a LTspice circuit that attempted to recreate results from other papers (an example is the experiment in [35] shown in figure 6.5). If the analyses comparing the simulation and the presented results did not match, the parameters were tuned in iterations until the model behaved as expected. It was then verified with its I-V circuit, and that it still behaved correctly in that respect too.

After the analysis concluded that the model was behaving as intended when subject

to the same inputs as the real world analogue results reported in related works, the next step in data analysis was to start designing a ternary simulation Circuit, recording interesting findings (included in appendix B from B.47), and eventually comparing the results of this with the states and behaviours identified by the ternary research group that eventually were published in [4].

The following work, up until the paper in appendix A were iterative improvements, the result of this is mostly documented in the results chapter and in appendix B, changes were documented and analyzed along the way.

Finally, other than the recreation of some figures in order for the scale to be more appropriate, they needed to be easier to read, and so on. The last data generated as part of this work was in collaboration with S. Bos as the first author of the paper: "Post-Binary Robotics: Using Memristors With Ternary States for Robotics Control" [40]. This started with replication of the binary controller from [53], another voltage level was added, another op-amp to form a window comparator for ternary output with a voltage regulator to shift the output to only positive values, as the Arduino cannot accept negative values. Lessons gained from analysis of the data gained so far in this thesis were then used to make changes to the circuit and make it function well in a simulation.

The data analysis in the paper consists of comparing LTspice results and breadboard implementations with one another, as well as to the results from [4, 35].

3 Justification for Methodological Choices

This type of data and method used to analyze it was chosen because it would facilitate experimentation and practical experience with the software and devices used, an exploration of the body of knowledge, and a path to understand the topic while gathering data; and because available open source quantitative data, tools required to produce quantitative data were not easily available to the author early in the process, and the required understanding required to properly perform quantitative research is higher then the initial requirement of qualitative research.

It was for these reasons concluded that a quantitative approach would be more appropriate for this work, and that it might pave the road to future research.

The greatest strengths of this approach is that: data is readily available, results are visual, and exploration of unknown or unfamiliar topics are easier to achieve. The greatest limitations is that findings and conclusions are to a greater extent subjective, data collection and analysis is more time consuming, and details and information can be lost when raw data is only analyzed through graphs and figures; this might cause important results to be overlooked or misinterpreted.

3.1 Choosing Simulation software

LTspice is an easy to use circuit simulator with good tutorials and learning material, it is freeware, and it was the simulation option that had the best description on how to start working with memristors.

Another noteworthy simulation tool is Xyce, there are several promising papers and some tutorials on how to implement memristor models here as well, but a understanding of Verilog-A is required to utilize the work. The need to learn a new hardware descriptive language, software, and how to relate all that to a new unknown and somewhat obscure component was deemed too ambitious for the scope of a thesis unless necessary.

For these reasons LTspice was chosen as the simulation tool to use in this work.

3.2 The path to selecting a memristor model

The first step after a brief exploration into memristors, before starting the process of modifying a memristor model to match the behaviour needed in order to answer the research questions, it was necessary to gain some basic experience with the LTspice memristor models made available through the collaborative github project "memristor-models-4-all" [44] (other models such as the one presented in [43] are interesting, but shared model files has not been found and recreating them from the papers was not within considered within the scope of this thesis and therefor not included).

The first objective was to investigate the behaviour of the models shared at [44] three

were investigated: the Knowm mean metastable switch memristor model, the University of Michigan memristor model, and the Yakopcic model. The after documenting the base models behaviour with different inputs, work on recreating behaviour seen in the physical memristor started. This was attempted by simulating a circuit with documented behaviour from multiple sources, and making small adjustment to the parameter values of the models and notating down observed changes.

First up was the Knowm MSS model, this model is defined by parameters that are easy to understand at a basic level, while also supported by a robust mathematical foundation; and after some experimentation, it rapidly showed promising results and a preliminary match was achieved.

Up next was the U.M. model, it proved more challenging however, after many adjustments to the parameters controlling the model were made with little headway, it was eventually decided that work had to progress to the nest stage, and the model was abandoned to allow the project to progress.

Last up was the Yacopcic model and it showed early promise for ternary simulation, due to a clear ability to have partial change in memristance when nudged, this can be seen in appendix B figures B.38 and B.39. Like the U.M. model however, experimentation with this model made little headway towards a match with the real counterpart.

For these reasons, and on account of achieving promising result with the Knowm MSS model, the Yacopcic model was allows left behind, and it was decided that work would proceed with the MSS model in order to progress in the research.

3.3 The Transition From Sine to Square

At this stage, the first working ternary circuit was able to reproduce the memristance levels described in [4]. The circuit utilized a half sine wave signal modified from [35] with the amplitudes described in [4]. Attempts were made to translate simulation elements such as sine sources and the corresponding MUX and sample control signals that required coordinated timing into circuitry that could be implemented on a real circuit without the a need for an expensive waveform generator. Not only did it prove

challenging to design, but the needed component and complexity added up rapidly.

The effect of replacing the sine waves with square waves had on writing to the memristor was explored. It did not help in simplifying the circuit, however, until a realisation that if a DC source with the desired amplitude was applied the MUX, the MUX would only output the voltage while the correct control sequence is applied. The voltage applied to the memristor would then effectively be a pulse without the need for timing, eliminating the need for complex circuitry.

Chapter 5

RESULTS

This chapter will present most of the results of the thesis, but the research paper included in appendix A is also part of the results and will be discussed in chapter 6.

Before any exploration into ternary started, results on how the model compares to real memristors and what limitations it faced needed data. Therefore, the chapter starts with the presentation of results from the parameter tweaking process. The chapter moves on to show data on the problems the model faces. The first section presents data on how the model responds in a setting similar to some situations found in other literature. The second section presents key data of ternary storage. Finally, this chapter shows the final circuit and the simulation results from [40].

More results that are not as directly relevant to the thesis, but might be interesting to others that want to continue on this work or recreate some of the experiments can be found in appendix B, and the public Github repository [54].

1 Parameter Tweaking

The first goal of this thesis was to match memristor response behaviour with that of the Knowm W/tungsten SDC memristor as used by the ternary research group. This journey required trial and error, incremental improvements, and dead ends. This section

will show key results produced throughout this process and data used to discuss the model's limitations.

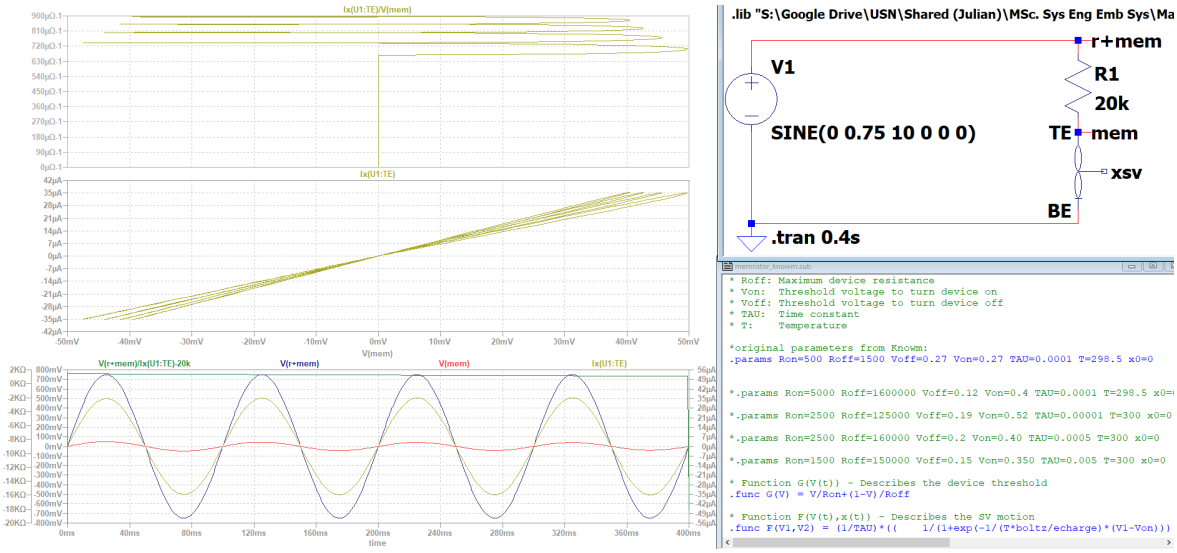


Figure 5.1: G-V, I-V, and time plot of a simple memristor circuit using default parameter settings.

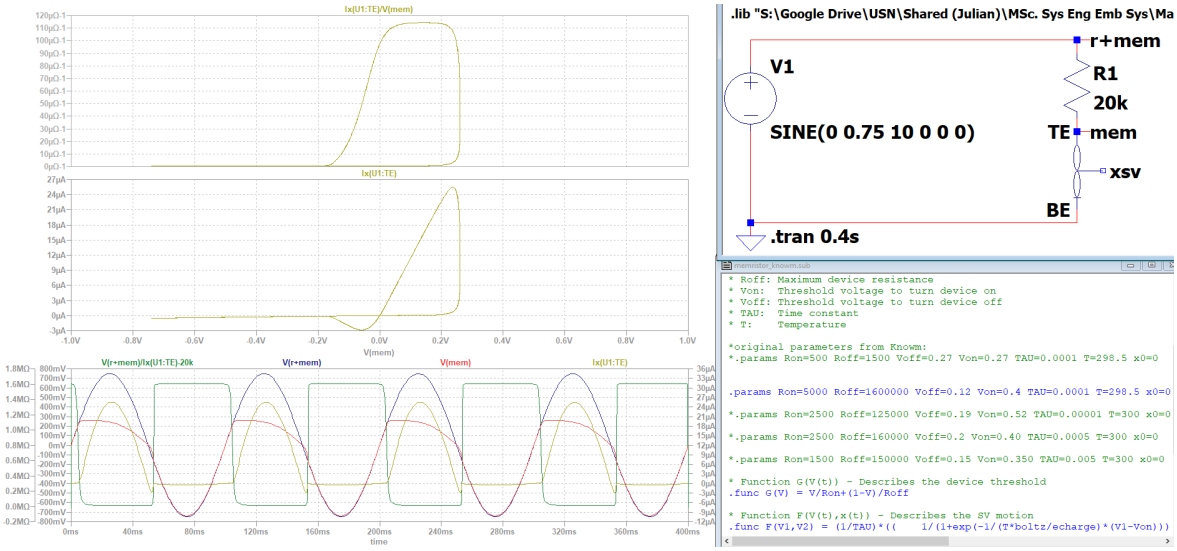


Figure 5.2: G-V, I-V, and time plot of a simple memristor circuit using a preliminary iteration of the parameter settings.

Figures 5.1, 5.2, and 5.3 are created using the same circuit but with with three different key parameter iterations. In figure 5.1 the base parameters of the model as presented in [44] is used, figure 5.2 uses an early iteration of the parameters, and figure 5.3 uses the final parameter settings.

All three figures are divided into the same 5 segments. The the top left plot shows

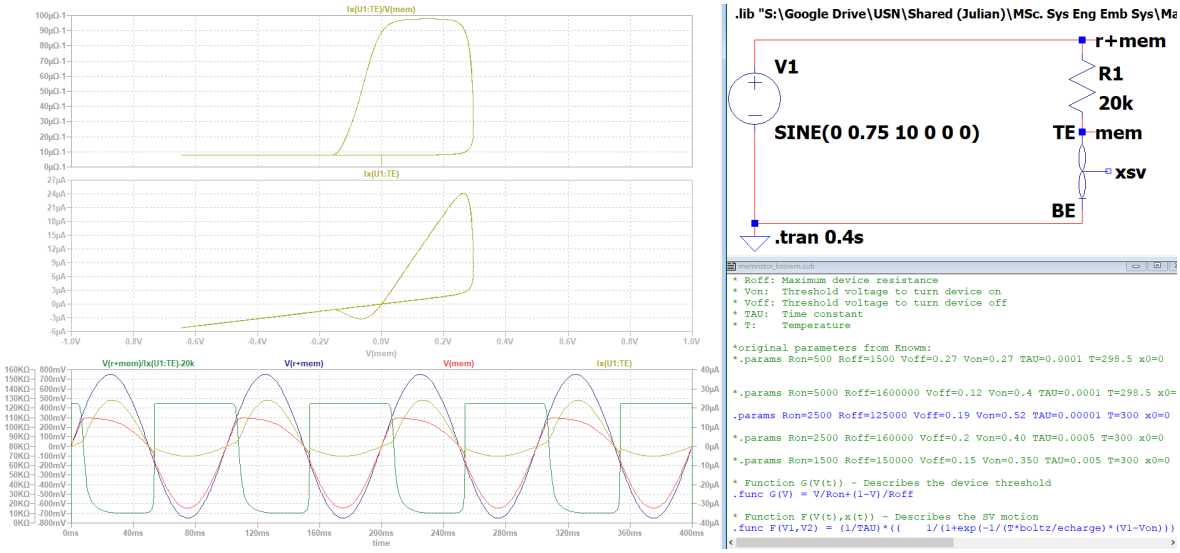


Figure 5.3: G-V, I-V, and time plot of a simple memristor circuit using the final iteration of the parameter settings.

the G-V curve over the memristor. The middle left plot shows the I-V curve over the memristor. The bottom left plot shows the calculated resistance of the memristor in green, the voltage over the resistor and memristor in blue, the voltage drop over the memristor in red, and the current through the memristor in yellow. The top right segment of the figures show the simulation circuit. Finally, the bottom right segment shows the memristor parameter settings.

A breadboard implementation of the circuit used in figures 5.1, 5.2, and 5.3 was created using a Knowm M+SDC Memristor 16 Discrete 32 DIP W/Tungsten chip, in order to supplement the external reference data, and gain practical experience with memristor circuits.

The analogue discovery was used with the waveform software to generate a 10Hz 0.75V sine wave, read the voltage over the memristor and 20k load resistor, as well as the voltage over the load resistor alone using the digital oscilloscope. The voltage drop over the memristor was calculated with $V_{mem} = V_{tot} - V_{R1}$. The momentary memristance was calculated using $R_{mem} = \frac{V_{mem} \cdot R1}{V_{R1}}$. Finally the current was calculated using $I = \frac{V_{tot}}{R1 + R_{mem}}$. The data was exported from waveforms in .CSV format (available on [54]). Finally the data was plotted using Matlab R2019b.

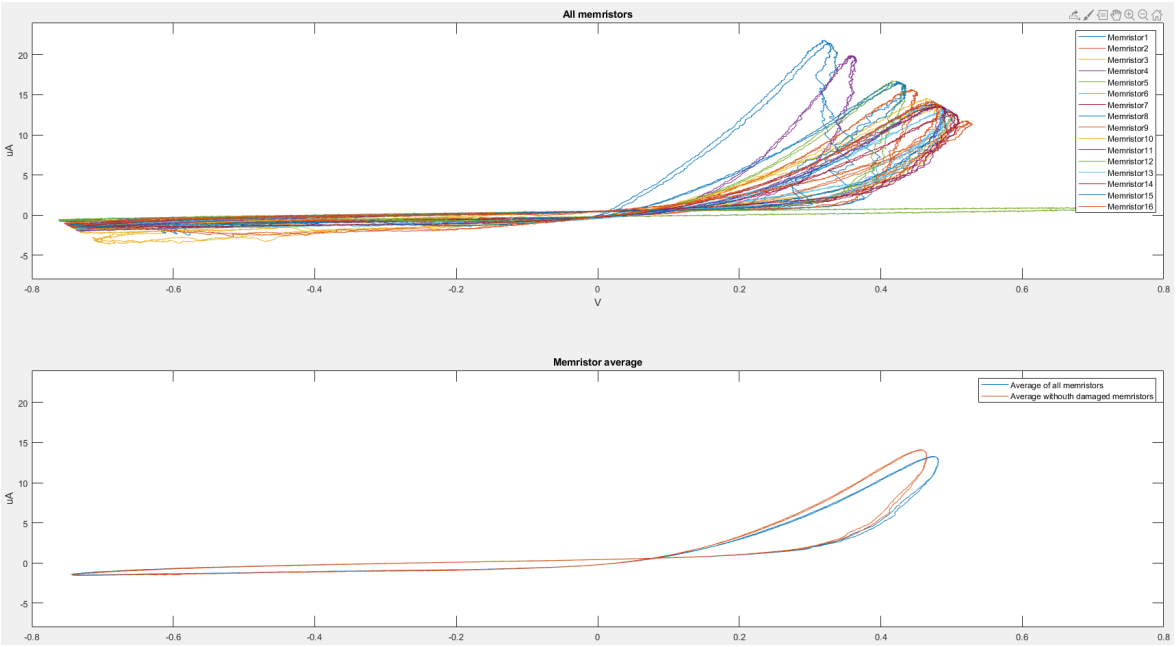


Figure 5.4: Individual and average I-V plot of all memristors on a Knowm M+SDC Memristor 16 Discrete 32 DIP W/Tungsten chip [39]

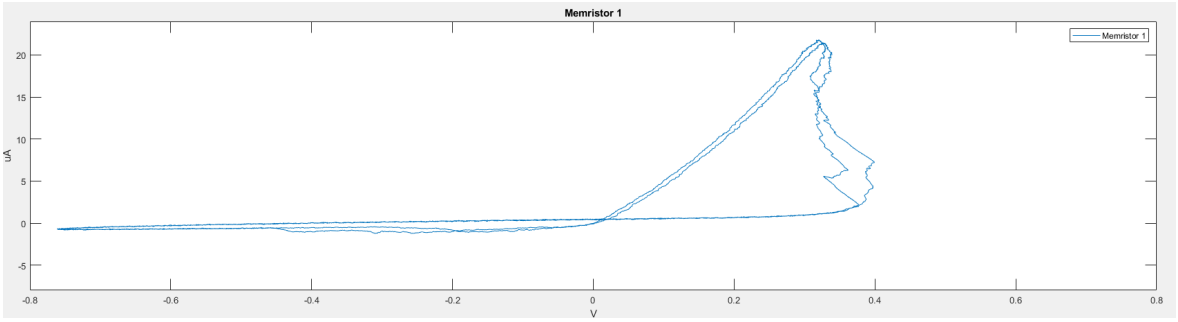


Figure 5.5: An isolated view of memristor 1

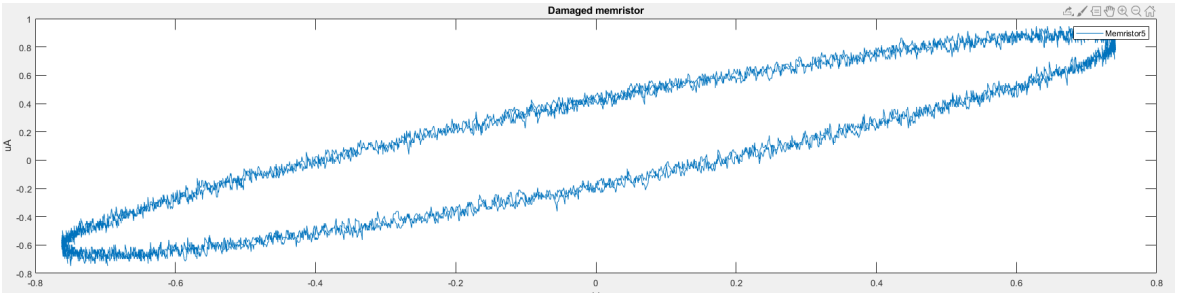


Figure 5.6: A zoomed in, isolated view of the damaged memristor seen in figure 5.4.

The results are shown in figures 5.4, 5.5, and 5.6. The top graph in figure 5.4 shows the plot of all memristors on a Knowm M+SDC Memristor 16 Discrete 32 DIP W/Tungsten chip [39]. The bottom graph shows the average plot of all 16 memristors in blue, and an

average excluding the damaged memristor in orange. Figure 5.5 shows an isolated view of memristor 1. Finally figure 5.6 shows a zoomed in, isolated view of the damaged memristor seen in figure 5.4.

Throughout the process of modifying the parameter settings, various simulation experiments were performed to assess the performance of the model, identify weaknesses, and validate results.

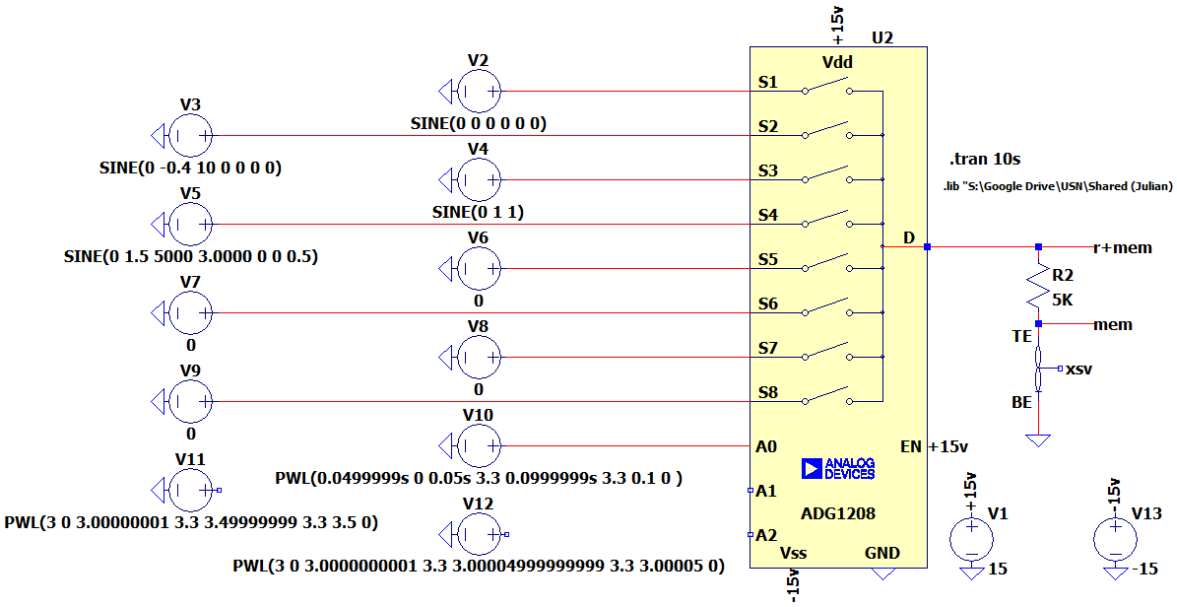


Figure 5.7: The LTspice circuit used to generate 5.8, 5.9, and 5.10

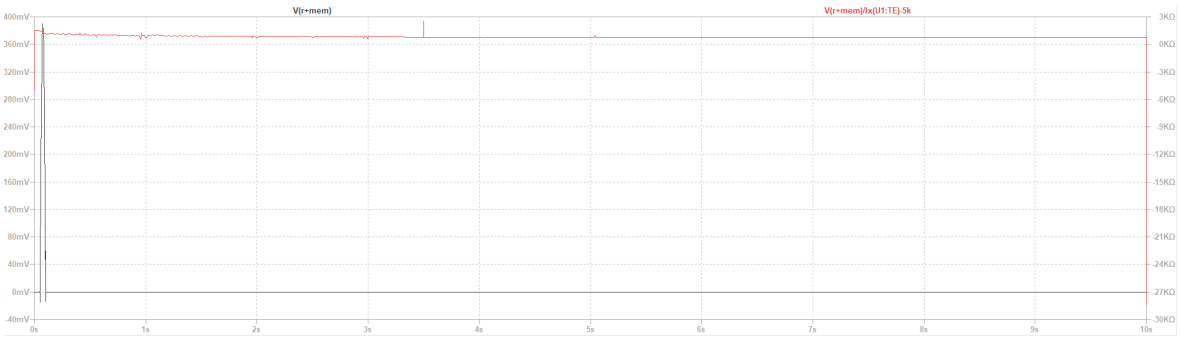


Figure 5.8: Drift experiment using the Knowm MMS model with parameters: "Ron=500 Roff=1500 Voff=0.27 Von=0.27 TAU=0.0001 T=298.5 x0=0".

One such experiment used the circuit shown in figure 5.7. Three of the results from this experiment is can be seen in figures 5.8, 5.9, and 5.10. This experiment shows the effect the TAU parameter has on the drift behavior found in the Knowm MMS model.

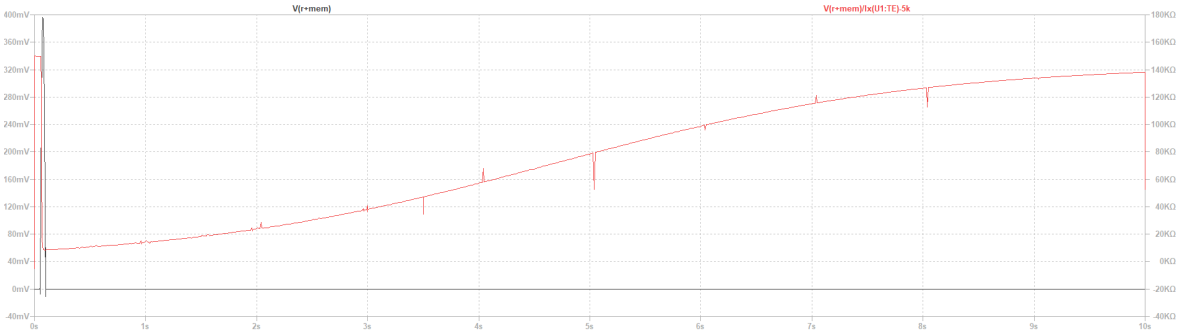


Figure 5.9: Drift experiment using the Knowm MMS model with parameters: "Ron=1500 Roff=150000 Voff=0.15 Von=0.350 TAU=0.005 T=300 x0=0".

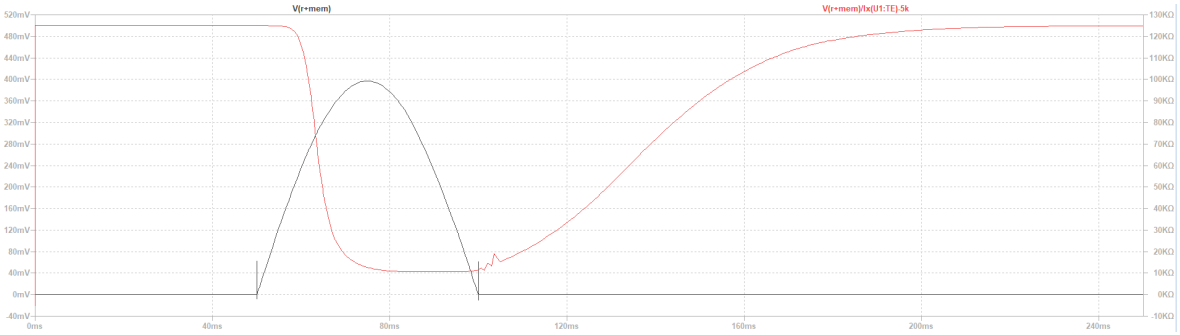


Figure 5.10: Drift experiment using the Knowm MMS model with parameters: "Ron=2500 Roff=125000 Voff=0.19 Von=0.52 TAU=0.00001 T=300 x0=0".

Another key experiment attempts to recreate figure 6.5 from Campbells paper [35]. This was done in order to assess how well the model can recreate a binary results were the memristor is written to and read for, before moving on to ternary experimentation.

Two results from this experiment will be included in this thesis and can be seen in figures 5.11, and 5.12. The black plot-line shows the voltage over memristor and load resistor, while the red plot-line represents the voltage drop over the load resistor alone.

Figure 5.11: A recreation of 6.5 in LTspice using the Knowm MMS model with parameters: ".params Ron=1500 Roff=150000 Voff=0.15 Von=0.350 TAU=0.005 T=300 x0=0".

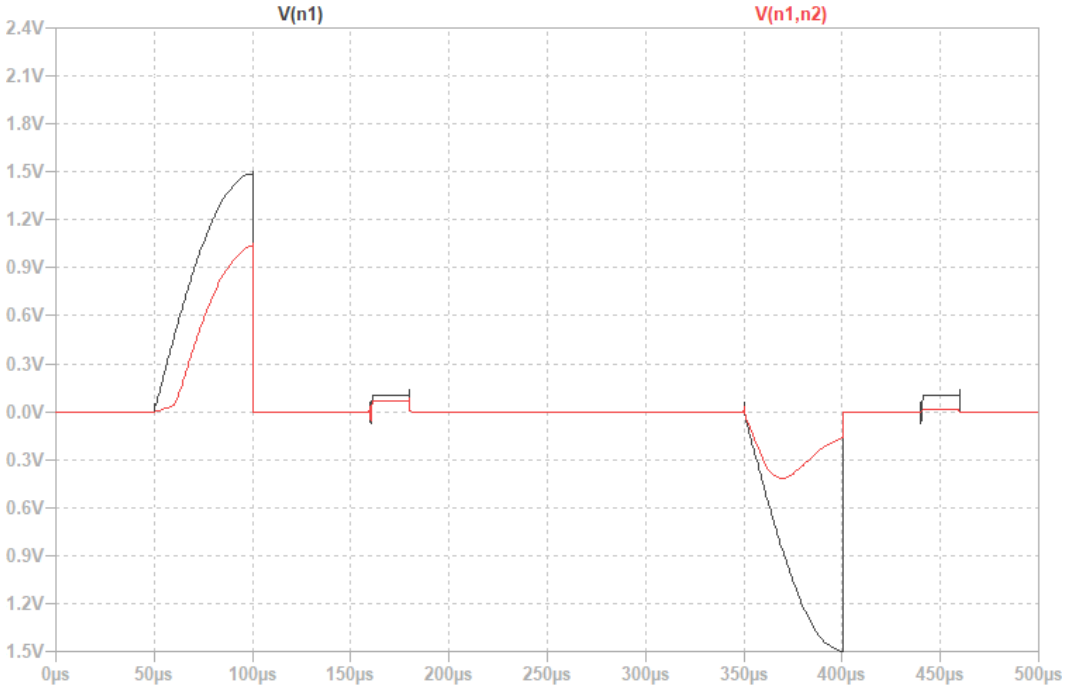


Figure 5.12: A recreation of 6.5 in LTspice using the Knowm MMS model with parameters: ".params Ron=2500 Roff=125000 Voff=0.19 Von=0.52 TAU=0.00001 T=413.15 x0=0".

2 Ternary Storage

This section will show the key results relating to ternary storage.

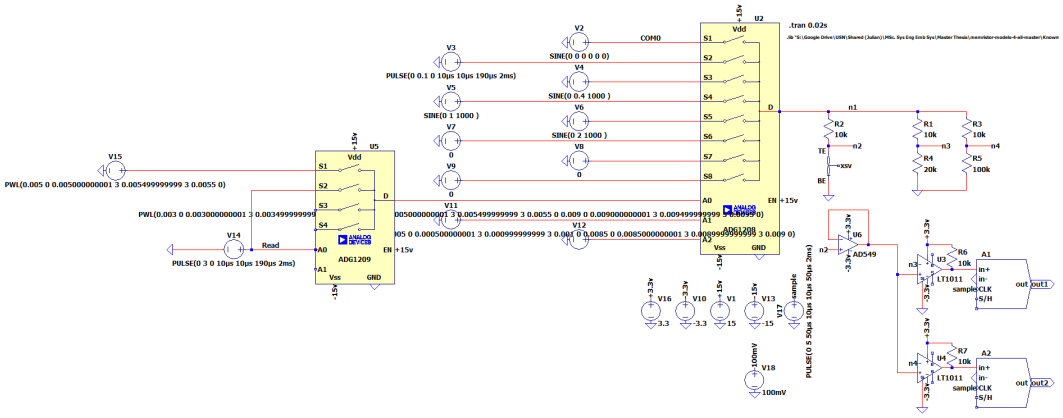


Figure 5.13: The first ternary circuit capable of writing to a memristor, reading from it, and outputting the logic state.

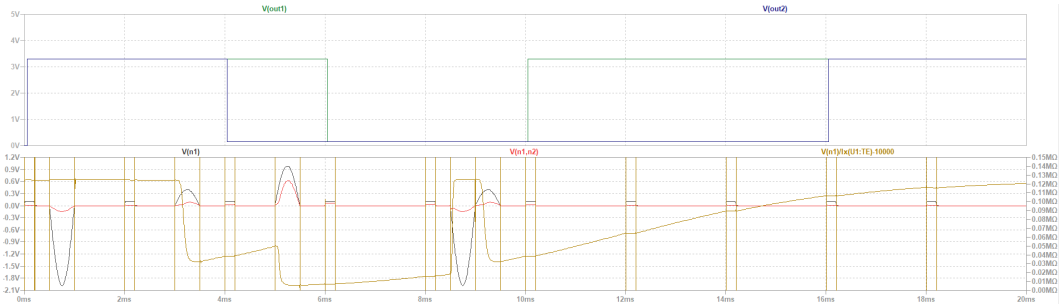


Figure 5.14: Simulation graph from the circuit in figure 5.13. The top plot shows output levels. The bottom plots MUX voltage, load resistor voltage drop, and the calculated memristance.

The first circuit that was able to read and write trits on a memristor is show in figure 5.14. In this iteration, the write was done with half sine pulses of -2V, 0.4V, and 1V; they were produced in AC voltage sources connected to a MUX. The read was generated as 0.1V square wave read pulses with a 2ms period and 50% duty cycle. Timed pulses on the control pins of the MUX then controlled what source was applied to the memristor and reference resistors.

The read memristor and reference resistors are all in series with there own 10kΩ load resistor to limit the current over the memristor, and allow the voltage drop over each branch to be compared. It is critical that the three load resistors have minimal differences in there resistances.

When a read pulse was outputted by the MUX, the voltage drop across the memristor was compared to the drop of each of the reference resistors using a comparator for each. The comparators then goes high or low depending on the memristance of the memristor with respect to the relevant reference, and the output is held until the next read pulse.

The simulation result of figure 5.13 are shown in figure 5.14, and is divided into two graphs. the top graph shows the output of the sample-hold shown in figure 5.13. When both outputs are high, the memristors ternary logic state is read as high (”2”). when out1 is high and out2 is low, the logic level read from the memristance is medium (”1”), and when both outputs are low the stored logic state is read as low (”0”). The bottom graph shows the voltage over the memristor and load resistor is black, and the voltage drop over the load resistor in red, and it shows the calculated memristance level of the memristor in yellow.

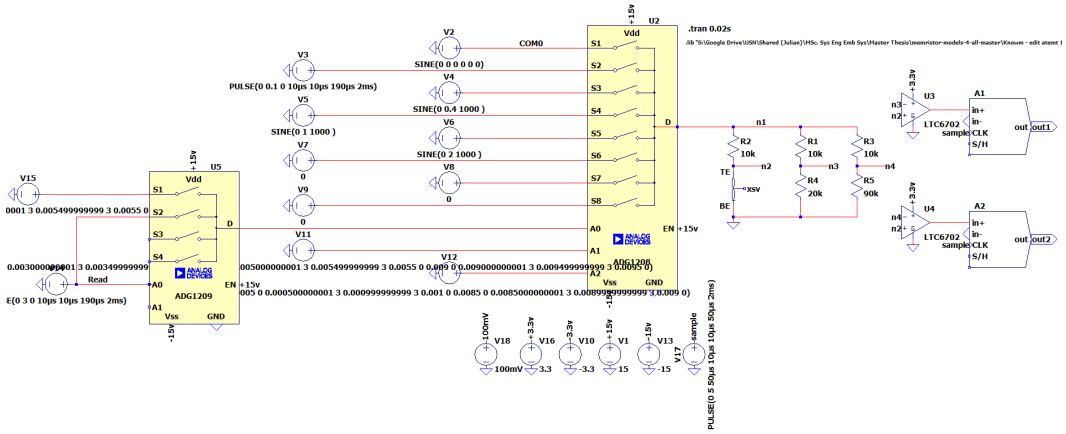


Figure 5.15: An iteration of the previous circuit with complexity in read section.

The circuit shown in figure 5.15 is of a later iteration of the circuit with a lower component count and a comparator that does not require a negative voltage rail.

When the simulation was first run with this circuit, the high reference resistor was still set to 100kΩ, the simulation result seen in figure 5.16 no longer reported a high state. The bottom plot plane of this figure shows a zoomed in view of the voltage level at n2, n3, and n4 shown in 5.15.

After reducing the resistance of the high reference resistor to 90kΩ, the output, as seen

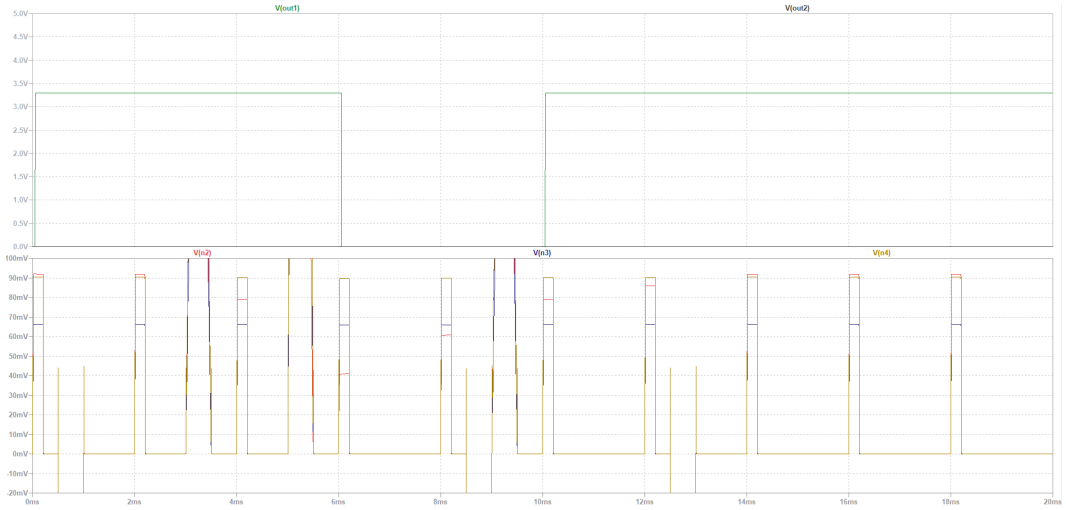


Figure 5.16: Simulation graphs from the circuit in figure 5.15 using a high reference resistor of 100kΩ. The top graph shows output levels. The bottom plot-plane shows a zoomed in view of the voltage drop across the memristor and reference resistors.

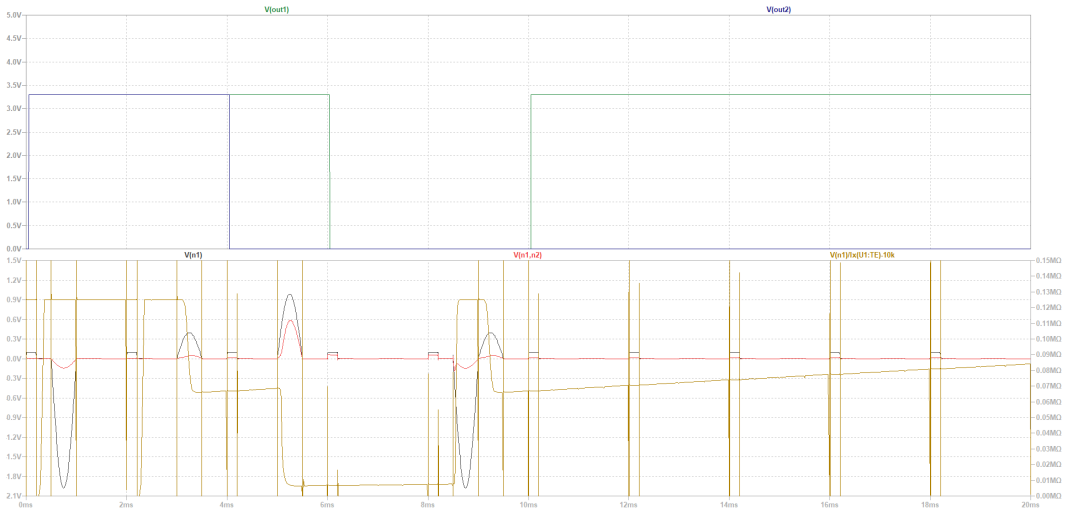


Figure 5.17: Simulation graph from the circuit in figure 5.15. The top plot shows output levels. The bottom plots MUX voltage, load resistor voltage drop, and the calculated memristance.

in figure 5.17 operated as expected again.

Figure 5.18 shows the final iteration of this simulation circuit, after the square wave realisation depicted in chapter 3.3.

The memristance plot seen in 5.19 shows clear calculation errors, but the level is where expected during each read pulse, suggesting minor leakage current from the MUX,

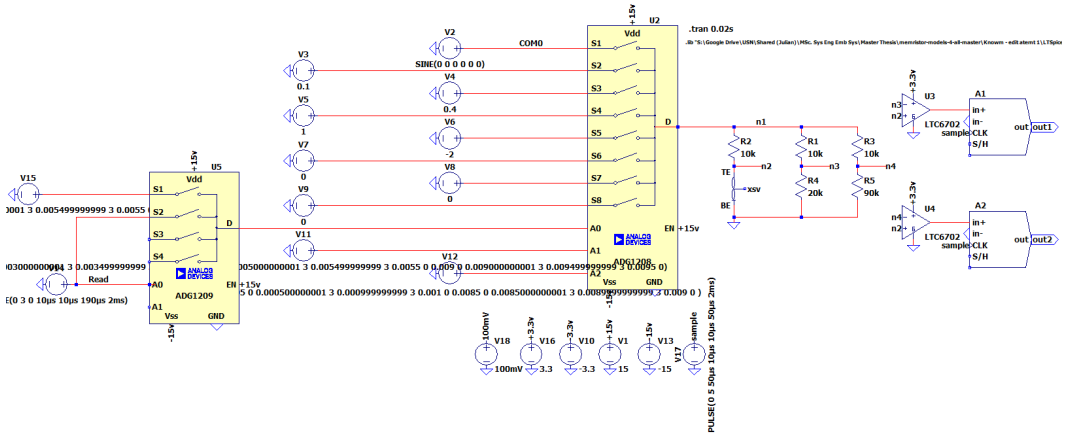


Figure 5.18: The final iteration of first working ternary circuit design.

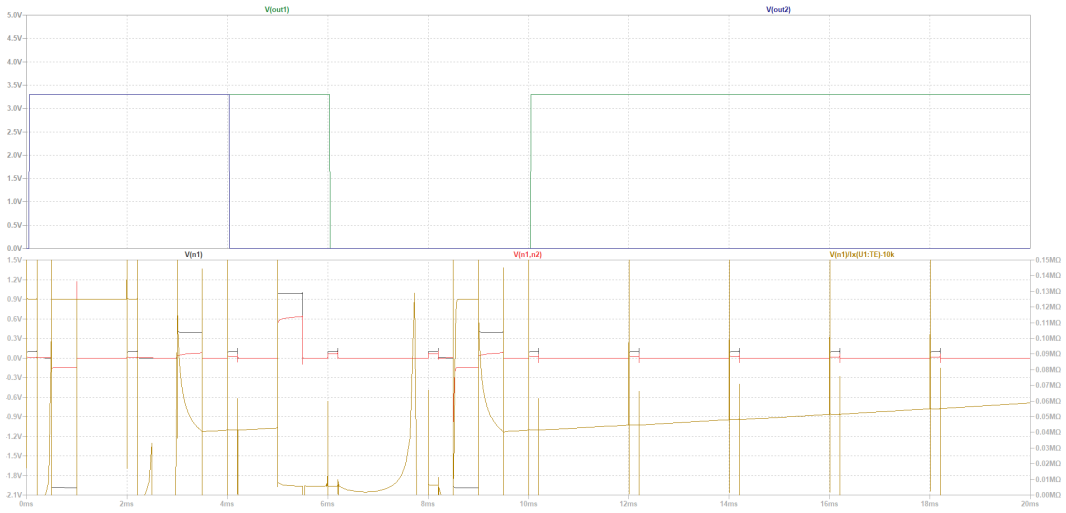


Figure 5.19: Simulation graph from the circuit in figure 5.18. The top plot shows output levels. The bottom plots MUX voltage, load resistor voltage drop, and the calculated memristance.

throwing of the calculation. This does not interfere with the model state or behavior however, and is deemed acceptable.

The circuit presented in figure 5.20 is included from the paper published together with S. Bos as first author [40], the paper is included as appendix A.

Figure 5.21 shows the LTspice simulation result of figure 5.20. The top plot-plane shows circuit output from the window comparator in green, the computed memristance in red, and the current over memristor in purple. The bottom plot-plane shows the inputs into the circuit. With the read pin shown in green, logical "0" pin in light blue, logical "1"

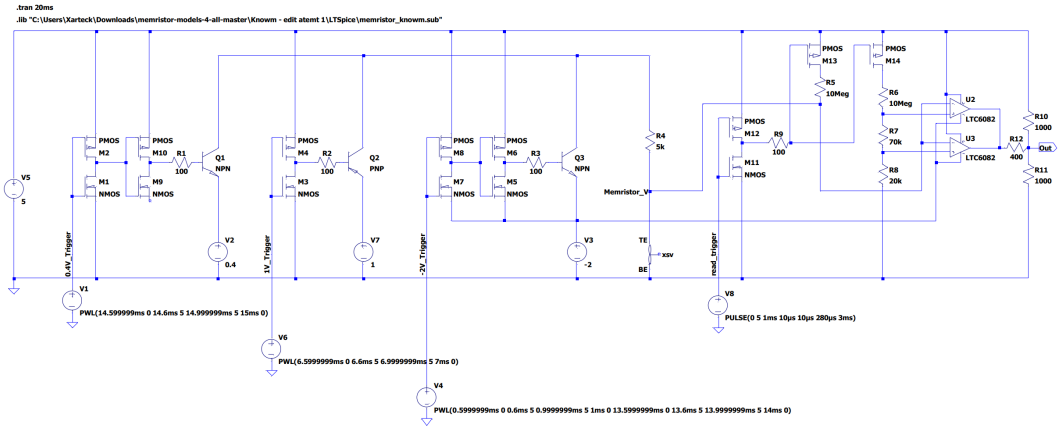


Figure 5.20: Memristor memory circuit with ternary read and write. This figure is developed in cooperation with S. Bos, and published in [40].

in dark green, and logical "2" pin in yellow.

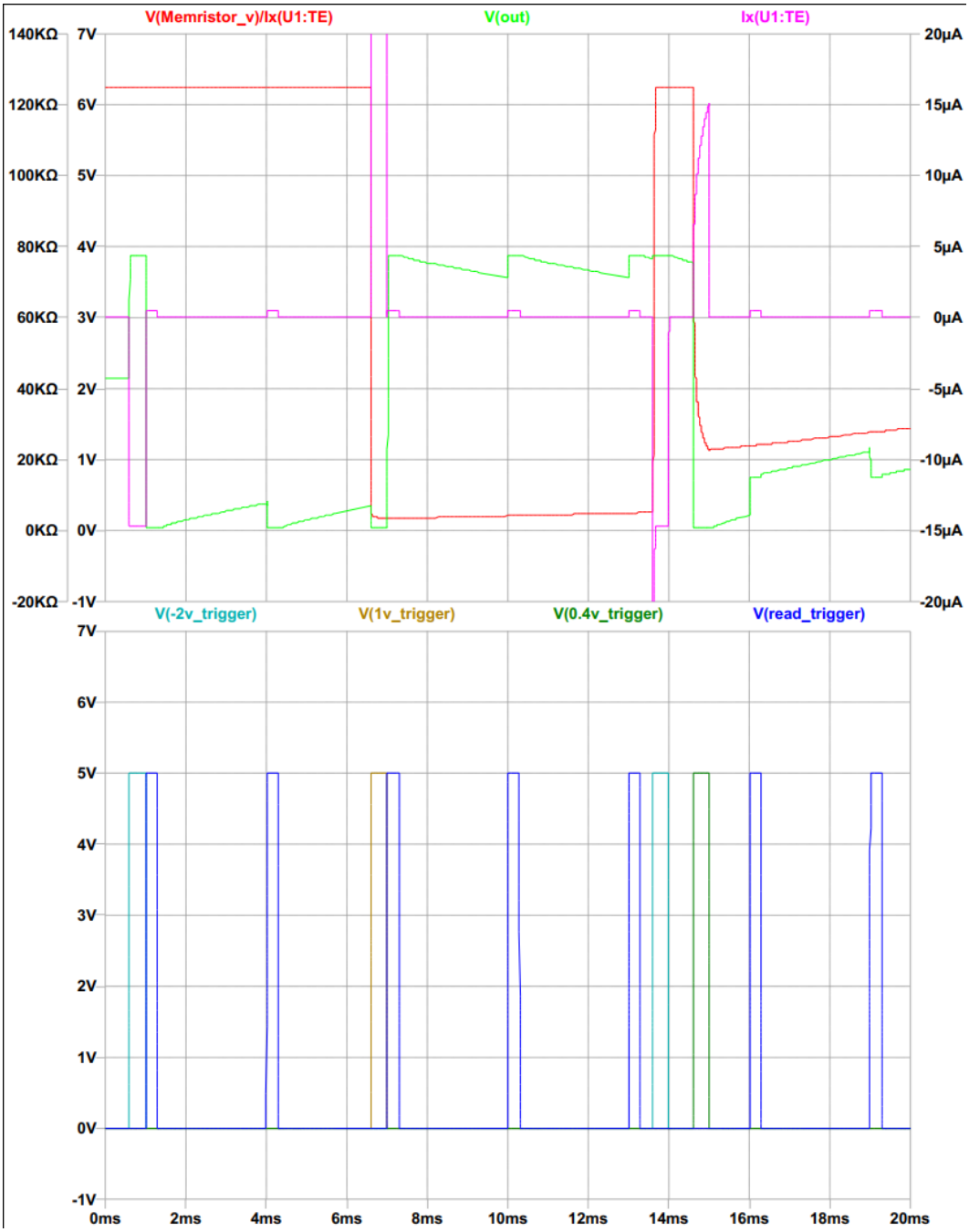


Figure 5.21: Simulation graph from the circuit in figure 5.20. The top plot shows calculated memristance, the ternary output, and the current over the memristor. The bottom plot shows read and write triggers.

Chapter 6

DISCUSSION

This chapter will discuss the results, compare real world memristors and results from other research to simulation results, and the simulation results published in our paper [40].

The main topics of focus in this chapter is:

- How the memristor model parameters affect the behaviour of the model, were they provide a good match to real memristor observations, and why they ended up the way they did.
- Limitations the model faces with the parameter settings that was chosen.
- Analysis of ternary memristor based memory circuits, their strengths and weaknesses, and how the model held up in this work.

1 Matching Model Behaviour to the Real Memristors

The first goal of this thesis, before work on designing a circuit capable of reading and writing ternary data to a memristor could begin, was to investigate too what degree the chosen memristor model [44, 55] could be made fit the response of the chosen Knowm SDC W/tungsten memristor [39].

1.1 Hysteresis response

The first set of data analyzed to reach this goal was comparisons between the I-V and G-V plots of the model and real memristors.

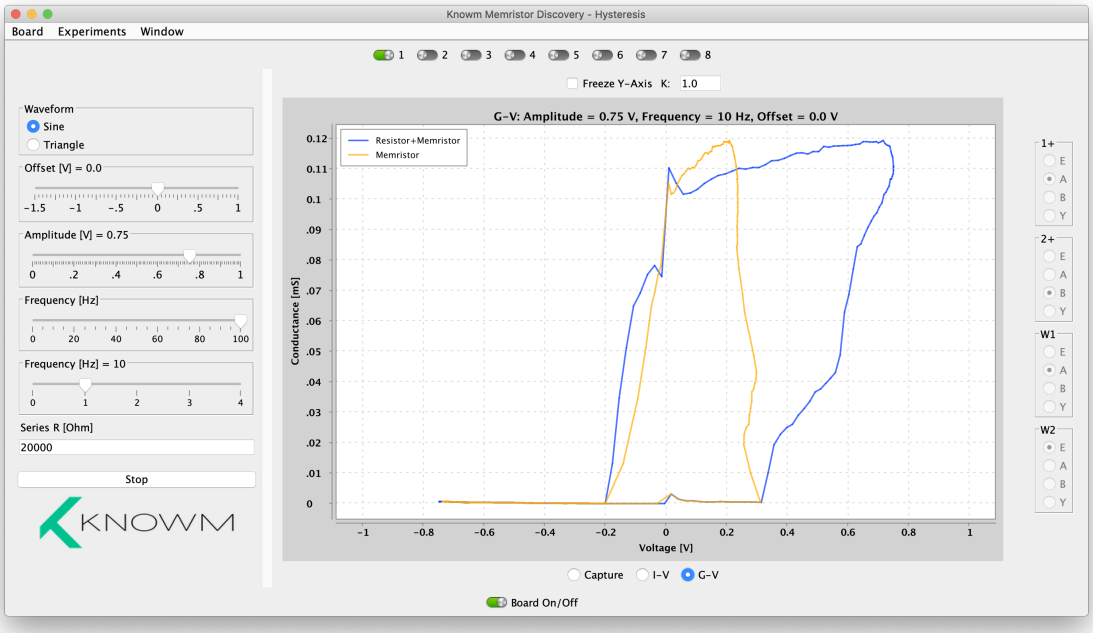


Figure 6.1: The G-V plot of a 0.75V 10Hz sine wave applied to a Knowm SDC W/tungsten memristor in yellow, and in series with a 20kΩ resistor in blue. The figure is supplied by Eric Johnson at Knowm Inc.

Figures 6.2, 6.1, and 6.3 are captured on the same circuit, with a Analogue Discovery 2 and Knowm Memristor Discovery software.

In figure 5.1, the responses given from the chosen memristor model with its base parameter setting is shown. It can be seen that the response does not closely match the response of the real memristor, but rather that the memristance is nudged slightly down for each sine period, with no load resistor, a similar behavior is only seen well

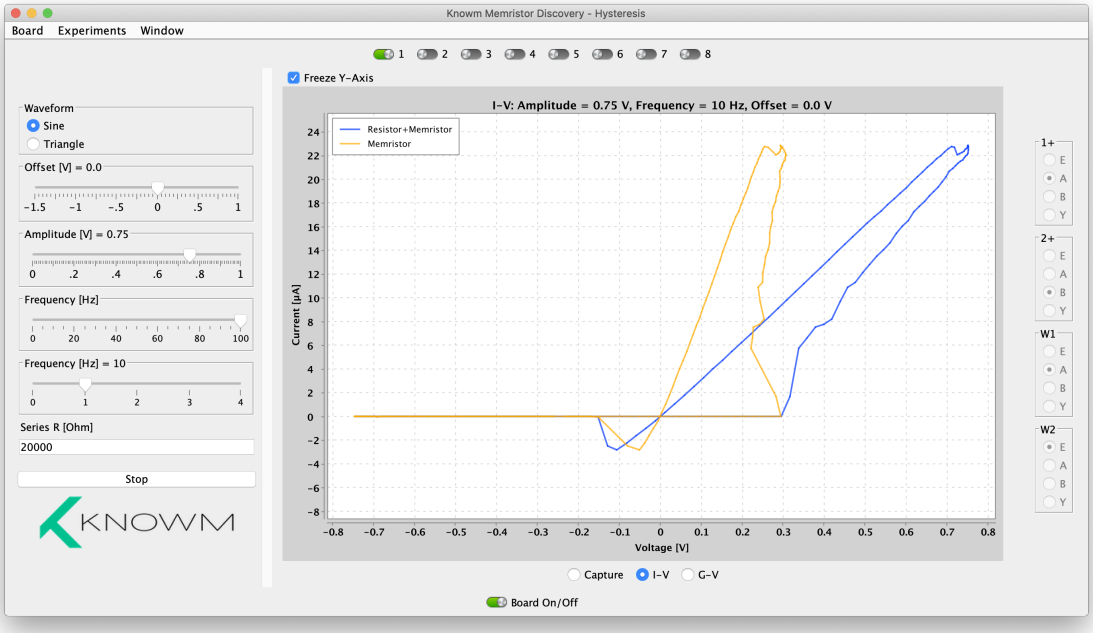


Figure 6.2: The I-V plot of a 0.75V 10Hz sine wave applied to a Knowm SDC W/tungsten memristor in yellow, and in series with a 20kΩ resistor in blue. The figure is supplied by Eric Johnson at Knowm Inc.

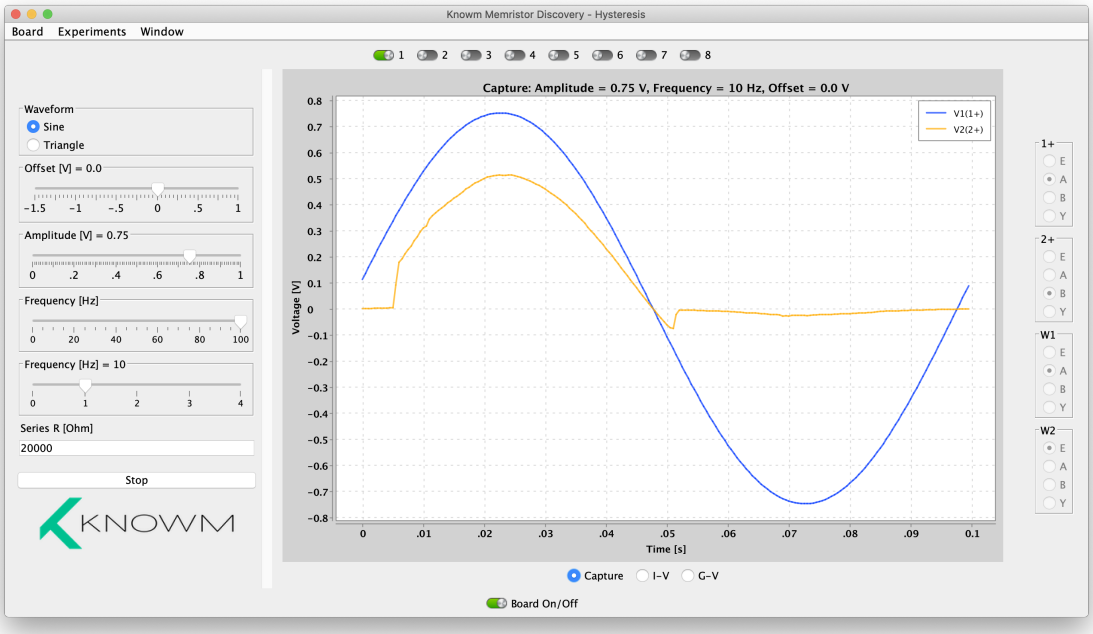


Figure 6.3: A plot of the voltage in blue, and current in yellow with respect to time over a Knowm SDC W/tungsten memristor in series with a 20kΩ resistor. The figure is supplied by Eric Johnson at Knowm Inc.

above 1000Hz (see figures B.1 to B.9 in appendix B), it is postulated that the behaviour appears here due to the voltage being low over the memristor due to the large resistance of load resistor compared to the "Roff" in the base model.

Figure 5.2 shows an early iteration of parameter tweaking of the model. Now the response is closer to what is plotted in figures 6.2, 6.1, and 6.3; as well as 6.4, 5.4, and 5.5.

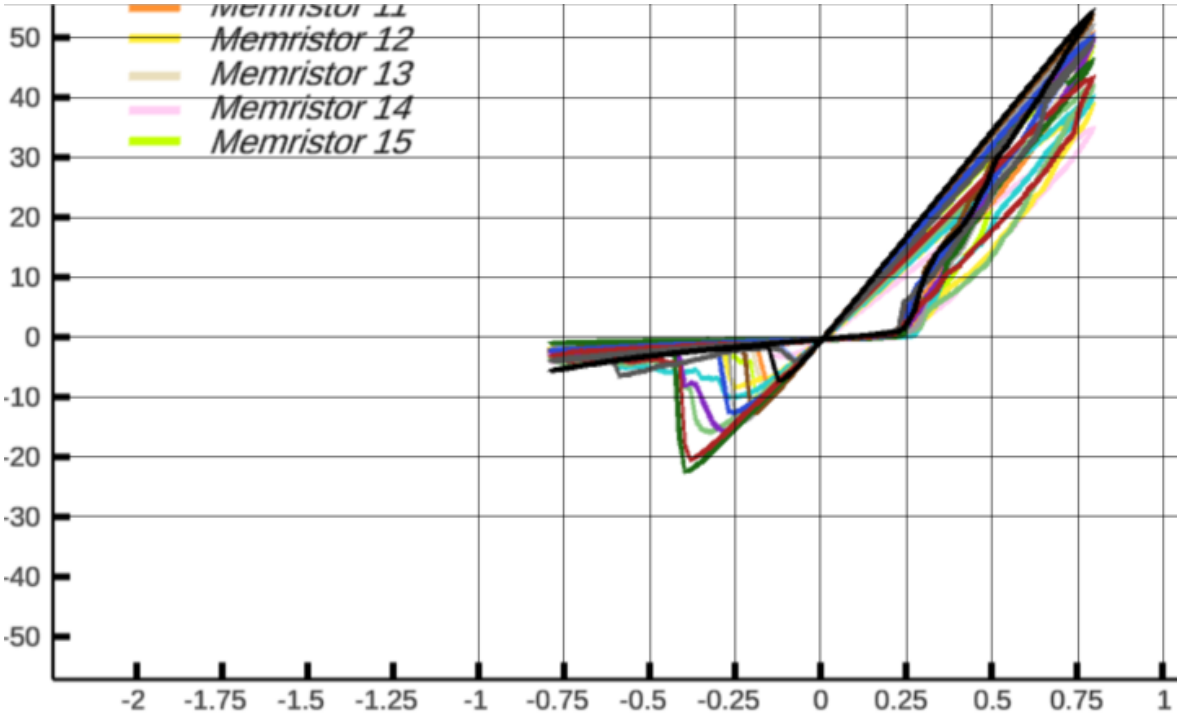


Figure 6.4: The IV-plot over a Knowm SDC W/tungsten memristor. The figure is taken from the uMemristorToolbox paper of S. Bos 2020 [4].

However, when compared to the memristance values found in early experiments of the research group on the memristor they had, the high memristance level of figure 6.2 was an order of magnitude higher then the memristors used by the group, the I-V curves of there memristors can be seen in figure 6.4 taken from [4] (note that this graph is made with the voltage over the memristor and the load resistor much like the blue plot in 6.2, not just the voltage drop over the memristor alone as seen in 6.2 yellow plot and the plots in 5.4).

The reason for the difference in Roff between the graph provided by Knowm and the

same type of memristor used in [4] is believed to be due to the forming process. Knowm reports that the memristance of the memristor is around 1MΩ but that if it is driven with "high" current approaching 1mA the Roff will reduce [39, 55].

The early work by the research group, later published in the aforementioned paper [4] defines logical 2 as memristance values between 0-8kΩ, logical 1 as values between 8k-100kΩ, and logical 0 as memristance values above 100kΩ. The final "Roff" parameter was therefor set at 125000Ω.

1.2 Memristance drift

Another problem with the first iteration of the parameter settings is the relatively high "TAU" parameter. As seen in figures 5.9, and 5.10, the memristance of the models drifts even if the applied voltage is 0. It is believed to happen due to how the change in state probability is a funcion of voltage, and the probability not reaching 0 at 0V. To illustrate this, figures 5.8, 5.9, and 5.10 from an LTspice simulation are included in chapter 5, the circuit used is shown in figure 5.7. In this figures, a half sine with an amplitude of 0.4V and a period of 50ms is applied to the memristor in series with a 5kΩ load resistor. The first figure is added for reference, it uses the base memristor model parameters ("params Ron=500 Roff=1500 Voff=0.27 Von=0.27 TAU=0.0001 T=298.5x0=0) are used. With these parameters the range of the memristor is too low, and the change in memristance is not meaningful.

The second figure uses parameters with a relatively high TAU (params Ron=1500 Roff=150000 Voff=0.15 Von=0.350 TAU=0.005 T=300x0=0). Now the memristance manages to drop to about 8kΩ, and creep back towards the high memristance level over the 10s graphed. Compare that to the third figure using the final parameters (.params Ron=2500 Roff=125000 Voff=0.19 Von=0.52 TAU=0.00001 T=413.15 x0=0) where the memristance drifts up to about 75% in as little as 60ms, and are almost at the high memristance level after 150-200ms.

Experimental results found when investigating the drift behaviour suggest that the drift might be related to the relationship between the parameters "Von" and "Voff" and the MSS probability function when the parameters are not equidistant to 0.

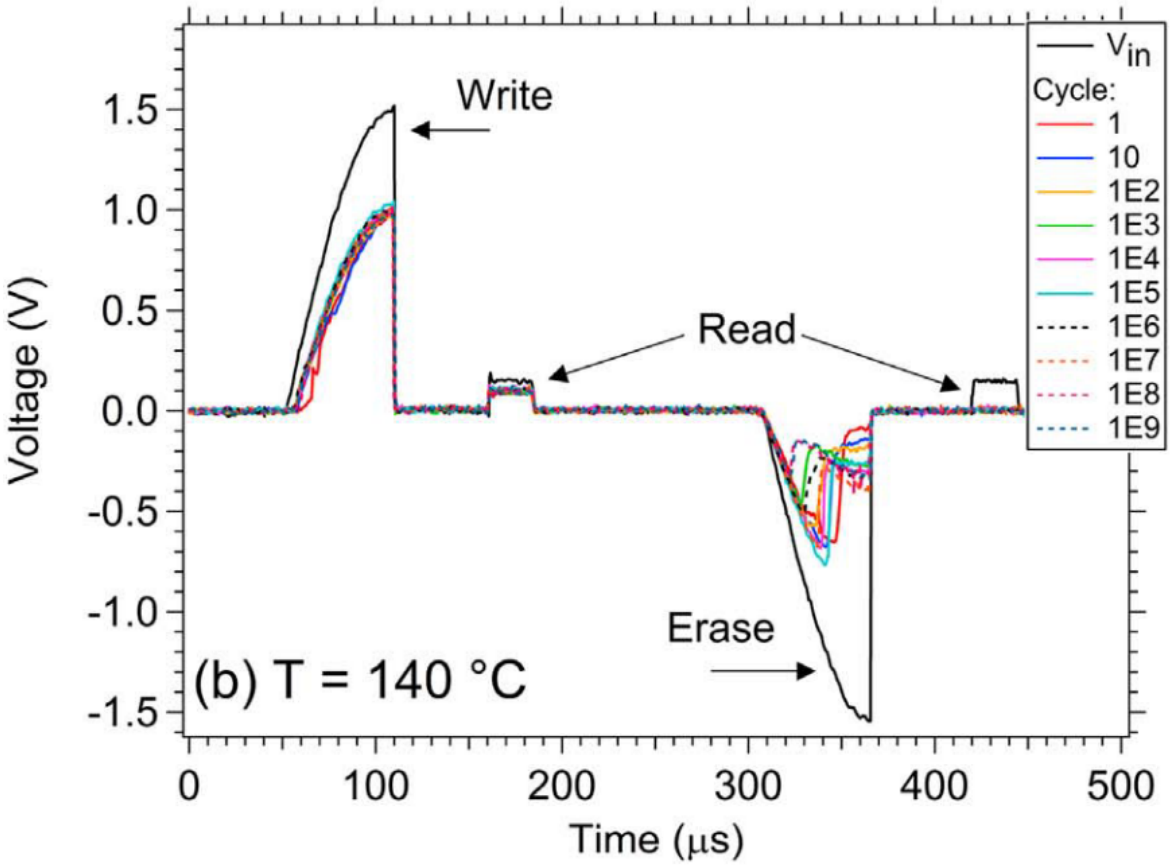


Figure 6.5: The black plot-line shows the voltage applied to a SDC memristor in series with a 10kΩ load resistor. The coloured plot-lines show the voltage drop over the load resistor after increasing decades of cycling, all at 140C°. The figure is taken from [35].

Validation with Campbell results

It could be concluded that the first option would be better due to a slower drift, but when the same parameters are used to create figures 5.11, and 5.12 in order to validate the parameter settings against figure 6.5 taken from Campbells 2017 paper [35], it can be seen in figure 5.11 that the first parameter settings does not respond fast enough, and the read remains unchanged. The results using the final parameter settings seen in figure 5.12 however is a good match to the behaviour seen in her paper.

1.3 Reevaluation of hysteresis response

The last step performed in tweaking the memristor parameters was to compare the simulation graph of the circuit used in figures 5.1, and 5.2; now using the final parameters shown in figure 5.3, with the graphs in 6.2, 6.4, and to validate it all with data from a real circuit shown and explained in 5.4, 5.5, and 5.6.

1.4 Section summary and last remarks

To sum up the results of the parameter tweaking. The I-V curve of the model is a good match for the average of the memristors in 6.4, as well as memristor 1 in 5.4 and 5.5. The average plot in 5.4 however is flatter, and none of the memristors show the same dip seen in the I-V plot of the model, B.44, or 6.4.

There is also more variance between the memristors in 5.4 then there is in those found in 6.4, this is believed to be due to memristor forming, [4] performed a forming routine on all memristors, this was not done on the set found in 5.4.

Finally, the model parameters appears to match the memristor used in [35], but they are not perfect, most notably the memristors used there seem to go to a high memristive state more abruptly during the erase, but the result are close enough to be tolerable for simple response simulations and as a tool for design.

Figure 5.6 shows the I-V curve of a damaged memristor, it shows that the Pinched hysteresis loop that defines memristors are no longer present, if this memristor was used in an experiment, the results would not be what is expected from the system. Damaged memristors are an element that needs to be taken into account when working with them, and more research is needed, solutions like those found in [52] could be implemented, and a system can not rely on every memristor in a crossbar array, remember which ones are damaged and store data accordingly.

2 Developing a Ternary Memristor Storage Circuit

Now that the tuning of the memristor model was complete, and the resulting limitations had been identified; work on ternary storage could finally commence. This section will discuss the results of this process.

2.1 The first ternary memory circuit

First up was to create a simulation circuit able to write ternary values to a memristor, and extract that data without corrupting it.

Figure 5.14 shows simulation results outputted by a circuit based on the one used to generate 5.11. The result of the circuit modification can be seen in figure 5.13. It adopted the amplitude and duration for its write pulses from early experiments of the ternary research group, findings of these experiments were later published in [4].

Although the relative rapid drift in memristance caused by the model is apparent in the figure, it shows that the response of the model approximately puts it into the same three states reported in [4] when similar pulses was applied. Figure 5.14 also shows that the trit stored in the memristor can be read from the model.

Work on simplifying the read part of the circuit eventually led to the circuit found in figure 5.15. This iteration only uses two comparators rather than the three operational amplifiers. A side effect of the change is that the comparators are less accurate when the levels are very close a result of this can be seen in figure 5.16, here it shows that out2 never goes high even when it can clearly be seen that n2 is greater then n3; to solve this the reference resistor was set to 90kΩ, this output is shown in figure 5.17.

Final iteration

Further work on developing the circuit eventually lead to the realisation detailed in 3.3. The end result of this journey is shown in figure 5.18, and the simulation results of that circuit are shown in 5.19.

This final iteration of this ternary circuit uses a MUX with 15v Vcc that controls what voltage is applied to the memristor and reference resistors, swapping between ground,

-2V, 0.1V, 0.4V, and 1V. In the LTspice simulation the MUX is controlled by PWL timed pulses, this would be done with a micro-controller or similar devise in a physical implementation. While the read signal is active, a shorter pulse is sent to A1 and A2 in order to sample the read.

Advantages and drawbacks of this circuit

The biggest advantage is that the outputs set during a read pulse, this is then held stable until the next read pulse is generated, it also outputs the ternary state stored in the memristor as two binary logic signal, this allows the solution to be integrated with already existing devices such as servo controllers, micro controllers and more; but it also means that the output is not ternary in and off it self, limiting its usefulness to future ternary systems.

Another drawback is that the output does not change after the circuit is written to but rather after the next read pulse. The solution also has little to no leakage current over the memristor or reference resistors when the device is not written too or being read.

2.2 Post Binary: The second ternary memory circuit

Figure 5.20 shows the LTspice circuit used to generate 5.21, this circuit is the one used for [40]. This circuit uses CMOS inverters and transistors inspired from [53] but modified to work with ternary. This replaces the the MUX used in the first circuit circuit 5.18, eliminating the need for 15V (9V min) positive and negative rails. The circuit places its reference resistors in series, and uses a two op-amp window comparator as output (with a voltage divider added in to be able to read using an Arduino because it only handles positive voltages).

The window comparator output is not as stable as the output from the set hold in the earlier circuit, and it is affected by write pulses, so a need to capture or use the memristance state during a read pulse still remain; but it also reduces the complexity and facilitates true ternary output from the circuit.

Chapter 7

CONCLUSION

This work attempt's to answer three questions:

- How will the memristor simulation compare to real memristor behaviour?
- What are the limitations and strengths of the memristor model explored in this work?
- Can current memristor models be used to simulate ternary, memristor based memory?

To answer the first question, the thesis presents a modification to the parameters of the Knowm mean metastable switch memristor model [44, 55]. This modification allows the model match behaviour of the Knowm SDC W/tungsten memristor [39, 56], observed both through experimentation with the device, and with the results reported by other researchers.

Various iterations of the parameter settings are analyzed with respect relevant date from several sources. Although an early iteration of parameter settings gave a close match to I-V curves provided by Eric Johnson at Knowm, an analysis between the Knowm results and those reported by Bos el al. [4] reviles why parameter settings with

a lower peak memristance "Roff" was selected to integrate with the research of the ternary research group.

Further analysis shows that, although a smaller time constant "TAU" parameter exacerbates the memristance drift observed in figures 5.9 and 5.10. An analysis of the results reported by Campbell in [35], and the recreations presented in figures 5.11 and 5.12 shows why a sufficiently small "TAU" is required.

The main limitation of the model when modified with the final parameter settings. The memristance state drifts rapidly, even when no voltage or current is applied. Experimental data suggest this is caused by a relationship between the voltage threshold parameters "Von" and "Voff", and the probabilistic manner the metastable switches of the model change state.

The drift causes the model to be unusable for simulating how memristor solutions behave over time, but is tolerable when using it to simulate short term use cases, behaviour in read and write cases, and as a tool for design of such circuits.

When moving on from parameter tuning and model validation, work to answer the final, main research question. To achieve this two different circuits are presented. Several iterations of the first is discussed. Already after discussing the results of the first iteration with respect to the findings in [4], it could be concluded that the M-MSS model with the selected parameters could be deemed viable for simulating memristor response to ternary read and write pulses. This alone makes the model a valuable tool for developing ternary read and write circuits.

The model was was used to develop the first ternary circuit to a point were it could be implemented without the need for complex components like wave-form generators.

The main benefit of the first circuit is that the output is stable after a read pulse, and that the output of the ternary data stored is outputted as two binary signals, this makes it able to interface directly with current equipment such as a servo motor; but this is

also a drawback for future ternary systems were a ternary output would be preferred.

The second ternary circuit published in [40] and shown in 5.20 solves some of these problems and introduces some new ones, it removes the mux that require a minimum of 9V making it more flexible for mobile and battery powered robotics applications, it also outputs a ternary signal, however the voltage divider included to shift the output to only positive voltages have a substantial leakage current, which is unwanted in battery powered systems.

This was intro because the op amp on the breadboard circuit was unable to sink its voltage threshold low enough for an accurate logical "1" read, without a negative voltage bias. The LTspice simulation did not show this problem, but the error is related to the op-amp, not the simulated memristor. In every other aspect simulation was a good prediction of actual memristor performance.

The paper [40] validates with the breadboard circuit that the memristor model responds to the pulses and puts memristor model and real world memristor in states that match, though the breadboard implementation also shows that there is a read error rate of 25%, clearly indicating that the circuit should be considered first generation and future work is needed to make the system more reliable.

Neither circuit require a waveform generator such as the Analogue Discovery, and can be implemented as is on a breadbord with the the components shown in the circuits and a micro controller.

1 Future Work

Throughout this paper, the drift problem identified in memristor model, causes it to be unsuited for modeling memristor behaviour over time. Weather this is caused by the "Voff" and "Von" needs further investigation, might suggest that different parameters settings can eliminate or minimize the drift. A quantitative analysis might be the most fitting approach.

If an optimal parameter setting can not be found, reworking the model might be required to eliminate the drift while still matching SDC memristor behaviour. Alternatively exploring other models such as the rest in [44], the one described in [43], and the one described in [45, 47]; or creating a new model will be a great benefit when simulating and designing more complex or time dependent systems.

The published paper [40] reports an error rate in the read of 25%. Work to improve the reliability of this system will be critical practical applications of any memristor memory controller. Forms of validation hardware are therefor needed.

Due to variations between memristors, a system that uses rigid pulses in both duration and amplitude might not store and read the same value when applied to the two different memristors, even if they are of the same type. This needs to be verified. A possible solution is to use shorter pulses that only nudge the memristance in the correct direction and have a feedback to verify that it for to the right state, and if not repeat the pulses until it is in a correct state. Another approach could be to read the memristance while writing to the memristor with a smaller amplitude signal, and stop the signal when the memristor reaches the desired state, is this feasible? What approach is fastest and most accurate?

Another aspect worth while exploring is to look at the window comperator of the second circuit seen in figure 5.20. As it stands the leakage due to the voltage divider is quite high, this will cause unnecessary heat production and will drain the battery of any mobile device. Exploring ways to hold the output in a ternary manner, preferably while sticking to positive voltages would be a great gain. One approach worth investigating to achieve this might look like the one demonstrated in [57]. As is is demonstrated it would need modifications, but would it be possible to use ground as the low threshold and another voltage as reference to achieve an all positive ternary output? If it is not, it might still be worth looking into as the approach uses only one op-amp, which would reduce the component count and possibly price of the circuit.

A third and critical avenue to investigate is multi memristor circuits, this is needed for

any storage solution more complex then storing simple commands like "left", "straight, and "right" for example in robotics. There are several aspects that need investigation to create a proper memory controller, the system needs to identify and work around damaged memristors, ways to use the read and write circuitry across multiple memristors without unintentionally corrupting the data stored in others in the same system (for example in a crossbar array).

With the previous avenues explored, applying the memristor based ternary memory, and logic to robotics is exiting. Developing a physical battery powered robot capable of receiving wireless instructions, storing them in ternary memory, and taking actions based on the stored data. How would the system perform with respect to power consummation, response time and complexity compared with a binary transistor system?

Can memristor based logic, such as described in [58, 59] be applied to ternary logic? If it can be, how does it compare to equivalent binary solutions, and to other ternary logic solutions? Investigations into power efficiency, component count, durability and much more is interesting here.

For some applications where small errors are not critical, an investigation into a modified version of the method presented in [52] would be very interesting. Is it possible to store multiple trits in one memristor using a series of logical pulses (for example +0.6V for logical "+", +0.4V for logical "0", and -0.25V for logical "-")? What would be needed to make such a system reliably store data? Is it even feasible?

Replicate, explore, and utilize the memristor based logic circuits described in [58, 59]. Explore the use for robotics applications, and investigate ternary logic feasibility.

Bibliography

[1] "USN Ternary Research Group - USN Ternary Research Group." [Online]. Available: https://ternaryresearch.com/

[2] H. Gundersen and S. Bos, "Ternary Computing ? Its Status and its Future," *Unpublished*, 2020.

[3] H. Gundersen, "FROM CMOS to CNTFET," 2019.

[4] S. Bos, H. Gundersen, and F. Sanfilippo, "uMemristorToolbox : Open source framework to control memristors in Unity for ternary applications," *International Symposium on Multiple-ValuedLogic (ISMVL)*, vol. 50th, pp. 1–6, 2020.

[5] I. ENGDAL, Henning GUNDERSEN, and Kiran RAJA, "Ternary Computing Assisted Secure Access Control," Ph.D. dissertation, University of South Eastern Norway - USN, 2019.

[6] Electric Videos, "What is a CMOS? [NMOS, PMOS] - YouTube," 2016. [Online]. Available: https://www.youtube.com/watch?v=docgmTprR5o

[7] P. Mathys, "CMOS Transistors - YouTube," 2014. [Online]. Available: https://www.youtube.com/watch?v={_}SwY-WfWTQo

[8] D. B. Strukov, G. S. Snider, D. R. Stewart, and R. S. Williams, "The missing memristor found," *Nature*, vol. 453, no. 7191, pp. 80–83, 2008.

[9] M. H. Moaiyeri, R. F. Mirzaee, A. Doostaregan, K. Navi, and O. Hashemipour, "A universal method for designing low-power carbon nanotube FET-based multiple-valued logic circuits," *IET Computers and Digital Techniques*, vol. 7, no. 4, pp. 167–181, 2013.

[10] G. Hills, C. Lau, A. Wright, S. Fuller, M. D. Bishop, T. Srimani, P. Kanhaiya, R. Ho, A. Amer, Y. Stein, D. Murphy, Arvind, A. Chandrakasan, and M. M. Shulaker, "Modern microprocessor built from complementary carbon nanotube transistors," *Nature*, vol. 572, no. 7771, pp. 595–602, aug 2019. [Online]. Available: http://dx.doi.org/10.1038/s41586-019-1493-8

[11] R. Martel, T. Schmidt, H. R. Shea, T. Hertel, and Ph. Avouris, "Single- and multi-wall carbon nanotube field-effect transistors," *Applied physics letters*, vol. 73, no. 17, pp. 2447–2449, 1998.

[12] S. J. Wind, J. Appenzeller, R. Martel, V. Derycke, and P. Avouris, "Vertical scaling of carbon nanotube field-effect transistors using top gate electrodes," *Applied Physics Letters*, vol. 80, no. 20, pp. 3817–3819, 2002.

[13] Z. Chen, D. Farmer, S. Xu, R. Gordon, P. Avouris, and J. Appenzeller, "Externally assembled gate-all-around carbon nanotube field-effect transistor," *IEEE Electron Device Letters*, vol. 29, no. 2, pp. 183–185, 2008.

[14] S. Kim, T. Lim, and S. Kang, "An optimal gate design for the synthesis of ternary logic circuits," *Proceedings of the Asia and South Pacific Design Automation Conference, ASP-DAC*, vol. 2018-January, no. January, pp. 476–481, 2018.

[15] M. H. Moaiyeri, A. Doostaregan, and K. Navi, "Design of energy-efficient and robust ternary circuits for nanotechnology," *IET Circuits, Devices and Systems*,

vol. 5, no. 4, pp. 285–296, 2011.

[16] S. Lin, Y. B. Kim, and F. Lombardi, "A novel CNTFET-based ternary logic gate design," *Midwest Symposium on Circuits and Systems*, no. 1, pp. 435–438, 2009.

[17] S. Y. Lee, S. Kim, and S. Kang, "Ternary Logic Synthesis with Modified Quine-McCluskey Algorithm," *Proceedings of The International Symposium on Multiple-Valued Logic*, vol. 2019-May, pp. 158–163, 2019.

[18] H. N. Risto. (2020, September) Automated synthesis of netlists for ternary-valued n-ary logic functions in cntfet circuits. [Online]. Available: https://doi.org/10.5281/zenodo.4015574

[19] D. Etiemble, "Ternary circuits: why R=3 is not the Optimal Radix for Computation," *ArXiv*, no. 1908.06841, 2019. [Online]. Available: http://arxiv.org/abs/1908.06841

[20] D. W. Jones, "The Ternary Manifesto," 2012. [Online]. Available: http://homepage.divms.uiowa.edu/{~}jones/ternary/

[21] N. P. Brousentsov, S. P. Maslov, R. A. J, and E. A. Zhogolev, "DEVELOPMENT OF TERNARY COMPUTERS AT MOSCOW STATE UNIVERSITY." [Online]. Available: https://computer-museum.ru/english/setun.htm

[22] O. &. retro Computers, "The Bit Bucket - 2 - Setun (soviet computers part 2) - YouTube," 2017. [Online]. Available: https://www.youtube.com/watch?v=a2nrCrNDAHk

[23] P. Taxel, "File:Two-terminal non-linear circuit elements.svg - Wikimedia Commons." [Online]. Available: https://commons.wikimedia.org/wiki/File:Two-terminal{_}non-linear{_}circuit{_}elements.svg

[24] L. Chua, "Memristor-The missing circuit element," *EEE TRANSACTIONS ON CIRCUIT THEORY*, vol. Ct-18, no. 5, pp. 507–519, 1971.

[25] ——, "The Chua Lectures, Part 2: Everything You Wish to Know About Memristors but are Afraid to Ask - YouTube," 2016. [Online]. Available: https://www.youtube.com/watch?v=iyYfAiZ9wrM

[26] Y. N. Joglekar and S. J. Wolf, "The elusive memristor: Properties of basic electrical circuits," *European Journal of Physics*, vol. 30, no. 4, pp. 661–675, 2009.

[27] L. Chua, "The Chua Lectures, Part 1: Everything you wish you knew about Memristors," 2016. [Online]. Available: https://www.youtube.com/watch?v=Rg3Vsc515YY

[28] ——, "The Chua Lectures, Part 3: 10 Things You Didn't Know about Memristors - YouTube," 2016. [Online]. Available: https://www.youtube.com/watch?v=IV1OW{_}MN5Jc

[29] M. Kikuchi, M. Saito, H. Okushi, and A. Matsuda, "Polarized (letter '8') memory in cdse point contact diodes," *Solid State Communications*, vol. 9, no. 10, pp. 705 – 707, 1971. [Online]. Available: http://www.sciencedirect.com/science/article/pii/003810987190250X

[30] L. Chua, "10 Things you didn't know about memristors," *Optics InfoBase Conference Papers*, vol. Part F77-I, 2017.

[31] S. Vongehr and X. Meng, "The missing memristor has not been found," *Scientific Reports*, vol. 5, no. July, 2015. [Online]. Available: http://dx.doi.org/10.1038/srep11657

[32] B. P. Abbott and et.al, "Observation of gravitational waves from a binary black hole merger," *Physical Review Letters*, vol. 116, no. 6, pp. 1–16, 2016.

[33] I. Abraham, "The case for rejecting the memristor as a fundamental circuit element," *Scientific Reports*, vol. 8, no. 1, pp. 1–9, 2018.

[34] L. Chua, "The Chua Lectures, Part 4: 10 More Things You Didn't Know about Memristors - YouTube," 2016. [Online]. Available: https://www.youtube.com/watch?v=YcOtbbkaFRE{\&}t=314s{\&}ab{_}channel=HPETechnology

[35] K. A. Campbell, "Self-directed channel memristor for high temperature operation," *Microelectronics Journal*, vol. 59, no. November 2016, pp. 10–14, 2017. [Online]. Available: http://dx.doi.org/10.1016/j.mejo.2016.11.006

[36] L. Chua, "Resistance switching memories are memristors," *Applied Physics A: Materials Science and Processing*, vol. 102, no. 4, pp. 765–783, 2011.

[37] Z. Biolek, D. Biolek, and V. Biolková, "SPICE model of memristor with nonlinear dopant drift," *Radioengineering*, vol. 18, no. 2, pp. 210–214, 2009.

[38] Knowm, "Self Directed Channel Memristors," Knowm, Tech. Rep., 2019. [Online]. Available: https://knowm.org/downloads/Knowm{_}Memristors.pdf

[39] "M+SDC Memristor 16 Discrete 32 DIP – Knowm Inc," 2020. [Online]. Available: https://knowm.com/collections/frontpage/products/m-sdc-memristor-16-discrete-32-dip

[40] S. Bos, J. B. Nilsen, and H. Gundersen, "Post-Binary Robotics : Using Memristors With Ternary States for Robotics Control," in *ESTC*, 2020.

[41] T. Molter and A. Nugent, "The Mean Metastable Switch Memristor Model in Xyce – Knowm.org," 2017. [Online]. Available: https://knowm.org/the-mean-metastable-switch-memristor-model-in-xyce/

[42] ——, "The Mean Metastable Switch Memristor Model in Xyce – Knowm.org," 2017.

[Online]. Available: https://knowm.org/the-mean-metastable-switch-memristor-model-in-xyce/

[43] B. Mohammad, D. Homouz, and H. Elgabra, "Robust hybrid memristor-CMOS Memory: Modeling and design," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 21, no. 11, pp. 2069–2079, 2013.

[44] A. Nugent, "GitHub - knowm/memristor-models-4-all: This project aims to be a collaborative effort to collect under one roof all the memristor models published to date." 2019. [Online]. Available: https://github.com/knowm/memristor-models-4-all

[45] T. Wang and J. Roychowdhury, "Well-Posed Models of Memristive Devices," *CoRR*, vol. abs/1605.0, 2016. [Online]. Available: http://arxiv.org/abs/1605.04897

[46] T. Molter, "The Joglekar Resistance Switch Memristor Model in LTSpice – Knowm.org," 2016. [Online]. Available: https://knowm.org/the-joglekar-resistance-switch-memristor-model-in-ltspice/

[47] ——, "Well-posed Memristor Modeling with Xyce and Verilog-A – Knowm.org," 2016. [Online]. Available: https://knowm.org/well-posed-memristor-modeling-with-xyce-and-verilog-a/

[48] ——, "The Joglekar Resistance Switch Memristor Model in LTSpice – Knowm.org," 2016. [Online]. Available: https://knowm.org/the-joglekar-resistance-switch-memristor-model-in-ltspice/

[49] ——, "The Generalized Metastable Switch Memristor Model – Knowm.org," 2015. [Online]. Available: https://knowm.org/the-generalized-metastable-switch-memristor-model/

[50] M. A. Nugent and T. W. Molter, "AHaH computing-from metastable switches to

attractors to machine learning," *PLoS ONE*, vol. 9, no. 2, 2014.

[51] I. E. Ebong and P. Mazumder, "Self-controlled writing and erasing in a memristor crossbar memory," *IEEE Transactions on Nanotechnology*, vol. 10, no. 6, pp. 1454–1463, 2011.

[52] N. Taherinejad, P. D. Manoj, and A. Jantsch, "Memristors' potential for multi-bit storage and pattern learning," *Proceedings - EMS 2015: UKSim-AMSS 9th IEEE European Modelling Symposium on Computer Modelling and Simulation*, pp. 450–455, 2016.

[53] D. Radakovits and N. Taherinejad, "Implementation and Characterization of a Memristive Memory System," *2019 IEEE Canadian Conference of Electrical and Computer Engineering, CCECE 2019*, pp. 1–4, 2019.

[54] J. B. Nilsen, "Projects · Xarteck/XarteckTernaryResearch," 2020. [Online]. Available: https://github.com/Xarteck/XarteckTernaryResearch/projects

[55] T. Molter, "Memristor Models in LTSpice – Knowm.org," 2017. [Online]. Available: https://knowm.org/memristor-models-in-ltspice/

[56] A. Nugent, "Knowm Memristor Discovery Manual," Knowm.com, Tech. Rep., 2019.

[57] Craig, "Single OpAmp Window Comparators — Analog Zoo," 2015. [Online]. Available: http://www.analogzoo.com/2015/12/single-opamp-window-comparators/

[58] M. Arendall, "Simulating Knowm AHaH Memristor-Based Logic Using Qucs-S and Xyce – Knowm.org," 2017. [Online]. Available: https://knowm.org/simulating-knowm-ahah-memristor-based-logic-using-qucs-s-and-xyce/

[59] ——, "Comparing Xyce and JSpice Simulations of Knowm AHaH Memristor-Based Logic Using Qucs-S – Knowm.org," 2017. [Online]. Available: https://knowm.org/comparing-xyce-and-jspice-simulations-of-knowm-ahah-memristor-based-logic-using-qucs-s/

Appendix A

Post Binary Robotics Using Memristors With Ternary States for Robotics Control

Post-Binary Robotics: Using Memristors With Ternary States for Robotics Control

Steven Bos
Dept. of Science and Industry systems
University of South-Eastern Norway
Kongsberg, Norway
steven.bos@usn.no

Julian Breivold Nilsen
Dept. of Science and Industry systems
University of South-Eastern Norway
Kongsberg, Norway
juliannilsen@hotmail.com

Henning Gundersen
Dept. of Science and Industry systems
University of South-Eastern Norway
Kongsberg, Norway
henning.gundersen@usn.no

***Abstract*—This paper presents a method to read and write ternary (three-valued) signals on memristors to control a robotic actuator in real-time. The paper is a continuation of earlier work by [1] and implements a ternary memory controller for memristors in hardware. This post-binary approach with non-volatile memory is used to program a memristor as a "trit". The paper contributes to the state-of-the-art in memristor controlled robotics by reporting an entropy gain of log2(3)= 58% information at (20 versus 14)= 43% more component cost compared to binary. This advantage (eg. less wire complexity) increases when multi-trit architectures are considered.**

This article demonstrates both an LTspice simulation of the circuit and implementation with source code. The memristor programmer circuit writes a state to the memristor using a pattern of 100us pulses at 3 different amplitudes. The memristor read circuit sends 500 nA pulses and converts these using two reference resistors to three logic levels using an op amp window comparator. An Arduino Mega microcontroller ADC pin converts the analog output to a digital trit. Strenuous effort was made for predictable and replicable applied memristor research in pursuit of a post-binary robotics era. The multiple-valued circuit has safety features to prevent harm to the memristance state, standardized forming of new memristors and programming flexibility by sending patterns of different pulse amounts, pulse width and pulse amplitudes.

***Keywords*-multiple-valued logic, ternary computing, ternary storage, memristor ciruit, LTspice simulation**

I. Introduction

Since the discovery of the modern binary number system by Leibniz in 1679, numerous binary operating automata have been invented. Some of these were very practical, such as the weaving machine of Jacquard in 1804 leading to the first programmable binary machine using punch cards. Others were more theoretical, such as the EDVAC featuring a binary implementation of Alan Turings famous universal Turing machine. Other numeral systems for automata were also considered; the predecessor of the EDVAC, the ENIAC was base 10 and the first electrical ternary (base 3) computer, the Setun, dates back to 1958.

In a post-binary world a compute paradigm shift is needed. We see this slowly happening with new technology drivers based on memristors as multiple-valued storage and carbon nanotube field-effect transistors as elementary compute elements. Like many others before, we use the radix economy argument [2] to position ternary as the next compute paradigm.

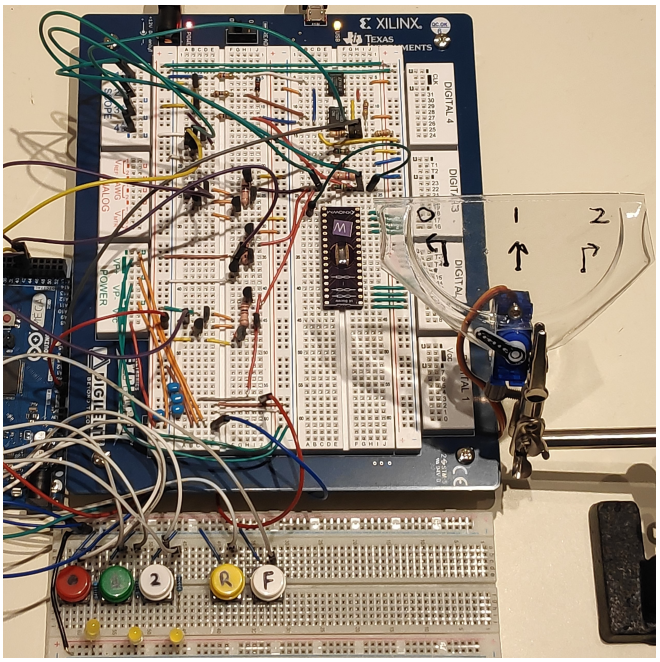


Fig. 1: Implementation of ternary valued memory circuit for reading and writing to memristors

Specifically, in this paper we address component count of a ternary memristor circuit in direct comparison to binary.

Academic discourse on memristor-based circuit designs is mostly proven with simulators [3] as memristive devices until a few years ago were not commonly available. In addition they are still costly, extremely sensitive and have a steep learning curve. The potential impact of three-valued memristor circuit for memory applications are significant, especially for robotics. Memristors require extremely small die space, little power and are fast enough for real-time actuator control. In addition, they can be stacked on top of the CMOS layer and thus decrease CPU-memory latency [4]. We show that both regular 30 FPS control signals can be send to the controller as well as irregular, interactive control signals.

II. Related work

When Hewlett Packard published about a physical implementation of a memristive device in 2008 [5], it stirred up the scientific field. Was the memristive device a memristor like the one Leon Chua hypothesized in 1971 [6]? It lead to still ongoing discussion on memory resistor scope [7], redefinitions of terms [8] and test to determine if the ideal memristor exist [9]. Regardless, systems of memory resistors, resistive memory (ReRAM) or variable resistors have proven to be extremely useful. A non exhaustive list of recent applications are memory systems [10], neural networks [11], robotics control [12] [13] and computing [14]. In computing for instance, a memristor might replace the classic transistor, augment it or both. Breaking the von Neumann architecture with crossbar in-memory computing [15], results in more parallelism and thus increased performance, reduced energy consumption and cost.

Both Moore's Law (doubling transistors in a die every 2 years) and Koomey's Law (halving battery consumption for a compute operation every 1,5 year) have trouble to keep up. Memristive circuit elements could extend both since they can be produced using traditional CMOS processes [10] and take much less die space eg. $2x2nm^2$ [16] than for instance 64-layer triple cell NAND flash memory. Although memristors can be made with many different memristive materials (see taxonomy [17]), limited energy consumption for reading and writing a stable state is a key property. The non-volatility property is another key property to reduce energy [10] as it doesn't require continuous power to retain its state, even after a read operation.

The holy grail of memristors are analog memory properties such as "multi bits". Traditional memory have 2 stable states, in this paper we experiment with 3 stable states (entropy of log2(3) = 1.58 bits or 58% gain), but research by Stathopoulos et al. [18] report 92 stable states (entropy of log2(92) =6.5 bits or 650% gain). The limits of memristors vary and depend on the memristive material. Clear is that temperature influences the memristive device for both initial forming of the device, reading and writing. When evaluating memristive circuit elements for implementations, stable state count, state retention, switching speed, energy consumption and random writes are key metrics to be measured at different temperatures, next to the physical die space and manufacturing cost. Most of these metrics are discussed in this paper, but measured only at room temperature.

Campbell [19] provides a great resource to program SDC memristors. She mentions two programming techniques for non-binary resistance states: DC compliance current limiting and pulsed operation. With regards to pulsed operation she mentions: "Consecutive pulsing can selectively place a device into a desired resistance range, either through consecutive erase or write pulses. This range can be selected by varying the number of pulses applied, the pulse width, and/or the pulse amplitude.". This gives memory controllers such as the one described in this paper a few parameters to tweak. Other examples of memristive circuits that explicitly publish about read / write control operations are [3] and [20] where the circuit is simulated.

A brief survey searching for "memristor" on the open source platform Github showed that several memristor simulation projects can be found. Most notably, a nice overview of memristor models for SPICE simulation is found in [21]. However, memristor control software to experiment with the physical memristive devices are scarce. We only found the Memristor-Discovery project by memristor manufacturer Knowm [22] while developing a Unity C# based port of it, uMemristorToolbox [1]. This port and implemented ternary memory controller was tailored for the Digilent Analog Discovery 2 (AD2) and used its waveform generator and scopes for writing and reading. In this paper we create our own waveform generator and scopes circuits and thus do not require an AD2 anymore.

III. LTspice simulation

A. Memristor model

The LTspice model used in this paper uses a modified version of the Knowm mean metastable switch model [21], [23], a spice implementation of the Xyce model of the same name [24]. The mean field simulation model is again a continuation of the generalized metastable switch memristor model [25], a model that was created because the creators found that existing models did not suit their need and wanted to address some problems caused by previous approaches to modeling [25], [26].

The metastable switch model is, like the real 16x1 SDC-W (Tungsten) memristor used in this paper produced by Knowm. Installation and basic use of the model is well documented in Knowm blog posts. With some tweaking of the parameters, the model was able to reproduce the results of the (binary) read and write pulse experiments, as those documented in Campbell's key paper from 2017 [27]. By adding a third power source and choosing the parameters as published in [1], ternary operations was also proven to be possible [28].

Although the model is good at simulating a memristor in short-term use cases there, it does not work accurately in long-term simulations. The model uses voltage to direct its state, causing rather rapid drift in the memristance towards a voltage dependent value; This can even be observed in figure 3 in the few ms between write and read pulses when there is no voltage or current over the memristor. Regular writes are needed to keep its state and thus non-volatility experiments cannot be run accurately. The model uses voltage to control its state. The read operations will influence the memristance state if the voltage is too high, hence our usage of 10 MΩ resistors to keep the read voltage low.

B. Read/Write experiment

The memristor parameter settings used in the simulation are as follows: ".params Ron=2500 Roff=125000 Voff=0.19 Von=0.52 TAU=0.00001 T=300 x0=0"

The LTspice write section of the circuit is controlled by PWL timed voltage sources that control the CMOS at the input pin of the circuit. The timing of the these sources can be found

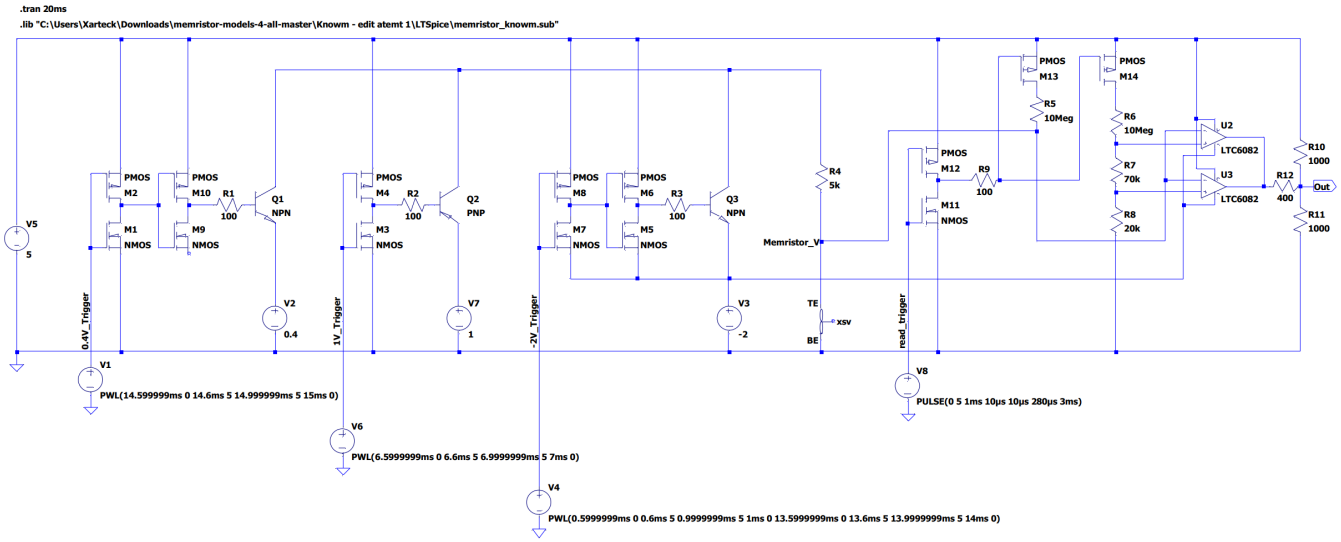


Fig. 2: The LTspice circuit used for the LTspice simulation. The circuit uses PWL voltage sources to control the input pins of the circuit

in Fig. 2. The voltage sources PWL patterns mimic the output pins of a microcontroller.

The read section is controlled by a pulse generator generating a 280 µs square wave plus a 10 µs rise and 10 µs fall (just like in the circuit implementation below) with a 3 ms period. Note that this read pattern (dark blue line) results in two read actions after after a logical 0 (write -2 V, turquoise line), then three read actions after a logical 2 (write +1 V, gold line) and finally two read actions after a reset (a logical 0 pulse) followed by a logical 1 (write +0.4 V green line).

C. Results

The output of the circuit, the upper part of Figure 3 shows three signals. The current through the memristor (pink line, in µA) show 4 pulses, a -15 µA (logical 0), 110 µA (logical 2), another -15 µA and a +15 µA (logical 1). Note that in 3 an additional logical 0 is written first when writing a logical 1 when the previous state was a logical 2, as no negative voltage source has been implemented for this state change. The additional pulse is not needed when the previous state was the logical 0 state. The resistance across the memristor (red line, in kΩ) show the drift behavior that is present in the simulated memristor but not in the actual memristor. This is clearly visible after writing a logical 1 or 2 where the line immediately converges to logical 0 over time. Although the drift velocity decreases with shorter pulses, the reached memristrance state is also lower, hence closer to the switch point of the op amp [28]. Some undesired state change over time in actual memristors has been reported as leakage current [29]. Finally,the op amp output (green line, in V) shows how the op amp is affected by the resistance across the memristor as well as natural drifting of the op amp after a read. The op amp has three levels set by the two reference resistors a logical 0@0V, logical 1@0.8V and logical 2@3.9V.

IV. Breadboard Implementation

The software implementation to control the Knowm SDC memristors for ternary operation created in [1] used the Digilent Analog Discovery 2 to read and write. The benefit of this solution is great flexibility in memristor programming, ease of use for novice memristor developers with safety features to prevent "burning" and a controlled environment where experiments can easily be repeated and compared. For memristors to be embedded in robotic applications a smaller hardware footprint is needed and hence the work in this paper to construct both a (square) waveform generator and oscilloscope.

We started our design process with replicating the memristor controller for binary operation by [29]. After simulation in National Instrument's Multisim Live with a variable resistor to mimic a resistor, we added a third voltage source for writing the middle state with a NPN transistor (BC547B). We use the BC557A as PNP transistor in the +1 V bridge. The voltage bridge is driven by two MOSFETS (ZVP4424A PMOSFET and 2N7000 NMOSFET). We added a second op amp (LT1012CN8) to create an op amp window comparator with two reference resistors. Since the Arduino Mega microcontroller ADC (analog) pin only accepts positive voltage, we added a voltage divider circuit to shift the output. Finally, we added CMOS inverters for the +0.4 V source, -2 V source and read circuit such that all voltage sources are isolated from the memristor by default when powering the circuit. We also added a 5 kΩ series load resistor in front of the memristor to protect it when writing to it as described in the Knowm documentation and Campbell's reference paper [27]. The Knowm documentation also reversed the orientation of the memristor in their latest reference wiring schematic such that the "anode" side is directly connected to ground. Lastly we changed the 1 MΩ resistors in [29] to 10 MΩ resistors

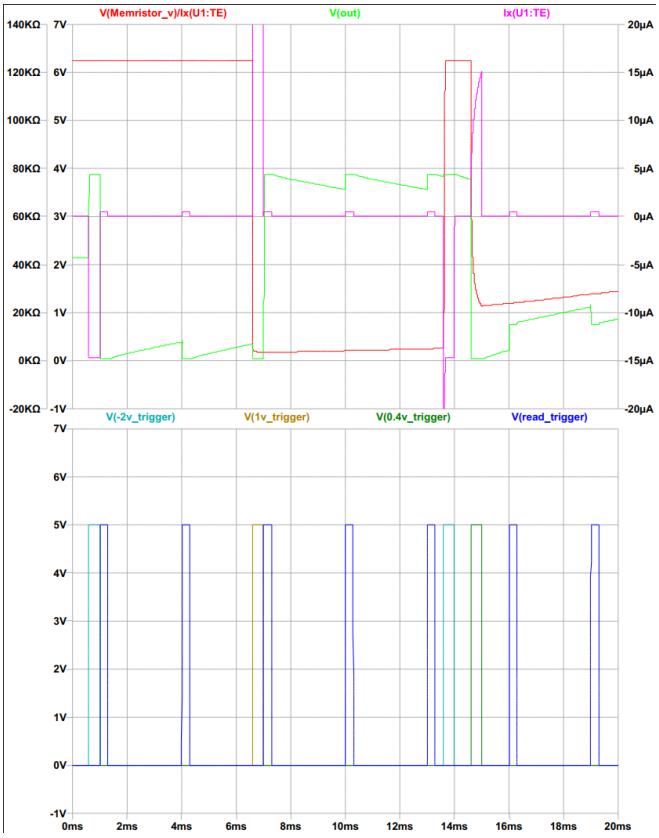


Fig. 3: The LTspice graph from the simulation of Figure 2. The top graph shows op amp output (green) computed memristance (red) and current over memristor (purple). The bottom graph shows the input pins: read (green), logical 0 (light blue), logical 1 (dark green) and logical 2 (yellow).

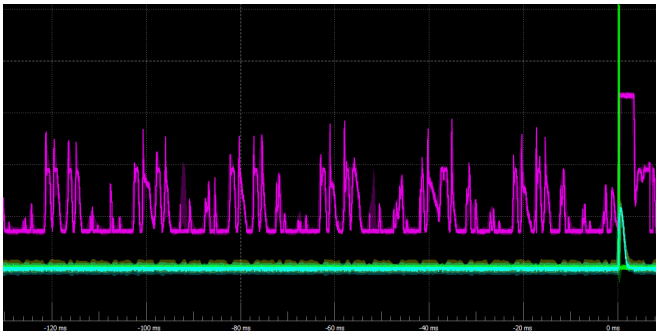


Fig. 4: A read pulse showing correct memristance of logical 2 state after 130ms of inactivity. The op amp (purple) only shows correct state after a read trigger (green)

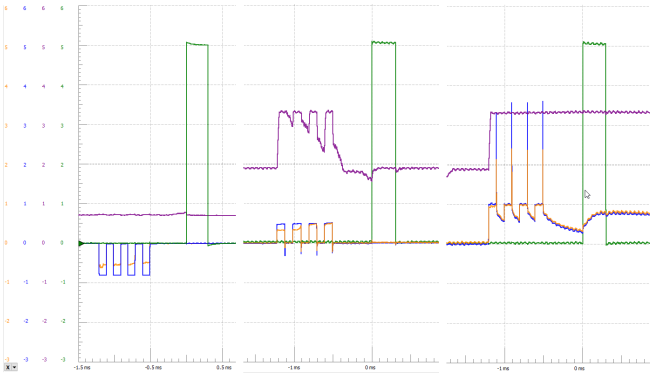


Fig. 5: All three write operations and read operations on a Knowm SDC-W memristor showing a clear voltage ceiling when driving it to logical states (yellow). The op amp output (purple) is triggered by a read pulse (green).

to further reduce the effects of read operations. For ternary operation this is critical as a read operation overwrites a logical 0 state to a logical 1 state when using a 1 MΩ resistor. This read effect can also be simulated in LTspice. We choose 20 kΩ and 70 kΩ reference resistors to separate the resistance spectrum in three regions (see Fig. 5):

- Logical 2: < 25 kΩ (3.31V@op amp)
- Logical 1: 25 kΩ <= x <= 120 kΩ (1.87V@op amp)
- Logical 0: > 120 kΩ (0.68V@op amp)

The op amp voltage levels closely resemble the simulated LTspice results. The reference resistors are important and dependent on the dynamic range of the memristor. We choose 25 kΩ as all our memristors were able to reach this threshold unless they were *burned*. For example, [29] report an *on* resistance of 6.8 kΩ, 16.7 kΩ and 31.7 kΩ when driven to lowest resistance level and an *off* resistance of >125 kΩ for all three memristors when driven to the highest level. We suspect that with a standardized forming method, standardized read/write operations, a series load resistor before the memristor and future memristor manufacturing, the dynamic range of the memristor becomes more predictable. We prototyped on the Digilent Electronics Explorer board since it conveniently provides three out of the required four power sources as well as four scopes for data visualisation. We filtered DC noise with 10 uF MLCC 50 V capacitors on all four power lines. A standard off-the-shelf Arduino Mega microcontroller was used to control the I/O pins and for its ADC feature to read the op amp output. If we compare the bare minimum ternary controller in comparison to a bare minimum binary controller in [29]: 1 NPN transistor, 1 opamp with 1 reference resistor, 1 NPN gate protection resistor and two MOSFETS are needed extra. This is in increase of +43% (20 vs 14 components) while the information gain is Log2(3) = +58%, and thus significantly larger than the increased costs.

A. *Debugging the circuit*

We added 3 resistors on the breadboard (5 kΩ, 33 kΩ and 200 kΩ to mimic three memristance states. This minimized burning memristors while our circuit was still in development. We used all four scopes of the Digilent Electronics Explorer board: read trigger, op amp output, voltage before/after reference resistor.

B. Interface

A five button interface has been created to manually interact with the memristor controller. Three buttons are used to program a logical 0,1 or 2. One button is dedicated for a single read operation and one button is dedicated for forming a new memristor with a high-low pattern of a single 5 ms logical 2 pulse, 5 ms idle, followed by 5 ms logical 0 and 5 ms idle. This is repeated 20 times. We added three LEDs to the interface board to directly show Arduino ADC output.

C. Robotic control use case: irregular control signals for robotic steering

In this use case we manually steer a power conscious robot, represented as a small robotic actuator. The actuator rotates between -60°(left, logical 0), 0 (straight, logical 1) and +60°(right, logical 2) after each read operation. Energy is saved by doing irregular read actions done manually. In one recording of 81 random writes and 321 interleaved reads we recorded 64 incorrect reads (25%), of which the majority was caused by incorrect logical 1 states. This shows that further research is needed to better differentiate that state. The test has been recorded in [30].

V. Discussion

Working with memristors is still not a trivial matter as they are highly sensitive to program. We have switched the 20 KΩ for a 10 KΩ reference resistor to reduce errors and suspect that neither is optimal. The logical 1 state is often seen as a logical 0. Exploration of optimal write 1 voltage level and pulse pattern as well as reference resistors is needed. We burned three memristors (significantly reduced their dynamic memristance range) by accidentally exposing it to high voltage spikes repeatedly, compared to just one when experimenting with the Digilent Analog Discovery 2 (AD2). We have not found much literature on recognizing malfunctioning or "de-sensitized" memristors. As a sanity check, we connected a suspected burned memristor to the AD2 and tried to produce a pinched hysteris loop, the hallmark pattern of a memristor. With the result in this paper we can now compare pulsed operation with waveform operation. Our preliminary results indicate improved operation with less energy to switch states and higher switching speed, while (high) state retention and state correctness being similar to [1]. The largest development issues were controlling the voltage levels and smoothing the spikes since they affect the memristance, see Fig. 6.

The voltage shift circuit is not an ideal solution for providing only positive op amp output voltage to Arduino's analog ADC pins. It consumes significant current while other solutions exist, such as described in [31]. It was needed to offset a little bias in the op amp, preventing it to sink its voltage below the threshold for a logical 0. Interestingly, the LTspice simulation did not show this bias, while every other aspect was a good prediction of actual memristor performance. Possibly the use of generic components is the root cause of this anomaly.

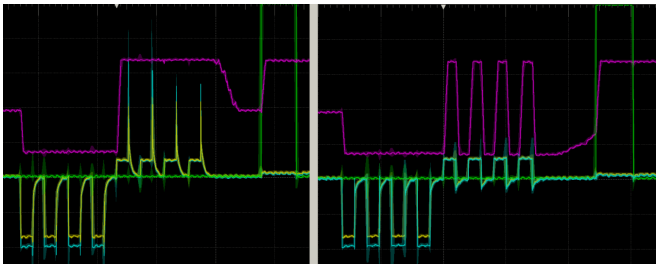


Fig. 6: One of the issues with 100 µs +0.4V pulses we fixed were high spikes overwriting a logical 1 with logical 2. This test was done with a 5k resistor resembling a memristor programmed at logical 2 to prevent damage to the actual device, hence the incorrect read pulses

```
Write 1 (write 4 pulses 0, then 4 pulses 1
[CORRECT] Read logical: 1 ( 1.86523 V.) Groundtruth = 1
[CORRECT] Read logical: 1 ( 1.86035 V.) Groundtruth = 1
[CORRECT] Read logical: 1 ( 1.86523 V.) Groundtruth = 1
Write 2 (write 4 pulses 2)
[CORRECT] Read logical: 2 ( 3.29102 V.) Groundtruth = 2
[CORRECT] Read logical: 2 ( 3.30566 V.) Groundtruth = 2
[CORRECT] Read logical: 2 ( 3.30566 V.) Groundtruth = 2
[CORRECT] Read logical: 2 ( 3.29102 V.) Groundtruth = 2
```

Fig. 7: Arduino terminal output showing correct ADC after irregular reads

VI. Conclusion

In this paper we present a complete LTspice simulation and matching breadboard implementation to control a memristor for ternary valued (0,1,2) operation. We show one use case with 3-valued robotic control signals. In that use case irregular control signals are executed with a random, irregularly timed write (81 times) and repeated irregular read signals (257 times) over a period of a nearly a minute. We report [64/257=25%] read errors, clearly indicating that this work should be considered as the first generation. This paper has three key contributions:

1) LTspice simulation of a ternary memristor controller, source: [32]
2) breadboard implementation of a ternary memristor controller including firmware, source: [32]
3) Early results of using the ternary memristor controller for irreguar robotic control signals

With the work described in this paper we demonstrate the feasibility of a new direction for robotics we call "post-binary robotics" (PBR). Robotic control with memristors has been proposed before eg. [12] [13]. In PBR, we only evaluate multiple-valued solutions that cost the same or less compared to binary solutions from the very start. This work briefly addresses the radix economy and complexity argument that have long favoured binary operation over multiple-valued. Scaling and maturing this technology to multi-trit architectures enable an era where even smaller and more energy efficient robots could be achieved.

VII. Future Work

Throughout this paper various shortcomings in simulation and implementation have been identified that should be addressed. This is the first iteration of a ternary memristor controller and results are under active research as no longitudinal studies have been performed yet. We can only refer to the inventor of the Knowm SDC-W memristor, [19], where write endurance of 1 billion operations is claimed. A logical step towards embedded memristor controllers is further miniaturization to PCB and eventually IC level. For practical applications, multiple memristors must be accessible in parallel. Also, for non-volatile memory applications, power, battery and refresh circuits (as found in SSD technology) is needed. Finally, since programming memristors is a non-deterministic activity, some form of state validation in hardware is needed.

VIII. Acknowledgment

This research was fully funded by the University of South-East Norway (USN). We thank Dr. Mehdi Azadmehr for his feedback on our circuit simulation. We thank the members of SMART research group, Cyber Physical and Applied Smart Systems research group, and the Ternary Research Group for their feedback on this paper.

References

[1] S. Bos, H. Gundersen, and F. Sanfilippo, "umemristortoolbox: Open source framework to control memristors in unity for ternary applications," in *2020 IEEE 50th International Symposium on Multiple-Valued Logic (ISMVL)*, 2020.

[2] B. Hayes, "Computing science: Third base," *American Scientist*, vol. 89, no. 6, pp. 490–494, 2001.

[3] N. TaheriNejad and D. Radakovits, "From behavioral design of memristive circuits and systems to physical implementations," *IEEE Circuits and Systems Magazine*, vol. 19, no. 4, pp. 6–18, Fourthquarter 2019.

[4] D. B. S. et al, "Hybrid cmos memristor circuits," *2010 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 1967–1970, 2010.

[5] D. R. S. D. B. Strukov, G. S. Snider and R. S. Williams, "The missing memristor found," *Nature*, vol. 453, no. 7191, pp. 80–83, 2008.

[6] C. L., "Memristor - the missing circuit element," *IEEE Transactions on Circuit Theory*, vol. 18, no. 5, pp. 507–519, 1971.

[7] S. Vongehr, "The missing memristor has not been found," *Scientific Reports*, vol. 5, p. 11657, 06 2015.

[8] L. Chua, "Resistance switching memories are memristors," *Applied Physics A*, vol. 102, pp. 765–783.

[9] Y. V. Pershin and M. D. Ventra, "A simple test for ideal memristors," *Journal of Physics D: Applied Physics*, vol. 52, no. 1, p. 01LT01, 2018.

[10] B. Mohammad, D. Homouz, and H. Elgabra, "Robust hybrid memristor-cmos memory: Modeling and design," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 21, pp. 2069–2079, 2013.

[11] A. Thomas, "Memristor-based neural networks," *Journal of Physics D: Applied Physics*, vol. 46, no. 9, p. 093001, feb 2013.

[12] A. Ascoli, D. Baumann, R. Tetzlaff, L. O. Chua, and M. Hild, "Memristor-enhanced humanoid robot control system – part i: Theory behind the novel memcomputing paradigm," *International Journal of Circuit Theory and Applications*, vol. 46, no. 1, pp. 155–183, 2018. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/cta.2431

[13] T. Li, S. Duan, J. Liu, L. Wang, and T. Huang, "A spintronic memristor-based neural network with radial basis function for robotic manipulator control implementation," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 46, no. 4, pp. 582–588, April 2016.

[14] S. Kvatinsky, D. Belousov, S. Liman, G. Statat, N. Wald, E. Friedman, A. Kolodny, and U. Weiser, "Magic—memristor-aided logic," *IEEE Trans. Circuits Syst. II, Express Briefs*, vol. 61, no. 11, pp. 895–899, 11 2014.

[15] E. Linn, R. Rosezin, S. Tappertzhofen, U. Böttger, and R. Waser, "Beyond von neumann—logic operations in passive crossbar arrays alongside memory operations," *Nanotechnology*, vol. 23, no. 30, p. 305205, jul 2012.

[16] S. Pi, C. Li, H. Jiang, W. Xia, H. Xin, J. J. Yang, and Q. Xia, "Memristor crossbars with 4.5 terabits-per-inch-square density and two nanometer dimension," 04 2018.

[17] M. Zidan, A. Chen, G. Indiveri, and W. Lu, "Memristive computing devices and applications," *J Electroceram*, vol. 39, pp. 4–20, 2017.

[18] S. Stathopoulos, A. Khiat, M. Trapatseli, S. Cortese, A. Serb, I. Valov, and T. Prodromakis, "Multibit memory operation of metal-oxide bi-layer memristors," *Sci Rep*, vol. 7, p. 17532, 2017.

[19] K. A. Campbell, "Self-directed channel memristor for high temperature operation," *Microelectronics journal*, vol. 59, pp. 10–14, 2017.

[20] P. Ebong, I. E. ; Mazumder, "Self-controlled writing and erasing in a memristor crossbar memory," *IEEE Transactions on Nanotechnology*, vol. 10, no. 6, pp. 1454–1463, 2011.

[21] K. Inc. (2019) Memristor models github project. [Online]. Available: https://github.com/knowm/memristor-models-4-all

[22] ——. (2019) Memristor discovery github project. [Online]. Available: https://github.com/knowm/memristor-discovery

[23] T. Molter, "Memristor Models in LTSpice – Knowm.org," 2017. [Online]. Available: https://knowm.org/memristor-models-in-ltspice/

[24] T. Molter and A. Nugent, "The Mean Metastable Switch Memristor Model in Xyce – Knowm.org," 2017. [Online]. Available: https://knowm.org/the-mean-metastable-switch-memristor-model-in-xyce/

[25] T. Molter, "The Generalized Metastable Switch Memristor Model – Knowm.org," 2015. [Online]. Available: https://knowm.org/the-generalized-metastable-switch-memristor-model/

[26] T. Wang and J. Roychowdhury, "Well-Posed Models of Memristive Devices," *CoRR*, vol. abs/1605.0, 2016. [Online]. Available: http://arxiv.org/abs/1605.04897

[27] K. A. Campbell, "Self-directed channel memristor for high temperature operation," *Microelectronics Journal*, vol. 59, no. November 2016, pp. 10–14, 2017. [Online]. Available: http://dx.doi.org/10.1016/j.mejo.2016.11.006

[28] J. B. Nilsen, "On ternary valued memristors and simulation (to be published)," Master's thesis, USN, Norway, 2020.

[29] D. Radakovits and N. TaheriNejad, "Implementation and characterization of a memristive memory system," in *2019 IEEE Canadian Conference of Electrical and Computer Engineering (CCECE)*, 2019, pp. 1–4.

[30] H. G. Steven Bos, Julian Nilsen, "Demonstration of irregular read and write of our ternary memristor controller circuit," https://youtu.be/RcDCvRd7P0Q, August 2020.

[31] N. Taherinejad, M. P. D. Sai, M. Rathmair, and A. Jantsch, "Fully digital write-in scheme for multi-bit memristive storage," in *2016 13th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE)*, 2016, pp. 1–6.

[32] S. Bos. (2020, Augustus) Ternarymemristorcontroller. [Online]. Available: https://github.com/aiunderstand/TernaryMemristorController.git

Appendix B

SIMULATION AND CIRCUIT DATA

1 Memristor Model

In the next three sections, simulation results using a few of the LTspice memristor models out there will be tested and graphed. The parameters of the memristor models are unchanged from their base setting in these graphs.

1.1 Knowm Mean Metastable Switch Memristor Model

The LTspice model used, uses a modified version of the Knowm mean metastable switch model [44, 55], a spice implementation of the Xyce model of the same name [41]. The mean field simulation model is again a continuation of the generalized metastable switch memristor model [49], a model that was created because the creators found that existing models did not suit their need and wanted to address some problems caused by previous approaches to modeling [45, 49].

Base Knowm mean memristor model, no resistor, 0.5V sine wave, no offset

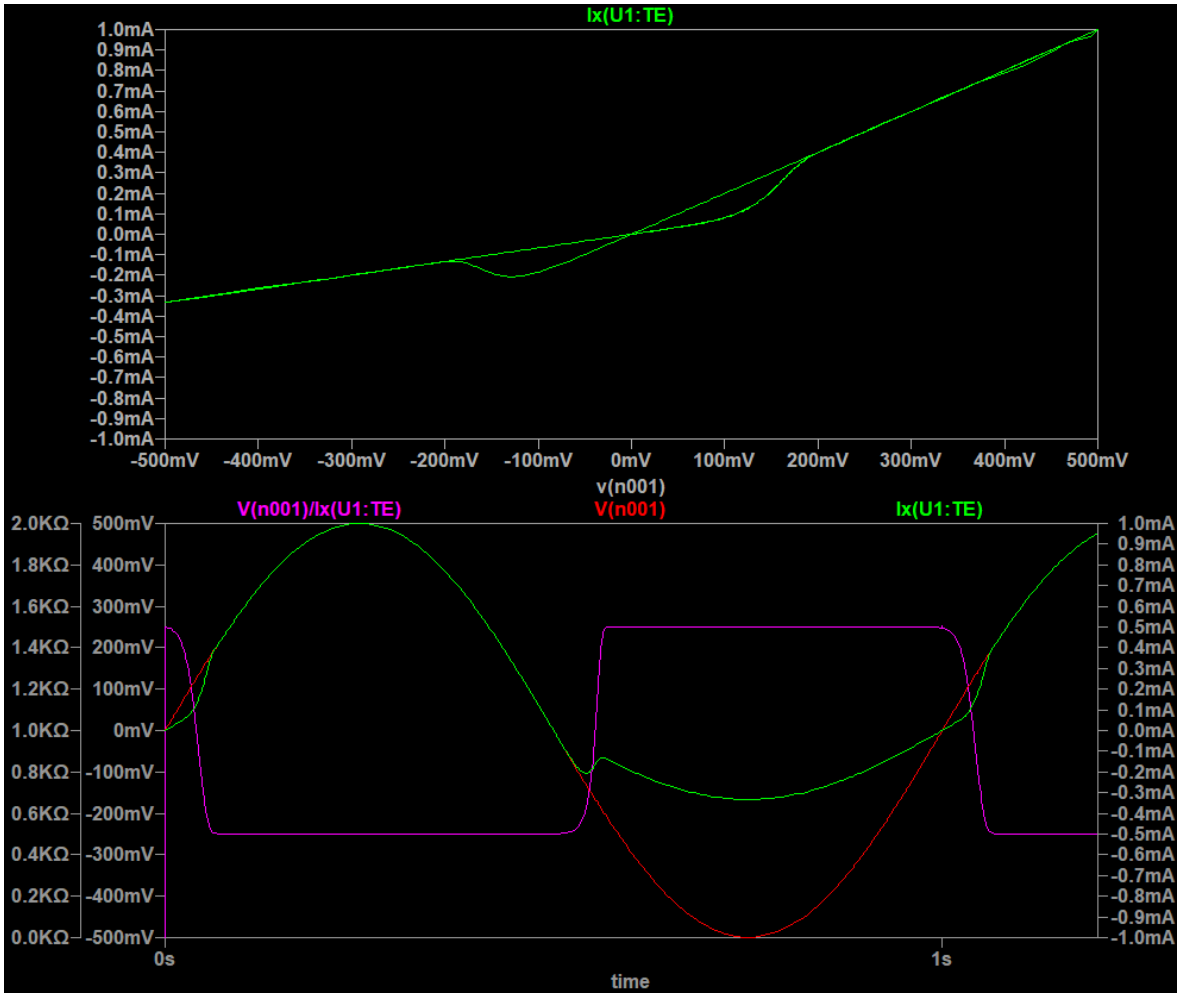


Figure B.1: Base Knowm mean memristor model, no resistor, 1Hz 0.5V sine wave, no offset.

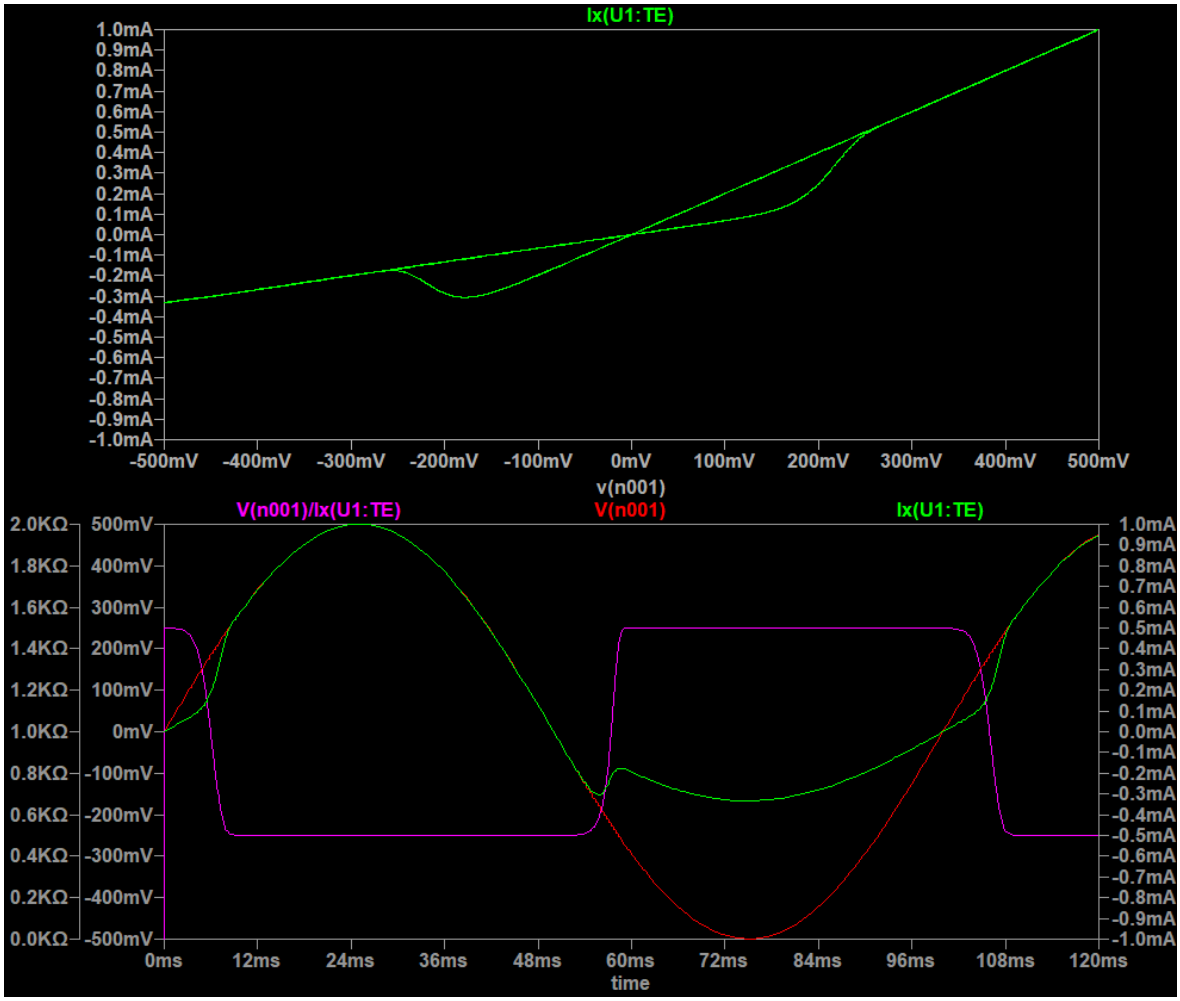


Figure B.2: Base Knowm mean memristor model, no resistor, 10Hz 0.5V sine wave, no offset.

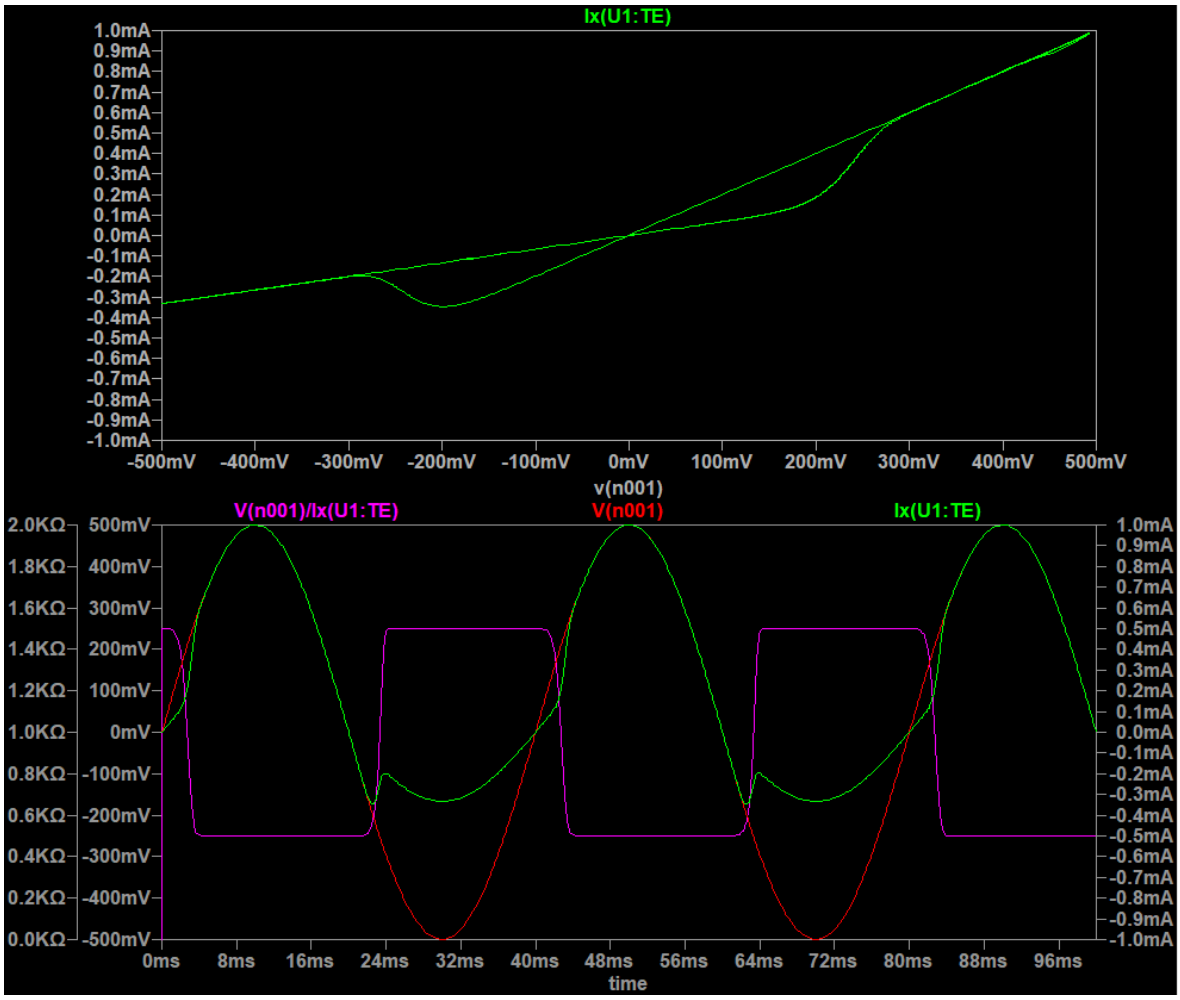


Figure B.3: Base Knowm mean memristor model, no resistor, 25Hz 0.5V sine wave, no offset.

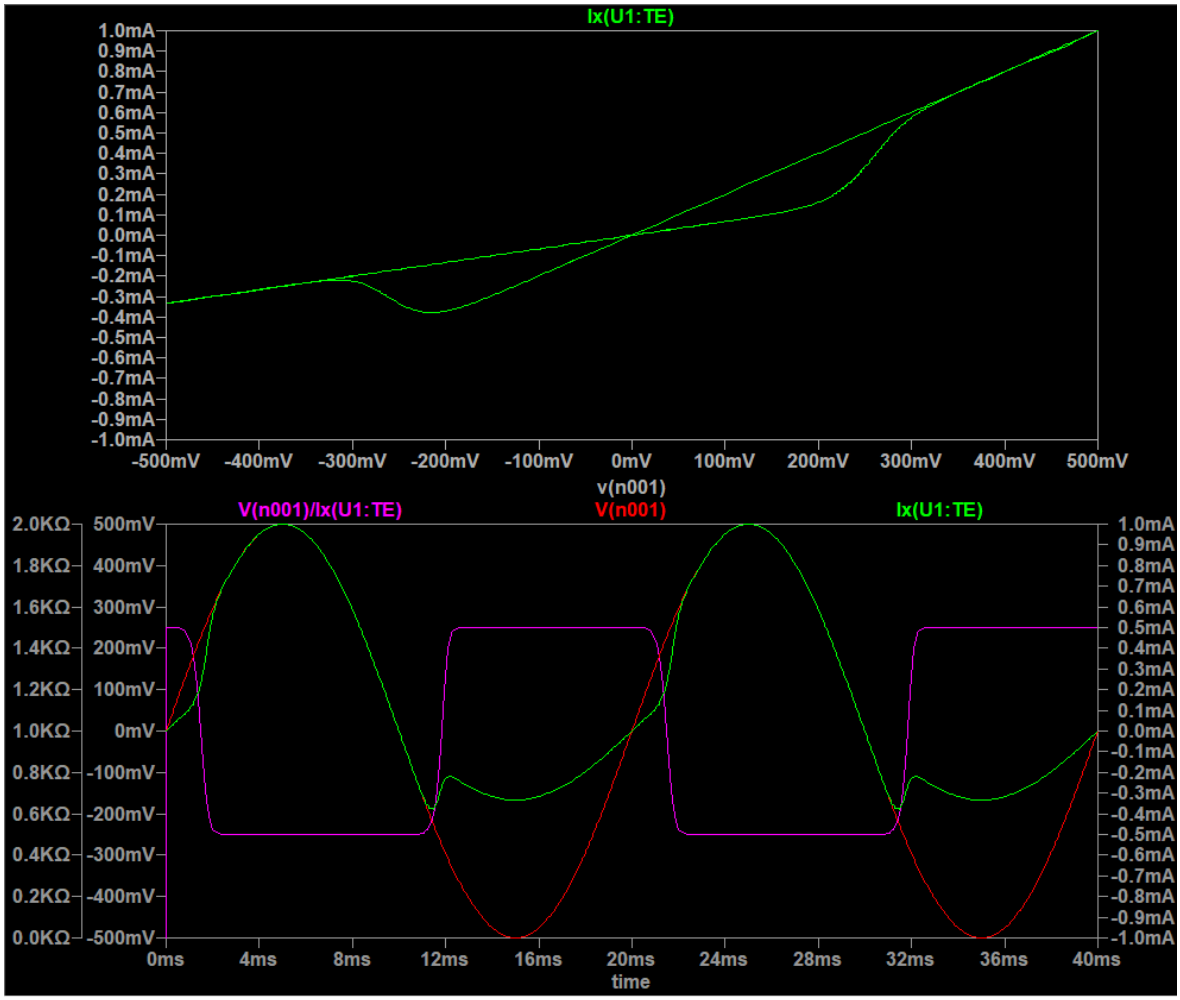


Figure B.4: Base Knowm mean memristor model, no resistor, 50Hz 0.5V sine wave, no offset.

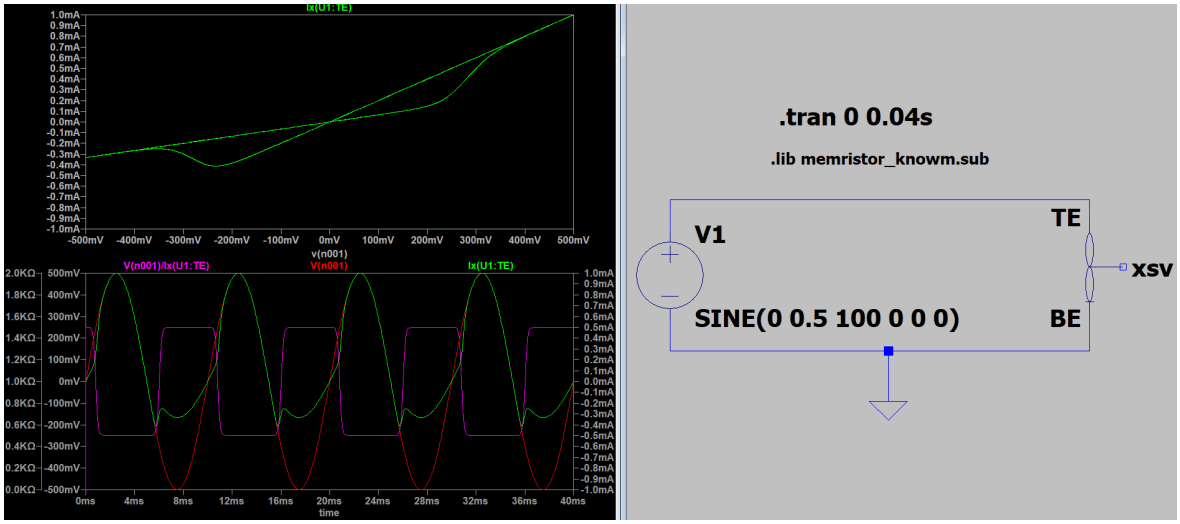


Figure B.5: Base Knowm mean memristor model, no resistor, 100Hz 0.5V sine wave, no offset.

Figure B.6: Base Knowm mean memristor model, no resistor, 1500Hz 0.5V sine wave, no offset.

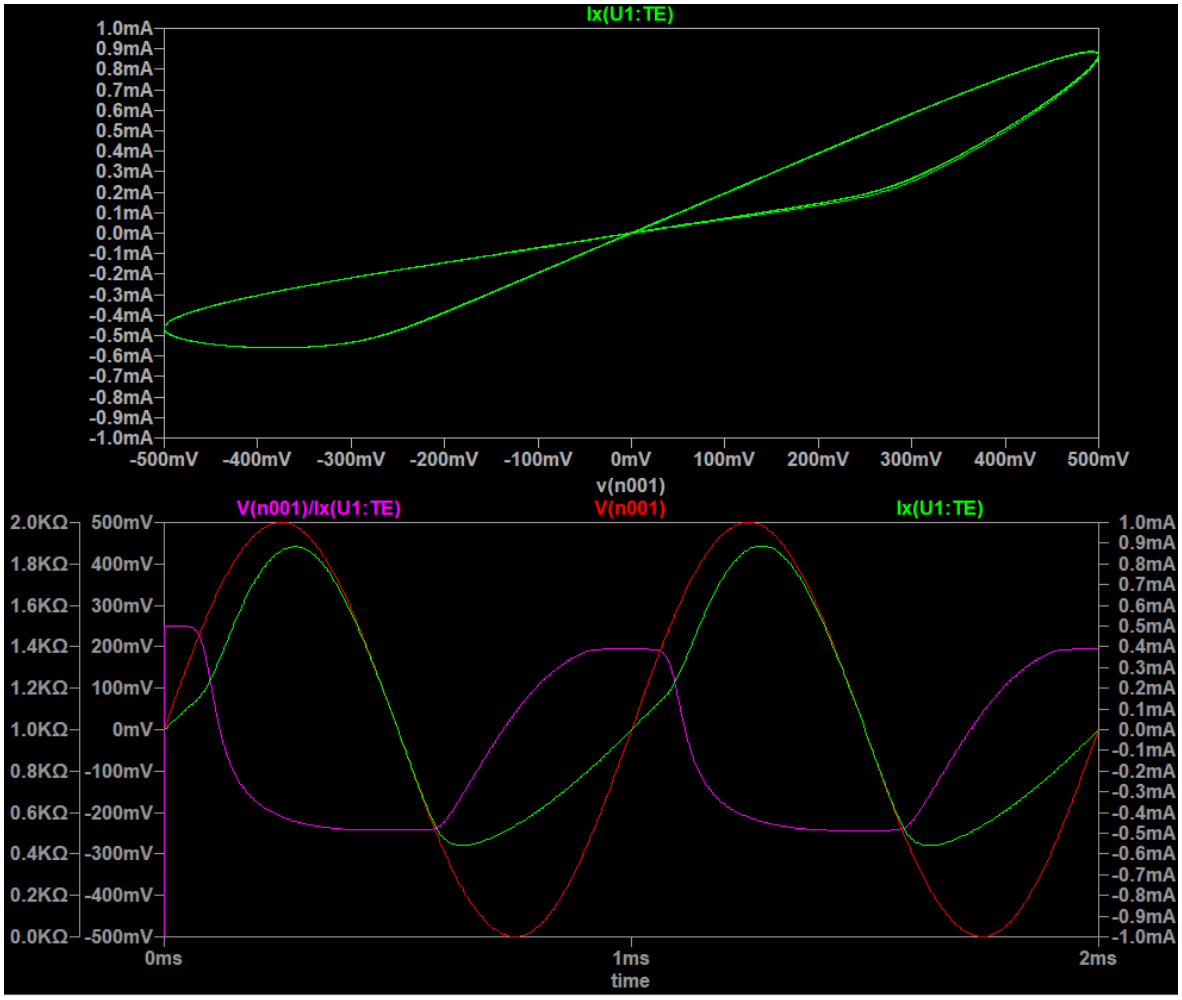


Figure B.7: Base Knowm mean memristor model, no resistor, 1000Hz 0.5V sine wave, no offset.

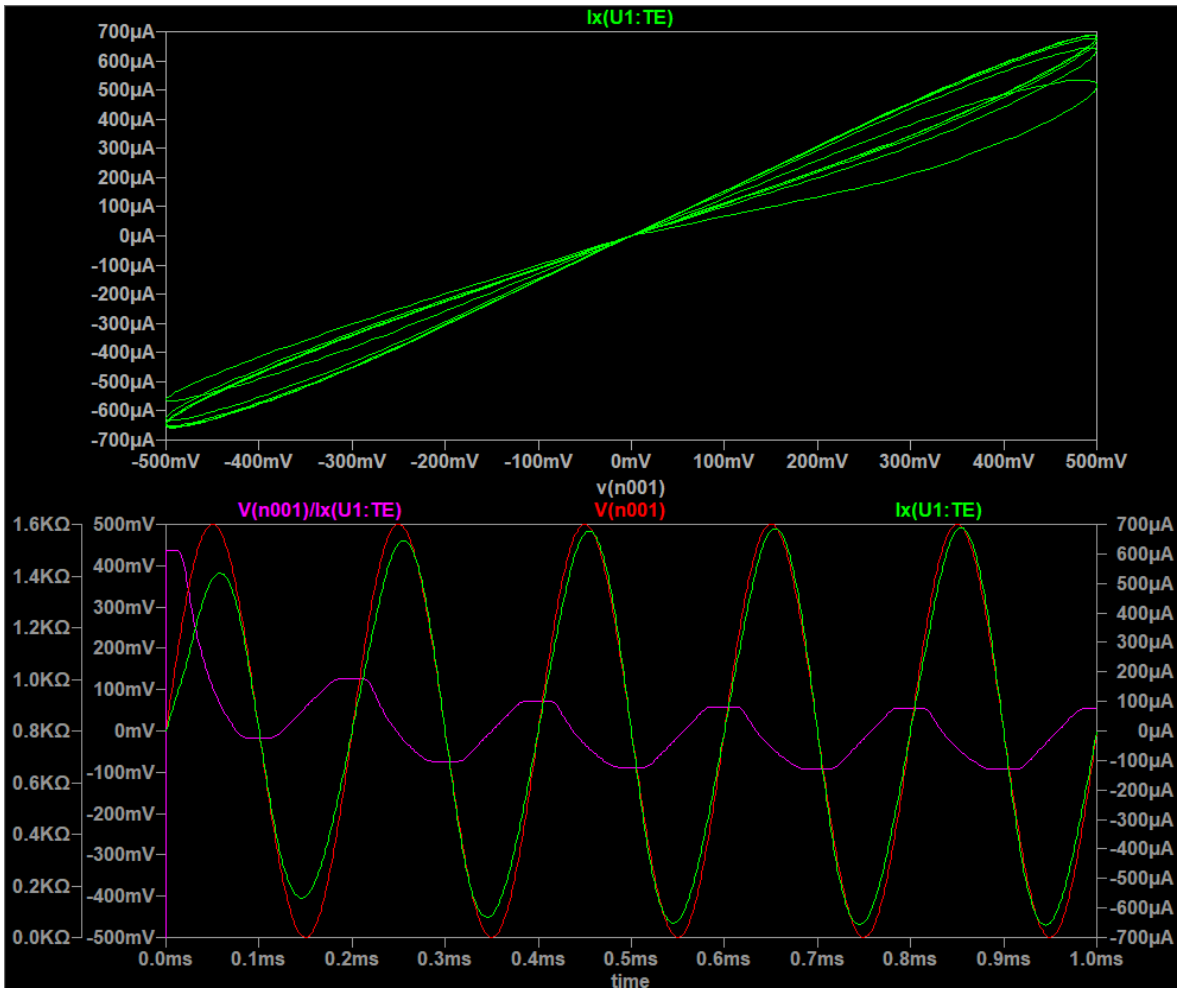


Figure B.8: Base Knowm mean memristor model, no resistor, 5000Hz 0.5V sine wave, no offset.

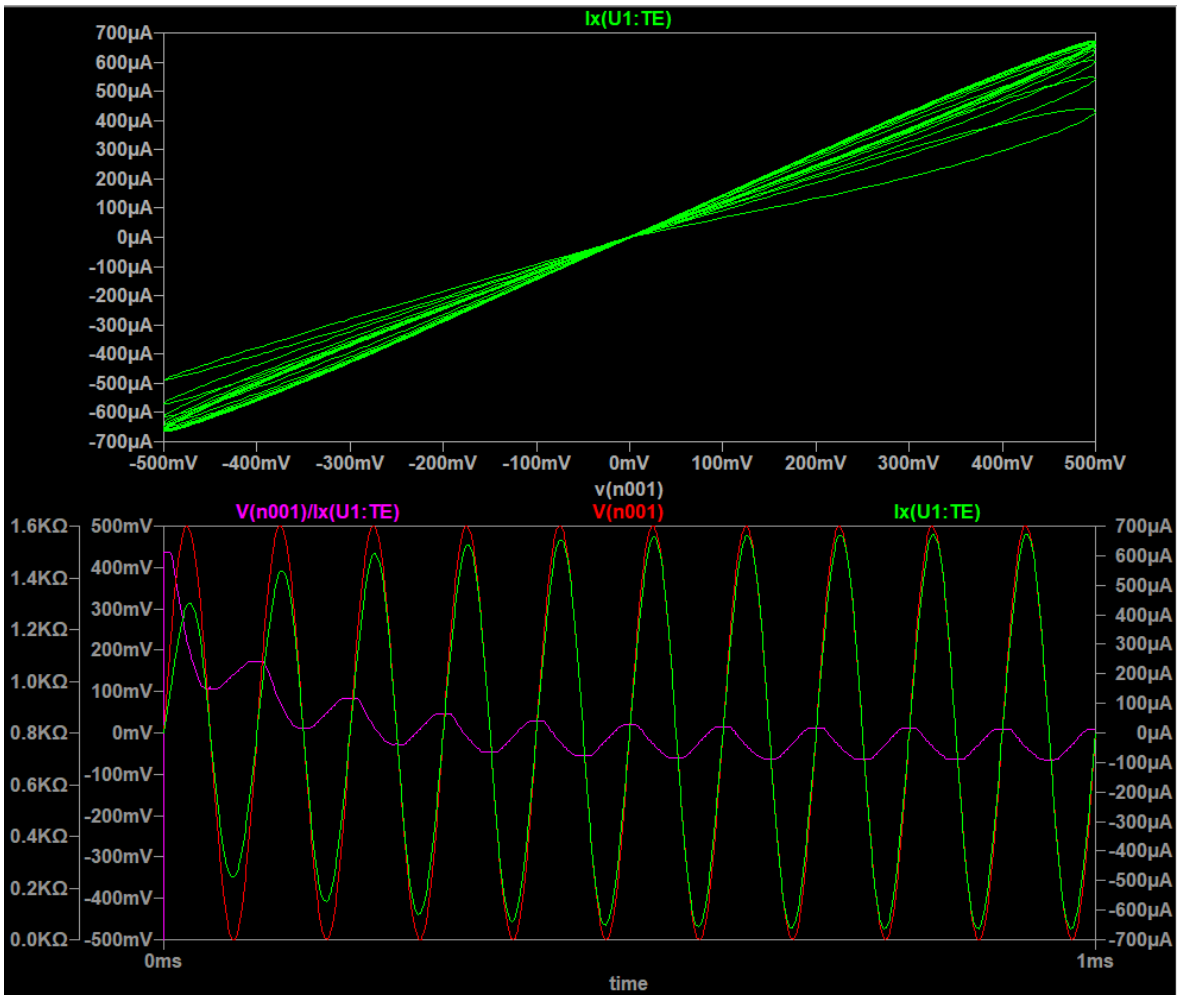


Figure B.9: Base Knowm mean memristor model, no resistor, 10000Hz 0.5V sine wave, no offset.

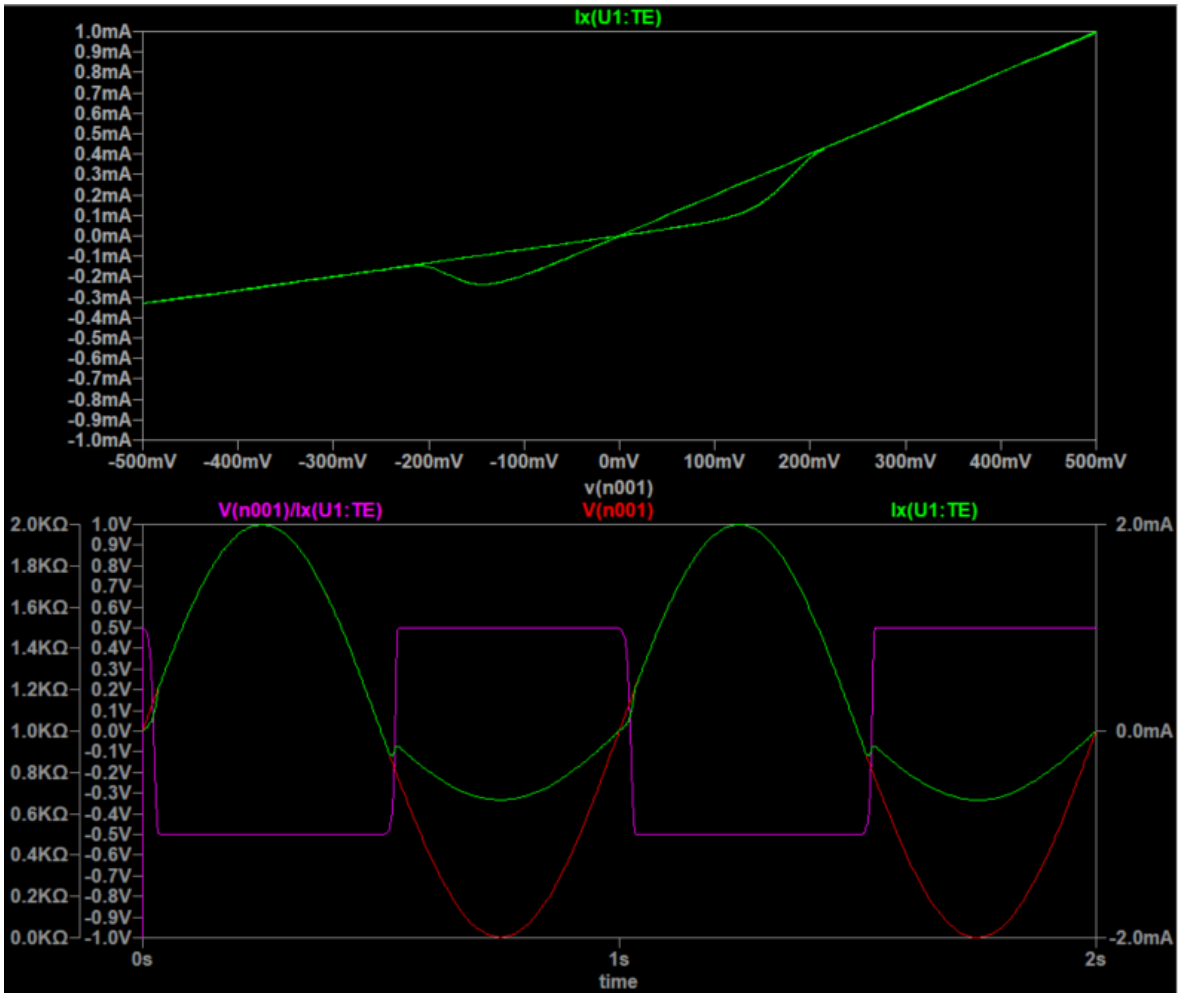


Figure B.10: Base Knowm mean memristor model, no resistor, 1Hz 1V sine wave, no offset.

Base Knowm mean memristor model, no resistor, 1V sine wave, no offset

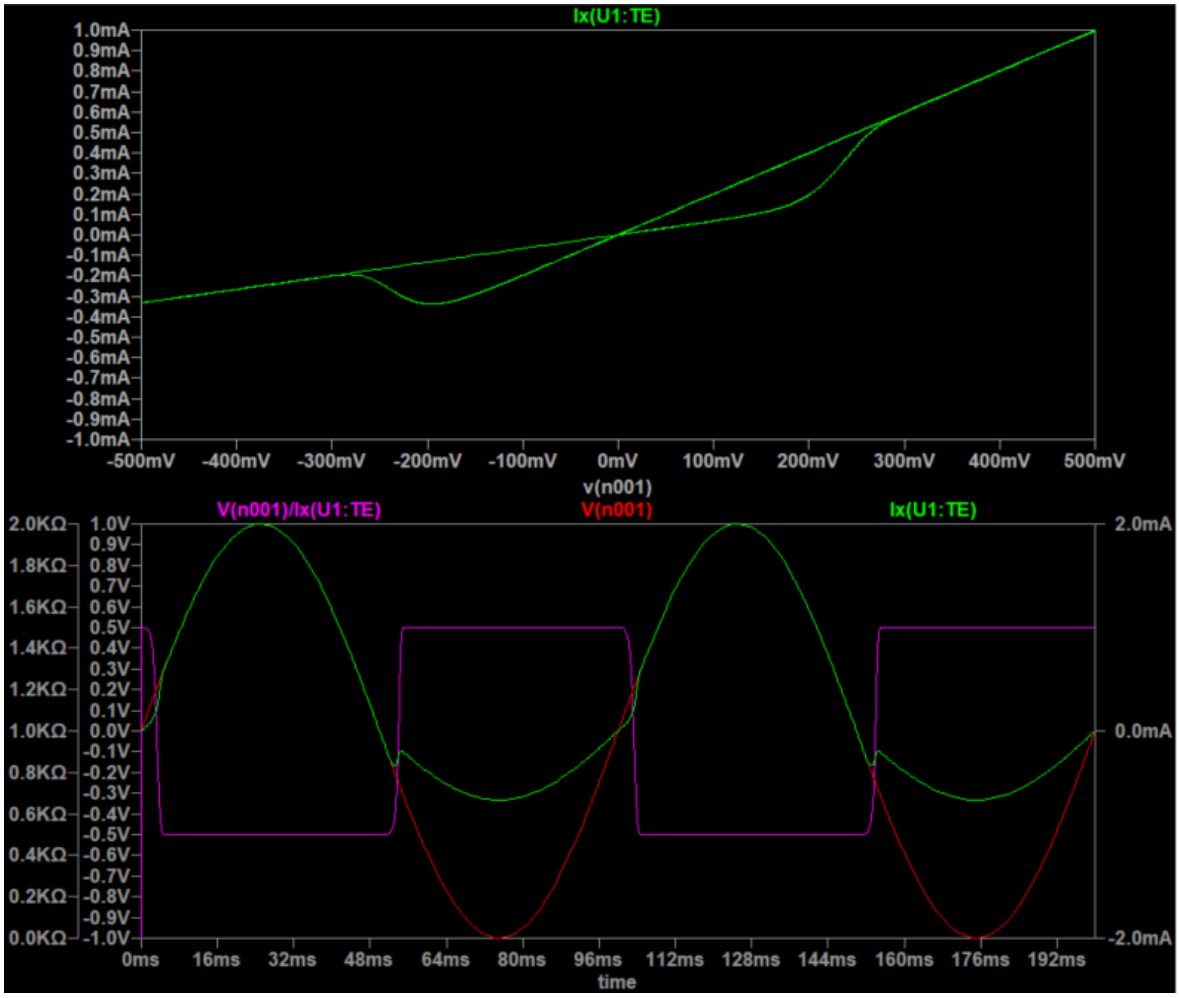


Figure B.11: Base Knowm mean memristor model, no resistor, 10Hz 1V sine wave, no offset.

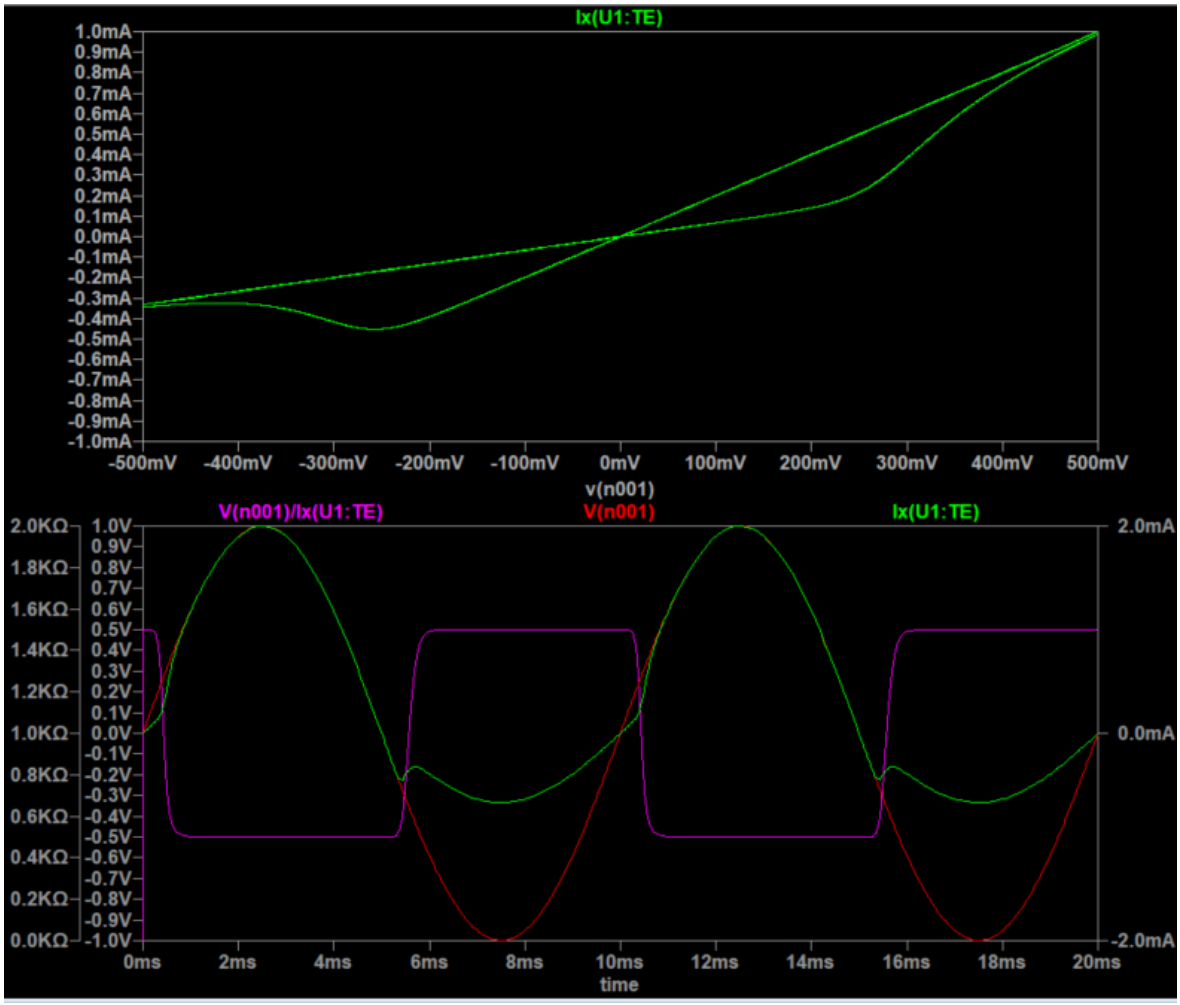


Figure B.12: Base Knowm mean memristor model, no resistor, 100Hz 1V sine wave, no offset.

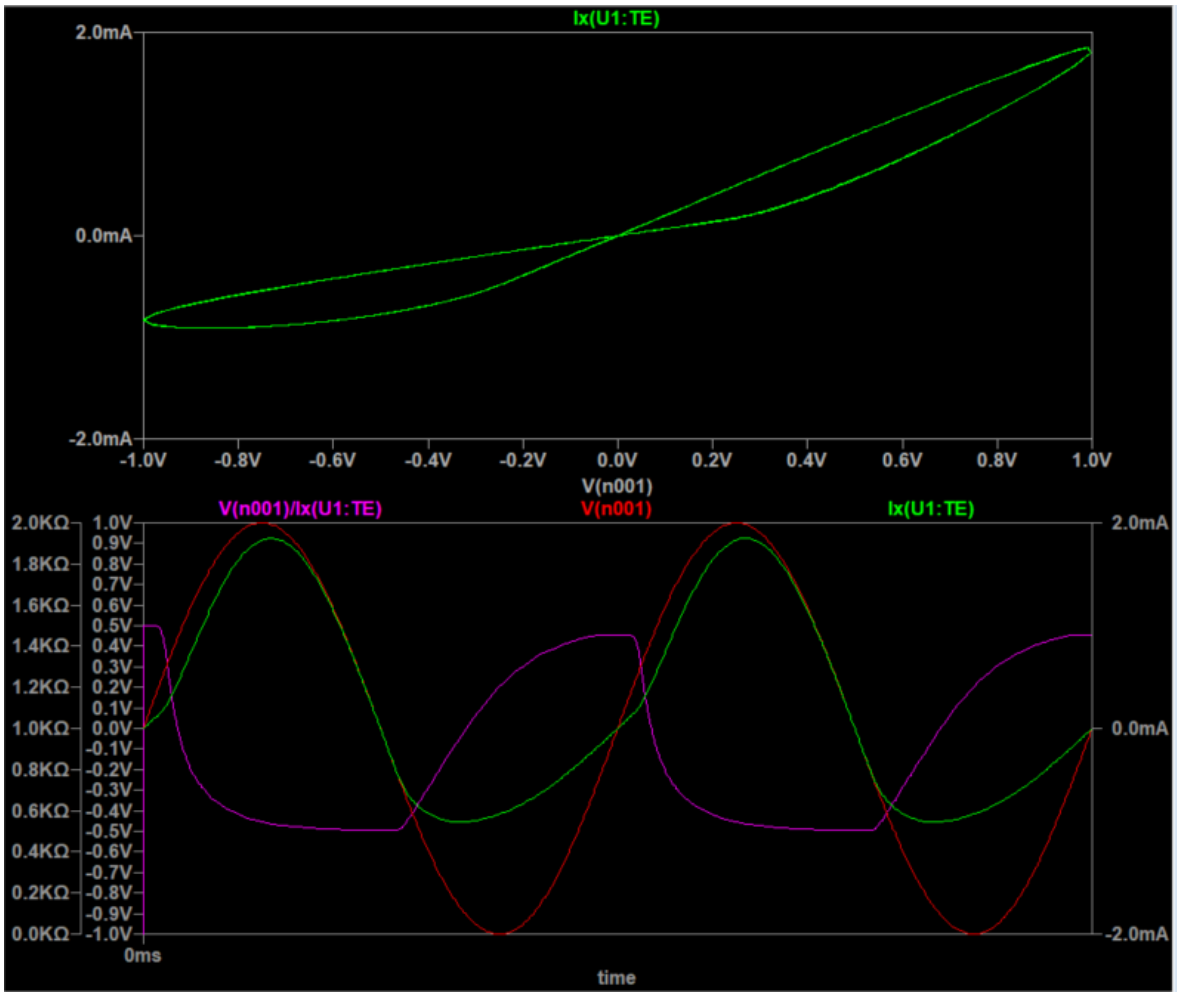


Figure B.13: Base Knowm mean memristor model, no resistor, 1000Hz 1V sine wave, no offset.

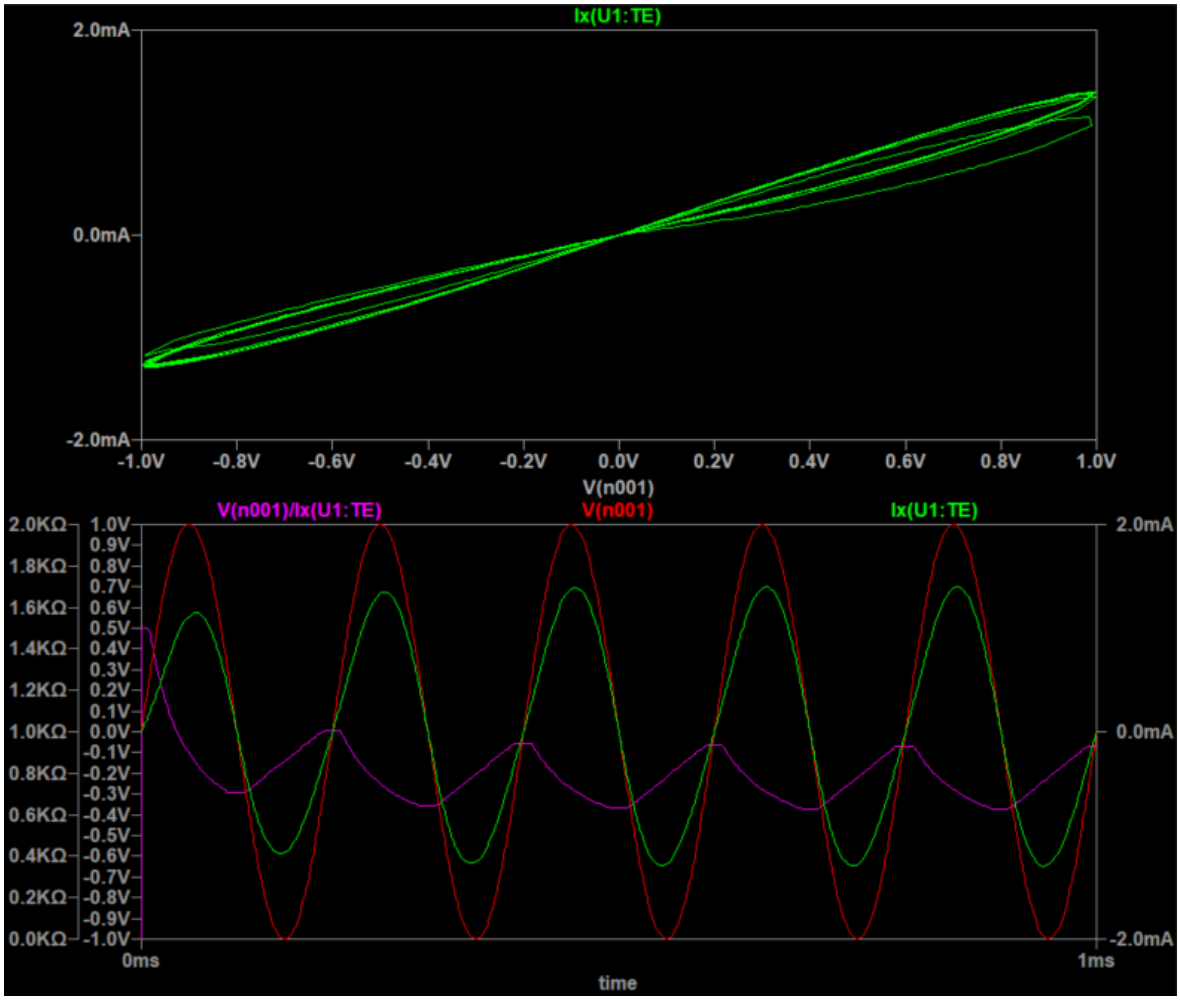


Figure B.14: Base Knowm mean memristor model, no resistor, 5000Hz 1V sine wave, no offset.

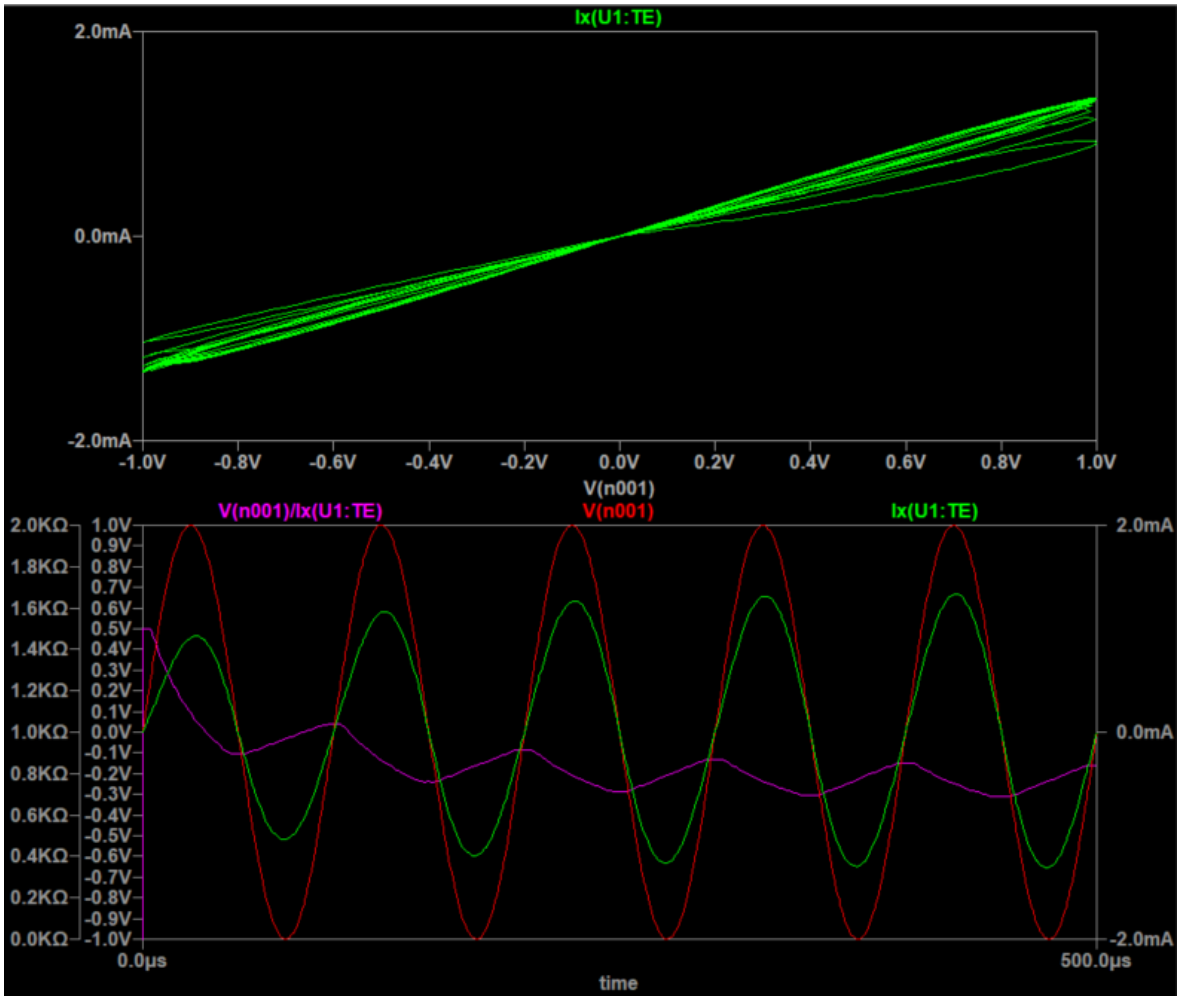


Figure B.15: Base Knowm mean memristor model, no resistor, 10000Hz 1V sine wave, no offset.

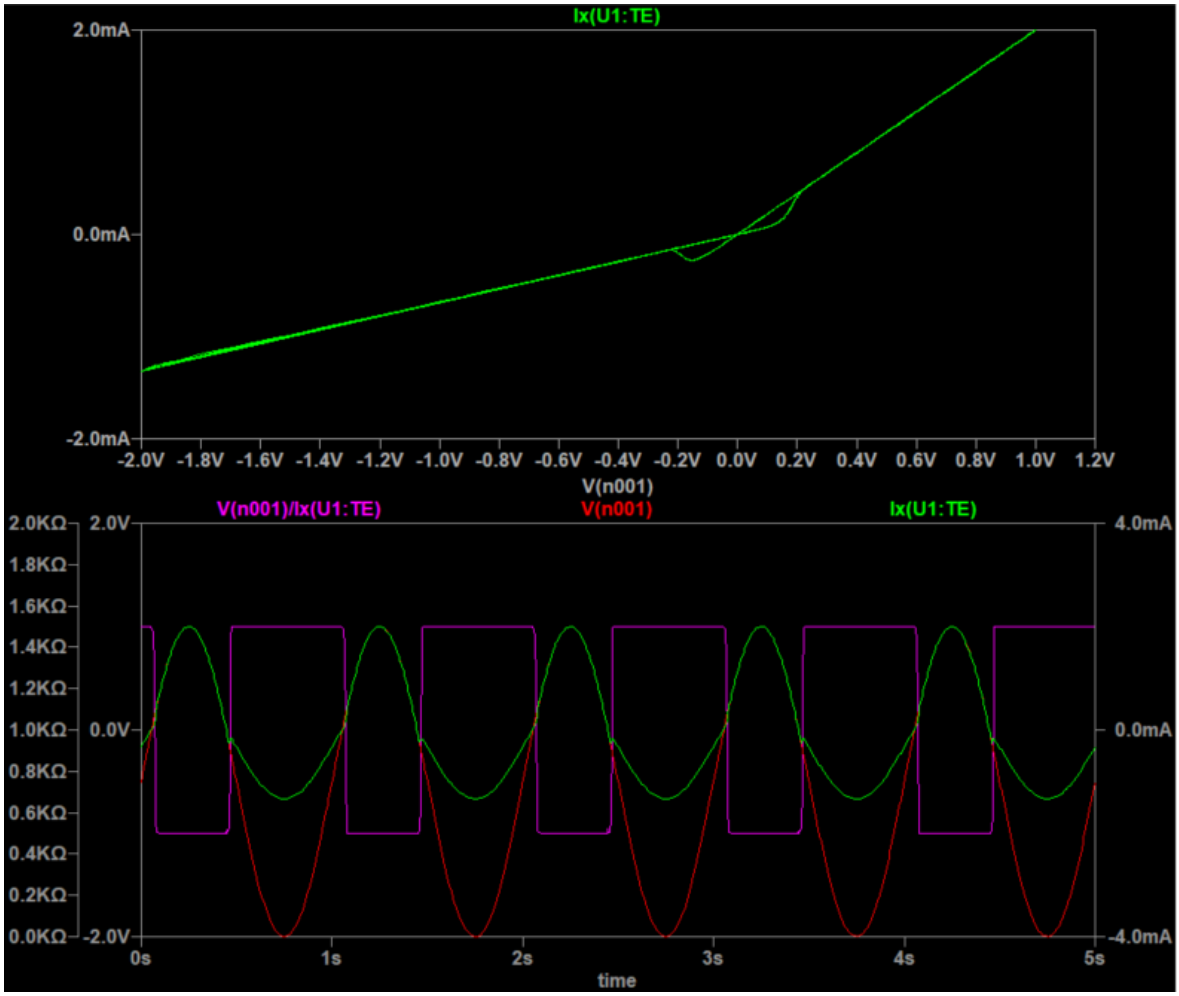


Figure B.16: Base Knowm mean memristor model, no resistor, 1Hz 1.5V sine wave, -0.5V offset.

Base Knowm mean memristor model, no resistor, 1.5V sine wave, -0.5V offset

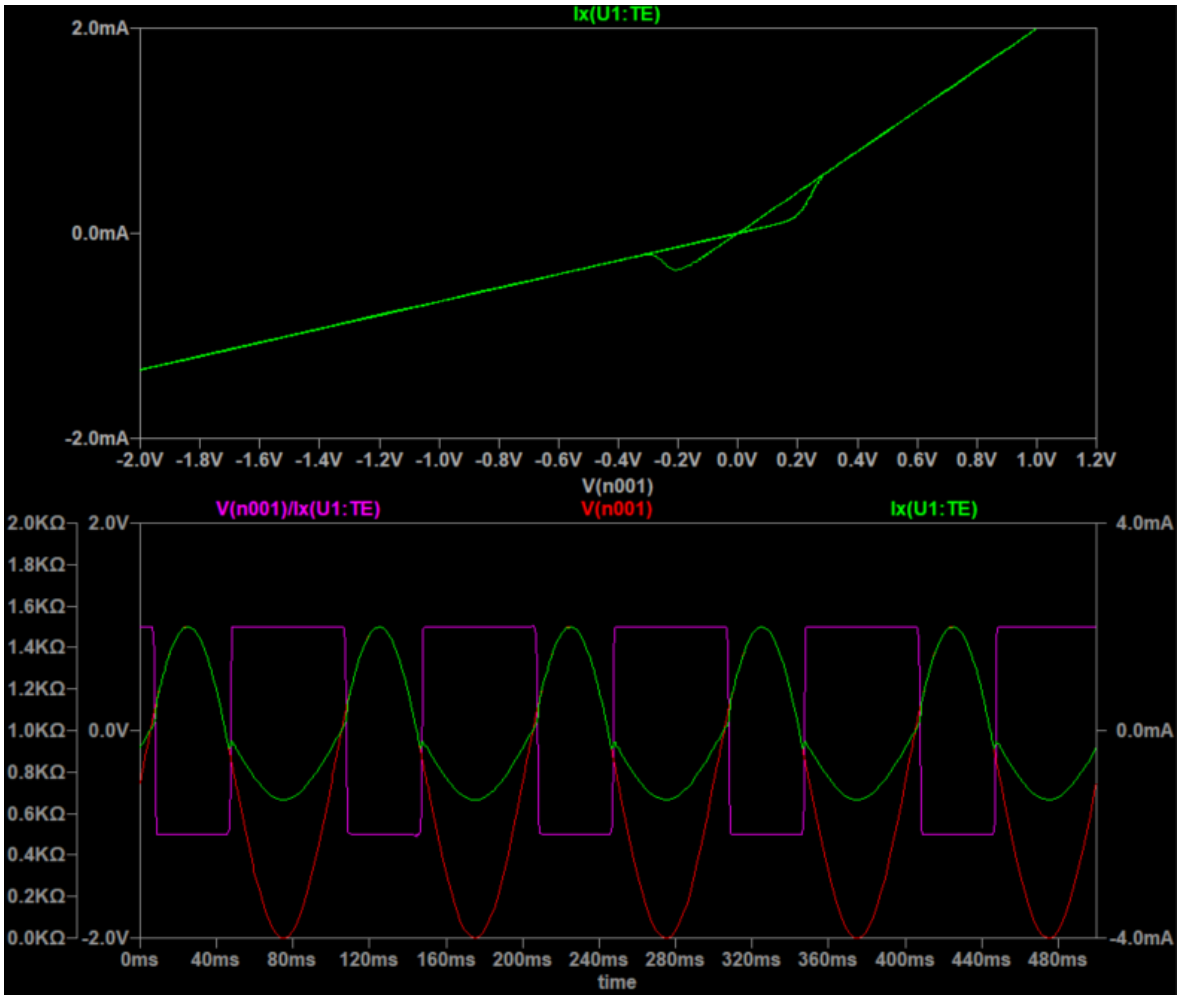


Figure B.17: Base Knowm mean memristor model, no resistor, 10Hz 1.5V sine wave, -0.5V offset.

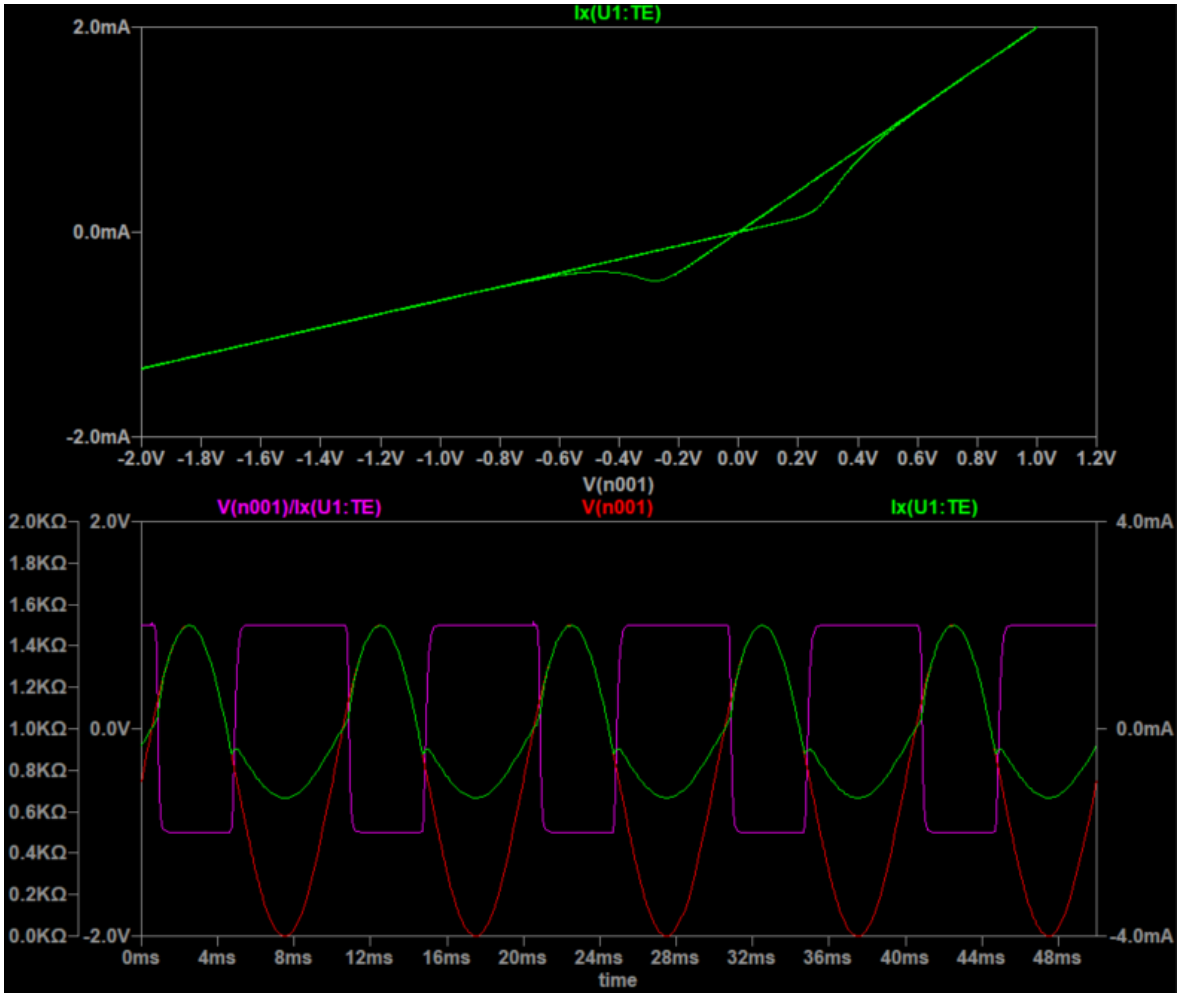


Figure B.18: Base Knowm mean memristor model, no resistor, 100Hz 1.5V sine wave, -0.5V offset.

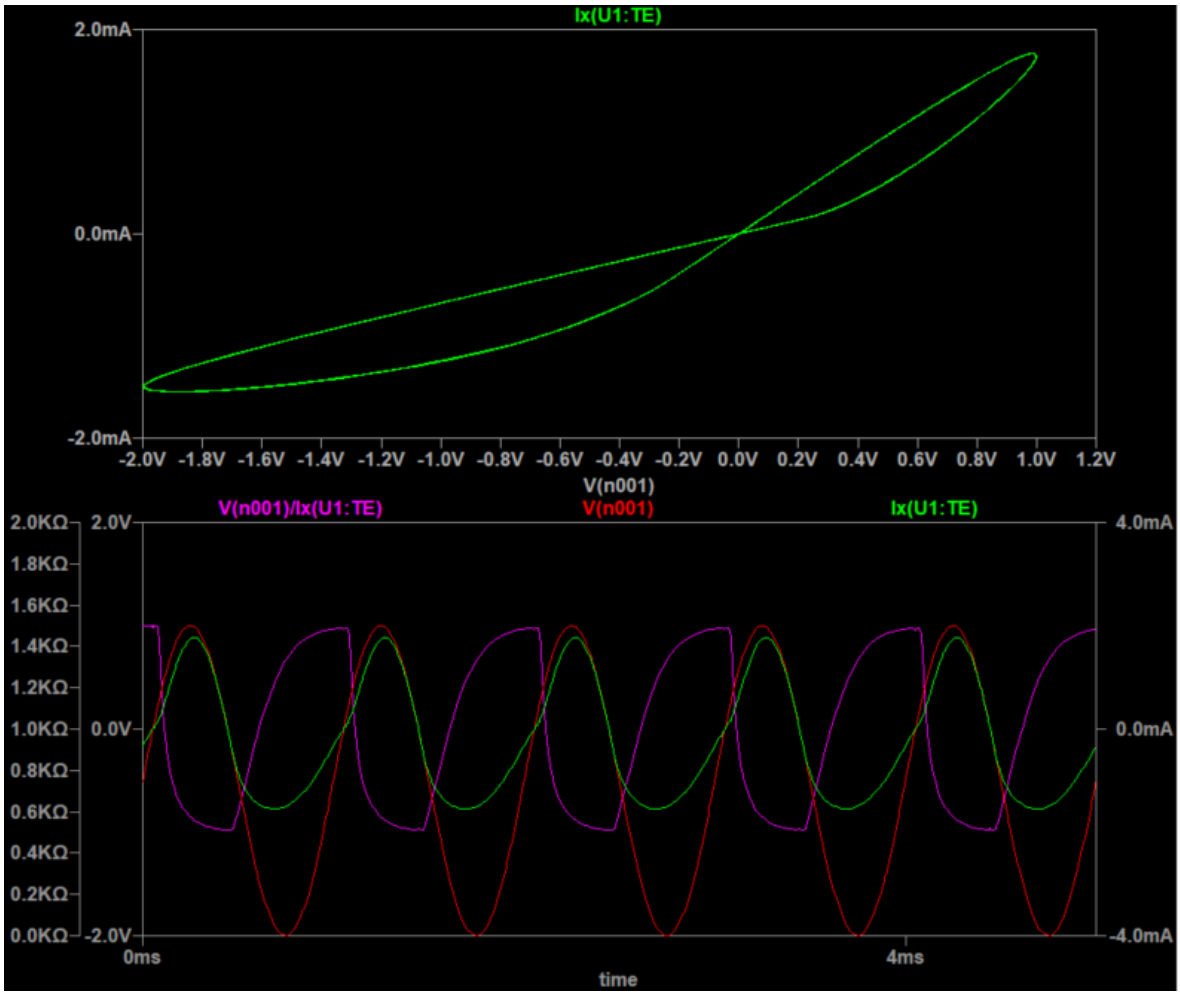


Figure B.19: Base Knowm mean memristor model, no resistor, 1000Hz 1.5V sine wave, -0.5V offset.

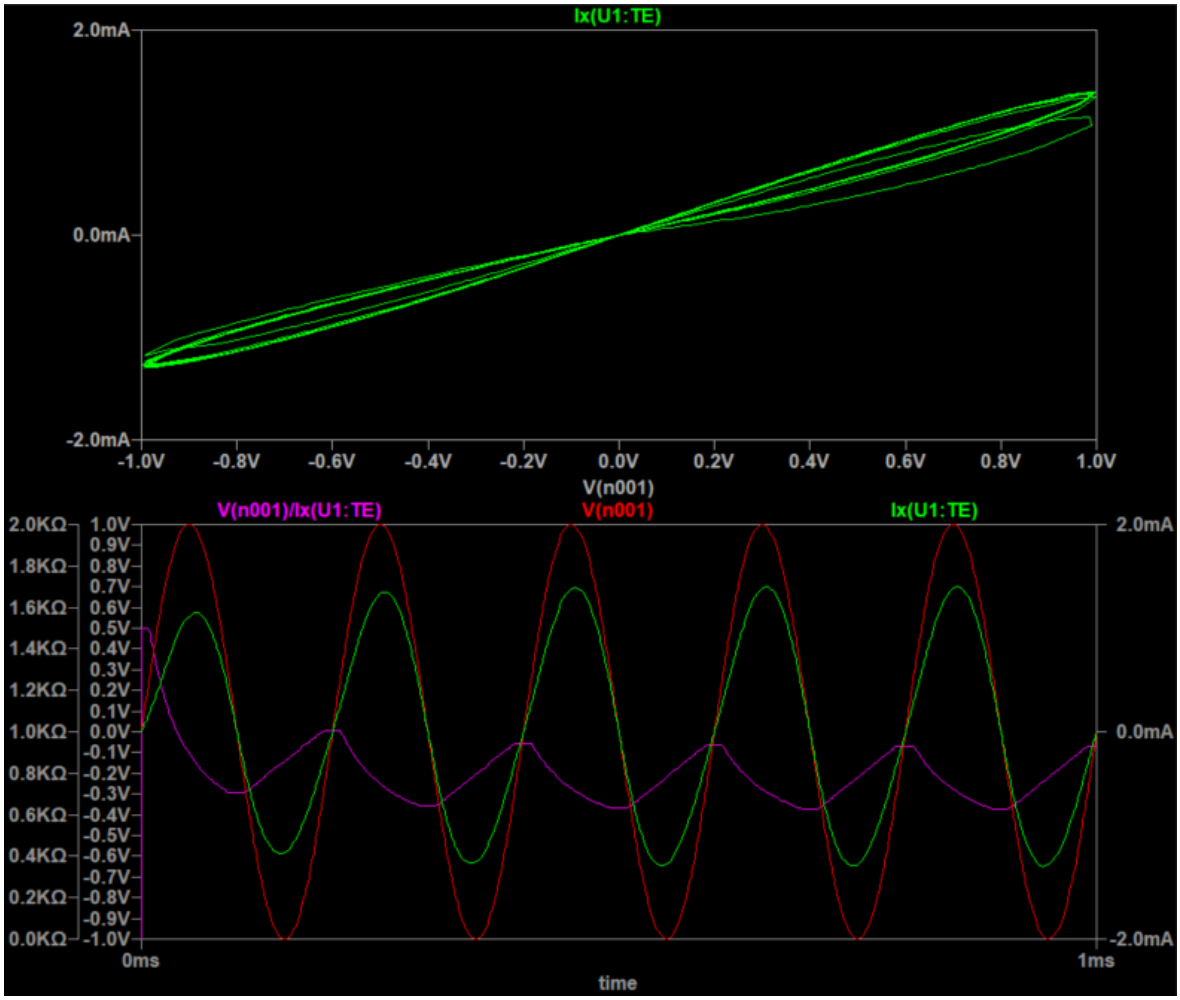


Figure B.20: Base Knowm mean memristor model, no resistor, 5000Hz 1.5V sine wave, -0.5V offset.

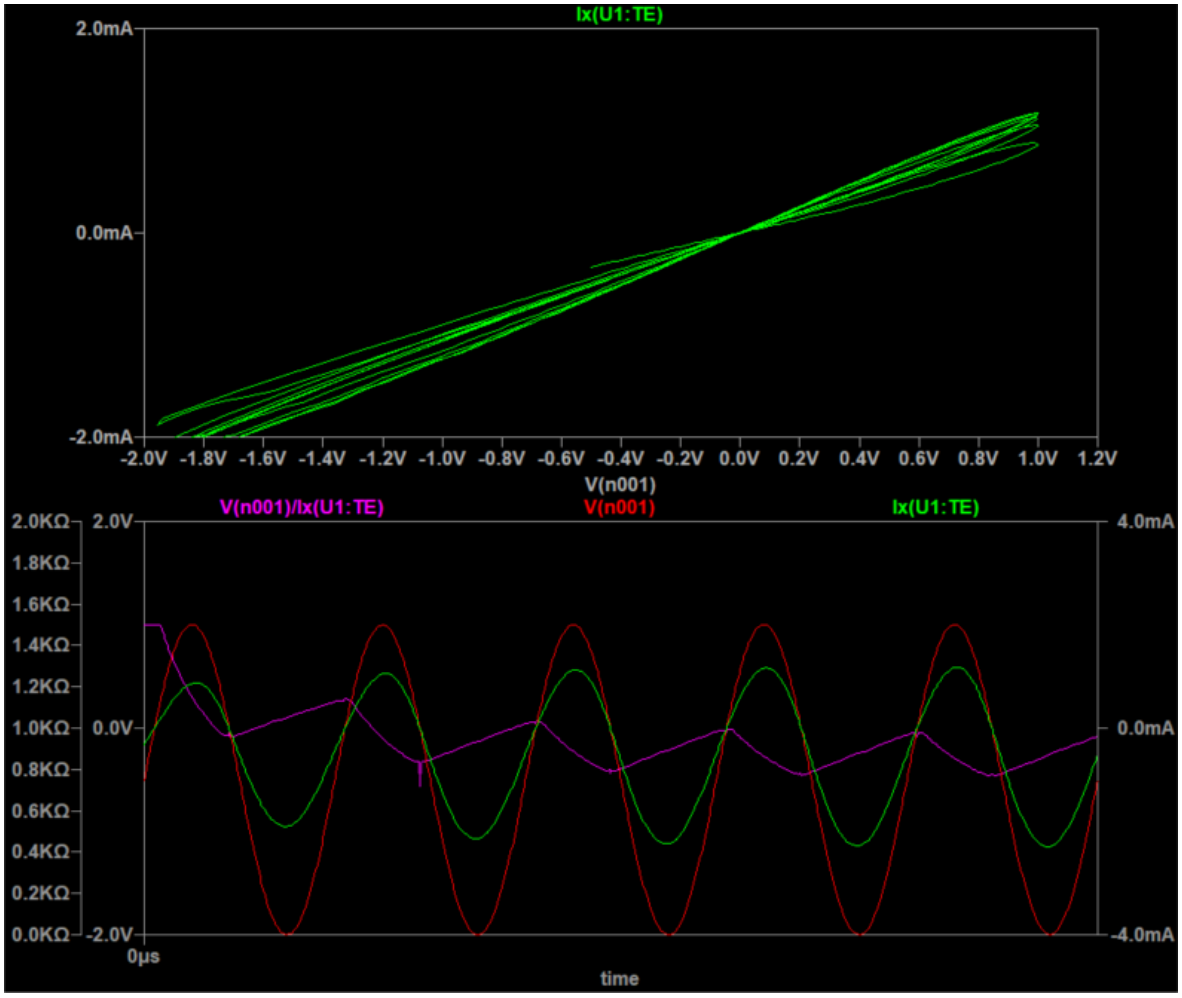


Figure B.21: Base Knowm mean memristor model, no resistor, 10000Hz 1.5V sine wave, 0.5V offset.

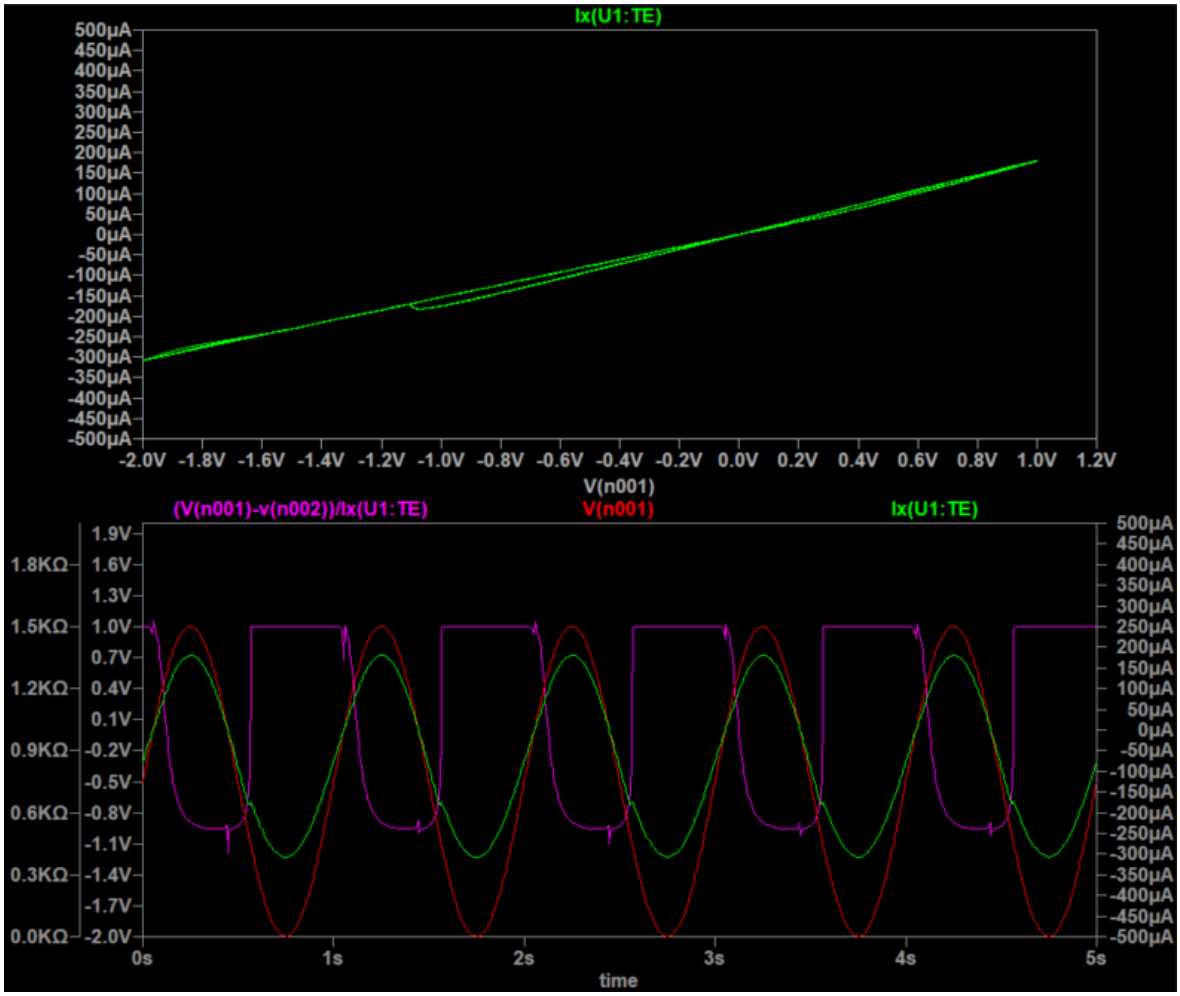


Figure B.22: Base Knowm mean memristor model, 5KΩ resistor, 1Hz 1.5V sine wave, -0.5V offset.

Base Knowm mean memristor model, 5KΩ resistor, 1.5V sine wave, -0.5V offset

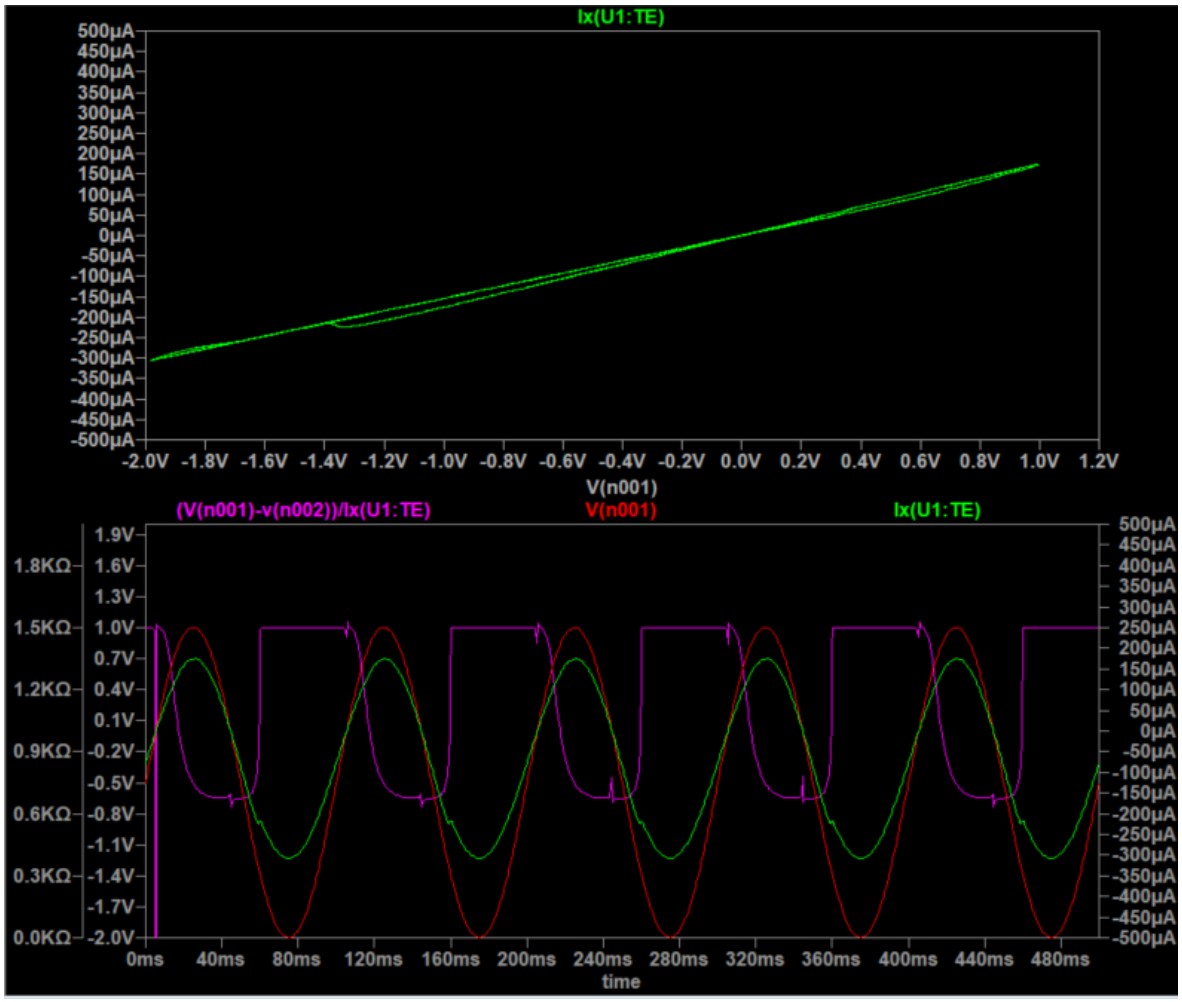


Figure B.23: Base Knowm mean memristor model, 5KΩ resistor, 10Hz 1.5V sine wave, -0.5V offset.

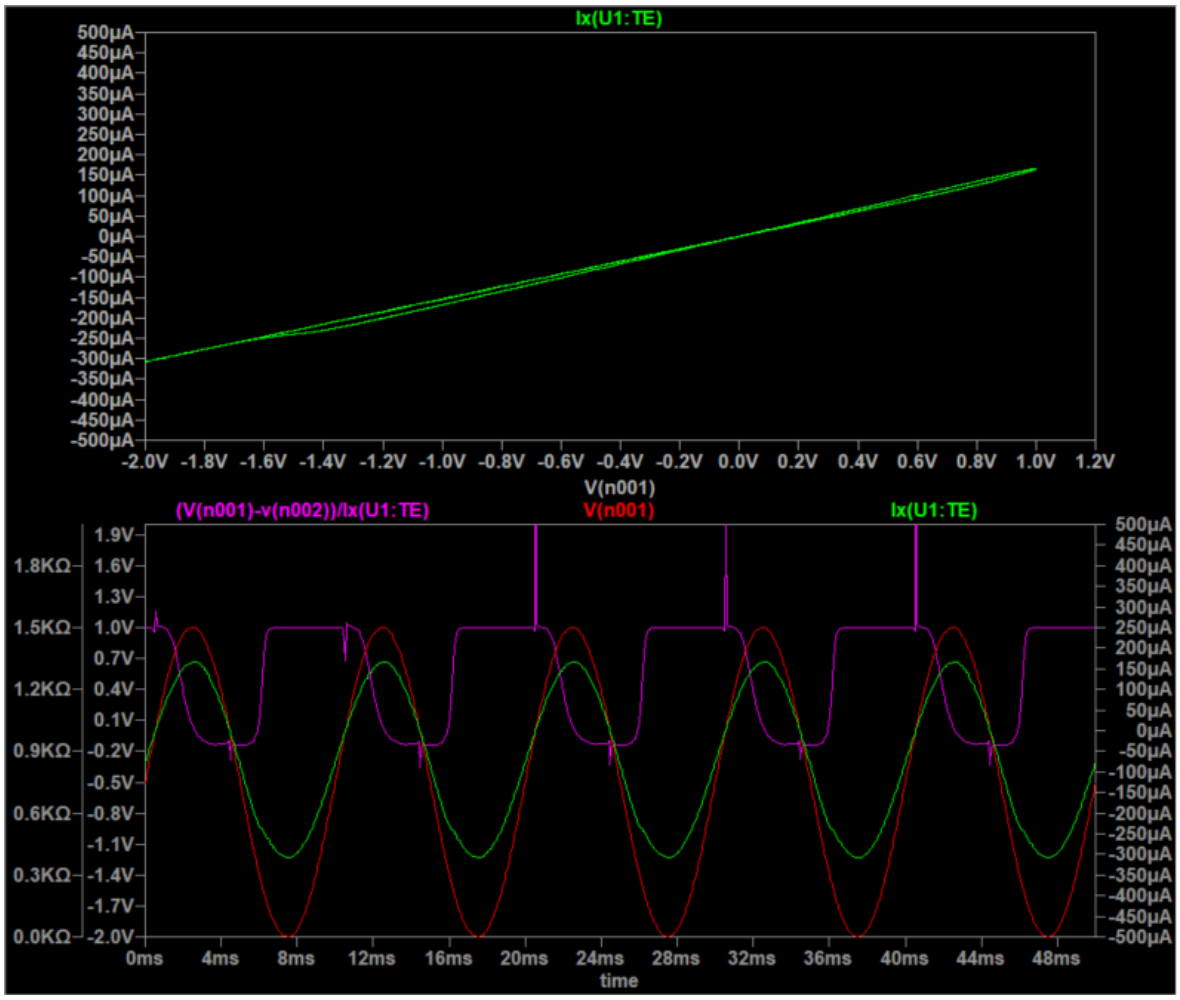


Figure B.24: Base Knowm mean memristor model, 5KΩ resistor, 100Hz 1.5V sine wave, -0.5V offset.

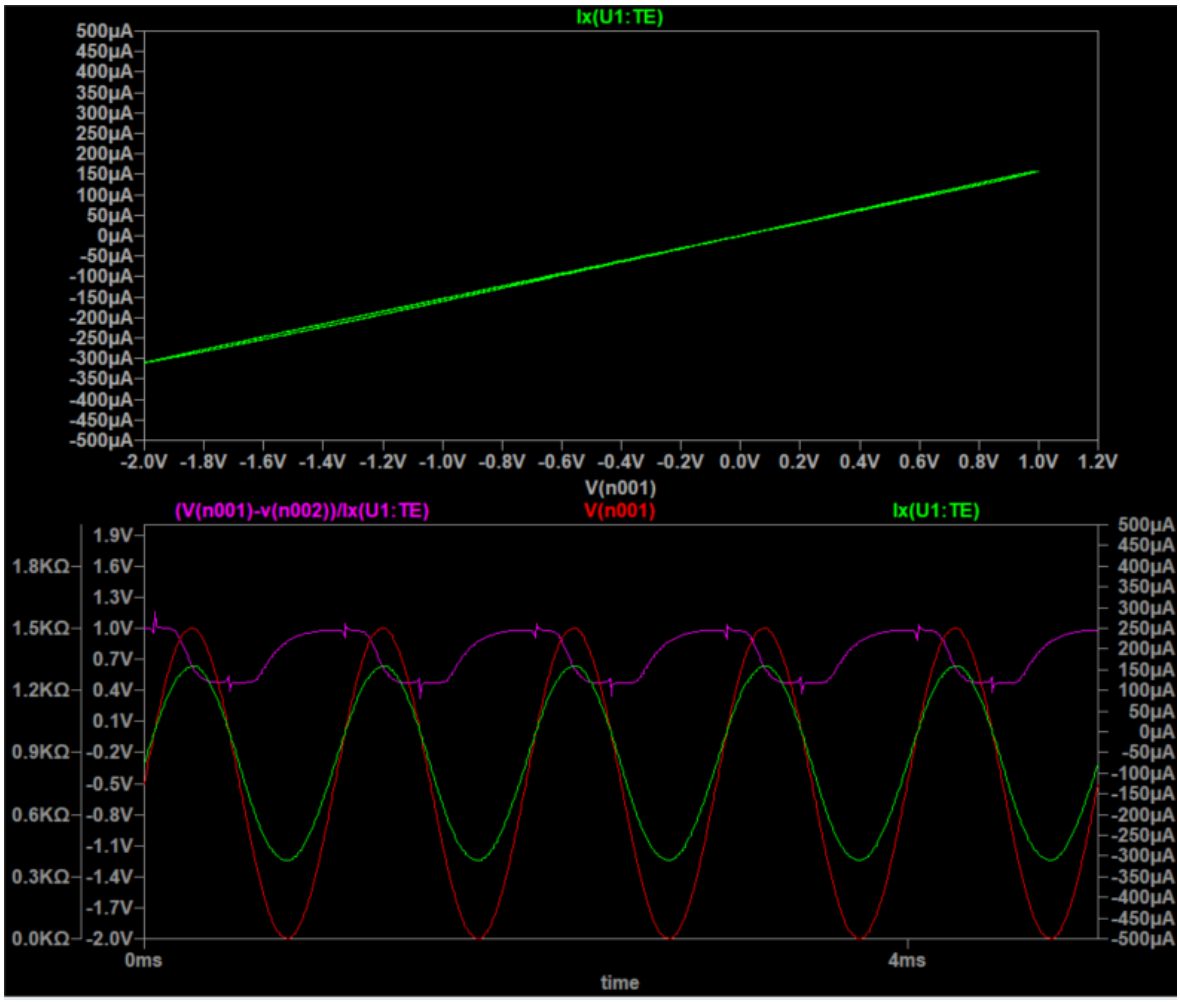


Figure B.25: Base Knowm mean memristor model, 5KΩ resistor, 1000Hz 1.5V sine wave, -0.5V offset.

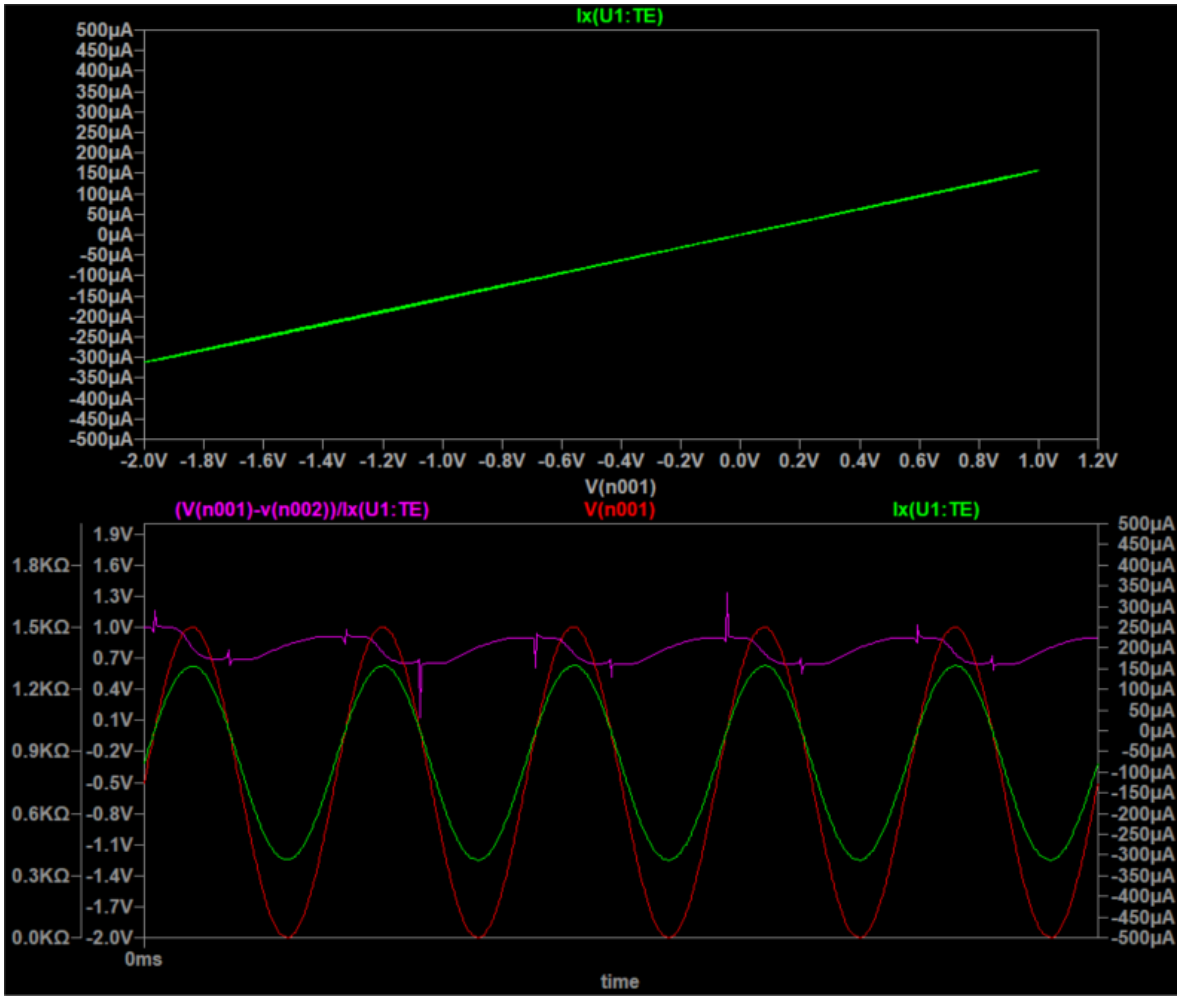


Figure B.26: Base Knowm mean memristor model, 5KΩ resistor, 2500Hz 1.5V sine wave, -0.5V offset.

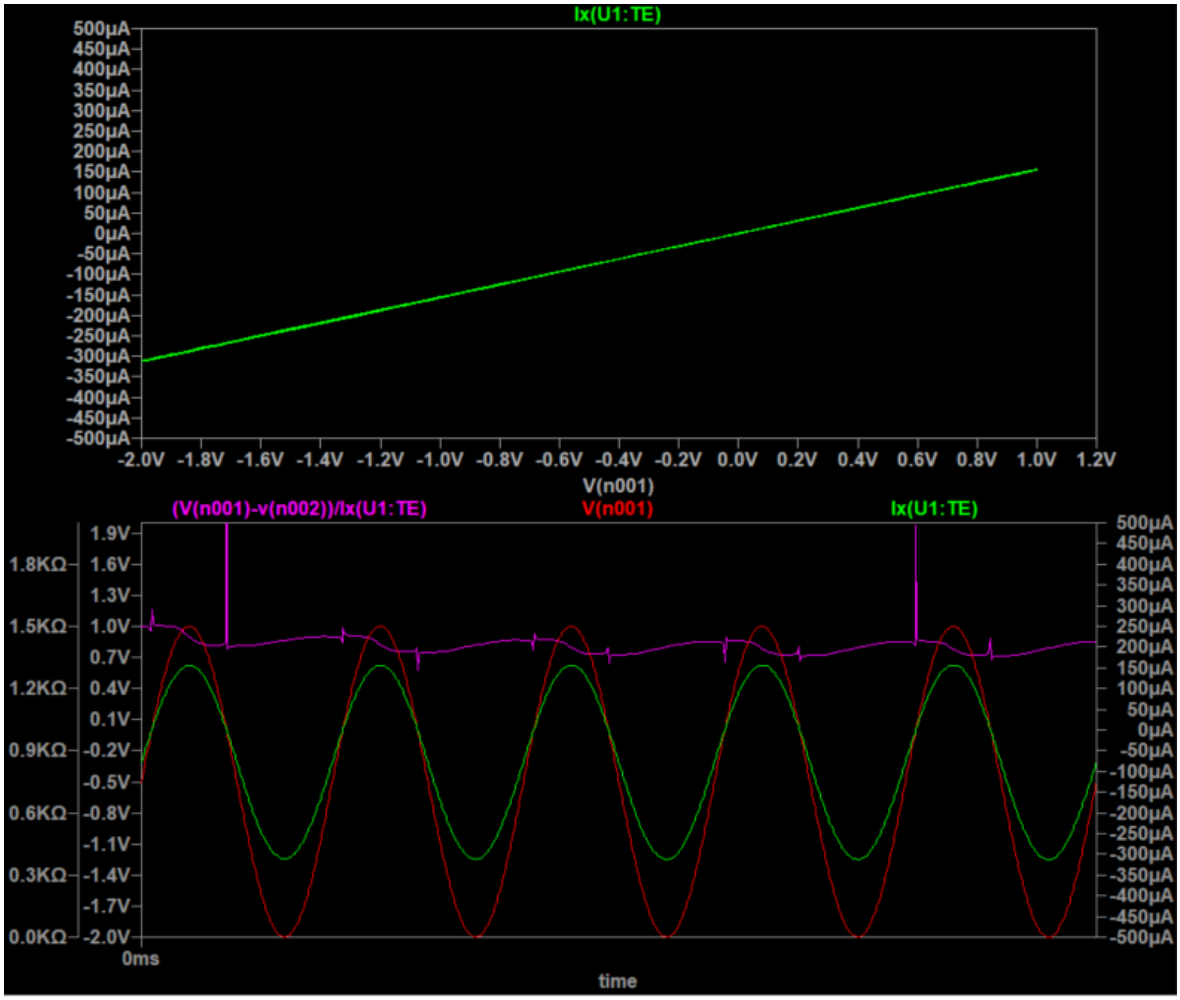


Figure B.27: Base Knowm mean memristor model, 5KΩ resistor, 5000Hz 1.5V sine wave, -0.5V offset.

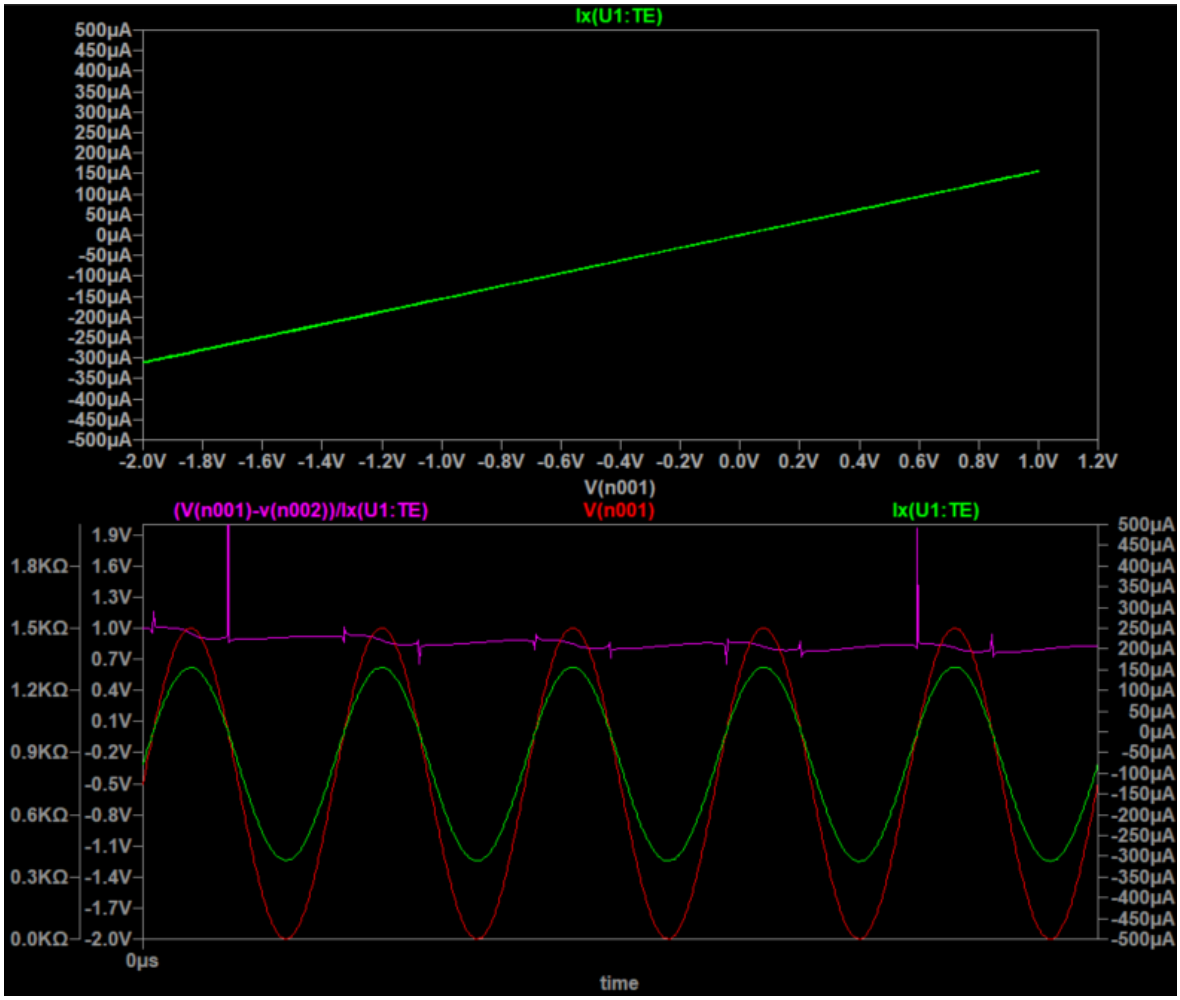


Figure B.28: Base Knowm mean memristor model, 5KΩ resistor, 10000Hz 1.5V sine wave, 0.5V offset.

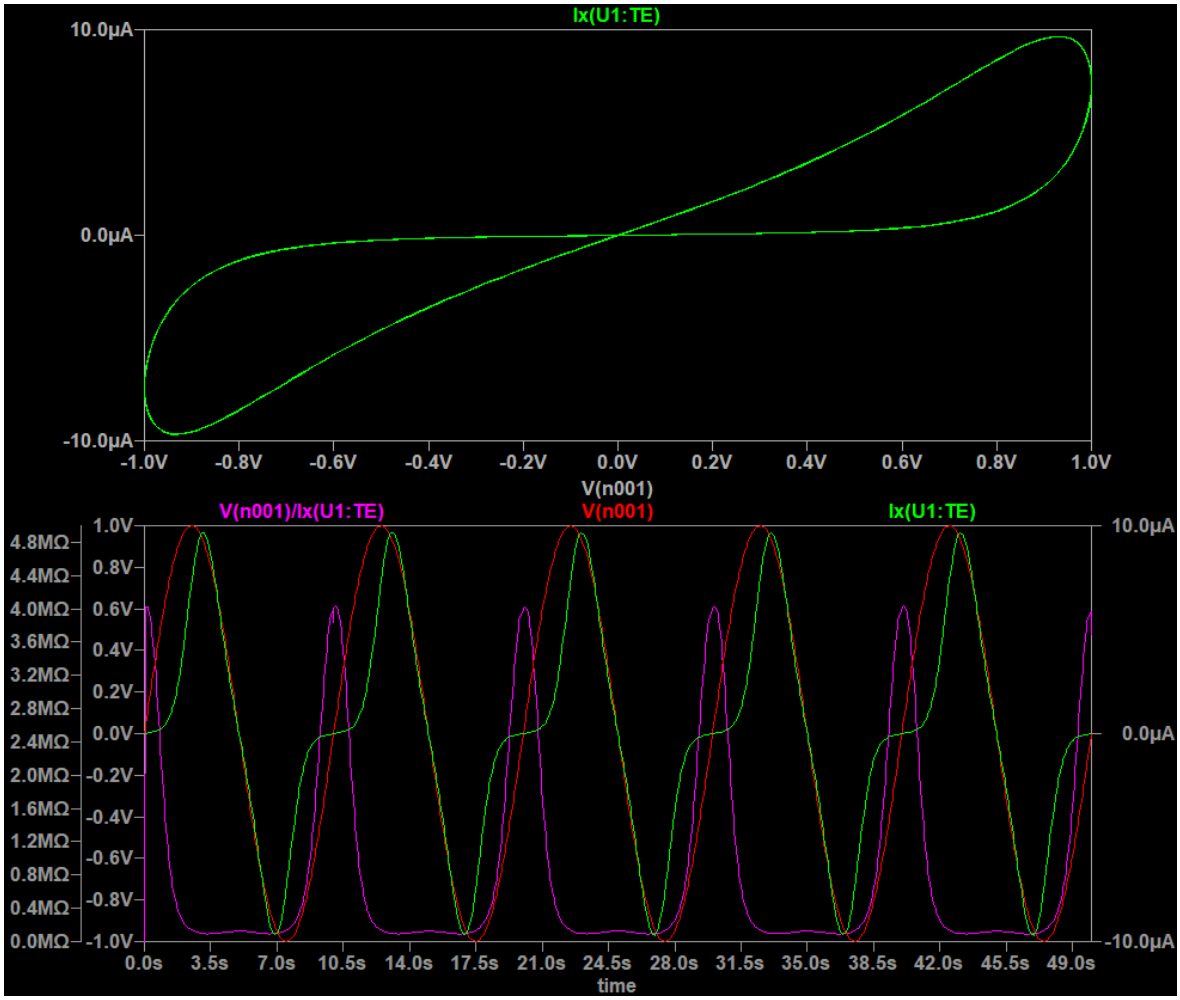


Figure B.29: Memristor model from the University of Michigan found at [44]. No resistor, 0.1Hz 1V sine wave, no offset.

1.2 University of Michigan Memristor Model

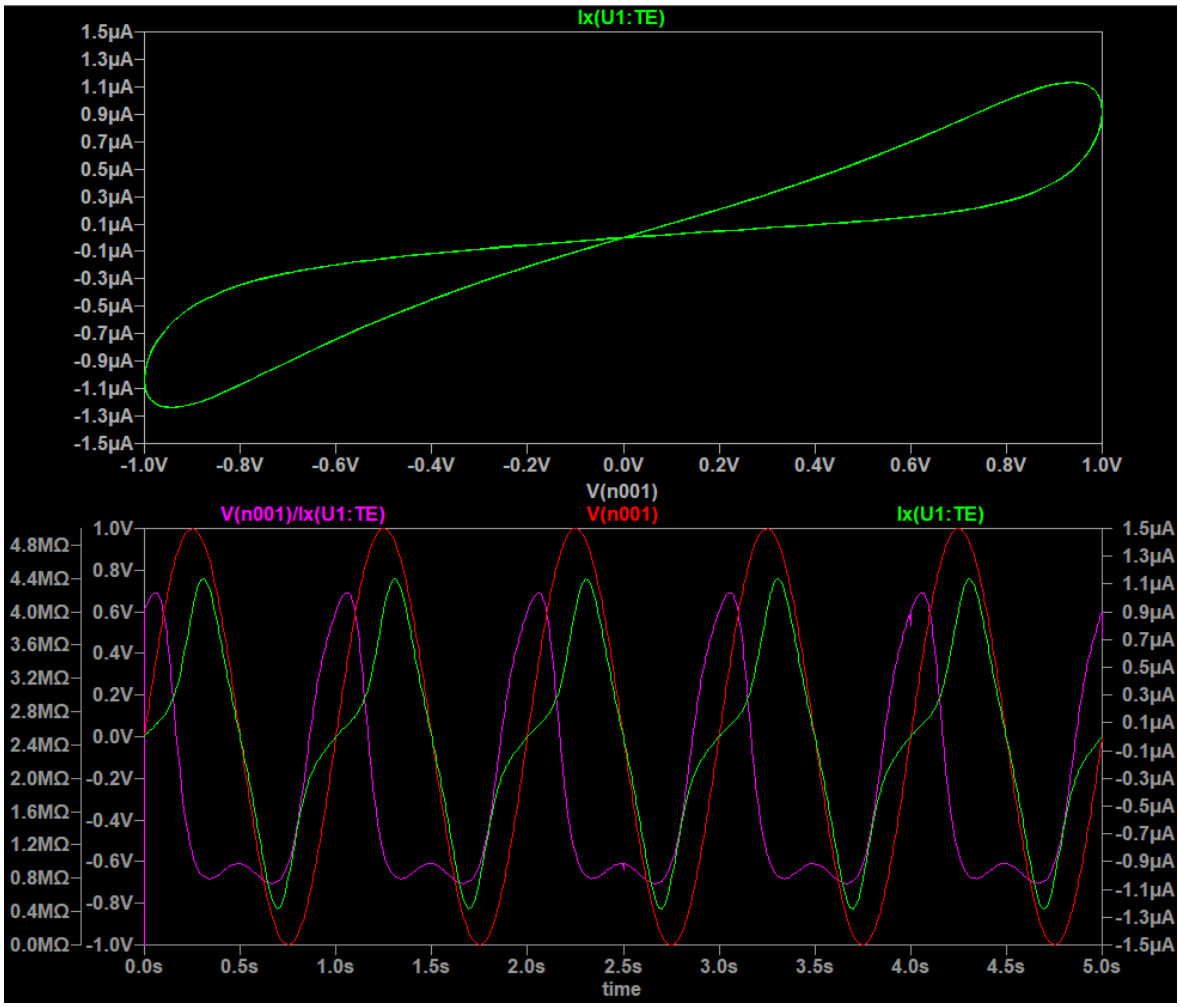


Figure B.30: Memristor model from the University of Michigan found at [44]. No resistor, 1Hz 1V sine wave, no offset.

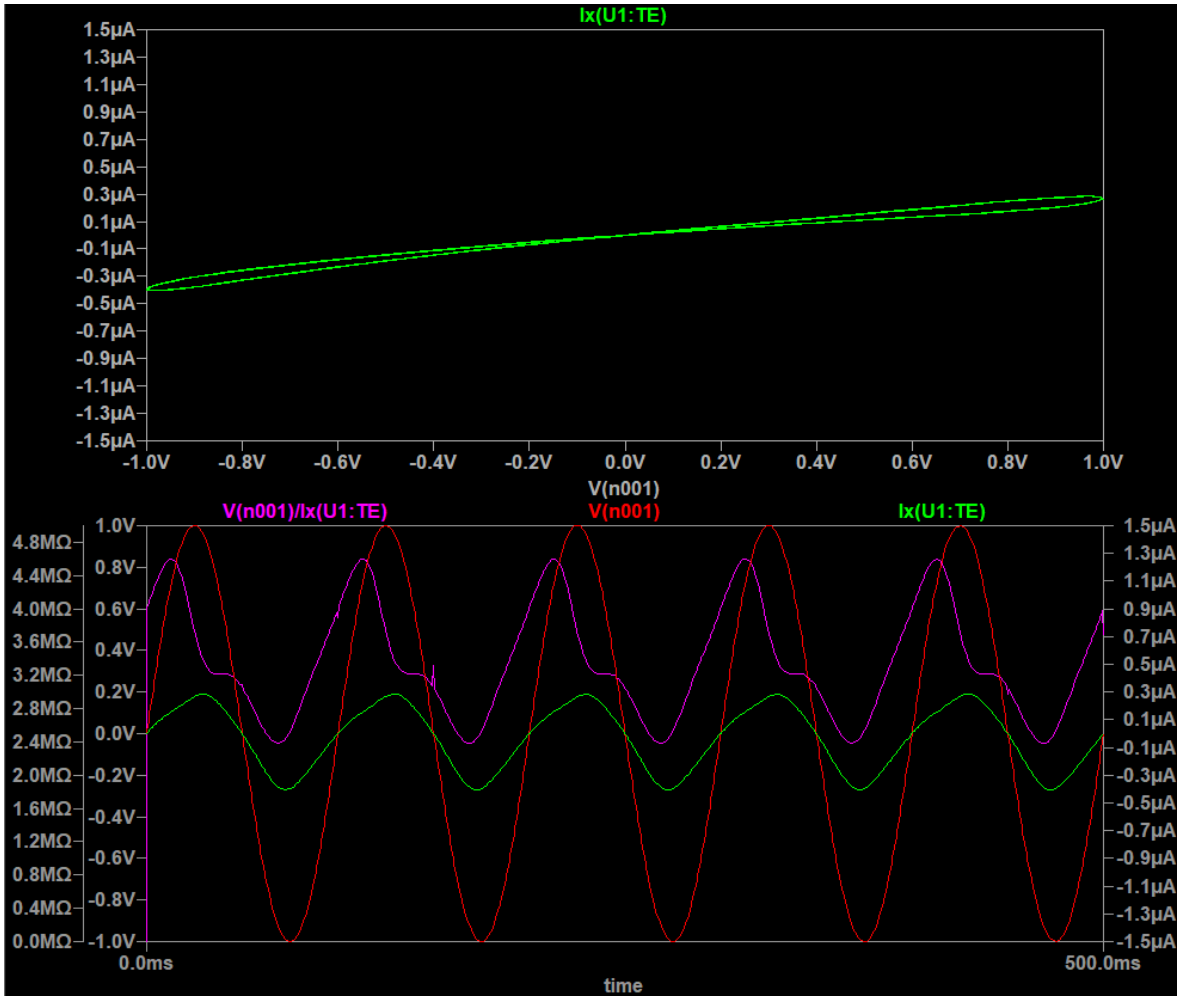


Figure B.31: Memristor model from the University of Michigan found at [44]. No resistor, 10Hz 1V sine wave, no offset.

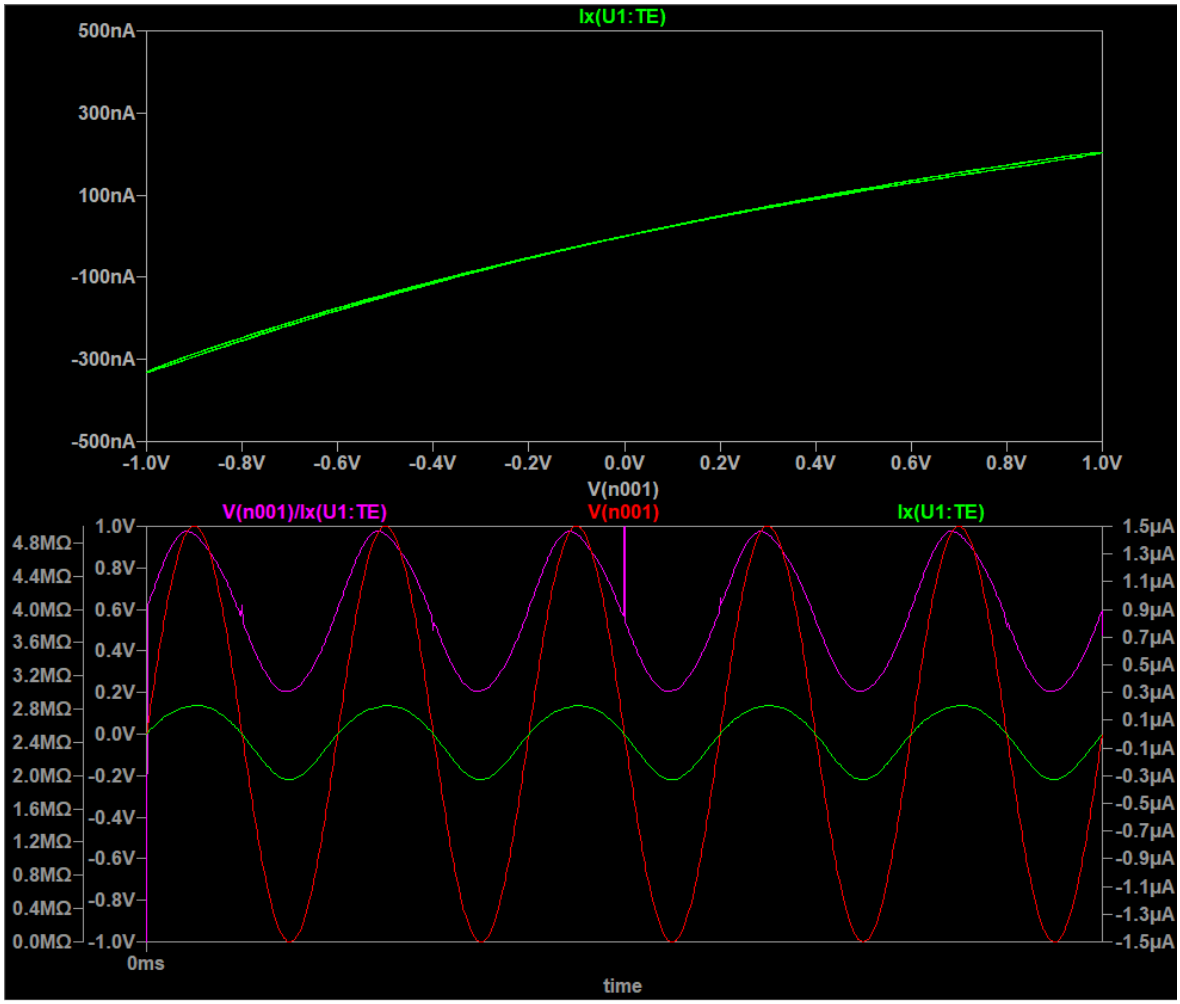


Figure B.32: Memristor model from the University of Michigan found at [44]. No resistor, 100Hz 1V sine wave, no offset.

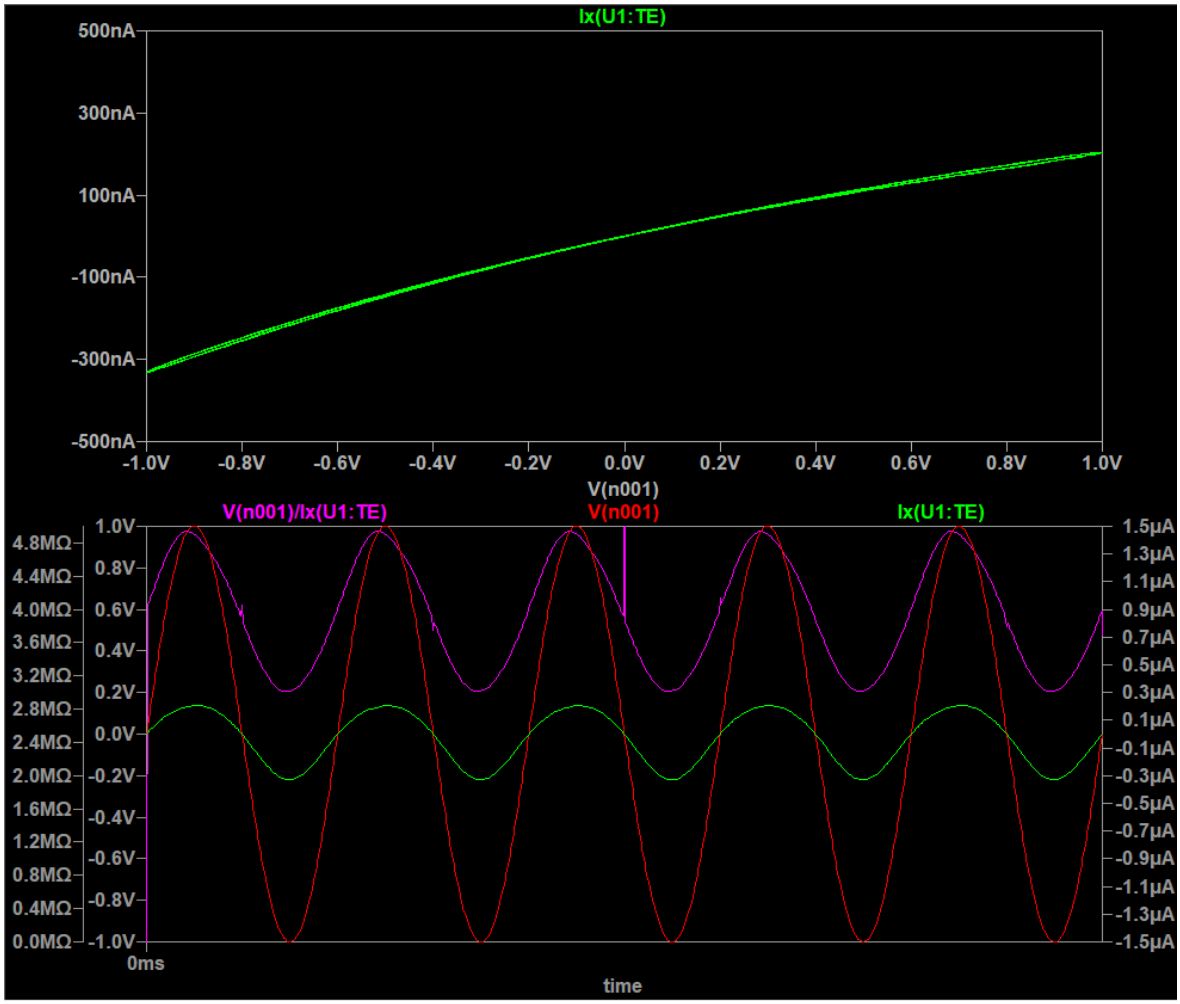


Figure B.33: Memristor model from the University of Michigan found at [44]. No resistor, 1000Hz 1V sine wave, no offset.

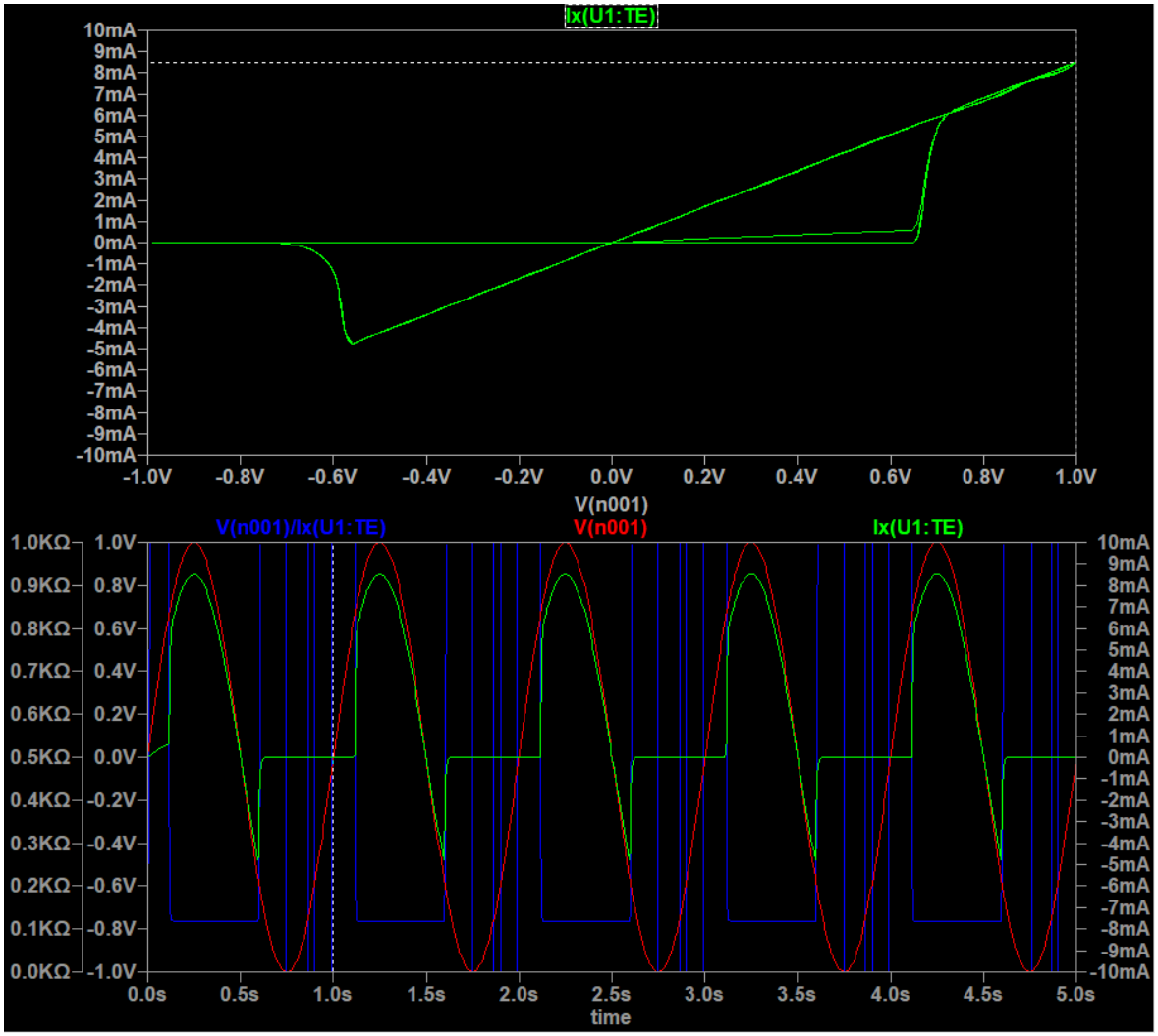


Figure B.34: Memristor model from the from Yakopcic found at [44]. No resistor, 1Hz 1V sine wave, no offset.

1.3 The Yakopcic Model

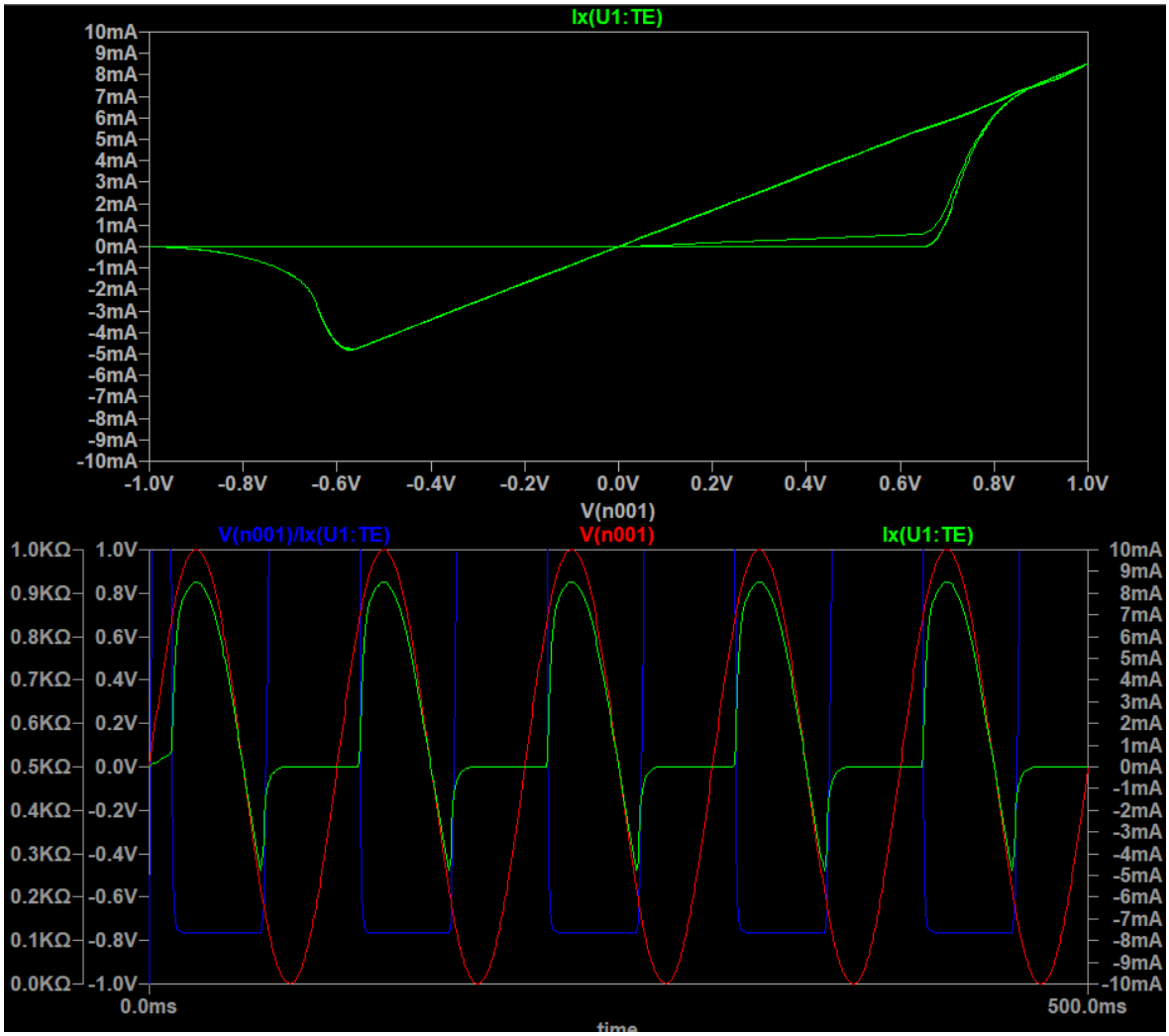


Figure B.35: Memristor model from the from Yakopcic found at [44]. No resistor, 10Hz 1V sine wave, no offset.

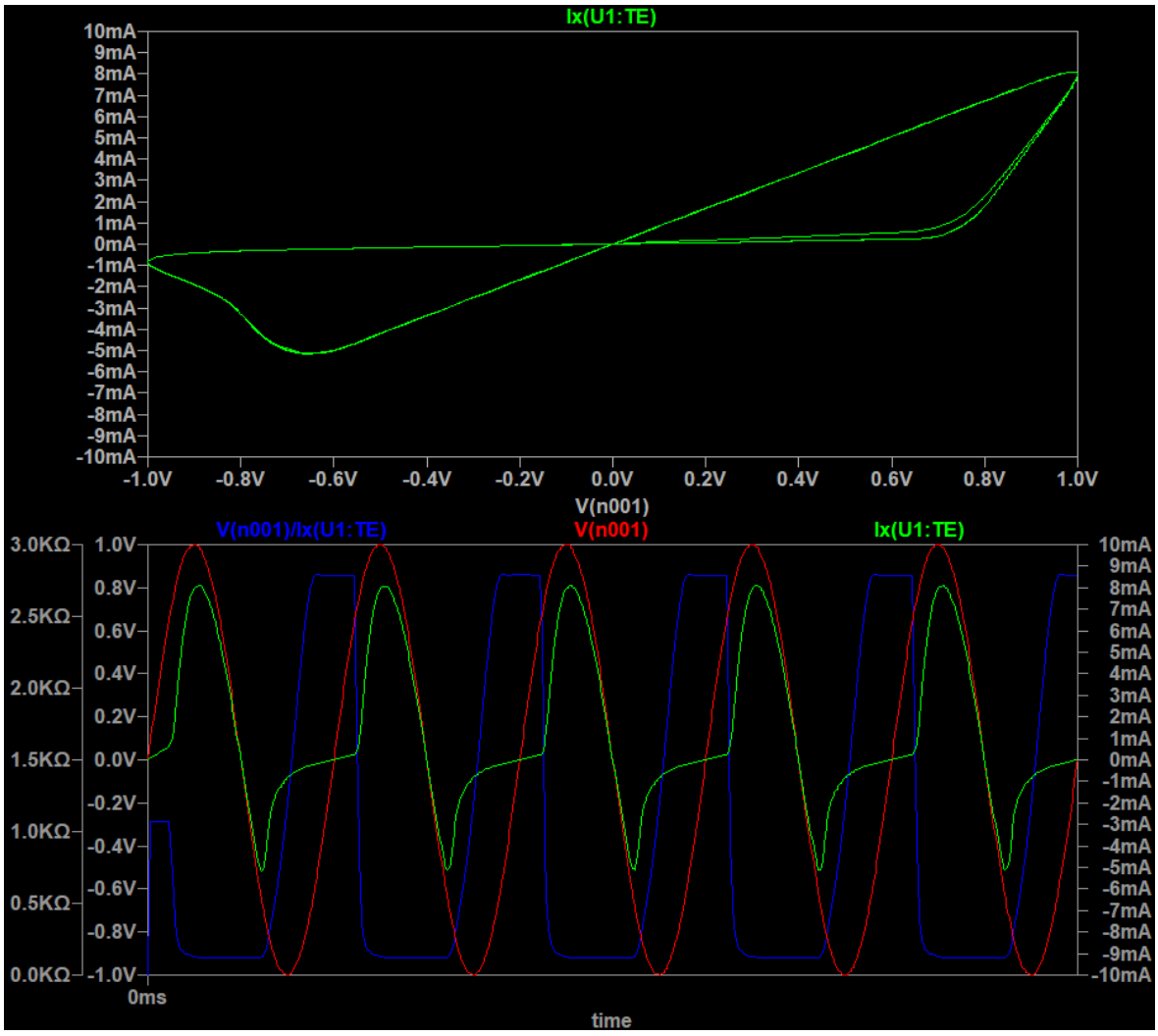


Figure B.36: Memristor model from the from Yakopcic found at [44]. No resistor, 100Hz 1V sine wave, no offset.

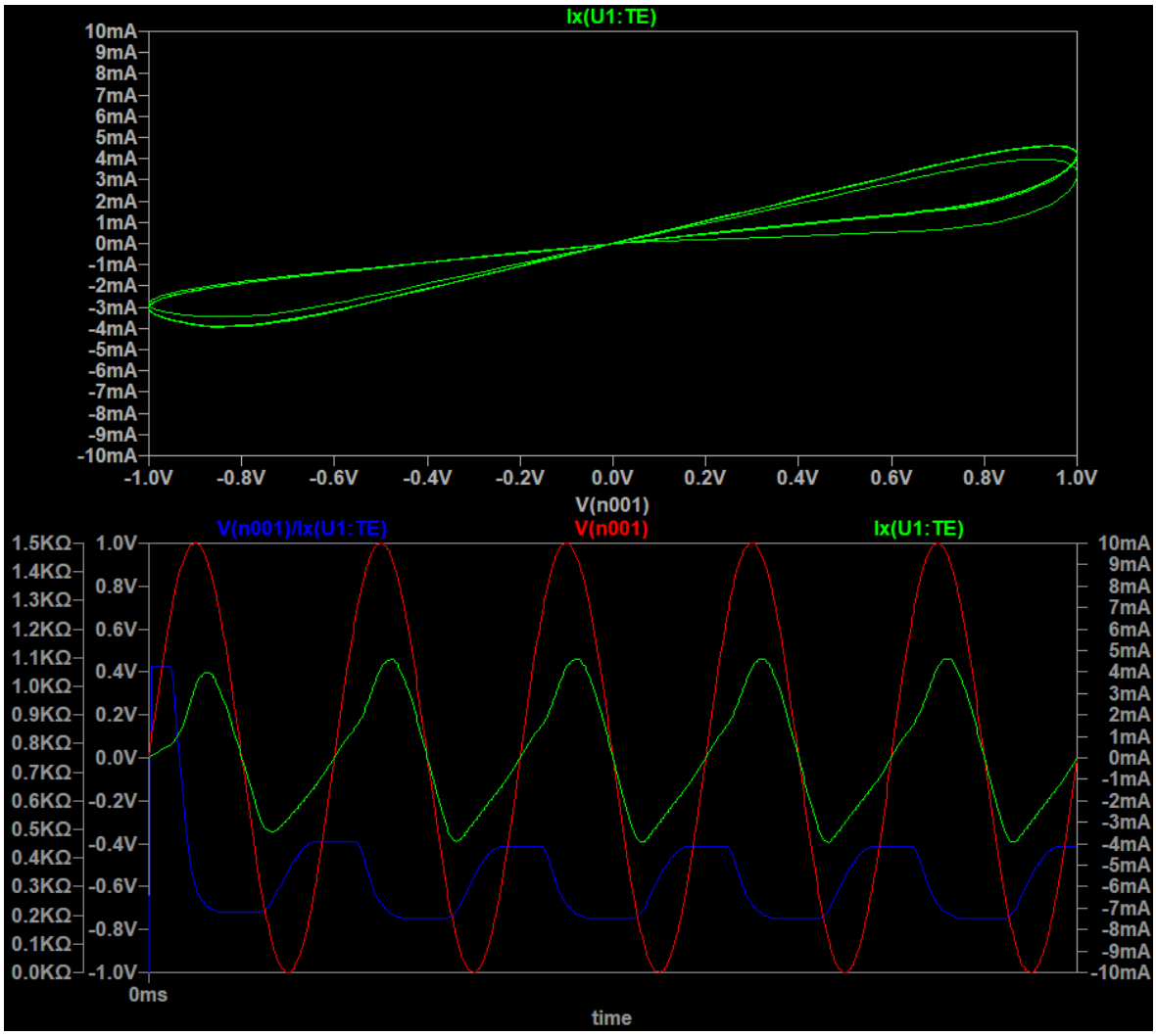


Figure B.37: Memristor model from the from Yakopcic found at [44]. No resistor, 1000Hz 1V sine wave, no offset.

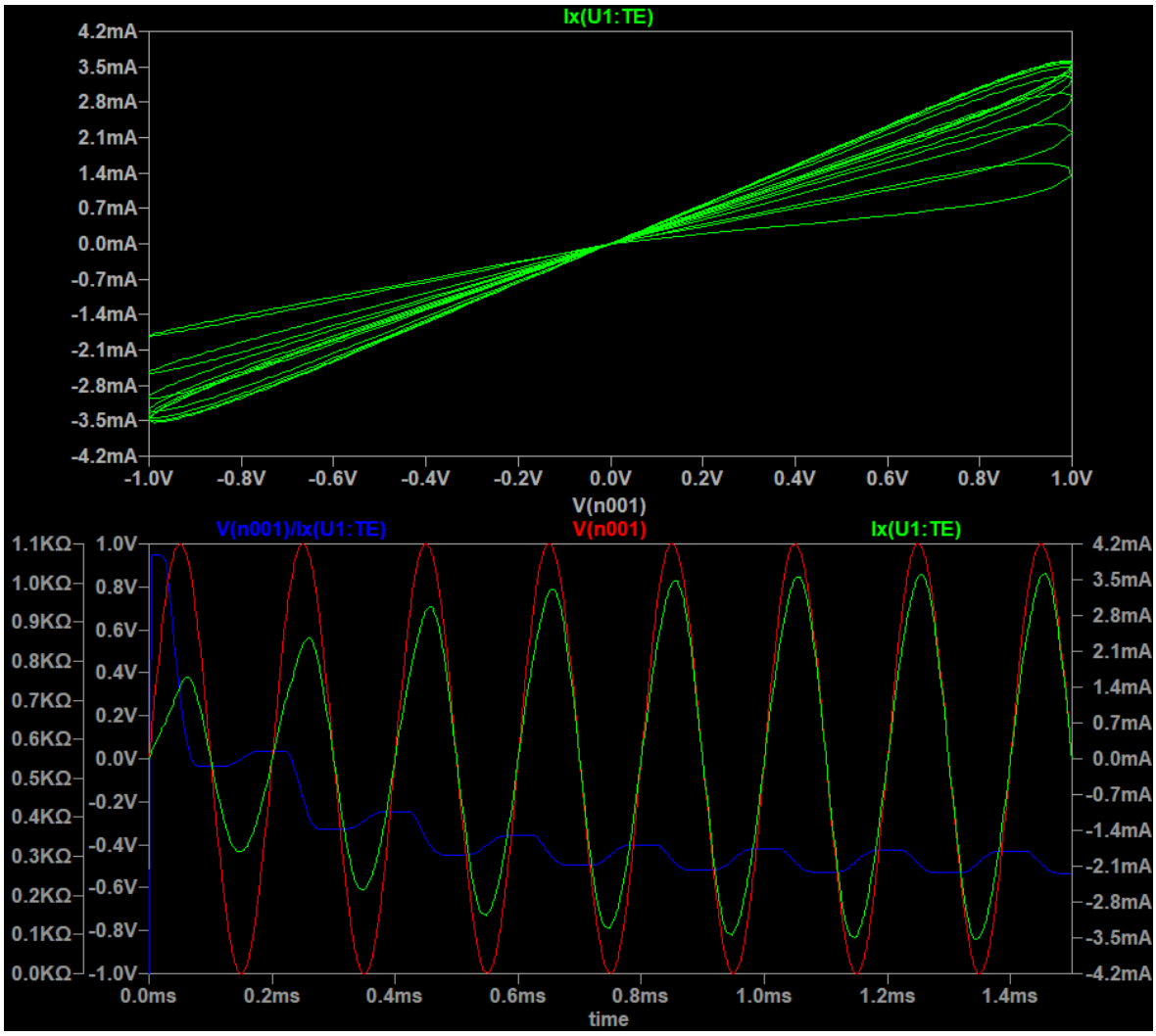


Figure B.38: Memristor model from the from Yakopcic found at [44]. No resistor, 5000Hz 1V sine wave, no offset.

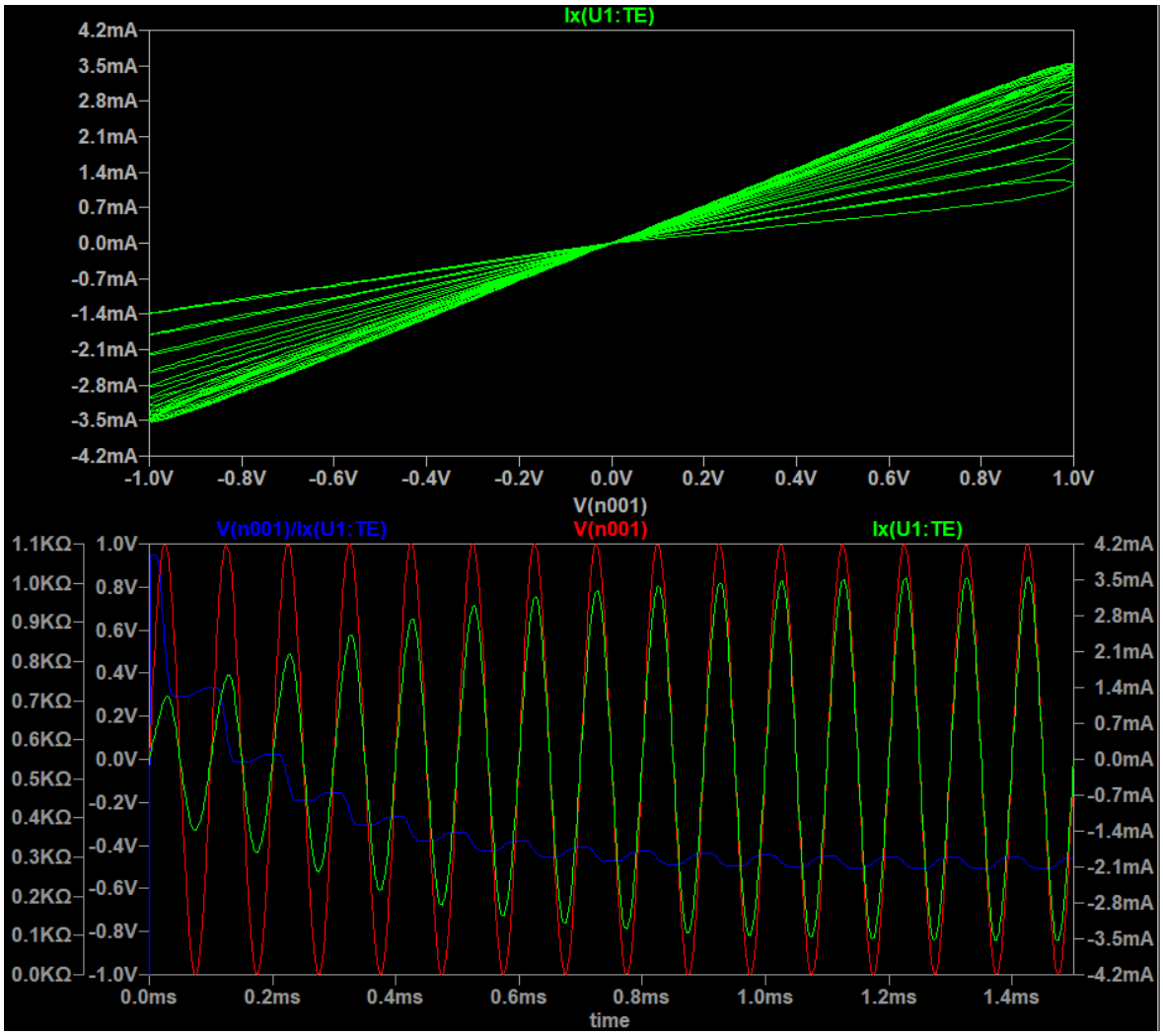


Figure B.39: Memristor model from the from Yakopcic found at [44]. No resistor, 10000Hz 1V sine wave, no offset.

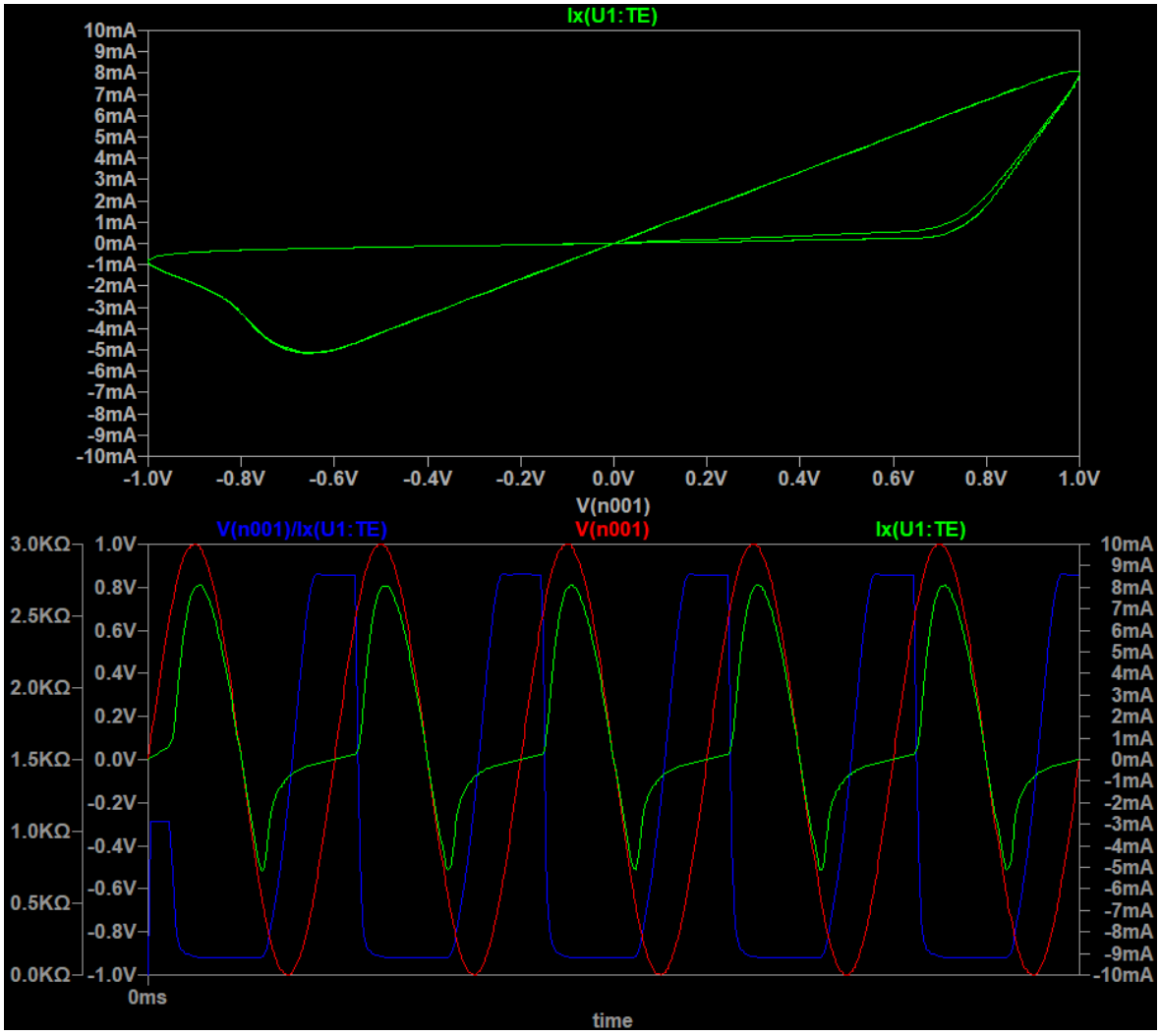


Figure B.40: The graph shows that the memristor has a "relatively" low maximum memristance of below 3KΩ at 100Hz

Infinite memristance

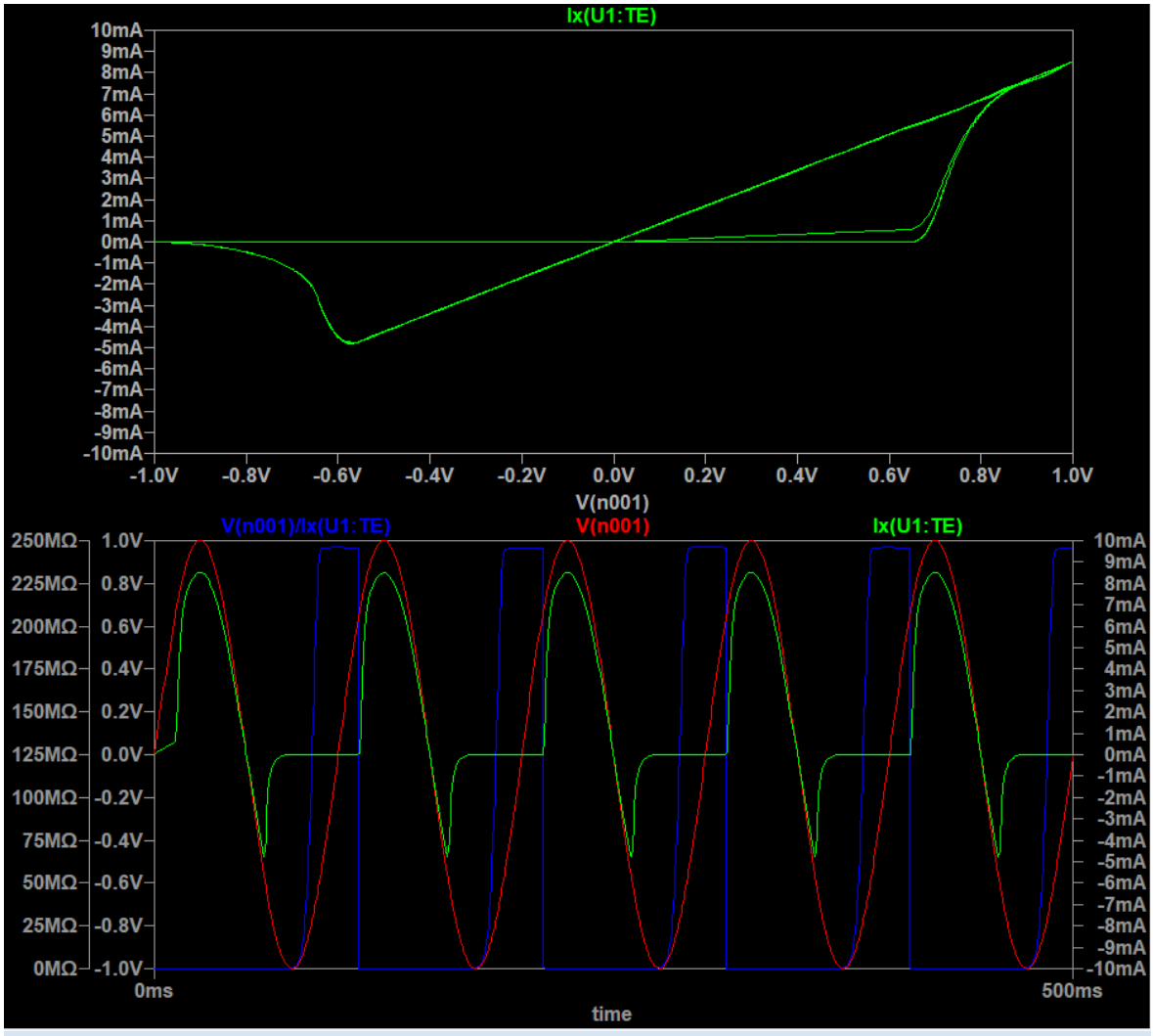


Figure B.41: In this graph it is shown that the memristor has a similar shape to the memristor line in blue at 10Hz. The maximum memristance is here at just below 25MΩ

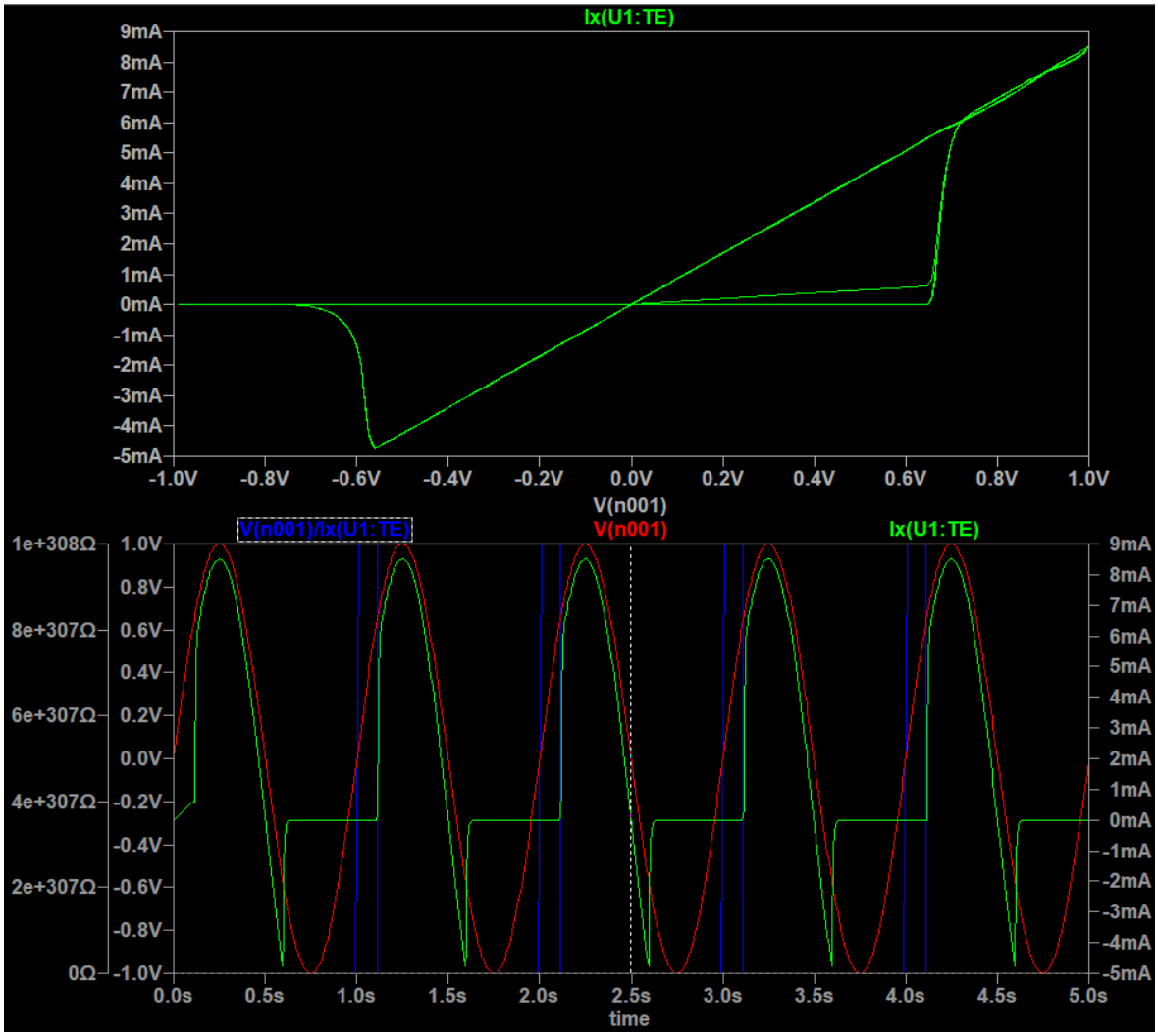


Figure B.42: In this graph it is shown that the memristor has a sine at 1Hz. The maximum memristance is now outside the limit of LTspice, we can view this as infinite as it is more then $10^{308}\Omega$

2 Making the Circuit

In the following section figures made during the design of the circuit, both from LTspice and the real circuit is included with a brief explanation of why it is done and/or what we see. These are included in this appendix to make recreating results easier as well as being available for anyone who might want to use the data.

The order is chronological to when they where produced and there are some logical jumps as new approaches where discovered or realised.

The older figures are black before I figured out I can change the visuals in LTspice, the ones I then used in the paper are recreated to be white in order to be more visually pleasing in the master thesis.

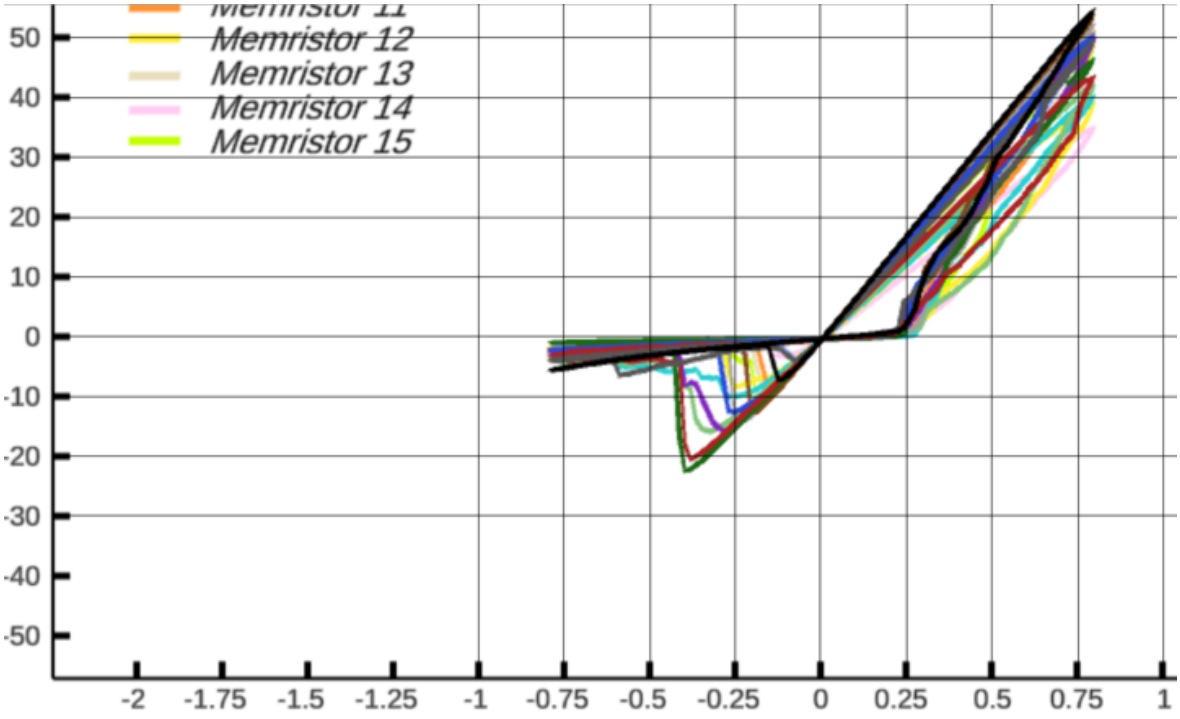


Figure B.43: IV-curves from the uMemristorToolbox paper of S. Bos 2020 [4]. This is the same memristor type used in this paper and the graph was one of the data sources used to try to tweak the simulation parameters of the LTspice model.

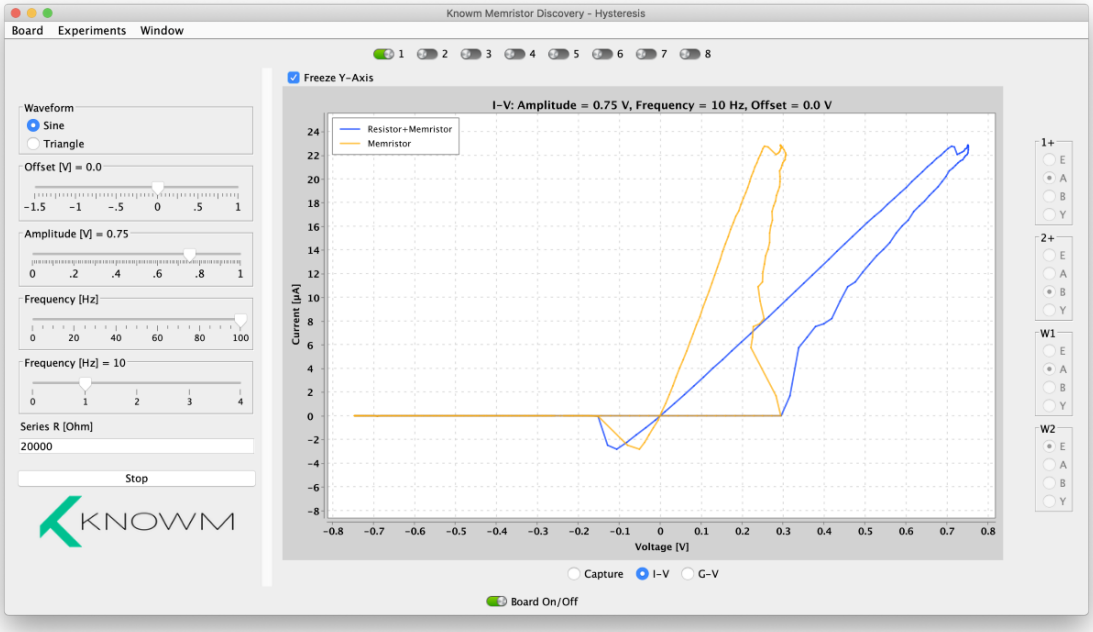


Figure B.44: IV-curve from Knowm using the same memristor used in the thesis circuit.

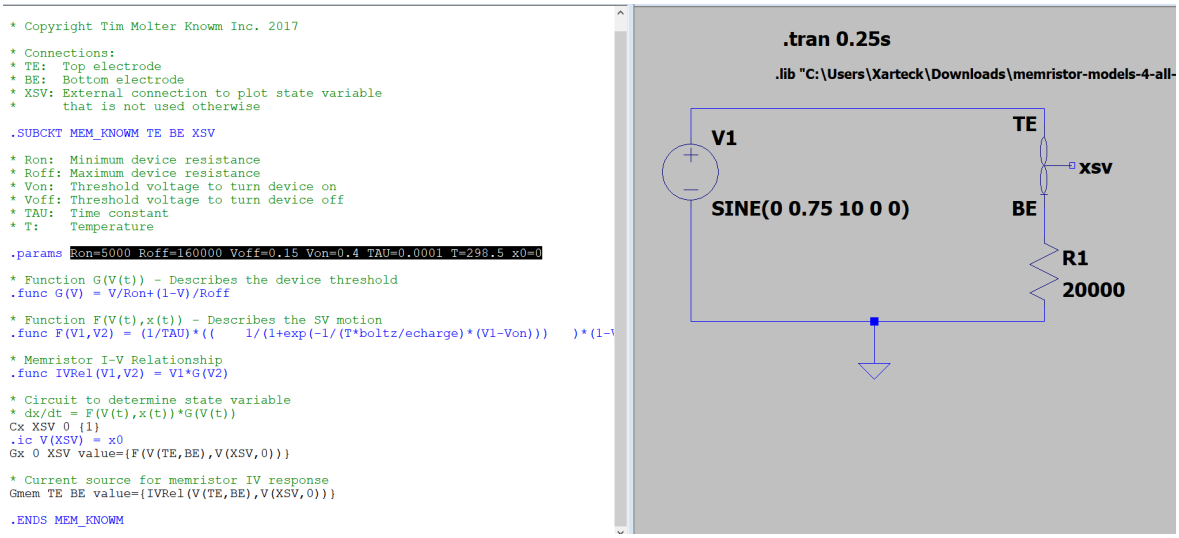


Figure B.45: The parameters the model ended up at after this attempt to tweak them as well as the circuit.

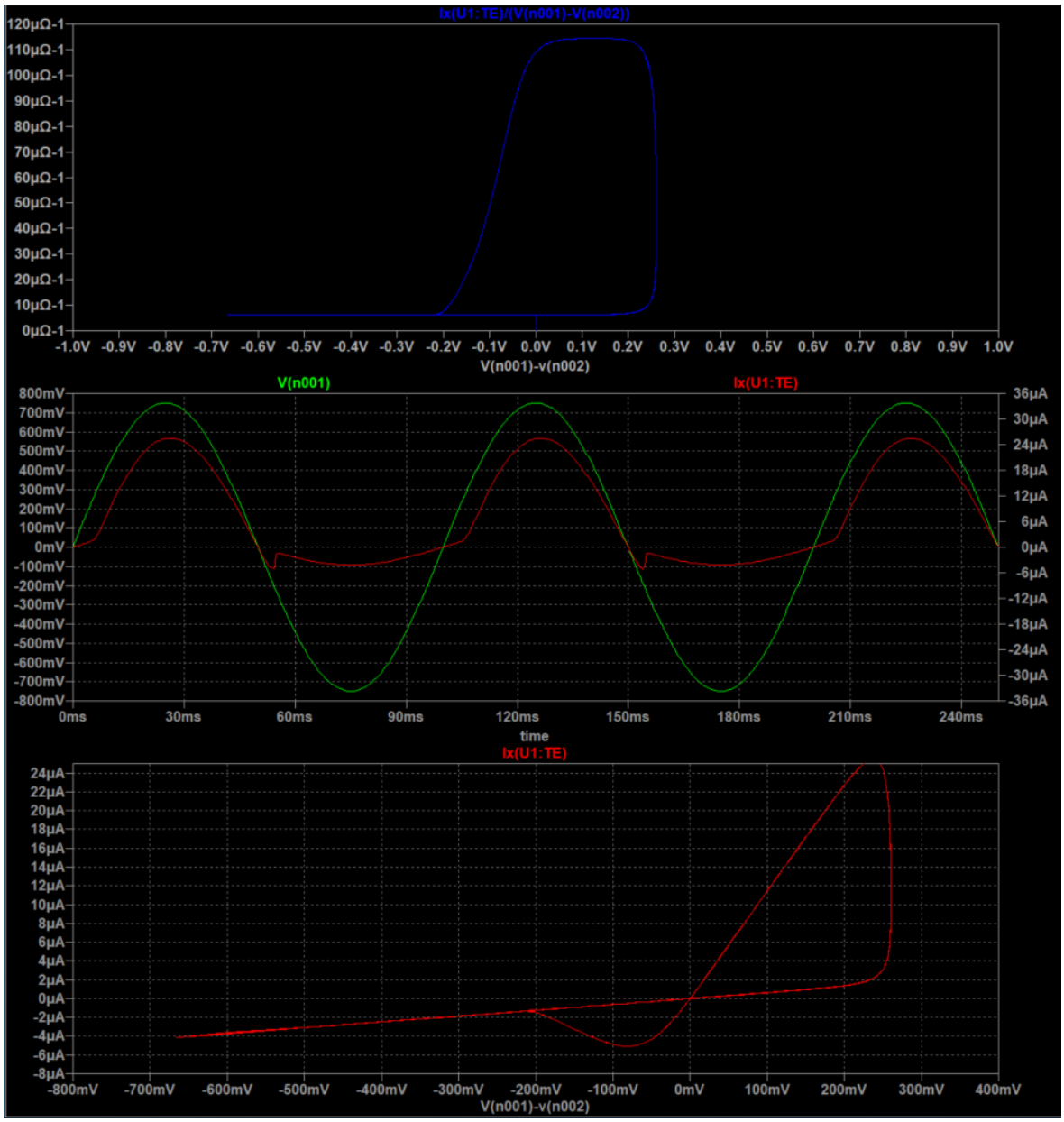


Figure B.46: Final graphs after trying to recreate the behaviour of the memristor model to more mach the Knowm tungstein memristor used in the thesis, in this graphs a 0.75V sine wave with a 20K Ω series resistor at 10Hz with no DC offset; The first graph is the SI-curve of the memristor; The second graph shows the voltage and current over the memristor, over the run; The last graph shows the IV-graph of the memristor model.

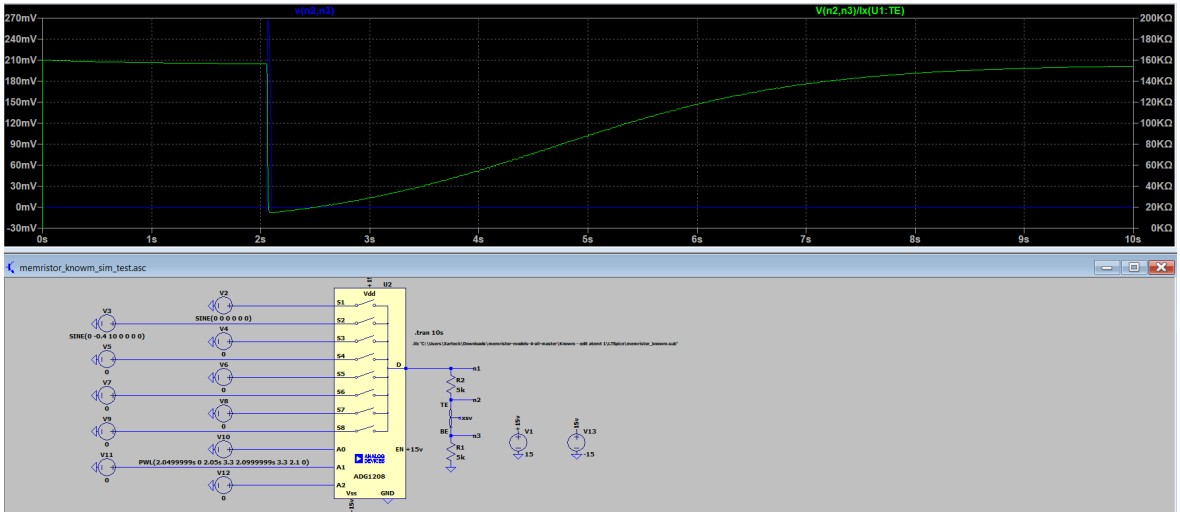


Figure B.47: This figure shows the memristor changing it's memristance over time even when the voltage over it is 0V.

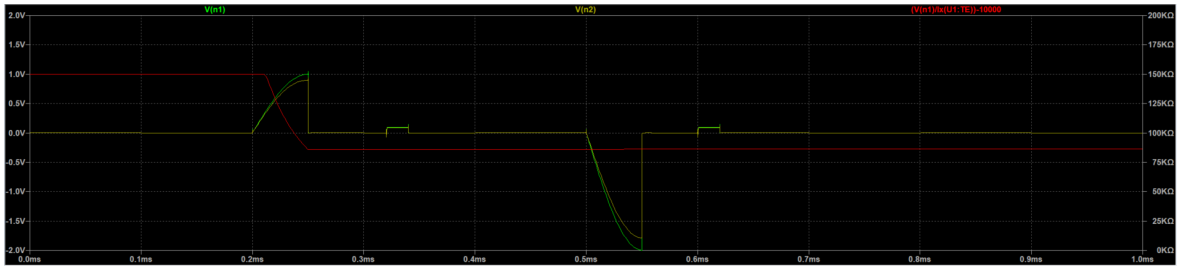


Figure B.48: 1v write -2v erase with 0,1v 20μs read pulses on the Knowm memristor model using the parameters: .params Ron=1500 Roff=150000 Voff=0.15 Von=0.350 TAU=0.005 T=300 x0=0

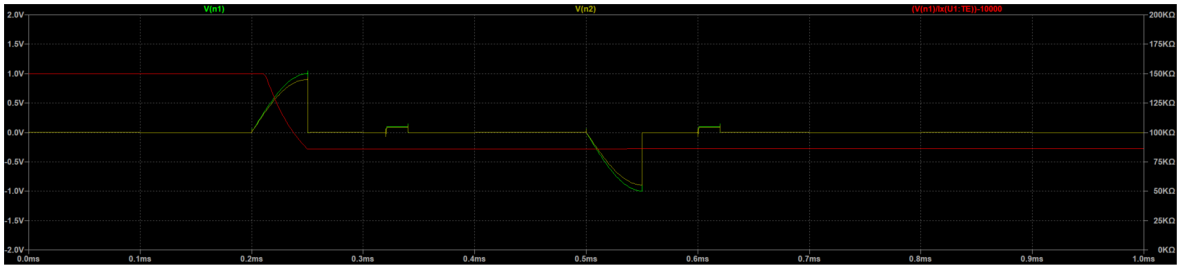


Figure B.49: 1v write -1v erase with 0,1v 20μs read pulses on the Knowm memristor model using the parameters: .params Ron=1500 Roff=150000 Voff=0.15 Von=0.350 TAU=0.005 T=300 x0=0

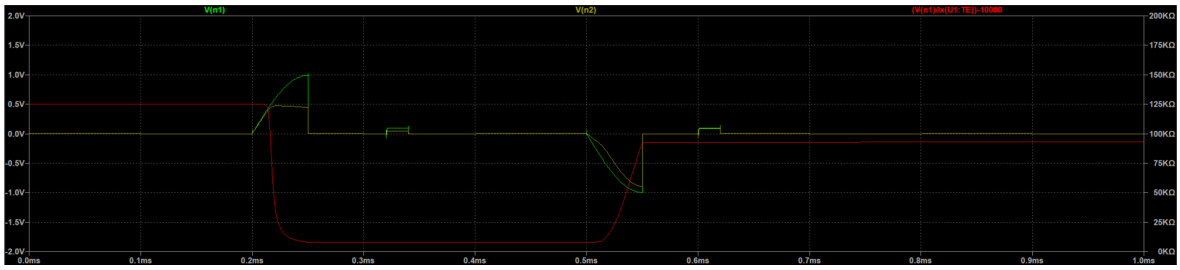


Figure B.50: 1v write -1v erase with 0,1v 20μs read pulses on the Knowm memristor model using the parameters: .params Ron=2500 Roff=125000 Voff=0.19 Von=0.52 TAU=0.00001 T=300 x0=0

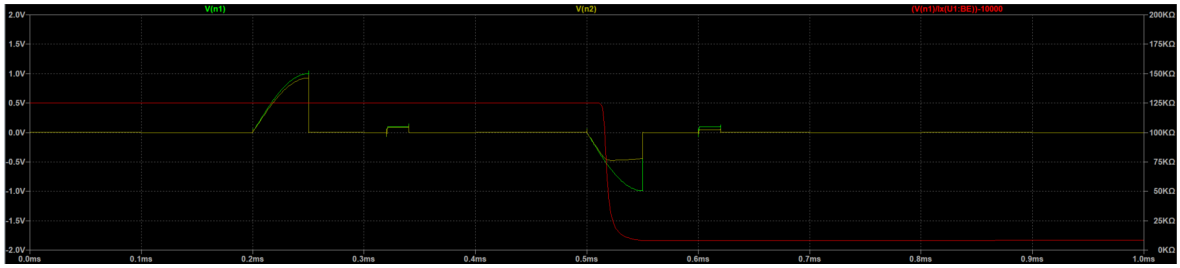


Figure B.51: Flipped memristor. 1v write -1v erase with 0,1v 20μs read pulses on the Knowm memristor model using the parameters: .params Ron=2500 Roff=125000 Voff=0.19 Von=0.52 TAU=0.00001 T=300 x0=0

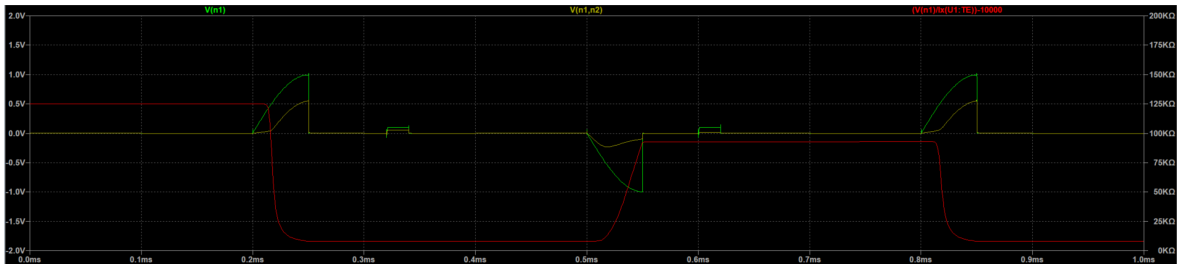


Figure B.52: Breakthrough, flipped back to normal but read the paper right and read the voltage drop over the load resistor, not the memristor. 1v write -1v erase with 0,1v 20μs read pulses on the Knowm memristor model using the parameters: .params Ron=2500 Roff=125000 Voff=0.19 Von=0.52 TAU=0.00001 T=300 x0=0

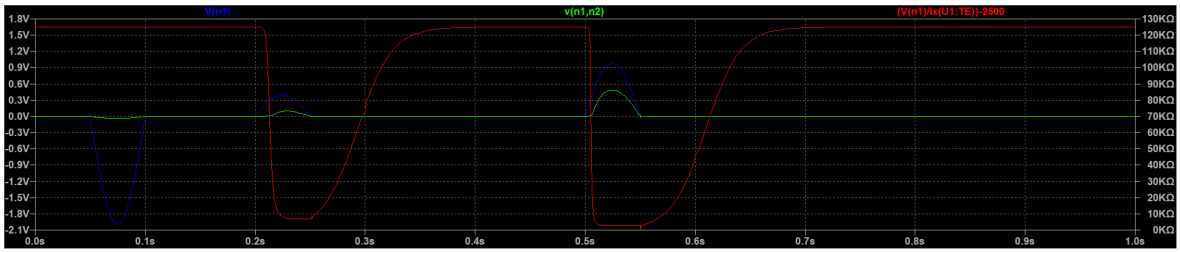


Figure B.53: First real atempt to recreate ternary storage using the 1, 0.4, and -2 10Hz half sine waves used in the uMemristor toolbox. The figure shows the model loosing its stored value instanly, going to high resistance within 0.15s. parameters: .params Ron=2500 Roff=125000 Voff=0.19 Von=0.52 TAU=0.00001 T=300 x0=0

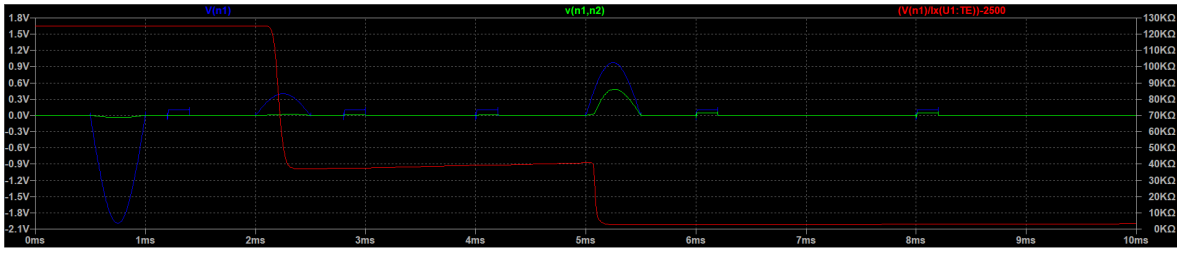


Figure B.54: same settings as 7 but at 1000Hz, here there are 3 states

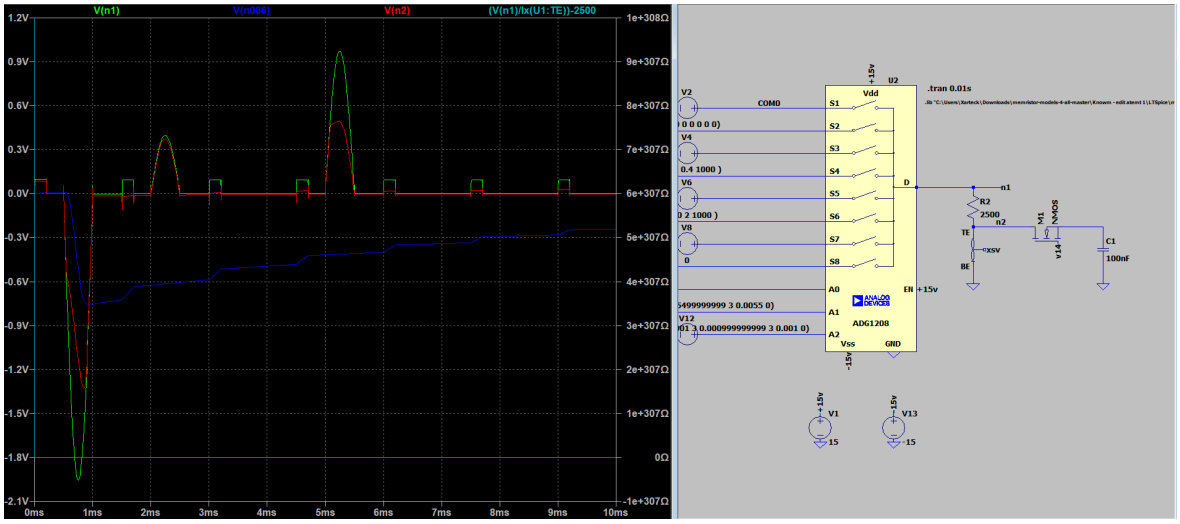


Figure B.55: Trying to add a sample_hold element to the memristor volage, not sure what is going on. used a 100nF capasitor

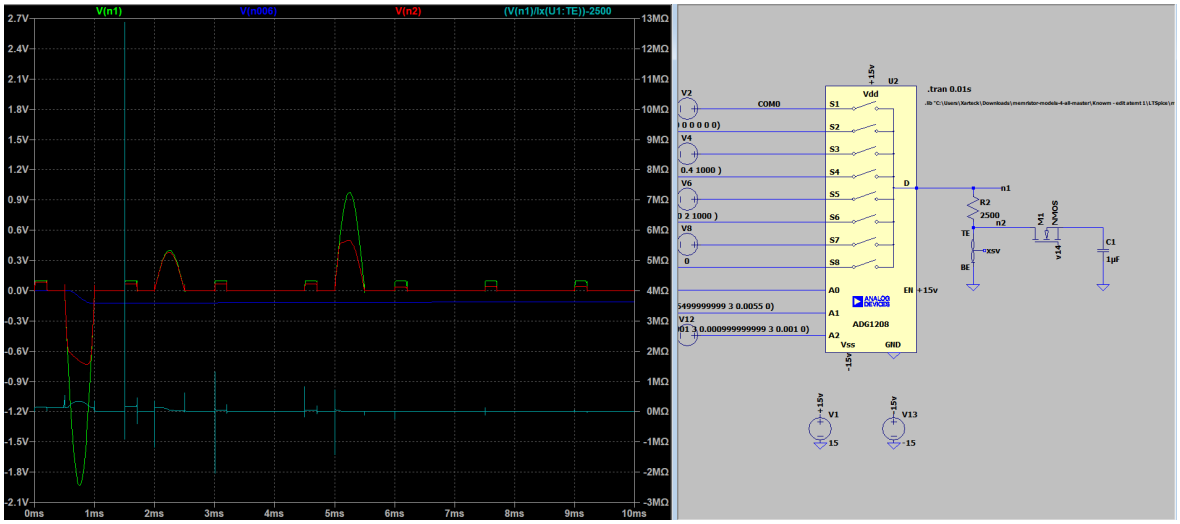


Figure B.56: Changed the capasitor to 1μf, still confused but now there is some memristance?

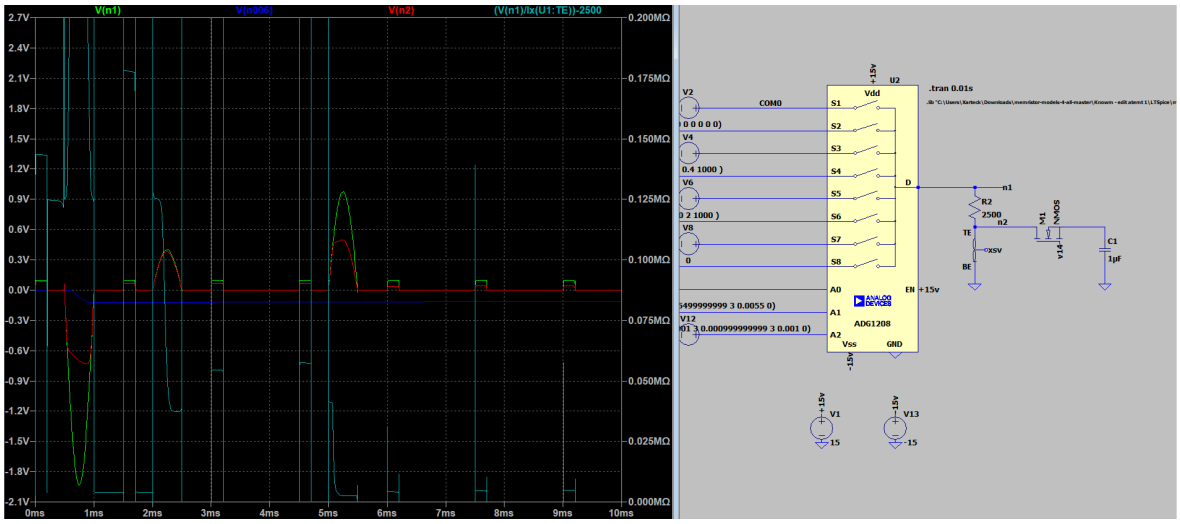


Figure B.57: Changed the NMOS to PMOS, still confused.

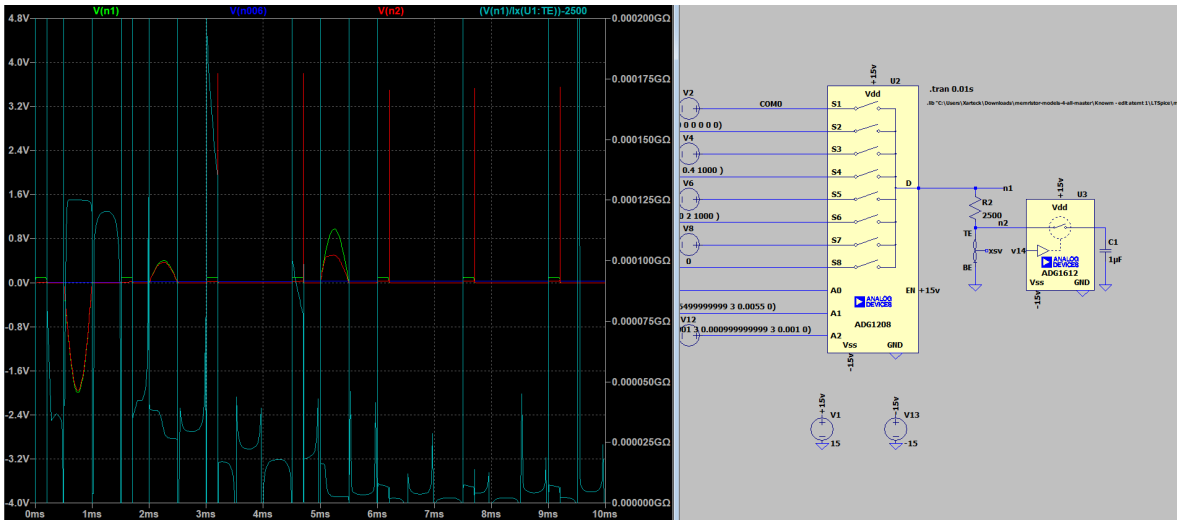


Figure B.58: Changed the MOS to a single-pole/single-throw switch. I think there is bleed through?

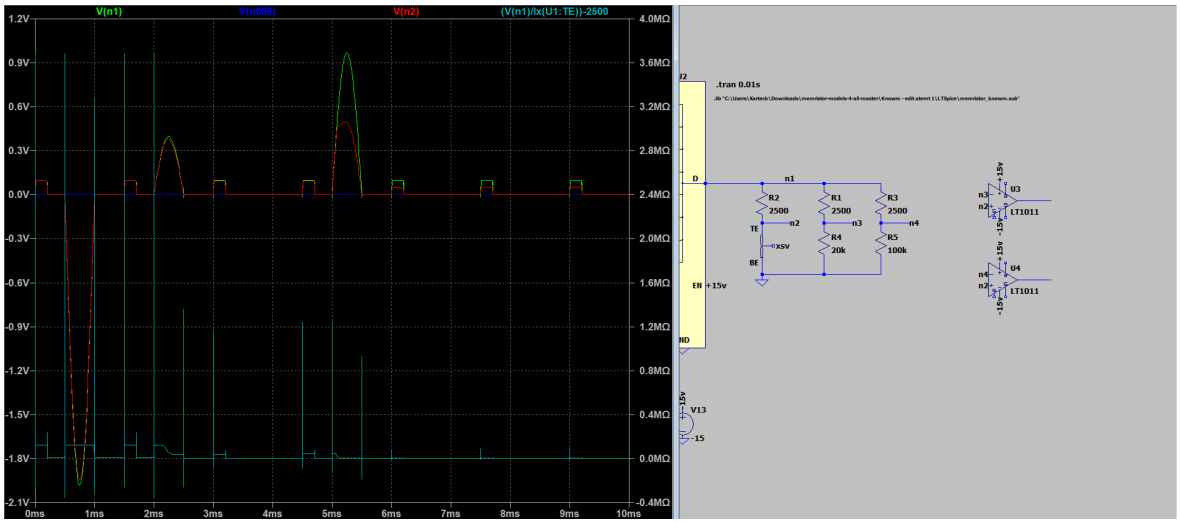


Figure B.59: Here I deside to try to compare the signals first and will rather hold the outputs but I think there is leakage here too?

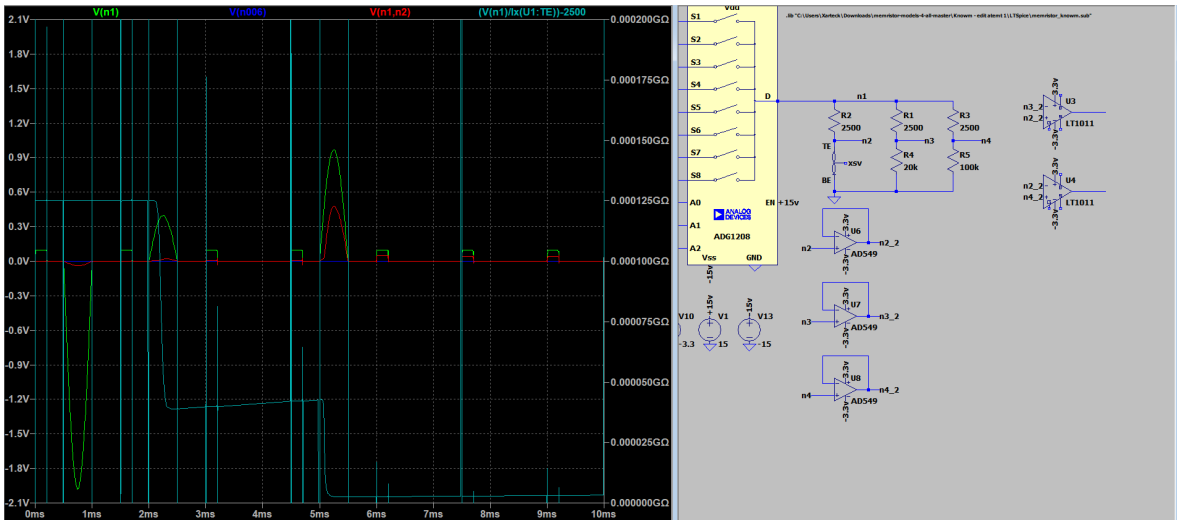


Figure B.60: In order to atempt to mitigate the problem I sent the node values through opamps to shield the system and the readings, this seam to have work well as the graph looks good exept for some artifacts in the resistance calculations when the system changes, but it does not seem to affect the real values.

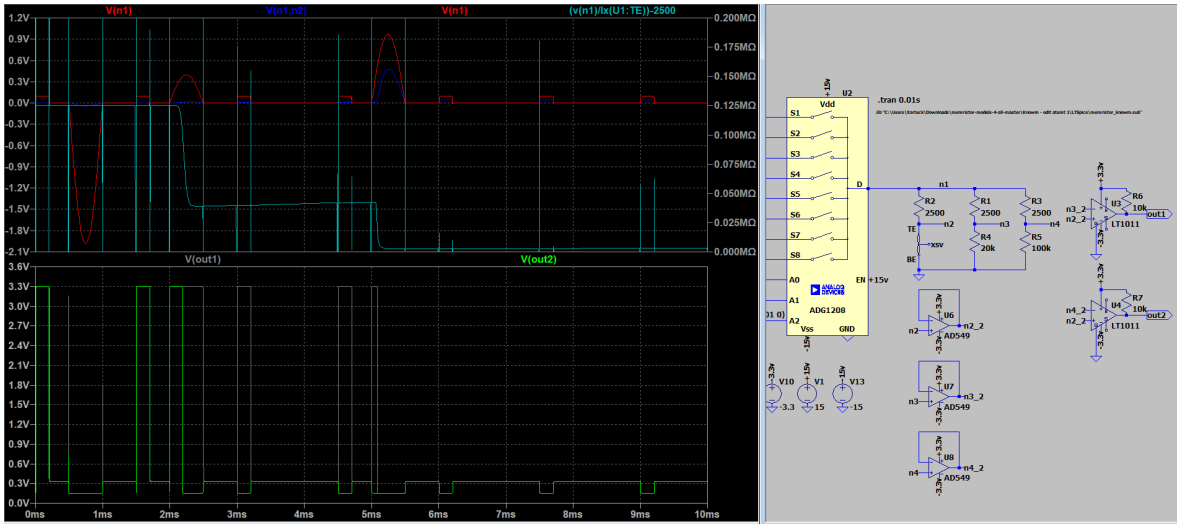


Figure B.61: In this iteration I have fixed the comperators and graph thair output, there is no hold or selection on when to compare in this iteration.

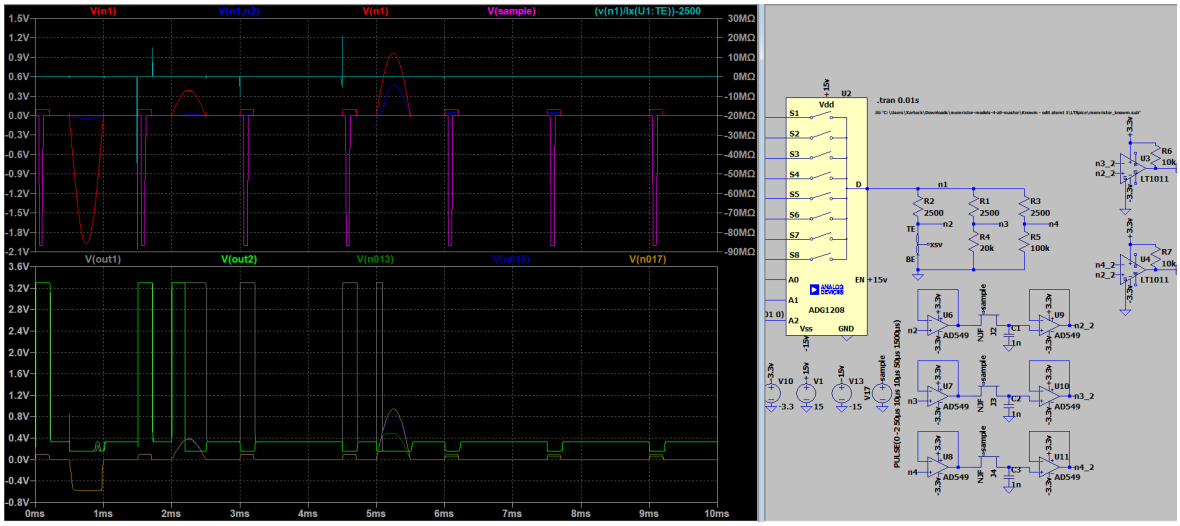


Figure B.62: In this itteration I added switches to atempt to only compare the signals during the read pulse. the switches do not seem to isolate the read pulses as shown on the bottom graph being sampled after the switch. other options need to be explored

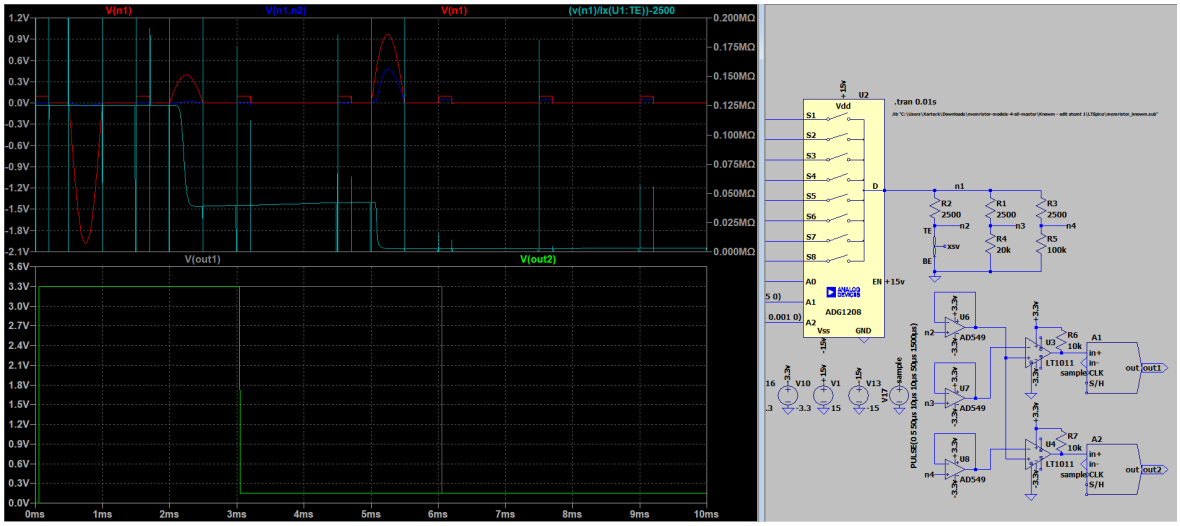


Figure B.63: Here I decided to use state/hold elements on the output insted of before the comparison because the levels before the comparison are more crutial that they are correct and tle outputs do not need to be as exact, this seem to work well and the states save between read pulses well. the read circuit seems to funtion correctly now.

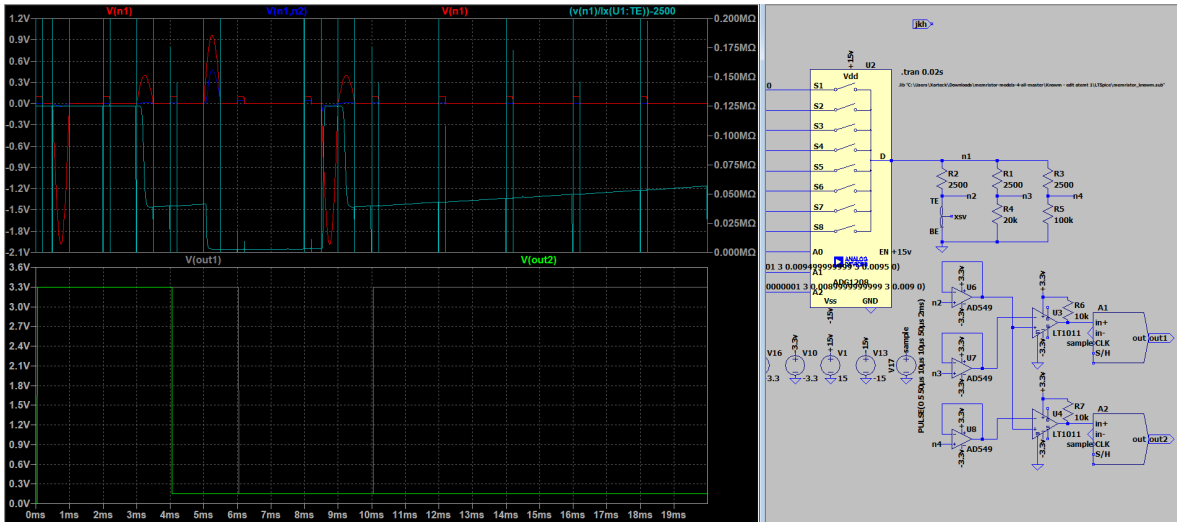


Figure B.64: Same setup as in 17 but I extended the period between read cycles so that I could erase the memristor and write a new value to it between two read pulses, this is shown in the graphs with a -2V pulse followed by a 0.4v pulse at 8ms to go from a low resistanse state to a mid resistance state.

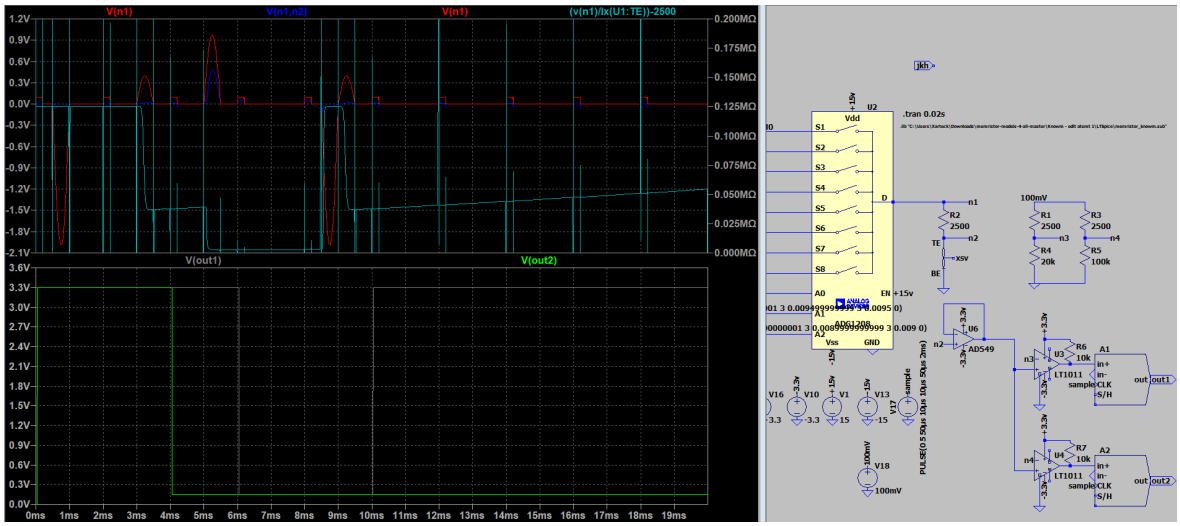


Figure B.65: An atempt to seperate the resistors that are used to compare the memristor value to determin the state and output. Also then put the compare values directly into the comperator eliminating two op-amps. It seems to work well

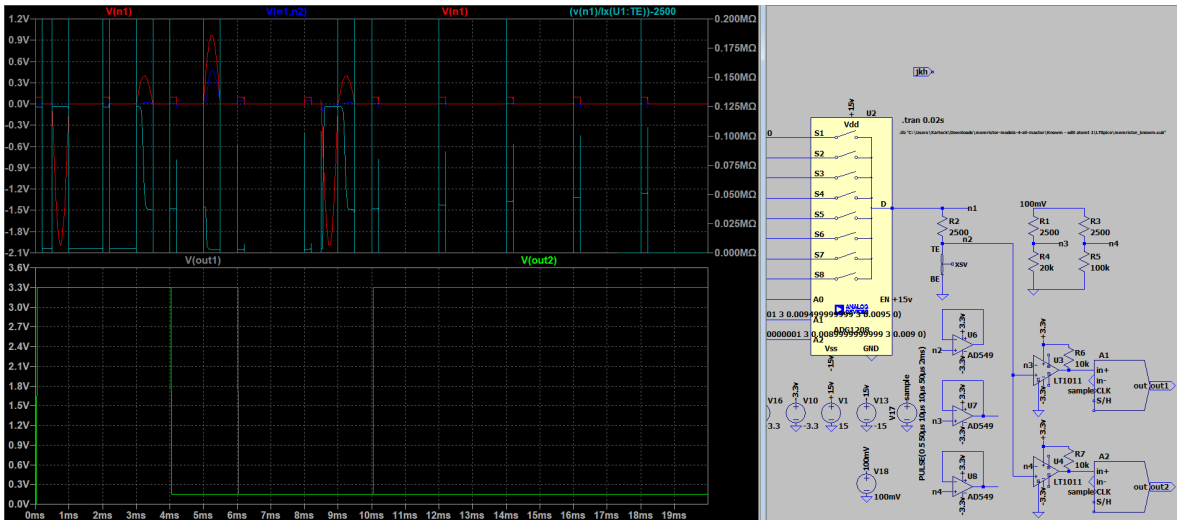


Figure B.66: In this iteration I atempted to remove the op amp on the memristor as well, at first glance it seams to mess up the memristor value completly; But at a closer glance it the memristance only seem wrong due to a calculation error when there is no voltage across the devive, the output states and the memristance value can be seen as correct during the read pulses and write operations.

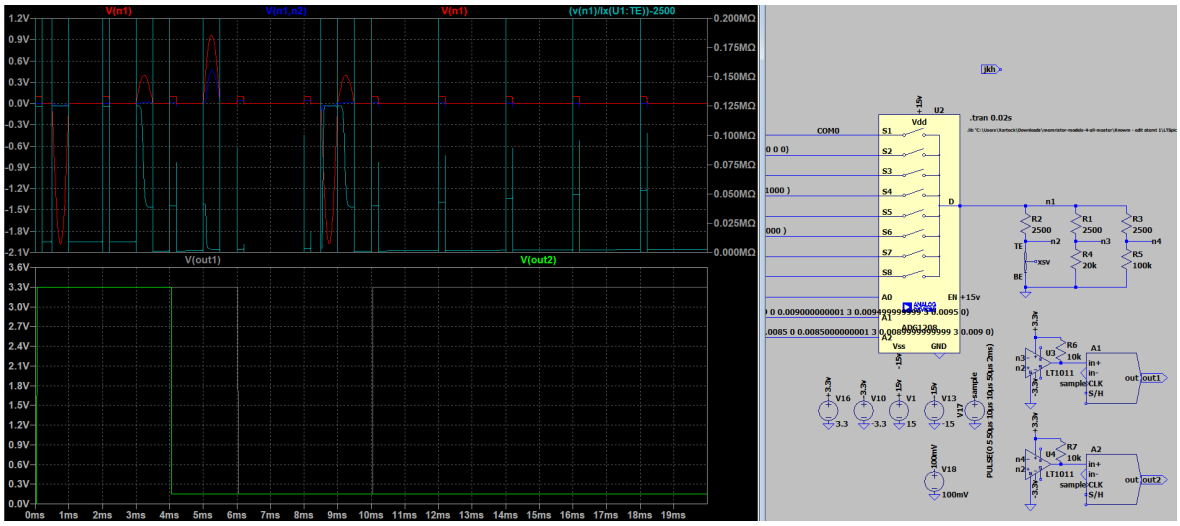


Figure B.67: In this iteration the resistors where returned to the memristor circuit to see if it disturbed the memristor value. The main advantage ti this is that voltage over the resistors are only high durring read and write events rather then continiously. This should save energy. One observation is that is semas that the memristance decays faster when they are attached, I am unsure if this is due to the problems with the model or if it will be true with a real memristor.

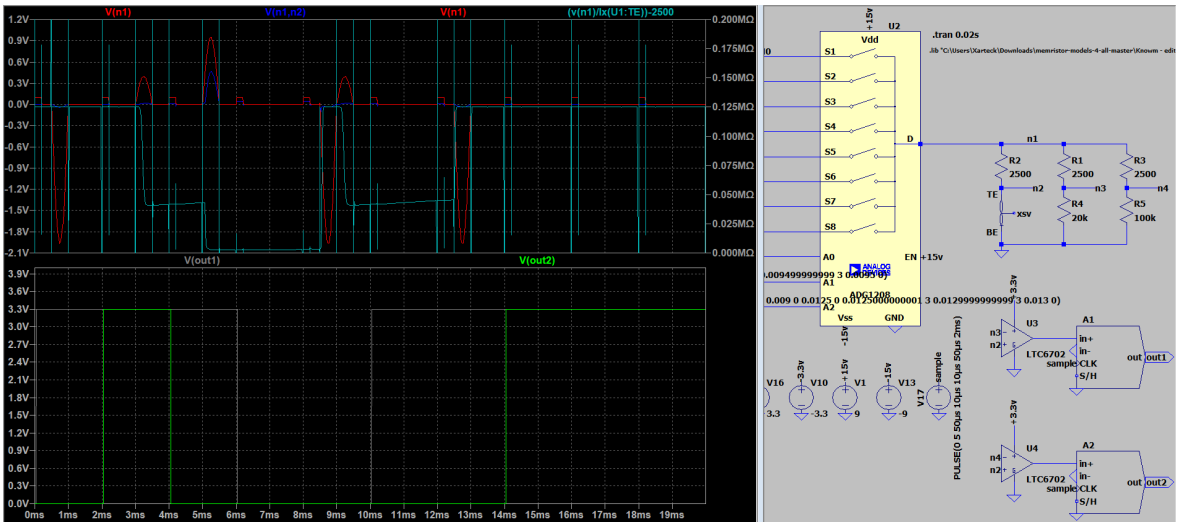


Figure B.68: In this iteration I swapped out the comperators to ones that only accesses the positive rail and ground, eliminating the need for a negative voltage rail at that level. It allso does not require a resistor between the output and the positive rail, making the output close to 0 voltage in the low state. This samev power and cleans up the value.

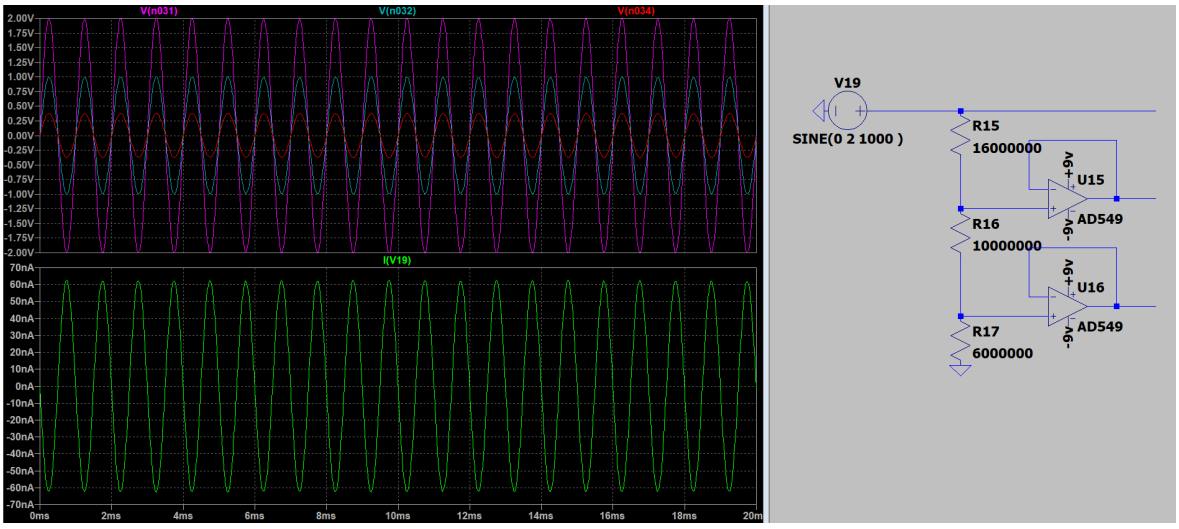


Figure B.69: In this Figure is the first atempt at spliting a single sine voltage source into all sine waves needed for the write and erase cycles. In this iteration I use a voltage devider to limit the voltage down from 2 voltage. The 2 and 1 voltage waves look right on the mark, but the 0.4 amplitude wave is a little short at 0.37V. The leakage current is also at 60nA at the peak of the sine wave.

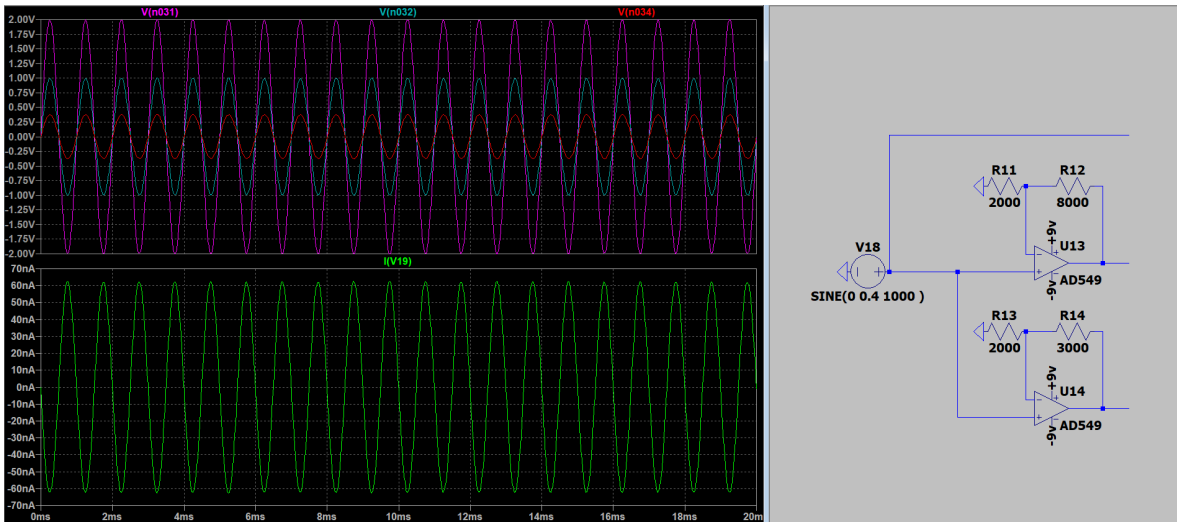


Figure B.70: In this second atempt I insted use the amplifying setup of the op-amps to boost a 0.4V sine wave. In this iteration all voltage levels are spot on and I've left in the amp of the voltage devider iteration for comperation in green. As the graph shows this solution has considerably lower leakage current at 1.4nA, but it introduces a new leakage over the op-amp that adds up to 3μA with the current resistor levels.

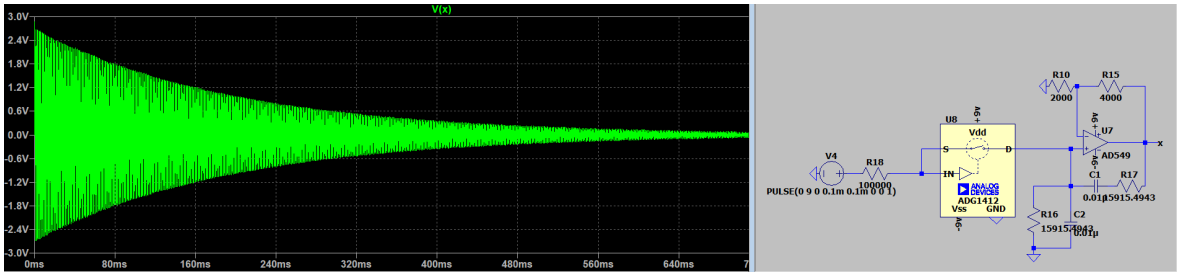


Figure B.71: This is the first atempt at a sine ocelator using a Wien bridge oscillator, one problem is that it seems to need a pulse to get it going, and then its voltage level drops rapidly as it goes. This does not seem like a good solution got the circuit.