university of groningen

faculty of science and engineering

mathematics and applied mathematics

Networks of Memristive Devices as Building Blocks for Neuromorphic Computing

Master's Project Applied Mathematics

April 2019

Student: M.A. Huijzer

First supervisor: Dr.ir. B. Besselink

Second assessor: Dr. A. E. Sterk

Abstract

In this thesis we give an introduction to neuromorphic computing, a new computing paradigm. We suggest memristive devices as potential building blocks for this new computing paradigm. To motivate that, we exhibit four different mathematical models which describe the mathematical behavior of a memristive device. Here, we make a link between memristive devices and spike-timing dependent plasticity. Furthermore, we will consider networks of memristive devices and develop tools for modelling and analysis of their external behavior. We will see that the dynamical behavior of networks of memristive devices is different from that of a single memristive device. This suggests that proper design of the network structure can be used to achieve desired memristive behavior needed for using these devices as building blocks for neuromorphic computing.

Contents

1 Introduction

1.1 Background

Nowadays, almost all of us are in possession of electronic devices such as computers and mobile phones. We are excited about the handy apps and tools of our devices and most of us use the Internet on a daily basis. However, not many people are aware of the large amount of energy consumed by these devices and the Internet. To give an example, doing one search by Google consumes the same amount of energy as a 60 Watt light to burn for 17 seconds [1], not even taking into account the energy consumed by the device itself. On top of that, large amounts of energy are consumed by the data centers which are used to store digital data. This is a motivation for looking into options to make our devices and data centers more energy efficient. In addition to this, we are always searching for manners to improve the processing speed of our devices.

Currently, the most recognized electronic device is the computer; a device build out of networks of transistors which use Boolean algebra. Improving the processing speed of a computer can be done by increasing the number of these transistors. Gordon Moore, the co-founder of Intel, laid out his prediction that the number of transistors on integrated processing units would double every two years, leading to exponential increases in processing speed of computers [2], known as Moore's law. This increasing transistor count will not only increase the complexity of the computer, but it will also require more space and energy. Until now, the increasing transistor count is compensated by the development of faster transistors of a decreasing size. However, the transistor is approaching physical limits to further miniaturization [3], and this complex network of transistors is not very energy efficient as laid out before. This motivates the search for an alternative computing paradigm which has both a high energy efficiency as well as a high processing speed.

A potential computing paradigm is neuromorphic computing, which is inspired by the biological concepts of the human brain and hopes to learn from the efficiency of it. This computing paradigm has potential to be used for the storage and processing of large amounts of digital information. This makes the paradigm not only interesting for usage in electronic devices as laptops, tablet, and mobile phones, but also for edge computing, i.e. a method for moving the control of data processing from centralized data centres to the last edge nodes of the Internet where data is collected and connected to the physical world [4]. Moving the control from data centers to edge devices, e.g. routers, will not only limit the necessary growth of data centres, but it will also save energy since neuromorphic computing devices are expected to be more efficient with data storage than the current computer architectures. In addition, the data will be stored nearer to the costumer which makes it cheaper to reach the data [4].

In order to create a computing paradigm based on the brain, the brain has to be studied more thoroughly. Open questions as in [5] should be addressed. In addition, physical building blocks should be developed for neuromorphic computing devices. Suggested building blocks are memristive devices [4], [6], [7], i.e. resistors with memory storage. These devices are more energy efficient than the transistors used in the current computing architectures because of their passivity property. In addition, their dynamical structure gives them potential to mimic the brain. A drawback of the use of memristors is that there is relatively little known about these circuits elements and how they can be efficiently used in a circuit. The coupling between the memristors in a network namely still consumes a lot of energy [8]. Therefore, research needs be done to manners how we can couple memristors, which mimic the human brain.

1.2 Contribution

In this thesis, we will introduce memristive devices as potential building blocks for neuromorphic computing. In order to create optimal neuromorphic computing devices, the memristive devices should have a certain dynamical behavior. This behavior might not be achieved by a single memristive device. Therefore, in this thesis we will consider networks of memristive devices and we will develop tools for modelling and analysis of their external behavior.

This thesis is organized as follows. Chapter 2 will introduce the idea of neuromorphic computing by briefly explaining biological neural networks and linking this to artificial neural networks. In Chapter 3 we will introduce memristive devices and we will exhibit different mathematical models which can be used to describe the behavior of such a device. Here, we will also make a link between memristive devices and synaptic plasticity in the human brain.

In Chapter 4 we will introduce some mathematical and electrical network properties, these will be utilized in Chapter 5 where we study the dynamical behavior of networks of memristive devices. Finally, Chapter 6 depicts simulation results of the external behavior of networks of memristive devices.

2 Neuromorphic computing

Neuromorphic computing is the name of a computing paradigm that is inspired by the human brain. It hopes to learn from the efficiency of the human brain to utilize it for tasks as data analysis. To give some more insight into these ideas, the following chapter will spend time on the functionality of the human brain to the extend it is known. This will be the starting point for introducing neural networks, as known from computer science, to eventually sketch the idea of neuromorphic computing.

2.1 Biological neurons and networks

Neuromorphic computing is inspired by key aspects of biological neurons and networks. Therefore this section, based on [9], [6] [10], will give an introduction to the functioning of the human brain relevant for understanding neuromorphic computing. We will do so by first considering the connection between two neurons to later extend this to a network of neurons.

Consider two neurons connected by a synapse, illustrated in Figure 1. Both neurons consist of a cell body, the so-called soma, and have incoming dendrites and an outgoing branched axon. The dendrites bring information to the neuron, where the axon sends information to other neurons. The cell body controls how many of the information received from the dendrites will be send towards the axon; this depends on the strength of the information. This information comes in the form of neurotransmitters, a form of chemicals, and are regarded as electrical pulses. These pulses, send by the cell body of a pre-synaptic neuron towards a post-synaptic neuron are called action potentials or spikes. The number of pulses send by a neuron depends on its membrane potential, the difference between the voltages at the in- and outside of the cell membrane. When the pre-synaptic membrane potential exceeds a certain threshold, it will send a pre-synaptic spike through one of its axon branches towards the synaptic cleft, a small gap that is adjacent to another neuron. The large voltage potential of this spike causes a variety of membrane channels to open and close allowing many ionic and molecular substances to flow. Among these substances are "packages" of neurotransmitters which are kept in small sacs, called synaptic vesicles. At moments of large voltage potentials, these synaptic vesicles inside the pre-synaptic cell fuse with the membrane in such a way that the "packages" of neurotransmitters are released in the synaptic cleft. The neurotransmitters are then collected by the post-synaptic cell and change the post-synaptic membrane potential.

The effect of the arriving spike on the post-synaptic membrane potential does depend on the properties of the synapse. An excitatory pulse increases the post-synaptic membrane potential, where an inhibitory pulse decreases the post-synaptic membrane potential. Both excitatory and inhibitory

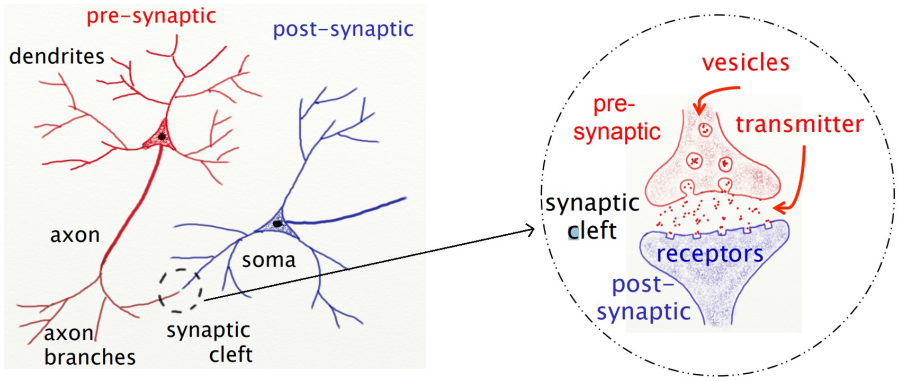

Figure 1: Schematic illustration of neurons (pyramidal cells) and their connections [9].

synapses can have varying strength (or weight) $\boldsymbol{w}$, but a synapse is either excitatory or inhibitory and this does not change through time. The weight $\boldsymbol{w}$ determines the efficacy of a pre-synaptic neuron in contributing to the membrane potential of the post-synaptic neuron; it can be interpreted as the number of neurotransmitter packages released during a pre-synaptic spike. The post-synaptic neuron rests when its membrane potential is sub-threshold and it fires spikes when the potential exceeds threshold.

Mathematically, the membrane potential at neuron i can be described as the weighted sum of incoming activities of the surrounding pre-synaptic neurons as

$$\sum_j \boldsymbol{w}_{ij}(t) S_j(t),$$

where $S_j(t)$ represents the mean activity of neuron j at time t, and $\boldsymbol{w}_{ij}(t)$ is the synaptic weight between neuron i and j. The mean activity of a neuron i, $S_i(t)$, represents whether neuron i is at rest or firing spikes. It can for example be chosen as the rate of generated spikes by neuron i per millisecond, as depicted in Figure 2. The synaptic weight between neuron i and j is given by

$$\boldsymbol{w}_{ij}(t) \begin{cases} > 0 & \text{excitatory synapse,} \\ = 0 & \text{no connecting synapse,} \\ < 0 & \text{inhibitory synapse.} \end{cases}$$

Considering discrete time-steps, S_i depends non-linearly on the mean activity of the other neurons as

$$S_i(t+1) = h\left(\sum_j \boldsymbol{w}_{ij}(t) S_j(t)\right),$$

where $h(\cdot)$ is a sigmoidal activation function. Figure 3 shows an example of a sigmoidal activation function, i.e. $S_i(t+1) = \tanh\left(\gamma(x_i(t) - \theta)\right)$ where $x_i(t) = \sum_j \boldsymbol{w}_{ij}(t) S_j(t)$. Here, the parameter

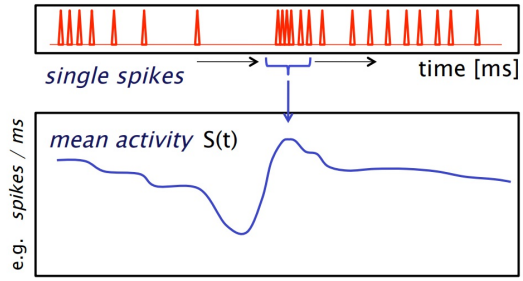

Figure 2: Schematic representation of how the activity of a neuron $S_i(t)$ can depend on its spike pattern [9].

γ represents the gain or steepest slope of the curve and θ gives the value of $x_i(t)$ at which the steepest slope is achieved. The parameter θ is sometimes referred to as the threshold of the activation function, butnote that this does not directly corresponds to the threshold value in the spike generation. The sigmoidal activation function is bounded by two horizontal asymptotes. The lower

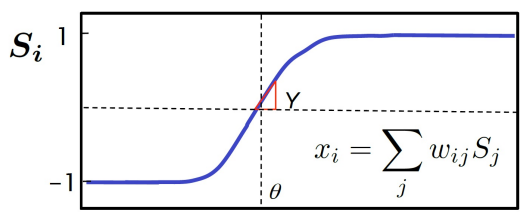

Figure 3: Example of a sigmoidal activation function with gain γ and threshold θ [9].

asymptote corresponds to minimal activity of a neuron, the neuron being at rest, where the upper asymptote correspond to maximal activity, the maximal number of spikes generates every second. Figure 2 shows an example of a sigmoidal activation function with $S_i = -1$ as the lower and $S_i = 1$ as the upper asymptote. The values of these asymptotes can be changed by scaling the activation function to for example $S_i = 0$ for the resting state and $S_i = 1$ for maximal activity.

2.1.1 Spike-timing dependent plasticity

The synaptic weight $\boldsymbol{w}_{ij}$ between two neurons changes in time as a function of the spiking activity of pre- and post-synaptic neurons. In 1949, Donald O. Hebb found the relation that if neuron i and j fire at the same time then the excitatory synaptic strength $\boldsymbol{w}_{ij}$ increases [11]: "What wires together fires together". Following this idea, the change of the synaptic strength was described to be proportional to the product of the firing rates of the pre- and post-synaptic neurons, i.e.

$\Delta \boldsymbol{w}_{ij} \propto S_i S_j$. Spike-timing dependent plasticity (STDP) is a refinement of this Hebbian learning rule which takes into account the relative timing of individual pre- and post-synaptic spikes, and not their mean activity over time. In STDP the change in synaptic weight $\Delta \boldsymbol{w}$ is not expressed in terms of the mean activities S_i and S_j, but as a function of the arrival time of a pre-synaptic spike and the generation of a post-synaptic spike, i.e. the time difference between the post-synaptic spike at t_{pos} and the pre-synaptic spike at t_{pre}. The difference between these times is given by $\Delta T = t_{\text{pos}} - t_{\text{pre}}$ as depicted in Figure 4. The update of the synaptic weight can be expressed as a function of ΔT as $\Delta \boldsymbol{w} = \xi(\Delta T)$. The shape of the function $\xi(\Delta T)$ can be interpolated from

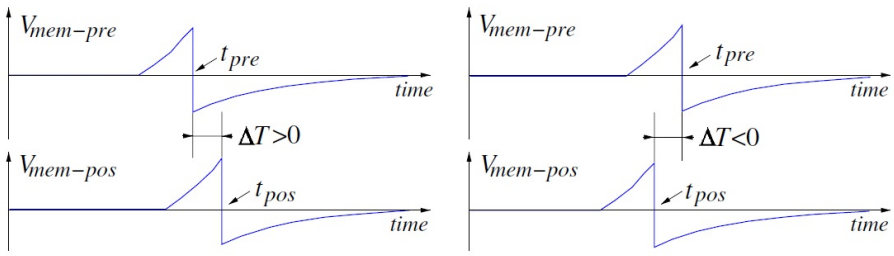

Figure 4: Pre- and post-synaptic membrane voltages, i.e. $V_{\text{mem-pre}}$ and $V_{\text{mem-post}}$, for the situations of positive and negative ΔT [6].

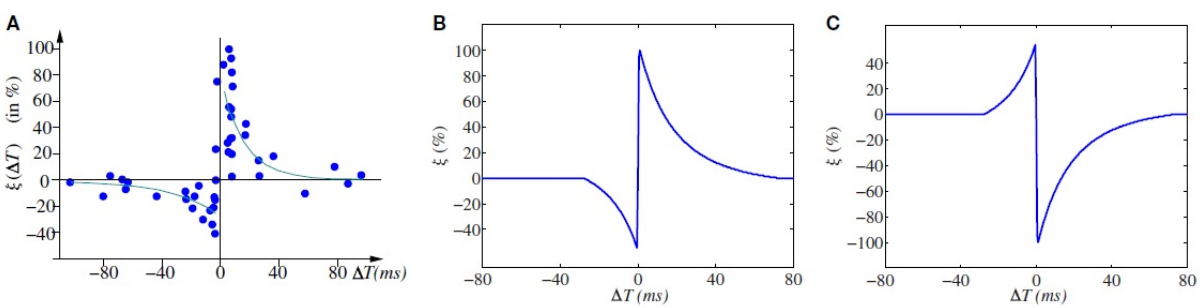

Figure 5: **A** Experimentally measured STDP function $\xi(\Delta T)$ on biological synapses [12]. **B** STDP update function for excitatory synapses, i.e. $\boldsymbol{w} > 0$. **C** STDP update function for inhibitory synapses, i.e. $\boldsymbol{w} < 0$. [6].

experimental data as shown in Figure 5. Mathematically it can be described as

$$\Delta \boldsymbol{w} = \xi(\Delta T) = \begin{cases} \alpha_{\text{pos}} e^{-\Delta T/\tau_{\text{pos}}} & \text{if } \Delta T > 0, \\ -\alpha_{\text{neg}} e^{-\Delta T/\tau_{\text{neg}}} & \text{if } \Delta T < 0, \end{cases} \tag{1}$$

where α_{pos}, α_{neg}, τ_{pos}, $\tau_{\text{neg}} \in \mathbb{R}$ are positive constants. Consider Figure 5B, the STDP update function for excitatory synapses. For positive relative timing, ΔT, the pre-synaptic spike has an

important role in producing the post-synaptic spike, i.e. the strength of the synapse is increased. For negative relative timing, the pre-synaptic spike is irrelevant for the generation of the post-synaptic spike, i.e. the strength of the synapse is decreased. Furthermore, the change $\Delta \boldsymbol{w}$ is higher for small values of ΔT which corresponds to the idea of Donald O. Hebb, namely if two neurons fire at the same time then the excitatory strength of the synapse between them increases. In the case of inhibitory synapses the relation (1) should be reversed, see Figure 5C, but similar observations can be made for the reversed diagram. More information about STDP and its history can be found in [13].

2.2 Artificial neurons and networks

In the previous section the key aspects of biological neurons and networks have been considered. These aspects allowed for a mathematical formulation of neural activity and synaptic interactions. This enables us to simulate the dynamics of neurons and their synaptic connections with artificial neural networks. In the following, based on [9], simple recurrent and feed-forward neural networks will be discussed to give an idea of the field of artificial neural networks.

2.2.1 Recurrent neural networks

Consider a network assembled from artificial neurons connected by artificial synapses with weight $\boldsymbol{w}$ of which an example is shown in Figure 6. The network with very high and unstructured connectivity forms a dynamical system of neurons which influence each other through synaptic interaction. The manner in which this happens is based on how biological neurons influence each other. Namely, considering discrete time-steps, one obtains the activities of the neurons to be of

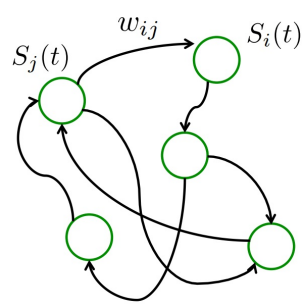

Figure 6: A networks of $N = 5$ neurons with unstructured partial connectivity [9].

the form

$$S_i(t+1) = g\left(\sum_j \boldsymbol{w}_{ij}(t) S_j(t)\right) \tag{2}$$

where $g(\cdot)$ is a sigmoidal activation function. Here, the sum $\sum_j \boldsymbol{w}_{ij}(t)S_j(t)$ ranges over all $j \in \{1, \ldots, N\}$ for which $w_{ij}(t) \neq 0$, where N is the number of neurons in the network. In words, the activity of a particular neuron depends on the synaptic strengths of the incoming edges. Since the activity of a particular neuron is given as a function of the preceding activities of the neurons, this network is called a recurrent neural network. The synaptic strength $\boldsymbol{w}_{ij}$ between neuron i and j may vary over time by adapting it following Hebbian learning, STDP, or another learning rule.

Figure 7 depicts an illustration of an application of recurrent neural networks. Data from a noisy initial configuration of neuron activities, $S(0) = \begin{pmatrix} S_1(0) & \ldots & S_N(0) \end{pmatrix}$ at time $t = 0$, is considered, the dynamics (2) generates a sequence of neuron activities which converge towards a clean configuration. This result is obtained since the weights $\boldsymbol{w}$ between the more active neurons in the configuration will be stronger where the weights $\boldsymbol{w}$ between the other neurons will become weaker, as discussed in Hebbian learning and STDP.

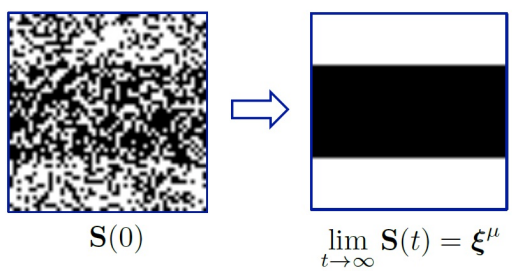

Figure 7: Illustration of a retrieval of noisy initial configuration of neuron activities [9].

2.2.2 Feed-forward neural networks

Another type of network architectures are feed-forward neural networks. In these networks, neurons are arranged in layers and information is processed in a well-defined direction. The network is determined via synaptic interactions and activations of the form

$$S_i^{(k)}(t) = g^{(k)}\left(\sum_j \boldsymbol{w}_{ij}^{(k)}(t) S_j^{(k-1)}(t)\right),$$

where $g^{(k)}$ represents the activation function of layer k and $S_j^{(k-1)}(t)$ represents the activity of neuron j in layer $k-1$. The sum $\sum_j \boldsymbol{w}_{ij}^{(k)}(t)S_j^{(k-1)}(t)$ ranges over all $j \in \{1, \ldots, N^{(k-1)}\}$ for which $\boldsymbol{w}_{ij}^{(k)}(t) \neq 0$, where $N^{(k-1)}$ is the number of neurons in layer $k-1$ of the network. This means

that the activity $S_i^{(k)}(t)$ of neuron i in layer k is only determined from the weighted sum of the activities in the previous layer $k-1$. As a consequence, a feed-forward neural network can be seen as a function that maps the vector of input activations to a single or several output units (neurons). Figure 8 shows an example of a four-layer feed-forward neural network. The first and last layer

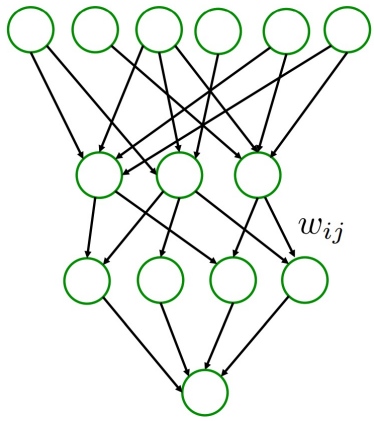

Figure 8: Schematic illustration of a four-layer feed-forward neural network [9].

represent the input and output layer of the network, respectively. The other two layers, which are neither input nor output layers, are called hidden layers. The illustration displays only two hidden layers and a single output unit (neuron), but the extension to a network with more hidden layers or a multiple output units is straightforward.

An application of a feed-forward neural network can be to classify vecotrs in discrete classes. Namely, if the output activities are discretized, for instance by a step function as activation function in the last layer, i.e. $S^{(k)} \in \{1, 2, \ldots, C\}$ where C_i represents a class, then the feed-forward neural network represents the classification of input activations $\begin{pmatrix} S_1^{(0)}(t) & \ldots & S_N^{(0)}(t) \end{pmatrix}$ into one of the C classes.

The accuracy and efficacy of a feed-forward neural network will depend on the number of hidden layers, the number of units per layer, the activation functions at the different layers, and the choice of the weights. The determination of these parameters might be done using an example data set with given input and outputs. Tests might be done for different values of the parameters and different activation functions after which the results should be compared. The term learning in neural networks is used for this adaptation or fitting process.

Note, that the recurrent and feed-forward neural networks depicted above are just simple networks inspired by the human brain. The human brain is more complex, it contains in the order of 10^{12} neurons and the network between them is highly connected, with each neuron having 1000

neighbours [9]. Also, there are still a lot of open questions about the functionality of the brain and the mathematical description of it [5].

To summarize, we exhibited two examples of artificial neural networks which are inspired by the functionality of the human brain. The usage of these computing algorithms which mimic the brain is referred to as neuromorphic computing. Possible physical building blocks for these artificial neural networks are the so-called memristive devices. They have the desired properties to mimic the STDP weight update rule as we will see in the next chapter.

3 Memristive devices

In the previous chapter, memristive devices were introduced as the potential building blocks for neuromorphic computing devices, i.e. the functionality of memristive devices can be used to mimic synapses in the human brain. In this chapter memristive devices will be introduced. In addition, an overview of different memristive device models as found in the literature will be given.

3.1 Memristor

In 1971, Leon O. Chua introduced a new two-terminal circuit element called the memristor [14]. This element is characterized by a relationship between the charge $q(t) = \int_{-\infty}^{t} i(\tau)d\tau$ and flux-linkage $\varphi(t) = \int_{-\infty}^{t} v(\tau)d\tau$, i.e. a link between the physical property of matter that causes it to experience a force when placed in an electromagnetic field and the strength of the electromagnetic field. The name memristor is a contraction for memory and resistance. This name is introduced because the memristor can be seen as a resistor whose resistance changes through time depending on the resistances in the past. Chua argued that the memristor can been seen as the fourth basic two-terminal circuit element, next to the resistor, capacitor and inductor, as depicted in Figure 9. All the four two-terminal elements are defined in terms of a relationship between two of the four

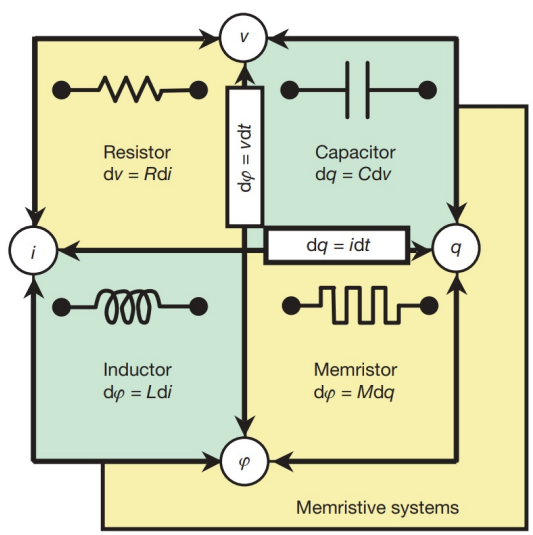

Figure 9: The four fundamental two-terminal circuit elements: resistor, capacitor, inductor and memristor [15]. Here, the voltage, current, charge and flux-linkage are represented by v,i, q and φ respectively.

fundamental circuit variables, the current i, voltage v, charge q and flux-linkage φ. Here, a two-terminal element or one-port can been seen as a black box connected with two nodes, i.e. terminals

to an outside circuit. The current flowing into the two nodes must be equal and opposite [16, Chapter 2].

The memristor is characterized by a relation of the type $g(q(t), \varphi(t)) = 0$. It is said to be charged-controlled (flux-controlled) if this relation can be expressed as a single valued function of the charge (flux-linkage). The voltage across a charge-controlled memristor is given by

$$v(t) = M(q(t))i(t),$$

where

$$M(q) = \frac{d\varphi(q(t))}{dt}.$$

Here, $M(q)$ has the unit of resistance, since resistance has the relationship $v(t) = Ri(t)$, it will henceforth be called the memristance. Similarly, the current of a flux-controlled memristor is given by

$$i(t) = W(\varphi(t))v(t),$$

where

$$W(\varphi(t)) = \frac{dq(\varphi(t))}{dt}.$$

In the above, $W(\varphi)$ has the unit of conductance, since conductance has the relationship $i(t) = \frac{1}{R}v(t)$, it will henceforth be called the memductance. Since the charge $q(t) = \int_{-\infty}^{t} i(\tau)d\tau$ and flux-linkage $\varphi(t) = \int_{-\infty}^{t} v(\tau)d\tau$ depend on the history of the current and voltage, the memristance and memductance indeed depend on the memory of the device.

3.2 Memristive device

In 1976, Leon O. Chua concluded that the memristor is only a special case of a much more general class of dynamical systems [17], called memristive devices, defined by

$$\dot{x}(t) = f(x(t), u(t), t), \tag{3}$$

$$y(t) = g(x(t), u(t), t)u(t), \tag{4}$$

where $u(t) \in \mathbb{R}$ and $y(t) \in \mathbb{R}$ define the input and output of the system, and $x(t) \in \mathbb{R}^n$ denotes the state with $x(t_0) = x_0$ as the initial condition. Both g and f represent continuous functions. We assume that (3) has an unique periodic solution with the same period as that of the input signal for any initial condition $x_0 \in \mathbb{R}^n$. In the special case that the one-port is time-invariant the memristive device is given by

$$\begin{aligned} \dot{x}(t) &= f(x(t), u(t)), \\ y(t) &= g\left(x(t), u(t)\right) u(t), \end{aligned} \tag{5}$$

where g and f are again continuous functions.

Having this general definition of a memristive device in mind, we consider two different types of time-invariant memristive devices, namely the charge- and flux-controlled memristive device. A charge-controlled time-invariant memristive device is represented by

$$\begin{aligned}\dot{x}(t) &= f(x(t), i(t)),\\ v(t) &= R(x(t), i(t))i(t),\end{aligned}$$

and a flux-controlled time-invariant memristive devices is given as

$$\begin{aligned}\dot{x}(t) &= f(x(t), v(t)),\\ i(t) &= G(x(t), v(t))v(t),\end{aligned}$$

where $i(t)$ denotes the current and $v(t)$ the voltage through the device. The functions R and G are defined to be continuous and represent the memristance and memductance of the devices, respectively. The function f represents a continuous function as before.

Examples of mathematical models of memristive devices will be treated in Section 3.4. First, some general properties of memristive devices will be discussed.

3.3 General properties

The memristive device has been introduced as a class of dynamical systems, however, it has some properties which distinguishes these systems from the wide range of dynamical systems. The most relevant properties are laid out in this section and more can be found in [17].

3.3.1 Hysteresis loop

As will be depicted in the simulation results in Section 3.4, memristive devices are characterized by a hysteresis loop going through the origin in an i, v diagram. The hysteresis loop of a memristive device can be explained by the fact that we assumed that (3) has an unique periodic solution with the same period as that of the input signal. Namely, consider a voltage-controlled (flux-controlled) memristive device having a periodic input signal i with the amplitude I_0 (v with the amplitude V_0), then since the input signal crosses each value $i_c \in [-I_0, I_0]$ ($v_c \in [-V_0, V_0]$) only twice during one period, and $x(t)$ is assumed to be an unique periodic solution with the same period as that of the input signal, there exist at most two distinct values for v (i) for each value of the input signal.

The fact that the hysteresis loop goes through the origin follows from (4); this equation ensures that the output y of the system is equal to zero whenever the input u is equal to zero. Figure 10A

illustrates a typical hysteresis loop going through the origin. Figure 10B cannot correspond to a memristive device, because there is a point i^* which corresponds to more than two distinct values for v. Furthermore, it can be shown that the area of the hysteresis loop of a memristive device decreases with the frequency f_c of the input signal, and tends to a straight line as $f_c \to \infty$, for all bipolar periodic signals and for all valid initial conditions [18].

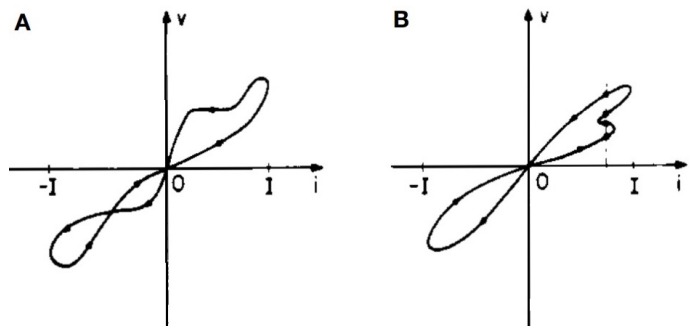

Figure 10: **A**. Illustration of the hysteresis loop of a current-controlled memristive device. **B**. Incorrect illustration the hysteresis loop of a current-controlled memristive device. [17]

3.3.2 Passivity

One remarkable property of the memristive device is that it is passive, i.e. it is possible to develop a memristive device without internal power supply [17]. This among other properties makes memristive devices an interesting option for building blocks for neuromophic computing, since these devices are more energy-efficient than devices with internal power supply. In order to understand this idea better, we will first introduce the concept of passivity based on [19], and subsequently prove that memristive devices are indeed passive.

Consider a time-invariant dynamical system Σ described by

$$\dot{x}(t) = f(x(t), u(t)),$$
$$y(t) = g(x(t), u(t)),$$

where f and g represent continuous functions, $x(t) \in \mathbb{R}^n$ is the state of the system with the initial condition $x(0) = x_0$, and $u(t) \in \mathbb{R}$ and $y(t) \in \mathbb{R}$ are the input and output of the system, respectively. Define $s(u(t), y(t)) : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ as the supply rate function of this system which represents the amount of energy supplied to the dynamical system at a certain time. An example of an supply rate function is given

$$s(u(t), y(t)) = u(t)y(t).$$

Furthermore, define $S : \mathbb{R} \to \mathbb{R}^+$ as the storage function which represents the stored energy in the system as a function of time. The energy balance of a system given the supply rate $s(u(t), y(t)) = u(t)y(t)$ is represented by

$$S(x(t_1)) = S(x(t_0)) + \int_{t_0}^{t_1} u(t)y(t)dt - \int_{t_0}^{t_1} d(t)dt. \tag{6}$$

Here, $d(t) \in \mathbb{R}$ indicates the dissipated energy from the system, which is negative if there is internal creation of energy in the system. This equation expresses that the stored energy at time t_1, $S(x(t_1))$, is equal to the stored energy at time t_0, $S(x(t_0))$, plus the total externally supplied energy during the time interval $[t_0, t_1]$, minus the energy dissipated during the that time interval. We say that a system is passive if there is no internal creation of energy possible, i.e. $d(t) \geq 0$ for all $t \in \mathbb{R}$. This leads to the following definition:

Definition 3.1. *A dynamical system of the form Σ is said to be passive if there exists a function $S : \mathbb{R} \to \mathbb{R}^+$, called the storage function, such that for all $x_0 \in \mathbb{R}^n$, all $t_1 \geq t_0$, and all input functions u,*

$$S(x(t_1)) \leq S(x(t_0)) + \int_{t_0}^{t_1} u(t)y(t)dt, \tag{7}$$

where $x(t_0) = x_0$ and $x(t_1)$ is the state of the system at time t_1 for the initial condition x_0 and the input function u.

The inequality (7) is called the dissipation inequality. In terms of electrical circuit elements passivity means that we are considering a circuit element which consumes energy but does not generate it.

It follows that a time-invariant current-controlled memristive device

$$\begin{aligned} \dot{x}(t) &= f(x(t), i(t)), \\ v(t) &= R(x(t), i(t))i(t), \end{aligned}$$

is said to be passive if there exists a function $S : \mathbb{R} \to [0, \infty)$, such that for all $x_0 \in \mathbb{R}$, all $t_1 \geq t_0$ and all input signals i,

$$S(x(t_1)) \leq S(x(t_0)) + \int_{t_0}^{t_1} i(t)v(t)dt.$$

Leon O. Chua described the following result on the passivity of the above depicted current-controlled memristive devices in [17].

Theorem 3.1. *A time-invariant current-controlled memristive device is passive if and only if $R(x, i) \geq 0$ for all input signals i, for all $t_1 \geq t_0$, and $R(x(t), i) = 0$ only if $i = 0$*[1].

[1]This assumption is made in order to not conflict with the hysteresis loop characteristic.

Proof. Assume that $R(x,i) \geq 0$ for all input signals $i(t)$ and all $t_1 \geq t_0$. It follows that

$$\int_{t_0}^{t_1} i(t)v(t)dt = \int_{t_0}^{t_1} R(x,i)i^2(t)dt \geq 0 \tag{8}$$

for all input signals $i(t)$ and all $t_1 \geq t_0$. Define

$$S(x) = \sup_{i(t),t_1 \geq t_0} -\int_{t_0}^{t_1} i(t)v(t)dt \tag{9}$$

as the storage function of the device. This function defines a storage function since it will always take a positive value (take $t_0 = t_1$ in (9)). Furthermore, by (8) it follows that $S(x) = 0$ for all input signals $i(t)$ and all $t_1 \geq t_0$. Hence, the energy balance (6) of the system is given by

$$\int_{t_0}^{t_1} d(t)dt = \int_{t_0}^{t_1} i(t)v(t)dt \geq 0$$

for all input signals i and all times $t_1 \geq t_0$. Since this result holds for every time-interval $[t_0, t_1]$, we must have that $d(t)$ is non-negative. This implies that there is no internal creation of energy possible in the system, hence the system is dissipative.

Assume that the device is passive and $R(x(t), i(t)) < 0$ for some input signal $i(t)$ on the time interval $[t_0, t_1]$. Let the storage function $S(x(t))$ be defined such that $x(t_0)$ is the state of minimum energy storage, i.e. $S(x(t_0)) = \min_{t \in \mathbb{R}} S(x(t))$ [20], hence

$$S(x(t_1)) \geq S(x(t_0)) \text{ for all } t_1 \geq t_0. \tag{10}$$

Our assumption that $R(x,i) < 0$ implies that

$$\int_{t_0}^{t_1} i(t)v(t)dt < 0. \tag{11}$$

Adding (11) to both sides of (10) gives

$$S(x(t_1)) \geq S(x(t_1)) + \int_{t_0}^{t_1} i(t)v(t)dt \geq S(x(t_0)) + \int_{t_0}^{t_1} i(t)v(t)dt \text{ for all } t_1 \geq t_0$$

which contradicts our assumption that the device is passive. We conclude that the current-controlled memristive device can only be passive if and only if $R(x(t), i(t)) \geq 0$ for all input signals i and all $t_1 \geq t_0$. □

3.3.3 Polarity

Figure 11 depicts the symbol of a memristive device. Here, we make the distinction between the positive (left) and negative (right) side of the device, called the polarity of the device. Due to this polarity, a memristive device will show different hysteresis loops depending on the direction of the

input current. Therefore, polarity is an important characteristic which need to be considered when coupling memristive devices, as shown in [18], [21]. In this thesis, we will include the polarity of a memristive device in the time-invariant memristive device model (5) as

$$\dot{x}(t) = f(x(t), pu(t)),$$
$$y(t) = g(x(t), u(t))u(t),$$

where $p = 1$ when the direction of the memristive device and input current are equal, and $p = -1$ when they are opposite.

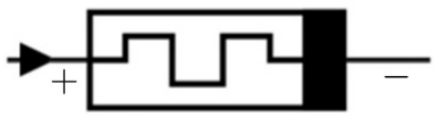

Figure 11: Symbol of a memristive device [22].

Note that the polarity of the memristive device should not be confused with the polarity, i.e. the sign, of the input signal applied to the device. Since the behavior of a memristive device depends on its history, applying a (slightly) different input signal will influence the hysteresis loop of the device. Hence, the polarity of the input signal will also change the hysteresis loop of the device.

3.4 Mathematical models

In the previous section, memristive devices and their general properties are introduced. In this section, various memristive device models will be exhibited, compared, and their relationship with the synaptic strength in the human brain will be discussed. We will start with an affine model and move towards memristive device models that satisfy the general properties of memristive devices while also mimicking the synaptic strength in the human brain following the STDP weight update rule.

3.4.1 Affine model

One of the simplest memristive device models is introduced in [21]. This model represents a flux-controlled memristive device where the memductance $W(\varphi(t))$ is an affine function of the flux-linkage $\varphi(t)$ through the device. The affine model is given by

$$\frac{d\varphi(t)}{dt} = v(t),$$
$$i(t) = W(\varphi(t))v(t),$$

where

$$W(\varphi(t)) = \alpha\varphi(t) + \beta,$$

and $i(t)$ is the current through the device and $v(t)$ the voltage drop across it. The parameter α determines the variation rate of the memductance and β is regarded as the initial memductance value. Figure 12 shows simulation results obtained for the affine model having the input $v(t) = 4\sin(2\pi t f_c + \pi/3)$ with the frequency $f_c = 20$ Hz. We note that this simulation result indeed depicts a hysteresis loop going through the origin as introduced in Section 3.3.1.

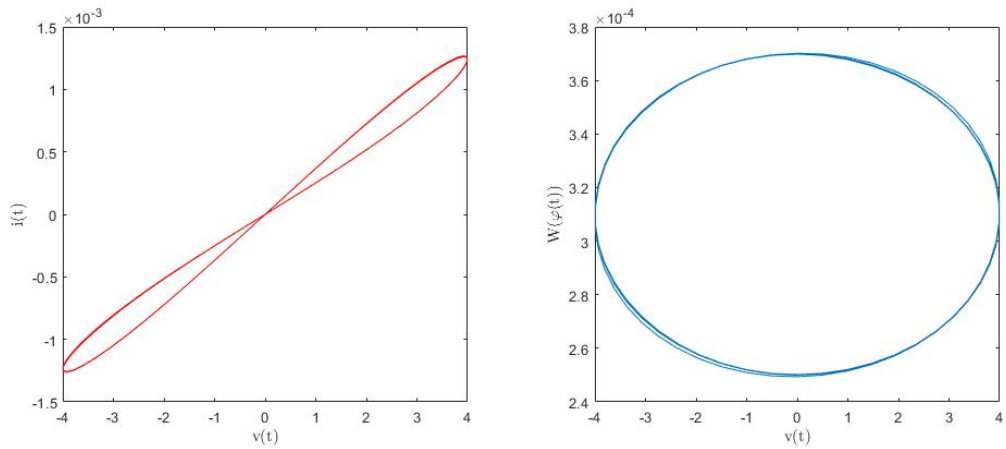

Figure 12: Results obtained for the affine model. Left, $v(t)$-$i(t)$ curves. Right, memductance compared with the memristor voltage. In this simulation, $\alpha = 1.884\ mS/W_b$, $\beta = 0.25\ mS$, and $\varphi_0 = 0$.

3.4.2 Moving wall model

The next memristive device model considered is the so called moving-wall model. This model was first introduced by scientists at HP labs in 2008 [15] and later widely used by [7], [23], [6], [10], and others.

The model described in [15] is a charge-controlled memristive device given by

$$\frac{d\boldsymbol{w}(t)}{dt} = \mu_v \frac{R_{\text{ON}}}{D} i(t), \tag{12}$$

$$v(t) = \left(R_{\text{ON}} \frac{\boldsymbol{w}(t)}{D} + R_{\text{OFF}} \left(1 - \frac{\boldsymbol{w}(t)}{D} \right) \right) i(t), \tag{13}$$

where $i(t)$ is the current through the device, $v(t)$ the voltage drop across it, and $\boldsymbol{w}(t)$ is a state variable. This model is based on the idea that the hysteresis property of the memristive device requires the device to have some sort of atomic rearrangement which modulates the electrical

current [15]. In order to realize this atomic rearrangement, we consider a thin semiconductor film

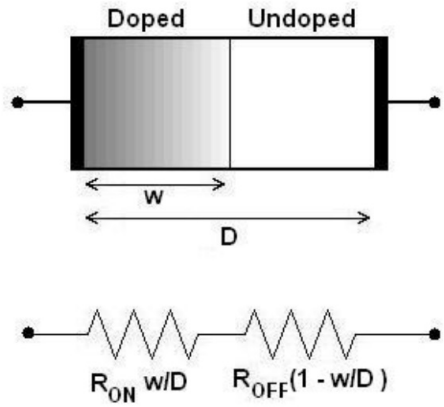

Figure 13: Illustration of a semiconductor film with regions having high and low concentration of dopants, the doped and undoped region respectively, and their corresponding time-varying resistances in series [23].

of thickness D sandwiched between two metal contacts, as shown in Figure 13. The film is separated by a moving wall into a region with a high concentration of dopants (for example positive ions) having a low resistance R_{ON} and a region with a low concentration of dopants having a higher resistance R_{OFF}. The applied current on the memristive device will move the wall which separates the two regions, i.e. change the variable $\boldsymbol{w}$, causing the dopants to drift. The total memristance of this device is dependent on the position of the wall and is determined as a combination of the two variable resistors connected in series, as shown in Figure 13, and is described by

$$M(\boldsymbol{w}(t)) = R_{\mathrm{ON}}\frac{\boldsymbol{w}(t)}{D} + R_{\mathrm{OFF}}\left(1 - \frac{\boldsymbol{w}(t)}{D}\right), \tag{14}$$

where we assume that $0 \leq \boldsymbol{w}(t) \leq D$ in order to ensure that $M(\boldsymbol{w}(t))$ is positive. This function can be expressed as a function of $q(t)$ by substituting $\boldsymbol{w}(t)$, obtained by (12) as

$$\boldsymbol{w}(t) = \mu_v \frac{R_{\mathrm{ON}}}{D} q(t) + \boldsymbol{w}_0,$$

into (14) which gives

$$M(q(t)) = (R_{\mathrm{ON}} - R_{\mathrm{OFF}})\left(\frac{\mu_v R_{\mathrm{ON}}}{D^2} q(t) + \frac{\boldsymbol{w}_0}{D}\right) + R_{\mathrm{OFF}}. \tag{15}$$

By our assumptions that $R_{\mathrm{OFF}} > R_{\mathrm{ON}}$ and $0 \leq \boldsymbol{w}(t) \leq D$ it follows that the memristance becomes larger for higher dopant mobilities μ_v and smaller D. Figure 14 depicts the hysteresis loop going through the origin and the memristance compared with the memristor current obtained for the moving-wall model having the input $i(t) = 10^{-3}\sin(2\pi t f_c)$ with $f_c = 1$ Hz.

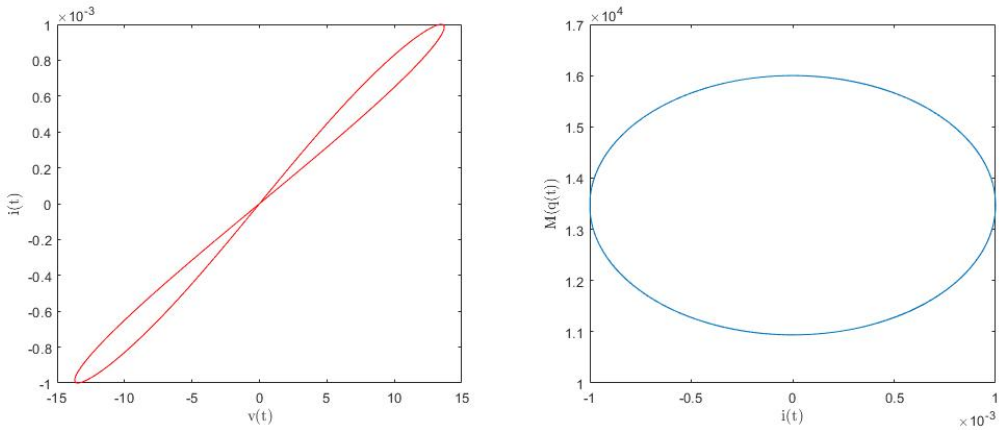

Figure 14: Results obtained for the moving-wall model. Left, $v(t)$-$i(t)$ curves. Right, memristance compared with the memristor current. In this simulation, $\mu = 10^{-10}\ cm^2/sV$, $D = 10\ nm$, $R_{\text{ON}} = 100\ \Omega$, $R_{\text{OFF}} = 16000\ \Omega$, and $x_0 = 0$.

3.4.3 Spike-timing dependent plasticity model

Keeping the previous two memristive device models in mind, we will make a link between memristive devices and STDP, this will be done based on [6] and [10].

Consider the flux-controlled memristive device described by

$$\frac{d\boldsymbol{w}}{dt} = f(\boldsymbol{w}, v),$$
$$i = g(\boldsymbol{w}, v)v,$$

where i is the current through the device, v the voltage drop across it, $\boldsymbol{w}$ is a state variable, and g is its memductance. For readability the dependence of i, v, and $\boldsymbol{w}$ on t is omitted here. The function f consists of two parts, f_{STDP} and f_{sat}, and may describe the ionic drift under electric fields, as we saw in (12), where the rate of change of $\boldsymbol{w}$ was modeled by a linear dependence on i. In reality it is more likely that this dependence grows exponentially after a certain threshold boundary. For the flux-controlled device we are considering here, this dependence can be described as

$$f_{\text{STDP}}(v) = \begin{cases} I_o \text{sign}(v) \left(e^{|v|/v_o} - e^{v_{\text{th}}/v_o} \right) & \text{if } |v| > v_{\text{th}}, \\ 0 & \text{otherwise,} \end{cases}$$

where v_{th} is the threshold boundary, and I_o, v_o are parameters which may depend on $\boldsymbol{w}$. The shape of this function is depicted in Figure 15A. Note that the polarity of the applied voltage is included in this function. Namely, if a sufficiently large positive voltage is applied to the memristive device then $\boldsymbol{w}$ will increase, and for sufficiently large negative voltage it will decrease.

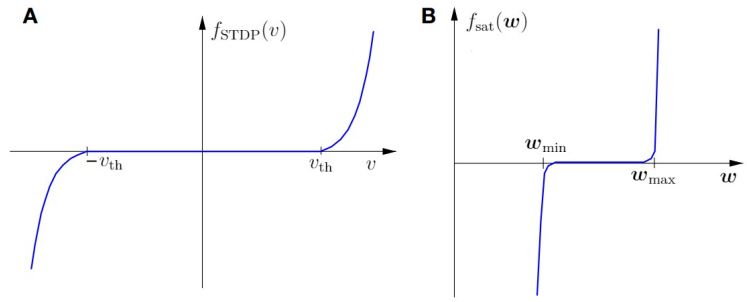

Figure 15: **A.** Weight update function with exponential growth and thresholds. **B.** Saturation function for limiting the range of the weight. [6]

This function f_{STDP} can be used to make a link between memristive devices and STDP. The relationship between those two is that the function f_{STDP} may describe the ionic drift under electric fields, i.e. the drift of dopants while moving the wall in Figure 14, where STDP represents drift of packages of neurotransmitters. To make this link precise, consider a neural spike with a shape of the type shown Figure 16, mathematically expressed as

$$\mathrm{spk}(t) = \begin{cases} A^{+}_{\mathrm{mp}} \frac{e^{t/\tau^{+}} - e^{-t^{+}_{\mathrm{ail}}/\tau^{+}}}{1-e^{-t^{-}_{\mathrm{ail}}/\tau^{+}}} & \text{if } -t^{+}_{\mathrm{ail}} < t < 0, \\ -A^{-}_{\mathrm{mp}} \frac{e^{-t/\tau^{-}} - e^{-t^{-}_{\mathrm{ail}}/\tau^{-}}}{1-e^{-t^{-}_{\mathrm{ail}}/\tau^{-}}} & \text{if } 0 < t < t^{-}_{\mathrm{ail}}, \\ 0 & \text{otherwise.} \end{cases} \tag{16}$$

During a time t^{+}_{ail}, the membrane voltage increases exponentially towards a peak amplitude A^{+}_{mp} after which the pre-synaptic neuron fires a spike towards the post-synaptic neuron. The membrane voltage then changes quickly towards the negative peak amplitude A^{-}_{mp} and returns smoothly to its resting potential during a time t^{-}_{ail}. The parameters τ^{-}_{ail} and τ^{+}_{ail} modify the curvature of the graph.

The memristive device voltage potential can then be expressed as a function of (16) and is given

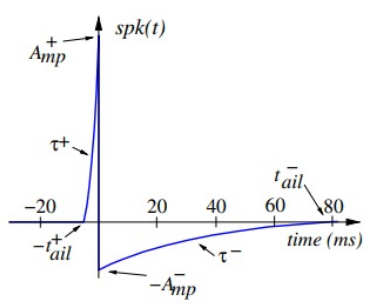

Figure 16: Illustration of the membrane voltage action potential [6].

$$v(t, \Delta T) = \alpha_{\text{pos}}\text{spk}(t) - \alpha_{\text{pre}}\text{spk}(t + \Delta T),$$

where α_{pos}, α_{pre} are parameters and $\Delta T = t_{\text{pos}} - t_{\text{pre}}$ as before in Section 2.1.1. The synaptic strength $\boldsymbol{w}$ update can be computed as

$$\frac{d\boldsymbol{w}}{dt} = f(v(t, \Delta T)) = \frac{d\xi(\Delta T)}{dt},$$

where $\xi(\Delta T)$ is given by (1).

So the weight $\boldsymbol{w}$ of a synapse between two neighboring neurons depends on its history and more explicitly on the total amount of neurotransmitters received from the pre-synaptic neurons. In similar fashion, the strength of a memristor, its memristance, depends on the amount of charge q or flux-linkage φ that flowed through it.

Having the relation between f_{STDP} and STDP in mind, we need to introduce a new function f_{sat} aimed to keep $\boldsymbol{w}$ inside the boundary $[\boldsymbol{w}_{\min}, \boldsymbol{w}_{\max}]$ and so keeping the memristance of the device between limits. The function f_{sat} is depicted in Figure 15B, and is mathematically described as

$$f_{\text{sat}}(\boldsymbol{w}) = \begin{cases} e^{\boldsymbol{w}_{\min}/w_o} - e^{|\boldsymbol{w}|/w_o} & \text{if } \boldsymbol{w} < \boldsymbol{w}_{\min}, \\ 0 & \text{if } \boldsymbol{w}_{\min} \leq \boldsymbol{w} \leq \boldsymbol{w}_{\max}, \\ e^{|\boldsymbol{w}|/w_o} - e^{\boldsymbol{w}_{\max}/w_o} & \text{if } \boldsymbol{w} > \boldsymbol{w}_{\max}. \end{cases}$$

The complete memristive device model, as described in [6], is then given by

$$\begin{aligned} \frac{d\boldsymbol{w}}{dt} &= \frac{1}{C}\left(f_{\text{STDP}}(v) - f_{\text{sat}}(\boldsymbol{w})\right), \\ i &= \frac{1}{k_r(\boldsymbol{w} + \boldsymbol{w}_o)} v, \end{aligned}$$

where C and k are constant parameters. The variable $\boldsymbol{w}$ in this model represents the weight of a synapse in artificial neural networks. In addition, if we rescale the variable $\boldsymbol{w}$ from the bounded interval $[\boldsymbol{w}_{\min}, \boldsymbol{w}_{\max}]$ to the interval $[0, D]$, then it represents the parameter of the wall position, as described in Section 3.4.2. Figure 17 shows simulation results obtained for the STDP model having the input $v(t) = \sin(2\pi t f_c)$ with $f_c = 10$ Hz. Again, this simulation result indeed depicts a hysteresis loop going through the origin as introduced in Section 3.3.1. Note that this model does not fully correspond to the model given in [6] because of another way of defining the function $f_{\text{sat}}(\boldsymbol{w})$.

3.4.4 Improved spike-timing dependent plasticity model

In addition to the STDP model we will introduce an improved STDP model as discussed in [24]. The authors of this paper claim that there are significant discrepancies between the models we described

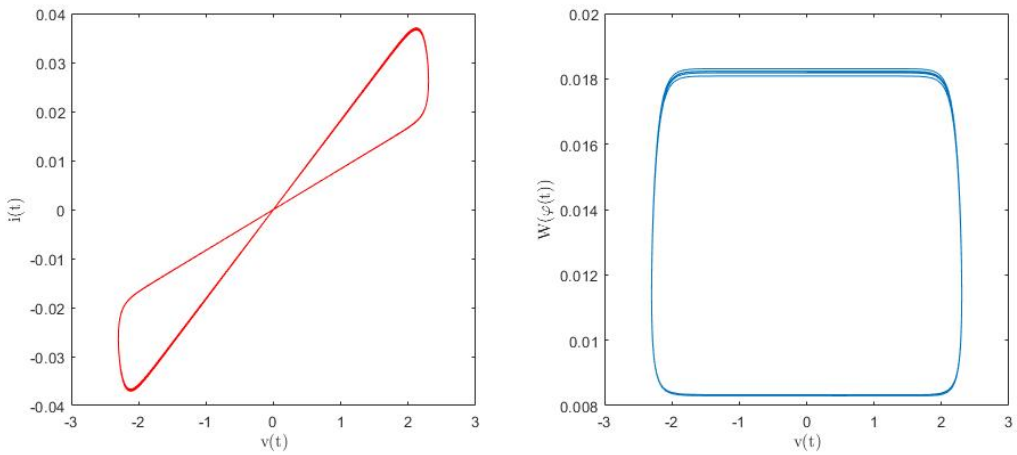

Figure 17: Results obtained for the STDP model. Left, $v(t)$-$i(t)$ curves. Right, memductance compared with the memristor voltage. In this simulation, $v_{\text{th}} = 1\ V$, $I_o = 10\ \mu A$, $v_o = 0.1\ V$, $\boldsymbol{w}_{\max} = 10\ V$, $\boldsymbol{w}_{\min} = -10\ V$, $k_r = 4.50\ A$, $w_o = 12.2\ V$, $C_{\text{MR}} = 10\ mF$, and $w_0 = 0$.

so far and published memristive device characterization data. Before, all simulation results for the different memristive device models showed symmetric $v(t)$-$i(t)$ curves, i.e. the motion of the state variable was equivalent regardless of whether it was moving in the positive or negative direction. However, published memristive device characterization data showed that the motion of the state variable depends on both its value and the polarity of the applied voltage [25]. The model presented here will take these different motions of the state variables into account. Also, this model will take into account that the MIM (metal-insulator-metal) structure of a memristive device imposes an increase in conductivity beyond a certain voltage threshold. In addition, the model uses many fitting parameters and therefore it can be used to simulate the effect of a wide class of devices.

In order to take the MIM structure of a memristive device into account, the $v(t)$-$i(t)$ relationship is given by

$$i(t) = \begin{cases} \alpha_1 x(t) \sinh bv(t) & \text{if } v(t) \leq 0, \\ \alpha_2 x(t) \sinh bv(t) & \text{if } v(t) < 0, \end{cases} \tag{17}$$

where α_1, α_2 and b are parameters used to fit different memristive device structures. The hyperbolic sinusoid depicted in Figure 18 is used to model an increase in conductivity beyond a voltage threshold b which is caused by the MIM structure of a memristive device. Devices appear to be more conductive when $v(t) > 0$ which is reflected in our model by choosing the parameters α_1 and α_2 accordingly. Note that the $v(t)$-$i(t)$ relation (17) does not satisfy the general memristive device model as introduces in Section 3.1. However due to the shape of the hyperbolic sinusoid, shown in

Figure 18, the $v(t)$-$i(t)$ curve still depicts a hysteresis loop going through the origin.

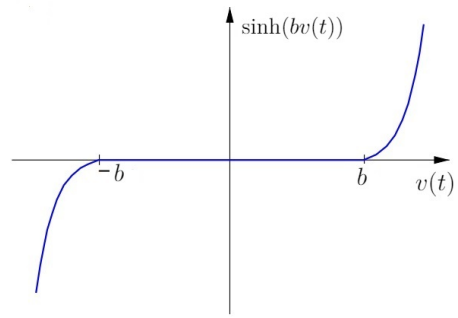

Figure 18: Illustration of the hyperbolic sinusoid $\sinh(bv(t))$. The hyperbolic sinusoid takes small values on the interval $[-b, b]$ and grows exponentially outside of that interval.

The $v(t)$-$i(t)$ relationship also depends on the state variable $x(t)$, which takes values between 0 and 1 and provides the change in memductance based on the physical dynamics of a device. The change in the state variable is based on two different functions $g(v(t))$ and $f(x(t), v(t))$ as

$$\frac{dx(t)}{dt} = g(v(t))f(x(t), v(t)).$$

The function $g(v(t))$ is included to impose that the state $x(t)$ does not change until a certain threshold is exceeded and is given by

$$g(v(t)) = \begin{cases} -A_n \left(e^{-v(t)} - e^{v_n}\right) & \text{if } v(t) < -v_n, \\ 0 & \text{if } -v_n \leq v(t) \leq v_p, \\ A_p \left(e^{v(t)} - e^{v_p}\right) & \text{if } v(t) > v_p. \end{cases}$$

In order to include the polarity of the applied voltage, v_p represents the threshold for positive $v(t)$ and v_n for negative $v(t)$. These thresholds are viewed as the minimum energy required to impose a change on the physical structure of the device [24]. The parameters A_p and A_n indicate how quickly the state changes once the threshold is surpassed.

The function $f(x(t), v(t))$ which ensures $x(t)$ to stay between 0 and 1 is described as

$$f(x(t), v(t)) = \begin{cases} e^{-\alpha_p(x(t)-x_p)} \left(\frac{x_p - x(t)}{1-x_p} + 1\right) & \text{if } x(t) \leq x_p, v(t) > 0, \\ 1 & \text{if } x(t) < x_p, v(t) > 0, \\ e^{-\alpha_n(x(t)+x_n-1)} \left(\frac{x(t)}{1-x_n}\right) & \text{if } x(t) \leq 1 - x_n, v(t) \leq 0, \\ 1 & \text{if } x(t) > 1 - x_n, v(t) \leq 0. \end{cases}$$

This function is constructed such that it remains constant until $x(t)$ reaches the point x_p or x_n after which $x(t)$ is limited. By picking x_p and x_n differently, this function provides the possibility to model the state variable dependent of the polarity of the applied voltage. The choice of x_p and x_n needed to keep $x(t)$ between 0 and 1 depends on the choice of the parameters in $g(v(t))$.

Figure 19 shows an example of simulation results obtained for the improved STDP model having the input $v(t) = \sin(2\pi t f_c)$ with $f_c = 3$ Hz. It depicts an hysteresis loop going through the origin, which is unlike the simulation results of the previous models not symmetric around the origin because this model takes the polarity of the input voltage into account. More examples of simulated devices can be found in [24] where it is shown that this model can fit different memristive devices structures. Also, the parameters in this model can be adjusted such that its state variable mimics the synaptic weight between two (artificial) neurons.

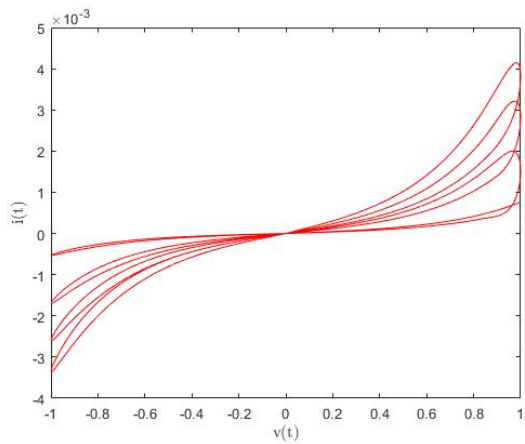

Figure 19: $v(t)$-$i(t)$ curves obtained for the improved STDP model. In this simulation, $v_p = 0.9\ V$, $v_n = 0.2\ V$, $A_p = 0.1$, $A_n = 10$, $x_p = 0.15$, $x_n = 0.25$, $\alpha_p = 1$, $\alpha_n = 4$, $a_1 = 0.076$, $a_2 = 0.06$, $b = 3$, and $x_0 = 0.001$.

To summarize, we have compared four different memristive device models. The first two where solely based on the evidence that a memristive device should exist and its expected physical structure, where the last two models have showed a link between memristive devices and the STDP weight update rule in (artificial) neural networks. Note that the memristive device models exhibited above are just some (of the most important) examples of the broad range of models available. In addition, the simulations above are done for only one set of parameters, the fitting parameters in all the models can be adjusted to mimic other memristive devices.

4 Electrical networks and graph theory

In this chapter we will introduce some mathematical preliminaries and preliminaries from electrical networks.

4.1 Mathematical preliminaries and graph theory

In this section we will introduce graph theory and some other mathematical preliminaries.

4.1.1 Graph theory

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a undirected graph, where $\mathcal{V} = \{1, 2, \ldots, N\}$ represents the set of distinct nodes of the graph and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the set of edges, which consists of unordered pairs (i, j) where i,$j \in \mathcal{V}$. The graph is called simple if it does not contain self-loops, i.e. $(i, i) \notin \mathcal{E}$ for all $i \in \mathcal{V}$, nor multiple edges between two nodes. In addition, the graph is called connected if for any pair of nodes i, $j \in \mathcal{V}$ there exists a path connecting them, i.e. there is a sequence of p nodes $\{n_1, n_2, \ldots, n_p\}$ such that $(n_l, n_{l+1}) \in \mathcal{E}$ for all $l \in \{1, 2, \ldots, p-1\}$ with $n_1 = i$ and $n_p = j$.

Consider a simple undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with N nodes and M edges. Assume that there are labels assigned to all the edges such that one of its ends is labeled positive and the other negative. The incidence matrix $\mathcal{D} = (d_{ik}) \in \mathbb{R}^{N \times M}$ of the graph is defined as

$$d_{ik} = \begin{cases} 1 & \text{if } i \text{ is the positive end of edge } k, \\ -1 & \text{if } i \text{ is the negative end of edge } k, \\ 0 & \text{otherwise.} \end{cases}$$

For a given node $i \in \mathcal{V}$ its neighboring set $\mathcal{N}_i$ is defined as $\mathcal{N}_i := \{j \in \mathcal{V} \mid (i, j) \in \mathcal{E}\}$. The cardinality of the set $\mathcal{N}_i$, denoted as $\#\mathcal{N}_i$, gives the number of neighbors of node i. The Laplacian matrix $\mathcal{L} = (l_{ij}) \in \mathbb{R}^{N \times N}$ of the graph is defined by

$$l_{ij} = \begin{cases} \#\mathcal{N}_i & \text{if } j = i, \\ -1 & \text{if } j \in \mathcal{N}_i, \\ 0 & \text{otherwise,} \end{cases}$$

and can be calculated as $\mathcal{L} = \mathcal{D}\mathcal{D}^\intercal$.

An undirected weighted graph $\mathcal{G} = \left(\mathcal{V}, \mathcal{E}, \{w_{ij}\}_{(i,j)\in\mathcal{E}}\right)$ is a graph with a positive weight associated

to every edge. The weighted Laplacian matrix $\mathcal{L} \in \mathbb{R}^{N\times N}$ of a weighted graph is defined by

$$l_{ij} = \begin{cases} \sum_{j\in\mathcal{N}_i} w_{ij} & \text{if } j = i, \\ -w_{ij} & \text{if } j \in \mathcal{N}_i, \\ 0 & \text{otherwise,} \end{cases}$$

and can be calculated as $\mathcal{L} = \mathcal{D}\mathcal{W}\mathcal{D}^\intercal$, where $\mathcal{W} = \operatorname{diag}\left(\{w_{ij}\}_{(i,j)\in\mathcal{E}}\right) \in \mathbb{R}^{M\times M}$ is a diagonal matrix with $\{w_{ij}\}_{(i,j)\in\mathcal{E}}$ as its entries.

The rank of the (weighted) Laplacian matrix of a graph depends on the number of connected components of the graph, which is equal to one for a connected graph. From [26] we have the following result about the rank of the incidence matrix of a graph.

Theorem 4.1. *Let $\mathcal{G}$ be a graph with N nodes and c connected components. If $\mathcal{D}$ is the incidence matrix representing the graph structure of $\mathcal{G}$ then rank $(\mathcal{D}) = N - c$.*

4.1.2 Schur complement

In linear algebra the Schur complement of a block matrix is defined as follows.

Definition 4.1. (Schur complement, [27, Chapter 0]) *Consider the partitioned matrix*

$$M = \begin{pmatrix} A & B \\ C & D \end{pmatrix}$$

with $A \in \mathbb{R}^{p\times p}$ and $D \in \mathbb{R}^{q\times q}$. If A^{-1} exists then its Schur complement is defined as $M/A := D - CA^{-1}B$. If D^{-1} exists then its Schur complement is defined as $M/D := A - BD^{-1}C$. In addition, if A^{-1} exists, then

$$\textit{rank}\ (M) = \textit{rank}\ (A) + \textit{rank}\ (D - CA^{-1}B),$$

and if D^{-1} exists, then

$$\textit{rank}\ (M) = \textit{rank}\ (D) + \textit{rank}\ (A - BD^{-1}C).$$

The Schur complement is a handy tool to solve linear matrix equalities as

$$y_1 = Ax_1 + Bx_2, \tag{18}$$

$$y_2 = Cx_1 + Dx_2. \tag{19}$$

Namely, if D^{-1} exists then x_2 can be expressed as

$$x_2 = D^{-1}y_2 - D^{-1}Cx_1. \tag{20}$$

Substitution of this in (18) results in the matrix equality

$$y_1 - BD^{-1}y_2 = (A - BD^{-1}C)x_1, \tag{21}$$

which is an equation of order p where our original system was of order $p+q$. After solving (21) for x_1 one can simply calculate x_2 by substitution of the values of y_2 and x_1 in (20). Thus, the Schur complement can be used as a tool to reduce the order of a system of linear equations, making it easier to solve one.

In addition, the Schur complement can be used to compute the inverse of a block-matrix as:

Theorem 4.2. (Inverse of a block matrix, [27, Theorem 1.2]) *Consider the partitioned matrix*

$$M = \begin{pmatrix} A & B \\ C & D \end{pmatrix},$$

with $A \in \mathbb{R}^{p\times p}$ and $D \in \mathbb{R}^{q\times q}$. Assume that M^{-1} exists, then

$$M^{-1} = \begin{pmatrix} (A - BD^{-1}C)^{-1} & -(A - BD^{-1}C)^{-1}BD^{-1} \\ -D^{-1}C(A - BD^{-1}C)^{-1} & D^{-1}C(A - BD^{-1}C)^{-1}BD^{-1} + D^{-1} \end{pmatrix}.$$

4.1.3 Moore-Penrose inverse

The Moore-Penrose inverse $A^\dagger$, or pseudoinverse, is a generalization of the matrix inverse A^{-1} to singular matrices A. The Moore-Penrose inverse $A^\dagger$ is defined as the solution to a certain set of equations as follows.

Definition 4.2 (Moore-Penrose inverse, [28])**.** *The unique solution X to the four equations*

$$\begin{aligned} AXA &= A, \\ XAX &= A, \\ (AX)^\intercal &= AX, \\ (XA)^\intercal &= XA, \end{aligned}$$

is called the Moore-Penrose inverse, denoted as $A^\dagger$.

It can be verified that when A has linearly independent columns, i.e. $A^\intercal A$ is invertible, $A^\dagger$ can be computed as

$$A^\dagger = (A^\intercal A)^{-1}A^\intercal,$$

and constitutes a left inverse of A since $A^\dagger A = I$. Similarly, when A has linearly independent rows, i.e. $AA^\intercal$ is invertible, $A^\dagger$ can be computed as

$$A^\dagger = A^\intercal(AA^\intercal)^{-1},$$

and constitutes a right inverse of A since $AA^\dagger = I$.

4.2 Electrical Networks

In this section we will introduce some preliminaries from electrical networks which will be utilized when studying networks of memristive devices.

4.2.1 Kirchhoff's laws

In this section we will introduce Kirchhoff's current and voltage law based on [16]. These laws are important since they can be utilized while studying electrical networks and networks of memristive devices.

Theorem 4.3 (Kirchhoff's current law)**.** *For any electrical circuit, for any of its nodes, and at any time, the algebraic sum of all currents across the edges connected to the node is zero.*

Theorem 4.4 (Kirchhoff's voltage law)**.** *For any electrical circuit, for any of its loops, and at any time, the algebraic sum of the voltages across the edges around the loop is zero.*

In the next section, Kirchhoff's voltage law will be used to find a relation between the voltage potentials at the nodes and the voltages across the edges of an electrical circuit. Similarly, Kirchhoff's current law will be used to find a relation between the currents at the nodes and across the edges.

4.2.2 Graph theory

Having in mind the theory of graphs, we want to express resisitive circuits and sequentially, in Chapter 5, networks of memristive devices in terms of graphs.

Consider a simple, undirected, connected and weighted graph $\mathcal{G} = \left(\mathcal{V}, \mathcal{E}, \{w_{ij}\}_{(i,j)\in\mathcal{E}}\right)$ with N nodes, M edges and let the graph structure be specified by the incidence matrix $\mathcal{D} \in \mathbb{R}^{N\times M}$. Let $V = \begin{pmatrix} v_1 & \dots & v_M \end{pmatrix}^\intercal$ define the voltages across the edges and $I = \begin{pmatrix} i_1 & \dots & i_M \end{pmatrix}^\intercal$ the current across the edges. In addition, consider the vector of nodal voltage potentials $\psi = \begin{pmatrix} \psi_1 & \dots & \psi_N \end{pmatrix}^\intercal$ and the vector of nodal currents $J = \begin{pmatrix} j_1 & \dots & j_N \end{pmatrix}^\intercal$.

An example of this notation is given in Figure 4.2.2. Here, v_2 and i_2 denote the voltage through and current across edge 2, respectively. Furthermore, j_3 denotes the nodal current at node 3, which defines the current that can be extracted from the network at node 3. The voltage potential at node 3, denoted by ψ_3, defines the difference between the voltages at the surrounding edges.

It follows from Kirchhoff's voltage law, that the voltages across the edges are given by

$$V = \mathcal{D}^\intercal \psi,$$

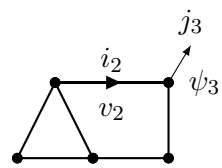

Figure 20: Simple electrical network. The voltage through and current across edge 2 are denoted by v_2 and i_2, respectively. Furthermore, j_3 and ψ_3 denote the nodal current and voltage potential at node 3.

and similarly, by Kirchhoff's current law, the nodal currents are given by

$$J = \mathcal{D}I. \tag{22}$$

Now, consider networks of resistors described by the above class of graphs. Let $\mathcal{W} = \operatorname{diag}\left(\{w_{ij}\}_{(i,j)\in\mathcal{E}}\right) \in \mathbb{R}^{M\times M}$ be a matrix whose entries represent the weights of the resistors on the edges of these networks with the unit of conductance. It then follows by Ohm's law that the current and voltage through the edges are related as

$$I = \mathcal{W}V.$$

Putting this all together, one obtains

$$J = \mathcal{D}\mathcal{W}\mathcal{D}^{\intercal}\psi$$

as the relation between the nodal currents and voltage potentials, where the matrix $\mathcal{L} = D\mathcal{W}D^{\intercal}$ is called the weighted Laplacian of the graph $\mathcal{G}$.

In order to describe the external behavior of the network, we split the nodes into N_C internal connection and N_B boundary nodes. Correspondingly, we split the nodal currents J and the nodal voltage potentials ψ as

$$J = \begin{pmatrix} J_B \\ J_C \end{pmatrix}, \quad \psi = \begin{pmatrix} \psi_B \\ \psi_C \end{pmatrix}$$

and the incidence matrix into $\mathcal{D} = \begin{pmatrix} \mathcal{D}_B^{\intercal} & \mathcal{D}_C^{\intercal} \end{pmatrix}^{\intercal}$. Using this notation, the relation between the nodal currents J and voltage potentials ψ is given by

$$\begin{pmatrix} J_B \\ J_C \end{pmatrix} = \begin{pmatrix} \mathcal{L}_{BB} & \mathcal{L}_{BC} \\ \mathcal{L}_{CB} & \mathcal{L}_{CC} \end{pmatrix} \begin{pmatrix} \psi_B \\ \psi_C \end{pmatrix}, \tag{23}$$

where

$$\begin{pmatrix} \mathcal{L}_{BB} & \mathcal{L}_{BC} \\ \mathcal{L}_{CB} & \mathcal{L}_{CC} \end{pmatrix} = \begin{pmatrix} \mathcal{D}_B \\ \mathcal{D}_C \end{pmatrix} \mathcal{W} \begin{pmatrix} \mathcal{D}_B^{\intercal} & \mathcal{D}_C^{\intercal} \end{pmatrix}.$$

This partition of the weighted Laplacian matrix can be used to find a relation between the current and voltage potentials at the boundary nodes N_B without considering the current and voltage potentials at the interconnected nodes, as we will see in the next section.

4.2.3 Kron reduction

In this section, we will utilize the Schur complement in order to find a relation between the current and voltage potentials at the boundary nodes N_B without considering the current and voltage potentials at the interconnected nodes N_C.

As before, the relation between the nodal current J and voltage potentials ψ is equal to (23). Assuming that the inverse of the block $\mathcal{L}_{CC}$ exists, Gaussian elimination of (23) with respect to the interconnected nodes gives

$$J_B = \mathcal{L}_{\text{red}}\psi_B + \mathcal{L}_{BC}\mathcal{L}_{CC}^{-1}J_C, \tag{24}$$

where $\mathcal{L}_{\text{red}}$ is the Schur complement of $\mathcal{L}$ with respect to the block $\mathcal{L}_{CC}$, that is,

$$\mathcal{L}_{\text{red}} = \mathcal{L}_{BB} - \mathcal{L}_{BC}\mathcal{L}_{CC}^{-1}\mathcal{L}_{CB} \in \mathbb{R}^{N_B \times N_B}.$$

By Kirchhoff's current law, the sum of the incoming currents at a node is equal to zero, hence it follows that the currents at the interconnected nodes are equal to zero, i.e. $J_C = 0$. Substitution of $J_C = 0$ in (24) gives

$$J_B = \mathcal{L}_{\text{red}}\psi_B,$$

which is a relation between the boundary currents and voltage potentials which does not depend on the internal current and voltage potentials. In addition, the accompanying matrix

$$\mathcal{L}_{\text{ac}} = -\mathcal{L}_{CC}^{-1}\mathcal{L}_{CB} \in \mathbb{R}^{N_C \times N_B}$$

gives a relation between the internal and boundary currents as

$$\psi_C = \mathcal{L}_{\text{ac}}\psi_B.$$

This reduction of an electrical network via the Schur complement of the associated weighted Laplacian is called Kron reduction due to its introduction by Gabriel Kron in 1939 [29].

5 Networks of Memristive Devices

In this chapter, we will use electrical circuit and graph theory to model a network of memristive devices. We will show that a network of memristive devices can be represented by a single memristive device, and we will develop a method to compute the effective memristance of such a network.

5.1 Circuit-theoretic properties

Consider the memristive device given by (3), (4) which is characterized by the relation $g(q(t), \varphi(t)) = 0$. In 1971, Leon O. Chua showed that a network of memristive devices connected to the outside world with the same input as output current is a memristive device itself.

Theorem 5.1 (Closure theorem, [14])**.** *A one-port*[2] *containing only memristive devices is equivalent to a memristive device.*

Proof. Consider an one-port containing K memristive devices. Let $i_k(t)$, $v_k(t)$, $q_k(t)$, and $\varphi_k(t)$ denote the current, voltage, charge and flux-linkage of the k-th memristive device, where $k = 1, 2, \ldots, K$, and let $i(t)$ and $v(t)$ denote the port current and voltage, i.e. the current and voltage going through the terminals. Assuming that the network is connected, it follows from Theorem 4.1 that we can write $N-1$ independent equations based on Kirchhoff's current law, see (22) and recall that $J_C = \mathcal{D}_c I = 0$; namely,

$$\alpha_{j0} i(t) + \sum_{k=1}^{K} \alpha_{jk} i_k(t) = 0, \tag{25}$$

for $j = 1, 2, \ldots, N-1$, where α_{jk} is either 1, if the k-th edge is an incoming edge, -1 if it is an outgoing edge, or 0 if the k-th edge is not linked to node j. Here, N is the total number of nodes in the network. Similarly, we can write $K - N + 2$, which is the minimum number of loops in a connected graph plus one loop including the two terminals, independent equations based on Kirchhoff's voltage law; namely,

$$\beta_{j0} v(t) + \sum_{k=1}^{K} \beta_{jk} v_k(t) = 0, \tag{26}$$

for $j = 1, 2, \ldots, K - N + 2$, where β_{jk} is either 1, -1, or 0 if the k-th edge is an incoming, outgoing edge, or not linked to node j, respectively.

[2] An one-port or two-terminal is a black box connected with two nodes, i.e. terminals to an outside world. The current flowing into the two nodes must be equal and opposite.

If we integrate both equation (25) and (26) with respect to time, we obtain

$$\begin{aligned}
\alpha_{j0}q(t) + \sum_{k=1}^{K} \alpha_{jk}q_k(t) + q_C = 0 \text{ for } j = 1, 2, \ldots, N-1, \\
\beta_{j0}\varphi(t) + \sum_{k=1}^{K} \beta_{jk}\varphi_k(t) + \varphi_C = 0 \text{ for } j = 1, 2, \ldots, K-N+2,
\end{aligned} \tag{27}$$

where q_C and φ_C are constants of integration. Without loss of generality, assume that the devices are charge-controlled[3]. Recall that they can be described by

$$\begin{aligned}
\dot{x}_k(t) &= f(x_k(t), i_k(t)), \\
v_k(t) &= R(x_k(t), i_k(t), t)i_k(t),
\end{aligned} \tag{28}$$

for $k = 1, 2, \ldots, K$ with f and R continuous functions. Furthermore, recall that

$$\varphi_k(t) = \int_{-\infty}^{t} v_k(\tau)d\tau, \tag{29}$$

for $k = 1, 2, \ldots, K$. By substitution of (28) in (29) we can find a relation between $\varphi_k(t)$ and $q_k(t)$ as

$$\begin{aligned}
\varphi_k(t) &= \int_{-\infty}^{t} v_k(\tau)d\tau \\
&= \int_{-\infty}^{t} R(x_k(\tau), i_k(\tau), \tau)i(\tau)d\tau \\
&= h(q_k(\tau), \dot{q}_k(\tau)),
\end{aligned}$$

for $k = 1, 2, \ldots, K$ where we used that $q_k(t) = \int_{-\infty}^{t} i_k(\tau)d\tau$, and h defines a function. Substitution of this result into (27) gives

$$\begin{aligned}
\alpha_{j0}q(t) + \sum_{k=1}^{K} \alpha_{jk}q_k(t) + q_C = 0 \text{ for } j = 1, 2, \ldots, N-1, \\
\beta_{j0}\varphi(t) + \sum_{k=1}^{K} \beta_{jk}h(q_k(t), \dot{q}_k(t)) + \varphi_C = 0 \text{ for } j = 1, 2, \ldots, K-N+2,
\end{aligned}$$

hence we have a system of $N-1+K-N+2 = K+1$ independent equations with K unknowns $q_k(t)$, and the unknown $q(t)$. Hence, solving for $\varphi(t)$, we obtain a relation $g(q(t), \varphi(t)) = 0$. We conclude that the one-port containing only memristive devices represents a memristive device itself. □

Having this result in mind, we want to derive an expression for the behavior of a memristive device which is constructed out of a network of memristive devices. This will be done by applying Kron reduction to network modeling of flux-controlled memristive devices.

[3]The proof can be easily modified to flux-controlled memristive devices.

5.2 Mathematical description

Consider a time-invariant flux-controlled memristive device where G only depends on the state $\boldsymbol{w}$ of the device and not on the voltage through the device. This device is described as

$$\begin{aligned}\dot{\boldsymbol{w}} &= f(\boldsymbol{w}, v),\\ i &= G(\boldsymbol{w})v,\end{aligned}$$

where f and G are continuous functions. Here, G represents the memductance of the device; it is positive for every $\boldsymbol{w}$. For simplicity in notation, the time dependence of $\boldsymbol{w}$, i and v is omitted here.

Having this memristive device in mind, we will consider a network of memristive devices. This network will be described using a simple weighted graph with the edges of the graph representing memristive devices. We assume that a memristive device situated at the edge between node i and j is represented as

$$\begin{aligned}\dot{\boldsymbol{w}}_{ij} &= f(\boldsymbol{w}_{ij}, p_k v_k),\\ i_k &= G(\boldsymbol{w}_{ij})v_k,\end{aligned} \tag{30}$$

where k is the label associated to the edge between node i and j. The parameter p_k represents the polarity of the device, which is 1 if the applied current is inserted at the positive side of the device and -1 if the current is inserted at the negative side of the device.

Now, consider the simple undirected weighted graph $\mathcal{G} = \left(\mathcal{V}, \mathcal{E}, \{G(\boldsymbol{w}_{ij})\}_{(i,j)\in\mathcal{E}}\right)$ with N nodes, K edges, and let the graph structure be specified by the incidence matrix $\mathcal{D}$. Here, the orientation of the graph is based on the orientation of the memristive devices at the respective edges, i.e.

$$d_{ik} = \begin{cases} p_k & \text{if the positive end of the } k\text{-th memristive device is directed towards node } i,\\ -p_k & \text{if the negative end of the } k\text{-th memristive device is directed towards node } i,\\ 0 & \text{otherwise.}\end{cases}$$

The weights $G(\boldsymbol{w}_{ij})$ in this graph depend on the dynamics (30). Specifically, the state-dependent weighted Laplacian of this graph can be calculated as

$$\mathcal{L}(\boldsymbol{w}) = \mathcal{D}\mathcal{W}(\boldsymbol{w})\mathcal{D}^{\mathsf{T}},$$

where $\mathcal{W}(\boldsymbol{w}) = \operatorname{diag}\left(\{G(\boldsymbol{w}_{ij})\}_{(i,j)\in\mathcal{E}}\right)$.

As before, let $V = \begin{pmatrix} v_1 & \dots & v_M \end{pmatrix}^{\mathsf{T}}$ define the voltages across the edges and $I = \begin{pmatrix} i_1 & \dots & i_M \end{pmatrix}^{\mathsf{T}}$ the current across the edges. In addition, consider the vector of nodal voltage potentials $\psi =$

$\begin{pmatrix}\psi_1 & \dots & \psi_N\end{pmatrix}^\intercal$ and the vector of nodal currents $J = \begin{pmatrix}j_1 & \dots & j_N\end{pmatrix}^\intercal$. By Section 4.2.2 and (30), it follows that

$$\begin{aligned}\dot{\boldsymbol{w}} &= F(\boldsymbol{w}, PV),\\ J &= \mathcal{D}\mathcal{W}(\boldsymbol{w})\mathcal{D}^\intercal\psi,\end{aligned} \tag{31}$$

defines the relation between the nodal currents and voltage potentials. Here, F is the vector having as entries f evaluated at the different edges, and P is a diagonal matrix having as entries the polarities of the memristive devices.

5.3 Kron reduction

In order to find the external behavior of a network of memristive devices, we split the set of nodes into N_C internal connection and N_B boundary nodes as before. Here, N_B represents the set of nodes connected with terminals to the outside world. Accordingly, split the nodal currents J and the nodal voltages potentials ψ as

$$J = \begin{pmatrix}J_B\\ J_C\end{pmatrix}, \quad \psi = \begin{pmatrix}\psi_B\\ \psi_C\end{pmatrix}$$

and the incidence matrix into $\mathcal{D} = \begin{pmatrix}\mathcal{D}_B^\intercal & \mathcal{D}_C^\intercal\end{pmatrix}^\intercal$. Using this notation, the relation between the nodal currents J and voltage potentials ψ is given by

$$\begin{pmatrix}J_B\\ J_C\end{pmatrix} = \begin{pmatrix}\mathcal{L}_{BB}(\boldsymbol{w}) & \mathcal{L}_{BC}(\boldsymbol{w})\\ \mathcal{L}_{CB}(\boldsymbol{w}) & \mathcal{L}_{CC}(\boldsymbol{w})\end{pmatrix}\begin{pmatrix}\psi_B\\ \psi_C\end{pmatrix}, \tag{32}$$

where the weighted Laplacian is partitioned as

$$\mathcal{L}(\boldsymbol{w}) = \begin{pmatrix}\mathcal{L}_{BB}(\boldsymbol{w}) & \mathcal{L}_{BC}(\boldsymbol{w})\\ \mathcal{L}_{CB}(\boldsymbol{w}) & \mathcal{L}_{CC}(\boldsymbol{w})\end{pmatrix} = \begin{pmatrix}\mathcal{D}_B\\ \mathcal{D}_C\end{pmatrix}\mathcal{W}(\boldsymbol{w})\begin{pmatrix}\mathcal{D}_B^\intercal & \mathcal{D}_C^\intercal\end{pmatrix}.$$

This state-dependent weighted Laplacian has many properties, of which some are collected in the following theorem, which is a generalization of Theorem 3.1 in [30]. These properties will be key to the subsequent characterization of the external behavior of a memristive circuit.

Theorem 5.2. *Consider a simple graph $\mathcal{G} = \left(\mathcal{V}, \mathcal{E}, \{G(\boldsymbol{w}_{ij})\}_{(i,j)\in\mathcal{E}}\right)$ with incidence matrix $\mathcal{D}$. Let $\mathcal{W}(\boldsymbol{w}) = diag\left(\{G(\boldsymbol{w}_{ij})\}_{(i,j)\in\mathcal{E}}\right)$ be a positive definite matrix for each $\boldsymbol{w}$.*

1. *The state-dependent weighted Laplacian $\mathcal{L}(\boldsymbol{w}) = \mathcal{D}\mathcal{W}(\boldsymbol{w})\mathcal{D}^\intercal$ is symmetric, positive semi-definite, and independent of the orientation of the graph. Furthermore, it has all diagonal elements non-negative, all off-diagonal elements non-positive, and has zero row and column sums. Hence the vector $\mathbb{1}_N$, the vector of length N consisting of only ones, is in the kernel of $\mathcal{L}(\boldsymbol{w})$. In addition, the graph $\mathcal{G}$ is connected if and only if $\ker\mathcal{L}(\boldsymbol{w}) = span(\mathbb{1}_{\mathbb{N}})$.*

2. *If the graph $\mathcal{G}$ is connected, then all diagonal elements are positive. Furthermore, the Schur complement*

$$\mathcal{L}_{red}(\boldsymbol{w}) = \mathcal{L}_{BB}(\boldsymbol{w}) - \mathcal{L}_{BC}(\boldsymbol{w})\mathcal{L}_{CC}^{-1}(\boldsymbol{w})\mathcal{L}_{CB}(\boldsymbol{w})$$

is well defined, symmetric, positive semi-definite, with positive diagonal elements, non-positive off-diagonal elements, and with zero row and column sums. In particular, $\mathcal{L}_{red}(\boldsymbol{w})$ can be written as $\mathcal{D}_{red}\mathcal{W}_{red}(\boldsymbol{w})\mathcal{D}_{red}^{\intercal}$, with $\mathcal{D}_{red}$ the incidence matrix of a connected graph $\mathcal{G}_{red}$.

Proof. 1. Since $\mathcal{W}(\boldsymbol{w})$ is a diagonal matrix, it is evident that $\mathcal{L}(\boldsymbol{w})$ is symmetric. It is also easy to see that $\mathcal{L}(\boldsymbol{w})$ is positive semi-definite, namely

$$\begin{aligned} x^{\intercal}\mathcal{L}(\boldsymbol{w})x &= x^{\intercal}\mathcal{D}\mathcal{W}(\boldsymbol{w})D^{\intercal}x = x^{\intercal}\mathcal{D}\mathcal{W}(\boldsymbol{w})^{\frac{1}{2}}\mathcal{W}(\boldsymbol{w})^{\frac{1}{2}}\mathcal{D}^{\intercal}x \\ &= ((\mathcal{D}\mathcal{W}(\boldsymbol{w})^{\frac{1}{2}})^{\intercal}x)^{\intercal}(\mathcal{D}\mathcal{W}(\boldsymbol{w})^{\frac{1}{2}})^{\intercal}x = \|(\mathcal{D}\mathcal{W}(\boldsymbol{w})^{\frac{1}{2}})^{\intercal}x\|_2^2 \geq 0. \end{aligned}$$

Let an incidence matrix $\mathcal{D}$ represent the orientation of the graph $\mathcal{G}$ and let $\hat{\mathcal{D}}$ represent another orientation for the same graph. These matrices are then related as $\hat{\mathcal{D}} = S\mathcal{D}$ where $S \in \mathbb{R}^{M\times M}$ is a diagonal matrix with entries $\{s_k\}_{k=1}^{M}$ where $s_k = 1$ if edge k has the same orientation in $\mathcal{D}$ and $\hat{\mathcal{D}}$, and $s_k = -1$ otherwise. Since S is a diagonal matrix with all entries either 1 or -1, it follows that $S^{\intercal} = S$ and $SS = I$. The weighted Laplacian of the graph $\mathcal{G}$ is then given by

$$\begin{aligned} \hat{\mathcal{L}}(\boldsymbol{w}) &= \hat{\mathcal{D}}\mathcal{W}(\boldsymbol{w})\hat{\mathcal{D}}^{\intercal} = \mathcal{D}S\mathcal{W}(\boldsymbol{w})(DS)^{\intercal} = \mathcal{D}S\mathcal{W}(\boldsymbol{w})S^{\intercal}\mathcal{D}^{\intercal} \\ &= \mathcal{D}SS^{\intercal}\mathcal{W}(\boldsymbol{w})\mathcal{D}^{\intercal} = \mathcal{D}\mathcal{W}(\boldsymbol{w})\mathcal{D}^{\intercal} = \mathcal{L}(\boldsymbol{w}), \end{aligned}$$

where we used that $\mathcal{W}(\boldsymbol{w})$ and S are commutative since both matrices are diagonal. We conclude that the weighted Laplacian is independent of the orientation of the graph. Furthermore, from Section 4.1.1 it follows that the elements of $\mathcal{L}(\boldsymbol{w})$ are given by

$$l_{ij}(\boldsymbol{w}) = \begin{cases} \sum_{j\in\mathcal{N}_i} G(\boldsymbol{w}_{ij}) & \text{if } j = 1, \\ -G(\boldsymbol{w}_{ij}) & \text{if } j \in \mathcal{N}_i, \\ 0 & \text{otherwise,} \end{cases}$$

and hence all diagonal elements of $\mathcal{L}(\boldsymbol{w})$ are non-negative and the off-diagonal elements are non-positive. The row sum of $\mathcal{L}(\boldsymbol{w})$ is given by

$$\sum_{j=1}^{N} l_{ij}(\boldsymbol{w}) = l_{ii}(\boldsymbol{w}) + \sum_{j\in\mathcal{N}_i} l_{ij}(\boldsymbol{w}) = \sum_{j\in\mathcal{N}_i} G(\boldsymbol{w}_{ij}) - \sum_{j\in\mathcal{N}_i} G(\boldsymbol{w}_{ij}) = 0,$$

and hence $\mathbb{1}_N \in$ ker $(\mathcal{L}(\boldsymbol{w}))$. By the symmetry of $\mathcal{L}(\boldsymbol{w})$ it follows that the column sum of $\mathcal{L}(w)$ is equal to zero as well.

We are left to show that ker $\mathcal{L}(\boldsymbol{w}) = \mathbb{1}_N$ if and only if the graph $\mathcal{G}$ is connected. By Theorem 4.1 it follows that rank $(\mathcal{D}) = N - 1$ if and only if the graph $\mathcal{G}$ is connected. Hence, dim (ker $\mathcal{D}^\intercal$) $= 1$ if and only if the graph $\mathcal{G}$ is connected. Furthermore, if we can show that ker $\mathcal{D}^\intercal =$ ker $\mathcal{L}(\boldsymbol{w})$, then dim (ker $\mathcal{D}^\intercal$) = dim (ker $\mathcal{L}(\boldsymbol{w})$).

Let $x \in$ ker $\mathcal{L}(\boldsymbol{w})$, then

$$\begin{aligned} &\mathcal{L}(\boldsymbol{w})x = 0 \;\Rightarrow\; x^\intercal \mathcal{L}(\boldsymbol{w})x = 0 \;\Rightarrow\; \|\mathcal{W}(\boldsymbol{w})^{\frac{1}{2}}\mathcal{D}^\intercal x\|_2^2 = 0 \\ &\Rightarrow\; \mathcal{W}(\boldsymbol{w})^{\frac{1}{2}}\mathcal{D}^\intercal x = 0 \;\Rightarrow\; \mathcal{D}^\intercal x = 0 \;\Rightarrow\; x \in \text{ker } \mathcal{D}^\intercal, \end{aligned}$$

where we used that $\mathcal{W}(w)$ is a diagonal positive definite matrix. It follows that ker $\mathcal{L}(\boldsymbol{w}) \subseteq$ ker $\mathcal{D}^\intercal$. On the other hand, if $x \in$ ker $\mathcal{D}^\intercal$, then $\mathcal{L}(\boldsymbol{w})x = \mathcal{D}\mathcal{W}(\boldsymbol{w})\mathcal{D}^\intercal x = 0$, from which it follows that ker $\mathcal{D}^\intercal \subseteq$ ker $\mathcal{L}(\boldsymbol{w})$. We conclude that ker $\mathcal{D}^\intercal =$ ker $\mathcal{L}(\boldsymbol{w})$ which implies that dim (ker $\mathcal{L}(\boldsymbol{w})$) = dim (ker $\mathcal{D}^\intercal$). Since $\mathbb{1}_N \in$ ker $\mathcal{L}(\boldsymbol{w})$ and dim (ker $\mathcal{D}^\intercal$) $= 1$ if and only if the graph is connected, it follows that ker $\mathcal{L}(\boldsymbol{w}) =$ span $\mathbb{1}_N$ if and only if the graph $\mathcal{G}$ is connected.

2. If a graph $\mathcal{G}$ is connected, then for each node there exists at least one edge linking this node to another node, implying that each diagonal element of $\mathcal{L}(\boldsymbol{w})$ is positive. We will show that the Schur complement of $\mathcal{L}(\boldsymbol{w})$ with respect to the last diagonal element of $\mathcal{L}(\boldsymbol{w})$, $l_{NN}(\boldsymbol{w})$, is symmetric, positive semi-definite, with positive diagonal elements, non-positive off-diagonal elements, and with zero row and column sums.

   Partition the matrix $\mathcal{L}(\boldsymbol{w})$ as

$$\mathcal{L}(\boldsymbol{w}) = \begin{pmatrix} \mathcal{L}^{N-1}(\boldsymbol{w}) & l(\boldsymbol{w}) \\ l^\intercal(\boldsymbol{w}) & l_{NN}(\boldsymbol{w}) \end{pmatrix},$$

   where $l_{NN}(\boldsymbol{w})$ is the last diagonal entry of $\mathcal{L}(\boldsymbol{w})$, and $l(\boldsymbol{w})$ is the last column of $\mathcal{L}(\boldsymbol{w})$ minus its last element. Since $l_{NN}(\boldsymbol{w}) > 0$ for all $\boldsymbol{w}$, $l_{NN}^{-1}(\boldsymbol{w})$ exists, and the Schur complement of $\mathcal{L}(\boldsymbol{w})$ with respect to $l_{NN}(\boldsymbol{w})$ is given by

$$\hat{\mathcal{L}}(w) = \mathcal{L}^{N-1}(\boldsymbol{w}) - \frac{1}{l_{NN}(\boldsymbol{w})} l(\boldsymbol{w}) l^\intercal(\boldsymbol{w}).$$

   Since both $\mathcal{L}^{N-1}(\boldsymbol{w})$ and $l^\intercal(\boldsymbol{w})l(\boldsymbol{w})$ are symmetric, $\hat{\mathcal{L}}(\boldsymbol{w})$ is symmetric as well. In addition, all elements of $l(\boldsymbol{w})$ non-positive since they are off-diagonal elements in $\mathcal{L}(\boldsymbol{w})$, this implies that all elements in $l^\intercal(\boldsymbol{w})l(\boldsymbol{w})$ are positive. Furthermore, the off-diagonal elements of $\mathcal{L}^{N-1}(\boldsymbol{w})$ are non-positive, so the off-diagonal elements of $\hat{\mathcal{L}}(\boldsymbol{w})$ are non-positive.

   Next, we want to show that the row and column sums of $\hat{\mathcal{L}}(\boldsymbol{w})$ are zero. This will be done by using that the row sum of $\mathcal{L}(\boldsymbol{w})$ is equal to zero, i.e. $\mathbb{1}_N \in$ ker $\mathcal{L}(\boldsymbol{w})$. Namely, we have that

$$\begin{pmatrix} \mathcal{L}^{N-1}(\boldsymbol{w}) & l(\boldsymbol{w}) \\ l^\intercal(\boldsymbol{w}) & l_{NN}(\boldsymbol{w}) \end{pmatrix} \begin{pmatrix} \mathbb{1}_{N-1} \\ 1 \end{pmatrix} = 0,$$

which implies that

$$\mathcal{L}^{N-1}(\boldsymbol{w})\mathbb{1}_{N-1} + l(\boldsymbol{w}) = 0,$$
$$l^{\intercal}(\boldsymbol{w})\mathbb{1}_{N-1} + l_{NN}(\boldsymbol{w}) = 0,$$

and hence

$$\begin{aligned}\hat{\mathcal{L}}(\boldsymbol{w})\mathbb{1}_{N-1} &= \mathcal{L}^{N-1}(\boldsymbol{w})\mathbb{1}_{N-1} - \frac{1}{l_{NN}(\boldsymbol{w})} l(\boldsymbol{w}) l^{\intercal}(\boldsymbol{w})\mathbb{1}_{N-1} \\ &= -l(\boldsymbol{w}) + \frac{1}{l_{NN}(\boldsymbol{w})} l(\boldsymbol{w}) l_{NN}(\boldsymbol{w}) = 0,\end{aligned}$$

so $\mathbb{1}_{N-1} \in \ker \hat{\mathcal{L}}(\boldsymbol{w})$, and the row sum of $\hat{\mathcal{L}}(\boldsymbol{w})$ is equal to zero. By the symmetry of $\hat{\mathcal{L}}(\boldsymbol{w})$ it follows that the column sum of $\hat{\mathcal{L}}(\boldsymbol{w})$ is equal to zero as well. Also, since the off-diagonal elements of $\hat{\mathcal{L}}(w)$ are non-positive, the zero row sum implies that the diagonal elements of $\hat{\mathcal{L}}(\boldsymbol{w})$ are non-negative.

Furthermore, from the assumption that $\mathcal{G}$ is connected it follows that rank $\mathcal{L}(\boldsymbol{w}) = N - 1$, and clearly we have that rank $l_{NN}(\boldsymbol{w}) = 1$. Theorem 4.1 gives that

$$\text{rank } \mathcal{L}(\boldsymbol{w}) = \text{rank } l_{NN}(\boldsymbol{w}) - \text{rank } \hat{\mathcal{L}}(\boldsymbol{w}),$$

and hence rank $\hat{\mathcal{L}}(w) = N - 2$. This implies that $\dim\left(\ker \hat{\mathcal{L}}(\boldsymbol{w})\right) = 1$ from which, together with the fact that $\mathbb{1}_{N-1} \in \ker \hat{\mathcal{L}}(\boldsymbol{w})$, it follows that $\ker \hat{\mathcal{L}}(\boldsymbol{w}) = \text{span}\,(\mathbb{1}_{N-1})$.

Now, define the undirected graph $\hat{\mathcal{G}}$ with an edge between the nodes i and j if and only if the (i, j)-th element of $\hat{\mathcal{L}}(\boldsymbol{w})$ is nonzero. Furthermore, associate to this edge the weight given by the (i, j)-th element of $\hat{\mathcal{L}}(\boldsymbol{w})$ at time t. Finally, assign an arbitrary orientation to the graph. The matrices $\hat{\mathcal{D}}$ and $\hat{\mathcal{W}}(\boldsymbol{w})$ (at each time t) can then be constructed based on the graph $\hat{\mathcal{G}}$, and we have that $\hat{\mathcal{L}}(\boldsymbol{w}) = \hat{\mathcal{D}}\hat{W}(\boldsymbol{w})\hat{D}^{\intercal}$. Here, the graph $\hat{\mathcal{G}}$ is connected since we showed that $\ker \hat{\mathcal{L}}(\boldsymbol{w}) = \text{span}\,(\mathbb{1}_{N-1})$. This implies that the diagonal elements in $\hat{\mathcal{L}}(\boldsymbol{w})$ are positive.

The above procedure can be applied to all diagonal elements of $\mathcal{L}(w)$, hence we have proved the statement for the Schur complement with respect to any diagonal element of $\mathcal{L}(\boldsymbol{w})$. Notice that every Schur complement can be obtained by successive application of taking the Schur complement with respect to the diagonal entries. In the case we want to calculate the Schur complement of $\mathcal{L}(\boldsymbol{w})$ with respect to $\mathcal{L}_{CC}(\boldsymbol{w})$, we take the Schur complement with respect to the last diagonal entry of $\mathcal{L}(\boldsymbol{w})$ and repeat this procedure $N_C - 1$ times until we end up with $\mathcal{L}_{\text{red}}(\boldsymbol{w})$. As before, we associate the matrices $\mathcal{D}_{\text{red}}$ and $\mathcal{W}_{\text{red}}(\boldsymbol{w})$ with the graph $\mathcal{G}_{\text{red}}$, build out of $\mathcal{L}_{\text{red}}$, to obtain $\mathcal{L}_{\text{red}} = \mathcal{D}_{\text{red}}\mathcal{W}_{\text{red}}(\boldsymbol{w})\mathcal{D}_{\text{red}}^{\intercal}$.

□

Now, for any connected network of memristive device, we have the relation

$$J_B = \mathcal{L}_{\text{red}}(\boldsymbol{w})\psi_B$$

between the boundary currents and voltage potentials. Furthermore, as in Section 4.2.3, the accompanying matrix $\mathcal{L}_{ac}(\boldsymbol{w}) = -\mathcal{L}_{CC}^{-1}(\boldsymbol{w})\mathcal{L}_{CB}(\boldsymbol{w})$, which is well-defined by Theorem 5.2, gives a relation between the internal and boundary currents as

$$\psi_C = \mathcal{L}_{\text{ac}}(\boldsymbol{w})\psi_B. \tag{33}$$

This relation between the internal and boundary currents can be used to rewrite the state of the system (31) as a function solely depending on the voltage potentials at the boundary nodes; namely, substitution of (33) in (31) gives

$$\begin{aligned}
\dot{\boldsymbol{w}} &= F(\boldsymbol{w}, PV) = F\left(\boldsymbol{w}, P\mathcal{D}^{\intercal}\psi\right) \\
&= F\left(\boldsymbol{w}, P\begin{pmatrix}\mathcal{D}_B^{\intercal} & \mathcal{D}_C^{\intercal}\end{pmatrix}\begin{pmatrix}\psi_B \\ \psi_C\end{pmatrix}\right) \\
&= F\left(\boldsymbol{w}, P\begin{pmatrix}\mathcal{D}_B^{\intercal} \\ \mathcal{D}_C^{\intercal}\mathcal{L}_{\text{ac}}(\boldsymbol{w})\end{pmatrix}\psi_B\right).
\end{aligned}$$

It follows that the dynamics of any connected network of memristive devices can be described as

$$\begin{aligned}
\dot{\boldsymbol{w}} &= F\left(\boldsymbol{w}, P\begin{pmatrix}\mathcal{D}_B^{\intercal} \\ \mathcal{D}_C^{\intercal}\mathcal{L}_{\text{ac}}(\boldsymbol{w})\end{pmatrix}\psi_B\right), \\
J_B &= \mathcal{L}_{\text{red}}(\boldsymbol{w})\psi_B,
\end{aligned}$$

where F is the vector having as entries f evaluated at the different edges, and P is a diagonal matrix having as entries the polarities of the memristive devices.

5.4 Effective Memristance

One way of finding the external behavior of a network of memristive devices is by computing the effective memristance between the boundary nodes of the network.

Definition 5.1 (Effective memristance)**.** *The effective memristance $R_{ij}(\boldsymbol{w})$ between two nodes $i, j \in N$ of an undirected connected graph with a state-dependent weighted Laplacian $\mathcal{L}(\boldsymbol{w})$ is*

$$R_{ij}(\boldsymbol{w}) = (e_i - e_j)^{\intercal}\mathcal{L}^{\dagger}(\boldsymbol{w})(e_i - e_j), \tag{34}$$

where $\mathcal{L}^{\dagger}(\boldsymbol{w})$ is the Moore-Penrose inverse of $\mathcal{L}(\boldsymbol{w})$, and e_i and e_j represent the i-th and j-th unit vector in $\mathbb{R}^N$

The effective memristance between any of the nodes in a graph, and so the effective memristance matrix $R(\boldsymbol{w}) \in \mathbb{R}^{N\times N}$ of a graph, can be computed using this definition. By definition, $R(\boldsymbol{w})$ has zero diagonal elements, and since $\mathcal{L}^{\dagger}(\boldsymbol{w})$ is a symmetric matrix, $R(\boldsymbol{w})$ is a symmetric matrix as well. Since we are mainly interested in the external behavior of the network, i.e. the effective memristance between the boundary nodes, we would like to find a way to compute this without using the complete weighted Laplacian matrix. To simplify the computation of the effective memristance between the boundary nodes, we use Kron reduction, and obtain the following result:

Theorem 5.3 (Invariance of resistive properties under Kron reduction)**.** *Consider the state-dependent weighted Laplacian $\mathcal{L}(\boldsymbol{w})$ of a simple undirected connected graph, the corresponding Kron-reduced state-dependent weighted Laplacian $\mathcal{L}_{red}(\boldsymbol{w})$, and the effective memristance* (34)*. The effective memristance $R_{ij}(\boldsymbol{w})$ between any two boundary nodes is equal when computed for $\mathcal{L}(\boldsymbol{w})$ or $\mathcal{L}_{red}(\boldsymbol{w})$, that is, for any $i, j \in N_B$, we have that*

$$R_{ij}(\boldsymbol{w}) = (e_i - e_j)^{\intercal}\mathcal{L}^{\dagger}(\boldsymbol{w})(e_i - e_j) = (\bar{e}_i - \bar{e}_j)^{\intercal}\mathcal{L}^{\dagger}_{red}(\boldsymbol{w})(\bar{e}_i - \bar{e}_j),$$

where e_i, e_j, and $\bar{e}_i$, $\bar{e}_j$ represent the i-th and j-th unit vector in $\mathbb{R}^N$ and $\mathbb{R}^{N_B}$, respectively.

The previous result states that the effective memristance between the boundary nodes is invariant under Kron reduction of the interior nodes. Hence, the effective resistance matrix $R(\boldsymbol{w}) \in \mathbb{R}^{N_B\times N_B}$, which gives the effective memristance between the boundary nodes in the original network, can be computed from the Moore-Penrose inverse of the Kron-reduces weighted Laplacian $\mathcal{L}_{\text{red}}(\boldsymbol{w})$.

For the proof of Theorem 5.3, we need to establish some identities relating the weighted Laplacian $\mathcal{L}(\boldsymbol{w})$ and its Moore-penrose inverse $\mathcal{L}^{\dagger}(\boldsymbol{w})$, these results are based on [31].

Lemma 5.1. *Let $\mathcal{L}(\boldsymbol{w})$ define the state-dependent weighted Laplacian of a simple undirected connected graph, then a pseudo-inverse of $\mathcal{L}(\boldsymbol{w})$ is given by*

$$\mathcal{L}^{\dagger}(\boldsymbol{w}) = U(\boldsymbol{w})D(\boldsymbol{w})U(\boldsymbol{w})^{\intercal},$$

where $U(\boldsymbol{w})$ is an orthogonal matrix having as entries the eigenvectors of $\mathcal{L}(\boldsymbol{w})$, and $D(\boldsymbol{w}) = diag\,(1/\lambda_1(\boldsymbol{w}), \ldots, 1/\lambda_{n-1}(\boldsymbol{w}), 0)$ with $\lambda_1, \ldots, \lambda_{n-1}, 0$ the eigenvalues of $\mathcal{L}(\boldsymbol{w})$. Furthermore, $\mathbb{1}_N \in \ker \mathcal{L}^{\dagger}(\boldsymbol{w})$.

Proof. Consider the state-dependent weighted Laplacian $\mathcal{L}(\boldsymbol{w})$ of a connected graph. By Theorem 5.2, $\mathcal{L}(\boldsymbol{w})$ is symmetric, and its eigenvalues satisfy $\lambda_1(\boldsymbol{w}) \geq \ldots \geq \lambda_{n-1}(\boldsymbol{w}) > \lambda_n(\boldsymbol{w}) = 0$. Applying the singular value decomposition, see [32, Chapter 6.5], to $\mathcal{L}(\boldsymbol{w})$ gives that there exists an orthonormal matrix $U(\boldsymbol{w})$ such that

$$\mathcal{L}(\boldsymbol{w}) = U(\boldsymbol{w})\Sigma(\boldsymbol{w})U^{\intercal}(\boldsymbol{w}),$$

where $\Sigma = \operatorname{diag}(\lambda_1(\boldsymbol{w}), \ldots, \lambda_{n-1}(\boldsymbol{w}), 0)$. The columns $u_1(\boldsymbol{w}), \ldots, u_{n-1}(\boldsymbol{w})$ of $U(\boldsymbol{w})$ are then the eigenvectors corresponding to the eigenvalues $\lambda_1(\boldsymbol{w}), \ldots, \lambda_{n-1}(\boldsymbol{w})$. Furthermore, the column $u_n = \frac{1}{\sqrt{n}}\mathbb{1}_N$ is the eigenvector belonging to the eigenvalue 0. Note that this vector does not depend on $\boldsymbol{w}$ by the definition of $L(\boldsymbol{w})$.

The elements of $\mathcal{L}(\boldsymbol{w})$ can then be expressed as

$$(\mathcal{L}(\boldsymbol{w}))_{ij} = \sum_{k=1}^{n} \lambda_k u_{ik} u_{jk},$$

where the dependence of λ_k and u_{ij} is on $\boldsymbol{w}$ is omitted. Solving the four equations in Definition 4.2, we can find that the elements of the Moore-Penrose inverse of $\mathcal{L}(\boldsymbol{w})$ are given by

$$\left(\mathcal{L}^{\dagger}(\boldsymbol{w})\right)_{ij} = \sum_{k=1}^{n-1} \frac{1}{\lambda_k} u_{ik} u_{jk},$$

and hence

$$\mathcal{L}^{\dagger}(\boldsymbol{w}) = U(\boldsymbol{w}) D(\boldsymbol{w}) U(\boldsymbol{w})^{\intercal},$$

where $D(\boldsymbol{w}) = \operatorname{diag}(1/\lambda_1(\boldsymbol{w}), \ldots, 1/\lambda_{n-1}(\boldsymbol{w}), 0)$. Furthermore, $\mathbb{1}_N \in \ker \mathcal{L}^{\dagger}(\boldsymbol{w})$, since

$$\begin{aligned}
\mathcal{L}^{\dagger}(\boldsymbol{w})\mathbb{1}_N &= U(\boldsymbol{w}) D(\boldsymbol{w}) U^{\intercal}(\boldsymbol{w}) \mathbb{1}_N \\
&= U(\boldsymbol{w}) \begin{pmatrix} \frac{1}{\lambda_1(\boldsymbol{w})} & & & \\ & \cdots & & \\ & & \frac{1}{\lambda_{n-1}(\boldsymbol{w})} & \\ & & & 0 \end{pmatrix} \begin{pmatrix} u_1^{\intercal}(\boldsymbol{w}) \\ \cdots \\ u_{n-1}^{\intercal}(\boldsymbol{w}) \\ u_n^{\intercal}(\boldsymbol{w}) \end{pmatrix} \mathbb{1}_N \\
&= U(\boldsymbol{w}) \begin{pmatrix} \frac{1}{\lambda_1(\boldsymbol{w})} & & & \\ & \cdots & & \\ & & \frac{1}{\lambda_{n-1}(\boldsymbol{w})} & \\ & & & 0 \end{pmatrix} \begin{pmatrix} 0 \\ \cdots \\ 0 \\ \sqrt{n} \end{pmatrix} = U(\boldsymbol{w}) \begin{pmatrix} 0 \\ \cdots \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ \cdots \\ 0 \\ 0 \end{pmatrix},
\end{aligned}$$

where we used that the columns of $U(\boldsymbol{w})$ are orthonormal, and $u_n = \frac{1}{\sqrt{n}}\mathbb{1}_N$. □

The following lemma shows an example a way to adapt $\mathcal{L}(\boldsymbol{w})$ such that we obtain a non-singular matrix. This new matrix $\hat{\mathcal{L}}(\boldsymbol{w})$ is introduced, since we cannot compute the Schur complement of a singular matrix, consequently we will use the following lemma to prove Theorem 5.3.

Lemma 5.2. *Let $\mathcal{L}(\boldsymbol{w})$ define the state-dependent weighted Laplacian of a simple undirected connected graph. Then for any nonzero δ, the matrix*

$$\hat{\mathcal{L}}(\boldsymbol{w}) := \mathcal{L}(\boldsymbol{w}) + \frac{\delta}{n}\mathbb{1}_N \mathbb{1}_N^{\intercal}$$

is non-singular for any $\boldsymbol{w}$, and its inverse is given by

$$\hat{\mathcal{L}}^{-1}(\boldsymbol{w}) = \mathcal{L}^{\dagger}(\boldsymbol{w}) + \frac{1}{\delta n}\mathbb{1}_N\mathbb{1}_N^{\intercal}. \tag{35}$$

Proof. Consider the state-dependent weighted Laplacian $\mathcal{L}(\boldsymbol{w})$ of a simple undirected connected graph. Let $\lambda_1(\boldsymbol{w}), \ldots, \lambda_{n-1}(\boldsymbol{w}), 0$ represent the eigenvalues of $\mathcal{L}(\boldsymbol{w})$ and $u_1(\boldsymbol{w}), \ldots, u_{n-1}(\boldsymbol{w}), u_n(\boldsymbol{w})$ the corresponding eigenvectors. Then, for any value of δ, $\lambda_1(\boldsymbol{w}), \ldots, \lambda_{n-1}(\boldsymbol{w}), \delta$ are the eigenvalues of $\hat{\mathcal{L}}(\boldsymbol{w})$ corresponding to the eigenvectors $u_1(\boldsymbol{w}), \ldots, u_{n-1}(\boldsymbol{w}), u_n(\boldsymbol{w})$.
To see this: let $k \in \{1, \ldots, n-1\}$, then

$$\hat{\mathcal{L}}(\boldsymbol{w})u_k(\boldsymbol{w}) = \left(L(\boldsymbol{w}) + \frac{\delta}{n}\mathbb{1}_N\mathbb{1}_N^{\intercal}\right)u_k(\boldsymbol{w}) = \mathcal{L}(w)u_k(\boldsymbol{w}) + \frac{\delta}{n}\mathbb{1}_N\mathbb{1}_N^{\intercal}u_k(\boldsymbol{w}) = \lambda_k(\boldsymbol{w})u_k(\boldsymbol{w}),$$

since the eigenvectors $u_1(\boldsymbol{w}), \ldots, u_{n-1}(\boldsymbol{w})$ are orthogonal to $u_n = \frac{1}{\sqrt{n}}\mathbb{1}_N$, and hence $\mathbb{1}_N^{\intercal}u_k = 0$ for all $k \in \{1, \ldots, n-1\}$. For $k = n$, we have that

$$\begin{aligned}\hat{\mathcal{L}}(\boldsymbol{w})u_n &= \left(L(\boldsymbol{w}) + \frac{\delta}{n}\mathbb{1}_N\mathbb{1}_N^{\intercal}\right)u_n = \mathcal{L}(w)u_n + \frac{\delta}{n}\mathbb{1}_N\mathbb{1}_N^{\intercal}u_n \\ &= \frac{1}{\sqrt{n}}\mathcal{L}(w)\mathbb{1}_N + \frac{\delta}{n\sqrt{n}}\mathbb{1}_N\mathbb{1}_N^{\intercal}\mathbb{1}_N = \frac{\delta}{\sqrt{n}}\mathbb{1}_N = \delta u_n,\end{aligned}$$

hence δ is the eigenvalue belonging to the eigenvector u_n. We conclude that $\lambda_1(\boldsymbol{w}), \ldots, \lambda_{n-1}(\boldsymbol{w}), \delta$ are indeed the eigenvalues belonging to $\hat{\mathcal{L}}(\boldsymbol{w})$, since these eigenvalues are all positive, we conclude that $\hat{\mathcal{L}}(\boldsymbol{w})$ is non-singular.

In order to show that (35) defines the inverse of $\hat{\mathcal{L}}(\boldsymbol{w})$, we need to use that

$$\mathcal{L}(\boldsymbol{w})\mathcal{L}^{\dagger}(\boldsymbol{w}) = \mathcal{L}^{\dagger}(\boldsymbol{w})\mathcal{L}(\boldsymbol{w}) = I - \frac{1}{n}\mathbb{1}_N\mathbb{1}_N^{\intercal},$$

which is proved in [31]. It follows that

$$\begin{aligned}\hat{\mathcal{L}}^{-1}(\boldsymbol{w})\hat{\mathcal{L}}(\boldsymbol{w}) &= \left(\mathcal{L}^{\dagger}(\boldsymbol{w}) + \frac{1}{\delta n}\mathbb{1}_N\mathbb{1}_N^{\intercal}\right)\left(\mathcal{L}(\boldsymbol{w}) + \frac{\delta}{n}\mathbb{1}_N\mathbb{1}_N^{\intercal}\right) \\ &= \mathcal{L}^{\dagger}(\boldsymbol{w})\mathcal{L}(\boldsymbol{w}) + \mathcal{L}^{\dagger}(\boldsymbol{w})\frac{\delta}{n}\mathbb{1}_N\mathbb{1}_N^{\intercal} + \frac{1}{\delta n}\mathbb{1}_N\mathbb{1}_N^{\intercal}\mathcal{L}(\boldsymbol{w}) + \frac{1}{n^2}\mathbb{1}_N\mathbb{1}_N^{\intercal}\mathbb{1}_N\mathbb{1}_N^{\intercal} \\ &= \mathcal{L}^{\dagger}(\boldsymbol{w})\mathcal{L}(\boldsymbol{w}) + \frac{1}{n}\mathbb{1}_N\mathbb{1}_N^{\intercal} = I,\end{aligned}$$

where we used that $\mathbb{1}_N \in \ker \mathcal{L}^{\intercal}(\boldsymbol{w})$, and $\mathbb{1}_N \in \ker \mathcal{L}^{\dagger}(\boldsymbol{w})$. Similarly, it can be shown that $\hat{\mathcal{L}}(\boldsymbol{w})\hat{\mathcal{L}}^{-1}(\boldsymbol{w}) = I$. We conclude that $\hat{\mathcal{L}}^{-1}(\boldsymbol{w})$ defines the inverse of $\hat{\mathcal{L}}(\boldsymbol{w})$. □

We are now in the position to prove Theorem 5.3.

Proof of Theorem 5.3. Consider the state-dependent weighted Laplacian $\mathcal{L}(\boldsymbol{w})$ of a simple undirected connected graph. The effective memristance between two boundary nodes $i, j \in N_B$ of this graph can be computed as

$$R_{ij}(\boldsymbol{w}) = (e_i - e_j)^{\intercal}\mathcal{L}^{\dagger}(\boldsymbol{w})(e_i - e_j), \tag{36}$$

which using Lemma 5.2, is equal to

$$R_{ij}(\boldsymbol{w}) = (e_i - e_j)^\intercal \hat{\mathcal{L}}^{-1}(\boldsymbol{w})(e_i - e_j), \tag{37}$$

since $\mathbb{1}_N^\intercal(e_i - e_j) = 0$. Now, we want to use the Schur complement, see Theorem 4.1, to reduce the right-hand side of (37) to an equation solely depending on the boundary nodes of our network. In order to do so, we note that (37) can be written as

$$R_{ij}(\boldsymbol{w}) = (e_i - e_j)^\intercal v_{ij}, \tag{38}$$

with v_{ij} a solution to

$$\hat{\mathcal{L}}(\boldsymbol{w}) v_{ij} = e_i - e_j. \tag{39}$$

The matrix and vectors in (39) can be split into parts corresponding to the interconnected and boundary nodes as

$$\begin{pmatrix} \hat{\mathcal{L}}_{BB}(\boldsymbol{w}) & \hat{\mathcal{L}}_{BC}(\boldsymbol{w}) \\ \hat{\mathcal{L}}_{CB}(\boldsymbol{w}) & \hat{\mathcal{L}}_{CC}(\boldsymbol{w}) \end{pmatrix} \begin{pmatrix} v_{ij}^B \\ v_{ij}^C \end{pmatrix} = \begin{pmatrix} \bar{e}_i - \bar{e}_j \\ \mathbb{0}_{N_C} \end{pmatrix} \tag{40}$$

where we used that $i, j \in N_B$. Since $\hat{\mathcal{L}}(\boldsymbol{w})$ is non-singular, we can apply Theorem 4.1 on (40) to obtain

$$\begin{aligned} \bar{e}_i - \bar{e}_j &= \left(\hat{\mathcal{L}}_{BB}(\boldsymbol{w}) - \hat{\mathcal{L}}_{BC}(\boldsymbol{w}) \hat{\mathcal{L}}_{CC}^{-1}(\boldsymbol{w}) \hat{\mathcal{L}}_{CB}(\boldsymbol{w}) \right) v_{ij}^B \\ &= \hat{\mathcal{L}}_{\text{red}}(\boldsymbol{w}) v_{ij}^B. \end{aligned} \tag{41}$$

By Theorem 4.2 it follows that $\hat{\mathcal{L}}_{\text{red}}^{-1}$ exists, hence (41) can be rewritten as

$$v_{ij}^B = \hat{\mathcal{L}}_{\text{red}}^{-1}(\bar{e}_i - \bar{e}_j).$$

Substituting this in (38) gives

$$\begin{aligned} R_{ij}(\boldsymbol{w}) &= (e_j - e_j)^\intercal \begin{pmatrix} v_{ij}^B \\ v_{ij}^C \end{pmatrix} = \begin{pmatrix} (\bar{e}_i - \bar{e}_j)^\intercal & \mathbb{0}_{N_C} \end{pmatrix} \begin{pmatrix} v_{ij}^B \\ v_{ij}^C \end{pmatrix} \\ &= (\bar{e}_i - \bar{e}_j)^\intercal v_{ij}^B = (\bar{e}_i - \bar{e}_j)^\intercal \hat{\mathcal{L}}_{\text{red}}^{-1}(\bar{e}_i - \bar{e}_j). \end{aligned}$$

Then, since the Moore-Penrose inverse is a generalization of the regular matrix inverse, it follows that

$$R_{ij}(\boldsymbol{w}) = (\bar{e}_i - \bar{e}_j)^\intercal \hat{\mathcal{L}}_{\text{red}}^{\dagger}(\boldsymbol{w})(\bar{e}_i - \bar{e}_j). \tag{42}$$

Now, we notice that (36) is independent of δ, and hence (42) should be independent of δ as well. Therefore, we can take the limit of $\delta \to 0$, and we obtain:

$$\begin{aligned} R_{ij}(\boldsymbol{w}) &= \lim_{\delta \to 0} R_{ij}(\boldsymbol{w}) \\ &= \lim_{\delta \to 0} (\bar{e}_i - \bar{e}_j)^\intercal \hat{\mathcal{L}}^\dagger_{\text{red}}(\boldsymbol{w})(\bar{e}_i - \bar{e}_j) \\ &= \lim_{\delta \to 0} (\bar{e}_i - \bar{e}_j)^\intercal \mathcal{L}^\dagger_{\text{red}}(\boldsymbol{w})(\bar{e}_i - \bar{e}_j). \end{aligned}$$

Here, we used that

$$\begin{aligned} \lim_{\delta \to 0} \hat{\mathcal{L}}^\dagger_{\text{red}}(\boldsymbol{w}) &= \lim_{\delta \to 0} \Big(\hat{\mathcal{L}}_{BB}(\boldsymbol{w}) - \hat{\mathcal{L}}_{BC}(\boldsymbol{w}) \hat{\mathcal{L}}^{-1}_{CC}(\boldsymbol{w}) \hat{\mathcal{L}}_{CB}(\boldsymbol{w}) \Big) \\ &= \lim_{\delta \to 0} \Big(\mathcal{L}_{BB}(\boldsymbol{w}) + \frac{\delta}{n} \mathbb{1}_{N_B} \mathbb{1}^\intercal_{N_B} \\ &\quad - (\mathcal{L}_{CB}(\boldsymbol{w}) + \frac{\delta}{n} \mathbb{1}_{N_B} \mathbb{1}_{N_C})(\mathcal{L}_{CC}(\boldsymbol{w}) + \frac{\delta}{n} \mathbb{1}_{N_C} \mathbb{1}_{N_C})^{-1} (\mathcal{L}_{BC}(\boldsymbol{w}) + \frac{\delta}{n} \mathbb{1}_{N_C} \mathbb{1}_{N_B}) \Big) \\ &= \mathcal{L}_{BB}(\boldsymbol{w}) - \mathcal{L}_{CB}(\boldsymbol{w}) \mathcal{L}^{-1}_{CC}(\boldsymbol{w}) \mathcal{L}_{BC}(\boldsymbol{w}) \\ &= \mathcal{L}_{\text{red}}(\boldsymbol{w}), \end{aligned}$$

which proves the claimed identity (34). □

Theorem 5.3 allows us to compute the effective memristance between any two boundary nodes $r, s \in N_B$ in our original graph from the Kron reduced weighted Laplacian $\mathcal{L}_{\text{red}}(\boldsymbol{w})$; this can be used to find an expression for the external behavior of a network.

5.5 Terminal behavior

In this chapter, we used electrical circuit and graph theory to model a network of memristive devices. These networks are of interest since their external behavior, i.e. characteristic $i(t) - v(t)$ curves, might be different from that of single memristive devices which is of interest when considering memristive devices as synapses in artificial neural networks. In the previous section, we showed how the effective memristance between any pair of boundary nodes in a network of memristive devices can be computed. Here, we will show how the definition of effective memristance can be used to express the dynamics of a network of memristive devices by a single memristive device.

Consider a network of K memristive devices and recall that the memristive device situated at edge k between node i and j is described as

$$\begin{aligned} \dot{\boldsymbol{w}}_{ij} &= f(\boldsymbol{w}_{ij}, p_k v_k), \\ i_k &= G(\boldsymbol{w}_{ij}) v_k. \end{aligned}$$

In Section 5.3 we saw that any connected network then can be described by

$$\begin{aligned}\dot{\boldsymbol{w}} &= F\left(\boldsymbol{w}, P\begin{pmatrix}\mathcal{D}_B^\intercal \\ \mathcal{D}_C^\intercal \mathcal{L}_{\mathrm{ac}}(\boldsymbol{w})\end{pmatrix}\psi_B\right), \\ J_B &= \mathcal{L}_{\mathrm{red}}(\boldsymbol{w})\psi_B,\end{aligned}$$

where F is the vector having as entries f evaluated at the different edges, and P is a diagonal matrix having as entries the polarities of the memristive devices.

Then using Definition 5.3, we can compute the effective memristance between any pair of boundary nodes. From that it follows that the relation of the current i_{rs} and voltage v_{rs} between node $r, s \in N_B$ is given by

$$\begin{aligned}\dot{\boldsymbol{w}} &= F\left(\boldsymbol{w}, P\begin{pmatrix}\mathcal{D}_B^\intercal \\ \mathcal{D}_C^\intercal \mathcal{L}_{\mathrm{ac}}(\boldsymbol{w})\end{pmatrix}\psi_B\right), \\ i_{rs} &= \frac{1}{R_{rs}(\boldsymbol{w})} v_{rs},\end{aligned} \tag{43}$$

where $R_{rs}(\boldsymbol{w})$ denotes the effective memristance between node $r, s \in N_B$. This dynamical systems represents the external behavior of our original network between node $r, s \in N_B$ and can be seen as a single memristive device representing the dynamic behavior between two boundary nodes of the original network of memristive devices.

An important thing to note here is that we found an expression for the external behavior, i.e. the relation between v_{rs} and i_{rs} for $r, s \in N_B$, by reducing the network to a graph which solely contains the boundary nodes of our original graph. However, we did not reduce the number of state variables, weights, in the network; the state of the external behavior is still given by a K dimensional weight vector $\boldsymbol{w}$.

The next section will be used to illustrate these ideas with an example.

5.6 Example

Consider the electrical network depicted in Figure 21. As before, the dynamics of the k-th memristive device is described by

$$\begin{aligned}\dot{\boldsymbol{w}}_k &= f(\boldsymbol{w}, p_k v_k), \\ i_k &= G(w_k) v_k,\end{aligned}$$

which makes that the dynamics of the network is given by

$$\begin{aligned}\dot{\boldsymbol{w}} &= F(\boldsymbol{w}, PV), \\ J &= \mathcal{L}(\boldsymbol{w})\psi.\end{aligned}$$

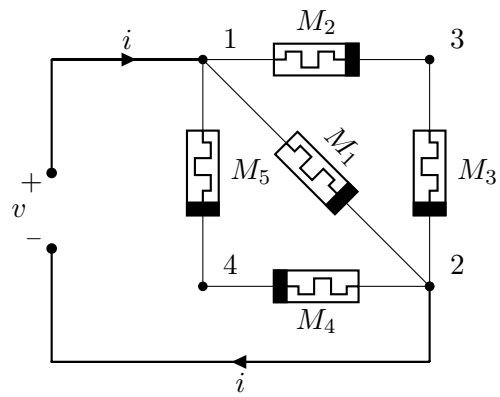

Figure 21: Interconnection of 5 memristive devices with $N_B = \{1, 2\}$. Here, i and v are the port current and voltage of the device, respectively.

Here, F is the vector having as entries f evaluated at the different edges, P is a diagonal matrix having as entries the polarities of the memristive devices, and $\mathcal{L}(\boldsymbol{w}) = \mathcal{D}\mathcal{W}(\boldsymbol{w})\mathcal{D}^\intercal$ denotes state-dependent the weighted Laplacian of the graph associated to the network. For this example, the incidence and weight matrix are given by

$$\mathcal{D} = \begin{pmatrix} 1 & 1 & 0 & 0 & -1 \\ -1 & 0 & -1 & 1 & 0 \\ 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 1 \end{pmatrix}, \text{ and } \mathcal{W}(\boldsymbol{w}) = \begin{pmatrix} G(\boldsymbol{w}_1) & & & & \\ & G(\boldsymbol{w}_2) & & & \\ & & G(\boldsymbol{w}_3) & & \\ & & & G(\boldsymbol{w}_4) & \\ & & & & G(\boldsymbol{w}_5) \end{pmatrix},$$

respectively. Hence, the state-dependent weighted Laplacian $\mathcal{L}(\boldsymbol{w})$ can be computed as

$$\mathcal{L}(\boldsymbol{w}) = \begin{pmatrix} G(\boldsymbol{w}_1) + G(\boldsymbol{w}_2) + G(\boldsymbol{w}_5) & -G(\boldsymbol{w}_1) & -G(\boldsymbol{w}_2) & -G(\boldsymbol{w}_5) \\ -G(\boldsymbol{w}_1) & G(\boldsymbol{w}_1) + G(\boldsymbol{w}_3) + G(\boldsymbol{w}_4) & -G(\boldsymbol{w}_3) & -G(\boldsymbol{w}_4) \\ -G(\boldsymbol{w}_2) & -G(\boldsymbol{w}_3) & G(\boldsymbol{w}_2) + G(\boldsymbol{w}_3) & 0 \\ -G(\boldsymbol{w}_5) & -G(\boldsymbol{w}_4) & 0 & G(\boldsymbol{w}_3) + G(\boldsymbol{w}_4) \end{pmatrix}.$$

Splitting the set of nodes in the boundary nodes $N_B = \{1, 2\}$ and internal connection nodes $N_C = \{3, 4\}$ gives the relation

$$\begin{pmatrix} J_B \\ J_C \end{pmatrix} = \begin{pmatrix} \mathcal{L}_{BB}(\boldsymbol{w}) & \mathcal{L}_{BC}(\boldsymbol{w}) \\ \mathcal{L}_{CB}(\boldsymbol{w}) & \mathcal{L}_{CC}(\boldsymbol{w}) \end{pmatrix} \begin{pmatrix} \psi_B \\ \psi_C \end{pmatrix},$$

where

$$\mathcal{L}_{BB}(\boldsymbol{w}) = \begin{pmatrix} G(\boldsymbol{w}_1) + G(\boldsymbol{w}_2) + G(\boldsymbol{w}_5) & -G(\boldsymbol{w}_1) \\ -G(\boldsymbol{w}_1) & G(\boldsymbol{w}_1) + G(\boldsymbol{w}_3) + G(\boldsymbol{w}_4) \end{pmatrix},$$

$$\mathcal{L}_{BC}(\boldsymbol{w}) = \begin{pmatrix} -G(\boldsymbol{w}_2) & -G(\boldsymbol{w}_5) \\ -G(\boldsymbol{w}_3) & -G(\boldsymbol{w}_4) \end{pmatrix},$$

$$\mathcal{L}_{CC}(\boldsymbol{w}) = \begin{pmatrix} G(\boldsymbol{w}_2) + G(\boldsymbol{w}_3) & 0 \\ 0 & G(\boldsymbol{w}_4) + G(\boldsymbol{w}_5) \end{pmatrix},$$

and $\mathcal{L}_{CB}(\boldsymbol{w}) = \mathcal{L}_{BC}^{\intercal}(\boldsymbol{w})$. It follows that the nodal currents and voltage potentials of the boundary nodes are related as

$$\begin{pmatrix} j_1 \\ j_2 \end{pmatrix} = \mathcal{L}_{\text{red}}(\boldsymbol{w}) \begin{pmatrix} \psi_1 \\ \psi_2 \end{pmatrix},$$

where

$$\mathcal{L}_{\text{red}}(\boldsymbol{w}) = \mathcal{L}_{BB}(\boldsymbol{w}) - \mathcal{L}_{BC}(\boldsymbol{w})\mathcal{L}_{CC}^{-1}(\boldsymbol{w})\mathcal{L}_{CB}(\boldsymbol{w}).$$

By explicitly calculating $\mathcal{L}_{\text{red}}(\boldsymbol{w})$, one finds that the effective memristance between the boundary nodes is given by

$$\begin{aligned} R(\boldsymbol{w}) &= \begin{pmatrix} 1 & -1 \end{pmatrix} \mathcal{L}_{\text{red}}^{\dagger}(\boldsymbol{w}) \begin{pmatrix} 1 \\ -1 \end{pmatrix} \\ &= \left(G(\boldsymbol{w}_1) + \frac{G(\boldsymbol{w}_2)G(\boldsymbol{w}_3)}{G(\boldsymbol{w}_2) + G(\boldsymbol{w}_3)} + \frac{G(\boldsymbol{w}_4)G(\boldsymbol{w}_5)}{G(\boldsymbol{w}_4) + G(\boldsymbol{w}_5)} \right)^{-1}. \end{aligned}$$

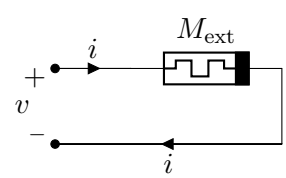

Figure 22: External behavior of the network depicted in Figure 21. Here, i and v are the port current and voltage of the device, respectively.

Now, the network in Figure 21 can be reduced to the memristive device M_{ext} depicted in Figure 22

which is mathematically described as

$$\begin{aligned}
\dot{\boldsymbol{w}} &= F(\boldsymbol{w}, PV), \\
i &= \frac{1}{R(\boldsymbol{w})} v \\
&= \left(G(\boldsymbol{w}_1) + \frac{G(\boldsymbol{w}_2)G(\boldsymbol{w}_3)}{G(\boldsymbol{w}_2) + G(\boldsymbol{w}_3)} + \frac{G(\boldsymbol{w}_4)G(\boldsymbol{w}_5)}{G(\boldsymbol{w}_4) + G(\boldsymbol{w}_5)} \right) v,
\end{aligned}$$

where i and v are the port current and voltage of the device. Here, the effective memristance $R(\boldsymbol{w})$ still depends on the state variables of all the memristive devices in our original network as we mentioned in the previous section.

Simulation results for this and other networks of memristive devices will be given and compared in the next chapter.

6 Simulations

In this chapter, we will show simulation results of the $v(t)$-$i(t)$ curves and the effective memristances for series, parallel interconnections and random graphs, consecutively. These results will be compared with the existing relations of effective resistance for networks of resistors. Furthermore, for the random graphs we will compare the effective memristance between different sets of boundary nodes, which shows the effect of the network structure on the effective memristance between two boundary nodes.

Throughout this chapter, we consider networks of memristive devices where a single memristive device is described by the affine model as

$$\begin{aligned}\dot{\boldsymbol{w}}_k &= p_k v_k,\\ i_k &= (\alpha \boldsymbol{w}_k + \beta) v_k.\end{aligned}$$

This makes that the total model can be described as

$$\begin{aligned}\dot{\boldsymbol{w}} &= PV = P\mathcal{D}^\intercal \psi,\\ J &= \mathcal{L}(\boldsymbol{w})\psi = \mathcal{D}\mathcal{W}(\boldsymbol{w})\mathcal{D}^\intercal \psi,\end{aligned}$$

where P is a diagonal matrix having as entries the polarities of the memristive devices. The incidence matrix $\mathcal{D}$ specifies the network structure, and the entries of $\mathcal{W}(\boldsymbol{w}) = \text{diag}\left(\{\alpha \boldsymbol{w}_k + \beta\}_{k=1}^K\right)$ give the memductances of the memristive devices in the network. Here, K is the number of memristive devices in the network.

The simulation results are obtained by using the effective memristance, and following the same calculation steps as in Section 5.6. Furthermore, we used that the differential equation of $\boldsymbol{w}$ can be rewritten as a function only depending on ψ_B and not on ψ_C, see Section 5.3.

All simulations are executed for the parameters $\alpha = 1.884\ mS/W_b$, $\beta = 0.25\ mS$, and the initial condition $w_0 = -0.03 \cdot \mathbb{1}_K$. Furthermore, a sinusoidal port-voltage $v(t) = 4\sin(2\pi f_c + \pi/3)$ with a frequency f_c of 20 Hz is applied to the circuit.

6.1 Series interconnection

Consider a series interconnection of K memristive devices, as depicted in Figure 23. Here, the memristive devices are placed such that the current enters the devices at their negative side.
Figure 24 shows simulation results of series interconnections consisting of $1, 2, 3$, and 4 memristive devices, respectively. It can be seen that the effective memristance between the boundary nodes, increases with an increasing number of memristive devices.

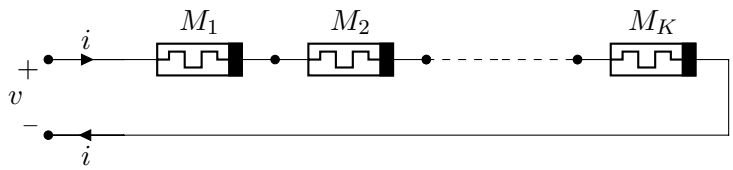

Figure 23: Series interconnection of K memristive devices. Here, i and v are the port current and voltage, respectively.

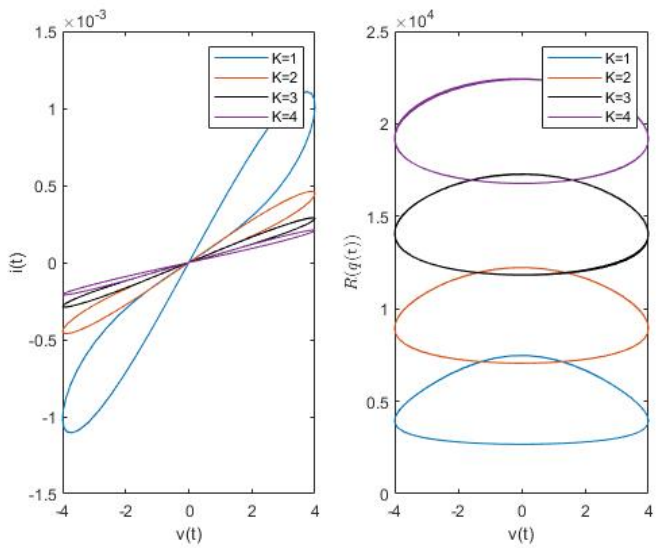

Figure 24: Results obtained for series interconnections of K memristive devices. Left, $v(t)$-$i(t)$ curves. Right, effective memristance between the boundary nodes compared with the port voltage.

This is comparable to the relation we have for networks of resistors. Namely, for a series interconnection of K resistors, the effective resistance is given by

$$R_{\text{tot}} = \sum_{k=1}^{K} R_k,$$

where R_k denotes the resistance of the k-th resistor in the interconnection, and K is the total number of resistors.

6.2 Parallel interconnection

Consider a parallel interconnection of K memristive devices, as depicted in Figure 25. Here, the memristive devices are placed such that the current enters the devices at their negative side. Figure 26 shows simulation results of parallel interconnections consisting of $1, 2, 3$, and 4 memristive

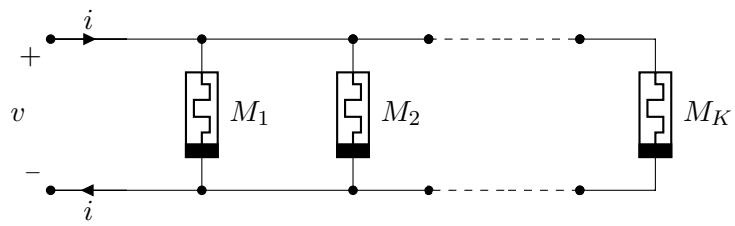

Figure 25: Parallel interconnection of K memristive devices. Here, i and v are the port current and voltage, respectively.

devices, respectively. It can be seen that the effective memristance between the boundary nodes, decreases with an increasing number of memristive devices. This is comparable to the relation we

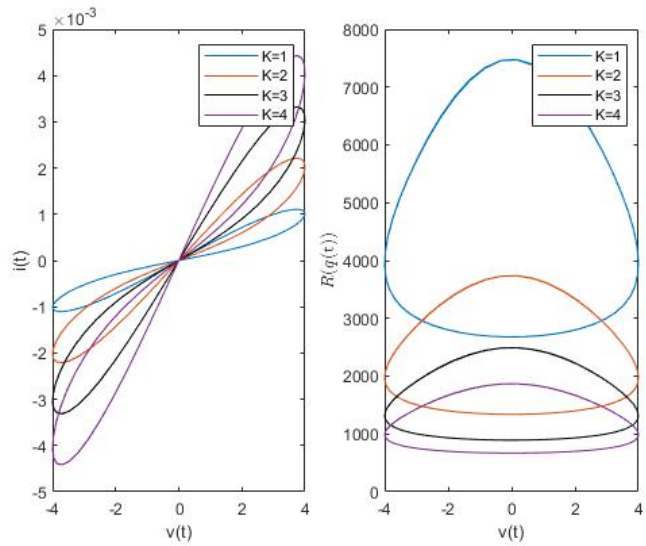

Figure 26: Results obtained for parallel interconnections of K memristive devices. Left, $v(t)$-$i(t)$ curves. Right, effective memristance between the boundary nodes compared with the port voltage.

have for networks of resistors. Namely, for a parallel interconnection of K resistors, the effective resistance is given by

$$\frac{1}{R_{\text{tot}}} = \frac{1}{\sum_{k=1}^{K} R_k},$$

where R_k denotes the resistance of the k-th resistor in the interconnection, and K is the total number of resistors. Besides the fact that the total memristance of the networks decreases as we increase the number of memristive devices in the interconnection, we see that the shape of the effective memristance curve in Figure 26 changes; when the number of memristive devices in the interconnection increases, the effective memristance curve gets flatter. This is an interesting

observation since this implies that the shape of the hysteresis loop of the total network of memristive devices is influenced by the network structure. We will also see this when looking to random graphs as we do in the next section.

6.3 Random graphs

In Section 5.6, we found a mathematical expression for the external behavior of the network depicted in Figure 27. Figure 28 depicts a simulation result of this external behavior. Also, the simulation

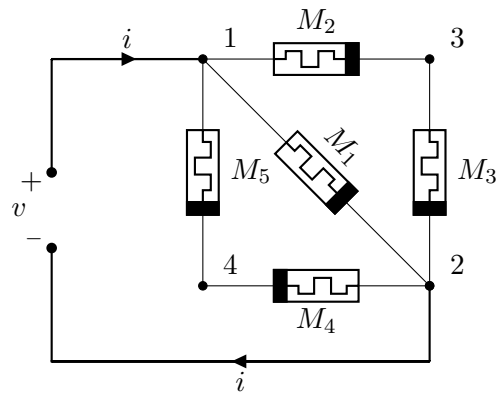

Figure 27: Interconnection of 5 memristive devices with $N_B = \{1, 2\}$. Here, i and v are the port current and voltage of the device, respectively.

result of a single memristive device described by the affine model and having the same parameters and input signal as the network is depicted in Figure 28. We note that the effective memristance of the network is smaller than that of a single memristive device. This can be explained by the fact that the network in Figure 27 is in fact a parallel connection with three memristive devices, and so since the port voltage is devided by those three paths in the graph, the effective memristance of the network is smaller than that of a single memristive device. Also, it is interesting to note that the shape of the curve is different for the network and the single device. This implies that we can design memristive devices with differently shaped hysteresis loops out of networks of equal devices, which is of interest when we consider memristive devices as building blocks for neuromorphic computing.

In addition, instead of considering the effective memristance of a network for only one pair of boundary nodes, we can consider the effective memristance of a network between different pairs of boundary nodes. Figure 30 shows a simulation result for the effective memristance between two different pairs of boundary nodes of the network depicted in Figure 29. We note that the effective memristance curves, and so the hysteresis loops, for these two different pairs of boundary nodes depict differently shaped figures. However, if we consider an (almost) symmetric graph as depicted

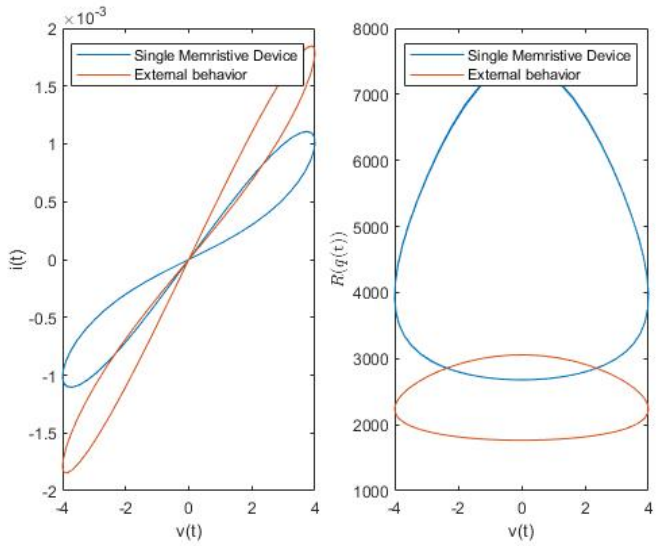

Figure 28: Results obtained for the electrical network depicted in Figure 27 compared with those of a single memristive devices, see Figure 12. Left, $v(t)$-$i(t)$ curves. Right, effective memristance between the boundary nodes compared with the port voltage.

in Figure 31, then the external behavior between different sets of boundary nodes which are mirrored to each other is similar, see Figure 32.

To summarize, we saw that the structure of a network of memristive devices influences its external behavior, i.e. different networks leads to a different dynamical behavior of the memristive device which represents the network of memristive devices. Furthermore, not only the structure of the network influences the external behavior, but also the set of boundary nodes which are considered. However, the effective memristance curve and the hysteresis loop depicted similar results when considering mirrored pairs of boundary nodes in (almost) symmetric graphs. Concluding, we saw that we can influence the shape of the characteristic hysteresis loop of a memristive device build out of a network of memristive device by considering different graph structures, and considering different pairs of boundary nodes.

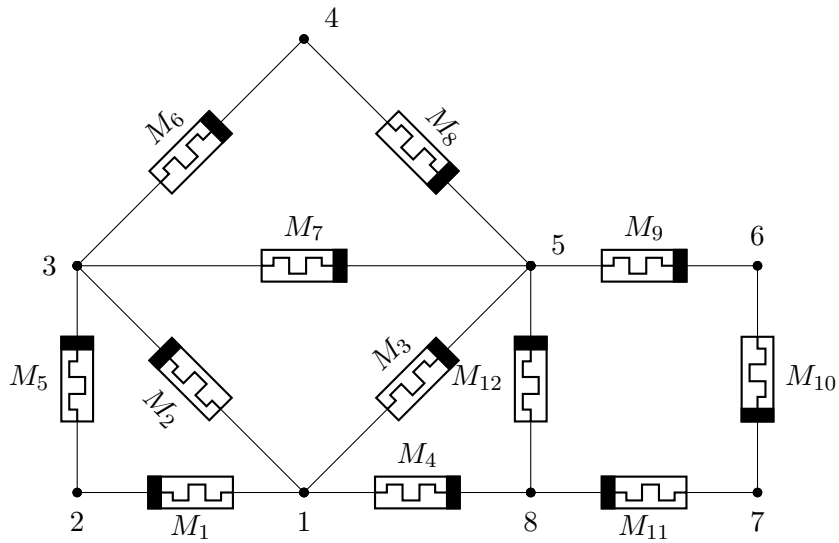

Figure 29: Interconnection of 12 memristive devices. Here, we consider $N_B = \{2, 4, 7\}$ as the set of boundary nodes.

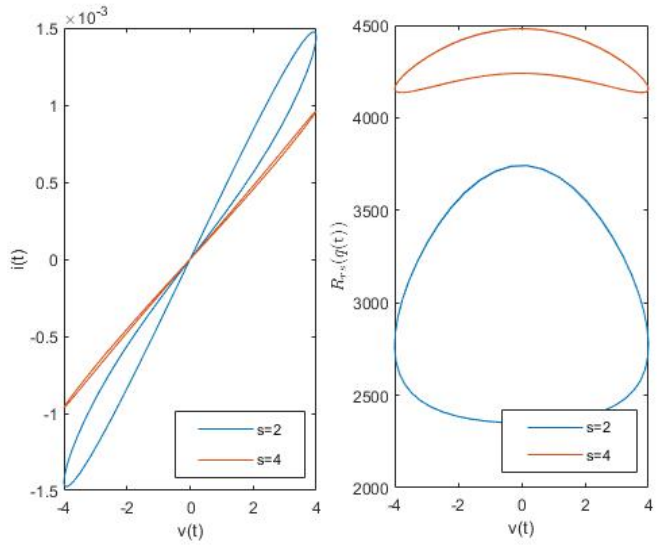

Figure 30: Results obtained for the electrical network depicted in Figure 29. Here, we compared the external behavior of the network between the boundary node $r = 7$, and $s = 2$ and $s = 4$, respectively. Left, $v(t)$-$i(t)$ curves. Right, effective memristance between the boundary nodes compared with the port voltage.

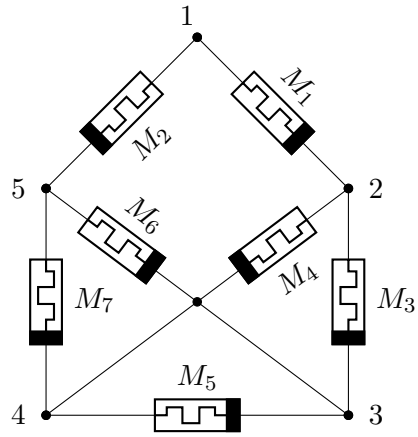

Figure 31: Interconnection of 7 memristive devices. Here, we consider $N_B = \{1, 2, 3, 4, 5\}$ as the set of boundary nodes.

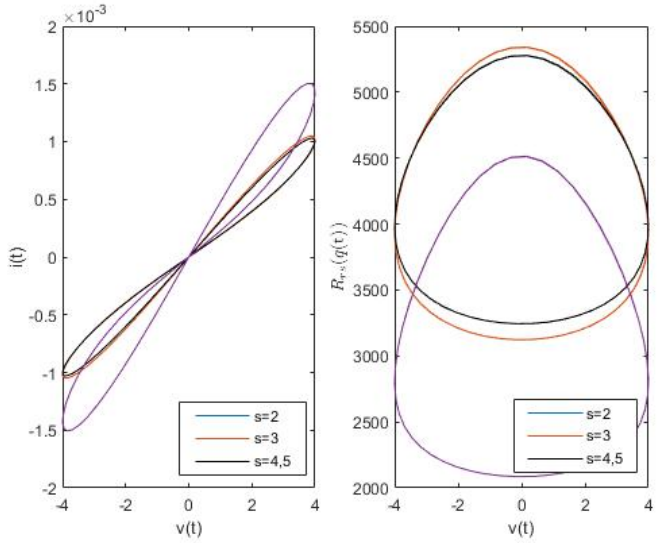

Figure 32: Results obtained for the electrical network depicted in Figure 31. Here, we compared the external behavior of the network between the boundary node $r = 1$, and $s = 2$, $s = 3$, $s = 4$, and $s = 5$, respectively. Left, $v(t)$-$i(t)$ curves. Right, effective memristance between the boundary nodes compared with the port voltage.

7 Conclusion

In this thesis, we introduced neuromorphic computing as a new computing paradigm whose architecture is based on artificial neural networks. We did this by briefly explaining biological neural networks and linking this to artificial neural networks.

Thereafter, we introduced memristive devices, and we saw that they are characterized by a hysteresis loop pinched at the origin. Various mathematical models exist for describing the behavior of memristive devices, some of which can be linked to the synaptic weight update rule spike-timing dependent plasticity in biological neural networks. Henceforth, we suggested to use memristive devices as the building blocks for neuromorphic computing architectures. More precisely, we suggested memristive devices as the synaptic weights in artificial neural networks.

Then, in order to realize artificial neural networks with an optimal functionality, the memristive devices should have a certain behavior, i.e. the hysteresis loop of a device should have a certain shape. This might not be achieved by single memristive devices which made us consider networks of memristive devices.

We showed that a network of memristive device represents a memristive device itself, and we developed tools to describe the dynamical behavior of this memristive device. This is realized by describing a network of memristive devices as a graph in which the memristive devices correspond to the edges. Thereafter, we used Kron reduction to reduce the network to a network only consisting of the boundary nodes of the original network. The notion of effective memristance was then introduced to derive a relation between the currents through and the voltage potentials across the edges in the reduced network. We concluded this section by using the notion of effective memristance to find a mathematical expression for the dynamical behavior of the single memristive device representing the original network of memristive devices, also called the external behavior of the network.

Finally, in the section with simulation results, we compared the effective memristance and characteristic hysteresis loops of various networks of memristive devices. We saw that the network dynamics of a network of memristive devices does influence the shape of its effective memristance curve and, consequently that of its hysteresis loop. In addition, we saw that the effective memristance between different pairs of boundary nodes in the same network already depicts differently shaped curves. This suggest that certain desired dynamic behavior of memristive devices can be achieved by proper design of networks of memristive devices.

A goal of future research can be to find mathematical tools to describe the network structure corresponding to desired dynamic behavior, i.e. specific hysteresis loops. When considering the work we

did until now, a extension of the results can be to not only reduce our original network to a network consisting of only the boundary nodes, but to also reduce the dimension of the state-dynamics of the network. Furthermore, it would be interesting to study how the developed mathematical tools in this thesis can be applied to more general memristive device models, and to networks consisting of different types of memristive devices as well as additional circuit elements such as capacitors and inductors.

References

[1] J. Glanz, "Google Details, and Defends, Its Use of Electricity," https://www.nytimes.com/2011/09/09/technology/google-details-and-defends-its-use-of-electricity.html, 2011, Accessed: 25-02-2019.

[2] G. E. Moore, "Cramming more components onto integrated circuits," *Proceedings of the IEEE*, vol. 86, no. 1, pp. 82–85, 1998.

[3] T. N. Theis and H.-S. P. Wong, "The end of Moore's law: A new beginning for information technology," *Computing in Science & Engineering*, vol. 19, no. 2, pp. 41–50, 2017.

[4] O. Krestinskaya, A. P. James, and L. O. Chua, "Neuro-memristive circuits for edge computing: A review," *arXiv preprint arXiv:1807.00962*, 2018.

[5] E. Tang and D. S. Bassett, "Control of dynamics in brain networks," *arXiv preprint arXiv:1701.01531*, 2017.

[6] B. Linares-Barranco, T. Serrano-Gotarredona, L. A. Camuñas-Mesa, J. A. Perez-Carrasco, C. Zamarreño-Ramos, and T. Masquelier, "On spike-timing-dependent-plasticity, memristive devices, and building a self-learning visual cortex," *Frontiers in Neuroscience*, vol. 5, p. 26, 2011.

[7] T. Chang, Y. Yang, and W. Lu, "Building neuromorphic circuits with memristive devices," *IEEE Circuits and Systems Magazine*, vol. 13, no. 2, pp. 56–73, 2013.

[8] T. Doorenbosch, "Huidige computers vreselijk uit de tijd," https://www.agconnect.nl/artikel/huidige-computers-vreselijk-uit-de-tijd, 2011, Accessed: 25-02-2019.

[9] M. Biehl, *Lecture notes Neural Networks and Computational Intelligence*, 2018-2019.

[10] T. Serrano-Gotarredona, T. Masquelier, T. Prodromakis, G. Indiveri, and B. Linares-Barranco, "STDP and STDP variations with memristors for spiking neuromorphic learning systems," *Frontiers in Neuroscience*, vol. 7, p. 2, 2013.

[11] D. O. Hebb *et al.*, *The organization of behavior.* New York: Wiley and Sons, 1949.

[12] G.-Q. Bi and M.-M. Poo, "Synaptic modification by correlated activity: Hebb's postulate revisited," *Annual Review of Neuroscience*, vol. 24, no. 1, pp. 139–166, 2001.

[13] H. Markram, W. Gerstner, and P. J. Sjöström, *Spike-timing-dependent plasticity: a comprehensive overview.* Frontiers, 2012, vol. 4.

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

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

[16] C. A. Desoer and E. S. Kuh, *Basic circuit theory.* McGraw-Hill, 1969.

[17] L. O. Chua and S. M. Kang, "Memristive devices and systems," *Proceedings of the IEEE*, vol. 64, no. 2, pp. 209–223, 1976.

[18] R. K. Budhathoki, M. P. Sah, S. P. Adhikari, H. Kim, and L. O. Chua, "Composite behavior of multiple memristor circuits," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 60, no. 10, pp. 2688–2700, 2013.

[19] A. J. van der Schaft and J. M. A. Scherpen, *Lecture notes Modeling and Control of Complex Nonlinear Engineering Systems*, 2015.

[20] J. C. Willems, "Dissipative dynamical systems part i: General theory," *Archive for Rational Mechanics and Analysis*, vol. 45, no. 5, pp. 321–351, 1972.

[21] D. Yu, H. H.-C. Iu, Y. Liang, T. Fernando, and L. O. Chua, "Dynamic behavior of coupled memristor circuits," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 62, no. 6, pp. 1607–1616, 2015.

[22] S. Saïghi, C. G. Mayr, T. Serrano-Gotarredona, H. Schmidt, G. Lecerf, J. Tomas, J. Grollier, S. Boyn, A. F. Vincent, and D. Querlioz, "Plasticity in memristive devices for spiking neural networks," in *Frontiers in Neuroscience*, vol. 9, 2015, paper 51.

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

[24] C. Yakopcic, T. M. Taha, G. Subramanyam, R. E. Pino, and S. Rogers, "A memristor device model," *IEEE Electron Device Letters*, vol. 32, no. 10, pp. 1436–1438, 2011.

[25] M. D. Pickett, D. B. Strukov, J. L. Borghetti, J. J. Yang, G. S. Snider, D. Stewart, and R. S. Williams, "Switching dynamics in titanium dioxide memristive devices," *Journal of Applied Physics*, vol. 106, no. 7, p. 074508, 2009.

[26] C. Godsil and G. F. Royle, *Algebraic graph theory.* Springer Science & Business Media, 2013, vol. 207.

[27] F. Zhang, *The Schur complement and its applications.* Springer Science & Business Media, 2006, vol. 4.

[28] R. Penrose, "A generalized inverse for matrices," *Mathematical Proceedings of the Cambridge Philosophical Society*, vol. 51, no. 3, pp. 406–413, 1955.

[29] G. Kron, *Tensor analysis of networks.* Wiley, 1939.

[30] A. J. van der Schaft, "Characterization and partial synthesis of the behavior of resistive circuits at their terminals," *Systems & Control Letters*, vol. 59, no. 7, pp. 423–428, 2010.

[31] I. Gutman and W. Xiao, "Generalized inverse of the laplacian matrix and some applications," *Bulletin (Académie serbe des sciences et des arts. Classe des sciences mathématiques et naturelles. Sciences mathématiques)*, pp. 15–23, 2004.

[32] S. Leon, *Linear algebra with applications*, 8th ed. Pearson, 2014.