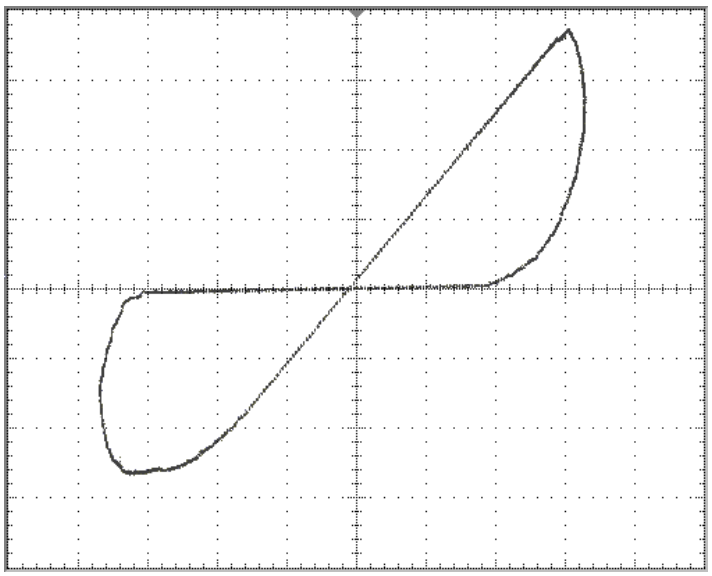
Universidade de Aveiro Departamento de Eletrónica, Telecomunicações e Informática
2013

João Diogo Capela Duarte

Memristors



Universidade de Aveiro
2013
Departamento de Eletrónica, Telecomunicações e Informática

João Diogo Capela Duarte

Memristores

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia Eletrónica e Telecomunicações, realizada sob a orientação científica do Doutor Luís Filipe Mesquita Nero Moreira Alves (orientador), Professor Auxiliar do Departamento de Eletrónica, Telecomunicações e Informática da Universidade de Aveiro e do Doutor Ernesto Fernando Ventura Martins (cooriendator), Professor Auxiliar do Departamento de Eletrónica, Telecomunicações e Informática da Universidade de Aveiro

Dedico este trabalho aos meus pais e à minha falecida madrinha.

o júri

Presidente	Professor Doutor José Carlos Esteves Duarte Pedro Professor Catedrático do Departamento de Eletrónica, Telecomunicações e Informática da Universidade de Aveiro
Vogal	Prof. Doutor Vítor Manuel Grade Tavares Professor Auxiliar do Departamento de Engenharia Eletrotécnica e de Computadores da Faculdade de Engenharia da Universidade do Porto
Vogal	Prof. Doutor Luís Filipe Mesquita Nero Moreira Alves Professor Auxiliar do Departamento de Eletrónica, Telecomunicações e Informática da Universidade de Aveiro

agradecimentos

Aos meus orientadores, os professores Luís Nero e Ernesto Martins, por me proporcionarem a oportunidade de trabalhar neste tema fascinante.
Aos colegas do Laboratório de Circuitos e Sistemas Integrados do Instituto de Telecomunicações pelo ambiente de camaradagem e pela ajuda prestada.
Ao Engenheiro João Lopes pela sua amizade e ajuda insubstituível ao longo do meu percurso académico.
Aos meus pais António Duarte e Ana Capela e à minha falecida madrinha Rosa Maria Capela por terem acreditado em mim, por me terem apoiado e por terem ajudado a tornar-me no homem que hoje sou.
A todos os meus amigos e colegas ao longo destes anos.

A todos um grande bem-haja.

palavras-chave

Curva tensão-corrente, emulador, função janela, Memristor, Dispositivos e Sistemas Memristivos, TiO_2.

resumo

O memristor foi proposto por Leon Chua em 1971 apenas por uma questão de complemento matemático, uma ideia que não teve grande aceitação na comunidade científica. Só décadas mais tarde, depois do anúncio da HP em 2008 é que os memristors começaram a ser vistos como elementos realizáveis e não como meras curiosidades matemáticas. Estes dispositivos apresentam características distintas dos demais dispositivos eletrónicos conhecidos. Além de serem elementos passivos, são caracterizados pelos seguintes postulados: existência de uma curva característica tensão-corrente com histerese e valor único na origem, diminuição gradual da área definida por esta curva com o aumento da frequência e comportamento puramente resistivo do memristor quando a frequência tende para infinito.

A resposta dos dispositivos memristivos depende bastante das características de amplitude e frequência do sinal de entrada e das suas próprias características internas. Por isso, há uma clara necessidade de descobrir procedimentos e atributos que permitam classificar e categorizar diferentes dispositivos memristivos. Estes atributos, na sua essência, semelhantes às figuras de mérito de dispositivos como díodos ou transístores, permitirão num futuro próximo selecionar dispositivos memristivos para aplicações específicas. Para tentar obter estes atributos, realizou-se uma análise morfológica da área e comprimento das curvas tensão-corrente de vários modelos teóricos de dispositivos memristivos em MATLAB variando as suas características internas, para conjuntos de valores de frequência e amplitude do sinal de entrada. De seguida construiu-se um emulador de um dispositivo memristivo para corroborar os resultados teóricos obtidos. Para tal mediram-se as curvas de tensão-corrente para vários valores de entrada e efetuou-se o cálculo das áreas e comprimentos dessas curvas.

keywords

Emulator, Memristor, Memristive Devices and Systems, TiO_2,voltage-current curve, window function.

abstract

The memristor was proposed by Leon Chua in 1971 only for the sake of mathematical complement, an idea that was not widely accepted by the scientific community. Only decades later, after HP's announcement in 2008 is that the memristors started to be seen as realizable elements and not as mere mathematical curiosities. These devices feature distinct characteristics from the other known electronic devices. Besides being passive, they are characterized by the following postulates: the existence of a characteristic voltage-current loop with hysteresis and single valued in the origin, gradual decrease of the area defined by the loop with the increasing of the frequency and simply resistive behaviour for infinite frequency.

As a memristive device's response depends greatly on the amplitude and frequency characteristics of the input signal and its own internal characteristics. Therefore there is a clear need to find procedures and attributes that allow to classify and categorize various memristive devices. These attributes, in their essence, similar to the figures of merit of devices like diodes and transistors, will allow in the near future to better choose memristive devices for specific applications. To try to obtain these attributes, a morphologic analysis of the voltage-current loops' area and length of several theoretical memristive devices models was made in MATLAB changing its internal characteristics, for arrays of frequency and amplitude values of the input signal. Afterwards, a memristor device emulator was built to corroborate the theoretical results obtained. To this end the voltage-current loops for several input values were measured and the calculation of the loops' areas and lengths was effectuated.

Index

List of figures

Glossary

ADC:	Analog-to-Digital Converter
BCM:	Boundary Condition-based Model
CAM:	Content-Addressable Memory
CMOS:	Complementary Metal-Oxide-Semiconductor
DAC:	Digital-to-Analog Converter
DC:	Direct Current
FPGA:	Field Programmable Gate Array
NMOS:	N-type Metal-Oxide-Semiconductor
ODE:	Ordinary Differential Equation
ReRAM:	Resistive Random-Access Memory
TEAM:	ThrEshold Adaptive Memristor Model

Chapter I

Introduction

The three basic two terminals elements in electronics are all known by the scientific community since the XIX century: the resistor, the capacitor and the inductor, being present in almost every schematic and electronic circuit. One of the characteristics shared by all these elements is the fact that their mathematic definition is the relation between two of the four fundamental variables: current (i), voltage (v), electric charge (q) and magnetic flux (φ).

Attempting to establish all the possible pair relations between these variables leads to well known equations: the definitions of resistance $R = dv/di$, capacitance $C = dq/dv$, inductance $L = d\varphi/di$, current $i = dq/dt$ and Faraday's law $v = d\varphi/dt$. As can be seen one relation is missing, between φ and q. This flaw was noticed by Leon Chua [1], who postulated for the sake of completeness the existence of a fourth two terminal element that would establish the missing relation. To this new electrical element Leon Chua called memristor ($M = d\varphi/dq$), because as he demonstrated, if the relationship between φ and q is nonlinear, it behaves like a variable resistor with memory. The various relations between the fundamental variables can be observed in Figure 1.1.

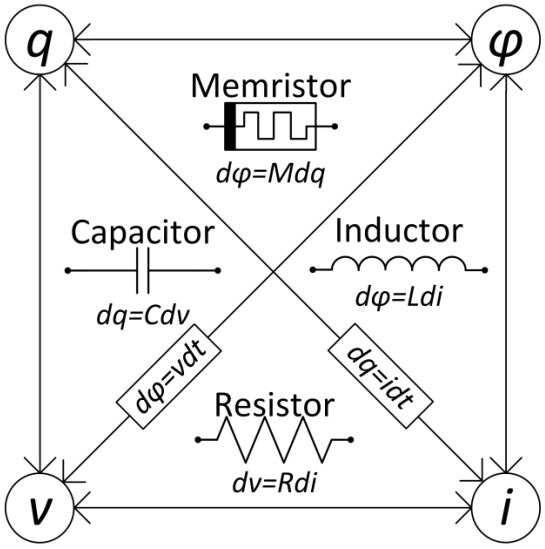


Figure 1.1: The various relations between v, i, q and φ.

1.1 - Motivation

The mathematical definition of the memristor by Leon Chua [1] and the announcement, by the HP Labs [2] in 2008 of the discovery of a device with memristive properties, sparked a renewed interest of the entire scientific community on the subject, specially the area of computers and electronics. The existence of a passive resistive two terminal element having memory capabilities, even when no energy is passing through it, opens a new world of possibilities. Foreseen applications such as high density memories, reconfigurable devices, sensors, neuromorphic circuits, amongst others, are currently under consideration. However, despite the fabulous planned applications and the eagerness of the community to see them fulfilled, there are no memristive systems yet considered ready for usage (although HP expects to launch devices in 2014 [3]). Moreover, the existing devices are not fully understood nor is their behaviour adequately modelled.

Nonetheless, there is continuous progress in this area and all indicators point to the question "when" and nor "if" will the memristor appear commercially. In the light of this reality, new questions are coming forward: "How do I use it?", "How do I place in the operation mode I require?" or "What are the physical limits of a specific memristor?". These are valid questions for any electronic element and as such, they have responses. For example, consulting a datasheet of a transistor reveals a wealth of data: operation modes, requirements for the transistor to achieve then, like v_{gs} or v_{ds} voltages and other parameters, like h_{FE} or r_{on}.

The same kind of specifications are required for the memristor. First, the available devices and mathematical models need to be analysed and studied, for figures of merit to be discovered. Then, these figures can be used as parameters to classify the memristor and quickly inform the engineers of its limitations, precautions of usage and suggestions of utilization. The intention of the work presented in this text is to lay the foundation for a study that may lead to the discovery of these parameters.

1.2 - Objectives

This dissertation has the main objective of studying the behaviour of the memristors and memristive devices models in literature for a multitude of different input signals, find consistencies in their response and point out figures of merit that can be tracked in these models, which can be used to classify or predict their behaviour. To do this the following task were proposed:

- Summarize the mathematical models of memristors and memristive devices existent in the literature;
- Choose one or more to study their behaviour to input signals of different frequencies and amplitudes, focusing in the study of the area and length or the v-i curves;
- Study the areas and lengths of the v-i curves for variations of the input signals and the devices' characteristics to find figures of merit;
- Summarize the emulators of memristive devices existent in literature;
- Choose one emulator and study the graphics of area and length of the v-i curves for different input signals to try to corroborate the results obtained from the analysis of the mathematical models.

1.3- Original work

During this work two papers were produced for IEEE (Institute of Electrical and Electronics Engineers) conferences. The first paper entitled "Frequency Characterization of Memristive Devices" was presented in ECCTD 2013 (European Conference on Circuit Theory and Design 2013), [4] and consists in the demonstration of a method of extracting figures of merit from the graphics of area and length *vs* the frequency of the input signal. The second paper,

entitled “Amplitude Characterization of Memristive Devices” was accepted and is, at the time of the writing of this document, scheduled to be presented in ICECS 2013 (International Conference on Electronics, Circuits, and Systems 2013), [5] and consists in the demonstration of the application of the same method to extract figures of merit, but with input signals of varying amplitude. In the annex section, at the end of the text are located copies of these articles.

1.4 - Structure of the Dissertation

The present dissertation is divided in six chapters.

- The second chapter lays out the state of the art, starting with a brief introductory historical background, a note on the distinctions between memristors and memristive devices, followed by the characteristics that define the memristor. Then explanations of its non-linearity and v-i curve are presented. Finally, it is listed the different types of memristive systems, based on their memristive mechanism and a summary of possible applications for the memristor both in the analog and digital fields;
- Chapter three comprehends a resume of the various mathematical models collected;
- Chapter four is dedicated to the study of the area and length of the v-i curves of the models studied, presenting the mathematical principles the calculations were based on, the equations, the basic structure of the MATLAB algorithm and the results obtained in this study;
- The fifth chapter covers the study and usage of the hardware emulator of a memristive system, starting by a search of some of the available systems, a brief explanation of the chosen circuit, passing to the explanation of the procedure of collecting and treating the data obtained and ending in an exposition of the results and comparison with the results from the MATLAB simulations;
- The last chapter presents the final conclusions of the work developed during the dissertation, a critical appreciation by the author and a layout of the work that is possible to develop in the future;

Chapter II

Analysis of the memristor's state of the art

The memristor, like all novel technologies or ideas passes through various stages of acceptance. First it was considered a mathematical curiosity then, an impossible and foolish concept. Afterwards it was considered plausible and worth some attention and finally, in the last decade, it has been considered a valid and worthy technology that will certainly come to market.

This chapter lays the basic history of the memristors journey from a mathematical theory to near ready technology. It also includes the distinguishing characteristics of the memristor defined by Leon Chua. Then some systems that have memristive behaviour are summarized. Finally, some of memristor's planned real world applications are listed.

2.1 - Historical background

The mathematical postulation of the memristor was introduced by Leon Chua in 1971 [1]. However, even before its postulation, materials and devices with memristive proprieties were discovered, with switching behaviour being reported in metal oxide films by T. Hickmott in 1964 [6], F. Argall in 1968 [7] and G. Dearnaley et al. in 1970 [8]. Despite these discoveries, the authors were never able to truly characterize the devices created, describing it as an anomalous switching behaviour or a negative resistance.

Later in 1976 [9] L. Chua and S. Kang expanded on the definition of memristor and announced a broader class of devices named memristive systems, in which the memristance could be dependant in more than one state variable, being the memristor a particular type of these systems. In this paper the authors also proposed an attempt to reclassify known devices and systems as one-port memristive systems, like the thermistor, some ionic systems, like the Hodgkin-Huxley axon circuit model and discharge tubes.

The following decades were slow in developments, however, just like before the mathematical postulation of the memristor, many studies were made of thin films of metal oxides and organic layers with variable resistance and voltage-current curves with hysteretic behaviour: [10]–[13] and many others. And just like before, the obtained results were many times classified as abnormal behaviours and the association with memristors or memristive systems was never made. These conclusions stemmed from the absent dialogue between the fields of science and electrical engineering.

After almost two decades of work, in 2008 a team from HP Labs announced in a paper published in the Nature journal [2], the first publicly known physical implementation of a device identified as a memristor and its mathematical model. After this announcement there has been a renewed interest around the memristor in the scientific community, sparking announcements of previously found memristive devices [14]; attempts to better model a memristor found in the HP Labs [15], [16] and others; creation of emulators using off-the-shelf components, [17]–[19] to try to study the memristor and validate possible circuit applications; and opinions of disbelieve and scepticism about the findings [20].

2.2 - Distinction between memristive systems and memristors

In 1976 [9] L. Chua and S. Kang realized that the memristor is a special case of a broader class of systems, which they named memristive systems. Considering the input signal *u(t)*, the output signal *y(t)* and the memristance given by the function *g* we have:

$$y(t) = g(x, u, t)u(t) \tag{2.1}$$

$$\dot{x} = f(x, u, t) \tag{2.2}$$

Where *x* is the vector of internal state variables of the memristive system, defined by:

$$x = (x_1, x_2, \dots, x_k), k \in \{1,2, \dots, n\} \tag{2.3}$$

Thus, a memristor is a memristive system in which the internal state variable is only the charge across the device, changing the equation (2.3) to:

$$x = q \tag{2.4}$$

As a last note, for now on, the term ideal memristor will be employed to refer to the charge-flux memristor defined in [1].

2.3 - What characterizes a memristor

According to L. Chua et al. [21] there are three fundamental characteristics that a device must have to be classified as a memristor. To these characteristics the authors called 'The Three Fingerprints of a Memristor'.

The first one is the presence of a 'pinched hysteresis loop' in the device's voltage-current characteristic when excited by a sinusoidal source (explained in more detail later in the next chapter). The 'pinched hysteresis loop' means that the v-i curve always presents two values of current or voltage (depending if the input signal is a voltage or current source respectively) for every value of input, except in one point, the origin of the axis, there $v = 0$ always corresponds to $i = 0$. This is the most distinguishable feature of a memristor, and one of the first to be noticed [9].

The second fingerprint relates to the area of the lobes of the v-i curve (part of the curve inside a single quadrant of the graph). According to [21], this area must decrease with the increase of input signal's frequency. A behaviour confirmed by the experiments (detailed in chapter 4).

The last fingerprint of the memristor is its behaviour with the continuous increase of the input signal's frequency. This fingerprint says that independently of the v-i curve's shape, when the frequency of the input signal tends to infinite, the v-i hysteresis loop collapses to a single valued function that passes through the origin, turning the memristor into a common resistance. Like the first fingerprint, this was one of the first characteristics to be noticed [9] and like the second, was observed in the experiments. A set of different v-i curves of the same device at different frequencies can be seen in Figure 2.1.

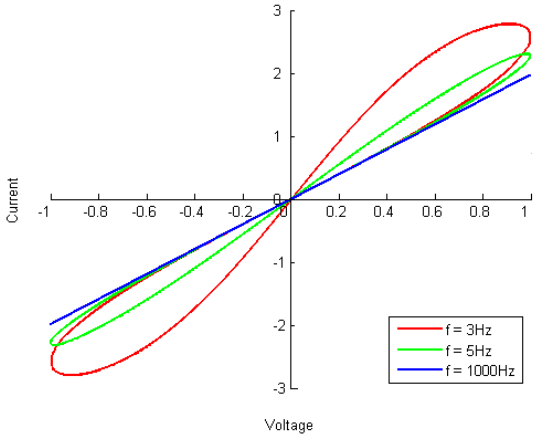


Figure 2.1: v-i curves of a memristor at different frequencies.

2.4 – The non-linearity of the ideal memristor

To introduce the memristor's equation, usually an axiomatic approach is used. This way however, the nonlinear characteristic is not easily noticeable, as such, bellow is presented a deduction of the memristor's equation where this characteristic is more easily noticeable [22]. In the following equations, to simplify the notation, the time dependence of $v(t)$, $\varphi(t)$, $i(t)$ and $q(t)$ will be omitted.

Considering that the flux φ and the charge q are related by a function f:

$$\varphi = f(q) \tag{2.5}$$

And deriving both terms of the equation with respect to the time t we have:

$$\frac{d\varphi}{dt} = \frac{df(q)}{dt} \Leftrightarrow \frac{d\varphi}{dt} = \frac{df}{dq}\frac{dq}{dt} \tag{2.6}$$

Remembering the equations of the current ($i = dq/dt$) and Faraday's law ($v = d\varphi/dt$) it is possible to change (2.6), becoming:

$$v = \frac{d\varphi}{dq} i \tag{2.7}$$

If the relationship between φ and q is linear, $\frac{d\varphi}{dq}$ is constant and it is called R, the common linear resistance. However, if the relationship between φ and q is not linear, $\frac{d\varphi}{dq}$ is called $M(q)$, memristance and the equation becomes:

$$v = M(q)i \tag{2.8}$$

From this equation it is possible to backtrack the steps given between equations (2.7) and (2.6), leading to the axiomatic definition already given, $d\varphi = Mdq$.

From the current ($i = dq/dt$) and Faraday's law ($v = d\varphi/dt$) equations, we have $q(t) = \int_{-\infty}^{t} i(\tau)d\tau$ and $\varphi(t) = \int_{-\infty}^{t} v(\tau)d\tau$ which allow us to rewrite equation (2.5) as:

$$\int_{-\infty}^{t} v(\tau)d\tau = f\left(\int_{-\infty}^{t} i(\tau)d\tau\right) \tag{2.9}$$

Thereby verifying the nonlinear relationship between the current and the voltage and the dependency of the current values with their past history.

2.5 – The voltage-current characteristic

To understand the memristor's voltage-current characteristic, we can start by rewriting (2.8):

$$v = M(q)i \Leftrightarrow v = M\left(\int_{-\infty}^{t} i(\tau)d\tau\right)i(t) \tag{2.10}$$

This implies that when $i(t) = 0, v(t) = 0$. Assuming that for $t \leq 0, i(t) = 0$, for a moment of time called *t'*, the voltage value *v* is:

$$v(t') = M\left(\int_{0}^{t'} i(\tau)d\tau\right)i(t') \tag{2.11}$$

Assuming that the current across the memristor is sinusoidal with period *T*, there is a moment in time $t'' = (T/2) - t',\ \forall t' \in [0, (T/2)]$, that verifies $i(t') = i(t'')$. However, for equation (2.10) we can take $v(t') \neq v(t'')$, because $M\left(\int_{0}^{t'} i(t)dt\right) \neq M\left(\int_{0}^{T/2-t'} i(t)dt\right)$. Thus, for any value of *i*, except $i = 0$, corresponds usually two values of *v*, provoking a hysteresis curve that is characteristic in these electrical elements, like the one in Figure 2.2.

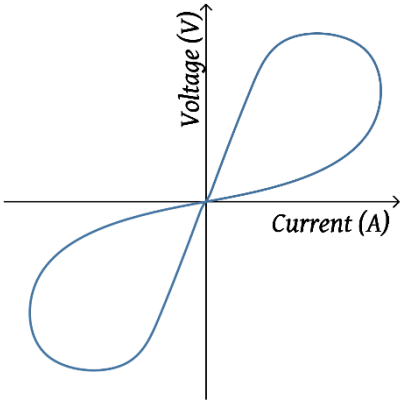


Figure 2.2: Typical v-i curve of a memristor.

2.6 - Types of memristive devices

There are many types of memristive devices and systems, each with a different physical behaviour that characterizes it. However they can be organized in five different groups, depending on the memristive mechanism that better fits them: thermal effect, chemical reaction, ionic transfer, spin polarization or phase transition.

2.4.1 - Thermistors and tungsten filament lamps

Thermistors have been identified as memristive systems since 1976, [9], later it was noticed that the tungsten filament lamps have the behaviour. These are thermal effect MDs, because the memristive proprieties of both stem from their self-heating. One particularity of these systems is that their voltage-current curves do not self-cross, as can be seen in Figure 2.3.

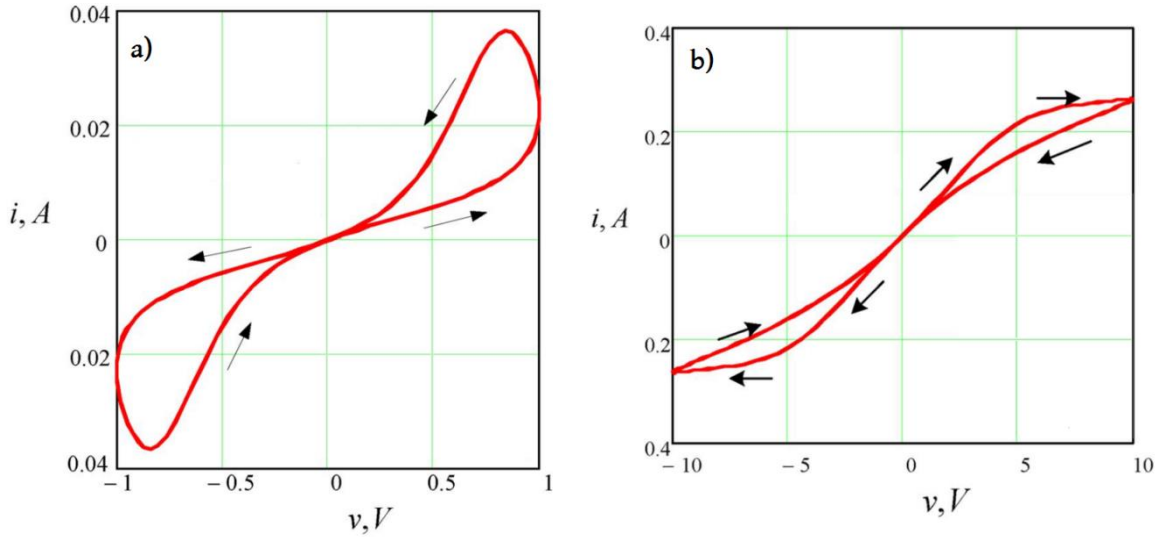


Figure 2.3: Voltage-current curves: a) of a thermistor, b) of a tungsten filament lamp, adapted from [21].

2.4.2 - Resistance switching thin films

The most common type of MD consists in a thin layer of memristive material sandwiched between two metallic electrodes (Figure 2.4). The memristive material can vary from oxides (TiO_2, SiO_2, CuO, NiO, CoO, Fe_2O_3, MoO, etc.), to sulphides (Cu_2S, Ag_2S), perovskite-type oxides, organics and others [23]. The mechanisms to switch resistance in these MDs are still not well known, being usually attributed to ionic transfer, chemical reactions or even thermal effects.

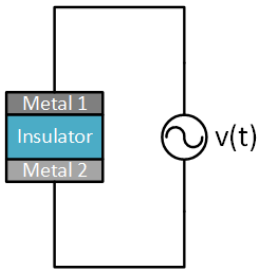


Figure 2.4: Thin film Memristor.

2.4.3 - Metal-insulator and Crystal-amorphous phase transition

Phase transition MDs are composed of materials that can change their phase with the variation of one or more environmental conditions, usually temperature. The most common examples are the MDs that change from metal to insulator and from crystal to amorphous states.

An example of a MD those material transitions between an insulator and a metal states was presented by T. Driscoll et al. in [24]. The MD is a thin film of VO_2 deposited on a sapphire substrate and then the temperature of the system is set to 340K, just below the temperature of phase transition of the VO_2. To programme the MD a series of high amplitude pulses is applied, increasing temporarily the temperature and as such, promoting the transition of a fraction of the VO_2 from insulator to metal, reducing the overall resistance across the device, as can be seen in Figure 2.5. This MD however suffers from a great flaw: because negative and positive pulses result both in an insulator to metal transition, the memristance of the device only moves in one direction.

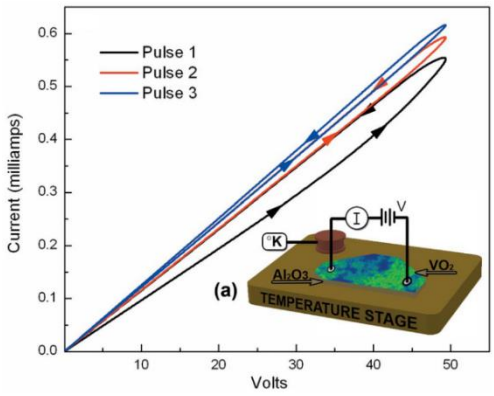


Figure 2.5: VO_2 MD and its v-i curve after three consecutive high voltage pulses, from [24].

MDs that rely in the transition from crystal to amorphous states rely in the Joule heating of a current flow to promote the switching. Some of the phase-change materials are $Ge_2Sb_2Te_5$, $AgSbSe_2$, Ag-In-Sb-Te and others [23].

2.4.4 - Spintronic systems

Spintronic devices are systems whose transport characteristic depends on the spin of the electrons. Unlike other systems, like the ones based on ionic transfer, spintronic systems are usually easier to control, allowing a fundamentally different degree of freedom [25]. There are two main kinds of spintronic MDs, semiconductor and metal based.

Semiconductor based spintronic MDs are usually composed of a junction between a semiconductor, where the electron spin is polarized and a half-metal (ideally ferromagnets with 100% spin polarization at the Fermi level), that acts as a spin filter, a scheme of such a system can be seen in Figure 2.6. These systems can function by spin-injection or spin-extraction, though the spin-extraction method is of greater interest. In this method, knowing that the half-metal only accepts electrons with one spin, when a current forces the electrons to travel from the semiconductor to the half-metal, the electrons with opposite spin accumulate in the junction. Because their passage is denied, if the current persists the accumulation of opposite spin electrons increases, hampering the passage of correctly polarized electrons and effectively increasing the resistance of the MD [25].

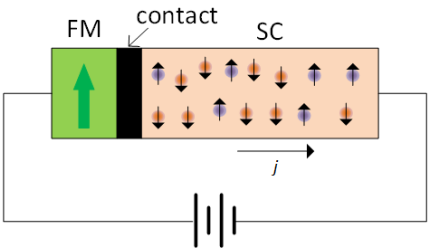


Figure 2.6: Scheme of a MD based on a semiconductor/half-metal spintronic system.

Metal spintronic systems are composed fully by metal, these systems can be subdivided in spin-torque transfer systems (Figure 2.7 a) and spin-torque-induced domain-wall motion (Figure 2.7 b) [23]. In the first one the resistance is given by the relative magnetization between opposite sides of a magnetic tunnel junction, the current flowing through the junction induces spin torque, which changes the relative magnetization. In the second one the resistance is dependent on the position of the barrier that divides the regions of opposite spin in the free layer, the position in the layer varies with the current that flows in the free and reference layers.

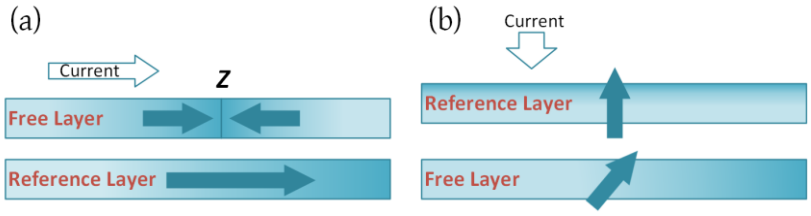


Figure 2.7: Scheme of MDs based on All-metal spintronic systems: (a) spin-torque transfer systems, (b) spin-torque-induced domain-wall motion. The blue arrows indicate the direction of spin polarization and Z the position of the barrier.

2.4.5 - Ionic channels

Perhaps the most curious and important MDs of all are the ionic channels, specially the potassium ion-channels and sodium ion-channels of the axons of the brain's neurons. The classification as MDs of these channels was first made by L. Chua and S. Kang in [9], when they analysed the Hodgkin-Huxley circuit model under the light of the memristor theory. The comprehension of these channels can be of the uttermost importance to the understanding of the neuron's operation and in turn the animal brain.

2.5 - Applications of memristors

Due to its unusual memory characteristic for a passive two terminal circuit element, the memristor is capable of replacing not only a simple resistor, but other circuit elements, like diodes and transistors, performing functions that before required a more or less complex circuit that could be active. The applications of the memristor presented here are just a sliver of what is proposed today and that even is just a small part of what it can do, as the interest of the scientific community increases and real devices eventually become available, examples of its applications will certainly grow. The list is organized in two areas: digital applications, where the memristor is expected to switching between two values of memristance; and analog applications, where more values of the memristor's memristance range are used.

2.5.1 - Digital applications

2.5.1.1 - Memory

The most popular use of the memristor is as a non-volatile memory, switching the device between its high memristance and low memristance values to store ones and zeros. The reasons for its popularity as a non-volatile memory come from its expected high storage density and its passivity, not requiring energy to store the information. Tests made with prototype devices can hold their information for times greater than ten years, as reported in [23]. A memristor based memory can be arranged in a crossbar fashion, where the electrodes are the lines and the memristors are located in the cross points (like in Figure 2.8 a). This crossbar arrangement was made by the team at HP Labs, later [26] members of the same team proposed a way to stack multiple crossbars and a CMOS addressing layer (Figure 2.8 b).

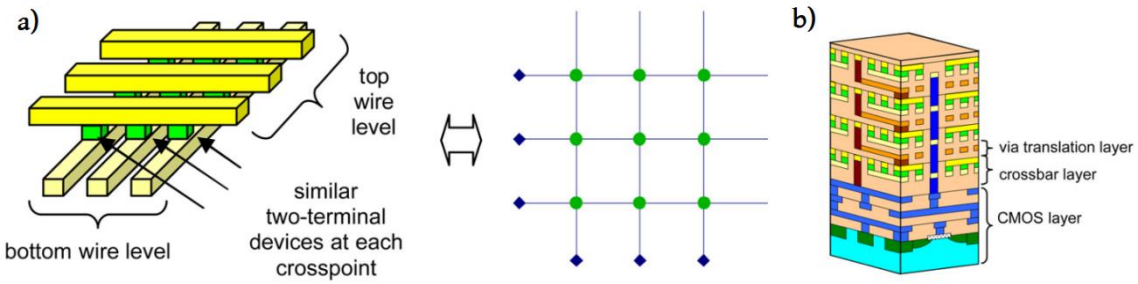


Figure 2.8: Crossbar arrangement of memristor memory: a) a simple layer, b) multiple layers with CMOS address layer, from [26].

Memristance based memory has been proposed to replace both volatile (in CAM architecture, for very high speed search applications [27]) and non-volatile memories, like ReRAM [28].

2.5.1.2 - Logic

In was proved that it is possible to perform logic operations with memristors in [23], [29]. Memristors were proved useful as well in FPGAs [30], [31], being used on the top of the CMOS

logic to replace the connections between the cell, greatly increasing the density of the logic in a FPGA.

2.5.2 - Analog applications

2.5.2.1 - Programmable analog systems

The memristor can be used as a variable resistor in analog circuits, replacing a linear resistor, allowing the circuit response to be configured during operation. Some examples of this include programmable gain amplifiers [32] (Figure 2.9), Schmitt triggers with programmable thresholds [33] (Figure 2.10), adaptive filters [34] and many others.

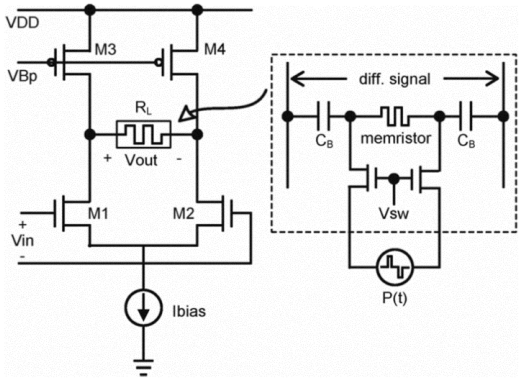


Figure 2.9: Gain amplifier with a memristor. The transistors in the details to the right are used to program the memristance value, from [32].

In the case of the programmable gain amplifier of Figure 2.9, the memristor works a programmable load resistor, changing the gain of the circuit by altering the memristance of the memristor.

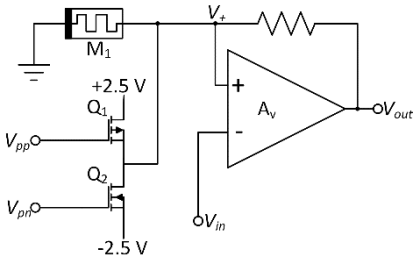


Figure 2.10: Schmitt trigger with a memristor. Transistors Q_1 and Q_2 are used to program the memristance value.

The programmable Schmitt trigger of Figure 2.10 can be configured by applying short DC bursts to the memristor, increasing its memristance. The increase of memristance will in its turn increase the absolute values of the circuit's threshold voltages, which will decrease the frequency of the trigger's output. An example of this behavior is represented in Figure 2.11.

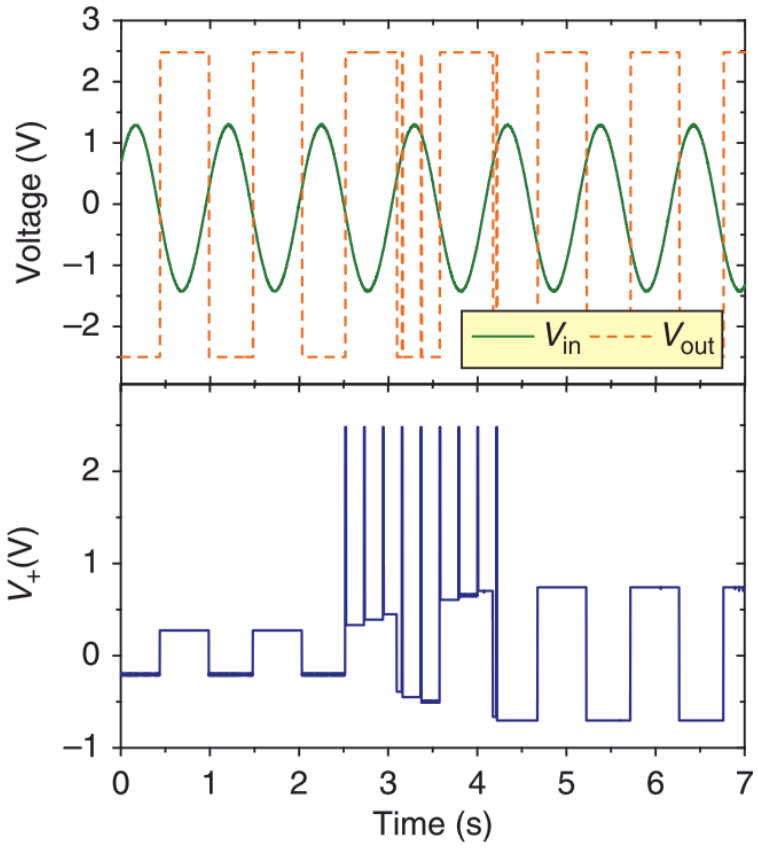


Figure 2.11: Voltages at various points of the circuit from Figure 2.1010 as the memristance is increased, from [33].

2.5.2.2 - Neuromorphic circuits

Neuromorphic circuits are probably the most interesting and radical application of the memristors. It is already known that the ionic channels of the neuron's axons are memristive systems, but more recently, associations have been made between memristance and the plasticity of synapses. These associations have been demonstrated in neuromorphic systems that use memristors as synapses and neurons made in CMOS technology, [35], [36]. These results prove that memristors can be used to create circuits that adapt and learn like the animal brain. A more simple and academic example was presented by Pershin and Di Ventra in [17]. In this paper the authors replicated Pavlov's experiment with dogs, using emulators for the neurons and the memristive synapses. As it can be seen in Figure 2.12, initially the output signal of the third neuron (salivation) is only triggered by the signal of one neuron (sight of food), but after a learning period, the output signal can be triggered by both the first neuron and the second (sound).

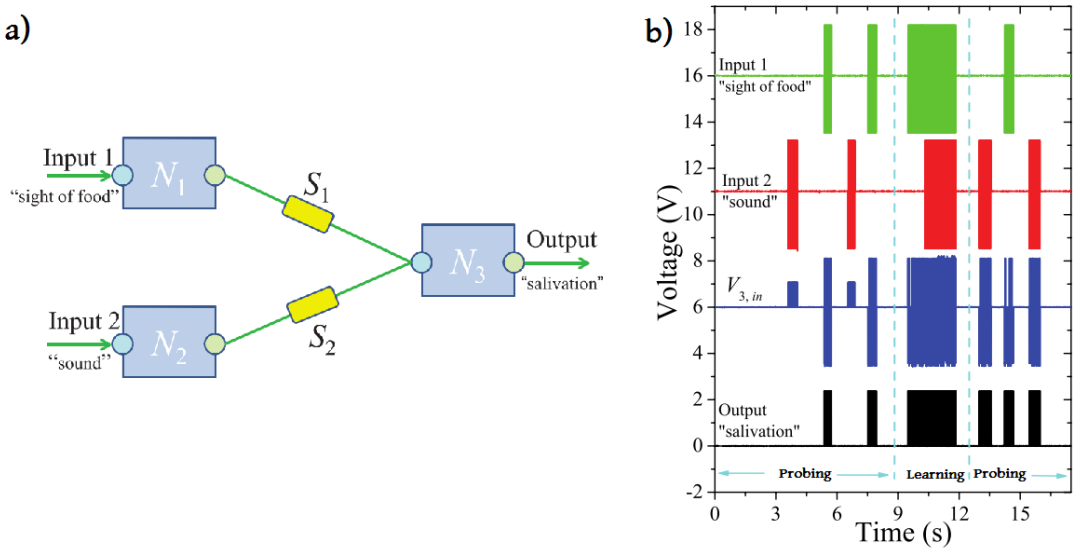


Figure 2.12: Emulation of Pavlov's experiment with memristors: a) scheme of the system, b) signals of voltage in the system *vs* time, adapted from [17].

2.6 – Final remarks

It is clear that the acceptance of the memristor as a valid circuit element was a harsh one. The reasons for this harshness ranged from the complexity of its mathematical definition (at least compared with the other basic two terminals elements), to the lack of dialogue between the various scientific areas. However, as was clearly shown, there are many systems that have memristive behaviour, some of them being used for many years without knowing of its intrinsic memristance.

The recognition of the memristor as a real circuit element triggered the curiosity of the scientific community, resulting in the proliferation of designs based upon its peculiar characteristics. This applications range from simple improvements to current systems to new systems, unfeasible with the current technology. This proves that the memristor is a stepping stone to the advancement of our technology.

Chapter III

Mathematical models

The mathematical models are used to describe the relation between the voltage across with the current through the MD. The need for these models emerged with the first TiO_2 based MD identified.

Further mathematical equations were proposed through time. Some, called window functions complemented the first model proposed, trying to correct some of its insufficiencies. Others were brand new models that either tried to better model the TiO_2 MD or other types of MDs.

3.1 - The memristive devices models

In this section some models are explained, details of their behaviours and mathematical equations. Most of these models address exclusively the Titanium Oxide MD, due to the fact that it is the model that has been attracting mostly the attention of the research community.

3.1.1 - The Strukov and Williams model

The Strukov and Williams model (or just Strukov) was made by the HP team that created the first TiO_2 MD that was announced and detailed in many publications, including [2]. It is one of the most used models in the academic and scientific circles, due to its simplicity and because it was the model of the very first claimed memristive device.

This MD consists in two superimposed layers of TiO_2, one pure, acting as an insulator and another with oxygen vacancies (TiO_{2-x}), acting a conductor, and two platinum electrodes, each

one making contact with a different layer of TiO_2. To change the memristance of the device, a voltage is applied. A positive voltage forces the oxygen vacancies to move into the insulating TiO_2 layer, decreasing the resistance across the device, while a negative voltage pulls together the oxygen vacancies, reducing the size of the TiO_{2-x} layer and increasing the resistance across the device [37], a graphical representation of these behaviours can be seen in Figure 3.1.

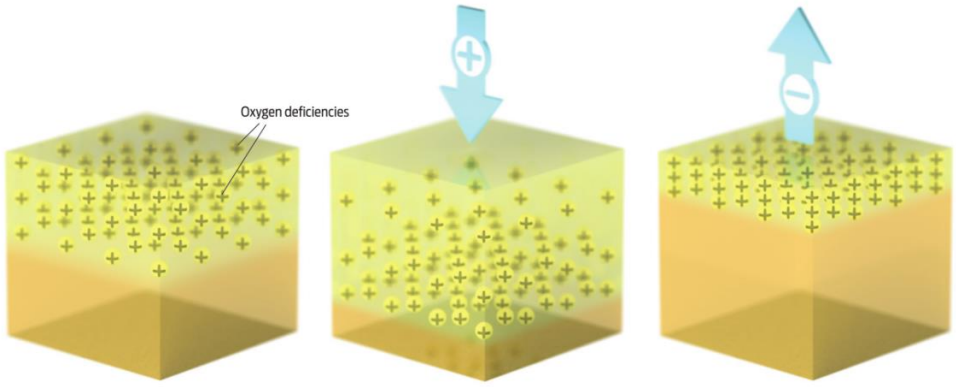


Figure 3.1: Representation of the movement of oxigen vacancies in the HP's TiO_2 memristor, from [37].

The Strukov model considers the memristor as two variable resistors in series. Each resistor represents a state of the memristor, R_{on} represents the TiO_2 fully doped with oxygen vacancies (the state of lowest resistance) and R_{off} represents the TiO_2 fully undoped (the state of highest resistance). Like in the explanation above, the increase of one layer in size results in the decrease in size of the other layer, in this model the resistors vary oppositely, but in sync: when one increases its value, the other decreases in the same proportion and vice versa. This proportion is defined by the length of the doped layer (also called the position of the interface between layers) divided by the total length of the device. Figure 3.2 depicts the model's constitution.

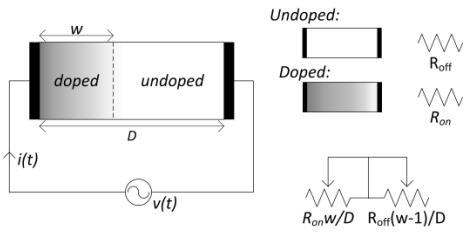


Figure 3.2: Representation of the Strukov and Williams model for the TiO_2 memristor.

This model assumes the application of an external bias *v(t)* and a simple ohmic electronic conduction with a linear ionic drift in a uniform field with average ion mobility μ_V [2]. To obtain the equation of the memristance we begin by changing equation (2.8) to:

$$v(t) = \left[R_{on}\frac{w(t)}{D} + R_{off}\left(1 - \frac{w(t)}{D}\right)\right] i(t) \tag{3.1}$$

Where D is the length of the memristor and w is the position of the interface and the vector of state variables. It is defined that *w(t)* changes with time according to the function:

$$\frac{dw(t)}{dt} = \eta\mu_V \frac{R_{on}}{D} i(t) \tag{3.2}$$

In this equation η is the polarity of the memristor, that is, which way the dopants drift when a positive voltage is applied to the device and it depends on the orientation of the device in the circuit. η can assume the values ±1, but for the sake of simplicity for now on, unless specified, η is supressed and assumes the value +1.

To obtain *w(t)* we can integrate both sides of (3.2), with the result:

$$w(t) = \mu_V \frac{R_{on}}{D} q(t) \tag{3.3}$$

with *q(t)* being the charge that passes through the memristor. Replacing *w(t)* in (3.1) with (3.3), simplifying and knowing that $M(q) = v(t)/i(t)$, we get:

$$M(q) = R_{on}\frac{\mu_V R_{on}}{D^2} q(t) + R_{off}\left[1 - \frac{\mu_V R_{on}}{D^2} q(t)\right] \tag{3.4}$$

Assuming that $R_{off} \gg R_{on}$, we finally have:

$$M(q) = R_{off}\left(1 - \frac{\mu_V R_{on}}{D^2} q(t)\right) \tag{3.5}$$

Although simple this model is far from perfect. Because of its linearity, the model does not take in account the non-linearity of the drift of dopants. To attempt to take that non-linearity in account, a window function is used in equation (3.2), changing it to:

$$\frac{dw(t)}{dt} = \eta\mu_V \frac{R_{on}}{D} i(t) F(x) \tag{3.6}$$

Where $x = w/D$. There are many window functions, each with different properties and characteristics, a few will be detailed in the section ahead.

3.1.2 - The Pickett model

This model was published by Pickett et al. from HP Labs [38], [39] and it models the same type of TiO_2 memristive device as the Strukov and Williams model. While the previous model was created for easy readability and understanding for the scientific community, this one was made to model more closely the device created, disregarding simplicity and readability of the used equations, but being more accurate by taking into account nonlinear dopant drift, asymmetric behaviour and contributions of the ionic diffusion.

A representation of the model can be seen in Figure 3.3 a). It consists in two main elements: a fixed value resistance (R_S, with a few hundreds of ohms) that results from electroforming the TiO_2, to cause the oxygen vacancies to create a conducting channel and a tunnel gap in the remaining TiO_2 between the conducting channel and the top electrode. This tunnel gap is modelled by a Simmons tunnel barrier model.

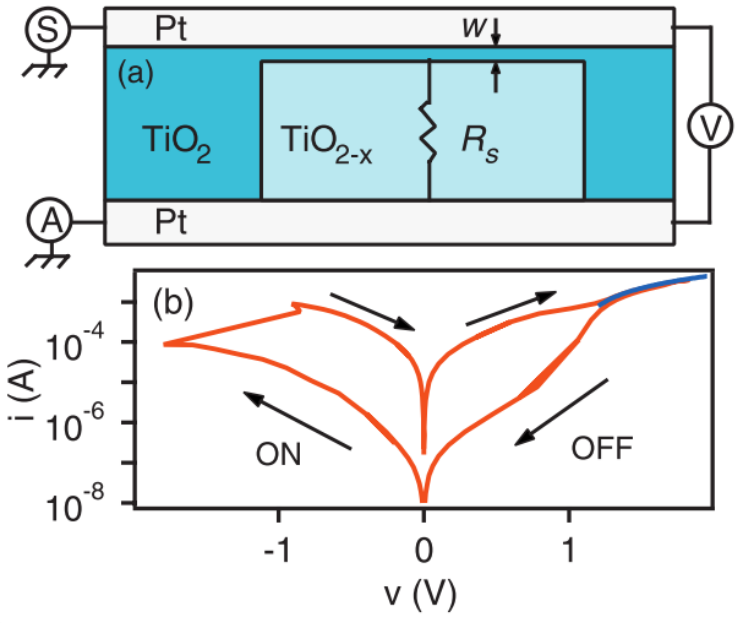


Figure 3.3: Pickett model: a) representation of the model, b) v-i curve from the physical device, [38].

The switching effect in this model is attributed to the variation of the width of the Simmons tunnel, *w*, being then used as the state variable (although in literature it shares the same name, *w*, with the state variable of the Strukov model, it does not have the same meaning).

The v-i relationship of the device is given by the equation,

$$i = \frac{j_0 A}{\Delta(w^2)}\{\varphi_1 \exp(-B\sqrt{\varphi_1}) - (\varphi_1 + ev_g)\exp(-B\sqrt{\varphi_1 + ev_g})\} \tag{3.7}$$

The voltage v_g across the tunnelling barrier is,

$$v_g = v - v_R = v - iR_S \tag{3.8}$$

Where v is the voltage across the device and v_R is the voltage across the internal channel series resistance. The terms of the Simmons tunnelling model are,

$$j_0 = e/(2\pi h) \tag{3.9}$$

$$\varphi_1 = \varphi_0 - ev_g\left(\frac{w_1 + w_2}{2w}\right) - \left(\frac{1.15\lambda w}{\Delta w}\right)\ln\left[\frac{w_2(w - w_1)}{w_1(w - w_2)}\right] \tag{3.10}$$

$$B = 4\pi\Delta\mathrm{w}\sqrt{2m}/h \tag{3.11}$$

$$\Delta\mathrm{w} = w_1 - w_2 \tag{3.12}$$

$$w_1 = \frac{1.2\lambda w}{\varphi_0} \tag{3.13}$$

$$w_2 = w_1 + w\left[1 - \frac{9.2\lambda}{(3\varphi_0 + 4\lambda - 2ev_g)}\right] \tag{3.14}$$

$$\lambda = e^2 \ln 2/8\pi k\varepsilon_0 w \tag{3.15}$$

where w is the state variable, k is the dielectric constant, φ_0 is the rectangular barrier height, R_s is the channel series resistance, A is the cross-section area of the tunnelling gap, e is the electron charge, h is the Planck constant and ε_0 is the vacuum permittivity. Finally, the equations of the state variable change with time are,

For $i > 0$:

$$\frac{dw}{dt} = f_{off}\sinh\left(\frac{i}{i_{off}}\right)exp\left(-exp\left(\frac{w - a_{off}}{w_c} - \frac{|i|}{b}\right) - \frac{w}{w_c}\right) \tag{3.16}$$

For $i < 0$:

$$\frac{dw}{dt} = f_{on}\sinh\left(\frac{i}{i_{on}}\right)exp\left(-exp\left(-\frac{w - a_{on}}{w_c} - \frac{|i|}{b}\right) - \frac{w}{w_c}\right) \tag{3.17}$$

With i being the current through the device and f_{off}, i_{off}, a_{off}, w_c, b, f_{on}, i_{on} and a_{on} being fitting parameters. A v-i curve measured from the device modelled by these equations can be seen in Figure 3.3 b).

3.1.3 - The Yakopcic model

Unlike the other models that focus only in the TiO_2 memristive device from HP, this one is intended to model various other devices, [40]. The v-i relationship is given by the following equation:

$$i(t) = \begin{cases} a_1 x(t) \sinh\big(bv(t)\big), & v(t) \geq 0 \\ a_2 x(t) \sinh(bv(t)), & v(t) < 0 \end{cases} \tag{3.18}$$

Because some devices have different conductivity depending on the bias of the input voltage, two different fitting parameters, a_1 and a_2 are used to model the amplitude of the conductivity, depending on the polarity of the input and b is used to control the intensity of the threshold function that relates conductivity with input voltage. The state variable, x, ranges between zero and one and has a different physical interpretation depending on the memristive device that is being modelled.

The change in the state variable depends on two functions, $g\big(v(t)\big)$ and $f\big(x(t)\big)$. The first one imposes a programming threshold on the model:

$$g\big(v(t)\big) = \begin{cases} A_p\big(e^{v(t)} - e^{V_p}\big), & v(t) > V_p \\ -A_n\big(e^{-v(t)} - e^{V_n}\big), & v(t) < -V_n \\ 0, & -V_n \leq v(t) \leq V_p \end{cases} \tag{3.19}$$

Where V_p, V_n are the threshold voltages and A_p, A_n are the magnitudes of the exponentials, both positive and negative respectively. The second function models the state variable, allowing the change of the variable to be dependent on the polarity of the input and is given by the equations:

$$f(x) = \begin{cases} e^{-\alpha_p (x - x_p)} w_p\big(x, x_p\big), & x \geq x_p \\ 1, & x < x_p \end{cases} \tag{3.20}$$

$$f(x) = \begin{cases} e^{\alpha_n (x - x_n - 1)} w_n(x, x_n), & x \leq 1 - x_n \\ 1, & x > 1 - x_n \end{cases} \tag{3.21}$$

The fitting parameters x_p and x_n represent the point where the change of state variable starts to suffer limitation (up to those point the model assumes constant change) and the fitting parameters α_p and α_n are the exponentials' rate of decay. The functions $w_p(x, x_p)$ and $w_n(x, x_n)$ are window functions that enforce $f(x) = 0$ for $x(t) = 1$ and $v(t) > 0$ and $f(x) = 0$ for $x(t) = 0$ and $v(t) < 0$ and are represented by:

$$w_p\big(x, x_p\big) = \frac{x_p - x}{1 - x_p} + 1 \tag{3.22}$$

$$w_n(x, x_n) = \frac{x}{1 - x_n} \tag{3.23}$$

Finally, combining $g\big(v(t)\big)$ and $f\big(x(t)\big)$ gives the equation of the change of the state variable with time:

$$\frac{dx}{dt} = g\big(v(t)\big) f\big(x(t)\big) \tag{3.24}$$

This model is simpler than the Pickett model, however due to the extensive use of exponentials and hyperbolic sine functions, it still is computationally intensive.

3.1.4 - The TEAM model

Like the Yakopcic model, the TEAM is a generic model, but unlike Yakopcic it is more accurate and less computationally intensive, by relying more on polynomials, [41]. The voltage-current relationship of this model is undefined and can be chosen from the relationships of other model, for example, if this model is to mimic the Strukov model, the v-i can be:

$$v(t) = \left[R_{on} + \frac{R_{off} - R_{on}}{x_{off} - x_{on}}(x - x_{on})\right] i(t) \tag{3.25}$$

While if the memristive device described by the Pickett model is to be modelled by TEAM, the v-i relationship can become:

$$v(t) = R_{on} e^{(\lambda / x_{off} - x_{on})(x - x_{on})} i(t) \tag{3.26}$$

Where λ is a fitting parameter and R_{on} and R_{off} are the terminal resistance values of the memristive device and are related by:

$$\frac{R_{off}}{R_{on}} = e^{\lambda} \tag{3.27}$$

Independently of the v-i relationship used, the derivative of the state variable with time is given by the equation:

$$\frac{dx(t)}{dt} = \begin{cases} k_{off}\left(\dfrac{i(t)}{i_{off}} - 1\right)^{\alpha_{off}} f_{off}(x), & 0 < i_{off} < i \\ 0, & i_{on} < i < i_{off} \\ k_{on}\left(\dfrac{i(t)}{i_{on}} - 1\right)^{\alpha_{on}} f_{on}(x), & i < i_{on} < 0 \end{cases} \tag{3.28}$$

Where k_{off}, k_{on}, α_{off} and α_{on} are fitting parameters, i_{off} and i_{on} are the threshold currents and x is the internal state variable, whose physical interpretation depends on the memristive device being modelled and v-i relationship used. The function $f_{off}(x)$ and $f_{on}(x)$ represent the dependence on the state variable and are fitting functions similar in behaviour to the window functions used by the Strukov model:

$$f_{off}(x) = exp\left[-exp\left(\frac{x - a_{off}}{w_c}\right)\right] \tag{3.29}$$

$$f_{on}(x) = exp\left[-exp\left(-\frac{x - a_{on}}{w_c}\right)\right] \tag{3.30}$$

Where a_{off} and a_{on} can be considered approximations for the terminal values of the internal state variable x: x_{off} and x_{on} respectively, although unlike the other model, x can exceed these values with no critical consequences.

3.2 - Window functions

The drift of the dopants across the device is not linear and can depend on many factors, however some mathematical models, like the Strukov model do not take this into account, having a behaviour too different from the memristive device that they are trying to capture. To take this non-linearity of the dopant drift into account, a window function is used, being usually applied to the equation of the interface position's derivative, $dw(t)/dt$. Some window functions are only dependent on generic numeric values and the relative position of the interface (w), to the total length of the device (D), a variable usually written as x, while other function are dependent on values, like threshold voltages, that are usually, making the drift of the dopants more dependent on the characteristics of the input voltage.

3.2.1 - HP window function

This was the first window function to be proposed and the most simple. It was first announced in the paper that first explained the HP memristor and the Strukov and Williams model, [2]. It consists of a simple quadratic equation that is only dependent on the position w of the interface:

$$F(w) = w\frac{(D - w)}{D^2} \tag{3.31}$$

After applying the normalization $x = w/D$ it simplifies to:

$$F(x) = x - x^2 \tag{3.32}$$

This window, cross the full range of x can be seen in Figure 3.4.

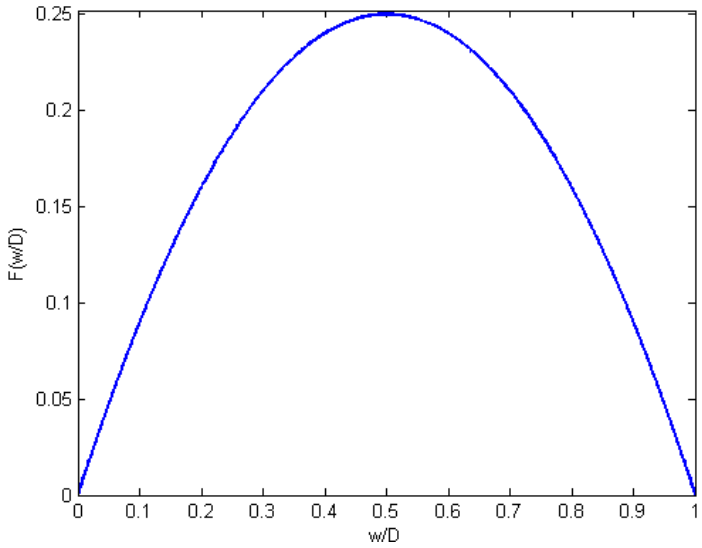


Figure 3.4: HP window function.

Observing this figure, it is noticeable that the drift of dopants is increasingly nonlinear the closed interface is to its limits, when they are reached, the drift is null. Another observation is that the function does not reach the value 1, which means that the drift is never linear. This is not a desirable behaviour and is addressed by other functions.

3.2.2 - Prodromakis window function

The Prodromakis window function, [15], is more scalable, besides using the relative position of the interface (x), it uses two fitting parameters: p and k, as can be see:

$$F(x) = k(1 - [(x - 0.5)^2 + 0.75]^p) \tag{3.33}$$

The control parameters allow a greater flexibility for the mathematical model to adapt to the memristive device. The parameter p can assume any positive real number and is used to choose the non-linearity of the model and the range of values of the window function. An example of

this can be seen in Figure 3.5, where the value of p is changed for a fixed $k = 1$. The k parameter can be used to fine-tune the change of $F(x)$, if desired. k can also assume any positive real value. In Figure 3.6 is the result of the window function for a set of values of k with $p = 10$.

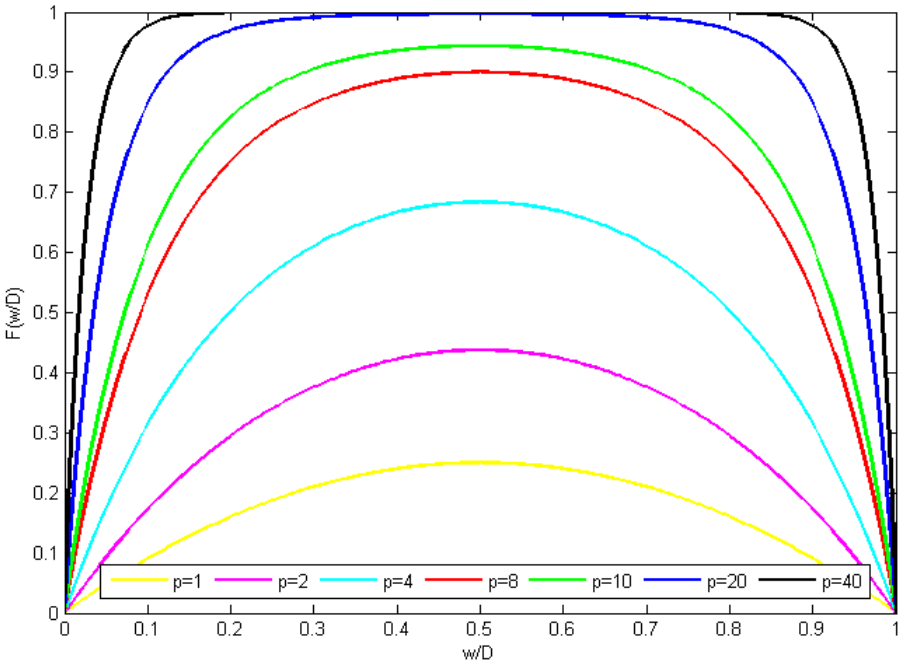


Figure 3.5: Prodromakis window function with *p* variable.

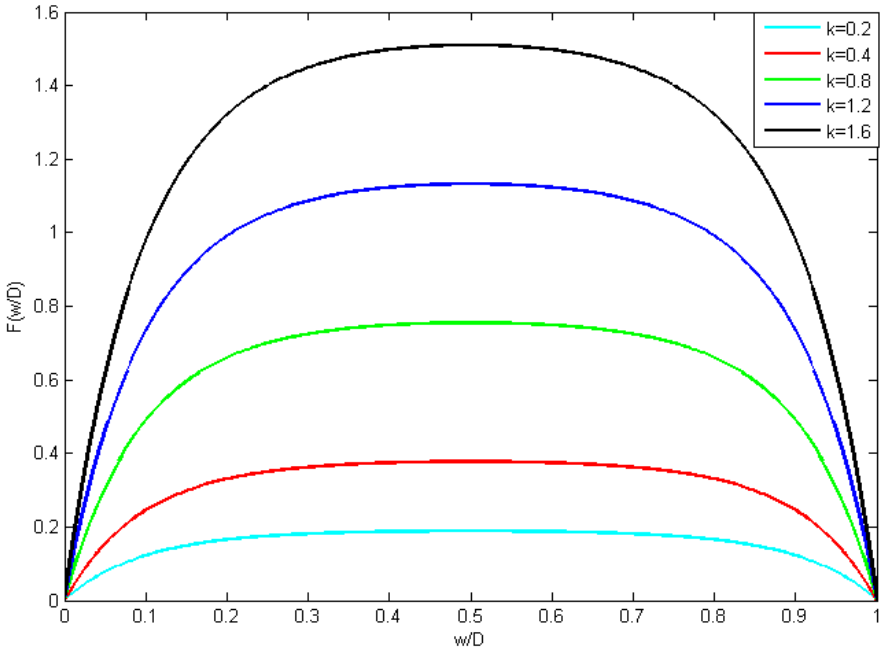


Figure 3.6: Prodromakis window function with *k* variable.

Like the previous function, the Prodromakis function guaranties a null drift in the boundaries of the device and the non-linearity of the drift increases as the interface approaches the said boundaries. However unlike the HP function, this one allows for the drift to be linear in the centre of the device, if that is desired.

3.2.3 - Joglekar window function

The window function of Joglekar and Wolf is one of the most simple and amongst the firsts to be announced in literature [16]. This function has the input variable x, but only one fitting parameter: p. It was created to ensure that the drift was null at the boundaries and that in the majority of the device's length the drift is close to, or linear. The function is the following:

$$F(x) = 1 - (2x - 1)^{2p} \tag{3.34}$$

Two things are of notice first, p can only assume positive integer numbers, second, as can be seen in Figure 3.7, no matter the value of p, there is always a value of x that makes $F(x)$ assume the value 1, which means that the linear drift away from the boundaries of the device is guaranteed, as the authors intended.

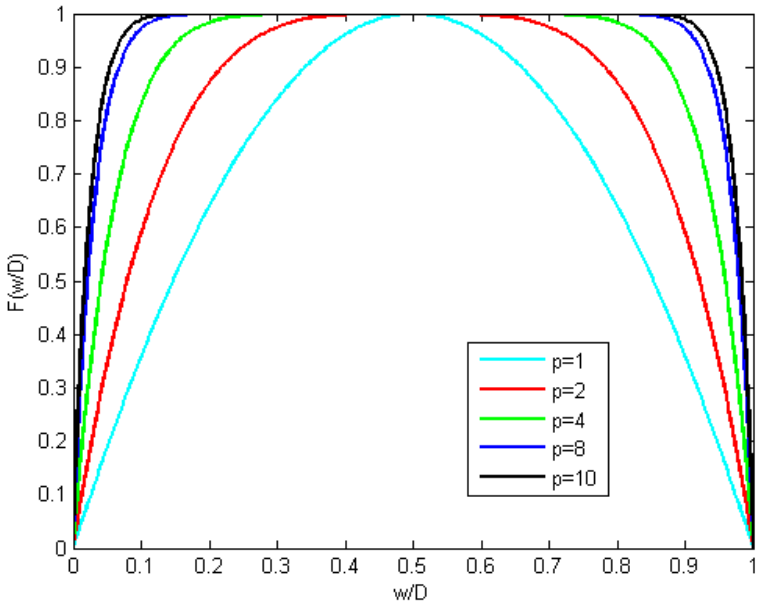


Figure 3.7: Joglekar window function.

According to Biolek et al. [42], this model has one great flaw, the terminal state problem: if the interface reaches one of the limits ($x = 0$ or $x = 1$), no external stimulus can change the memristive device to another state.

3.2.4 - Biolek window function

The Biolek window function was proposed by Biolek et al. in [42] to correct the flaws of the Joglekar window function, by introducing the current through the device as a new fitting parameter,

$$F(x) = 1 - \left(x - stp(-i)\right)^{2p} \tag{3.35}$$

Where p is a positive integer, i is the current through the memristive device and $stp(-i)$ is a function similar to the Heaviside step, except for $i = 0$:

$$stp(i) = \begin{cases} 1, & i \geq 0 \\ 0, & i < 0 \end{cases} \tag{3.36}$$

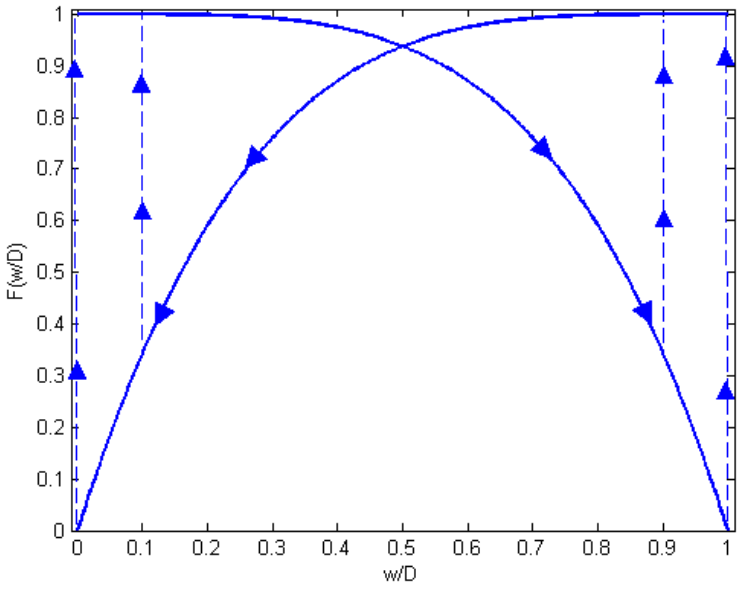


Figure 3.8: Biolek window function with *p* = 2;

Analysing the graphic of the window function in Figure 3.8 two characteristics can be extracted. First, the drift at the boundaries is null, just like in all the previous functions. Second, the curve of the drift of the interface approaching a boundary is different from the curve of the interface distancing from the same boundary. This is caused by the usage of the current as a

fitting parameter and allows the device modelled to retain the memory effect even when the interface reaches the boundaries, [42].

3.2.5 - Boundary condition window function

The boundary condition window function, or boundary condition-based model (BCM) was proposed by Corinto and Ascoli in [43]. This model is capable of having both a single-valued and a multi-valued memductance-flux characteristic. The function is unitary through all the possible positions of the interface, except in the boundaries of the device, where it can be null, depending on boolean conditions, which in turn depend on two fitting variables: threshold voltage 0 (v_{th0}) and threshold voltage 1 (v_{th1}). There are three conditions:

$$C_1 = \{x(t) \in]0,1[\vee (x(t) = 0 \wedge \eta v(t) > v_{th0}) \vee (x(t) = 1 \wedge \eta v(t) < -v_{th1})\} \tag{3.37}$$

$$C_2 = \{x(t) = 0 \wedge \eta v(t) \leq v_{th0}\} \tag{3.38}$$

$$C_3 = \{x(t) = 1 \wedge \eta v(t) \geq -v_{th1}\} \tag{3.39}$$

Where η is the polarity of the memristive device, $v(t)$ is the voltage across and $x(t)$ the relative position of the interface at the time *t*. These conditions affect the value of the window function according to the following equation:

$$F(x, \eta v) = \begin{cases} 1, & C_1 \; is \; True \\ 0, & C_2 \; or \; C_3 \; are \; True \end{cases} \tag{3.40}$$

3.3 – Final remarks

There are many mathematical models that attempt to characterize MDs, with some being over-simplistic and others far more detailed. Although the detailed models can characterize much better the memristive devices, they are not without problems. The most obvious of their issues is the extensive computation power required to obtain a solution, but other problem, no less important is the fact that such models usually are only usable for a single MD.

The more simplistic models might not be very rigorous, but they have some advantages. First of all, many of them can the adapted no various MDs, increasing their usefulness. Other positive characteristic is their lower requirements of computing power, despite the fairly good results of some of them. Another advantage, although not as important for the scientific

community is nonetheless relevant. Due to the fact that some of these simpler models have mathematical equations that are easier to understand, they can be a good introductory tool for students to the memristor area.

Chapter IV

Area and length of the v-i loop and MATLAB results

Currently there is no way to classify memristive devices. Most of the times, comparisons are made between different mathematical models, to check their adaptability to different configurations, like in [44]. Or at most, the behaviour of a single model is tested for various frequencies of the input signal, mainly the check the model's capacity of maintaining a hysteretic v-i loop, [32].

In the beginning of this work a mathematical model was exited with multiple input signals with different frequencies. One of the observations taken was that besides the reported decrease of the v-i loop hysteresis with the increase of frequency, this decrease was gradual and smooth. In the hopes of finding a behaviour that could help identify the frequencies that the model keeps a good sized hysteretic loop, the area of the v-i loop was calculated for a number of frequencies and the results were plotted in a two dimensional graph. The results of this graph were promising because, not only the area had a good behaved progression, that could be divided in distinct segments, but changes to the model's parameters led to a graph with different values, but similar behaviour. Later, the length of the loop was also computed to try to find other behaviours parallel with the changes in shape of the v-i loop.

This chapter details the process needed to perform the calculations, identifies the figures of merit that resulted from the analysis of the area and length graphs and presents the results for the MD models analysed in MATLAB.

4.1 – Evaluation of the area inside the v-i loop

The calculation of the v-i loop area is based on Green's Theorem:

$$\iint_A \left(\frac{\partial F}{\partial x} - \frac{\partial G}{\partial y}\right) \cdot ds = \oint_C G \cdot dx + F \cdot dy \tag{4.1}$$

Where G and F are functions of (x, y), A is the region surrounded by the loop C, and ds is an infinitesimal part of A ($ds = dx \cdot dy$), these elements are represented in Figure 4.1.

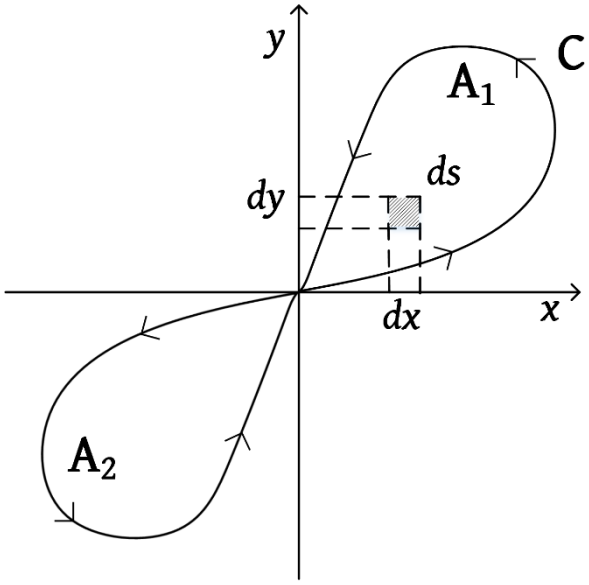


Figure 4.1: Hysteresis loop for area calculation.

For this theorem to be applicable to the area of A, some conditions need to be satisfied: The loop needs to be closed, piecewise smooth, with positive orientation and simple, which means that it cannot cross itself. This last condition is not respected, but can be solved by splitting the area in two sections, named A_1 and A_2, changing (4.1) to.

$$\iint_{A_k} \left(\frac{\partial F}{\partial x} - \frac{\partial G}{\partial y}\right) \cdot ds = \oint_{C_k} G \cdot dx + F \cdot dy \tag{4.2}$$

where $k \in \{1,2\}$ and C_k is the portion of C that encloses the respective A_k. To determine the area, F and G need to be defined to make the integrand of the left side of (4.2) unitary. For that the equations $F(x, y) = 0$ and $G(x, y) = -y$ are chosen, resulting in,

$$\iint_{A_k} ds = \oint_{C_k} -y \cdot dx \tag{4.3}$$

which is the well known Riemann integral. This can be simplified into,

$$A_k = -\oint_{C_k} y \cdot dx \tag{4.4}$$

Knowing that C can be represented parametrically by $\big(x(t), y(t)\big)$ and that the input signal is periodic, the equation can be further simplified into,

$$A_k = -\int_{(k-1)\cdot\frac{T}{2}}^{k\cdot\frac{T}{2}} \left(y \cdot \frac{dx}{dt}\right) \cdot dt \tag{4.5}$$

where $T = 2\pi/\omega$ is the frequency of the input signal. To obtain the total area of the v-i loop, all that is needed is to sum all the areas enclosed by C in the absolute sense,

$$A = |A_1| + |A_2| \tag{4.6}$$

4.2 – Evaluation of the length of the v-i loop

To calculate the length of the v-i loop, one can define that the length is the integral of all the infinitesimal dl segments that compose the loop:

$$L = \oint dl \tag{4.7}$$

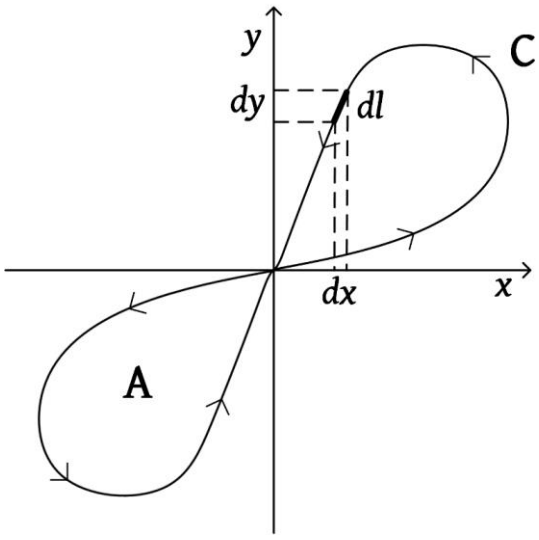


Figure 4.2: Hysteresis loop for length calculation.

Knowing that $dl^2 = dx^2 + dy^2$, as can be seen in Figure 4.2, (4.7) becomes,

$$L = \oint \sqrt{dx^2 + dy^2} \tag{4.8}$$

dividing and multiplying the argument of the square root by dx^2, applying limits to the integration and simplifying, (4.8) can be written as,

$$L = \int_{x_i}^{x_f} \sqrt{1 + \left(\frac{dy}{dx}\right)^2} \cdot dx \tag{4.9}$$

Using again the knowledge that C can be represented parametrically by $(x(t), y(t))$ and simplifying the result, (4.9) will become,

$$L = \int_{t_i}^{t_f} \sqrt{\left(\frac{dx}{dt}\right)^2 + \left(\frac{dy}{dt}\right)^2} \cdot dt \tag{4.10}$$

Knowing that the input signal is periodic, t_i and t_f can be replaced by 0 and T respectively, changing (4.10) finally to:

$$L = \int_{0}^{T} \sqrt{\left(\frac{dx}{dt}\right)^2 + \left(\frac{dy}{dt}\right)^2} \cdot dt \tag{4.11}$$

4.3 - The MATLAB algorithm

To perform the calculations of the area and length of the v-i loop, the values of voltage and current need to be found first. To begin, a time array of 2000000 equally spaced values is created, starting with 0 and spanning 2 periods of the input signal. This was done because it became evident that the system reached stationary regime after the first period.

To obtain the voltage values, a sinusoidal signal centred in 0 was created with the same length as the time array, resulting in an array of voltage values the same length of the time array:

```
v=v0*sin(w*t);
```

Where `v0` is the maximum amplitude of the signal, usually *1 V*, unless specified, `w` is the angular frequency of the signal and `t` is the time array. This code is included in a function called `stimvoltage` (present in the annex at the end of the text) that is capable of generating other kind of periodic signals, like square waves or sawtooths, if so is desired. However, for the purpose of these tests, only the sinusoidal signal was used.

To obtain the current flowing through the device the memristance needs to be found. The equation of the memristance varies from model to model, but all the models have a state variable that is a function of its derivatives, usually just the first derivative, which means that the memristance is specified by an ordinary differential equation (ODE). There are many methods to solve ODEs, but for the sake of simplicity the forward Euler method for first-order ODEs was used, [45]. The concept for this method is easy to understand: knowing that the differential equation is given by,

$$\dot{w}(t) = f\big(t, w(t)\big) \tag{4.12}$$

if a solution for the initial value problem exists, $w(t_0) = w_0$, a positive constant value, h, can be chosen to define each step of time as,

$$t_{n+1} = t_n + h \quad , \quad n \in \mathbb{N}_0 \tag{4.13}$$

This translates to a discreet step in the function $w(t)$ that is being solved by,

$$w_{n+1} = w_n + h \cdot f\big(t, w(t)\big) \tag{4.14}$$

Although this method is simple it is not exempt from problems. For the integration to have a satisfying degree of accuracy the step needs to be very small, which increases the computation time. Furthermore the forward Euler method can be numerically unstable for stiff equations. The last of these problems was observable in certain conditions, but some workaround precautions were put in place to avoid serious repercussions in the end results, namely restricting the total simulation time and imposing verifications in the retrieval measures.

Lastly, after knowing how to get the memristance, the current can be obtained by equation (2.8). Below is an extract of code used in the calculations described. The equations used were from the Strukov model, (3.1) and the Prodromakis window function (3.33):

```
W(1) = w0;
x = W(1)/D;
M(1) = Roff+(Ron-Roff)*x;
F(1) = Af*(1-((x-0.5)^2+0.75)^p);
for n = 2:N
    dW = K*i(n-1).*F(n-1);
    if ((x == 1) && (dW > 0))
        dW = 0;
    elseif ((x == 0) && (dW < 0))
        dW = 0;
    end
    W(n) = W(n-1)+dW*dt;
    x = W(n)/D;
    if x > 1
        x = 1;
    elseif x < 0
        x = 0;
```

```
    end
    M(n) = Roff+(Ron-Roff)*x;
    i(n) = v(n)./M(n);
    F(n) = Af*(1-((x-0.5)^2+0.75)^p);
end
```

Finally, with the values of current and voltage found, the calculations of area and length can be performed. To that end, first the voltage and current of a single period need to be isolated, more specifically the second period, for reasons already explained, the arrays `v2` and `i2` were created using the instructions,

```
N2 = N/2;
i2 = i(N2+1:end);
v2 = v(N2+1:end);
```

Then, using the MATLAB vectorized notation and difference vectors of the voltage, `dv2` and current, `di2`, the area and length are:

```
di2(1) = i2(1)-i(N2);
di2(2:end) = i2(2:end)-i2(1:end-1);
dv2(1) = v2(1)-v(N2);
dv2(2:end) = v2(2:end)-v2(1:end-1);
A(counter_variable,counter_v0,n) =
abs(i2(1:N2/2)*dv2(1:N2/2)')+abs(i2(N2/2:end)*dv2(N2/2:end)');
L(counter_variable,counter_v0,n) = (sum(sqrt((di2').^2+(dv2').^2)))';
```

A complete example of the code used is presented in the annex at the end of the document.

4.4 – Finding the figures of merit

An analysis of the model's response was conducted for two cases, first for a sweep of the input signal's frequency and then for a sweep of the input signal's maximum amplitude. For both cases the model used first to find the figures of merit was the Strukov model with the Prodromakis window function.

4.4.1 – Figures of merit for the frequency sweep

To check the behaviour of the model to a large range of frequencies, with a preliminary test ranging from *1 rad/s* to *100 rad/s*, the areas and lengths of the v-i loops were plotted against the frequency of the input signal. However, to make the results easier to interpret, a logarithmic conversion of the areas was made, according to the equation,

$$A(dBW) = 10 \cdot \log_{10}(|A|) \tag{4.15}$$

The results of the plot can be seen in Figure 4.3, below.

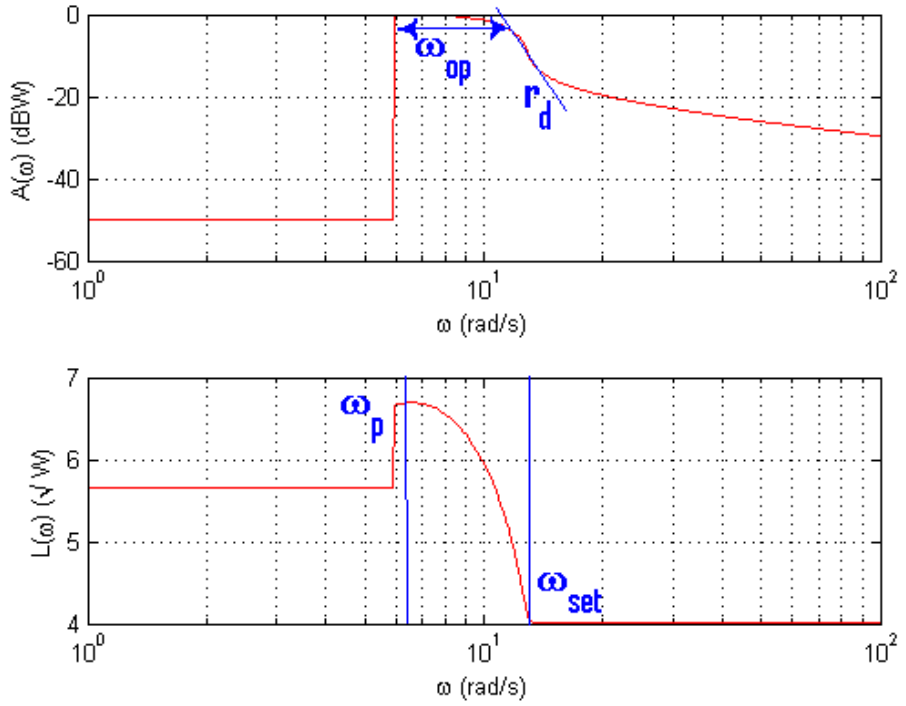


Figure 4.3: Area and Length of a memristor's v-i loop for a frequency sweep.

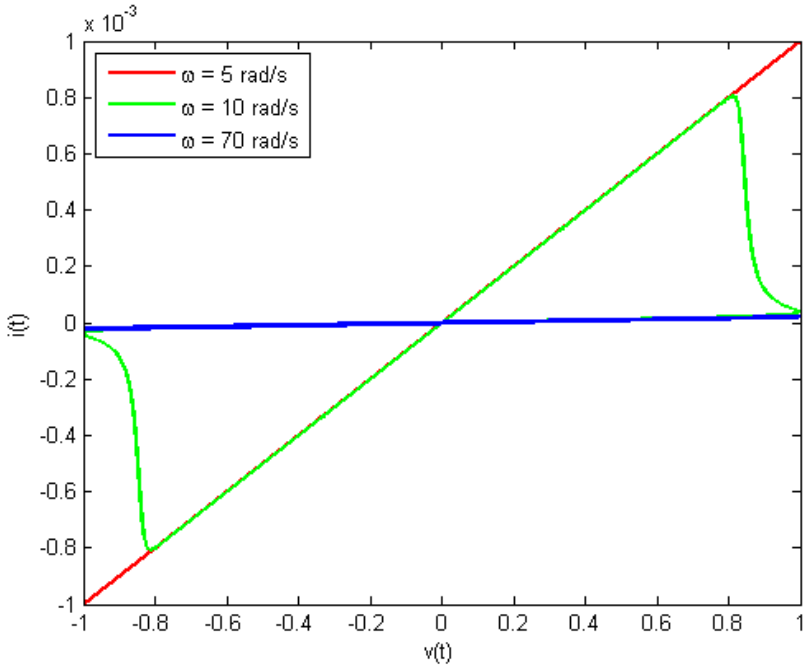


Figure 4.4: Details of v-i loops at some frequencies of Figure 4.3.

From Figure 4.3 and from v-i loops taken at strategic point which were plotted in Figure 4.4, some conclusions can be drawn. First, for lower values of frequency, there is no loop, only a line of inclination R_{on}, meaning that the memristor behaves just like a linear resistor with resistance R_{on}. After an increase of frequency the area and length suddenly rise, indicating that there is memristive behaviour. Both plots reach a peak and then slowly fall, until the loop length stabilises again and the area decreases gradually. This leads to believe that at higher frequencies the memristor has a behaviour increasingly linear, until it is indistinguishable from a resistor with resistance R_{off}, confirming two of the features mentioned before and published in [21]: that the memristor collapses into a resistor at high frequencies and that the loop area decreases with increasing frequencies. Finally, in the frequency band where the memristor displays the most significant area, four figures of merit can be considered. These figures are, the frequency at which the loop area and length are maximum (ω_p), the operational bandwidth around this frequency where the area is within *-3 dB* from its maximum (ω_{op}), the frequency at which the memristor collapses into a resistor (ω_{set}) and the rate of decay of the loop area from its peak to the collapse of the loop, defined by,

$$r_d = \frac{A(\omega_{set}) - A(\omega_p)}{\omega_{set} - \omega_p} \tag{4.16}$$

In order to simplify the analysis of the results, only the absolute value of r_d is taken in consideration.

4.4.2 – Figures of merit of the amplitude sweep

To find the figures of merit for the amplitude, a preliminary test similar to the one made for the frequency sweep was conducted. In this test the input signal frequency was kept constant at 100 Hz and the maximum voltage ranged from *10 V* to *1 kV*. This range of voltages was chosen to guaranty the capture of all the important changes of area and length with the amplitude of the input signal. The area was once again converted logarithmically by (4.15). The area and length results can be seen in Figure 4.5 and some selected v-i loops are plotted in Figure 4.6.

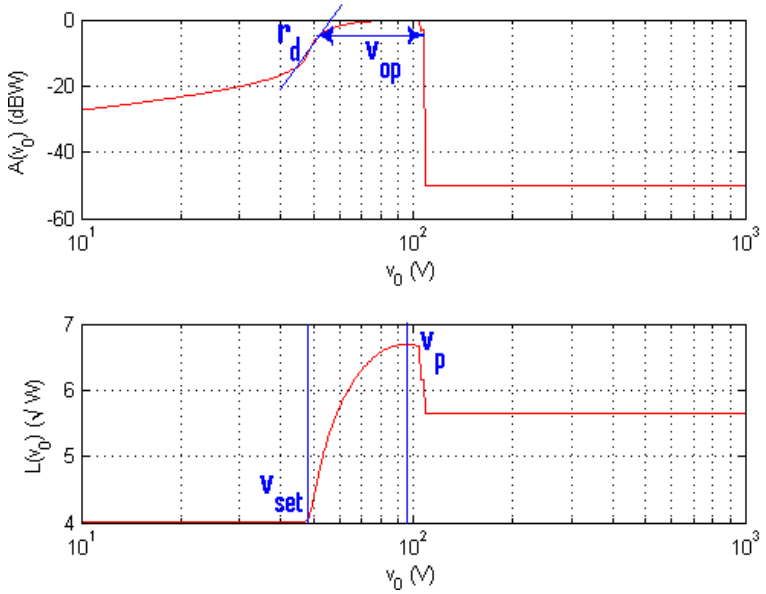


Figure 4.5: Area and Length of a memristor's v-i loop for an amplitude sweep.

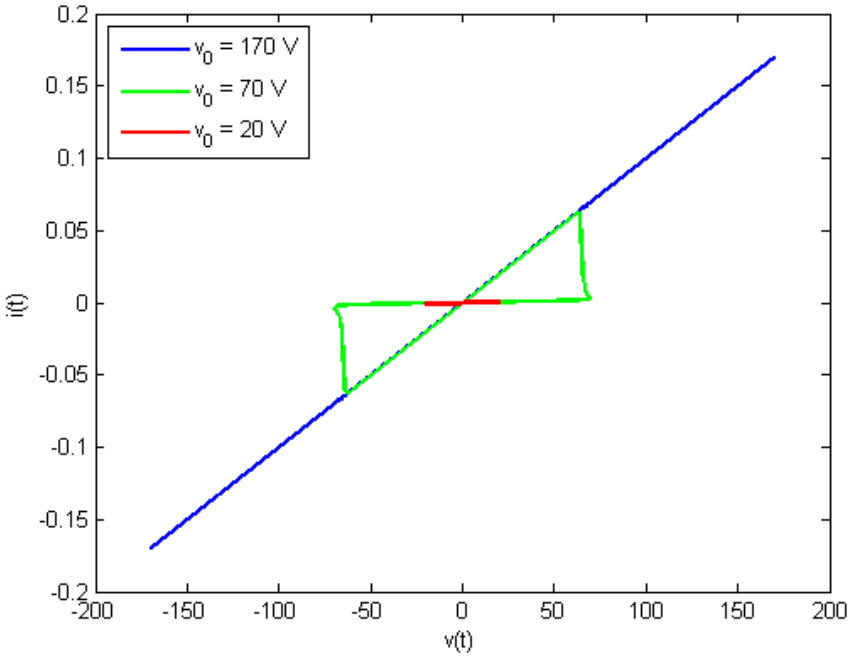


Figure 4.6: Details of v-i loops at some amplitudes of Figure 4.5.

The test results have a pattern remarkably similar to the previous, with the curves showing a similar shape although inverted. This simplifies greatly the analysis because many of the earlier conclusions can be reapplied with only minor changes. First we observe that the full memristive behavior only happens inside a very short window of input amplitudes. Below a certain value of amplitude the memristor collapses gradually into a resistor with resistance R_{off}. Above another higher value of amplitude, the memristive behavior in nonexistent and the memristor locks into the R_{on} value.

The final conclusion that can be dawn is that the figures of merit for the amplitude sweep continue to be four. These figures are the peak amplitude (v_p), the amplitude at which the area and length reach their highest value. The dynamic operation range (v_{op}), the dynamic range where the area maintains *-3 dB* from its maximum value. The settling amplitude (v_{set}), the minimum amplitude for the memristive behavior to be considerable. Finally the rate to decay of the memristive behavior (r_d), is the rate at which the v-i loop collapses into a line, as the amplitude decreases from v_p to v_{set}, defined by,

$$r_d = \frac{A(v_{set}) - A(v_p)}{v_{set} - v_p} \tag{4.17}$$

Like in the frequency analysis, only the absolute value of r_d is taken in consideration.

4.5 - MATLAB results

Knowing what figures of merit need to be found, the next step was to test the various models with and array of input signals and changing as many model parameters as possible, one at a time, to check and analyse the reaction of the models. The tests were conducted with the Strukov model and the window functions of Prodromakis, Joglekar, HP, Biolek and BCM.

Looking at equations (3.1) and (3.6), it is noticeable that the Strukov model has constants that represent physical characteristics of the device. Usually some of these characteristics can be modified, for instance, the device's length (D), the ion mobility (μ_v) (assuming it can be influenced), the initial position of the interface between the doped and undoped areas (w_0), the resistance of the device fully doped (R_{on}) and the ratio between the resistance of the device fully doped and fully undoped ($R_{ratio} = R_{on}/R_{off}$).

The window functions tested were configured as follows: the HP and Biolek window functions were used without changes; the parameters for the Prodromakis window function were set at $k = 1$ and $p = 10$; for the Joglekar function $p = 10$ and for the BCM function $v_{th0} = v_0/4$ and $v_{th1} = v_0/4$, with v_0 being the amplitude of the input signal.

The next step was to perform frequency and amplitude sweeps for different values of the physical parameters of the device. This was accomplished changing only one parameter for each test. For the parameters that were not being tested, default values were assigned as follows: $D = 10\ nm, w_0 = D/2\ \ nm, \mu_v = 10^{-10}\ \ cm^2/Vs, R_{on} = 100\ k\Omega$ and $R_{ratio} = 100$.

4.5.1 – Frequency sweep

The plots in Figure 4.7 through Figure 4.10 summarize the measurements taken from the simulations of the Strukov model with the different window functions. Each figure is the test results for the variation of a single physical parameter, for a sweep of 300 points in the input signal's frequency. The minimum and maximum sweep frequencies changed accordingly to the window function being tested, this was made to ensure that all figures of merit were observable in all tests.

A first analysis of all the plots reveals that, although the measurements for the all windows are different, for most of their development is very similar and consistent, with the

only exceptions being the functions with conditional transitions. A more careful observation of each individual set of plots reveals other conclusions.

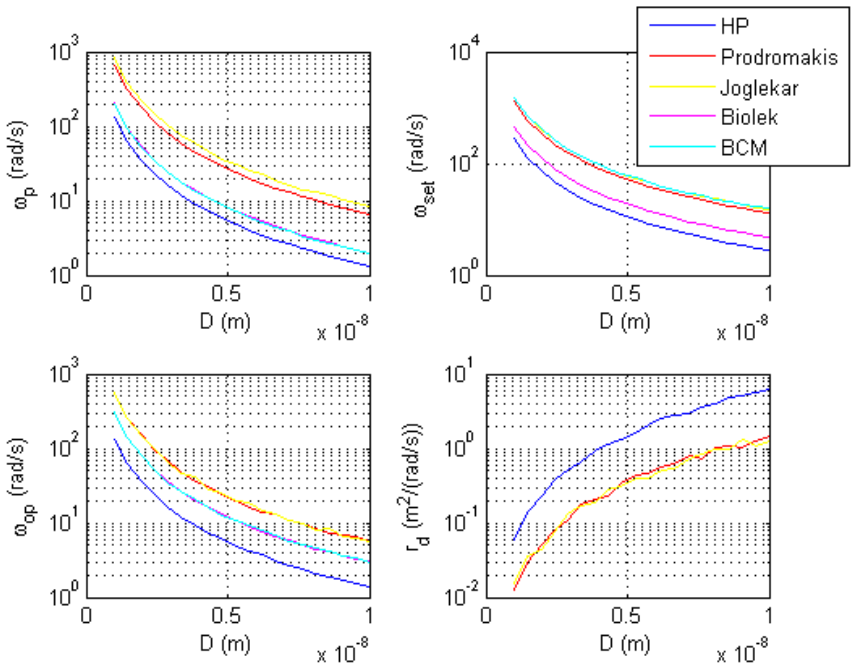


Figure 4.7: Measures for variable D with frequency sweep.

Figure 4.7 depicts the measured figures of merit for the devices' length sweep (D) from 1 nm to 10 nm. As can be seen, ω_{op}, ω_p and ω_{set} increase as D decreases, showing that smaller devices tend to have the peak of their memristive behaviour at higher frequencies. This is consistent to what is asserted in [2], that the time needed for the dopants to cross the full length of the device increases with D. The rate of decay (r_d) however, decreased with D, meaning that smaller devices are able to keep their memristive behaviour for wider bandwidths of the input frequency.

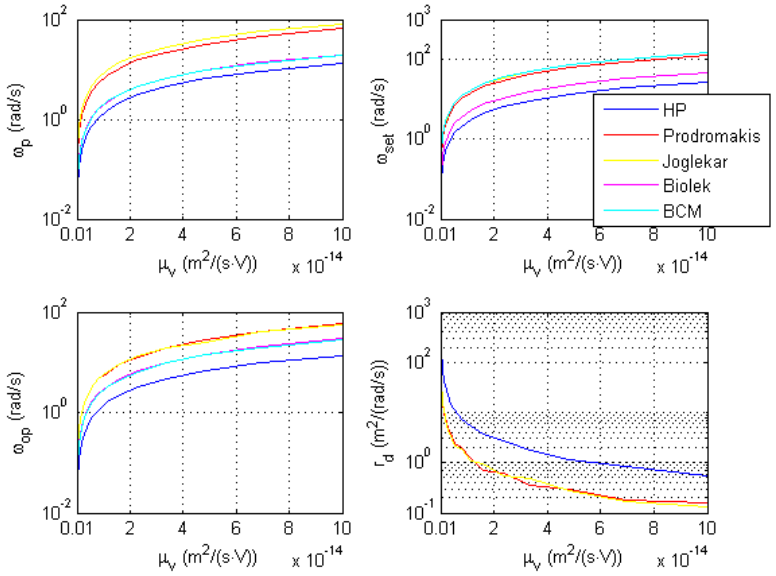


Figure 4.8: Measures for variable μ_v with frequency sweep.

Figure 4.8 displays the devices' dependence with the ion mobility between $10^{-12}\ cm^2/Vs$ and $10^{-9}\ cm^2/Vs$. If the ion mobility can be somehow increased, the devices' response changes. In this case, ω_{op}, ω_p and ω_{set} increase as well, this means that an higher electron mobility permits the devices to operate as an MD at higher frequencies. An increase of the ion mobility results in a decrease of the decay rate, meaning that the devices will not only be able to operate at higher frequencies, but keep the memristive behaviour for greater bandwidths.

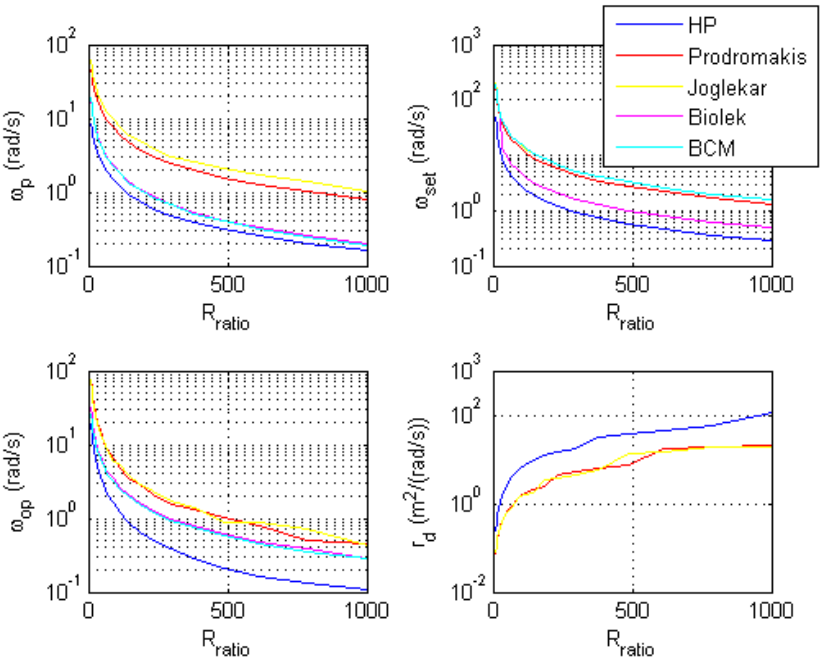


Figure 4.9: Measures for variable R_{ratio} with frequency sweep.

Figure 4.9 demonstrates the frequency behaviour of the devices with a changing ratio between the limit memristances. This ratio was changed from 10 to 1000. As it is shown, the greater the ratio between resistances, the lower are the figures of merit ω_{op}, ω_p and ω_{set}. This can be understood as a decrease of the devices' optimal operation frequencies if the ratio between the limit memristance values increases. An increase of the ratio translates into a faster decay of the hysteresis loop (r_d), meaning that the devices collapse faster into a resistor.

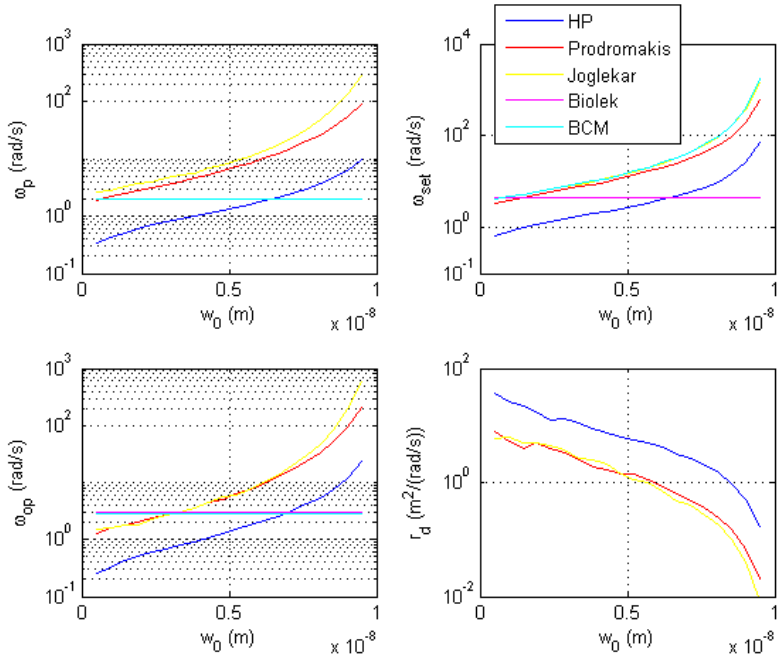


Figure 4.10: Measures for w_0 variable with frequency sweep.

Figure 4.10 presents the frequency performance against the initial position of the interface w_0, for values between *5%* and *95%* of *D*. As can be seen, the initial position of the interface has a huge contribution to the performance of the devices, except for the Biolek and BCM functions. As observed by the increase of ω_{op}, ω_p and ω_{set} and the decrease of r_d with w_0, higher doping profiles contributes to an enhancement of the memristive behaviour at higher frequencies. In the case of the two exceptions, a change of w_0 has no effect in their figures of merit, except for the settling frequency of the BCM window function.

The measurements concerning the dependency of the measures with R_{on} are not presented here, although these tests were conducted. They showed that changing R_{on} has no influence in the outcome. As can be seen in the example of Figure 4.11, two v-i loops of the HP window at the frequency of 1.5 rad/s with different values of R_{on} have a different shape, this

however does not translate in a change of the figures of merit. This lack of influence in the results from the variation of R_{on} alone, means that even though the shape of the v-i loop changes, the frequency at which the peak is reached, the frequencies that the area drops $-3\ dB$ from the maximum, the settling frequency and the rate of decay, all remain the same.

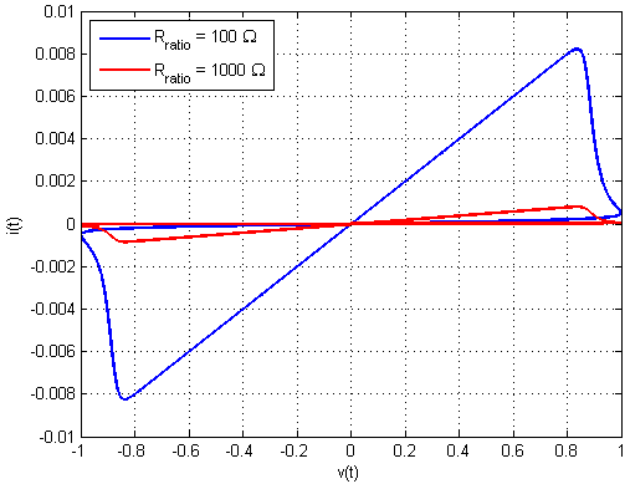


Figure 4.11: v-i loop of the HP window function for different values of R_{on}.

A measurement that is lacking is the rate of decay for the Biolek and BCM window functions. This is justified by the peculiar shape of their area and length curves in both cases (Figure 4.12), which makes the equation for the calculation of the rate of decay (4.17) not applicable.

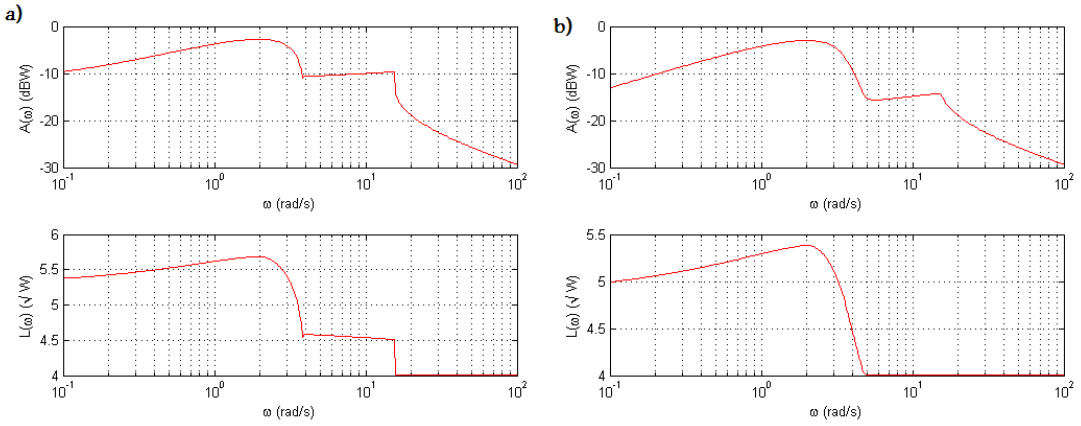


Figure 4.12: Length and area curves: a) BCM window, b) Biolek window.

This behaviour is caused by an initial asymmetry of the windows' v-i loops, which in turn is justified by the conditional transitions in the equations (3.36) and (3.40) of the models. Another interesting fact is that unlike in the Biolek window function, it is possible to change the threshold values of the BCM function, which can drastically change its behaviour. These variations are not explored in this chapter, but detailed area and length curves and samples of the v-i loops for the various window functions and for variations of the v_{th} values of the BCM function are present in annex at the end of the text.

4.5.2 – Amplitude sweep

The plots in Figure 4.13 through Figure 4.16 summarize the measurements taken from the simulations of the Strukov model with different window functions. This data has obtained for individual variations of the model's parameters and a sweep in the maximum amplitude of the input signal of three hundred points. To check the consistency of the results across various frequencies of the input signal, these tests were carried out for three frequencies: 1 Hz, 100 Hz, and 1 kHz. The test showed that despite the expected differences in the responses, the rates of change with the input amplitude were equal to all test frequencies. Therefore, to avoid clutter, only one test frequency (1 Hz) is presented below, the complete set is present, in the annex.

As it was observable with the frequency sweeps, with the exception of the Biolek and BCM window functions, the remaining ones have a similar response to the change of value for the models parameters. Another similarity is the lack of rate of decay for the Biolek and BCM functions. This is caused again by the peculiar shape of the associated area and length curves, which just like the example in section 4.4, is an inversion of the results from the frequency sweep.

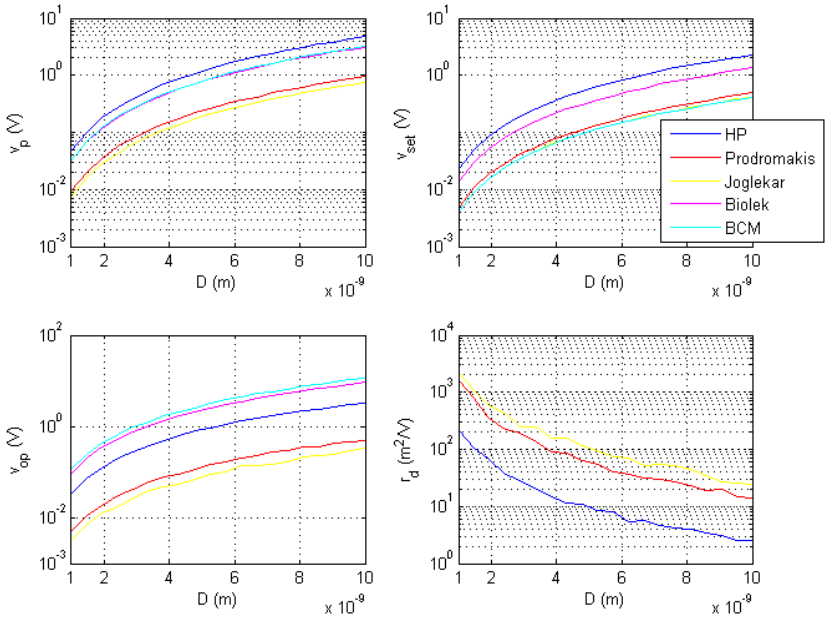


Figure 4.13: Measures for D variable with amplitude sweep and $f = 1\ Hz$.

Figure 4.13 depicts the figures of merit measured for the various window functions, for a sweep of D from 1 nm to 10 nm. An increase of D triggers an increase of v_p, v_{op} and v_{set} and an decrease of r_d. This means that the smaller devices can exhibit hysteretic behaviour without requiring high amplitudes of the input signal.

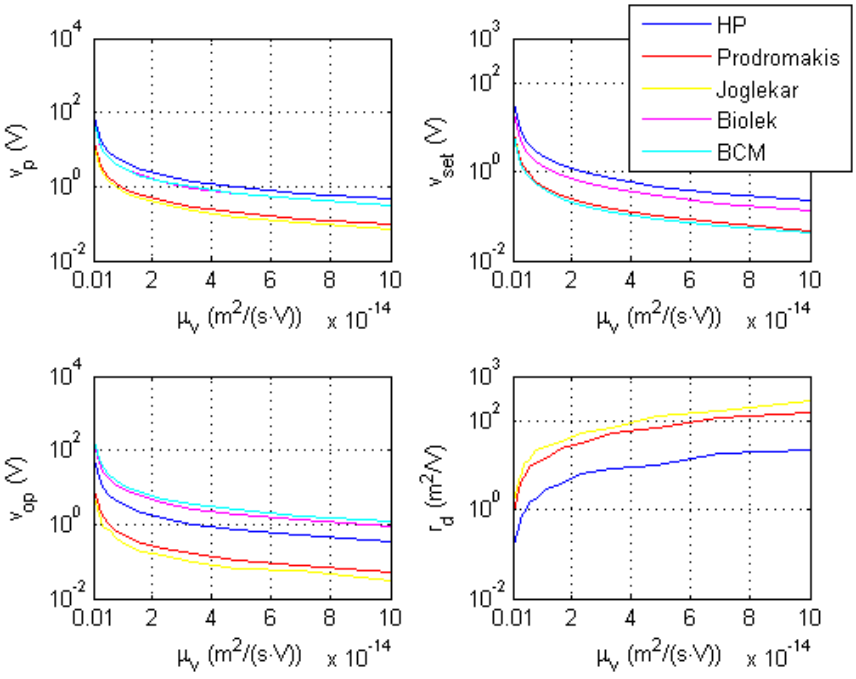


Figure 4.14: Measures for μ_v variable with amplitude sweep and$f = 1\ Hz$.

The plots in Figure 4.14 show the dependency of the ion mobility with the amplitude of the input signal, with a gradual variation of μ_v from $10^{-12}\ cm^2/Vs$ to $10^{-9}\ cm^2/Vs$. An analysis of the results demonstrates that devices with a fasted ion mobility have the peak of their memristive behaviour at lower amplitude of the input. This can be observed by the decrease of the v_p and v_{set} measures with an increase of the ion mobility. However, this increase of the mobility reduces v_{op} and r_d, meaning that the range of input values that guarantee a memristive behaviour is shortened.

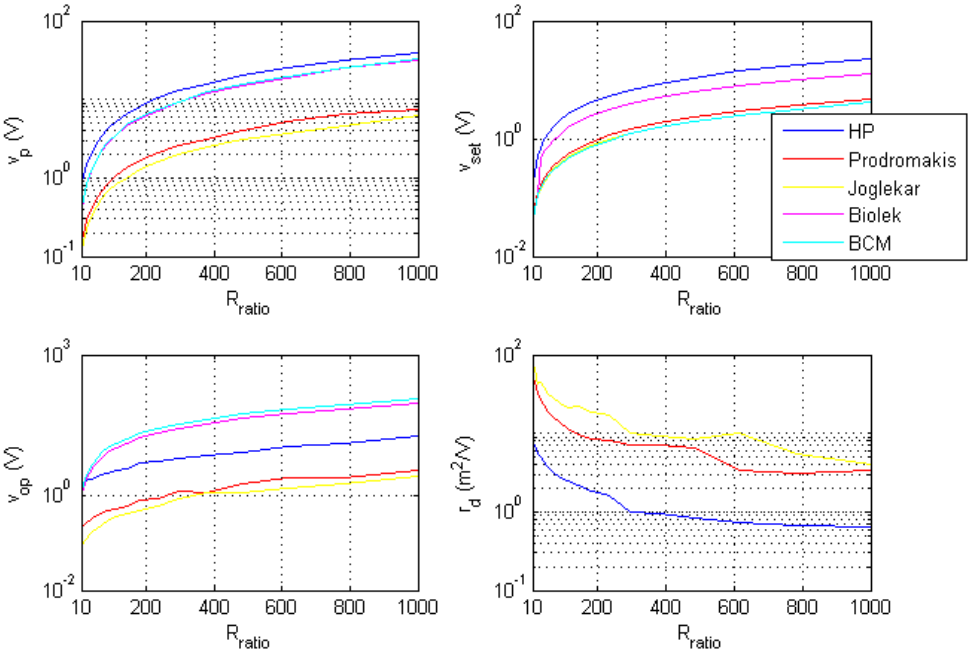


Figure 4.15: Measures for R_{ratio} variable with amplitude sweep and $f = 1\ Hz$.

In Figure 4.15, the figures of merit are plotted against the ratio between R_{on} and R_{off}, for values between 10 and 1000. It is noticeable that the hysteretic behaviour of the devices happens at higher amplitudes if the ratio increases. This is confirmed by the increase of v_p and v_{set}. The increased need of higher amplitudes for the memristive operation of the devices is accompanied by a growth of the operation range, as confirmed by the rise of v_{op} and decrease of r_d, as R_{ratio} increases.

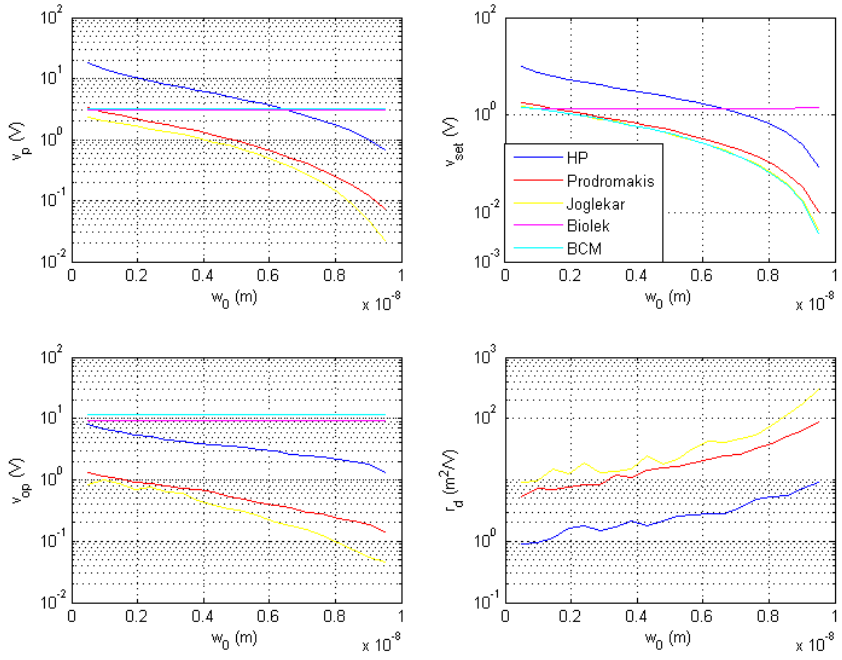


Figure 4.16: Measures for w_0 variable with amplitude sweep and $f = 1\ Hz$.

Figure 4.16 shows the measurements for an initial position of the interface, w_0 between *5%* and *95%* of *D*. An analysis of this figure leads to the conclusion that an increase of this parameter has the effects of lowering the amplitude needed for memristive operation, as observed by the decrease of v_p and v_{set}. This lower requirements of amplitude are accompanied by a reduction of the operation range, as seen by the decrease of v_{op} and increase of r_d. These interpretations differ for the Biolek and BCM windows. For BCM, at least for the case studied, $v_{th0} = v_{th1}$, changes of the initial position (w_0) have no effect in the device's response. In the case of the Biolek window only the device's settling amplitude suffers alterations, decreasing of value as w_0 increases.

The results pertaining to the analysis of the devices behaviour for different input amplitudes with a variable R_{on} are not shown here. Just like with the frequency sweep, variations of R_{on} modify the hysteresis of the v-i loop, but have no influence in the model's figures of merit for any window functions.

4.6 – Final remarks

The analysis of the Strukov MD model's v-i loop and some of the window functions that can be applied to it, showed that at least four figures of merit can be extracted from the area and length of the loop. This figures of merit can be extracted both for a variation of the input signal's frequency and for a variation of the input signal's amplitude. Another strong point of these measures stem from the fact that their progression followed a stable pattern when the parameters of the emulated MD model were changed.

However, these measures might not be foolproof. While the studied model and window functions showed a predictable pattern in the progression of the area and length of their v-i loops, nothing can be proved about other MD models untested. Furthermore, the results were generated in MATLAB, samples of voltage and current obtained from physical devices can have a lot of variations. This can result in area and length curves with a less predictable behaviour. These variations in the voltage and current can be caused by defects in the MD measured, electrical noise, or problems of accuracy and precision of the measures.

As a final observation about the results obtained, it is noticeable that the area and length curves of the frequency tests are symmetrical to the curves obtained in the amplitude tests. This symmetry could be exploited by running a test that could compromise results from both tests. This new test could be a sweep of the frequency normalized against the amplitude, $\Omega = \omega / v_0$.

Chapter V

The memristor emulator

Emulators are important, both as research and as academic tools. Their importance comes from many factors, including versatility and if the device emulated is hard to get or expensive, they provide a cheaper test alternative and a quicker obtainment of results.

In this chapter are present an overview of the available MD emulators, a detailed explanation of the emulator used and the results obtained from it.

5.1 - Emulators in literature

Given the fact that there are no memristive devices easily accessible, circuits with this component cannot be tested. One solution is to perform SPICE simulations. However, if a study with physical devices is required, SPICE simulations are not enough. This is where the memristor emulators become important, because although they do not fully replicate the behaviour of a memristor, their response can be close enough to test the feasibility for circuit design with memristors.

There are several emulators of memristive devices published in literature. They are all composed by active devices made with many different electrical components, ranging from microcontroller based to fully analog. Depending on the approach taken, these designs have different advantages and disadvantages.

5.1.1 – Pershin and Di Ventra emulator

This emulator was published in [46] and is a very simple and relatively cheap emulator. It consists in a digital potentiometer that acts as the variable resistance (memristance) of the memristor. To control the potentiometer wiper, the voltage at its terminals is sampled by an ADC and read by a microcontroller that changes the value of the potentiometer wiper accordingly. A diagram of the emulator is represented in Figure 5.1.

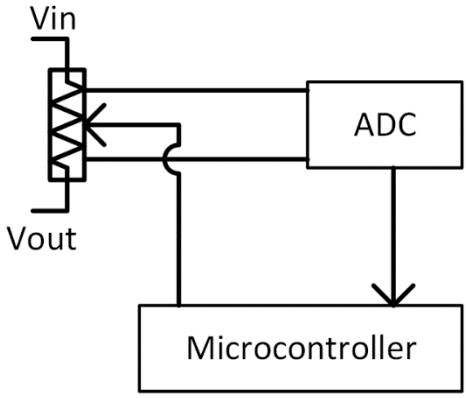


Figure 5.1: Diagram of the Pershin and Di Ventra emulator.

Despite its simplicity, this emulator has some shortcomings. The most obvious one is the very low operation frequency reported by the authors [46]. The other is the fact that it must operate with one grounded terminal [47]. However, its ease of use and low price has made the authors consider it as a teaching tool for students [33]. Furthermore, due to the fact that it is a microcontroller based approach, its behaviour can the adjusted if needed. Another advantage of this emulator is that it has room for improvement, its operation frequency can be increased if higher frequency microcontrollers and digital potentiometers are employed.

5.1.2 - Hyongsuk Kim emulator

This emulator was published by Hyongsuk Kim, et al. in [18]. It is based on the concept of an input resistance dependent on a control voltage. The basic principle of the circuit is represented in Figure 5.2.

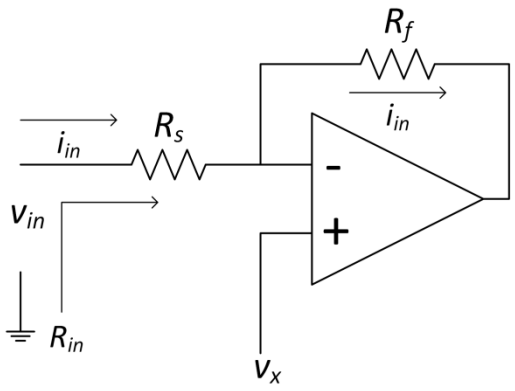


Figure 5.2: Diagram of the Hyongsuk Kim emulator.

The voltage equation at the input of the emulator is,

$$v_{in} = R_s i_{in} + v_x \tag{5.1}$$

Where i_{in} is the input current, R_s is the fixed resistance at the inverting terminal and v_x is the control voltage, applied to the positive terminal of the opamp.

Assuming that v_x is proportional to the input current, (5.1) can be rewritten as,

$$v_{in} = (R_s + m) i_{in} \tag{5.2}$$

Where m is a proportional constant and $v_x = m i_{in}$. From equation (5.2) it can be implied that the input resistance of the emulator is $R_s + m$. If m can be made a function of the input current's time integral, the circuit can behave as a memristive device. To this end a resistor, a capacitor and an analog multiplier are employed, as in Figure 5.3.

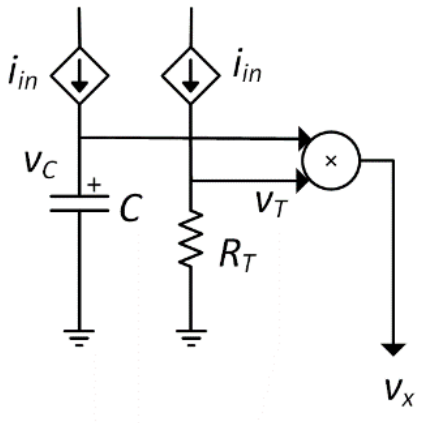


Figure 5.3: Detail of the control voltage of the Hyongsuk Kim emulator.

In Figure 5.3, the current sources mirror the input current i_{in}. The capacitor yields a voltage v_c resulting from the integration of i_{in} and the voltage across the resistor R_T is proportional to i_{in}. The result of the multiplication is the control voltage v_x,

$$v_x = \frac{q_c}{C} \times R_T i_{in} \tag{5.3}$$

where q_c is the charge stored in C.

Using (5.3) in (5.2) results in,

$$v_{in} = \left(R_s + \frac{q_c}{C} \times R_T\right) i_{in} \tag{5.4}$$

where $R_s + \frac{q_c}{C} \times R_T$ is the memristance of the emulator.

This emulator is far more complex than the one proposed by Pershin and Di Ventra and cannot be readjusted once build. However, it follows more closely the TiO_2 model. It can be used without grounded terminals and it has memristive behavior at high frequencies ($100\,Hz$), depending on the modelling employed.

5.1.3 – Mutlu and Karakulak emulator

The Mutlu and Karakulak model was published in [19]. It is based on the same principle of controlled input resistance of the Hyongsuk Kim emulator, but it is simpler. Consisting of two operational amplifiers, a diode, a capacitor an analog multiplier and a few resistors. The schematic of this emulator is replicated in Figure 5.4.

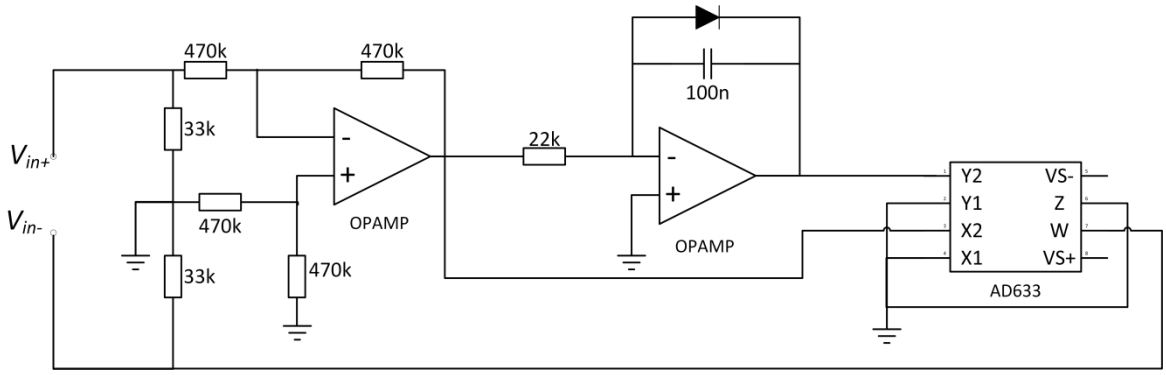


Figure 5.4: Schematic of the Mutlu and Karakulak emulator.

Its simplicity led to an attempt to replicate it, but unsatisfactory results and the lack of detail in the paper about its functioning and the procedures used by the authors to sample the current and voltage at the emulation terminals, led to its exclusion.

5.2 – The memristive device emulator implemented

The selected circuit is based upon the design presented by Sangho Shin et la. in [47]. It consists in a DAC of weighted resistors and an adjacent control and sampling circuit. Although it is as complex as the Hyongsuk Kim emulator, it presents the same benefits; it can be used without a grounded terminal and can emulate memristive behavior at high frequencies (from almost $1\,Hz$ to $1\,kHz$). In addition to these characteristics, its constitution is very well documented in the paper.

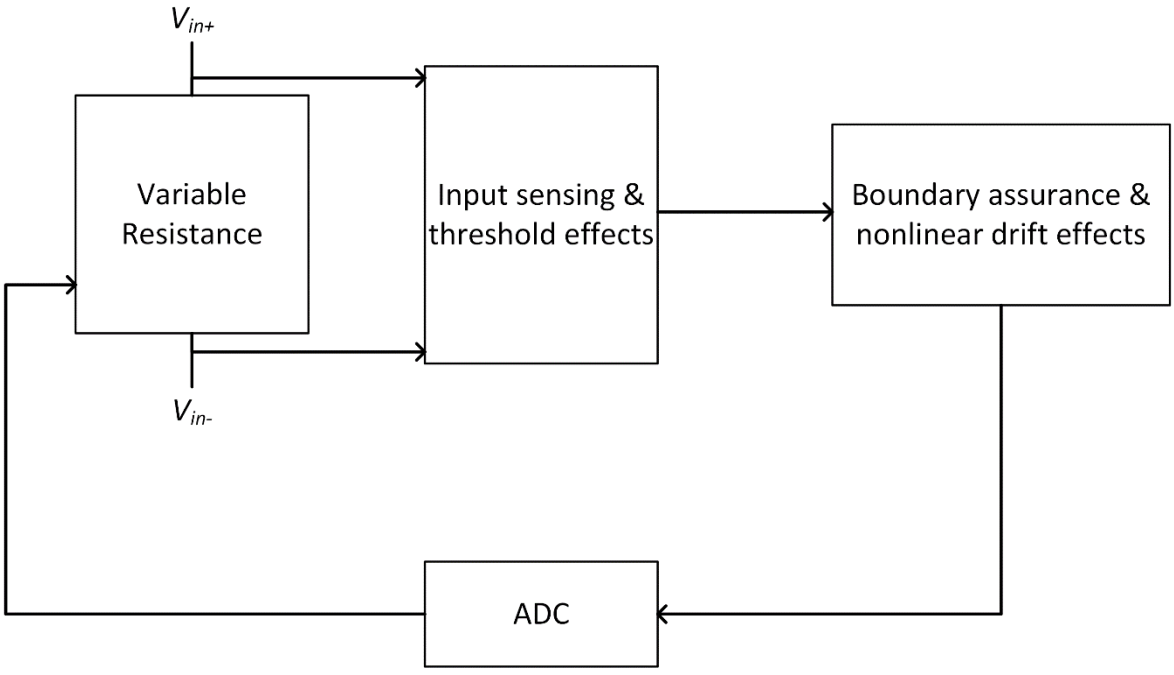


Figure 5.5: Block diagram of the emulador.

The emulator has several blocks, which are linked according to the diagram in Figure 5.5. The emulator supply voltages are $V_{CC} = +5V$ and $V_{EE} = -5V$ and the emulated circuit voltages

(variable resistance block) default maximum values are $\pm 2V$. In the following are described the various blocks, its functions and circuit elements.

5.2.1 – Input sensing and voltage threshold effects generator block

The memristor emulated by this circuit is voltage actuated. For this reason, the first block is a voltage sampler. To sample the voltage without distorting the signal, an instrumentation amplifier is used, with unit gain, as seen in Figure 5.6. After the amplifier, two diodes in anti-parallel are used to simulate the threshold voltages (using the forward voltage drop of the diodes) for the device's dynamics.

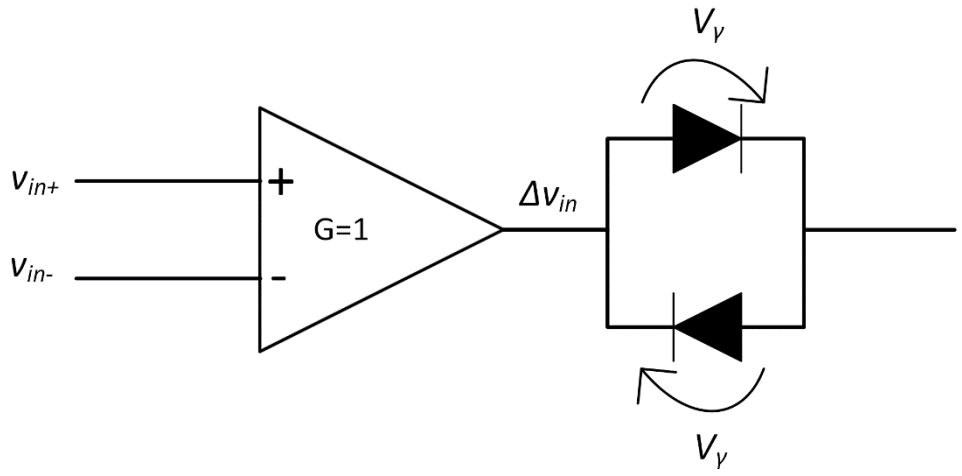


Figure 5.6: Detail of the emulator's first block.

Knowing that the gain of the amplifier is $G = 1$ and assuming the forward voltage drop of both diodes V_γ, the signal at the end of this block is:

$$v_1 = \begin{cases} G(v_{in+} - v_{in-}) - |V_\gamma|, & \Delta v_{in} > V_\gamma \\ 0, & -V_\gamma \leq \Delta v_{in} \leq V_\gamma \\ G(v_{in+} - v_{in-}) + |V_\gamma|, & \Delta v_{in} < -V_\gamma \end{cases} \quad (5.5)$$

This block was implemented with an AD623 instrumentation amplifier, with no gain resistor and two 1N4148 diodes (with forward voltage of $0.6\ V$).

5.2.2 – Boundary assurance and nonlinear drift effects generator block

The second block ensures that the values of the memristance remain within the desired bounds. In order to accomplish that, an integrator is used with the output restricted between two positive values. In the output of the integrator are connected two diodes (one in the anode another in the cathode), with the opposite terminals of the diodes connected to an individual bias voltage. These diodes limit the output of the integrator by providing a discharge path for the integrator's capacitor, when the voltage of the output is enough to allow the diodes to conduct. The nonlinear dopant drift effects at the boundaries are emulated by the diode's exponential switching behaviour. A diagram of this block is in Figure 5.7.

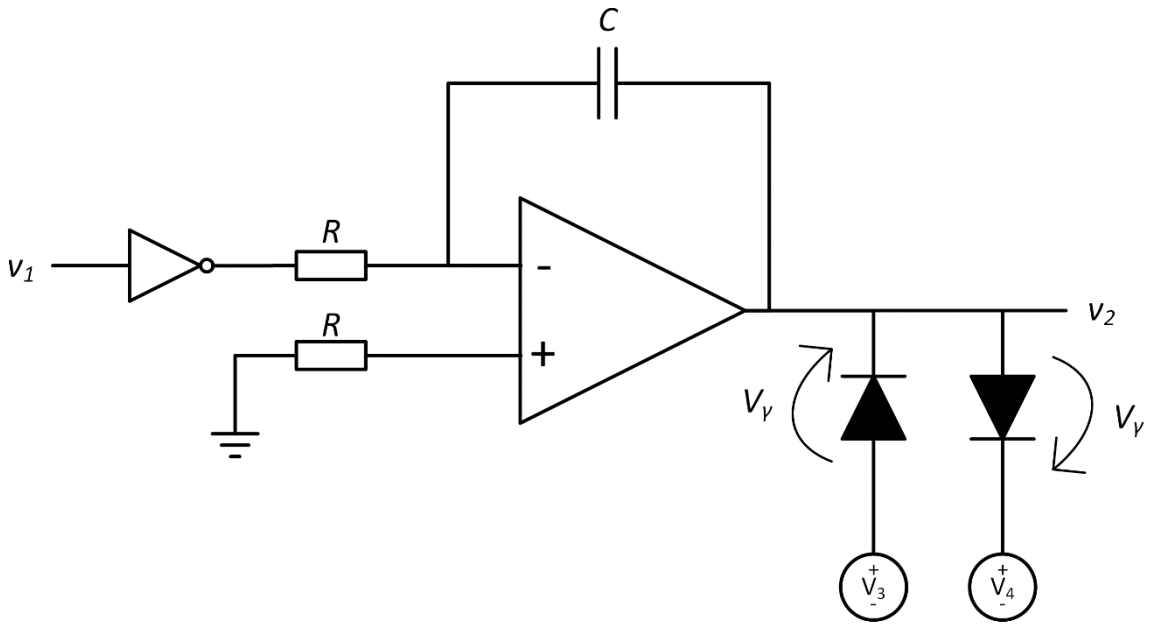


Figure 5.7: Detail of the emulator's second block.

As can be seen in the diagram, an additional inverter is present in the input of the block, it compensates the inversion of the signal in the integrator. The bias voltages, are represented by the DC sources V_3 and V_4.

Assuming the forward voltage drop of both diodes V_γ, the output of this block is,

$$v_2 = \begin{cases} V_4 + V_\gamma, & v_2 \geq V_4 + V_\gamma \\ V_3 + V_\gamma, & v_2 \leq V_3 - V_\gamma \\ \frac{1}{RC}\int v_1 dt\,, & V_3 - V_\gamma \leq v_2 \leq V_4 + V_\gamma \end{cases} \quad (5.6)$$

The following is a list of the components used to assemble this block. An OPAMP TL08x configured as an inverter and two $3k9\ \Omega$ resistors for the inverter. An OPAMP TL08x, a $1\ \mu F$ capacitor and two $560\ k\Omega$ resistors for the integrator, setting the integrator's time constant to $\tau_{int} = RC = 560\ ms$, the second resistor is used to compensate input offset. The two diodes used were 1N4148. The bias voltages chosen were $V_3 = 1.2V$ and $V_4 = 2.8V$. To implement V_3 a voltage divider with $15\ k\Omega$ and $47\ k\Omega$ resistors and a $1\ k\Omega$ potentiometer were used, with a TL08x OPAMP as a buffer in the output. The V_4 bias voltage was implemented in a similar way, replacing the $47\ k\Omega$ resistor with a $12\ k\Omega$ one. Finally, the two diodes used were 1N4148s.

5.2.3 – ADC block

Now, the v_2 signal must be converted into a 6-bit word to be read by the last block, for that, an ADC is needed. This block is composed by and ADC working in standalone mode, with an astable generating its clock signal and a maximum reference voltage close to the maximum of v_2. A diagram of this block is in Figure 5.8.

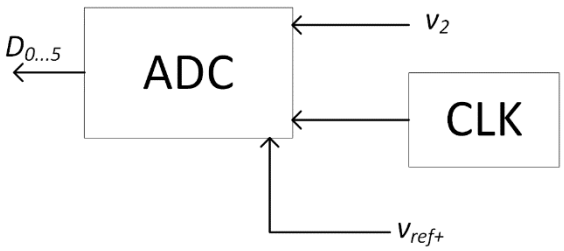


Figure 5.8: Detail of the emulator's third block.

This block is composed by an MAX150 ADC configured in write-read mode in stand-alone operation with reference to ground and V_{ref+}. The clock generator is an astable HEF4047 with a $680\ k\Omega$ resistor and a $101\ pF$ capacitor, generating a square wave with the frequency of $33.09\ kHz$. The reference voltage V_{ref+} is $3.6\ V$ and was created with a voltage divider with a $39\ k\Omega$ and $15\ k\Omega$ resistors, a $1\ k\Omega$ potentiometer and a TL08x OPAMP as a buffer.

5.2.4 – Variable resistance block

The last block of the emulator is responsible for emulating the memristance of the MD, which is a variable resistance. To this end a discreet DAC of weighted resistors was used. The DAC is composed of seven parallel resistances that, with the exception of one, are switched on or off, according to the signal received from the ADC block, changing the equivalent resistance at the terminals of the emulated circuit. The resistance of each parallel branch is half of the previous, with the exception of the first two branch, that have the same resistance ($R_0 = 250\ k\Omega$). These resistors in parallel allow the equivalent resistance of the emulated circuit to swing from $3k9\ \Omega$ to $250\ k\Omega$. A diagram of this block can be seen in Figure 5.9.

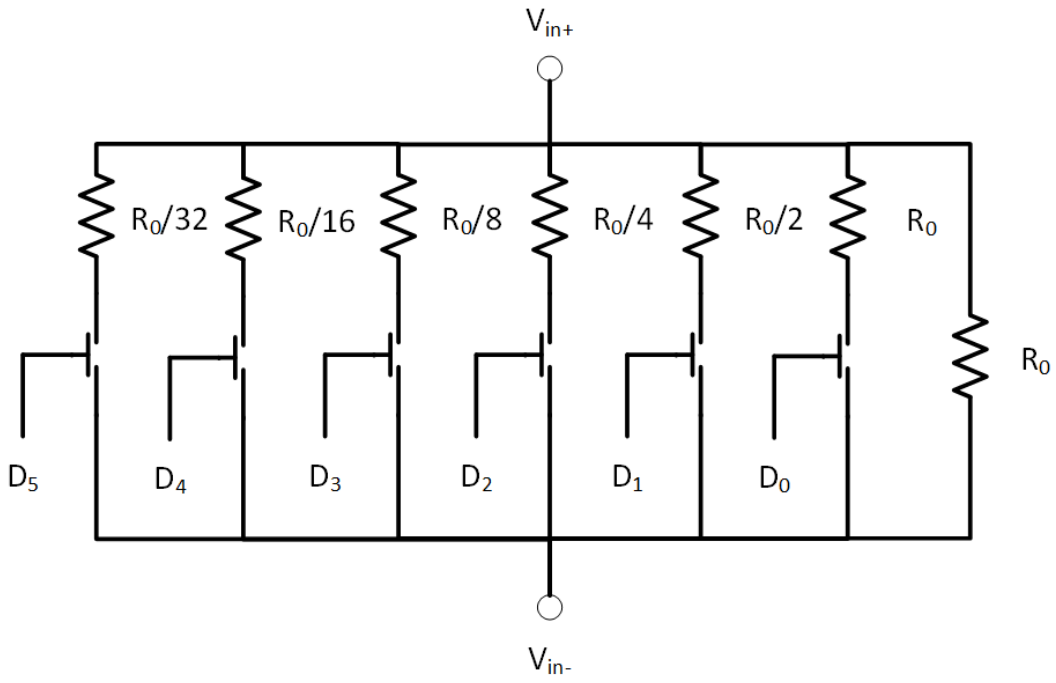


Figure 5.9: Detail of the emulator's fourth block.

To build this DAC, six NMOS transistors, from three ALD1106 ICs were used as switches, with the bits $D_{0\to5}$ coming from the previous block activating the gates. The resistors of each branch with switches were separated in two, to help polarize the transistors. The values of the resistors were chosen to be as close as possible to the ideal ones.

Finally, a photo of the circuit assembled can be seen in Figure 5.10, with additional electronics to allow the sensing of the voltage and current passing through the memristor.

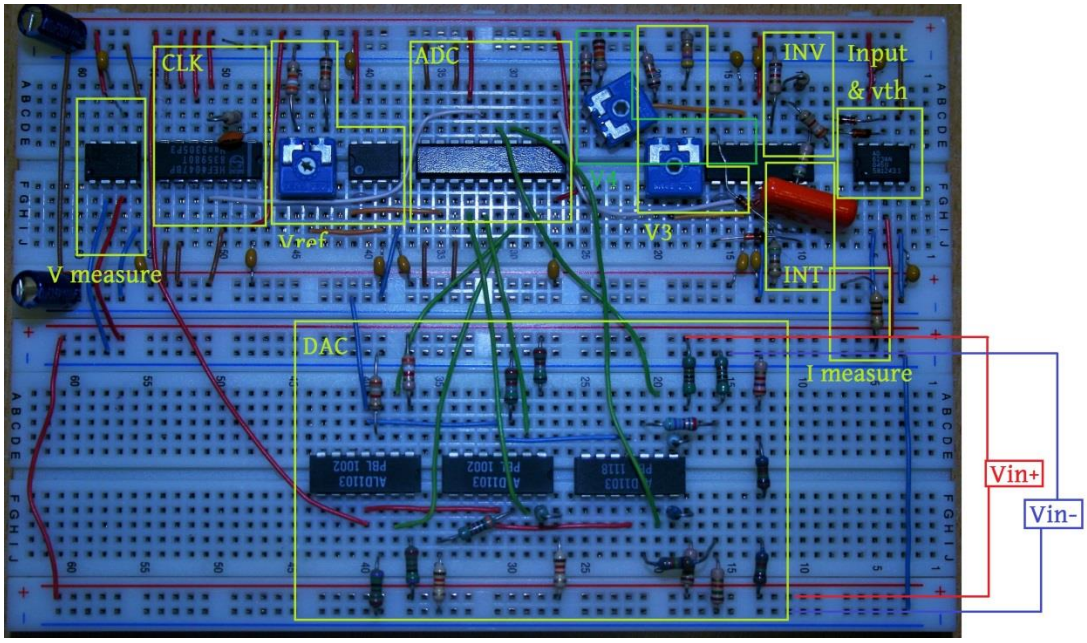


Figure 5.10: Photograph of the assembled emulator with sensing electronics.

The photo depicts not only the emulator, but additional electronics to allow external sensing of the current and voltage through the emulated MD. An explanation of the sensing electronics is in section 5.2 and the electronic schematic of the full circuit is in the annex.

5.2 – Measured Results

The emulator was assembled successfully and presented memristive behaviour for a wide bandwidth of the input voltage at its terminals, as it can be seen in Figure 5.11.

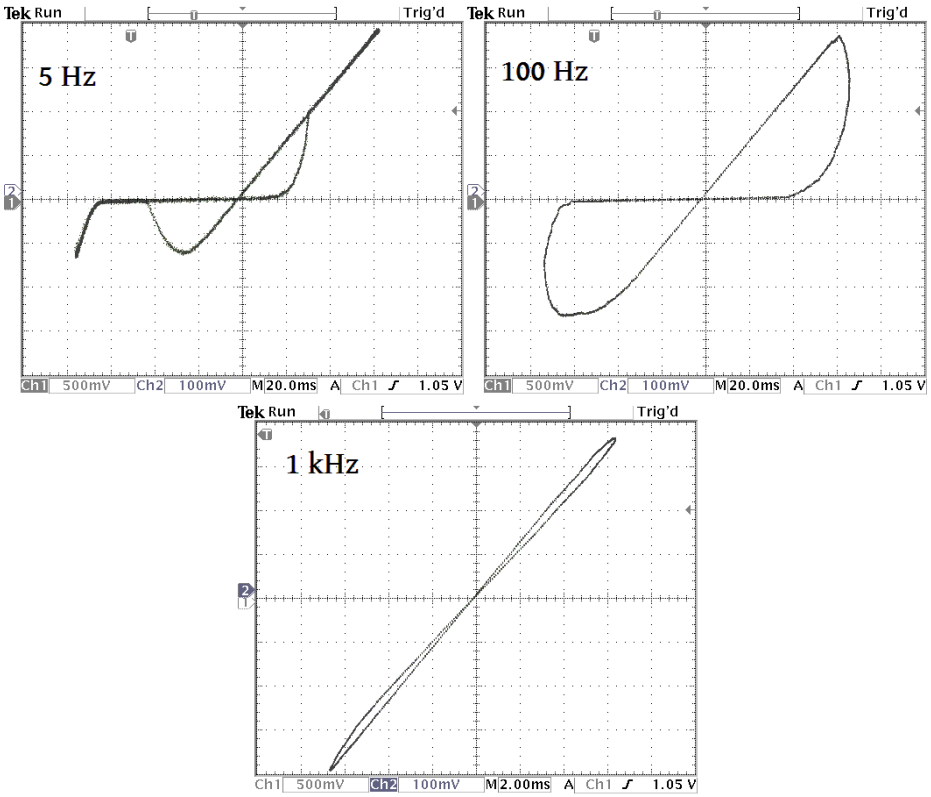


Figure 5.11: voltage-current curve of the emulator at different frequencies.

5.2.1 – Signals sampling

To sense the voltage and the current at the terminals of the emulator two sensing circuits were used as can be seen in Figure 5.12. To sense the current a $1\ k\Omega$ resistor was mounted in series with the emulation terminals and the voltage was sampled at its terminals with the aid of an oscilloscope. To compensate the value of resistance, this voltage was multiplied in the computer by 10^{-3}. Although $1\ k\Omega$ may seem a considerable value, given the fact that the memristance varies between $3.9\ k\Omega$ and $250\ k\Omega$, makes its effect negligible. To sample the voltage, an instrumentation amplifier (AD623) was connected to the emulation terminals and its output voltage was sampled with an oscilloscope, referring to the ground. To reduce noise the oscilloscope performed initially an average with 16 samples, but in an attempt to improve the quality of the results, later 512 samples were used.

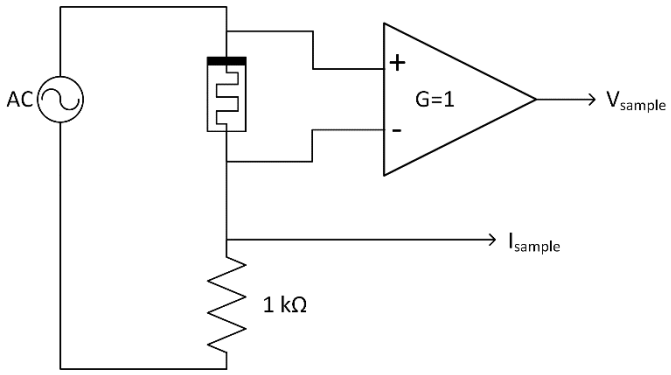


Figure 5.12: Circuit used to sense the current and voltage across the emulated MD.

5.2.2 – Data treatment and calculations of area and length

To prepare the voltage and current data in order to perform the area and length calculations, the data was first imported to MATLAB. Then a single period of the voltage and current data was isolated. The current array was compensated with a multiplication factor of 10^{-3} (as explained before). Afterwards, like in the calculations performed in the simulations of the MD model in MATLAB, both arrays were normalized, in this case against their maximum values. Then the difference vectors of the voltage and current were found and the area and length were calculated using the same method of chapter 4. An extract of the code used is below:

```
v = v/max(abs(v));
i = i/max(abs(i));
N = length(i);
di = zeros(N,1);
dv = zeros(N,1);
di(1,1)=i(1,1)-i(end,1);
di(2:end,1)=i(2:end,1)-i(1:end-1,1);
dv(1,1)=v(1,1)-v(end,1);
dv(2:end,1)=v(2:end,1)-v(1:end-1,1);
A(n,1) = abs(i(1:N/2)'*dv(1:N/2))+abs(i(N/2:end)'*dv(N/2:end));
L(n,1) = (sum(sqrt((di).^2+(dv).^2)))';
```

To plot the area a logarithmical conversion was used, just as before.

5.2.3 – Area and length versus frequency

Using the methodology explained before, various values of current and voltage of the emulated memristor were sampled with a frequency varying between 6 Hz and 1.3 kHz. The results are plotted in Figure 5.13 below, for a mean of 16 samples.

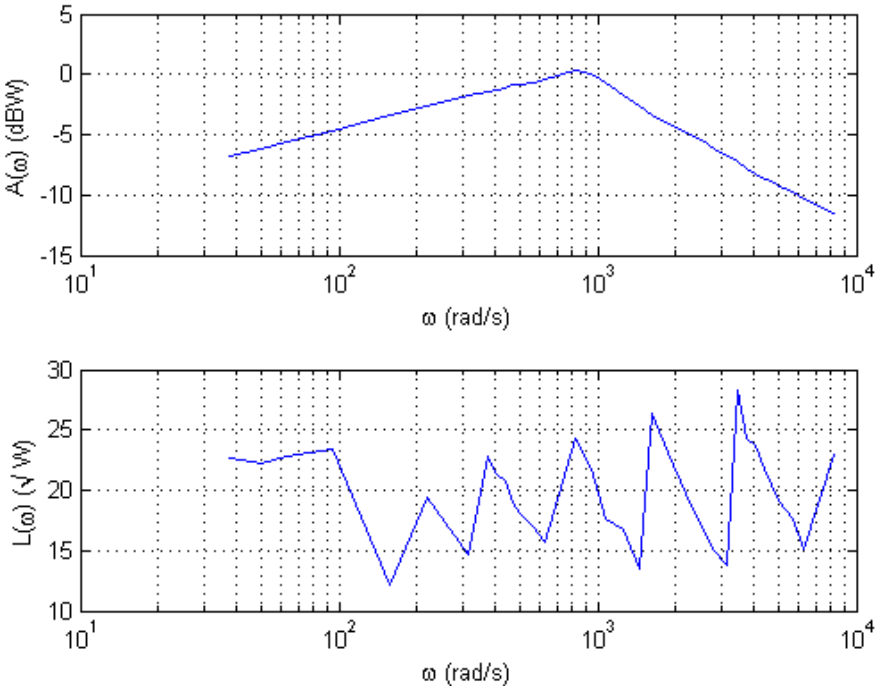


Figure 5.13: Area and length of the emulator's v-i curves for a frequency sweep and a mean of 16 samples.

Although the area has a progression within expectations, with a growing phase, similar to the results of window functions with thresholds, a peak and a steady decline after the peak, the results of the length do not follow the predicted pattern. This behaviour was not entirely unexpected, because the voltage and current from the emulator have much more noise than the curves from the MATLAB simulation. This noise translates into tiny variations to the v-i loop, that when added up increase the length significantly. These variations however, are so small that they do not contribute significantly to the shape of the loop, causing them to have a reduced effect on the area. A good example of the noise's effect on the length can be seen in Figure 5.14.

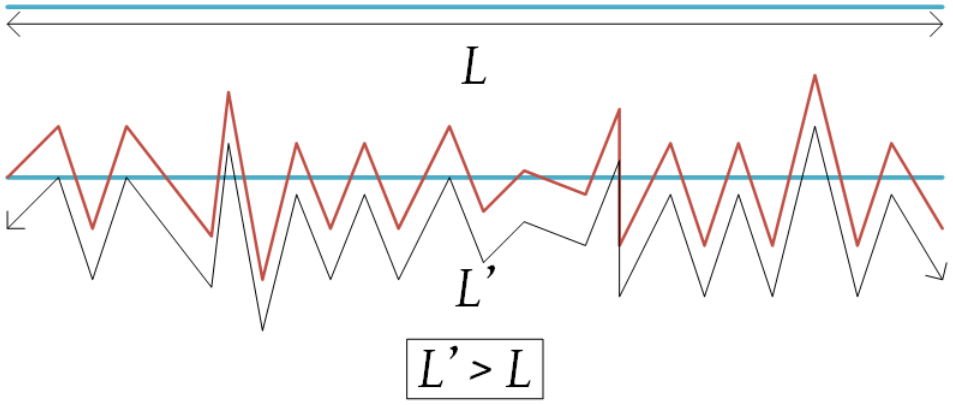


Figure 5.14: Example of the noise's contribution for the measurement of the length of a line segment.

As shown in Figure 5.14, a line segment measured with no noise has a length *L*. If the length of the same line segment is measured, but with tiny irregularities introduced by noise, the same measurement returns the value *L'*, which is bigger than *L*. This noise can be caused by many external factors, from electrical noise in the circuit to the precision of the oscilloscope's samples.

To attempt to mitigate the effects of the noise, the number of samples for voltage and current curves was increased to 512, with the results plotted in Figure 5.15.

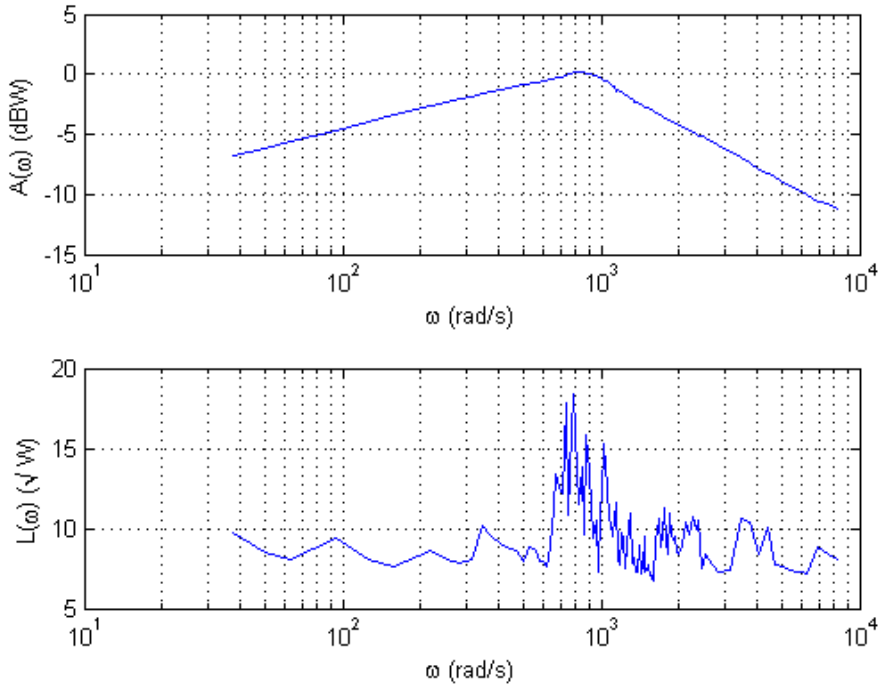


Figure 5.15: Area and length of the emulator's v-i curves for a frequency sweep and a mean of 512 samples.

As can be seen in Figure 5.15, an increased number of samples used in the mean did not have any effect in the end result of the loop's area. The loop's length was affected by the increased precision of the oscilloscope's results, but besides an overall reduction of the length values, they continued to have a chaotic behaviour. This means that a mean of 512, although beneficial, is not enough to produce satisfactory results.

As a final note to the frequency analysis, the full hard-switching behaviour was not observed for the range of tested frequencies. The closest behaviour was observed at $200\pi\ rad/s$ (Figure 5.11).

5.2.4 – Area and length versus amplitude

Using the same methods again, the test of the v-i loops' area and length for a sweep of the input signal's amplitude was conducted, for the emulated memristor, with the results present in Figure 5.16, for a mean of 16 samples of voltage and current. During this test the frequency of the input signal was kept at $200\pi\ rad/s$.

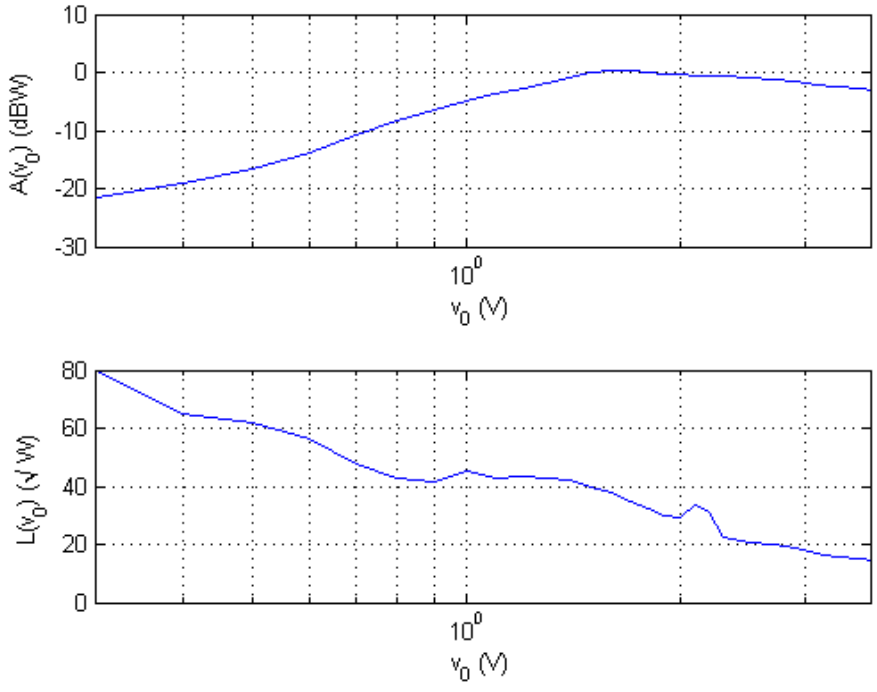


Figure 5.16: Area and length of the emulator's v-i curves for a frequency sweep and a mean of 16 samples.

The area graphic shows a progression similar to most theoretical models studied, a sharp incline, followed by a peak value and a decline similar to the one observed in window functions with thresholds. The length graphic shows an erratic behaviour, like in the previous test. This behaviour was caused again by the noise in the sampled signals reported previously. However, unlike before, the noise did not cause a chaotic measurement, but made the length plot steadily decline from the lowest input amplitude to the highest. This behaviour suggests that lower amplitudes are more susceptible to noise.

Just like before, the areas and lengths of the v-i loops were obtained for values resulting from the mean of 512 samples, with the results in Figure 5.17.

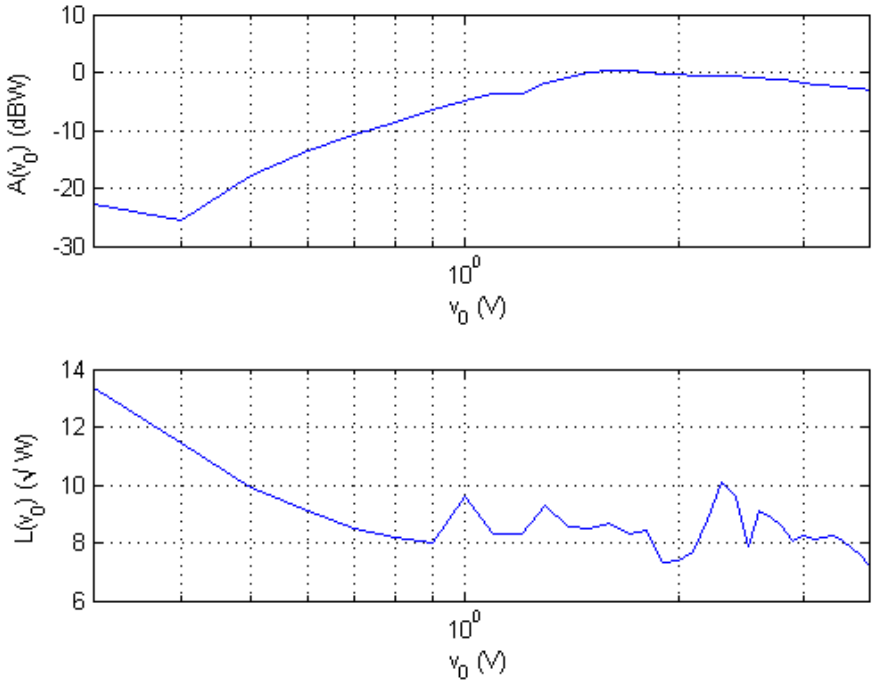


Figure 5.17: Area and length of the emulator's v-i curves for a frequency sweep and a mean of 512 samples.

The results of the variation of the number of values sampled are similar to the test for the frequency sweep. The area showed no noticeable changes, while the length presented an overall reduction of value, but the behaviour remained the same.

Unlike in the test of the frequency sweep, in this one the full hard-switching behaviour was observed from 2.5 V to 3.7 V. An example of this behaviour can be seen in Figure 5.18.

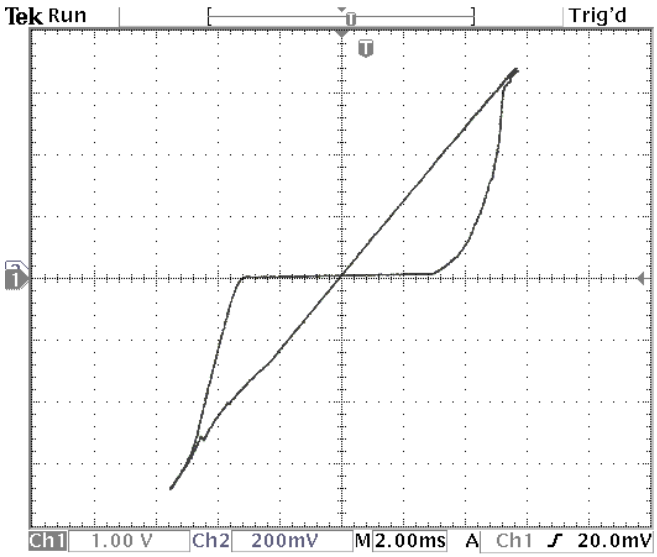


Figure 5.18: Emulator's v-i loop at $100\ Hz$ and $3.5\ V$ with hard-switching behaviour.

5.2.5 – Final remarks

After testing the response of the emulator, it is possible to state that the results are quite satisfactory. The tests concluded that the method to obtain the figures of merit is experimentally valid. However, there are two limiting factors.

The first one is the time required to perform the tests. The voltage and current have to be individually sampled, transferred to the computer and treated before the information of loop's area and length can be retrieved. The time required is increased by the need to perform the mean 512 consecutive values and wait for the stabilization of the results before being retrieved from the oscilloscope.

The second limiting factor is the noise present in the samples. Although the effect of the noise in the area calculation was tolerable, the effects on the length were disastrous, ruining the possibility of retrieving at least two of the figures of merit in each test, the settling value (ω_{set}/v_{set}) and the rate of decay (r_d). Effects of the noise can be mitigated by using an oscilloscope with better resolution, by increasing the number of samples required to calculate the mean value read from the oscilloscope or perform curve fitting using polynomials or splines. However, such options require either more expensive equipment, or even more time to run the tests.

Chapter VI

Conclusions

After terminating the work that served as the basis for this dissertation several conclusions can be drawn. First of all, the field of memristors and memristive devices has had an exciting and accelerated development in the last decade, that bursted with the HP's announcement in 2008 of the development of a physical device. This increased development is interesting and necessary for the emergence of commercial memristive devices. However the rapid appearance of new mathematical models, some of them putting the validity of others in question, the continuous opinions of disbelief and disagreement and the fact that no commercial devices have yet not arisen, make a study in this area difficult and any conclusions volatile.

Despite these problems, an analysis of some of the current mathematical models provided in the literature, showed that, except for a few cases, they have a fairly stable and expected behaviour in the determination of the areas and lengths of their v-i curves. This behaviour made possible the extraction of values of frequency and amplitude in moments of peak, settling and rate of decay. The stable evolution of these values while the internal characteristics of the model were changing, allowed their usage as classification attributes of the models of MDs used, making them prime candidates to characterize in a similar way future physical devices. Given this conclusion, it is safe to assume that the main goal of this dissertation was met with success.

The attempt to apply similar analysis techniques to the memristor emulator proved challenging. First of all, each emulator as a behaviour that depends on how it is build. Most of them are impossible to change, or even record the internal characteristics of the MD that they emulate, making them useful to emulate a single memristor.

Second, the precision that results from sampling values from a physical circuit compared with the precision of the values in a MATLAB simulation is smaller and the added noise can be significant. This led to v-i curves that are more difficult to analyse, as was observed in the erratic values of the loop lengths calculated, causing some figures of merit to become irretrievable.

In a final note, despite the success to obtain some figures of merit usable to classify theoretical MD models, their retrieval from a physical MD emulator was not entirely successful. The difficulty to obtain exact values of length in an acceptable timeframe, make the retrieval of some of the figures of merit very difficult, putting their validity in question. Nonetheless, it is possible to affirm that the work was completed with relative success.

6.1 – Future work

Although this work had a promising start, there are many shortcomings that can be corrected and new avenues to expand in.

One of the problems noticed is that not all the figures of measure proposed can be applied to the models studied. As was discussed, models that use window functions with conditional transitions (like the Biolek and BCM functions) have area and length curves that follow a different overall shape. This difference of shape is so that two of the proposed figures of merit: rate of decay (r_d) and settling value (ω_{set} and v_{set}) cannot be applied to them. So, further study of these models is needed, to understand them better and to find other figures that can better describe them.

In the vein of the previous proposal, the studies realized in this work only covered mathematical models of the TiO_2 MD. There are other types of MD devices in existence, each with different behaviours and characteristics. Because of these differences, the figures of merit proposed for the TiO_2 MD might not be valid for others. In light of these, studies of mathematical models that cover other types of MDs are necessary. A starting point might be the Yakopcic and TEAM models, because of their capacity to model different MDs, as reported in chapter 3.

Further improvements can be realized in the model analysis in MATLAB. The current method of finding the memristance of a MD for a certain input signal consists in an algorithm based upon the forward Euler method. Although this method is simple and produces results it is not without shortcomings. Its main problem comes from the fact that it is numerically unstable for stiff equations. This problem became evident in some measurement and even though a workaround solution was found, it is not an ideal situation. To attempt to solve the problem of instability associated with this method others can be used, namely Runge-Kutta methods or the MATLAB's own functions to solve ODEs.

Finally, it might be possible to improve the method of extraction of the current and voltage values from the terminals of the MD emulator. Although the presented method worked well to find the v-i loop area, it was not so successful in the discovery of the loop's length. Possible solutions to this problem might pass by the usage of an oscilloscope with greater resolution, capable of performing the mean of more than 512 samples, by applying a band-pass filter for the frequency of the input signal, or perform curve fitting of the voltage and current results using splines or polynomials.

Bibliography

[1] L. O. Chua, "Memristor-The Missing Circuit Element," *IEEE Trans. Circuit Theory*, vol. 18, no. 5, pp. 507–519, 1971.

[2] 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, May 2008.

[3] D. Johnson, "The Memristor's Fundamental Secrets Revealed," *IEEE Spectrum*, 2013. [Online]. Available: http://spectrum.ieee.org/nanoclast/semiconductors/nanotechnology/the-memristors-fundamental-secrets-revealed. [Accessed: 21-Nov-2013].

[4] J. C. Duarte, E. V. Martins, and L. N. Alves, "Frequency Characterization of Memristive Devices," in *2013 21st European Conference on Circuit Theory and Design (ECCTD 2013)*, 2013, pp. 1–4.

[5] J. C. Duarte, E. V. Martins, and L. N. Alves, "Amplitude Characterization of Memristive Devices," in *2013 IEEE International Conference on Electronics, Circuits, and Systems (ICECS 2013)*, 2013, pp. 1–4.

[6] T. W. Hickmott, "Potential Distribution and Negative Resistance in Thin Oxide Films," *J. Appl. Phys.*, vol. 35, no. 9, pp. 2679–2689, 1964.

[7] F. Argall, "Switching phenomena in titanium oxide thin films," *Solid. State. Electron.*, vol. 11, no. 5, pp. 535–541, 1968.

[8] G. Dearnaley, A. M. Stoneham, and D. V. Morgan, "Electrical phenomena in amorphous oxide films," *Reports Prog. Phys.*, vol. 33, no. 3, pp. 1129–1191, Sep. 1970.

[9] L. O. Chua and S.-M. Kang, "Memristive Devices and Systems," *Proc. IEEE*, vol. 64, no. 2, pp. 209–223, 1976.

[10] S. Thakoor, A. Moopenn, T. Daud, and a. P. Thakoor, "Solid-state thin-film memistor for electronic neural networks," *J. Appl. Phys.*, vol. 67, no. 6, pp. 3132–3135, 1990.

[11] F. A. Buot and A. K. Rajagopal, "Binary information storage at zero bias in quantum-well diodes," *J. Appl. Phys.*, vol. 76, no. 9, pp. 5552–5560, 1994.

[12] a. Beck, J. G. Bednorz, C. Gerber, C. Rossel, and D. Widmer, "Reproducible switching effect in thin oxide films for memory applications," *Appl. Phys. Lett.*, vol. 77, no. 1, pp. 139–141, 2000.

[13] L. D. Bozano, B. W. Kean, V. R. Deline, J. R. Salem, and J. C. Scott, “Mechanism for bistability in organic memory elements,” *Appl. Phys. Lett.*, vol. 84, no. 4, pp. 607–609, 2004.

[14] V. Erokhin and M. P. Fontana, “Electrochemically controlled polymeric device: a memristor (and more) found two years ago,” *eprint arXiv:0807.0333*, pp. 1–11, Jul. 2008.

[15] T. Prodromakis, B. P. Peh, C. Papavassiliou, and C. Toumazou, “A Versatile Memristor Model With Nonlinear Dopant Kinetics,” *IEEE Trans. Electron Devices*, vol. 58, no. 9, pp. 3099–3105, Sep. 2011.

[16] Y. N. Joglekar and S. J. Wolf, “The elusive memristor: properties of basic electrical circuits,” *Eur. J. Phys.*, vol. 30, no. 4, pp. 661–675, Jul. 2009.

[17] Y. V. Pershin and M. Di Ventra, “Experimental demonstration of associative memory with memristive neural networks.,” *Neural Netw.*, vol. 23, no. 7, pp. 881–886, Sep. 2010.

[18] H. Kim, M. P. Sah, C. Yang, S. Cho, and L. O. Chua, “Memristor Emulator for Memristor Circuit Applications,” *IEEE Trans. Circuits Syst. I Regul. Pap.*, vol. 59, no. 10, pp. 2422–2431, Oct. 2012.

[19] R. Mutlu and E. Karakulak, “A methodology for memristance calculation,” *Turkish J. Electr. Eng. Comput. Sci.*, pp. 1–24.

[20] S. Vongehr, “The Missing Memristor: Novel Nanotechnology or rather new Case Study for the Philosophy and Sociology of Science?,” *Adv. Sci. Lett.*, vol. 17, no. 1, pp. 285–290, Oct. 2012.

[21] S. P. Adhikari, M. P. Sah, H. Kim, and L. O. Chua, “Three Fingerprints of Memristor,” *IEEE Trans. Circuits Syst. I Regul. Pap.*, vol. 60, no. 11, pp. 3008–3021, Nov. 2013.

[22] G. Pazienza and J. Albo-Canals, “Teaching Memristors to EE Undergraduate Students [Class Notes],” *IEEE Circuits Syst. Mag.*, vol. 11, no. 4, pp. 36–44, 2011.

[23] Y. V. Pershin and M. Di Ventra, “Memory effects in complex materials and nanoscale systems,” *Adv. Phys.*, vol. 60, no. 2, pp. 145–227, Apr. 2011.

[24] T. Driscoll, H.-T. Kim, B.-G. Chae, M. Di Ventra, and D. N. Basov, “Phase-transition driven memristive system,” *Appl. Phys. Lett.*, vol. 95, no. 4, p. 043503 1–3, 2009.

[25] Y. V. Pershin and M. Di Ventra, “Spin memristive systems: Spin memory effects in semiconductor spintronics,” *Phys. Rev. B*, vol. 78, no. 11, pp. 113309–1–113309–4, Sep. 2008.

[26] D. B. Strukov and R. S. Williams, “Four-dimensional address topology for circuits with stacked multilayer crossbar arrays.,” *Proc. Natl. Acad. Sci. U. S. A.*, vol. 106, no. 48, pp. 20155–20158, Dec. 2009.

[27] K. Eshraghian, K. Cho, O. Kavehei, S.-K. Kang, D. Abbott, and S. S. Kang, “Memristor MOS Content Addressable Memory (MCAM): Hybrid Architecture for Future High Performance Search Engines,” *IEEE Trans. Very Large Scale Integr. Syst.*, vol. 19, no. 8, pp. 1407–1417, Aug. 2011.

[28] "Memristor - Panasonic Japan prepares to release ReRam Nonvolatile Memory with Microprocessor Evaluation Kit," *Memristor.org*. [Online]. Available: http://www.memristor.org/electronics/807/panasonic-reram-nonvolatile-memory-microprocessor-kit. [Accessed: 28-Oct-2013].

[29] E. Lehtonen and M. Laiho, "Stateful implication logic with memristors," in *2009 IEEE/ACM International Symposium on Nanoscale Architectures*, 2009, pp. 33–36.

[30] D. Strukov and K. Likharev, "CMOL FPGA circuits.," in *The 2006 International Conference on Computer Design (CDES'06)*, 2006.

[31] G. S. Snider and R. S. Williams, "Nano/CMOS architectures using a field-programmable nanowire interconnect," *Nanotechnology*, vol. 18, no. 3, p. 035204 1–11, Jan. 2007.

[32] S. Shin, K. Kim, and S.-M. Kang, "Memristor-Based Fine Resolution Programmable Resistance and Its Applications," in *International Conference on Communications, Circuits and Systems, 2009*, 2009, pp. 948–951.

[33] Y. V. Pershin and M. Di Ventra, "Teaching Memory Circuit Elements via Experiment-Based Learning," *IEEE Circuits Syst. Mag.*, vol. 12, no. 1, pp. 64–74, Jan. 2012.

[34] T. Driscoll, J. Quinn, S. Klein, H. T. Kim, B. J. Kim, Y. V. Pershin, M. Di Ventra, and D. N. Basov, "Memristive adaptive filters," *Appl. Phys. Lett.*, vol. 97, no. 9, p. 093502 1–3, 2010.

[35] B. Linares-barranco and T. Serrano-gotarredona, "Memristance can explain Spike-Time-Dependent-Plasticity in Neural Synapses," *Nat. Preced.*, pp. 1–4, 2009.

[36] S. H. Jo, T. Chang, I. Ebong, B. B. Bhadviya, P. Mazumder, and W. Lu, "Nanoscale memristor device as synapse in neuromorphic systems.," *Nano Lett.*, vol. 10, no. 4, pp. 1297–301, Apr. 2010.

[37] R. Williams, "How We Found The Missing Memristor," *IEEE Spectr.*, vol. 45, no. 12, pp. 28–35, Dec. 2008.

[38] M. D. Pickett, D. B. Strukov, J. L. Borghetti, J. J. Yang, G. S. Snider, D. R. Stewart, and R. S. Williams, "Switching dynamics in titanium dioxide memristive devices," *J. Appl. Phys.*, vol. 106, no. 7, p. 074508 1–6, 2009.

[39] M. D. Pickett, D. B. Strukov, J. L. Borghetti, J. J. Yang, G. S. Snider, D. R. Stewart, and R. S. Williams, "Switching Dynamics in Titanium Dioxide Memristive Devices - Supplementary Material," *J. Appl. Phys.*, vol. 106, no. 7, pp. 1–10, 2009.

[40] C. Yakopcic, T. M. Taha, G. Subramanyam, R. E. Pino, and S. Rogers, "A Memristor Device Model," *IEEE Electron Device Lett.*, vol. 32, no. 10, pp. 1436–1438, Oct. 2011.

[41] S. Kvatinsky, E. G. Friedman, A. Kolodny, and U. C. Weiser, "TEAM: ThrEshold Adaptive Memristor Model," *IEEE Trans. Circuits Syst. I Regul. Pap.*, vol. 60, no. 1, pp. 211–221, Jan. 2013.

[42] Z. Biolek, D. Biolek, and V. Biolková, "SPICE Model of Memristor with Nonlinear Dopant Drift," *Radioengineering*, vol. 18, no. 2, pp. 210–214, 2009.

[43] F. Corinto and A. Ascoli, "A Boundary Condition-Based Approach to the Modeling of Memristor Nanostructures," *IEEE Trans. Circuits Syst. I Regul. Pap.*, vol. 59, no. 11, pp. 2713–2726, Nov. 2012.

[44] A. Ascoli, F. Corinto, V. Senger, and R. Tetzlaff, "Memristor Model Comparison," *IEEE Circuits Syst. Mag.*, vol. 13, no. 2, pp. 89–105, 2013.

[45] "Euler method," *Wikipedia, the free encyclopedia*. [Online]. Available: http://en.wikipedia.org/wiki/Euler_method. [Accessed: 17-Nov-2013].

[46] Y. V Pershin and M. Di Ventra, "Practical Approach to Programmable Analog Circuits With Memristors," *IEEE Trans. Circuits Syst. I Regul. Pap.*, vol. 57, no. 8, pp. 1857–1864, Aug. 2010.

[47] S. Shin, L. Zheng, G. Weickhardt, S. Cho, and S.-M. "Steve" Kang, "Compact Circuit Model and Hardware Emulation for Floating Memristor Devices," *IEEE Circuits Syst. Mag.*, vol. 13, no. 2, pp. 42–55, 2013.

Annex

Window functions details

The following pages have samples of the area and length graphics for the voltage-current loops, in the second period of the input voltage, from the tested window function of the Strukov and Williams TiO_2 MD model. Below them are voltage-current loops taken at specific point of the test. Firstly are frequency sweep tests of the input voltage. Secondly the amplitude sweep tests of the same input voltage. The parameters used for the model were: $D = 10\,nm$, $w_0 = D/2\ nm$, $R_{on} = 1\ k\Omega$, $R_{off} = 100\ k\Omega$, $\eta = 1$ and $\mu_v = 1 \times 10^{-10}\ cm^2/Vs$.

Frequency sweeps

The input voltage used for these tests was defined by the function,

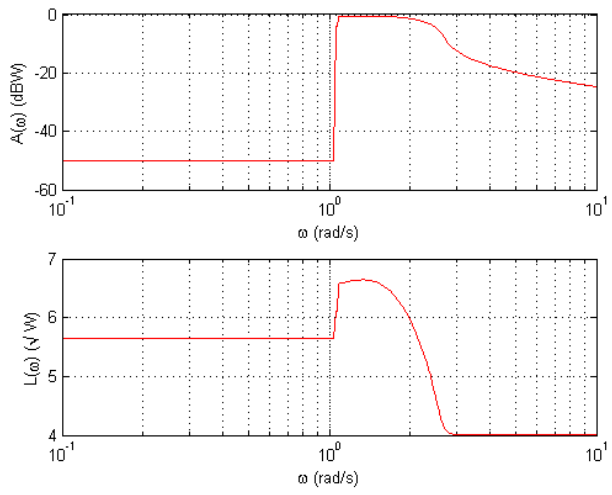
$$v(t) = v_0 \sin(\omega t)$$

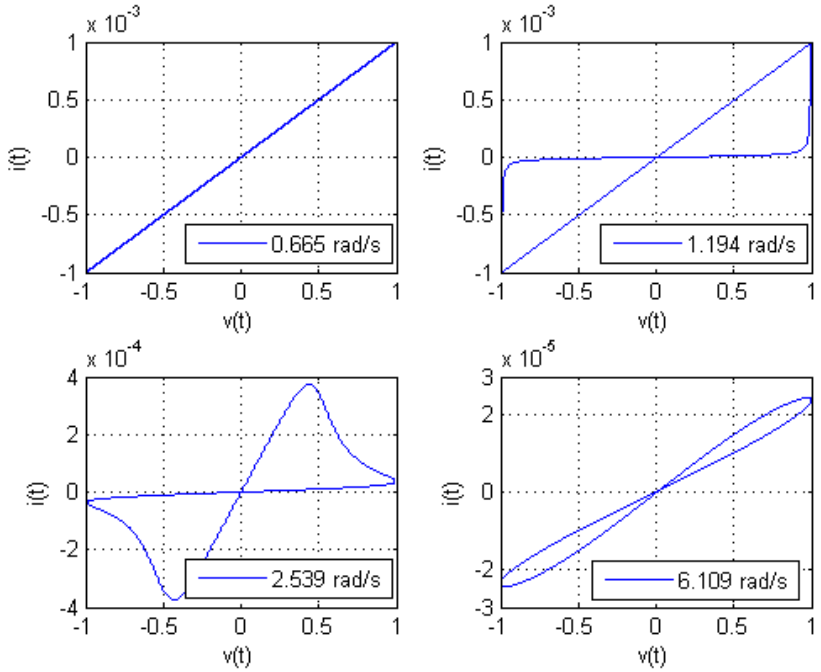
Where ω is the frequency of the signal, 300 values logarithmically spaced, $v_0 = 1\,V$ is the amplitude and t is the time vector, 2×10^6 values equally spaced amounting for the time of the first 2 periods of $v(t)$.

HP window function

This window function has no additional parameters.

This sweep was performed from $10^{-1}\ rad/s$ to $10^{1}\ rad/s$

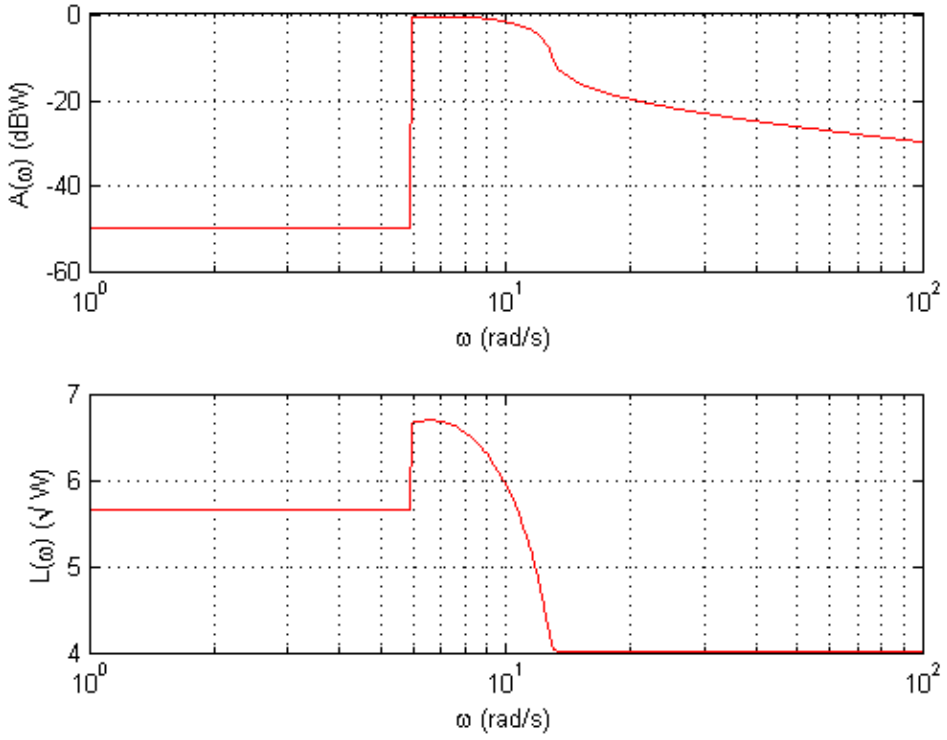


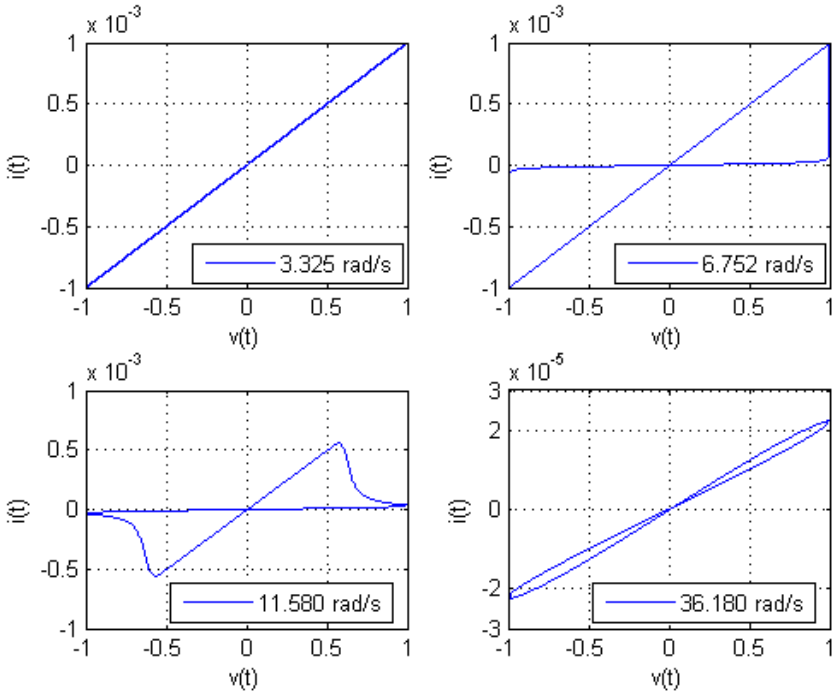


Prodromakis windows function

The window parameters used were $k = 1$ and $p = 10$.

This sweep was performed from $1\ rad/s$ to $10^2\ rad/s$

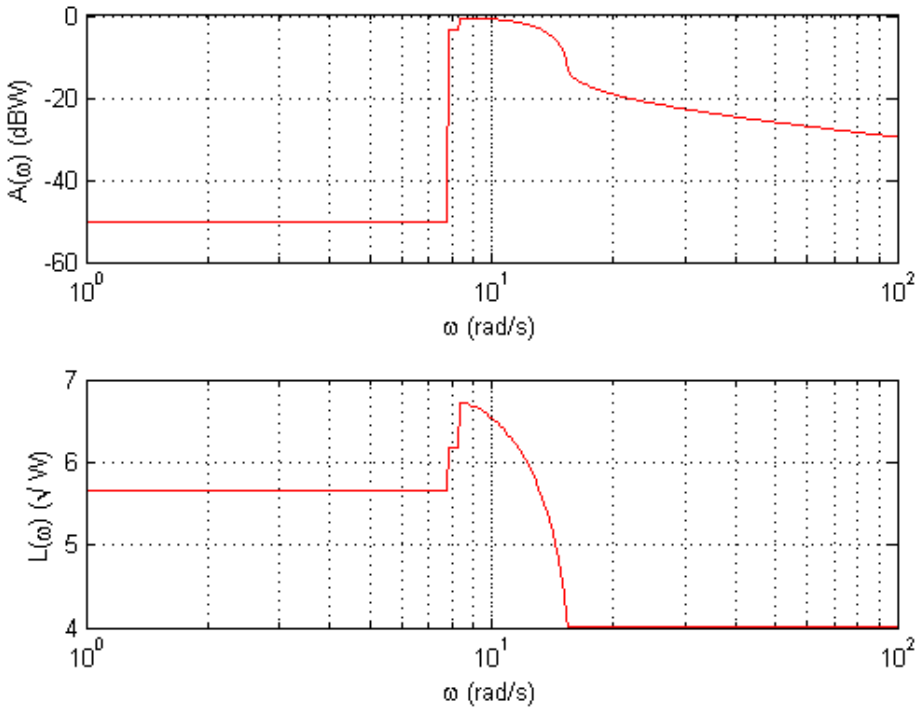


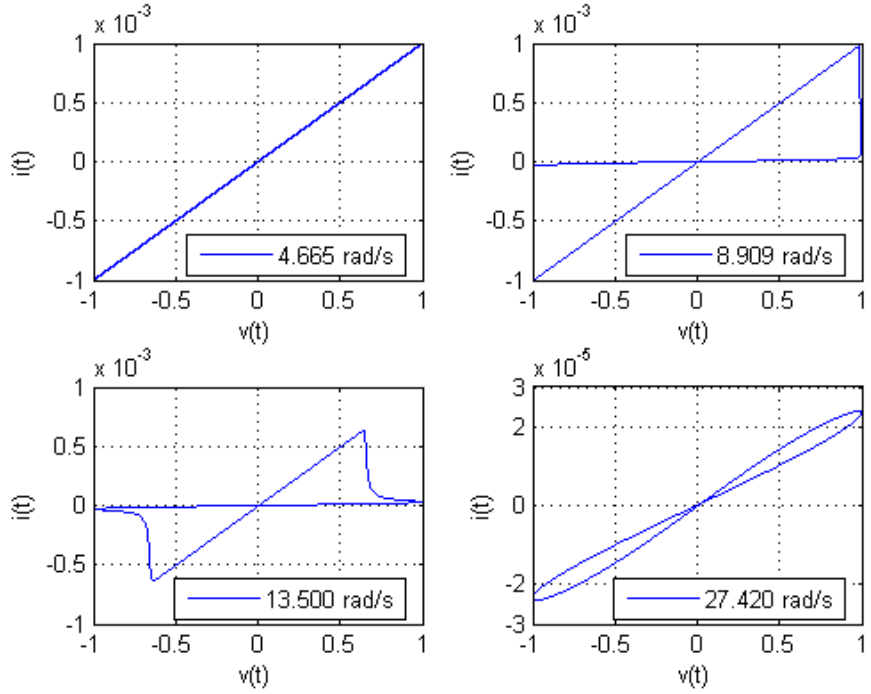


Joglekar window function

The window parameter used was $p = 10$.

This sweep was performed from $1\ rad/s$ to $10^2\ rad/s$

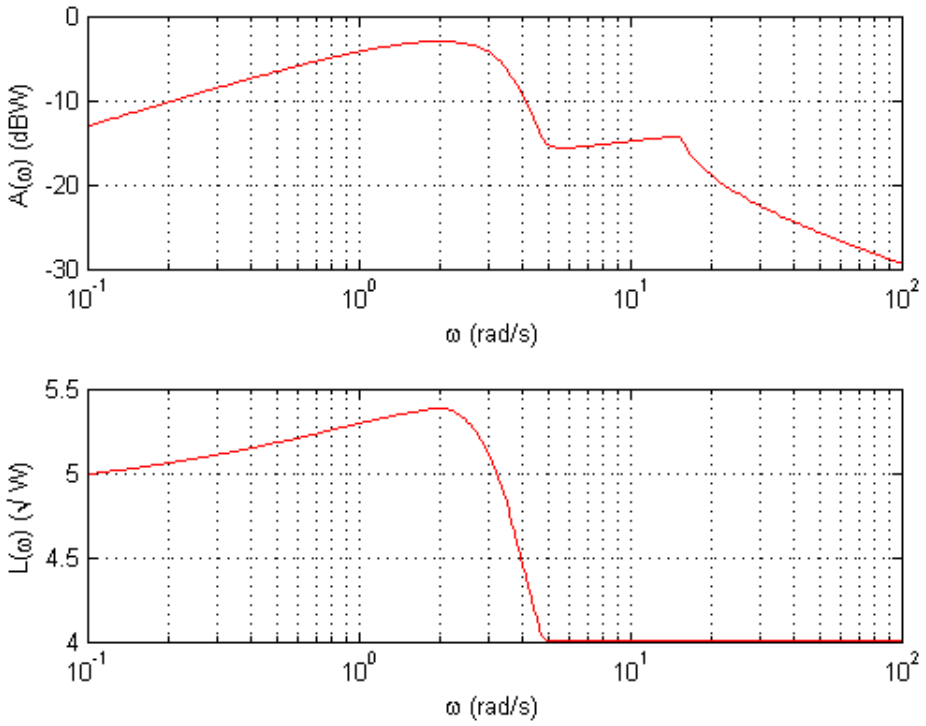


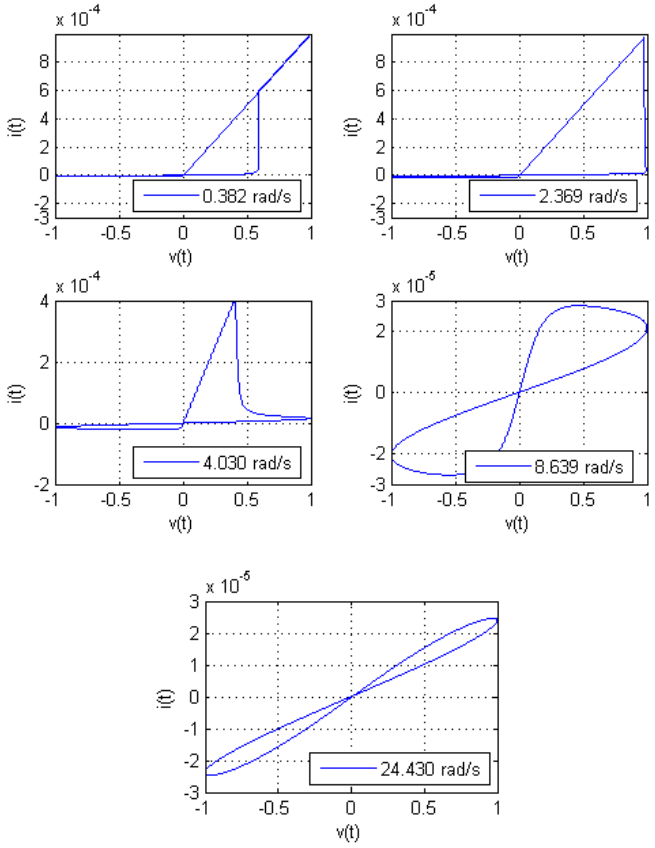


Biolek window function

The window parameter used was $p = 10$.

This sweep was performed from $10^{-1}\ rad/s$ to $10^{2}\ rad/s$

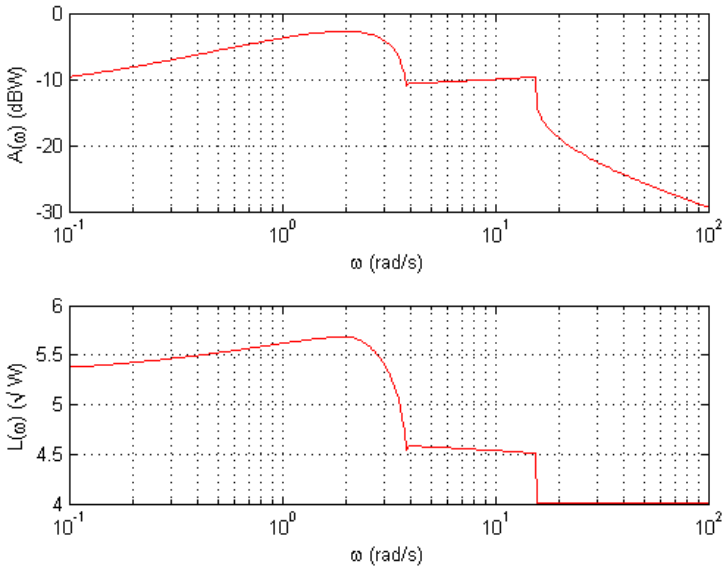


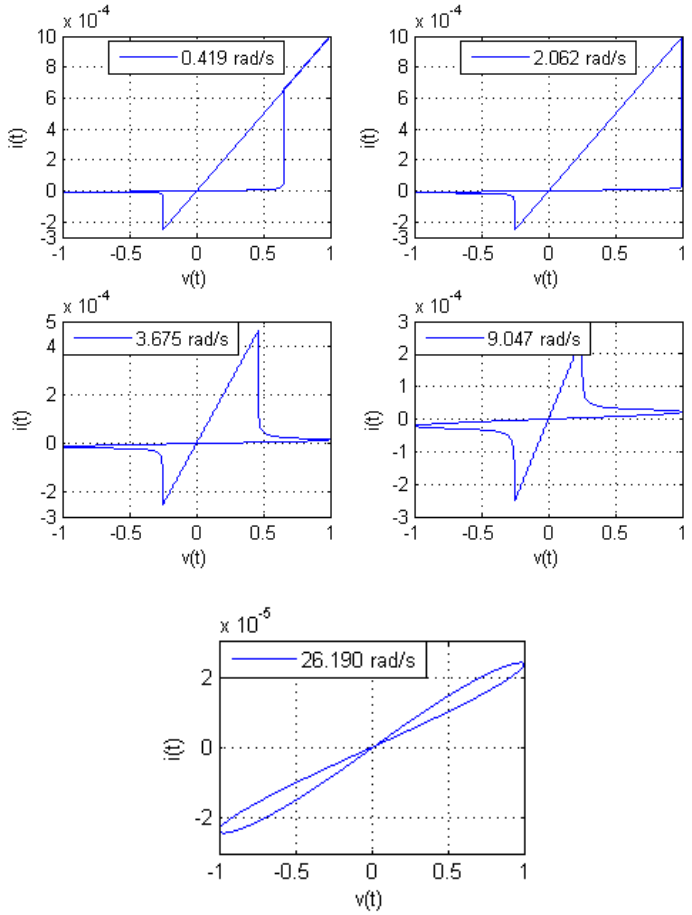


BCM window function

The window parameters used were $v_{th0} = v_0/4$ and $v_{th1} = v_0/4$.

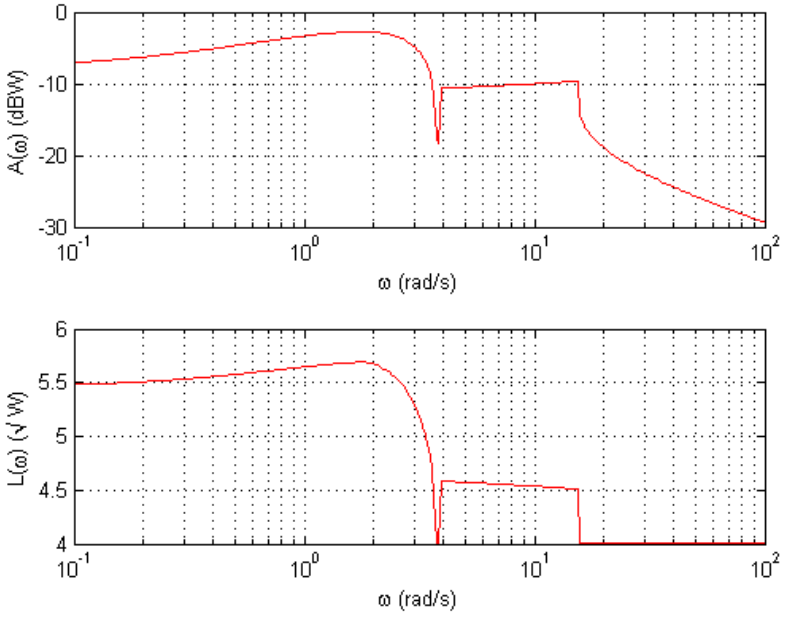
This sweep was performed from $10^{-1}\ rad/s$ to $10^{2}\ rad/s$

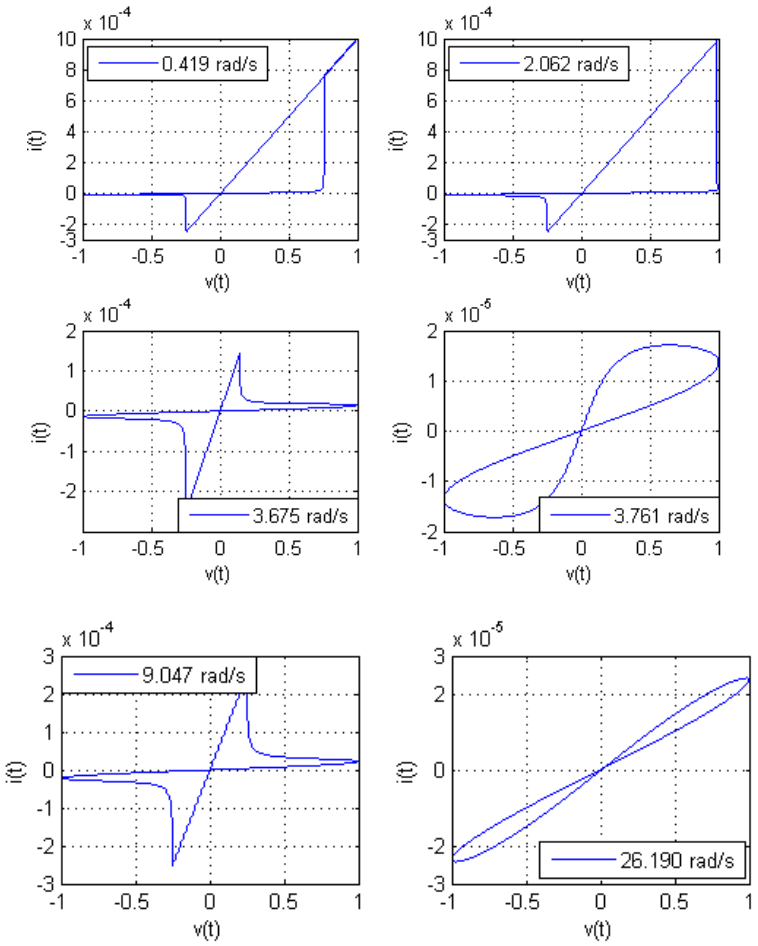




The window parameters used were $v_{th0} = v_0/2$ and $v_{th1} = v_0/4$.

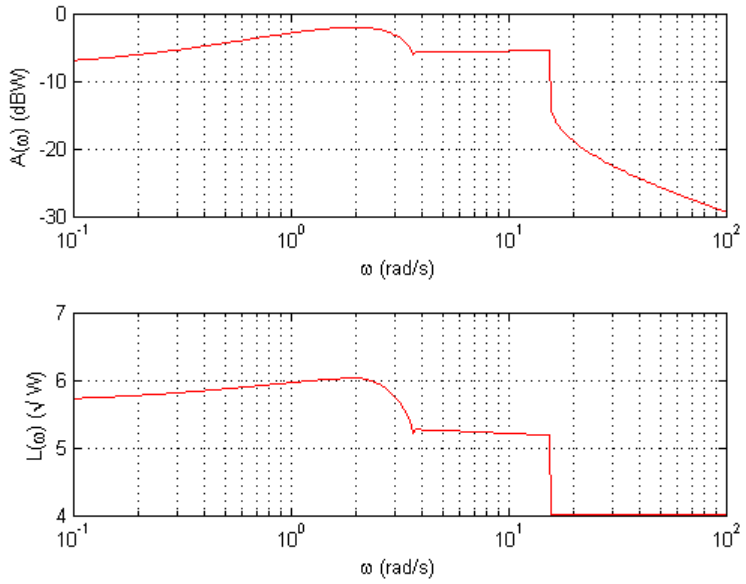
This sweep was performed from $10^{-1}\ rad/s$ to $10^{2}\ rad/s$

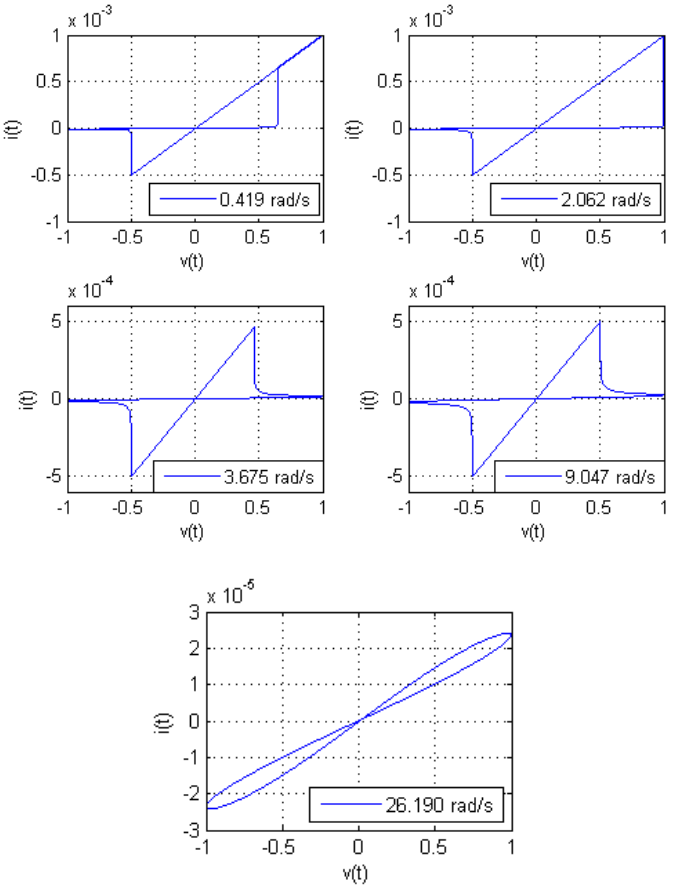




The window parameters used were $v_{th0} = v_0/4$ and $v_{th1} = v_0/2$.

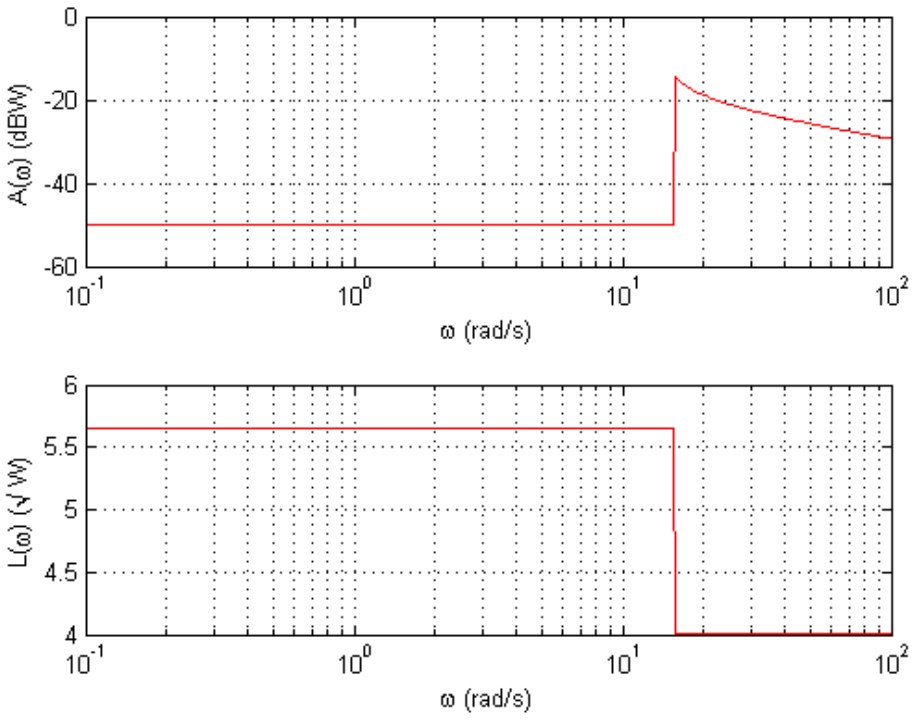
This sweep was performed from $10^{-1}\ rad/s$ to $10^{2}\ rad/s$

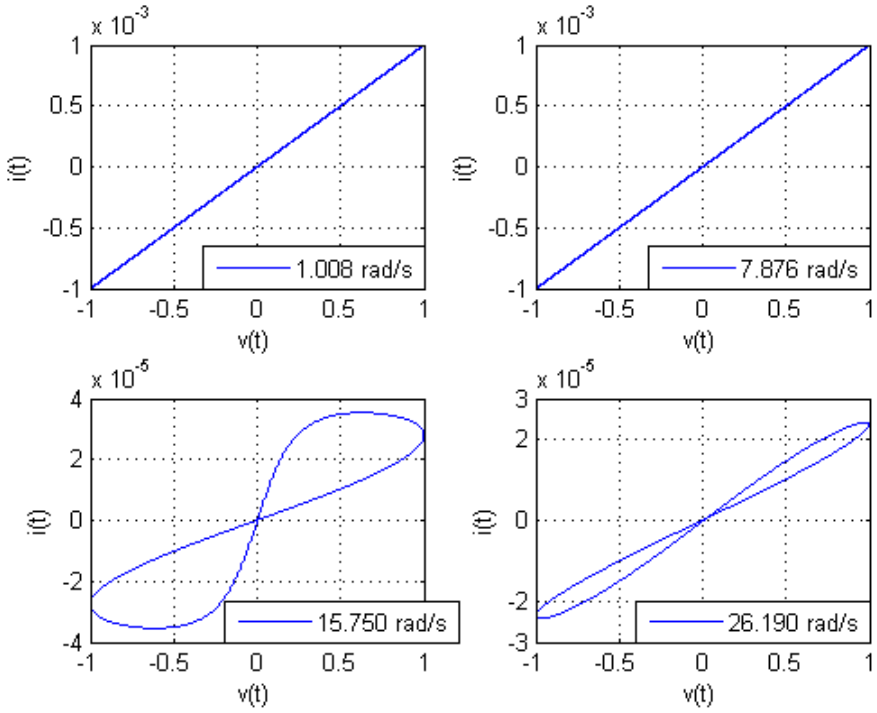




The window parameters used were $v_{th0} = 2 \times v_0$ and $v_{th1} = 2 \times v_0$.

This sweep was performed from $10^{-1}\ rad/s$ to $10^{2}\ rad/s$





Amplitude sweeps

In this section, besides the individual sweeps for each window function, are shown as well the figure of merit measures. There are four measures per window, each varying one of the model parameters, while the others are kept with their standard values.

The input voltage used for these tests was defined by the function,

$$v(t) = v_0 \sin(\omega t)$$

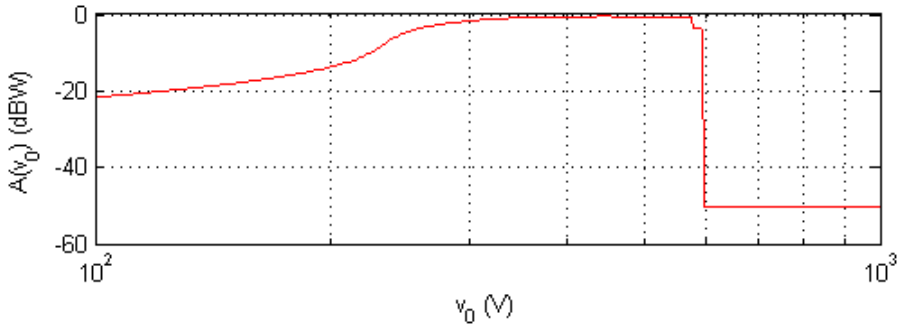
Where v_0 is the amplitude of the signal, 300 values logarithmically spaced, ω is the frequency of the signal and t is the time vector, 2×10^6 values equally spaced amounting for the time of the first 2 periods of $v(t)$. The samples of the voltage-current loops, area and length curves presented here were taken with $\omega = 100\ Hz$. However, for reasons discussed in chapter 4, the measures of the figures of merit were taken with three distinct frequencies: $1\ Hz$, $100\ Hz$ and $1\ kHz$.

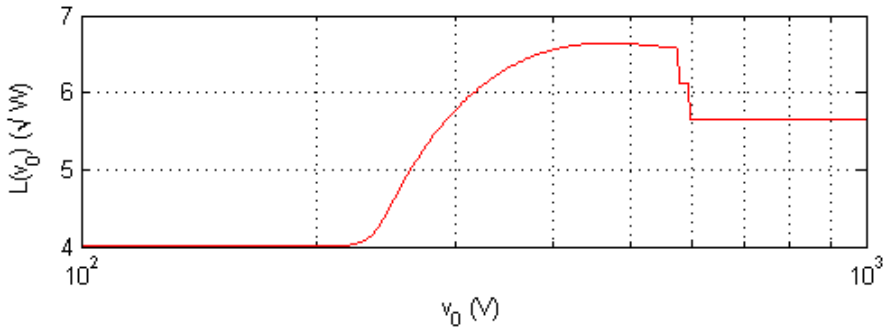
HP window function

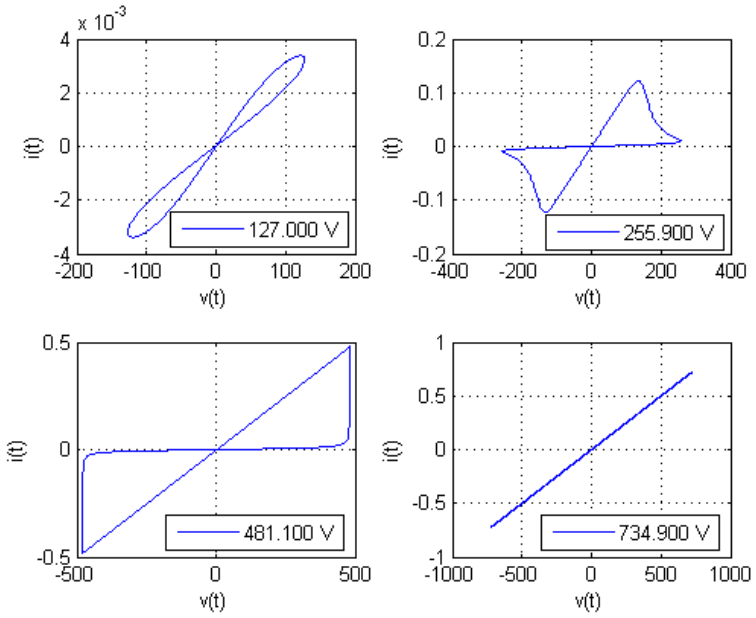
Area and length curves

This window function has no additional parameters.

This sweep was performed from $10^2\ V$ to $10^3\ V$



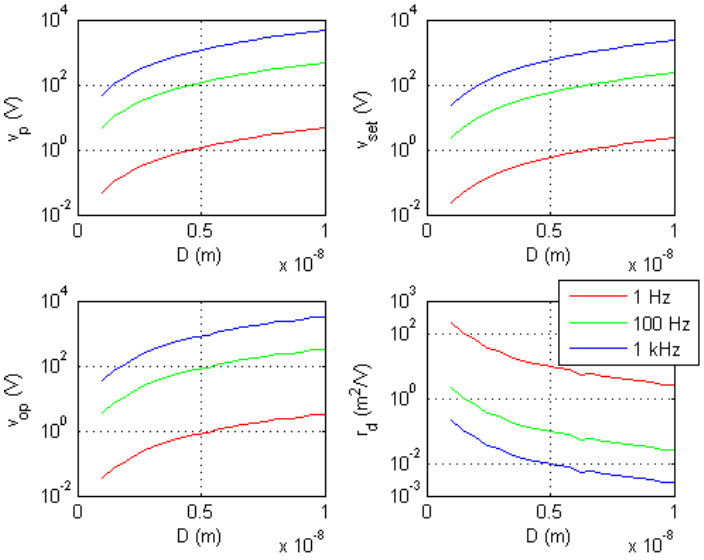




Figures of merit

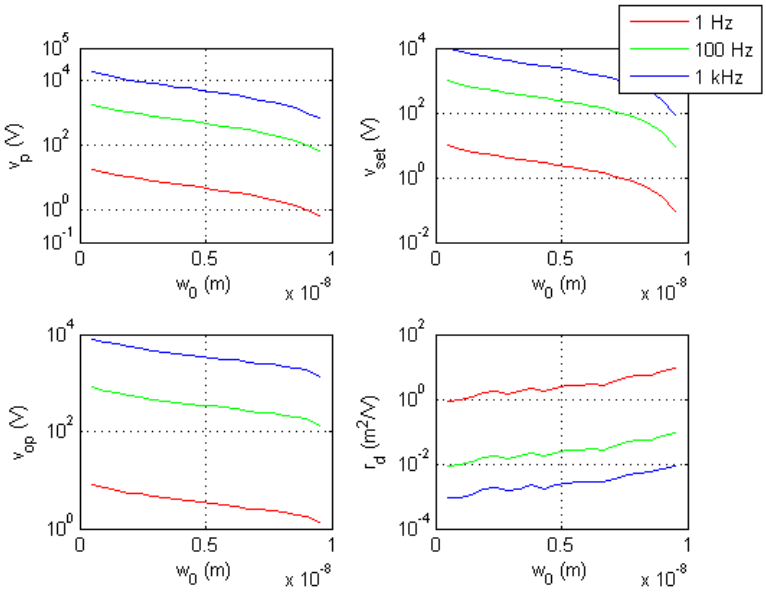
D variable

For these measures *D* took 20 values, from 1 nm to 20 nm.



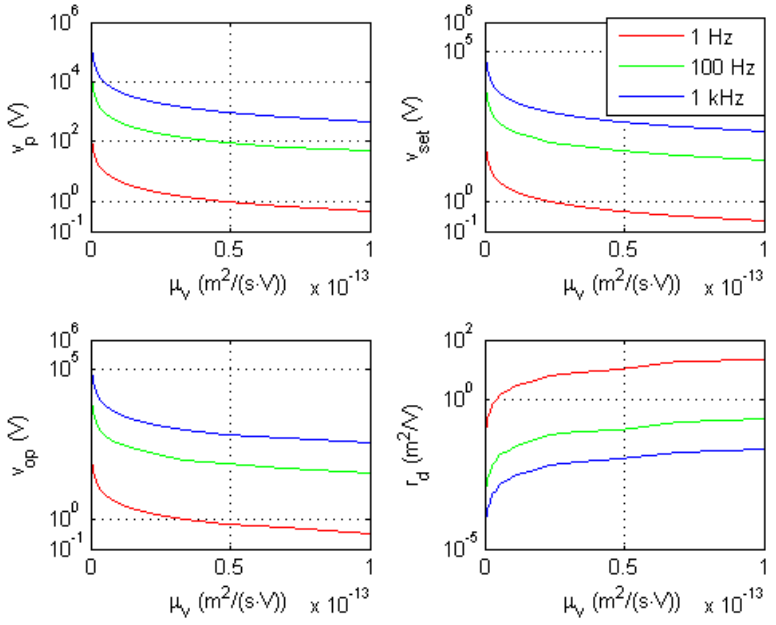
w_0 variable

For these measures w_0 took 20 values, from 5% of *D* to 95% of *D*.



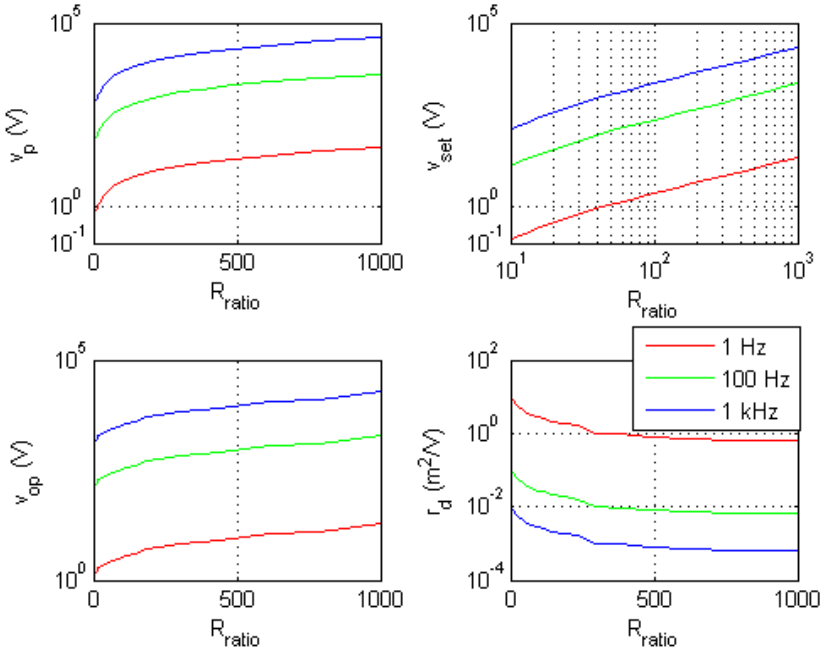
μ_v variable

For these measures μ_v took 20 values, from $1 \times 10^{-12}\ cm^2/Vs$ to $1 \times 10^{-9}\ cm^2/Vs$.



R_{ratio} variable

For these measures the ratio between R_{on} and R_{off} took 20 values, from 10 to 1000.

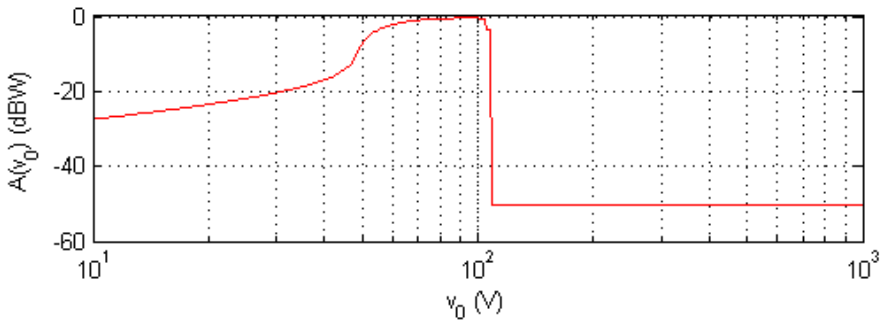


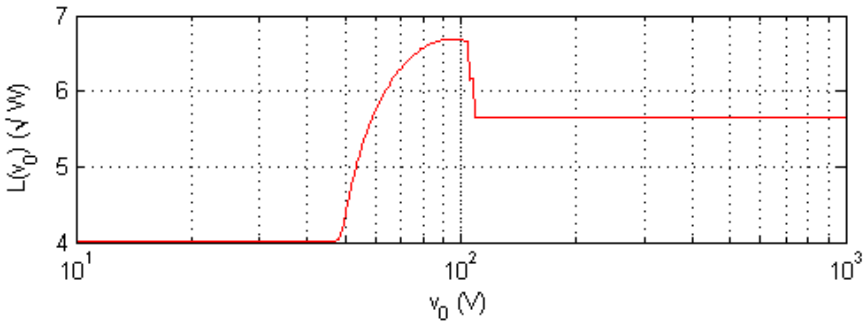
Prodromakis window function

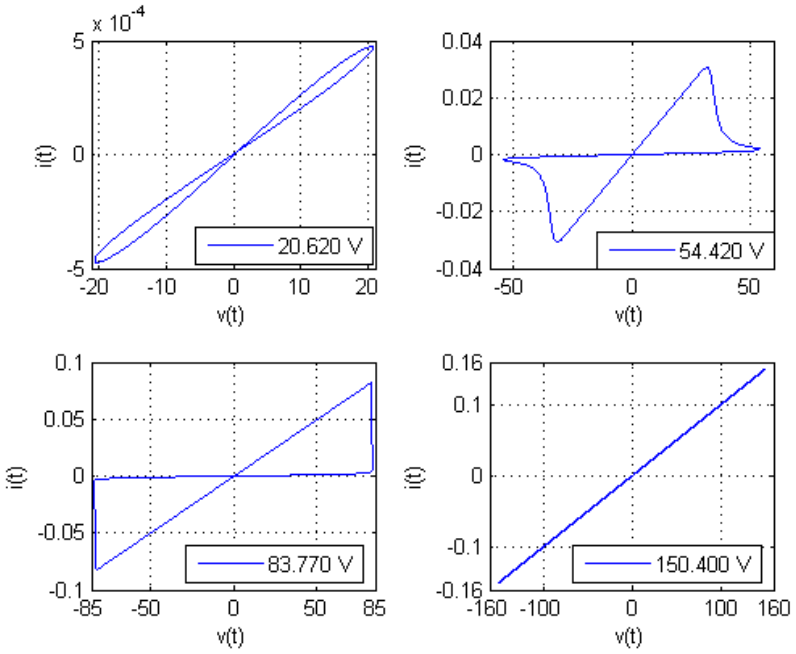
Area and length curves

The window parameters used were $k = 1$and $p = 10$.

This sweep was performed from $10^1\ V$ to $10^3\ V$



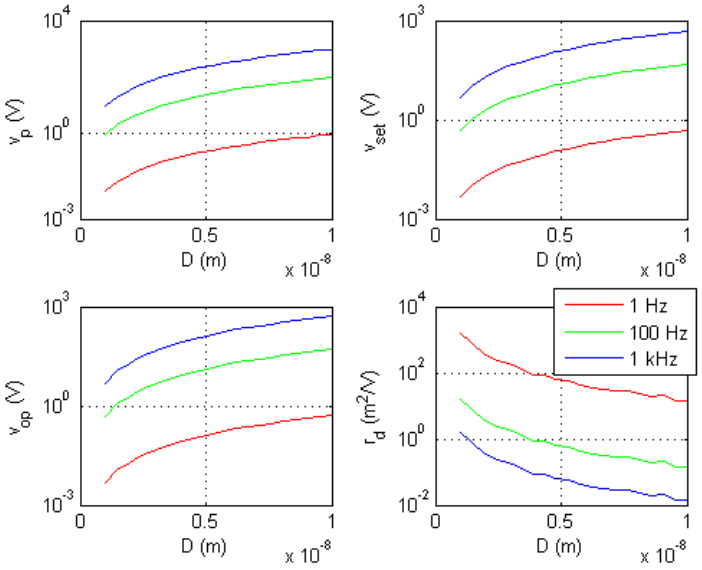




Figures of merit

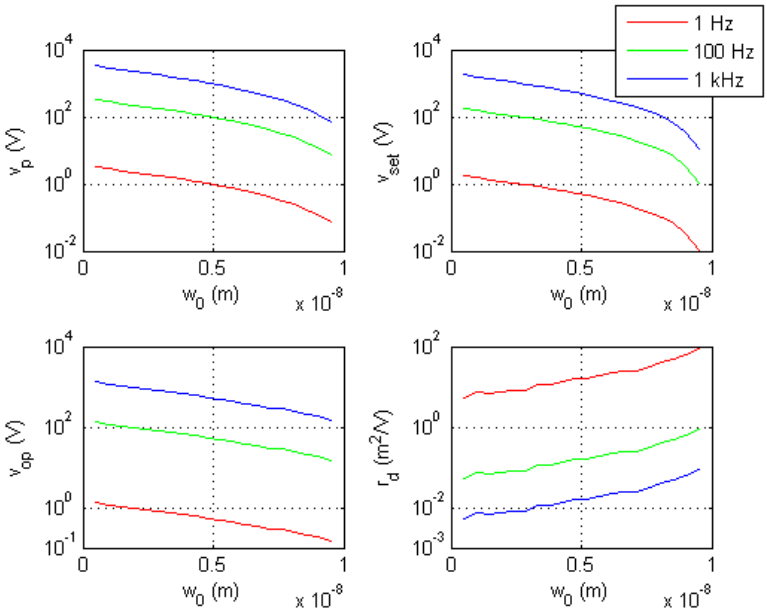
D variable

For these measures *D* took 20 values, from 1 nm to 20 nm.



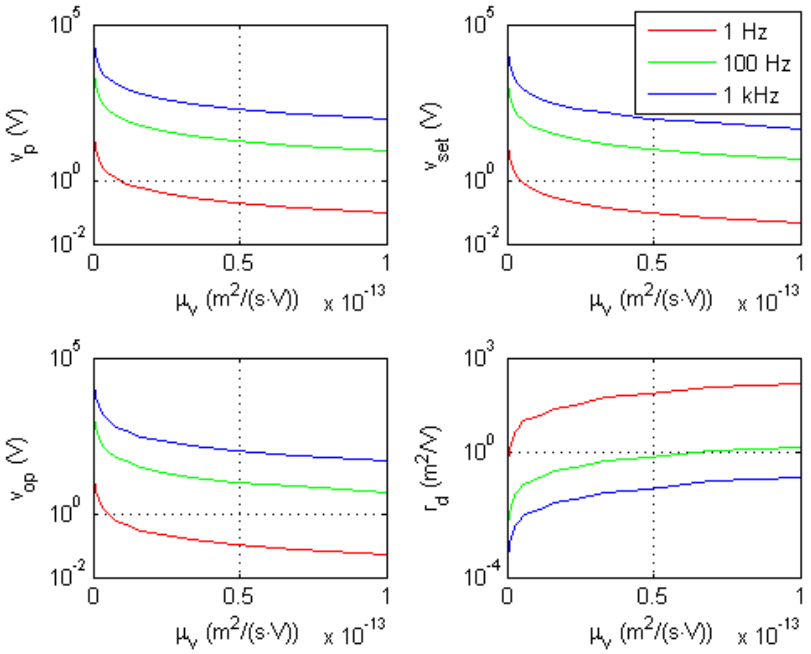
w_0 variable

For these measures w_0 took 20 values, from 5% of *D* to 95% of *D*.



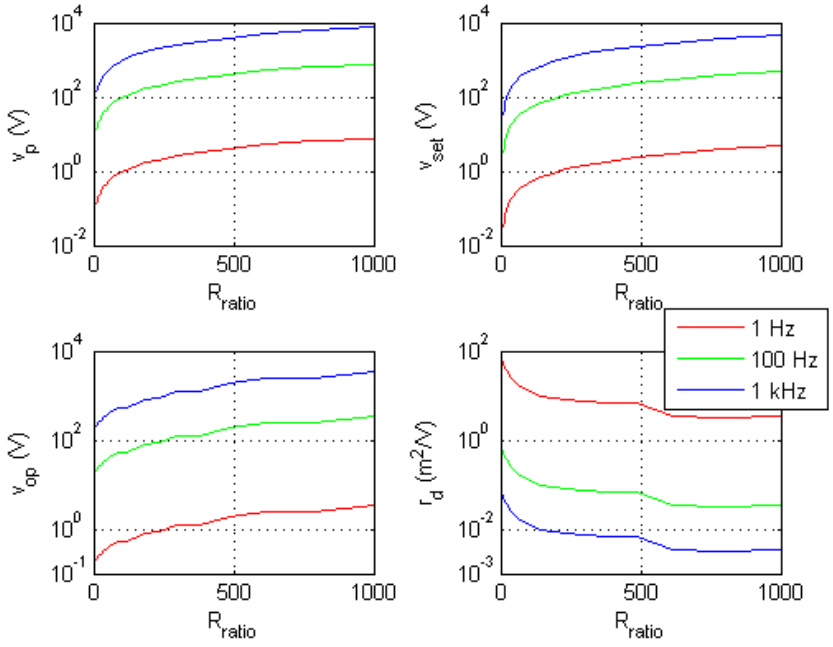
μ_v variable

For these measures μ_v took 20 values, from $1 \times 10^{-12}\ cm^2/Vs$ to $1 \times 10^{-9}\ cm^2/Vs$.



R_{ratio} variable

For these measures the ratio between R_{on} and R_{off} took 20 values, from 10 to 1000.

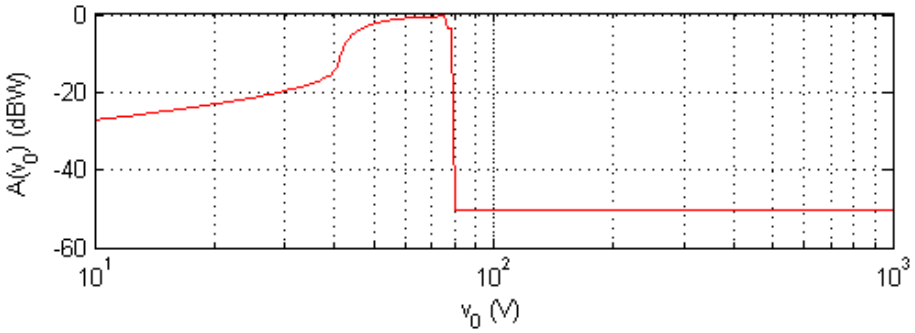


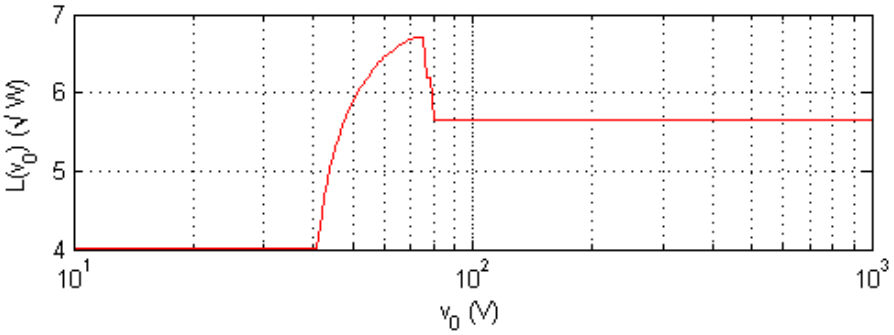
Joglekar window function

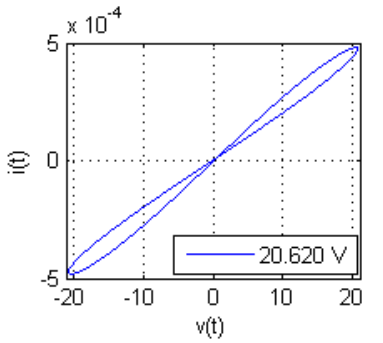
Area and length curves

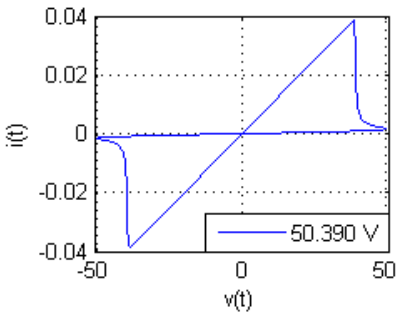
The window parameter used was $p = 10$.

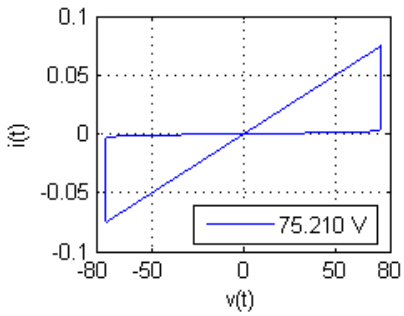
This sweep was performed from $10^1\ V$ to $10^3\ V$

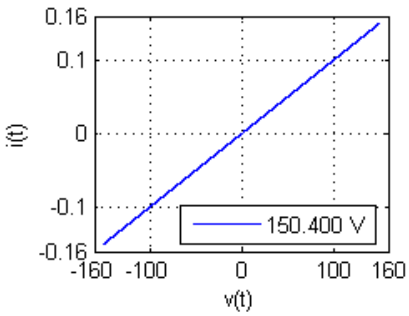








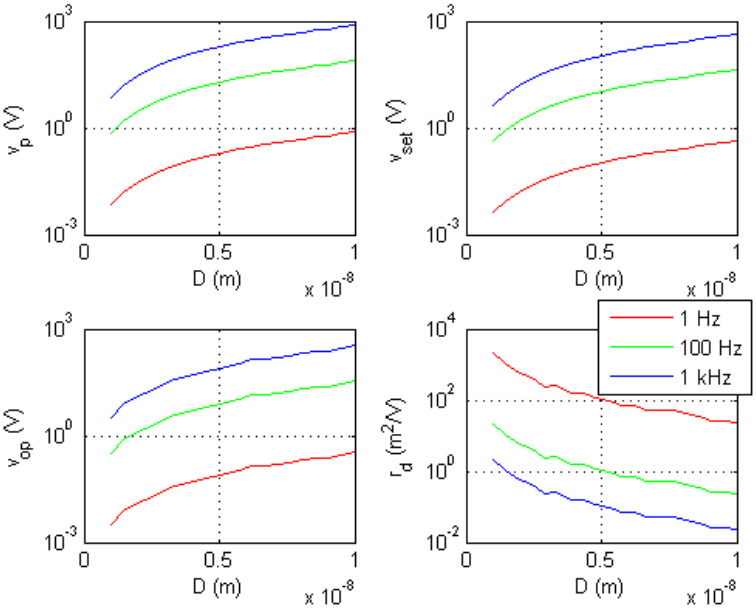




Figures of merit

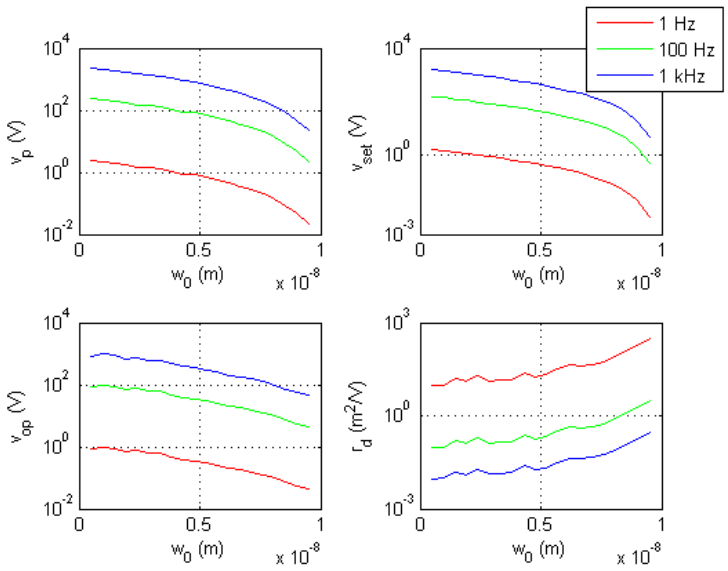
D variable

For these measures *D* took 20 values, from 1 nm to 20 nm.



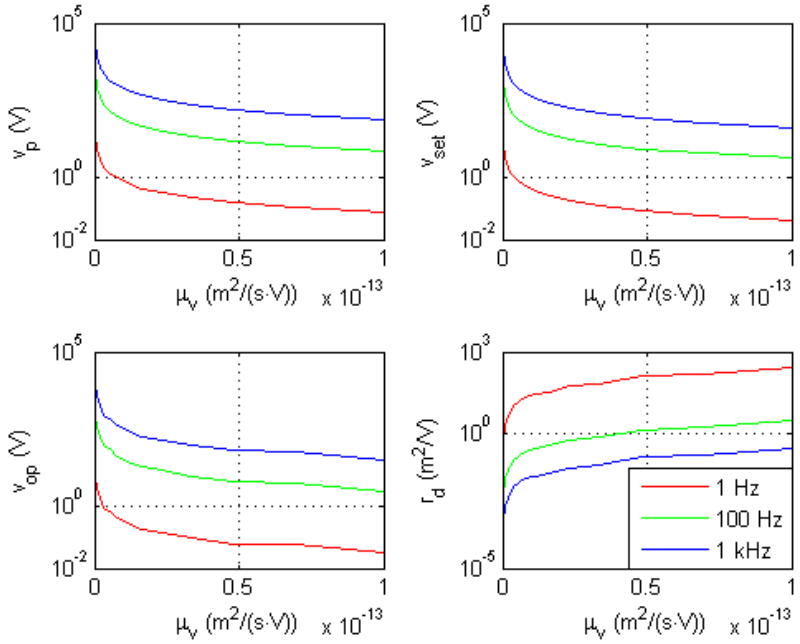
w_0 variable

For these measures w_0 took 20 values, from 5% of *D* to 95% of *D*.



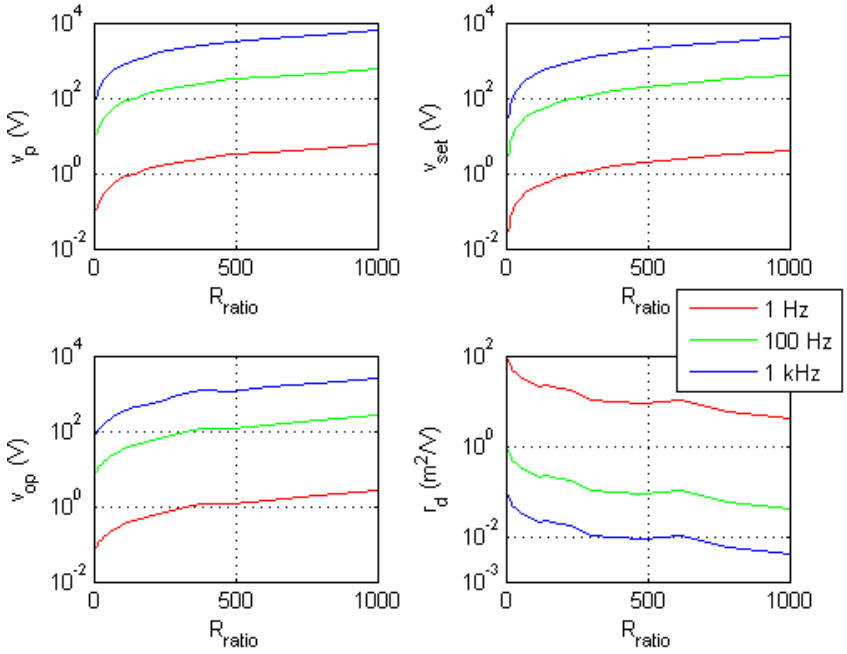
μ_v variable

For these measures μ_v took 20 values, from $1 \times 10^{-12}\ cm^2/Vs$ to $1 \times 10^{-9}\ cm^2/Vs$.



R_{ratio} variable

For these measures the ratio between R_{on} and R_{off} took 20 values, from 10 to 1000.

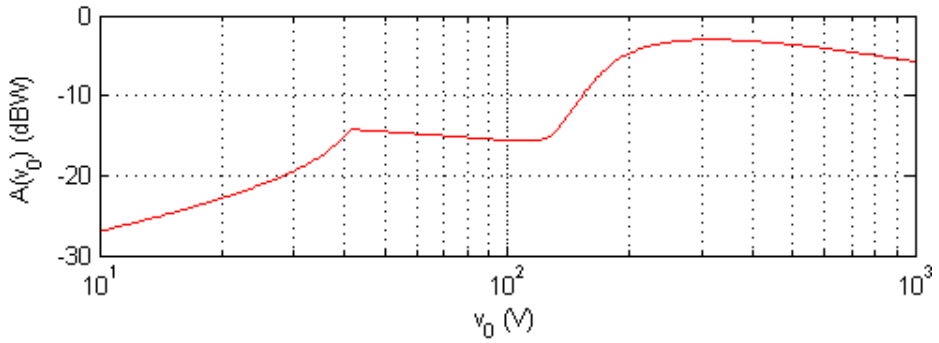


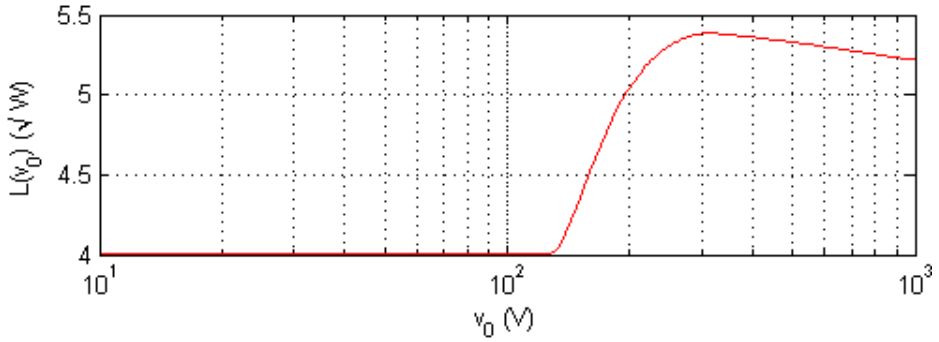
Biolek window function

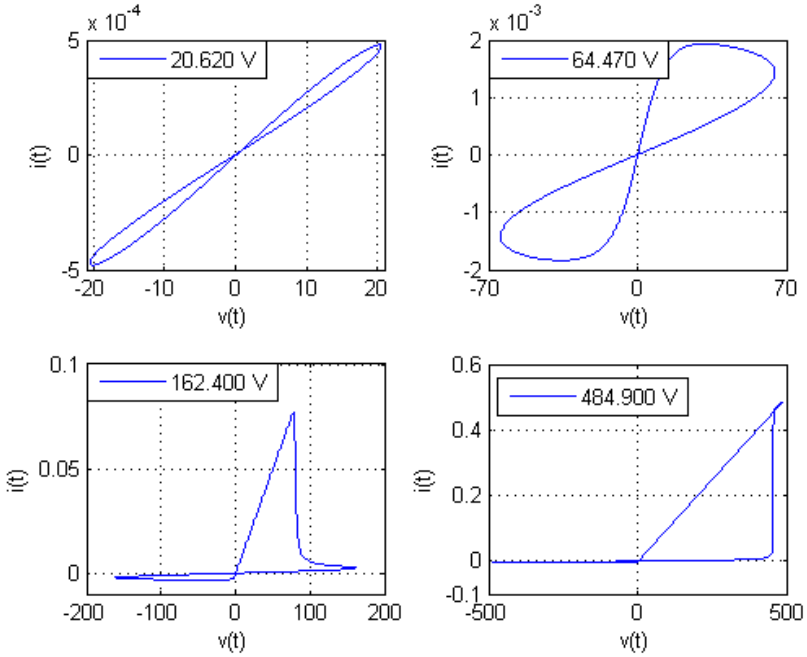
Area and length curves

The window parameter used was $p = 10$.

This sweep was performed from $10^1\ V$ to $10^3\ V$



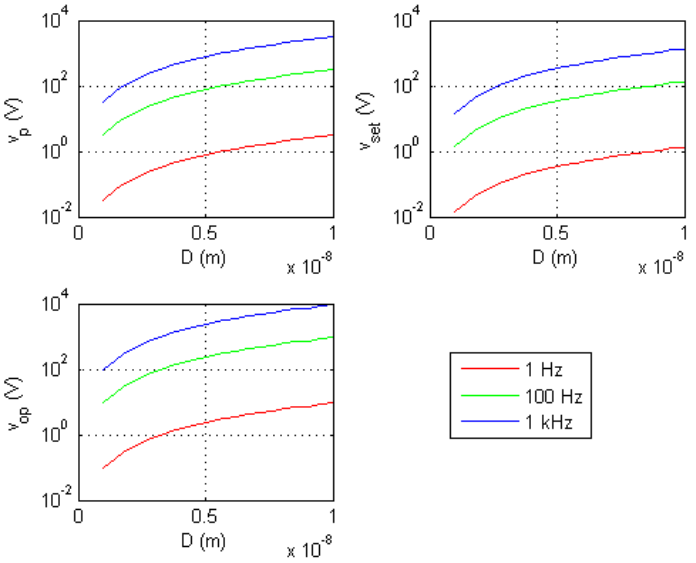




Figures of merit

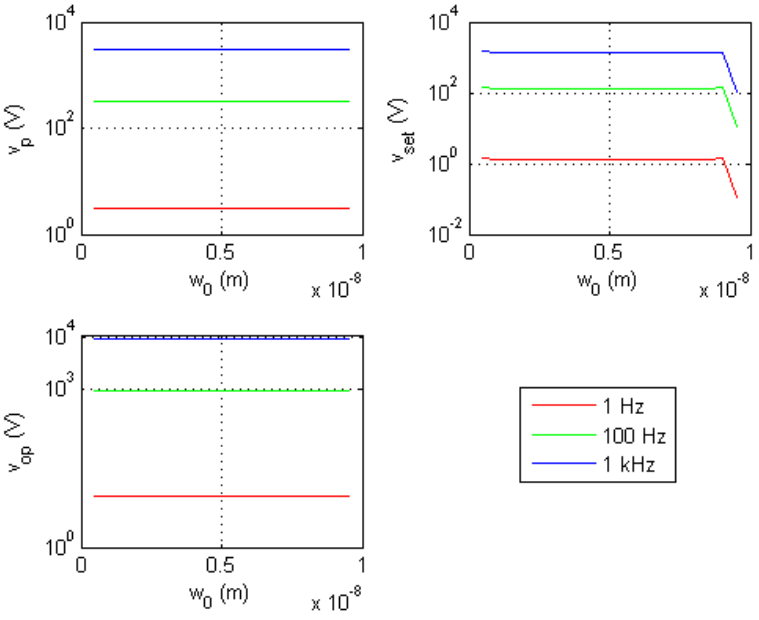
D variable

For these measures *D* took 20 values, from 1 nm to 20 nm.



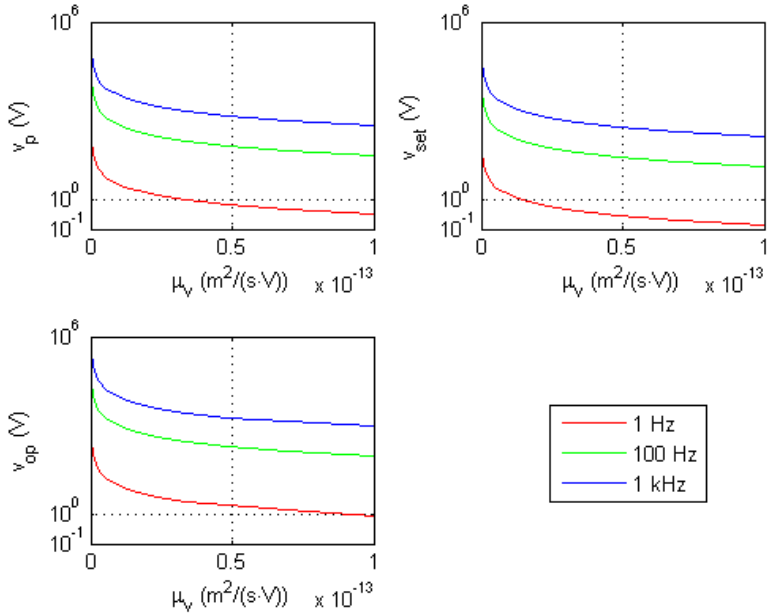
w_0 variable

For these measures w_0 took 20 values, from 5% of *D* to 95% of *D*.



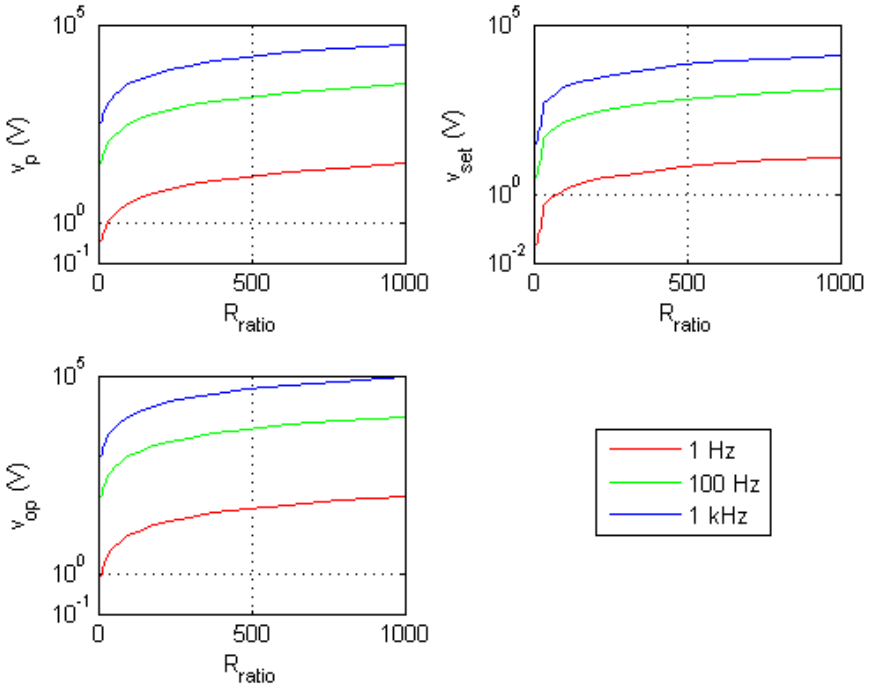
μ_v variable

For these measures μ_v took 20 values, from $1 \times 10^{-12}\ cm^2/Vs$ to $1 \times 10^{-9}\ cm^2/Vs$.



R_{ratio} variable

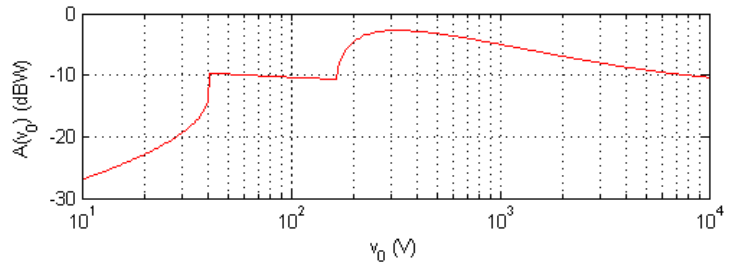
For these measures the ratio between R_{on} and R_{off} took 20 values, from 10 to 1000.

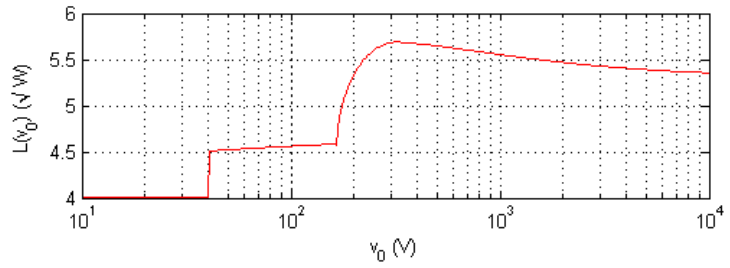


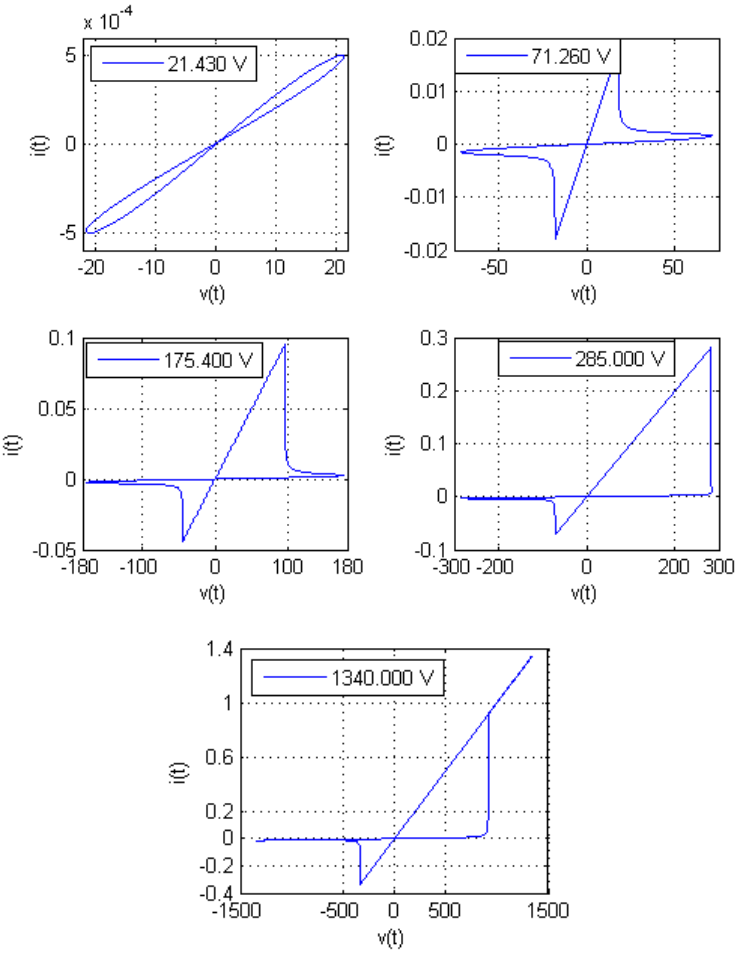
BCM window function

The window parameters used were $v_{th0} = v_0/4$ and $v_{th1} = v_0/4$.

This sweep was performed from $10^{-1}\ V$ to $10^{2}\ rad/s$

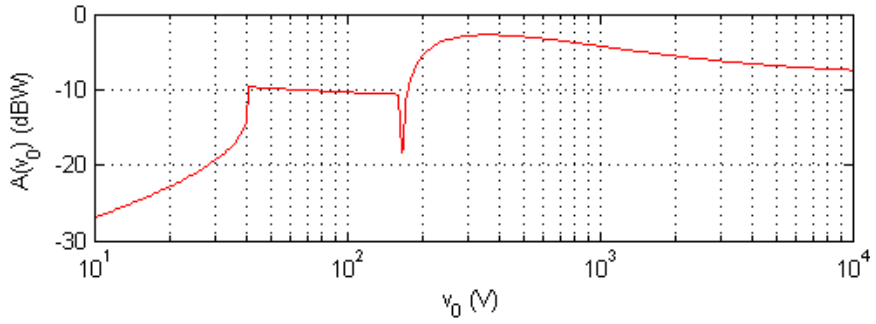


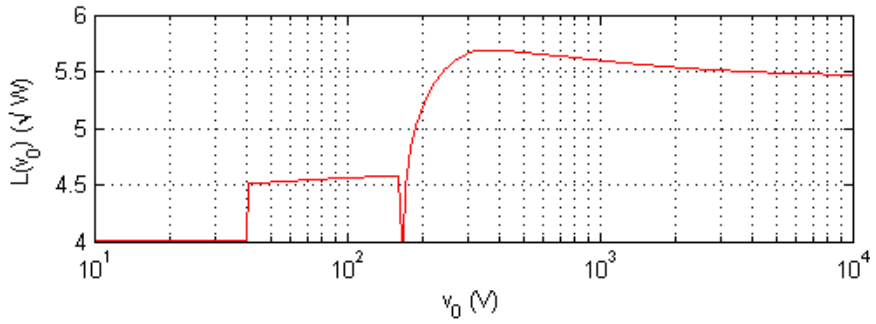


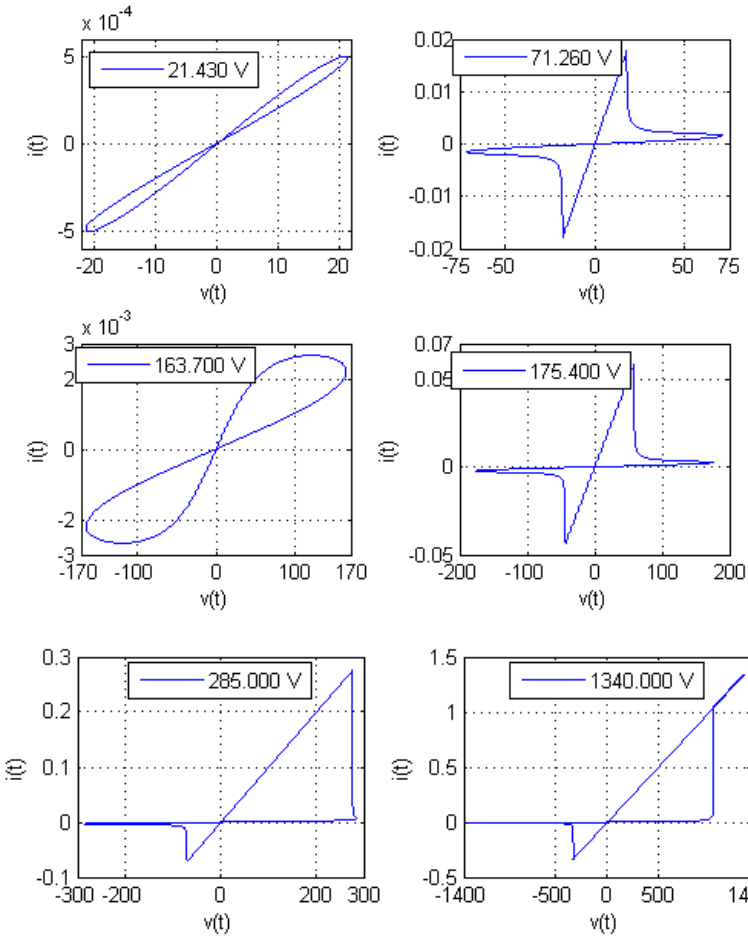


The window parameters used were $v_{th0} = v_0/2$ and $v_{th1} = v_0/4$.

This sweep was performed from $10^{-1}\ V$ to $10^{2}\ rad/s$

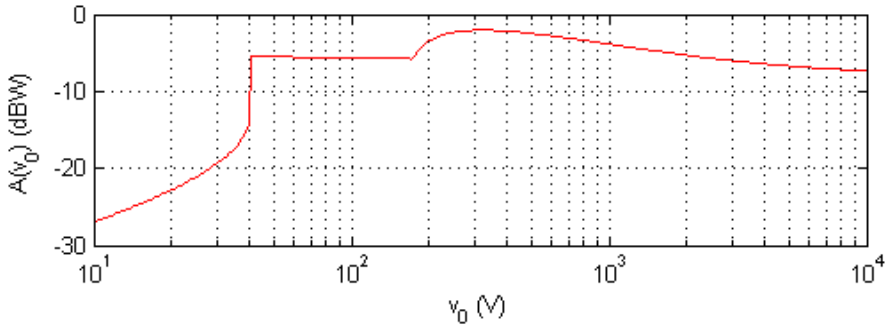


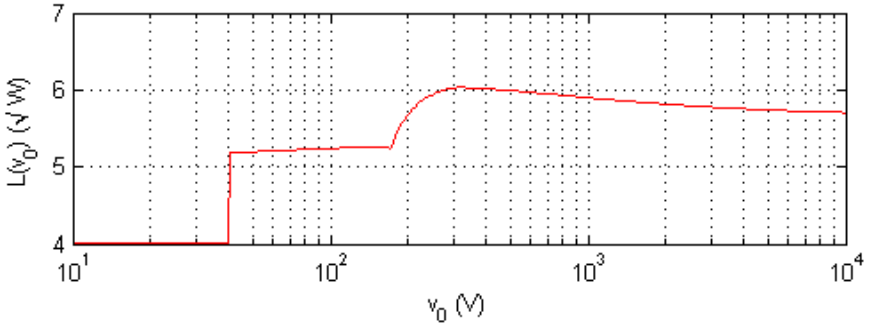


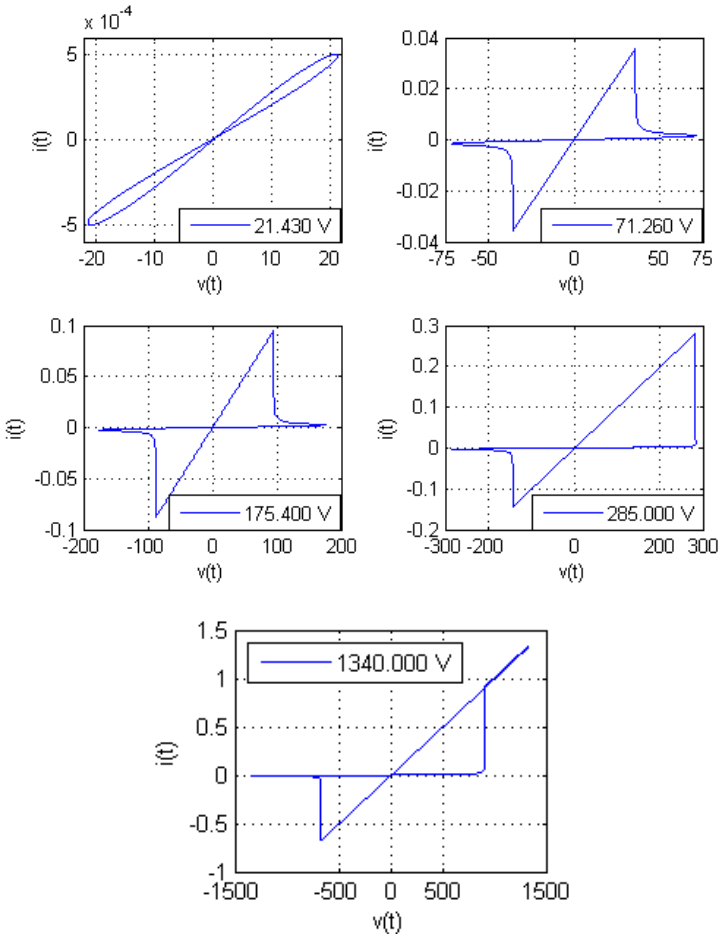


The window parameters used were $v_{th0} = v_0/4$ and $v_{th1} = v_0/2$.

This sweep was performed from $10^{-1}\ V$ to $10^{2}\ rad/s$

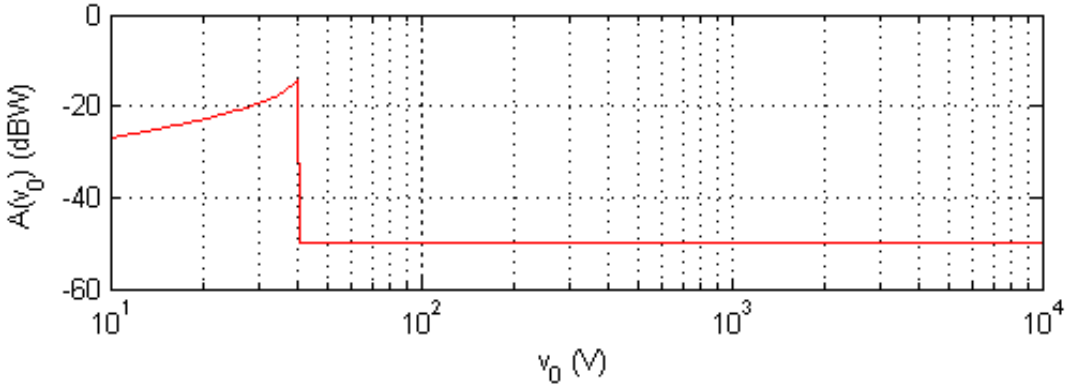


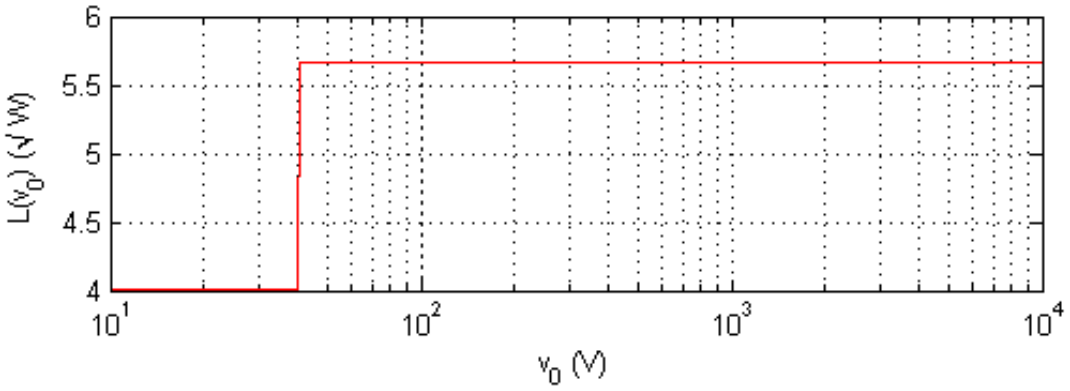


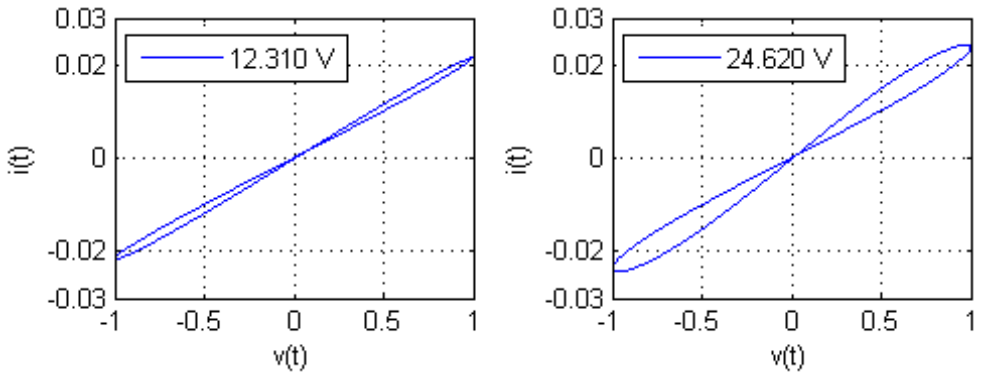


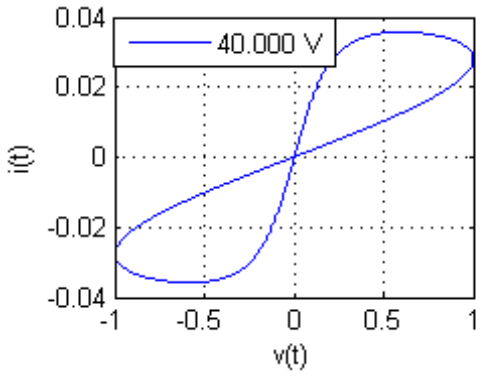
The window parameters used were $v_{th0} = 2v_0$ and $v_{th1} = 2v_0$.

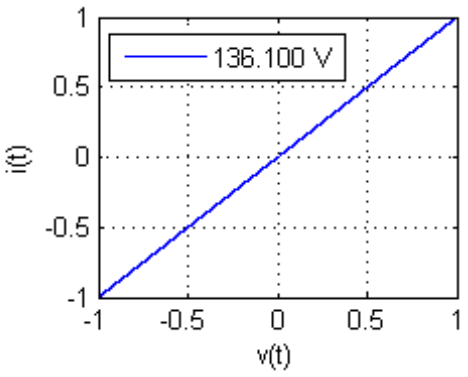
This sweep was performed from $10^{-1}\ V$ to $10^{2}\ rad/s$







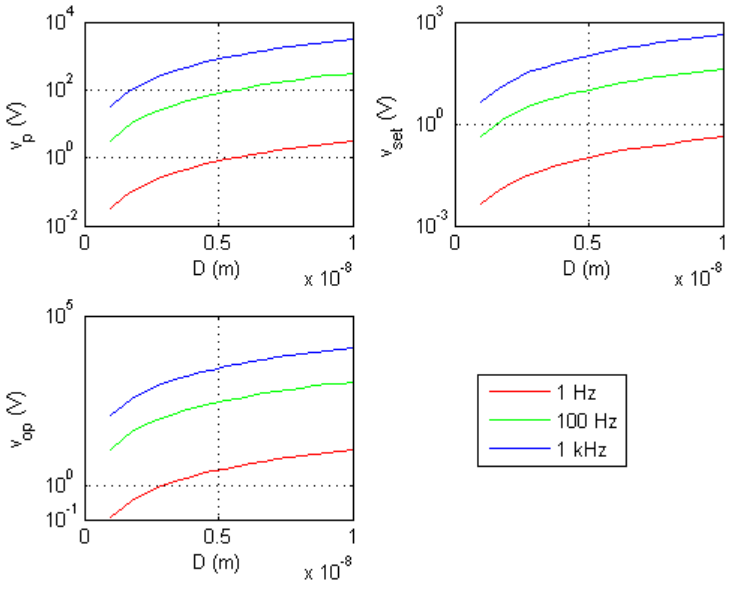




Figures of merit

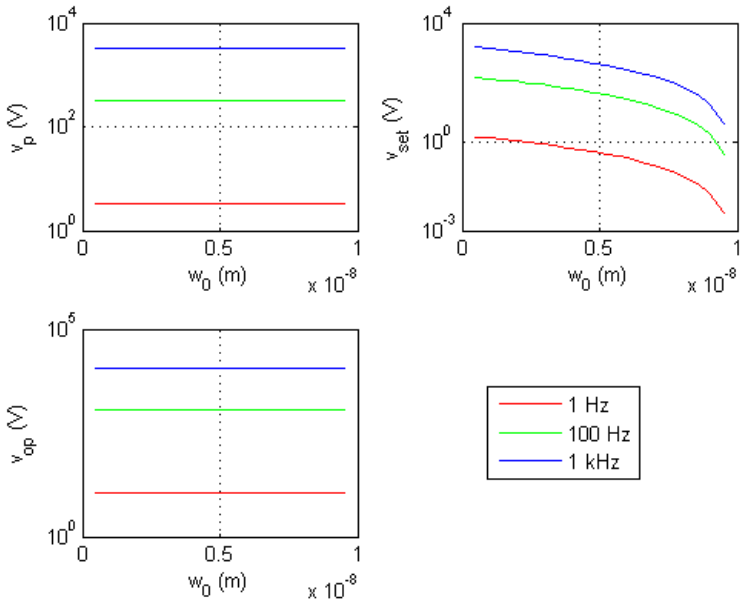
D variable

For these measures *D* took 20 values, from 1 nm to 20 nm.



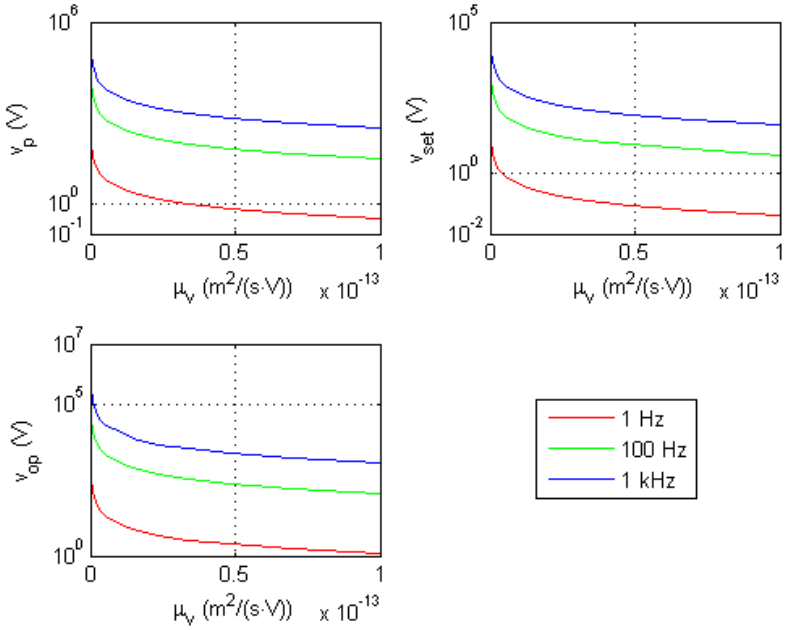
w_0 variable

For these measures w_0 took 20 values, from 5% of *D* to 95% of *D*.



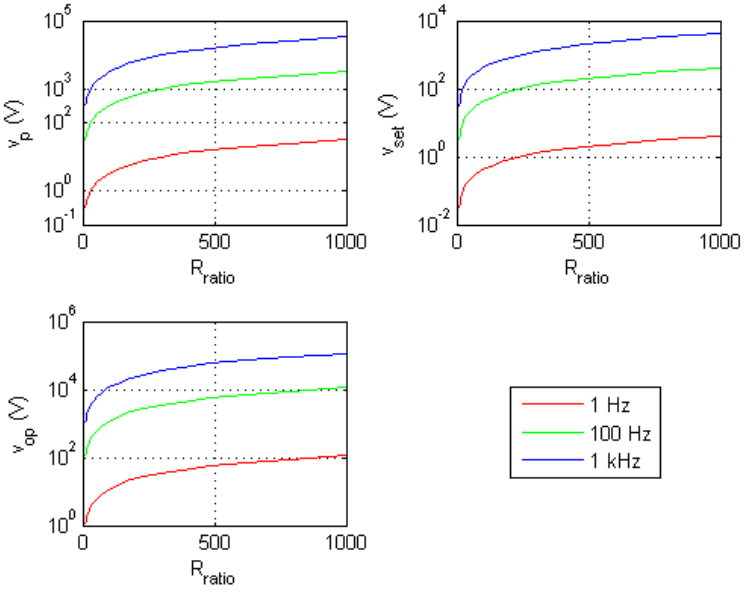
μ_v variable

For these measures μ_v took 20 values, from $1 \times 10^{-12}\ cm^2/Vs$ to $1 \times 10^{-9}\ cm^2/Vs$.



R_{ratio} variable

For these measures the ratio between R_{on} and R_{off} took 20 values, from 10 to 1000.



Matlab code for the frequency and amplitude tests

In this section are copies of the MATLAB code used to perform the frequency and amplitude sweeps for the various tests. It show be noted that some variables need to be specified to perform the individual tests.

Code for the frequency tests

test_file.m

```
clear all; clc;
%% Input parameters
global counter_variable counter_v0 N Ron Rratio D mu eta w0 stimulus_type
v0 P w_size w_min w_max err_frac err_frac_rd mem_model Af p vth0 vth1 A L
wop wh wl wp wset help_wset wm help_wm rd
% Memristor
type_of_variable = 1;    % 1 = D
                         % 2 = w0
                         % 3 = Mu
                         % 4 = Rratio
                         % 5 = Ron

if type_of_variable == 5
    Ron_total = logspace(1.69,3.7,20);
else
    Ron_total = 1000;
end
if type_of_variable == 4
    Rratio_total = logspace(1,3,20);
else
    Rratio_total = 100;
end
if type_of_variable == 1
    D_total = linspace(1e-9,10e-9, 20);
else
    D_total = 10e-9;
end
if type_of_variable == 3
    mu_total = logspace(-16,-13,2);
else
    mu_total = 1E-14;
end
if type_of_variable == 2
    w0_total = D_total*linspace(0.05,0.95,20);
else
    w0_total = D_total/2;
end
eta = 1;
% Input signal
N = 2000000;
```

```
w_size = 300;
w_min_total = 0;     % user defined
w_max_total = 2;     % user defined
v0_total = 1;
P = 2;  % number of periods
stimulus_type = 1;
% Measures variables
err_frac = 0.01;
err_frac_rd = 0.05;
% Models and Windows parameters
p = 10;          % For Prodromakis, Joglekar and Biolek
Af = 1;          % For Prodromakis
vth0 = v0/4;     % For BCM
vth1 = v0/4;     % For BCM
mem_model = 3;  % user defined
                 % mem_model=1 Strukov Model with HP window
                 % mem_model=2 Strukov Model with Joglekar window
                 % mem_model=3 Strukov Model with Prodromakis window
                 % mem_model=4 Strukov Model with Biolek window
                 % mem_model=5 Strukov Model with Boundary Condition
window type 1
%% Sweep personalization
if type_of_variable == 1
    variable_charac = D_total;
elseif type_of_variable == 2
    variable_charac = w0_total;
elseif type_of_variable == 3
    variable_charac = mu_total;
elseif type_of_variable == 4
    variable_charac = Rratio_total;
elseif type_of_variable == 5
    variable_charac = Ron_total;
end
%% Space allocation
L = zeros(length(variable_charac),length(v0_total),w_size);
A = zeros(length(variable_charac),length(v0_total),w_size);
wop = zeros(length(variable_charac),length(v0_total),1);
wh = zeros(length(variable_charac),length(v0_total),1);
wl = zeros(length(variable_charac),length(v0_total),1);
wp = zeros(length(variable_charac),length(v0_total),1);
wset = zeros(length(variable_charac),length(v0_total),1);
help_wset = zeros(length(variable_charac),length(v0_total),1);
wm = zeros(length(variable_charac),length(v0_total),1);
help_wm = zeros(length(variable_charac),length(v0_total),1);
rd = zeros(length(variable_charac),length(v0_total),1);
%% Calculations
for counter_variable = 1:length(variable_charac)
   if variable_charac == D_total
       D = D_total(counter_variable);
       w0 = w0_total(counter_variable);
       mu = mu_total(1);
       Rratio = Rratio_total(1);
       Ron = Ron_total(1);
   elseif variable_charac == w0_total
       D = D_total(1);
       w0 = w0_total(counter_variable);
       mu = mu_total(1);
       Rratio = Rratio_total(1);
       Ron = Ron_total(1);
```

```
    elseif variable_charac == mu_total
        D = D_total(1);
        w0 = w0_total(1);
        mu = mu_total(counter_variable);
        Rratio = Rratio_total(1);
        Ron = Ron_total(1);
    elseif variable_charac == Rratio_total
        D = D_total(1);
        w0 = w0_total(1);
        mu = mu_total(1);
        Rratio = Rratio_total(counter_variable);
        Ron = Rratio_total(1);
    elseif variable_charac == Ron_total
        D = D_total(1);
        w0 = w0_total(1);
        mu = mu_total(1);
        Rratio = Rratio_total(1);
        Ron = Ron_total(counter_variable);
    end
    for counter_v0 = 1:length(v0_total)
        v0 = v0_total(counter_v0);
        w_min = w_min_total;
        w_max = w_max_total;
        TiO2_Area_Length();
    end
end
%% Save Data to file
% save area and length results for future reference (takes time to
% simulate one run). file name includes file reference (LA_) plus date
and
% hour. Saved variables include A(w), L(w), w, memristor parameter
% settings and window function specs. Function LA_display , restore
variables,
% restores the graphics of A(w) and L(w) and presents a report on the
used
% parameters.
c=clock;
filename=sprintf('%s%d%d%d%s%d%s%d%s','LA_f_sweep_',c(1),c(2),c(3),'_',c(
4),'h',c(5),'.mat');
save(filename,'L','A','v0_total','w_size','w_min_total','w_max_total','ty
pe_of_variable','P','N','Ron_total','Rratio_total','D_total','w0_total','
mu_total','eta','p','Af','vth0','vth1','wop','wh','wl','wp','wset','help_
wset','wm','help_wm','rd','err_frac','err_frac_rd','stimulus_type','mem_m
odel');
%% Displays
LA_display(filename);
Display_Measures(filename);
```

TiO2_Area_Length.m

```
function TiO2_Area_Length()
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%
% Memristor parameters:
%
% Ron -> Ron resistance;
% dltR -> reason between Ron and Roff resistances (dltR = Roff/Ron);
```

```
% Roff -> Roff resistance;
% D -> width of the sample;
% mu -> mobility;
% eta -> doping type;
% K -> process parameter;
% W0 -> initial interface position;
%
% Stimulus specs:
%
% w_size -> number of elements of the sweep;
% w_min -> minimum value of the sweep;
% w_max -> maximum value of the sweep;
% v0 -> amplitude of the input signal;
% v0_total -> sweep array of the stimulus amplitude;
% t0 -> memristor characteristic time;
% w -> angular frequency - non-normalized;
% P -> number of periods;
% N -> number of sampled points;
% t -> time array;
% dt -> integration step;
%
% Normalizing parameters:
%
% i0 -> normalized current;
%
% Measurement parameters:
%
% err_frac -> percentage of the error tolerance for vset (0 to 1);
% err_frac_rd -> percentage of the error toleration for help_vset (0 to
1);
%
% Measures of data:
%
% wop -> frequency range where A(w) maintains amplitude above 3dB;
% wh -> frequency after peak where A(w) is first below 3dB;
% wl -> frequency before peak where A(w) is first below 3dB;
% wp -> peak frequency in L(w);
% wset -> settling frequency, the frequency at which the memristor
%         contracts into a linear resistance. Search method with err
%         tolerance based on L(inf);
% wm -> loop maintenance range wm=wset-wp;
% help_wm -> loop maintenance range used in rd;
% rd -> rate of decay rd=(A(wp)-A(wset))/wm;  (module)
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%

global counter_variable counter_v0 L A t N Ron Rratio Roff stimulus_type
P w_size w_min w_max w v0 v i
%% Inputs
% Memristor parameters
Roff = Ron*Rratio;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%
% Stimulus specs
w = logspace(w_min, w_max, w_size);
% Normalizing parameters
i0 = v0/Ron;
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%
% Measurement parameters
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%
%% Space allocation
%% Calculations
for n = 1:length(w)
    t = linspace(0,P*2*pi/w(n),N);
    v = stimvoltage(v0,w(n),t,stimulus_type);
    TiO2Models();
    di2 = zeros(1,N/2);
    dv2 = zeros(1,N/2);
    % normalizing i(t) and v(t)
    i = i/i0;
    v = v/v0;
    % second period
    N2 = N/2;
    i2 = i(N2+1:end);
    v2 = v(N2+1:end);
    % time derivatives of v2(t) and i2(t)
    di2(1) = i2(1)-i(N2);
    di2(2:end) = i2(2:end)-i2(1:end-1);
    dv2(1) = v2(1)-v(N2);
    dv2(2:end) = v2(2:end)-v2(1:end-1);

    % area integration - Green theorem: absolute value of the areas of
the
    % two lobes. Note: Biolek and BCM window functions may exhibit
    % different hard switching dynamics, which may constrain the ideal
    % periodic behaviour of M(q). For these cases integration must be
    % performed over the entire period of v(t), as shown below.
    A(counter_variable,counter_v0,n) =
abs(i2(1:N2/2)*dv2(1:N2/2)')+abs(i2(N2/2:end)*dv2(N2/2:end)');
    % length integration
    L(counter_variable,counter_v0,n) = (sum(sqrt((di2').^2+(dv2').^2)))';
    % Note: dt is not considered on time derivatives steps, since it will
    % cancel out during integration.
end
%% Measures of relevant data
LA_measures();
end
```

TiO2Models.m

```
function  TiO2Models()
% function TiO2Models()
% Auxiliary function to calculate the current through the TiO2 MD
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% Strukov Model equations
%
% v(t)=M(x)i(t)
% dx/dt=KF(x)i(t)
% M(x)=Roff-(Roff-Ron)x
% x=W/D is the state variable of the model, representing the normalized
```

```
% interface position 0<x<1
% F(x) represents the window function, used to model boundary conditions
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
global N t Ron Roff D mu eta i v w0 p Af vth0 vth1
K=eta*mu*Ron/D;     % process parameter

dt=t(end)/(N-1);

i = zeros(1,N);
W = zeros(1,N);
F = zeros(1,N);
M = zeros(1,N);
% v : from outside
% t : from outside

% Comment the unneeded window functions

%% Parabolic window
W(1) = w0;
M(1) = Roff+(Ron-Roff)*W(1)/D;
F(1) = W(1)/D-(W(1)/D)^2;
for n = 2:N
    W(n)=W(n-1)+K*i(n-1).*F(n-1)*dt;
    x = W(n)/D;
    M(n)=Roff+(Ron-Roff)*x;
    i(n)=v(n)./M(n);
    F(n)=x-x^2;
end
% % Joglekar window
W(1) = w0;
M(1) = Roff+(Ron-Roff)*W(1)/D;
F(1)=1-(2*W(1)/D-1)^(2*p);
for n = 2:N
    W(n)=W(n-1)+K*i(n-1).*F(n-1)*dt;
    x = W(n)/D;
    M(n)=Roff+(Ron-Roff)*x;
    i(n)=v(n)./M(n);
    F(n)=1-(2*x-1)^(2*p);
end
%% Prodromakis window
W(1) = w0;
x = W(1)/D;
M(1) = Roff+(Ron-Roff)*x;
F(1) = Af*(1-((x-0.5)^2+0.75)^p);
for n = 2:N
    dW = K*i(n-1).*F(n-1);
    if ((x == 1) && (dW > 0))
        dW = 0;
    elseif ((x == 0) && (dW < 0))
        dW = 0;
    end
    W(n) = W(n-1)+dW*dt;
    x = W(n)/D;
    if x > 1
        x = 1;
    elseif x < 0
        x = 0;
```

```
    end
    M(n) = Roff+(Ron-Roff)*x;
    i(n) = v(n)./M(n);
    F(n) = Af*(1-((x-0.5)^2+0.75)^p);
end
%% Biolek window
W(1) = w0;
M(1) = Roff+(Ron-Roff)*W(1)/D;
if eta*v(1)>0
    F(1) = 1-(W(1)/D)^(2*p);
else
    F(1) = 1-(W(1)/D-1)^(2*p);
end
for n = 2:N
    W(n)=W(n-1)+K*i(n-1).*F(n-1)*dt;
    x = W(n)/D;
    M(n)=Roff+(Ron-Roff)*x;
    i(n)=v(n)./M(n);
    if eta*v(n)>0
        F=1-(x)^(2*p);
    else
        F=1-(x-1)^(2*p);
    end
end
%% Boundary condition type 1
W(1) = w0;
x = W(1)/D;
M(1) = Roff+(Ron-Roff)*x;
F(1) = 2;
if ((0<x && x<1) || (x==0 && eta*v(1)>vth0) || (x==1 && eta*v(1)<-vth1))
    F(1) = 1;
elseif ((x==0 && eta*v(1)<=vth0) || (x==1 && eta*v(1)>=-vth1))
    F(1) = 0;
end
for n = 2:N
    dW = K*i(n-1).*F(n-1);
    if ((x == 1) && (dW > 0))
        dW = 0;
    elseif ((x == 0) && (dW < 0))
        dW = 0;
    end
    W(n) = W(n-1)+dW*dt;
    x = W(n)/D;
    if x > 1
        x = 1;
    elseif x < 0
        x = 0;
    end
    M(n)=Roff+(Ron-Roff)*x;
    i(n)=v(n)./M(n);
    F(n)=2;
    if ((0<x && x<1) || (x==0 && eta*v(n)>vth0) || (x==1 && eta*v(n)<-
vth1))
        F(n)=1;
    elseif ((x==0 && eta*v(n)<=vth0) || (x==1 && eta*v(n)>=-vth1))
        F(n)=0;
    end
end
end
```

stimvoltage.m

```
function v = stimvoltage( v0,w,t,type )
% Specifies the stimulus voltage shape
% vo   - Stimulus amplitude
% w    - Stimulus frequency
% t    - Time vector
% type - Stimulus type
%         1 - sinewave
%         2 - triangular wave
%         3 - square wave
%         4 - square wave with limited rise time (5% stimulus period)
%         5 - alternating sign sinewaves
switch type
    case 1
        v=v0*sin(w*t);
    case 2
        v=v0*sawtooth(w*t,0.5);
    case 3
        v=v0*square(w*t);
    case 4
        [Bf,Af]=butter(1,7*w,'s');
        v=lsim(tf(Bf,Af),v0*square(w*t),t);
    case 5
        v(t<3*pi/w)=v0*sin(w*t(t<3*pi/w)).*sin(w*t(t<3*pi/w));
        v(t>=3*pi/w)=-v0*sin(w*t(t>3*pi/w)).*sin(w*t(t>3*pi/w));
        % note: valid for six sinewave periods only
    otherwise
        v=zeros(1,N);
end
end
```

LA_measures.m

```
function LA_measures()
% Calculates morphological measures of TiO2 memristors hysteresis loops
% using Area A(v0) and Length L(v0) versus frequency plots.
%
% The input parameters are:
% L - L(w) length versus frequency vector, linear scale;
% A - A(w) area versus frequency vector, linear scale;
% w - frequency vector;
% err - error tolerance for wset search;
%
% The output parameters are:
% wop - amplitude range where A(w) maintains amplitude above 3dB;
% wp - peak frequency in L(w);
% wset - settling frequency, the frequency at which the memristor
%        contracts into a linear resistance. Search method with err
%        tolerance based on L(end);
% help_wset - settling frequency used in rd;
% wm - loop maintenance range wm=wp-wset;
% rd - rate of decay rd=(A(wp)-A(wset))/wm; (module)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
global counter_variable counter_v0 err_frac err_frac_rd L A w wop wh wl
wp wset help_wset wm help_wm rd
```

```
% dB conversion %
Adb(1,:)=10*log10(A(counter_variable,counter_v0,:));
% measures dependent on L(w)
L_temp(1,:) = L(counter_variable,counter_v0,:);
[~,ind_Lp] = max(L_temp);
wp(counter_variable,counter_v0,1) = w(ind_Lp);
% settling amplitude
err = L_temp(end)*err_frac;
wset_array = w(w > wp(counter_variable,counter_v0,1) & abs(L_temp-
L_temp(end)) >= err);
wset(counter_variable,counter_v0,1) = wset_array(end);
% loop maintenance range
wm(counter_variable,counter_v0,1) = wset(counter_variable,counter_v0,1)-
wp(counter_variable,counter_v0,1);
% measures dependent on A(w)
[MaxA,ind_A] = max(Adb);
% rate of decay
err_rd = L_temp(end)*err_frac_rd;
help_wset_array = w(w > wp(counter_variable,counter_v0,1) & abs(L_temp-
L_temp(end)) >= err_rd);
help_wset(counter_variable,counter_v0,1) = help_wset_array(1);
help_wm(counter_variable,counter_v0,1) =
help_wset(counter_variable,counter_v0,1)-
wp(counter_variable,counter_v0,1);
rd(counter_variable,counter_v0,1) = (Adb(ind_Lp) - Adb(w ==
wset(counter_variable,counter_v0,1)))/wm(counter_variable,counter_v0,1);

wl_array = w(w < w(ind_A) & Adb-MaxA <= -3);
stop_1 = 0;
wl_index = length(wl_array);
while stop_1 == 0
    wl_temp = wl_array(1,wl_index);
    index_temp = find(w == wl_temp);
    Cond1 = (Adb(index_temp-1) < Adb(index_temp));
    Cond2 = (abs(Adb(index_temp)-Adb(index_temp-1)) < 0.1);
    if (Cond1 || Cond2)
        wl(counter_variable,counter_v0,1) = wl_temp;
        stop_1 = 1;
    else
        wl_index = wl_index-1;
    end
end
wh_array = w(w > w(ind_A) & Adb-MaxA <= -3);
stop_2 = 0;
wh_index = 1;
while stop_2 == 0
    wh_temp = wh_array(1,wh_index);
    index_temp = find(w == wh_temp);
    Cond1 = Adb(index_temp+1) < Adb(index_temp);
    Cond2 = (abs(Adb(index_temp)-Adb(index_temp+1)) < 0.1);
    if (Cond1 || Cond2)
        wh(counter_variable,counter_v0,1) = wh_temp;
        stop_2 = 1;
    else
        wh_index = wh_index+1;
    end
end
wop(counter_variable,counter_v0,1) = wh(counter_variable,counter_v0,1)-
wl(counter_variable,counter_v0,1);
```

```
end
```

LA_display.m

```
function LA_display(filename)
% Function LA_display , restore variables,
% restores the graphics of A(w) and L(w)
load(filename);
% build graphics
for counter_v0 = 1:length(v0_total)
    txt=sprintf('%s %d','Window Type:',mem_model);
    figure('Name',txt);
    w(1,:) = logspace(w_min_total,w_max_total,w_size);
    if type_of_variable == 1
        variable_charac = D_total;
    elseif type_of_variable == 2
        variable_charac = w0_total;
    elseif type_of_variable == 3
        variable_charac = mu_total;
    elseif type_of_variable == 4
        variable_charac = Rratio_total;
    elseif type_of_variable == 5
        variable_charac = Ron_total;
    end
    subplot(2,1,1);
    for counter_variable = 1:length(variable_charac)
        a(1,:) = A(counter_variable,counter_v0,:);
        semilogx(w(1,:),10*log10(abs(a(:))),'r');
        hold on;
        wl_plot(1,:) = wl(counter_variable,counter_v0,1);
        semilogx(wl_plot,10*log10(a(w == wl_plot)),'bo');
        wh_plot(1,:) = wh(counter_variable,counter_v0,1);
        semilogx(wh_plot,10*log10(a(w == wh_plot)),'bo');
    end
    xlabel('\omega (rad/s)');
    ylabel('A(\omega) (dBW)');
    grid on;

    subplot(2,1,2);
    for counter_variable = 1:length(variable_charac)
        l(1,:) = L(counter_variable,counter_v0,:);
        semilogx(w(1,:),l(:),'r');
        hold on;
        wp_plot(1,:) = wp(counter_variable,counter_v0,1);
        semilogx(wp_plot,l(w == wp_plot),'bo');
        wset_plot(1,:) = wset(counter_variable,counter_v0,1);
        semilogx(wset_plot,l(w == wset_plot),'bo');
    end
    xlabel('\omega (rad/s)');
    ylabel('L(\omega) (\surd W)');
    grid on;
end
end
```

Display_Measures.m

```
function Display_Measures(filename)
% Function Display_Measures , restore variables,
% restores the graphics of:
% wop - frequency range where A(v0) maintains amplitude above 3dB
% wp - peak frequency in L(v0)
% wset - settling amplitude, the amplitude at which the memristor
%        contracts into a linear resistance. Search method with 5%
%        tolerance based on L(1).
% wm - loop maintenance bandwidth wm=wset-wp
% rd - rate of decay rd=(A(wp)-A(wset))/wm (module)
clc
load(filename);

if type_of_variable == 1
    x_plot = D_total;
    x_label_text = sprintf('D (m)');
elseif type_of_variable == 2
    x_plot = w0_total;
    x_label_text = sprintf('w_0 (m)');
elseif type_of_variable == 4
    x_plot = Rratio_total;
    x_label_text = sprintf('R_r_a_t_i_o');
end
for counter_v0 = 1:length(v0_total)

    subplot(2,2,1);
    hold on;
    wp_plot(1,:) = wp(:,counter_v0,1);
    semilogy(x_plot, wp_plot,'r');
    if type_of_variable == 3
        xlabel('\mu_v (m^2/(s\cdotV))');
    elseif type_of_variable == 5
        xlabel('R_o_n (\Omega)');
    else
        xlabel(x_label_text);
    end
    ylabel('\omega_p (rad/s)');
    grid on;

    subplot(2,2,2);
    hold on;
    wset_plot(1,:) = wset(:,counter_v0,1);
    semilogy(x_plot, wset_plot,'r');
    if type_of_variable == 3
        xlabel('\mu_v (m^2/(s\cdotV))');
    elseif type_of_variable == 5
        xlabel('R_o_n (\Omega)');
    else
        xlabel(x_label_text);
    end
    ylabel('\omega_s_e_t (rad/s)');
    grid on;

    subplot(2,2,3);
    hold on;
```

```
    wop_plot(1,:) = wop(:,counter_v0,1);
    semilogy(x_plot,wop_plot,'r');
    if type_of_variable == 3
        xlabel('\mu_v (m^2/(s\cdotV))');
    elseif type_of_variable == 5
        xlabel('R_o_n (\Omega)');
    else
        xlabel(x_label_text);
    end
    ylabel('\omega_o_p (rad/s)');
    grid on;

    subplot(2,2,4);
    hold on;
    rd_plot(1,:) = rd(:,counter_v0,1);
    semilogy(x_plot,rd_plot,'r');
    xlabel(x_label_text);
    ylabel('r_d (m^2/(rad/s))');
    grid on;
end
```

Code for the amplitude tests

test_file.m

```
clear all; clc;
%% Input parameters
global counter_variable counter_f N Ron Rratio D mu eta w0 stimulus_type
f P v0_size v0_min v0_max err_frac err_frac_rd mem_model Af p A L vop vh
vl vp vset help_vset vm help_vm rd
% Memristor
type_of_variable = 1;    % user defined
                         % 1 = D
                         % 2 = w0
                         % 3 = Mu
                         % 4 = Rratio
                         % 5 = Ron
if type_of_variable == 5
    Ron_total = logspace(1.69,3.7,20);
else
    Ron_total = 1000;
end
if type_of_variable == 4
    Rratio_total = logspace(1,3,20);
else
    Rratio_total = 100;
end
if type_of_variable == 1
    D_total = linspace(1e-9,10e-9, 20);
else
    D_total = 10e-9;
end
if type_of_variable == 3
    mu_total = logspace(-16,-13,2);
else
```

```
    mu_total = 1E-14;
end
if type_of_variable == 2
    w0_total = D_total*linspace(0.05,0.95,20);
else
    w0_total = D_total/2;
end
eta = 1;
% Input signal
N = 2000000;
v0_size = 300;
v0_min_total = [-3, -1, 0]; % user defined
v0_max_total = [4, 6, 7];   % user defined
f_total = [1, 100, 1e3];
P = 2;  % number of periods
stimulus_type = 1;
% Measures variables
err_frac = 0.01;
err_frac_rd = 0.05;
% Models and Windows parameters
p = 10;          % For Prodromakis, Joglekar and Biolek
Af = 1;          % For Prodromakis
mem_model = 1;   % user defined
                 % mem_model=1 Strukov Model with parabolic window
                 % mem_model=2 Strukov Model with Joglekar window
                 % mem_model=3 Strukov Model with Prodromakis window
                 % mem_model=4 Strukov Model with Biolek window
                 % mem_model=5 Strukov Model with Boundary Condition
window type 1
%% Sweep personalization
if type_of_variable == 1
    variable_charac = D_total;
elseif type_of_variable == 2
    variable_charac = w0_total;
elseif type_of_variable == 3
    variable_charac = mu_total;
elseif type_of_variable == 4
    variable_charac = Rratio_total;
elseif type_of_variable == 5
    variable_charac = Ron_total;
end
%% Space allocation
L = zeros(length(variable_charac),length(f_total),v0_size);
A = zeros(length(variable_charac),length(f_total),v0_size);
vop = zeros(length(variable_charac),length(f_total),1);
vh = zeros(length(variable_charac),length(f_total),1);
vl = zeros(length(variable_charac),length(f_total),1);
vp = zeros(length(variable_charac),length(f_total),1);
vset = zeros(length(variable_charac),length(f_total),1);
help_vset = zeros(length(variable_charac),length(f_total),1);
vm = zeros(length(variable_charac),length(f_total),1);
help_vm = zeros(length(variable_charac),length(f_total),1);
rd = zeros(length(variable_charac),length(f_total),1);
%% Calculations
for counter_variable = 1:length(variable_charac)
   if variable_charac == D_total
       D = D_total(counter_variable);
       w0 = w0_total(counter_variable);
       mu = mu_total(1);
```

```
        Rratio = Rratio_total(1);
        Ron = Ron_total(1);
    elseif variable_charac == w0_total
        D = D_total(1);
        w0 = w0_total(counter_variable);
        mu = mu_total(1);
        Rratio = Rratio_total(1);
        Ron = Ron_total(1);
    elseif variable_charac == mu_total
        D = D_total(1);
        w0 = w0_total(1);
        mu = mu_total(counter_variable);
        Rratio = Rratio_total(1);
        Ron = Ron_total(1);
    elseif variable_charac == Rratio_total
        D = D_total(1);
        w0 = w0_total(1);
        mu = mu_total(1);
        Rratio = Rratio_total(counter_variable);
        Ron = Rratio_total(1);
    elseif variable_charac == Ron_total
        D = D_total(1);
        w0 = w0_total(1);
        mu = mu_total(1);
        Rratio = Rratio_total(1);
        Ron = Ron_total(counter_variable);
    end
    for counter_f = 1:length(f_total)
        f = f_total(counter_f);
        v0_min = v0_min_total(counter_f);
        v0_max = v0_max_total(counter_f);
        TiO2_Area_Length();
    end
end
%% Save Data to file
% save area and length results for future reference (takes time to
% simulate one run). file name includes file reference (LA_) plus date
and
% hour. Saved variables include A(w), L(w), w, memristor parameter
% settings and window function specs. Function LA_display , restore
variables,
% restores the graphics of A(w) and L(w) and presents a report on the
used
% parameters.
c=clock;
filename=sprintf('%s%d%d%d%s%d%s%d%s','LA_v0_sweep_',c(1),c(2),c(3),'_',c
(4),'h',c(5),'.mat');
% Save
save(filename,'L','A','f_total','v0_size','v0_min_total','v0_max_total','
type_of_variable','P','N','Ron_total','Rratio_total','D_total','w0_total'
,'mu_total','eta','p','Af','vth0','vth1','vop','vh','vl','vp','vset','hel
p_vset','vm','help_vm','rd','err_frac','err_frac_rd','stimulus_type','mem
_model');
%% Displays
LA_display(filename);
Display_Measures(filename);
```

TiO2_Area_Length.m

```
function TiO2_Area_Length()
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%
% Memristor parameters:
%
% Ron -> Ron resistance;
% dltR -> reason between Ron and Roff resistances (dltR = Roff/Ron);
% Roff -> Roff resistance;
% D -> width of the sample;
% mu -> mobility;
% eta -> doping type;
% K -> process parameter;
% W0 -> initial interface position;
%
% Stimulus specs:
%
% v0_size -> number of elements of the sweep;
% v0_min -> minimum value of the sweep;
% v0_max -> maximum value of the sweep;
% v0 -> amplitude of the input signal;
% v0_total -> sweep array of the stimulus amplitude;
% t0 -> memristor characteristic time;
% w -> angular frequency - non-normalized;
% P -> number of periods;
% N -> number of sampled points;
% t -> time array;
% dt -> integration step;
%
% Normalizing parameters:
%
% i0 -> normalized current;
%
% Measurement parameters:
%
% err_frac -> percentage of the error tolerance for vset (0 to 1);
% err_frac_rd -> percentage of the error toleration for help_vset (0 to
1);
%
% Measures of data:
%
% vop -> amplitude range where A(v0) maintains amplitude above 3dB;
% vh -> amplitude after peak where A(v0) is first below 3dB;
% vl -> amplitude before peak where A(v0) is first below 3dB;
% vp -> peak amplitude in L(v0);
% vset -> settling amplitude, the amplitude at which the memristor
%         contracts into a linear resistance. Search method with err
%         tolerance based on L(1);
% vm -> loop maintenance range vm=vp-vset;
% help_vm -> loop maintenance range used in rd;
% rd -> rate of decay rd=(A(vp)-A(vset))/vm; (module)
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%

global counter_variable counter_f L A t N Ron Rratio Roff stimulus_type P
v0_size v0_min v0_max f v0 v i
```

```
%% Inputs
% Memristor parameters
Roff = Ron*Rratio;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%
% Stimulus specs
v0 = logspace(v0_min, v0_max, v0_size);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%
% Normalizing parameters
i0 = v0./Ron;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%
% Measurement parameters
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%
%% Space allocation
%% Calculations
for n = 1:length(v0)
    w = 2*pi*f;
    t = linspace(0,P*2*pi/w,N);
    v = stimvoltage(v0(n),w,t,stimulus_type);
    vth0 = v0(n)/4;     % For BCM
    vth1 = v0(n)/4;     % For BCM
    TiO2Models();
    di2 = zeros(1,N/2);
    dv2 = zeros(1,N/2);
    % normalizing i(t) and v(t)
    i = i/i0(n);
    v = v/v0(n);
    % second period
    N2 = N/2;
    i2 = i(N2+1:end);
    v2 = v(N2+1:end);
    % time derivatives of v2(t) and i2(t)
    di2(1) = i2(1)-i(N2);
    di2(2:end) = i2(2:end)-i2(1:end-1);
    dv2(1) = v2(1)-v(N2);
    dv2(2:end) = v2(2:end)-v2(1:end-1);

    % area integration - Green theorem: absolute value of the areas of
the
    % two lobes. Note: Biolek and Corinto window functions may exhibit
    % different hard switching dynamics, which may constrain the ideal
    % periodic behaviour of M(q). For these cases integration must be
    % performed over the entire period of v(t), as shown below.
    A(counter_variable,counter_f,n) =
abs(i2(1:N2/2)*dv2(1:N2/2)')+abs(i2(N2/2:end)*dv2(N2/2:end)');
    % length integration
    L(counter_variable,counter_f,n) = (sum(sqrt((di2').^2+(dv2').^2)))';
    % Note: dt is not considered on time derivatives steps, since it will
    % cancel out during integration.

    fprintf('passou %d vezes de %d \n', n, length(v0));
end
%% Measures of relevant data
LA_measures();
end
```

TiO2Models.m

```
function  TiO2Models()
% function TiO2Models()
% Auxiliary function to calculate the current through the TiO2 MD
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% Strukov Model equations
%
% v(t)=M(x)i(t)
% dx/dt=KF(x)i(t)
% M(x)=Roff-(Roff-Ron)x
% x=W/D is the state variable of the model, representing the normalized
% interface position 0<x<1
% F(x) represents the window function, used to model boundary conditions
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
global N t Ron Roff D mu eta i v w0 p Af vth0 vth1
K=eta*mu*Ron/D;     % process parameter

dt=t(end)/(N-1);

i = zeros(1,N);
W = zeros(1,N);
F = zeros(1,N);
M = zeros(1,N);
% v : from outside
% t : from outside

% Comment the unneeded window functions

%% Parabolic window
W(1) = w0;
M(1) = Roff+(Ron-Roff)*W(1)/D;
F(1) = W(1)/D-(W(1)/D)^2;
for n = 2:N
    W(n)=W(n-1)+K*i(n-1).*F(n-1)*dt;
    x = W(n)/D;
    M(n)=Roff+(Ron-Roff)*x;
    i(n)=v(n)./M(n);
    F(n)=x-x^2;
end
% % Joglekar window
W(1) = w0;
M(1) = Roff+(Ron-Roff)*W(1)/D;
F(1)=1-(2*W(1)/D-1)^(2*p);
for n = 2:N
    W(n)=W(n-1)+K*i(n-1).*F(n-1)*dt;
    x = W(n)/D;
    M(n)=Roff+(Ron-Roff)*x;
    i(n)=v(n)./M(n);
    F(n)=1-(2*x-1)^(2*p);
end
%% Prodromakis window
```

```
W(1) = w0;
x = W(1)/D;
M(1) = Roff+(Ron-Roff)*x;
F(1) = Af*(1-((x-0.5)^2+0.75)^p);
for n = 2:N
    dW = K*i(n-1).*F(n-1);
    if ((x == 1) && (dW > 0))
        dW = 0;
    elseif ((x == 0) && (dW < 0))
        dW = 0;
    end
    W(n) = W(n-1)+dW*dt;
    x = W(n)/D;
    if x > 1
        x = 1;
    elseif x < 0
        x = 0;
    end
    M(n) = Roff+(Ron-Roff)*x;
    i(n) = v(n)./M(n);
    F(n) = Af*(1-((x-0.5)^2+0.75)^p);
end
%% Biolek window
W(1) = w0;
M(1) = Roff+(Ron-Roff)*W(1)/D;
if eta*v(1)>0
    F(1) = 1-(W(1)/D)^(2*p);
else
    F(1) = 1-(W(1)/D-1)^(2*p);
end
for n = 2:N
    W(n)=W(n-1)+K*i(n-1).*F(n-1)*dt;
    x = W(n)/D;
    M(n)=Roff+(Ron-Roff)*x;
    i(n)=v(n)./M(n);
    if eta*v(n)>0
        F=1-(x)^(2*p);
    else
        F=1-(x-1)^(2*p);
    end
end
%% Boundary condition type 1
W(1) = w0;
x = W(1)/D;
M(1) = Roff+(Ron-Roff)*x;
F(1) = 2;
if ((0<x && x<1) || (x==0 && eta*v(1)>vth0) || (x==1 && eta*v(1)<-vth1))
    F(1) = 1;
elseif ((x==0 && eta*v(1)<=vth0) || (x==1 && eta*v(1)>=-vth1))
    F(1) = 0;
end
for n = 2:N
    dW = K*i(n-1).*F(n-1);
    if ((x == 1) && (dW > 0))
        dW = 0;
    elseif ((x == 0) && (dW < 0))
        dW = 0;
    end
    W(n) = W(n-1)+dW*dt;
```

```
    x = W(n)/D;
    if x > 1
        x = 1;
    elseif x < 0
        x = 0;
    end
    M(n)=Roff+(Ron-Roff)*x;
    i(n)=v(n)./M(n);
    F(n)=2;
    if ((0<x && x<1) || (x==0 && eta*v(n)>vth0) || (x==1 && eta*v(n)<-
vth1))
        F(n)=1;
    elseif ((x==0 && eta*v(n)<=vth0) || (x==1 && eta*v(n)>=-vth1))
        F(n)=0;
    end
end
end
```

stimvoltage.m

```
function v = stimvoltage( v0,w,t,type )
% Specifies the stimulus voltage shape
% vo   - Stimulus amplitude
% w    - Stimulus frequency
% t    - Time vector
% type - Stimulus type
%         1 - sinewave
%         2 - triangular wave
%         3 - square wave
%         4 - square wave with limited rise time (5% stimulus period)
%         5 - alternating sign sinewaves
switch type
    case 1
        v=v0*sin(w*t);
    case 2
        v=v0*sawtooth(w*t,0.5);
    case 3
        v=v0*square(w*t);
    case 4
        [Bf,Af]=butter(1,7*w,'s');
        v=lsim(tf(Bf,Af),v0*square(w*t),t);
    case 5
        v(t<3*pi/w)=v0*sin(w*t(t<3*pi/w)).*sin(w*t(t<3*pi/w));
        v(t>=3*pi/w)=-v0*sin(w*t(t>3*pi/w)).*sin(w*t(t>3*pi/w));
        % note: valid for six sinewave periods only
    otherwise
        v=zeros(1,N);
end
end
```

LA_measures.m

```
function LA_measures()
% Calculates morphological measures of TiO2 memristors hysteresis loops
% using Area A(v0) and Length L(v0) versus frequency plots.
```

```
%
% The input parameters are:
% L - L(w) length versus frequency vector, linear scale;
% A - A(w) area versus frequency vector, linear scale;
% w - frequency vector;
% err - error tolerance for wset search;
%
% The output parameters are:
% wop - amplitude range where A(w) maintains amplitude above 3dB;
% wp - peak frequency in L(w);
% wset - settling amplitude, the amplitude at which the memristor
%        contracts into a linear resistance. Search method with err
%        tolerance based on L(1);
% help_wset - settling amplitude used in rd;
% wm - loop maintenance range wm=wp-wset;
% rd - rate of decay rd=(A(wp)-A(wset))/wm; Atention, this is module!!
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
global counter_variable counter_f err_frac err_frac_rd L A v0 vop vh vl
vp vset help_vset vm help_vm rd
% Inputs

% dB conversion %
Adb(1,:)=10*log10(A(counter_variable,counter_f,:));
% measures dependent on L(v0)
L_temp(1,:) = L(counter_variable,counter_f,:);
[~,ind_Lp] = max(L_temp); % Lp replaced by ~, because it isn't used
vp(counter_variable,counter_f,1) = v0(ind_Lp);
% settling amplitude
err = L_temp(1)*err_frac;
vset_array = v0(v0 < vp(counter_variable,counter_f,1) & abs(L_temp-
L_temp(1)) >= err);
vset(counter_variable,counter_f,1) = vset_array(1);
% loop maintenance range
vm(counter_variable,counter_f,1) = vp(counter_variable,counter_f,1)-
vset(counter_variable,counter_f,1);
% measures dependent on A(v0)
[MaxA,ind_A] = max(Adb);
% rate of decay
err_rd = L_temp(1)*err_frac_rd;
help_vset_array = v0(v0 < vp(counter_variable,counter_f,1) & abs(L_temp-
L_temp(1)) >= err_rd);
help_vset(counter_variable,counter_f,1) = help_vset_array(1);
help_vm(counter_variable,counter_f,1) = vp(counter_variable,counter_f,1)-
help_vset(counter_variable,counter_f,1);
rd(counter_variable,counter_f,1) = (Adb(ind_Lp) - Adb(v0 ==
help_vset(counter_variable,counter_f,1)))/help_vm(counter_variable,counte
r_f,1);
% rd(counter_variable,counter_f,1) = (Adb(ind_Lp) - Adb(v0 ==
vset(counter_variable,counter_f,1)))/vm(counter_variable,counter_f,1);

vl_array = v0(v0 < v0(ind_A) & Adb-MaxA <= -3);
stop_1 = 0;
vl_index = length(vl_array);
while stop_1 == 0
    vl_temp = vl_array(1,vl_index);
    index_temp = find(v0 == vl_temp);
%     Cond1 = Adb(index_temp+1) > Adb(index_temp);
    Cond2 = (Adb(index_temp-1) < Adb(index_temp));
```

```
    Cond3 = (abs(Adb(index_temp)-Adb(index_temp-1)) < 0.1);
    if (Cond2 || Cond3)
        vl(counter_variable,counter_f,1) = vl_temp;
        stop_1 = 1;
    else
        vl_index = vl_index-1;
    end
end

vh_array = v0(v0 > v0(ind_A) & Adb-MaxA <= -3);
stop_2 = 0;
vh_index = 1;
while stop_2 == 0
    vh_temp = vh_array(1,vh_index);
    index_temp = find(v0 == vh_temp);
    Cond1 = Adb(index_temp+1) < Adb(index_temp);
    Cond2 = (abs(Adb(index_temp)-Adb(index_temp+1)) < 0.1);
    if (Cond1 || Cond2)
        vh(counter_variable,counter_f,1) = vh_temp;
        stop_2 = 1;
    else
        vh_index = vh_index+1;
    end
end

vop(counter_variable,counter_f,1) = vh(counter_variable,counter_f,1)-
vl(counter_variable,counter_f,1);
end
```

LA_display.m

```
function LA_display(filename)
% Function LA_display , restore variables,
% restores the graphics of A(w) and L(w)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Restores saved values
load(filename);
% build graphics

for counter_f = 1:length(f_total)
    txt=sprintf('%s %d %s %d%s','Window
Type:',mem_model,'Frequency',f_total(counter_f),'Hz');
    figure('Name',txt);
    v0(1,:) =
logspace(v0_min_total(counter_f),v0_max_total(counter_f),v0_size);
    if type_of_variable == 1
        variable_charac = D_total;
    elseif type_of_variable == 2
        variable_charac = w0_total;
    elseif type_of_variable == 3
        variable_charac = mu_total;
    elseif type_of_variable == 4
        variable_charac = Rratio_total;
    elseif type_of_variable == 5
        variable_charac = Ron_total;
    end
```

```
    subplot(2,1,1);
    for counter_variable = 1:length(variable_charac)
        a(1,:) = A(counter_variable,counter_f,:);
        semilogx(v0(1,:),10*log10(abs(a(:))),'r');
        hold on;
        vl_plot(1,:) = vl(counter_variable,counter_f,1);
        semilogx(vl_plot,10*log10(a(v0 == vl_plot)),'bo');
        vh_plot(1,:) = vh(counter_variable,counter_f,1);
        semilogx(vh_plot,10*log10(a(v0 == vh_plot)),'bo');
    end
    xlabel('v_0 (V)');
    ylabel('A(\omega) (dBW)');
    grid on;

    subplot(2,1,2);
    for counter_variable = 1:length(variable_charac)
        l(1,:) = L(counter_variable,counter_f,:);
        semilogx(v0(1,:),l(:),'r');
        hold on;
        vp_plot(1,:) = vp(counter_variable,counter_f,1);
        semilogx(vp_plot,l(v0 == vp_plot),'bo');
        vset_plot(1,:) = vset(counter_variable,counter_f,1);
        semilogx(vset_plot,l(v0 == vset_plot),'bo');
        help_vset_plot(1,:) = help_vset(counter_variable,counter_f,1);
        semilogx(help_vset_plot,l(v0 == help_vset_plot),'go');
    end
    xlabel('v_0 (V)');
    ylabel('L(\omega) (\surd W)');%^0^.^5
    grid on;
end
end
```

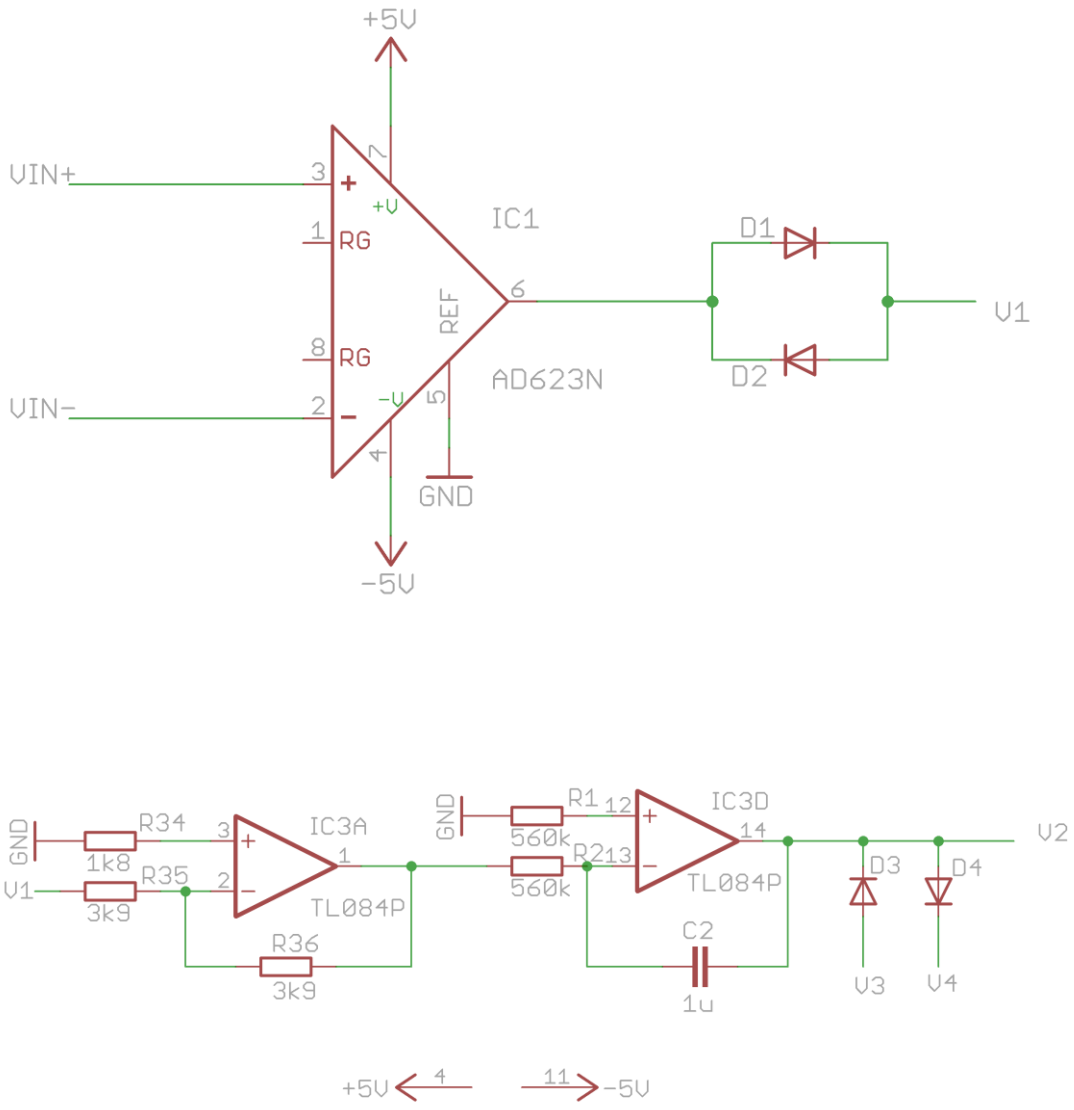
Display_Measures.m

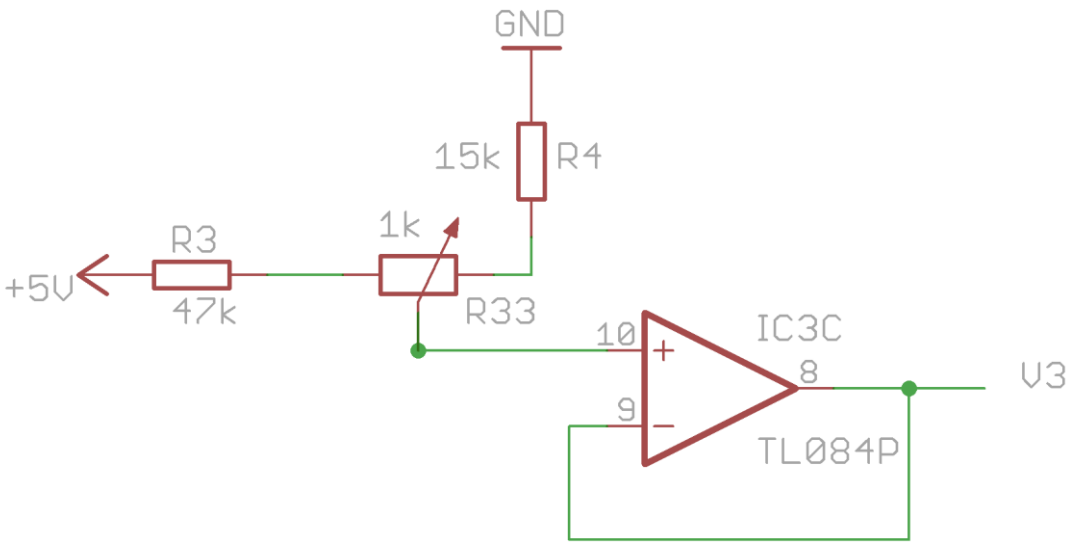
```
function Display_Measures(filename)
% Function Display_Measures , restore variables,
% restores the graphics of:
% wop - frequency range where A(v0) maintans amplitude above 3dB
% wp - peak frequency in L(v0)
% wset - settling amplitude, the amplitude at which the memristor
%        contracts into a linear resistance. Search method with 5%
%        tolerance based on L(1).
% wm - loop maintenance bandwidth wm=wset-wp
% rd - rate of decay rd=(A(wp)-A(wset))/wm
clc
load(filename);

if type_of_variable == 1
    x_plot = D_total;
    x_label_text = sprintf('D (m)');
elseif type_of_variable == 2
    x_plot = w0_total;
    x_label_text = sprintf('w_0 (m)');
elseif type_of_variable == 4
    x_plot = Rratio_total;
    x_label_text = sprintf('R_r_a_t_i_o');
```

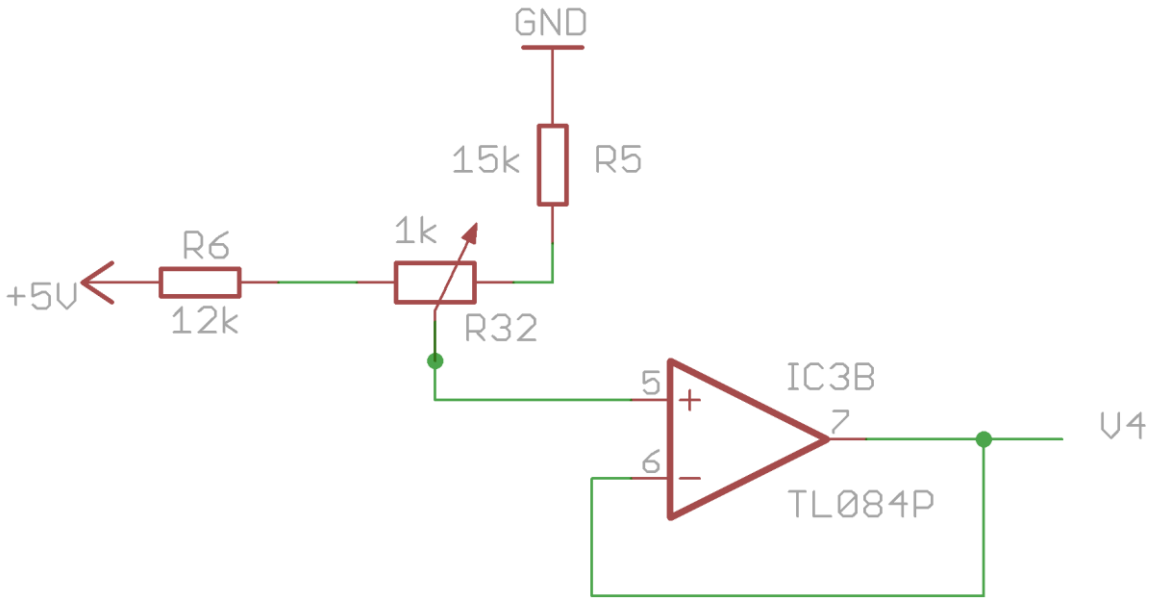
```
end

txt=sprintf('Window Type: %d, Measures of Amplitude', mem_model);
figure('Name',txt);
color = ['r','g','b'];
for counter_f = 1:length(f_total)
    subplot(2,2,1);
    vp_plot(1,:) = vp(:,counter_f,1);
    loglog(x_plot, vp_plot, color(counter_f));
    if type_of_variable == 3
        xlabel('\mu_v (m^2/(s\cdotV))');
    elseif type_of_variable == 5
        xlabel('R_o_n (\Omega)');
    else
        xlabel(x_label_text);
    end
    ylabel('v_p (V)');
    grid on;
    hold on;

    subplot(2,2,2);
    vset_plot(1,:) = vset(:,counter_f,1);
    loglog(x_plot, vset_plot, color(counter_f));
    if type_of_variable == 3
        xlabel('\mu_v (m^2/(s\cdotV))');
    elseif type_of_variable == 5
        xlabel('R_o_n (\Omega)');
    else
        xlabel(x_label_text);
    end
    ylabel('v_s_e_t (V)');
    grid on;
    hold on;

    subplot(2,2,3);
    vop_plot(1,:) = vop(:,counter_f,1);
    loglog(x_plot,vop_plot, color(counter_f));
    if type_of_variable == 3
        xlabel('\mu_v (m^2/(s\cdotV))');
    elseif type_of_variable == 5
        xlabel('R_o_n (\Omega)');
    else
        xlabel(x_label_text);
    end
    ylabel('v_o_p (V)');
    grid on;
    hold on;

    subplot(2,2,4);
    rd_plot(1,:) = rd(:,counter_f,1);
    loglog(x_plot,rd_plot, color(counter_f));
    if type_of_variable == 3
        xlabel('\mu_v (m^2/(s\cdotV))');
    elseif type_of_variable == 5
        xlabel('R_o_n (\Omega)');
    else
        xlabel(x_label_text);
    end
```

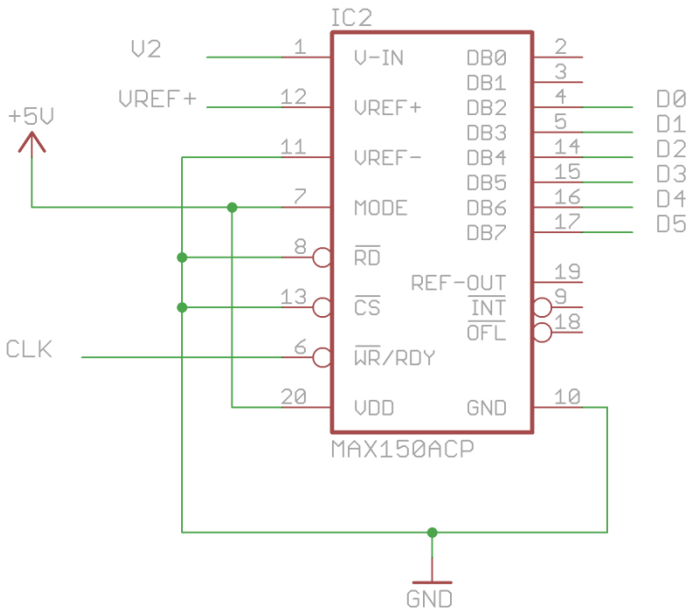
```
    ylabel('r_d (m^2/V)');
    grid on;
    hold on;
end
```

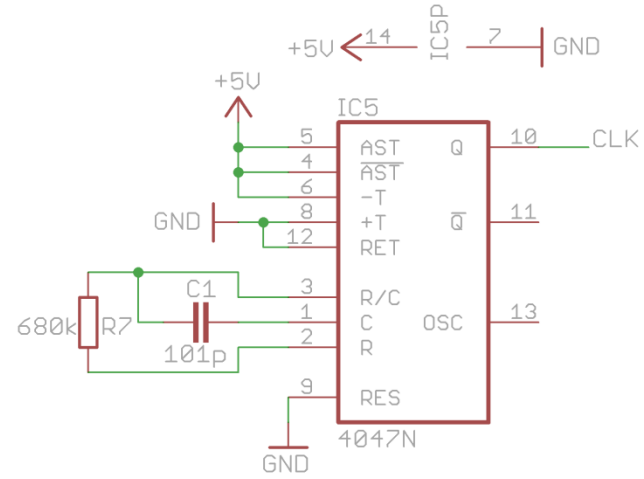
Schematics of the memristive device emulator

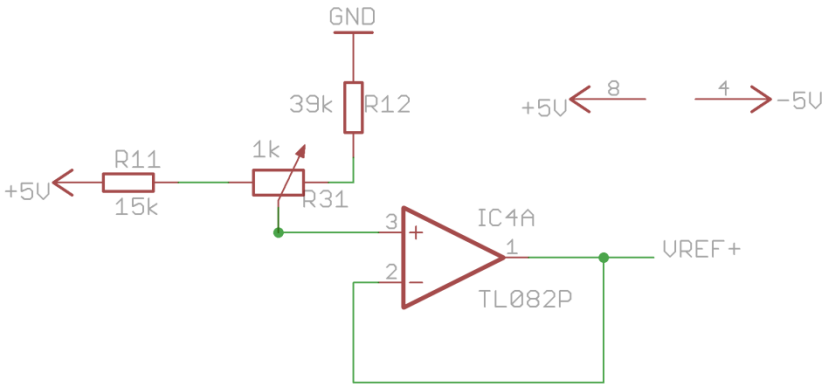


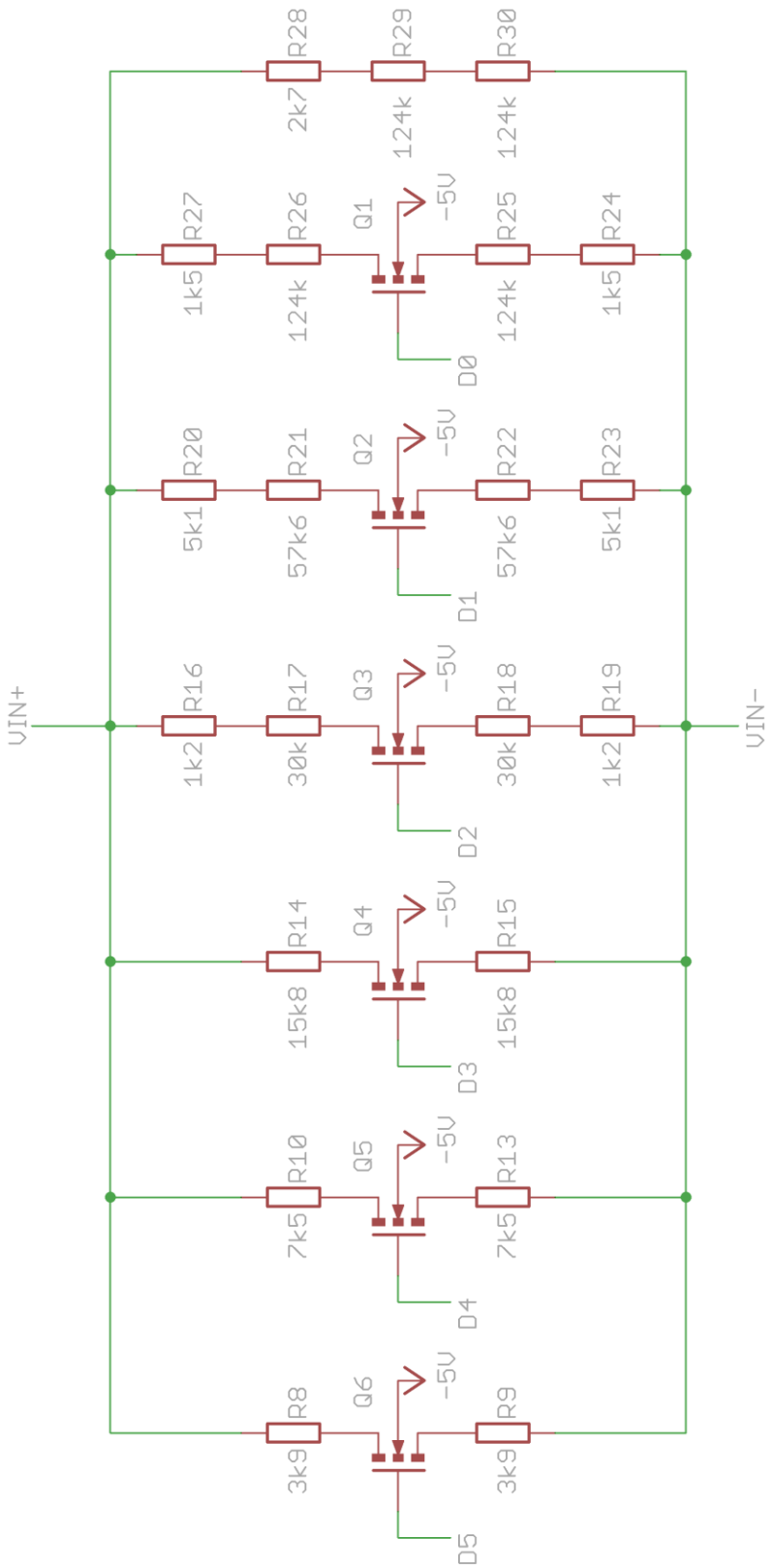

GND
15k
R4
1k
R3
+5V
47k
R33
10
IC3C
8
V3
9
TL084P

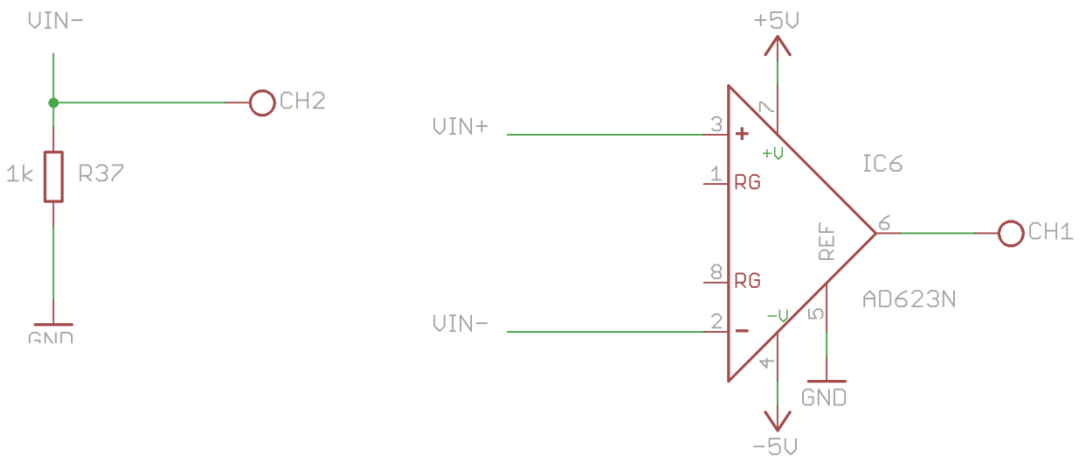

GND
15k
R5
1k
R6
+5V
12k
R32
5
IC3B
7
V4
6
TL084P


IC2
V2
VREF+
+5V
CLK
V-IN
VREF+
VREF-
MODE
RD
CS
WR/RDY
VDD
DB0
DB1
DB2
DB3
DB4
DB5
DB6
DB7
REF-OUT
INT
OFL
GND
MAX150ACP
D0
D1
D2
D3
D4
D5


+5V
IC5P
GND
IC5
AST
AST
-T
+T
RET
R/C
C
R
RES
Q
Q
OSC
CLK
680k
R7
C1
101p
4047N


GND
39k
R12
+5V
-5V
R11
15k
1k
R31
IC4A
TL082P
VREF+


VIN+
VIN-
R28
2k7
R29
124k
R30
124k
R27
1k5
R26
124k
Q1
-5V
R25
124k
R24
1k5
D0
R20
5k1
R21
57k6
Q2
-5V
R22
57k6
R23
5k1
D1
R16
1k2
R17
30k
Q3
-5V
R18
30k
R19
1k2
D2
R14
15k8
Q4
-5V
R15
15k8
D3
R10
7k5
Q5
-5V
R13
7k5
D4
R8
3k9
Q6
-5V
R9
3k9
D5


VIN-
CH2
1k
R37
GND
+5V
VIN+
3
+
7
+V
IC6
1
RG
6
REF
CH1
8
RG
AD623N
VIN-
2
-
-V
5
4
GND
-5V

Original Work

Frequency Characterization of Memristive Devices

João Capela Duarte[1], Ernesto Ventura Martins[1], Luis Nero Alves[1,2]
[1]Instituto de Telecomunicações, Aveiro Portugal
[2]Departamento de Eletrónica e Telecomunicações, Universidade de Aveiro
Aveiro, Portugal
joao.capela@ua.pt, evm@ua.pt, nero@ua.pt

***Abstract* – This paper proposes a new methodology suitable for frequency characterization of memristive devices (MDs) and systems. MDs are usually described by their associated hysteresis loops. Their distinctive memory properties stem from this unusual characteristic. Understanding the frequency behavior of these devices is of paramount importance for a multitude of different applications. This paper presents a morphological method, based on loop area and the length for the analysis of the frequency dependence of MDs. An example, considering thin film TiO_2 MDs reveals that the peak frequency (frequency where the loop has maximum area) of the device depends strongly on device dimensions and physical properties.**

Keywords – Memristive devices and systems, hysteretic loops, frequency characterization.

I. INTRODUCTION

Memristors and memristive systems have been introduced in the seventies by Leon O. Chua [1, 2]. In his first contribution to this theme [1], Chua identified a missing link between the four electric and magnetic variables. More specifically, considering the set of four independents variables, current (i), voltage (v), charge (q) and flux linkage (ϕ), there should be six formal two-by-two relations. Five out of these six relations were already known: two fundamental relations ($i=dq/dt$ and $v=d\phi/dt$) and three element definitions ($v=Ri$, $q=Cv$ and $\phi=Li$). The missing relation, establishing the connection between charge and flux, led to the introduction of the memristor ($\phi=Mq$). This missing relation has two possible interpretations: i) if the flux and charge are linearly related, the memristor behaves as a normal resistor; ii) if the flux and charge are not linearly related, then the memristor has distinctive properties, specifically, its current to voltage characteristic exhibits hysteresis behaviors. Hysteresis loops are often associated to elements exhibiting memory effects, in the memristor case, it translates into a resistance which is able to *"remember"* the current that has gone through it. In [2], Chua and Kang, extended the concept to memristive systems, which are systems exhibiting similar behaviors, without an explicit dependence on electric and magnetic variables.

More recently, memristive systems received a renewed allure due to the HP reported finding, published on the Nature journal, [3]. The HP MD was fabricated with a thin layer of TiO_2. This device exhibits a behavior similar to a voltage controlled memristor, displaying the characteristic pinched hysteresis loop. The authors in [3] present a mathematical model able to capture the distinctive features of the measured device. Further contributions on the modeling of TiO_2 MDs abound in the literature. Some of the most relevant contributions on this topic include references [3-6]. These contributions elaborate on the results of [3], adding to the discussion, modeling of boundary conditions and nonlinear dopant drift effects on the boundaries of the TiO_2 samples. Paper [6] suggests the use of window functions to mitigate these effects, while [4, 5] suggest alternative approaches.

Apart from the aforementioned achievements, understanding and characterizing the behavior of MDs still demands novel approaches. Particularly relevant is the topic of MD frequency response and frequency characterization. One of the foreseen applications of MDs, such as TiO_2 devices, is the development of new memory architectures, where data is stored as different resistance levels. Modern CMOS-based RAM technology, is able to provide memory devices with high density and fast access times. MDs have the undesirable property of degenerating into normal resistances, as the frequency increases [6-9]. This characteristic is generally dependent on the device physical parameters, but also on the stimulating signals (amplitude and frequency), as demonstrated in [2, 7, 8]. Evaluating the potentialities of these devices based on their frequency and switching characteristics is, in this sense, of paramount importance.

This paper proposes a morphological based method to address the frequency dependence characterization of MDs. The proposed approach analyses the area and length of the associated hysteresis loop. The ability of the device to maintain non-zero area as frequency increases is used as a figure of merit for frequency characterization. Loop length to discriminate the switching conditions of the MD. The paper is divided into five sections. Section II, presents the proposed methodology for area and length extraction and interpretation. Section III describes an application example based on TiO2 MD models. Section IV presents the achieved results using mathematic simulation. Section V draws the final conclusions.

II. PROPOSED CHARACTERIZATION METHODOLOGY

A generic model for a time invariant memristive one-port system involves three generic variables, x, y and a state variable w (for electrical circuits, these are i and v, for current controlled devices, or v and i, for voltage controlled devices, respectively). Following the axiomatic definition of Chua and Kang [2], a MD can be described by,

$$y = M(w)x \tag{1}$$

$$\frac{dw}{dt} = f(w,x) \tag{2}$$

where, $M(w)$ represents the memristance function, and $f(w,x)$ expresses the rate of change of the state variable w.

The analytical solution of (2) is possible for some special cases. In general the solution of (2) shows that both *x*, *y* and *w*, are periodic functions of time under special driving stimulus conditions. These conditions translate into both amplitude and frequency ranges of *x*. For moderate to low frequency regime, plotting *y* against *x*, reveals the characteristic hysteresis loop, depicted in fig. 1. An alternative representation resorts to the time dependent variables *x(t)*, *y(t)*, to which correspond the parametric representation of the loop, *C(x,y)=0*, also depicted in fig. 1. This hysteretic behavior is a manifestation of the memory property, revealing that *y* exhibits different values for the same value of the input *x*.

A possible method for frequency characterization of MDs consists of analyzing their hysteresis loop morphology against frequency. In this paper two morphological measures are adopted for this purpose: the loop area and the loop length [10].

A. Area of the Hysteresis Loop

Reporting to fig. 1, the area enclosed by the loop *C* can be determined using Green's theorem for two separated sections (represented as A_1 and A_2), which states that,

$$\iint_{A_k} \left(\frac{\partial F}{\partial x} - \frac{dG}{dy}\right) ds = \oint_{C_k} G dx + F dy \tag{3}$$

where, $k \in \{1,2\}$, C_k is the portion of *C* enclosing A_k, $ds=dxdy$ and *F* and *G* are two generic functions of *(x,y)*. The area $A_k(\omega)$ can be determined choosing *F* and *G* in such a way that the integrand of the left side of (3) becomes unity. For the present case, *F(x,y)=0* and *G(x,y)=-y*, resulting in,

$$A_k(\omega) = -\oint_{C_k} y dx \tag{4}$$

Since *C* is expressed parametrically by *(x(t), y(t))*, the contour integral in (4) reduces to,

$$A_k(\omega) = -\int_{(k-1)\frac{T}{2}}^{k\frac{T}{2}} \left[y \frac{dx}{dt}\right] dt \tag{5}$$

where, $T=2\pi/\omega$, represents the period of the input signal. The total area under *C* is just the sum of the absolute values of the areas under C_1 and C_2, that is, $A(\omega)=|A_1(\omega)|+|A_2(\omega)|$.

B. Length of the Hysteresis Loop

In order to measure the length of the hysteresis loop it will be necessary to integrate an infinitesimal length element, over the entire curve *C*. The length element can be expressed by $dl^2=dx^2+dy^2$. Dividing all by dx^2 and integrating over *C*,

$$L(\omega) = \int_{x_i}^{x_f} \sqrt{1 + \left(\frac{dy}{dx}\right)^2}\, dx \tag{6}$$

Using the parametric definition of *C* as before and performing the adequate variable changes, results in,

$$L(\omega) = \int_0^T \sqrt{\left(\frac{dx}{dt}\right)^2 + \left(\frac{dy}{dt}\right)^2}\, dt \tag{7}$$

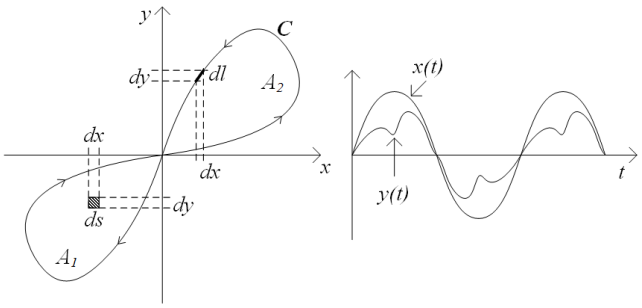


Fig. 1: Hysteresis loop and its parametric representation *(x(t), y(t))*.

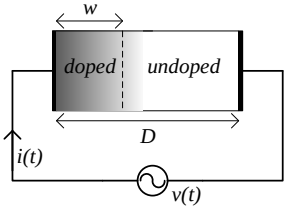


Fig. 2: TiO_2 memristor scheme, following [3].

The evaluation of equations (5) and (7) can benefit from the properties of MDs, in particular, the symmetry of the hysteresis loop. According to Chua's results, for time-invariant memristive one-ports the hysteresis loop has odd symmetry with respect to the origin [1, 2]. This implies that *x(t)* and *y(t)* also have symmetry properties which may be employed to simplify integration. For sinusoidal excitation, *x(t)* and *y(t)* are odd and even symmetric, respectively.

III. Application to Thin Film TiO_2 Memristive devices

Figure 2 depicts a simplified view of a thin film TiO_2 MD. The voltage *v(t)* impresses a current *i(t)* through the sample, modulating the boundary between doped and un-doped regions.

A. Device Modelling

The device model presented in [3], considers as state variable the position of the boundary, *w*, bounded between *0* and *D*, with *D* being the sample length. It also considers the dopant drift mobility, μ_v, the dopant polarity, η (to indicate if w increases or decreases with positive driving voltages, taking the values of *1* and *-1*, respectively), and the on and off state resistances, R_{on} and R_{off}, respectively. The model equations are,

$$i(t) = v(t)/M(x) \tag{8}$$

$$M(x) = R_{off} - (R_{off} - R_{on})x \tag{9}$$

$$\frac{dx}{dt} = \eta \frac{\mu_v R_{on}}{D^2} i(t) F(x) \tag{10}$$

where, $x=w/D$ and *F(x)* is a window function accounting for the non-linear nature of the dopant drift across the sample. It also forces *dx/dt* is zero at the boundaries *F(0)=F(1)=0*. There are several window functions models presented in the literature [3-6]. The present study, considers the window function presented by Prodromakis [4], defined by,

$$F(x) = k(1 - [(x - 0.5)^2 + 0.75]^p) \tag{11}$$

where, *k* and *p* are fitting parameters. The forgoing analysis assumes *k=1* and *p=10*.

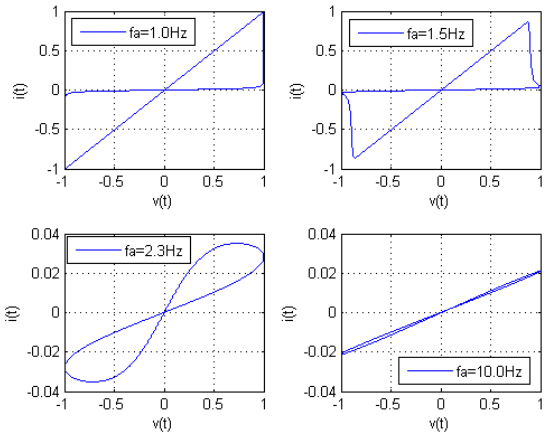


Fig. 3: Hysteresis loop for different frequencies.

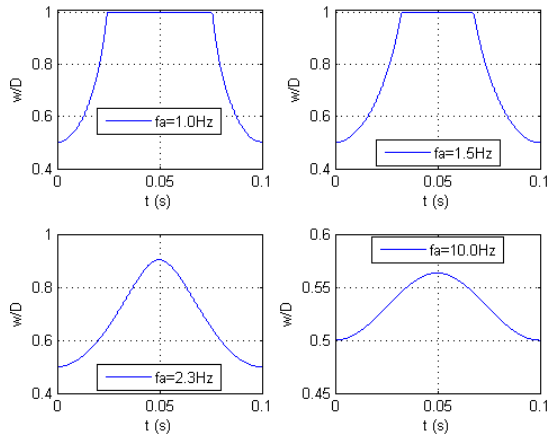


Fig. 4: Relative position of the interface, w/D, for different frequencies.

B. Model Behavior

To evaluate the performance of the TiO_2 MD model it is necessary to find *i* and *v*. This was accomplished following the procedure presented in [9], which resorts to the numerically integration of (10). For that purpose it was considered $v(t)=V_a sin(\omega_0 t)$ and constant integration step $dt=2\pi/(N\omega_a)$, with $N=10^6$ and $V_a=1$. The used model parameters were, $\mu_v=10^{-10} cm^2/Vs$, $\eta=1$, $D=10nm$, $w_0=5nm$ (initial position of the boundary), $R_{on}=1k\Omega$ and $R_{off}=100k\Omega$. Figures 3 and 4 show the qualitative behavior of the hysteresis loop for different frequency value, where both $i(t)$ and $v(t)$ were normalized with $i_0=V_a/R_{on}$ and V_a, respectively. For increasing values of the applied frequency, the loop changes from sharp to smooth behaviors. The sharp behavior, showing clearly the asymptotic values of R_{on} and R_{off}, is identified with hard switching conditions, occurring with w approaching the boundaries, as depicted on fig. 4. Decreasing further the frequency of the driving signal will lead to the partial collapse of the hysteretic behavior. For these cases, switching occurs only once, leaving the device in its on state, $M(w)=R_{on}$. For higher frequencies, the loop displays a smooth behavior as a result from soft switching conditions (when w takes values within the boundaries). Increasing further the frequency leads to the total collapse of the hysteretic behavior. For these cases, the device remains in its off state, with $M(w)=R_{off}$.

These behavioral changes can be captured from the plots of the loop area (converted to a decibel scale) and length versus frequency. Figure 5 shows the obtained plots for a frequency sweep ranging from *10mrad/s* to *1krad/s*. As it can be seen, the area and length plots area consistent with the above discussion. Under hard switching conditions the loop area assumes the maximum value, decreasing for both directions. The maximum length occurs for the hard switching limit where the loop changes from partial collapse to its sharpest form. For frequencies above this limit, the area and length decrease faster until the loop collapse. Possible characteristic measures are: i) the operational bandwidth, ω_{op}, defined as the frequency range where $A(\omega)$ maintains within *-3dB* from its maximum; ii) the peak frequency, ω_p, defined as the frequency of the maximum of $L(\omega)$; iii) the settling frequency, ω_{set}, defined as the frequency at which the loop collapses into a straight line (the MD collapses into a normal resistor); and iv) the rate of decay, r_d, defined as,

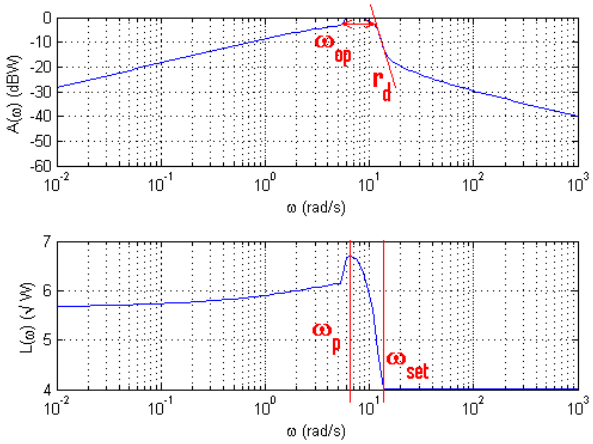


Fig. 5: Loop area and length versus frequency.

$$r_d = \frac{A(\omega_{set}) - A(\omega_p)}{\omega_{set} - \omega_p} \tag{12}$$

IV. Results and Discussion

Several simulations were performed in order to observe the dependencies of the aforementioned frequency response measures, ω_{bp}, ω_p, ω_{set} and r_d. All the simulations assumed as default TiO_2 model parameters the values presented on section III.B. The frequency response was evaluated following a parametric approach, changing *D*, w_0 and μ_v, one at a time for a set of *20* prescribed values. Both $A(\omega)$ and $L(\omega)$ were evaluated on a set of *200* logarithmic spaced frequencies. The achieved results are depicted on figures 6, 7 and 8. These results assume that it is possible to scale directly the TiO_2 model, against *D*, w_0 and μ_v. Although this assumption can lead to physical difficulties related to the fabrication of these devices for nanometer dimensions, it provides valuable guidelines for performance prediction, pertaining the behavior of MDs.

Figure 6 depicts frequency performance dependence against *D*, for values ranging from *1nm* to *10nm*. As it can be seen ω_{op}, ω_p and ω_{set}, decrease as *D* increases. This is consistent with what was presented in [3], the time taken to cross the MD length depends on *D*. On the contrary, r_d increases with *D*, showing that larger devices tend faster with frequency to the linear resistive behavior.

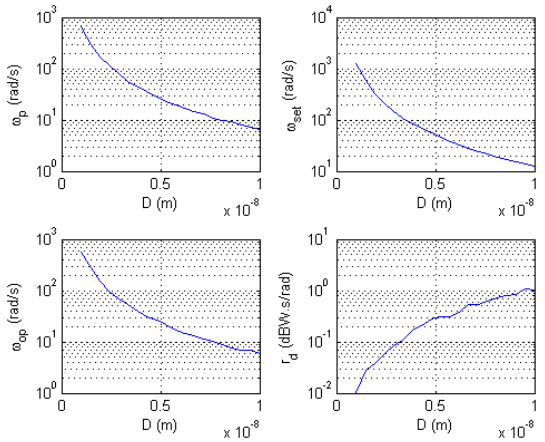


Fig. 6: Peak frequency, settling frequency, operational bandwidth and rate of decay against D.

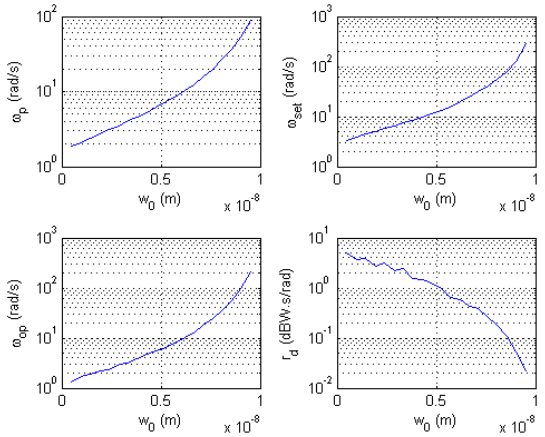


Fig. 7: Peak frequency, settling frequency, operational bandwidth and rate of decay against w_0.

Figure 7 displays the same frequency measures against the initial interface position, w_0, for values between *5%* and *95%* of D. The interface position between doped and undoped regions of TiO_2 MDs, plays a crucial role on their frequency behavior. As it can be seen ω_{op}, ω_p and ω_{set} increase with w_0, showing that the ability of the device to maintain is characteristic hysteretic behavior for wider bandwidths depends strongly on w_0. It is also observable that the limiting frequencies, ω_p and ω_{set} increase sharply for larger values of w_0. On the contrary, r_d decreases with w_0, showing that the transition to linear regime becomes faster for low values of w_0 and tend to slow for smaller ones.

Finally, figure 8 shows device dependence against the mobility factor. Assuming that it will be possible to tune the mobility in TiO_2 MDs. This can be achieved either, changing the dopant concentrations or selecting different materials. As it can be seen, ω_{op}, ω_p and ω_{set}, follow a linear trend with μ_v, showing that it is possible to design faster devices able to maintain their distinctive hysteretic behaviors for larger frequency ranges, using materials with larger mobility.

A curious observation is related to r_d. The achieved results show that the transition from hysteretic behavior to linear regime is faster for slower devices. This means, that devices able to operate for larger frequencies tend to maintain their memristive properties for wider frequency ranges.

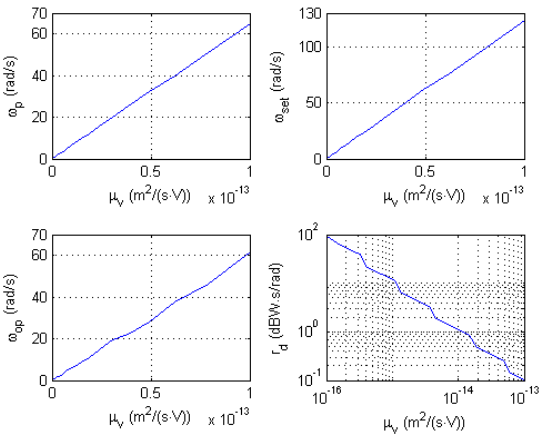


Fig. 8: Peak frequency, settling frequency, operational bandwidth and rate of decay against μ_v.

Further simulations revealed that ω_{op}, ω_p and ω_{set} decrease and r_d increases with the ratio R_{off}/R_{on}. The dependence was nearly logarithmic on R_{off}/R_{on} for all cases. These measures displayed negligible dependence on R_{on} with fixed R_{off}/R_{on}.

V. Conclusions

This paper presented a morphological approach to the frequency characterization of memristive devices. The proposed method extracts information from loop area and length frequency dependence of the device under test. The achieved results revealed important characteristics of TiO_2 devices, disclosing possible venues to optimize their frequency performance, in terms of maximum frequency and operational frequency ranges. Future work will explore the implementation of a suitable framework for frequency characterization of real devices, based on the proposed approach.

References

[1] L.O. Chua, "Memristor – The Missing Circuit Element", IEEE Transactions on Circuit Theory, Vol. CT-18, No. 5, September 1971.

[2] L.O. Chua, S.M. Kang, "Memristives Devices and Systems", Proceedings of the IEEE, Vol. 64, No. 2, February 1976.

[3] D.B. Strukov, G.S. Snider, D.R. Stewart, R.S. Williams, "The Missing Memristor Found", Nature Letters, Vol. 453, May 2008.

[4] T. Prodromakis, B.P. Peh, C. Papavassiliou, C. Toumazou, "A Versatile Memristor Model with Non Linear Dopant Kinetics", IEEE Transactions on Electron Devices, Vol. 59, No. 9, September 2011.

[5] F. Corinto, A. Ascoli, "A Boundary Condition-Based Approach to the Modeling of Memristor Nanostructures", IEEE Transactions on Circuits and Systems I: Regular Papers, Vol. 59, No. 11, November 2012.

[6] Y.N. Joglekar, S.J. Wolf, "The Elusive Memristor: Properties of Basic Electrical Circuits", European Jounal of Physics, Vol. 30, May 2009.

[7] Y.N. Joglekar, N. Meijome, "Fourier Response of a Memristor: Generation of High Harmonics with Increasing Weights", IEEE Transactions on Circuits and Systems II: Express Briefs, Vol. 59, No. .

[8] A.G. Radwan, M.A. Zidan, K.N. Salama, "On the Mathematical Modeling of Memristors", 22nd IEEE International Conference on Microelectronics, Cairo, Egypt, December 2010.

[9] S. Gazabare, R.J. Pieper, W. Wondmagegn, "Observations on Frequency Sensituvity of Memristors", 44th IEEE Southeastern Symposium on System Theory, Jacksonville, Florida, USA, March 2012.

[10] Z. Biolek, D. Biolek, V. Biolkova, "Computation of the Area of Memristor Pinched Hysteresis Loop", IEEE Trans. On Circ. Amd Syst. II: Express Briefs, Vol. 59, No. 9, September 2012.

Amplitude Characterization of Memristive Devices

João Capela Duarte[1,2], Ernesto Ventura Martins[2], Luis Nero Alves[1,2]
[1]Instituto de Telecomunicações, Aveiro Portugal
[2]Departamento de Eletrónica e Telecomunicações, Universidade de Aveiro
Aveiro, Portugal
joao.capela@ua.pt, evm@ua.pt, nero@ua.pt

***Abstract* – Memristive Devices (MDs) are usually described by their associated hysteresis loops. Their distinctive memory properties stem from this unusual characteristic, which depends on both stimulus frequency and amplitude. Understanding the behavior of these devices is of paramount importance for a multitude of different applications. This paper investigates the dependency of loop area and length on stimulus amplitude of MDs. The characterization methodology follows the morphological approach, introduced by the authors in [10], for frequency characterization. An example, considering thin film TiO_2 MDs reveals that the peak amplitude (amplitude where the loop has maximum area) of the device depends strongly on device dimensions and physical properties.**

Keywords – Memristive devices and systems, hysteretic loops, amplitude characterization.

I. INTRODUCTION

Memristors have been introduced in the early seventies by Leon O. Chua, as a missing link between the four electric and magnetic variables (current (*i*), voltage (*v*), charge (*q*) and flux linkage (φ)) [1]. This set of variables gives rise to six possible combinations pairs, from which the missing pair, establishing the connection between charge and flux, led to the introduction of the memristor ($\varphi=Mq$). When the flux and charge are linearly related, the memristor behaves as a normal resistor. However, if the flux and charge are not linearly related, then the memristor has distinctive properties, specifically, its current to voltage characteristic exhibits hysteresis behaviors. Chua's results were based on theoretical reasoning, by that time there was no memristor available. This led to the extension of the memristive systems concept by Chua and Kang, which, are systems exhibiting similar behaviors, without an explicit dependence on electric and magnetic variables [2].

In the last decade, memristive systems received renewed visibility due to HP's reported finding, published on the Nature journal, [3]. The HP MD was fabricated with a thin layer of TiO_2, and exhibits a voltage controlled memristor behavior, displaying the characteristic pinched hysteresis loop. The authors in [3] present a simple model able to capture the features of the measured device. Several contributions on the modeling of TiO_2 MDs followed [3-6]. These contributions elaborate on the results of [3], adding to the discussion, modeling of boundary conditions and nonlinear dopant drift effects of TiO_2 samples. Paper [6] suggests the use of window functions to mitigate these effects, while [4, 5] suggest alternative approaches based on Simmons tunneling model.

Apart from the aforementioned achievements, understanding and characterizing the behavior of MDs still demands novel approaches. Circuit design approaches often recur to device's figures of merit, such as cut-off frequencies, rise and fall times, 1dB compression and 3rd order intercept points, amongst others. Similar device figures of merit are lacking for Memristive devices. In particular the topic of MD response and characterization, both in frequency and amplitude is demanding novel approaches. MDs have the property of degenerating into normal resistances, as the amplitude decreases [6-9]. This characteristic is generally dependent on the device physical parameters, but also on the stimulating signals (amplitude and frequency), as demonstrated in [2, 7, 8]. Evaluating the potentialities of these devices based on their amplitude and switching characteristics is, in this sense, of paramount importance.

Similarly to the paper previously published by the authors [10], this paper proposes a morphological based method to characterize MDs, however, instead of studying the frequency dependence, this time the amplitude dependence is addressed. The proposed approach analyses the area and length of the associated hysteresis loop. The ability of the device to maintain non-zero area as the amplitude varies is used as a figure of merit for amplitude characterization. Just like in [10], loop length is used to discriminate the switching conditions of the MD. The paper is divided into five sections. Section II, presents the proposed methodology for area and length extraction and interpretation. Section III describes an application example based on TiO2 MD models. Section IV presents the achieved results using numerical simulation. Section V draws the final conclusions.

II. PROPOSED CHARACTERIZATION METHODOLOGY

A generic model for a time invariant memristive one-port system involves three generic variables, *x*, *y* and a state variable *w*. Following the axiomatic definition of Chua and Kang [2], a MD can be described by,

$$y = M(w)x \tag{1}$$

$$\frac{dw}{dt} = f(w,x) \tag{2}$$

where, *M(w)* represents the memductance function, and *f(w,x)* expresses the rate of change of the state variable *w*.

The analytical solution of (2) is possible for some special cases. In general the solution of (2) shows that both *x*, *y* and *w*, are periodic functions of time under special driving stimulus conditions. These conditions translate into both amplitude and frequency ranges of *x*. For a small window of frequency and amplitude values, plotting *y* against *x*, reveals the characteristic hysteresis loop, depicted in fig. 1. An alternative representation

resorts to the time dependent variables *x(t)*, *y(t)*, to which correspond the parametric representation of the loop, *C(x,y)=0*, also depicted in fig. 1. This hysteretic behavior is a manifestation of the memory property, revealing that *y* exhibits different values for the same value of the input *x*.

A possible method for amplitude characterization of MDs consists of analyzing their hysteresis loop morphology against amplitude. In this paper two morphological measures are adopted for this purpose: the loop area and length [10, 11].

A. Area of the Hysteresis Loop

Observing fig. 1, the area enclosed by the loop C can be determined using Green's theorem for two separated sections (A_1 and A_2),

$$\iint_{A_k} \left(\frac{\partial F}{\partial x} - \frac{\partial J}{\partial y}\right) ds = \oint_{C_k} J dx + F dy \tag{3}$$

where, $k \in \{1,2\}$, C_k is the portion of C enclosing A_k, $ds=dxdy$ and F and J are two generic functions of (x,y). The area $A_k(v_o)$ can be determined choosing F and J in such a way that the integrand of the left side of (3) becomes unity. For the present case, $F(x,y)=0$ and $J(x,y)=-y$, resulting in,

$$A_k(v_0) = -\oint_{C_k} y dx \tag{4}$$

knowing that x is a sinusoidal stimulus defined by,

$$x = v_0 \sin(\omega t) \tag{5}$$

using (1) and changing the variable x to $r=\sin(\omega t)$, the area $A_k(\omega)$ is given by,

$$A_k(v_0) = -v_0^2 \oint_{C_k'} G(w) r dr \tag{6}$$

where C_k' is a scaled version of C_k after the variable change. The total area under C is just the sum of the absolute values of the areas under C_1 and C_2, that is, $A(v_o)=|A_1(v_o)|+|A_2(v_o)|$. These result shows that the loop area is proportional to the square of the stimulus amplitude v_o.

B. Length of the Hysteresis Loop

The length of the hysteresis loop is simple the result of the integration of an infinitesimal length element, over the entire curve C. The length element can be expressed by $dl^2=dx^2+dy^2$. Dividing all by dx^2 and integrating over C,

$$L = \int_{x_i}^{x_f} \sqrt{1 + \left(\frac{dy}{dx}\right)^2}\, dx \tag{7}$$

Using (1), making the same change of variables as before and performing some calculations, results in,

$$L(v_0) = v_0 \int_{r_i}^{r_f} \sqrt{1 + (r\frac{dG(w)}{dr} + G(w))^2 dr} \tag{8}$$

which, shows that the length is proportional to the stimulus amplitude. The evaluation of equations (6) and (8) can benefit from the properties of MDs, in particular, the symmetry of the hysteresis loop. According to Chua's results, for time-invariant memristive one-ports the hysteresis loop has odd symmetry with respect to the origin [1, 2]. This implies that *x(t)* and *y(t)* also have symmetry properties which may simplify integration.

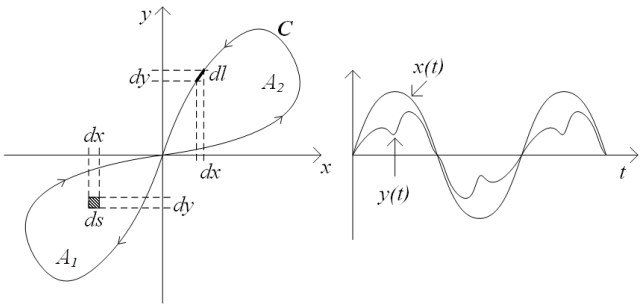


Fig. 1: Hysteresis loop and its parametric representation *(x(t), y(t))*.

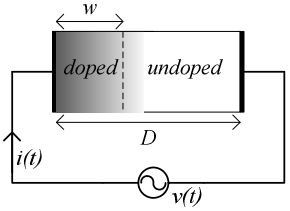


Fig. 2: TiO_2 memristor scheme, following [3].

III. Application to Thin Film TiO_2 Memristive Devices

Figure 2 depicts a simplified view of a thin film TiO_2 MD. The voltage *v(t)* impresses a current *i(t)* through the sample, modulating the boundary between doped and un-doped regions.

A. Device Modelling

The device model presented in [3], considers as state variable the position of the boundary, *w*, bounded between *0* and *D*, with *D* being the sample length. It also considers the dopant drift mobility, μ_v, the dopant polarity, η, and the on and off state resistances, R_{on} and R_{off}. The equations of the linear model used in this paper are,

$$i(t) = v(t)/M(x) \tag{9}$$

$$M(x) = R_{off} - (R_{off} - R_{on})x \tag{10}$$

$$\frac{dx}{dt} = \eta \frac{\mu_v R_{on}}{D^2} i(t) F(x) \tag{11}$$

where, $x=w/D$ and $F(x)$ is a window function accounting for the non-linear nature of the dopant drift across the sample. It also forces *dx/dt* to zero at the boundaries, $F(0)=F(1)=0$. There are several window functions models presented in the literature [3-6]. The present study, considers the window function presented by Prodromakis [4], defined by,

$$F(x) = k(1 - [(x - 0.5)^2 + 0.75]^p) \tag{12}$$

where, k and p are fitting parameters. The forgoing analysis assumes $k=1$ and $p=10$.

B. Model Behavior

To evaluate the performance of the TiO_2 MD model it is necessary to find *i* and *v*. This was accomplished following the procedure presented in [9], which resorts to the numerically

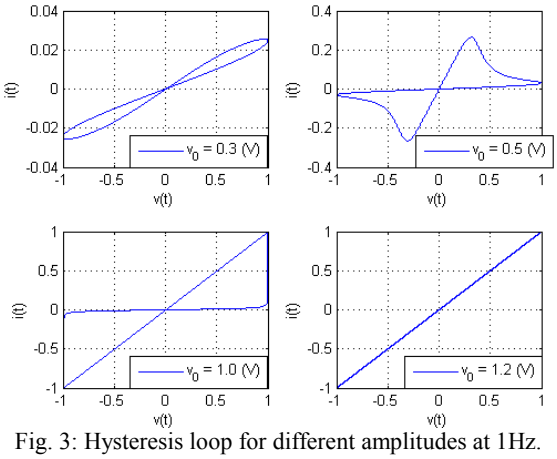


Fig. 3: Hysteresis loop for different amplitudes at 1Hz.

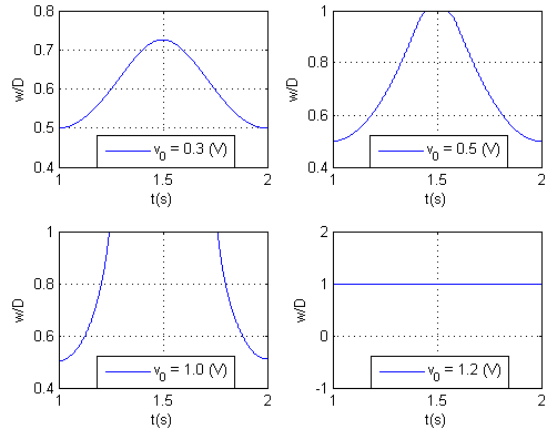


Fig. 4: Relative position of the interface, w/D, for the different amplitudes in fig 3.

integration of (11), furthermore, to get more accurate measurements, only samples after the first period of time were considered. For that purpose it was considered $v(t)=v_0 sin(\omega t)$ and constant integration step $dt=2\pi/(N\omega)$, with $N=2\times10^6$. The used model parameters were, $\mu_v=10^{-10} cm^2/Vs$, $\eta=1$, $D=10nm$, $w_0=5nm$ (initial position of the boundary), $R_{on}=1k\Omega$ and $R_{off}=100k\Omega$. Figures 3 and 4 show the qualitative behavior of the hysteresis loop for different amplitude values, for a fixed test frequency, where both $i(t)$ and $v(t)$ were normalized with $i_0=v_0/R_{on}$ and v_0, respectively. For increasing values of the applied amplitude, the loop changes from smooth to sharp behaviors. The sharp behavior, showing the asymptotic values of R_{on} and R_{off}, is identified with hard switching conditions, occurring with w approaching the boundaries, as depicted on fig. 4. Increasing further the amplitude of the driving signal will lead to the collapse of the hysteretic behavior. For these cases the device is in its on state, $M(w)=R_{on}$. For lower amplitudes, the loop displays a smooth behavior as a result from soft switching conditions (when w takes values within the boundaries). Further decreasing the amplitude leads to the total collapse of the hysteretic behavior. For these cases, the device remains in its off state, with $M(w)=R_{off}$.

These behavioral changes can be captured from the plots of the loop area (converted to a decibel scale) and length versus amplitude. Figure 5 shows the obtained plots for an amplitude sweep ranging from *0.1V* to *10V* (it is implicit that such high voltages may imply device destruction. However, it is assumed

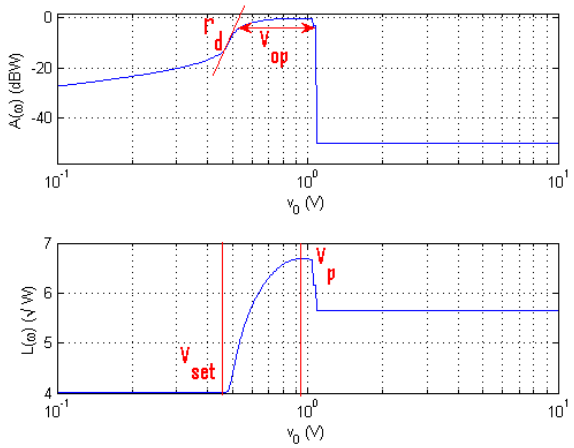


Fig. 5: Loop area and length versus amplitude.

for demonstration purposes that such behaviors are not present. Furthermore, the simple TiO_2 model does not include such effects, which may be present on real devices. Practical exploitation of this characterizing methodology may reveal such limitations). As it can be seen, the area and length plots area consistent with the above discussion. Under hard switching conditions the loop area assumes the maximum value, decreasing for both directions. The maximum length occurs for the hard switching limit where the loop changes from hard switching to partial collapse. For lower amplitudes, in the soft switching behavior zone, the area and length decrease sharply until the loop collapse. Possible characteristic measures are: i) the dynamic range, v_{op}, defined as the amplitude range where $A(v_0)$ maintains within *-3dB* from its maximum; ii) the peak amplitude, v_p, defined as the amplitude of the maximum of $L(v_0)$; iii) the settling amplitude, v_{set}, defined as the amplitude at which the loop collapses into a straight line (the MD collapses into a normal resistor); and iv) the rate of decay, r_d, defined as,

$$r_d = \frac{A(v_{set}) - A(v_p)}{v_{set} - v_p} \tag{14}$$

IV. Results and Discussion

Several simulations were performed in order to observe the dependencies of the aforementioned amplitude response measures, v_{op}, v_p, v_{set} and r_d. All the simulations assumed as default TiO_2 model parameters the values presented on section III.B, to check if the behavior of the MD model was consistent, independently of the frequency of the system used, three values of f were used: *1*, *100* and *1000Hz*. The amplitude response was evaluated following a parametric approach, changing D, w_0 and μ_v, one at a time for a set of *20* prescribed values. Both $A(v_o)$ and $L(v_o)$ were evaluated on a set of *300* logarithmic spaced frequencies. The achieved results are depicted on figures 6, 7 and 8. Like the previous study [10], these results assume that it is possible to scale directly the TiO_2 model, against D, w_0 and μ_v. Although this assumption can lead to physical difficulties related to the fabrication of these devices for nanometer dimensions, it provides valuable guidelines for performance prediction, pertaining the behavior of MDs.

Figure 6 depicts amplitude performance dependence against D, for values ranging from *1nm* to *10nm*. As it can be

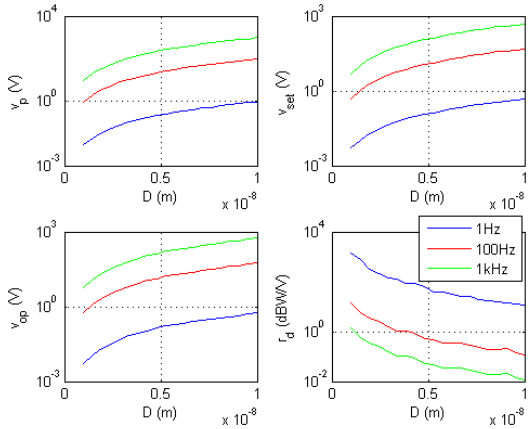


Fig. 6: Peak amplitude, settling amplitude, dynamic range and rate of decay against D.

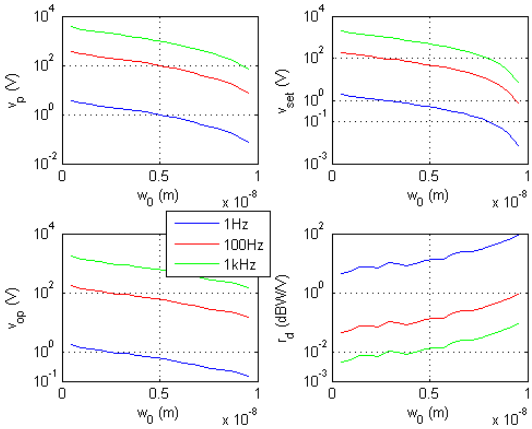


Fig. 7: Peak amplitude, settling amplitude, dynamic range and rate of decay against w_0.

seen v_{op}, v_p and v_{set}, increase as D increases. On the contrary, r_d decreases with D, showing that smaller devices exhibit hysteretic behaviors at lower stimulus amplitudes.

Figure 7 displays the same amplitude measures against the initial interface position, w_0, for values between *5%* and *95%* of D. The interface position between doped and undoped regions of TiO_2 MDs, plays a crucial role on their amplitude behavior. As it can be seen v_{op}, v_p and v_{set} decrease with w_0, showing that the ability of the device to maintain its characteristic hysteretic behavior for wider amplitudes depends strongly on w_0. It is also observable that the limiting amplitudes, v_p and v_{set} decrease sharply for larger values of w_0. On the contrary, r_d increases with w_0, showing that the transition from linear to hysteretic regimes becomes faster for higher values of w_0 and tend to slow down for smaller ones.

Finally, figure 8 shows device dependence against the mobility factor. Assuming that it will be possible to tune the mobility in TiO_2 MDs. This can be achieved either, changing the dopant concentrations or selecting different materials. As it can be seen, v_{op}, v_p and v_{set}, have an exponential decrease with the growth of μ_v, showing that devices that use materials with faster mobility can keep their distinctive hysteretic behaviors for lower amplitude ranges of input.

Unlike the other measures, r_d increases with the mobility. This result shows that the transition from linear to hysteretic behavior is faster for faster devices. This means, that devices able to operate with smaller amplitudes tend to maintain their

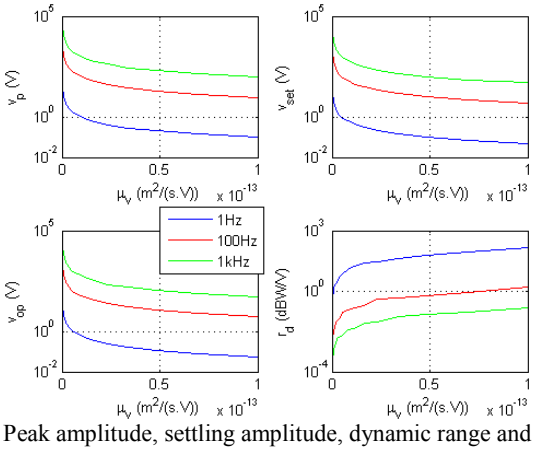


Fig. 8: Peak amplitude, settling amplitude, dynamic range and rate of decay against μ_v.

memristive properties for wider amplitude ranges. Simulations revealed that v_{op}, v_p and v_{set} increase and r_d decreases with the ratio R_{off}/R_{on}. The dependence was nearly linear on R_{off}/R_{on} for the first 3 cases and logarithmical for r_d. These measures displayed no dependence on R_{on} with fixed R_{off}/R_{on}.

V. CONCLUSIONS

This paper presented a morphological approach to the amplitude characterization of memristive devices. The proposed method extracts information from loop area and length amplitude dependence of the device under test. The achieved results revealed important characteristics of TiO_2 devices, disclosing possible venues to optimize their amplitude performance, in terms of stimulus amplitude ranges.

REFERENCES

[1] L.O. Chua, "Memristor – The Missing Circuit Element", IEEE Trans. on Circ. Theory, Vol. CT-18, No. 5, Sep. 1971.

[2] L.O. Chua, S.M. Kang, "Memristive Devices and Systems", Proc. of the IEEE, Vol. 64, No. 2, Feb. 1976.

[3] D.B. Strukov, G.S. Snider, D.R. Stewart, R.S. Williams, "The Missing Memristor Found", Nature Letters, Vol. 453, May 2008.

[4] T. Prodromakis, B.P. Peh, C. Papavassiliou, C. Toumazou, "A Versatile Memristor Model with Non Linear Dopant Kinetics", IEEE Trans. on Elect. Dev., Vol. 59, No. 9, Sep. 2011.

[5] F. Corinto, A. Ascoli, "A Boundary Condition-Based Approach to the Modeling of Memristor Nanostructures", IEEE Trans. on Circ. and Sys., Vol. 59, No. 11, Nov. 2012.

[6] Y.N. Joglekar, S.J. Wolf, "The Elusive Memristor: Properties of Basic Electrical Circuits", European Jounal of Physics, Vol. 30, May 2009.

[7] Y.N. Joglekar, N. Meijome, "Fourier Response of a Memristor: Generation of High Harmonics with Increasing Weights", IEEE Trans. on Circ.s and Sys. II., Vol. 59, No. 11, Nov 2012.

[8] A.G. Radwan, M.A. Zidan, K.N. Salama, "On the Mathematical Modeling of Memristors", 22nd IEEE Intern. Conf. on Microelectronics, Cairo, Egypt, Dec. 2010.

[9] S. Gazabare, R.J. Pieper, W. Wondmagegn, "Observations on Frequency Sensituvity of Memristors", 44th IEEE South. Symp. on Sys. Theory, Jacksonville, Florida, USA, Mar. 2012.

[10] J. C. Duarte, E. V. Martins, and L. N. Alves, "Frequency Characterization of Memristive Devices", in 21st Euro. Conf. on Circ. Theory and Des., 2013 (accepted).

[11] Z. Biolek, D. Biolek, V. Biolkova, "Computation of the Area of Memristor Pinched Hysteresis Loop", IEEE Trans. On Circ. and Syst. II, Vol. 59, No. 9, Sep. 2012.