THESIS FOR THE DEGREE OF LICENTIATE OF ENGINEERING

Theoretical simulations of environment-sensitive dynamical systems for advanced reservoir computing applications

VASILEIOS ATHANASIOU

Department of Microtechnology and Nanoscience - MC2
Chalmers University of Technology
Gothenburg, Sweden, 2018

Theoretical simulations of environment-sensitive dynamical systems for advanced reservoir computing applications
VASILEIOS ATHANASIOU

Department of Microtechnology and Nanoscience – MC2
EMSL Laboratory
SE-412 96 Gothenburg, Sweden
Telephone + 46 (0)31-772 1000

ISSN 1652-0769
Technical Report MC2 - 395

Printed in Gothenburg, Sweden 2018

Theoretical simulations of environment-sensitive dynamical systems for advanced reservoir computing applications

Vasileios Athanasiou

Department of Microtechnology and Nanoscience - MC2

Chalmers University of Technology

Abstract

The possibility of building intelligent sensing substrates that both collect information about an environment and analyze it in real-time has been investigated theoretically. In a typical setup, a dynamical system is assumed to interact with the environment over time. The system operates as a reservoir computer acting as a reservoir of states. Due to the reservoir-environment interaction, the information about the environment is encoded in the state of the reservoir. The information stored in the system can be inferred (decoded) by analyzing the reservoir state, which is done by observing how a system responds to an external stimulus being an external drive signal. This signal is optimized to ensure that under different environmental conditions the reservoir visits distinct regions of the configuration space. If such a behavior is possible, then a relatively simple readout layer can be used to achieve efficient sensing. These ideas have been examined theoretically by simulating various networks of environment-sensitive elements: the memristor, the capacitor, the constant phase element, and the organic field effect transistor element. It was found that heterogeneity of the network is important for sensing. The simulations were done in the context of ion sensing, which is an extremely complex, many-body, and multi-scale modeling problem. A generic electrical circuit simulator has been developed with a focus on understanding transient dynamics. The constant phase element has been identified as an important primitive that is essential for modeling the experimental data. A new algorithm has been develop to model its transient behavior. Likewise, the same was done for the organic electrochemical transistor. To quantify the sensing capacity of an environment sensitive network a precise mathematical measure has been introduced, the state separability index, and evaluated in numerical experiments. The theoretical work has been supported by the related set of experiments.

"When you sail for Ithaca, wish for the journey to be long, full of adventures, full of knowledge. "

Constantinos P. Cavafis

Acknowledgements

Firstly, I would like to express my sincere gratitude to my supervisor Zoran Konkoli for the continuous support of my doctoral studies and related research, for listening to me and spending time for discussions, for his patience, motivation, and immense knowledge. His guidance helped me in all the time of research and writing of this thesis. I could not have imagined having a better supervisor for my doctoral studies.

I thank my co-supervisors Aldo Jesorka and Dag Winkler for their useful comments.

I thank the collaborators of the RECORD-IT project for the stimulating scientific discussions and for supplying me with useful experimental data.

Last but not the least, I would like to thank my family: my wife Christina for spending her life with me and supporting me, my son Alexandros who was born almost in the beginning of my doctoral studies and gave me the power to do research for a better future of the humanity, to my parents and my brother’s family for supporting me spiritually throughout writing this thesis and my life in general.

Contents

List of Abbreviations

MFB	memristor element connected with a delay feedback mechanism
NOMFET	nanoparticle organic memory field-effect transistor
OECT	organic electrochemical transistor
SWEET	state weaving environment echo tracker
CPE	constant phase element
MNA	modified nodal analysis
SC	sensing capacity
ODE	ordinary differential equation
RLC	resistor-inductor-capacitor
RC	resistor-capacitor
DC	direct current
AC	alternating current

List of publications

I. Vasileios Athanasiou & Zoran Konkoli (2017) On using reservoir computing for sensing applications: exploring environment-sensitive memristor networks, International Journal of Parallel, Emergent and Distributed Systems, DOI: 10.1080/17445760.2017.1287264

II. V. Athanasiou, Z. Konkoli, On the use of collaborative interactions for embedded sensing applications: Memristor networks as intelligent sensing substrates, Submitted, Under review

III. Athanasiou V, Konkoli Z. On the efficient simulation of electrical circuits with constant phase elements: The warburg element as a test case. Int J Circ Theor Appl. 2018;46:1072–1090. https://doi.org/10.1002/cta.2474

IV. V. Athanasiou, S. Pecqeuer, D. Vuillaume, Z. Konkoli, On a generic theory of the Organic Electrochemical Transistor dynamics, manuscript under preparation

Chapter 1

Introduction

1.1 Background

Moore's law predicts that the number of transistors at the chip doubles roughly every second year. [21] However, it is likely that this trend will slow down, owing to practical engineering limitations or specific physical effects pertinent to small scales, such as wiring problems or electron tunneling. There are problems that are simply too complex and that cannot be handled by the traditional CMOS technology. In somewhat technical terms, one says that there are information processing applications that do not scale according to the Moore law. Typical examples include problems in distributed, real-time, or embedded information processing applications. Accordingly, there is a need to develop alternative information processing solutions by using non-CMOS based technologies. In the information processing industry, and especially semiconductor industry, one talks about *functional diversification*. The field of unconventional computation has emerged as a response to this functional diversification challenge. Up to date unconventional computation encompasses a plethora of computing frameworks, such as neuromorphic computing, molecular computing, reaction-diffusion computing, or quantum computing, and is ever increasing in its scope. [28, 9, 10, 1, 2, 17]

In particular, reservoir computing has gained a considerable interest among the unconventional computation community in the recent decade, both as a model of computation and as a remarkably practical approach for realizing neuromorphic computation. Historically, the field of reservoir computing started as an insight about behavior of synaptic weights during the neural network training process. [13, 12, 20, 29] During the supervised learning the weights need to be adjusted to achieve a desired computation. It has been realized that only a limited set of weights is adjusted in the network training process. While it is true that a neural network is essentially a geometry free object, the weights that change belong mostly to the links in the network that can be naturally described as an "outer layer". This led to the further insight that instead of neural network one could use an arbitrary dynamical system as the core, and augment it with the outer layer. The dynamical system used this way is referred to as a reservoir, and the outer layer is referred to as the readout layer. The modern understanding of reservoir computing emphasizes the fact that a Turing universal expressive power can be achieved in the context of time-series data processing, if the reservoir exhibits separation property on the state of inputs. This separation property is rather generic, it is not limiting in the engineering sense, and is often realized by complex systems at the edge of chaos. [19]

Though the foundational ideas behind reservoir computing have been around for quite some time, this is still an aggressively developing field which is gaining in momentum. A reservoir computer is essentially a pattern recognition device. It consists of two parts, a dynamical system, referred to as the reservoir, that can be driven by the external signal, and an easily trainable readout layer. The external signal is the input accepted by the system. By assumption, the readout layer is the only part of the device that can be optimized. The external signal drives the system to a specific region of the configuration space, which constitutes the act of computation since the information stored in the external signal is transformed into the information stored in the internal state of the reservoir. The term "reservoir" stands for the reservoir of states. The readout layer is only used to assess in which state the reservoir is. The key claim is that any computation can be realized this way, provided the dynamical system is complex enough. Due to the inherent flexibility and the ease of use the reservoir computing approach is being applied frequently in many applications that require neuromorphic computation. The reservoir computer can be used with a minimum of preparation as an artificial intelligence unit that process external information. From the theoretical point of view, the specific feature that makes reservoir computing popular is the ease of training. Likewise, from the engineering point of view, the readout layer can be a relatively simple structure.

The sensing reservoir concept: The work done in this thesis focuses on other, entirely novel aspect of reservoir computing. We investigate whether it is possible to use reservoir computing to build intelligent sensing substrates that can both collect and analyze information at the same time. The key idea is that the environment one wishes to sense interacts with a dynamical system, which we refer to as a *sensing reservoir*, or a *state weaver*. The sensing reservoir accumulates, or weaves in, the information about the environment into its internal state over time. In such a setup the flow of information is not linear (from the sensor to the artificial intelligence unit), but the sensor and the associated information processing intelligence are the same. More details on these ideas can be found in [16, 18] and in section 3.

1.2 The relation to the RECORD-IT research project

This thesis work has been strongly aligned with the activities in the RECORD-IT research project. The aim of the RECORD-IT project is to develop an intelligent biocompatible device sensitive to the environmental changes in ion concentrations. Thus many of the ideas presented in the thesis have been motivated by a need to understand the RECORD-IT experiments. The systems of interest feature elements that can be described as wet nanoparticle organic memory field-effect transistors (NOMFETs), coated Si nanowires, self-conjugated polymers, or arrays of photocells. These elements are combined to build powerful sensing devices. A natural theoretical paradigm for modelling these structures is a network of environment sensitive electrical components. This defines the context in which this thesis work has been done.

1.3 Aim and Scope

The overarching aim has been to exploit theoretically the possibility of using environment sensitive electrical circuit networks as intelligent sensing substrates. A key feature that is investigated is how the coupling between the elements affects the sensing capacity of the device. The hypothesis is that the existing interactions between the network elements should increase the sensing capacity of the network. The key idea is that the spatial-temporal information about the environment can be accumulated in the state of the reservoir if the system is arranged properly. [18, 16] This could happen if small pieces of information that are scattered over time, and that might be ignored in the standard sensing setup, can be accumulated, amplified, and ultimately stored in the state of the network. When should one use such a device? The use of such devices would be advantageous in situations where embedded information processing is necessary, e.g. in the case of distributed sensing networks. If the information collected by sensors can be pre-processed *in situ*, this would reduce the necessary communication bandwidth, provide real-time analysis options, and accordingly make the whole system much more responsive. Moreover, from the engineering point of view, such sensing solutions could be more flexible and easier to implement, be low-power, or be bio-compatible.

1.4 Content of the Thesis

The content of the thesis follows the structure of the papers I - III which have been either published or submitted. This is augmented with a highlights from the very recent on-going work, a manuscript in preparation, paper IV.

In paper I it is demonstrated theoretically on a very simple classification problem that a single memristor can be used to classify the environment it is exposed to. Only two environmental conditions have been considered, describing a static and a varying environment. First, a suitable drive signal has been identified based on intuitive analysis of the memristor dynamics. Then, a rigorous mathematical optimization problem has been set up, and solved using genetic algorithms. Interestingly, the optimization algorithm produced another drive signal. The two drive signals are different from each other because the intuitive drive was a square-wave while the optimization algorithm was allowed to search in the space of more complex sinusoidal drive signals. Under both drives the memristance is driven to two different regions of the one-dimensional state space (under the influence of the two environmental conditions). The environment can be easily inferred by monitoring the memristance value, i.e. if it is “high” or “low”. The separation only occurs if there is a synchronization between the drive and the environmental signals. To quantify the magnitude of the separation, a quality of sensing index was introduced: The ability to sense depends critically on the synchronization between the drive and environmental conditions. If this synchronization is not maintained the quality of sensing deteriorates.

In paper II, the cooperative behavior between memristor components has been explored achieving an additional functionality. In particular, we investigated how the interplay between various network features influences the sensing capacity of the device. The features of interest included the number of network elements, their connectivity pattern, and the complexity of the individual element. The big questions which have been addressed are as follows:

- Is it possible to quantify in a rigorous mathematical sense the sensing capacity of the device?

- How much information about the environment can be ultimately stored in the state of the substrate?

- How is the sensing capacity of the device depends on the number of the environment sensitive elements in the network?

- Which topological features of the network strongly influence the sensing capacity?

In paper III, a new method has been introduced for the efficient simulation of electronic circuits which contain Constant Phase Elements (CPEs). Finally, this paper suggests an algorithm for simulating circuits with CPEs based on the Modified Nodal Analysis (MNA). The algorithmic complexity of the suggested algorithm is linear with the total time of the transient simulation. This algorithm is compared to a simple method found in the literature: the consideration of resistance-capacitor circuits with equivalent impedance to the CPEs. The comparison has been done both in terms of the accuracy and the algorithmic complexity.

Paper IV is a typical device modelling paper. A simple dynamical model of the Organic Electrochemical Transistor (OECT) element has been suggested and implemented for simulation purposes. The model has been systematically improved through a series of carefully designed numerical experiments. The key outcome of this work is a rigorous simulation algorithm that can be used to predict the response of the OECT changes in time, depending on the voltages that are applied at its pins. A key challenge that has been addressed was to explain intriguing peaks in the experimental data for the drain current as a function of time.

Chapter 2

Mathematical Primitives

In here some key mathematical **primitives** of the reservoir computing approach for sensing are explained. These mathematical primitives feature frequently in the work and manifest themselves in many different forms.

The filter is a mapping that converts an input series of data $q \equiv \{q(t)\}_{t \in I}$ into an output time series data $x \equiv \{x(t)\}_{t \in I}$,

$$q \xrightarrow{\mathcal{F}} x \tag{2.1}$$

where I denotes the index set. The individual values in the series are indexed by the variable t, e.g. as $q(t)$ or $x(t)$. In the following the index set will be omitted when irrelevant for a discussion. The operation of the filter is denoted as

$$x = \mathcal{F}[q] \tag{2.2}$$

and a specific value indexed by t can be selected as $x(t) = \mathcal{F}[x](t)$. Further, the input and output data types do not need to match at all. For example, a filter $\mathcal{F}$ can take a vectorial data as the input, a series of values $\{(q(t), u(t)\}_t$ with $q, u \in \mathbf{R}$ and produce a single valued series $\{x(t)\}_t$ with $x \in \mathbf{R}$ as the output, with $\mathbf{R}$ being the set of real numbers. For a given index t one has $x(t) = \mathcal{F}[q, u](t)$.

The reservoir is a special type of a filter. It is a generic dynamical system, to be denoted by $\mathcal{R}$, that responds to a time-dependent external signal $q(t)$ in a way that the state of the system x at a particular time instance t depends on the way the system has been driven in the past. Using the filter notation introduced above, this behavior is represented as

$$x(t) = \mathcal{R}[q](t) \tag{2.3}$$

In this case the index set I is meant to describe the flow of time. Note that the equation does not read $x(t) = \mathcal{R}(q(t))$, which would imply that the state of the system is an instantaneous function of the drive.

The reservoir should have another important property: one should be able to query its state. Further, the apparatus used to query the state should be something simple, with a low degree of computational complexity, and presumably something that is easy to engineer. **The readout layer**, to be denoted by ψ, analyses the instantaneous state of the device x and produces the output y

$$y(t) = \psi(x(t)) \tag{2.4}$$

Note that the equation does not read $y(t) = \psi[x](t)$ which would imply that ψ represents a filter.

The key claim of reservoir computing: The abstract mathematical formulations introduced above formalize the reservoir computing ideas. Clearly, without stating what the expressive power of this model of computation is, the mathematical primitives are an empty shell without substance. What gives substance to the field is the claim that if the filter $\mathcal{R}$ has some well-defined mathematical properties, notably if it separates the input, then any computation is possible with one and the same reservoir $\mathcal{R}$. Thus for every desired pattern recognition task $\Phi[q](t)$, it is possible to find a related readout layer ψ_Φ such that

$$\Phi[q](t) = \psi_\Phi(\mathcal{R}[q](t)) \tag{2.5}$$

This implies that a single dynamical system has, in principle, infinite computing power, i.e. it can be used to compute anything. At first this might sound as an impossible claim, but in fact this key insight from the liquid state machine model rests on rigorous mathematical foundations of the Stone Weierstrass Approximation theorem. [29].

The sensing goal: Every sensing procedure is done with a certain goal in mind. For example, one might be interested in inferring whether a solution containing ions is static or changes in time. Thus a sensing procedure can be formally described as a pattern recognition task, described by the filter ϕ,

$$\varphi(t) = \Phi[q](t) \tag{2.6}$$

The filter is constructed so that its output, the variable $\varphi(t)$, convey the pattern recognition information. For example, the filter could be constructed to output $\varphi \approx 0$ for static ion concentration, or $\varphi \approx 1$ for a varying one. It is useful to think of Φ as an infinitely "intelligent" neural network that can be trained for any pattern recognition task.

The sensing reservoir is a special dynamical system that can be driven by an external input and interacts with the environment. Thus the state of the system, in the filter notation, can be written as

$$x(t) = \mathcal{R}[u, q](t) \tag{2.7}$$

The sensing performed by the reservoir is represented as

$$y(t) = \mathcal{S}[u, q](t) = \psi(x(t)) \tag{2.8}$$

where $y(t)$ is the variable that conveys the result of the sensing measurement. *The key idea pursued in this thesis is that all the sensing functionality should be done by the reservoir, and not the readout layer.* Thus the readout layer should be something simple to engineer with a low degree of computational complexity. As an example of a reservoir acting as a filter see Fig. 2.1.

Sensor optimization: The goal is to optimize the sensing reservoir so that it mimics the behavior of Φ: Formally, one tries to achieve that $y(t) \approx \varphi(t)$ to the largest extent possible, uniformly over time. This defines a rigorous mathematical optimization problem, where the goal is to find the drive such that

$$u_* = \text{argmin}_u \delta[u, q] \tag{2.9}$$

where $\delta[u, q]$ is a measure of how well the prediction of the sensing reservoir matches the desired classification,

$$\delta[u, q] \equiv ||\mathcal{S}[u, q](t) - \Phi[q](t)||_t \tag{2.10}$$

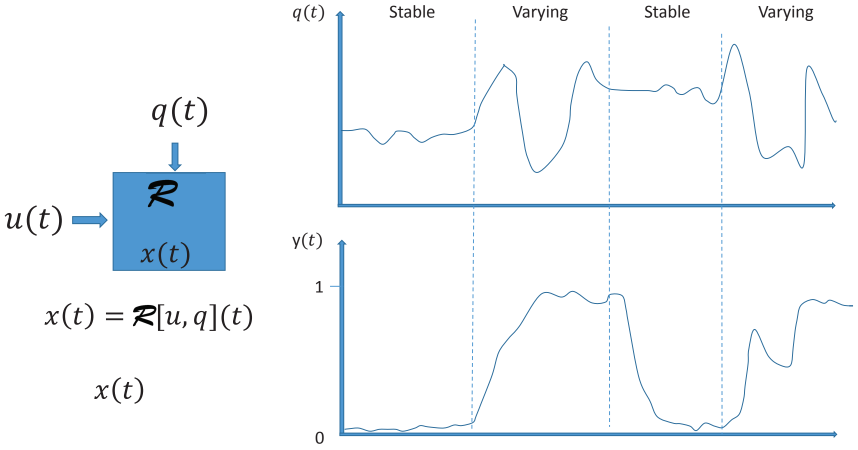

FIGURE 2.1: A hand drawn illustration of the sensing reservoir concept (a modification of a figure from [16]). The reservoir is denoted by $\mathcal{R}$ with the state of the reservoir at time t denoted by $x(t)$. This state depends on the whole history of the drive signal $u(t)$ and the environmental condition signal $q(t)$. To optimize the sensor, a drive signal $u(t)$ has to be found such that the output $y(t)$ is driven to 1 if the environmental condition is a varying one, and to 0 if the environmental condition can be characterized as a stable one.

with $||...||_t$ being a measure of the distance between two filters, the one realized by the reservoir $\mathcal{R}$ and the one by Φ. The subscript on the distance symbol indicates that one should in some sense provide a distance estimate over all times. For example, one could define the distance as

$$||\mathcal{S}[u,q](t) - \Phi[q](t)||_t^2 \equiv \lim_{T\to\infty} \frac{1}{T} \int_0^T dt \, \{\mathcal{S}[u,q](t) - \Phi[q](t)\}^2 \tag{2.11}$$

which is the definition used in the thesis. The distance measure generated this way will be denoted as $\delta^*[u,q]$.

Chapter 3

Part A - Memristor networks as intelligent sensing substrates

Part A of this thesis summarizes the work done on memristor networks in papers I and II. We describe how the mathematical primitives from Chapter 2 have been implemented in the memristor network context, with the focus on advanced sensing application of ionic concentrations. The memristor is one of the simplest electronic elements which behaves as a filter. Further, it is straight forward to couple such elements into a network. The response of such a network, at a given time instance, resembles the one of a pure resistor network. The filter property, i.e. the memory of the past, resides in the resistances which change over time. We studied such networks to streamline the theoretical foundations of the reservoir computing for sensing approach, and critically evaluate the workings of the sensing reservoir idea.

3.1 The sensing reservoir model

An implementation of the sensing reservoir idea using a memristor network is shown in Fig. 3.1. The memristor network is used as a dynamical system, the reservoir $\mathcal{R}$. The memristor network naturally implements the filter primitive since the memristance changes in time depending on the voltages that are applied at the external contacts of the element: $\dot{R} = f(V_1, V_2)$, where here and in the following the dot over a symbol defines a time derivative. The law that describes the rate of the memristance change is shown in Fig. 3.2. Thus $x(t)$ depends on the whole past of the drive signal u and environmental condition q, i.e. the time-series of u, q and not their instantaneous values. In the case of the memristor network the most natural way to realize the external drive is by applying external voltages.

The memristor model used in this thesis has been suggested in [24]. The memristance R changes in time depending on the applied voltage across the element ΔV, and other device parameters α_M, β_M, V_{thr}, R_{min} and R_{max},

$$\dot{R}(t) = f_M[\Delta V(t), \alpha_M, \beta_M, V_thr]\ \theta[R_M(t) - R_{M,min}]\ \theta[R_{M,max} - R_M(t)] \tag{3.1}$$

where t denotes time, $\theta(x)$ is the step function $[\theta(x > 0) = 1, \theta(x \leq 0) = 0]$. Parameters α_M and β_M control how fast the memristance changes, V_{thr} is the threshold voltage, R_{min} and R_{max} denote the minimum and the maximum values of the resistance; the resistance of a memristor cannot be lower than R_{min} or higher than R_{max}. A typical behavior of f_M is illustrated in Fig. 3.2.

The environment model: By assumption, the network is affected by the environmental conditions of the ionic solutions surrounding it. The key challenge is to assume a suitable model for the environment-reservoir interaction. There are numerous options, and special care has been given to choosing an appropriate model.

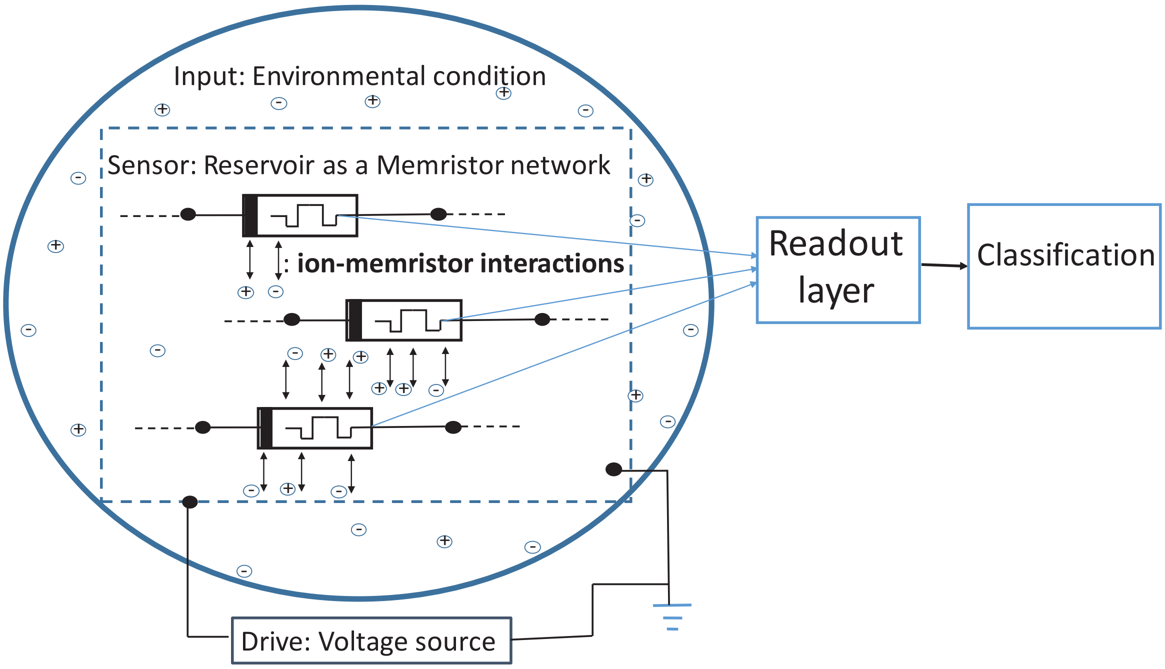
Input: Environmental condition
Sensor: Reservoir as a Memristor network
: ion-memristor interactions
Readout layer
Classification
Drive: Voltage source

FIGURE 3.1: The memristor network is driven by a drive signal and is affected by the environmental conditions of ionic solutions. The readout layer receives the instantaneous values of the network state and contributes to the classification of the environmental condition.

In paper I, based on a careful literature study, it has been argued that it is reasonable to assume that the rate of the memristance change should depend on the ion-concentration. Thus, for simulation purposes, it has been assumed that the parameter β_M is environment sensitive. Assuming that the variations of the environmental signal are small, one can use the standard working point model used in electronics: $\beta_M = a + bq(t)$. This behavior is illustrated in Fig. 3.2. The slope of graph changes depending on the environmental signal.

The state of the whole memristor network x at time t is described as an ordered list of resistances for each time instance

$$x(t) \equiv (R_1(t), R_2(t), \cdots, R_{N_R}(t)) \tag{3.2}$$

where N_R is the dimensionality of the state and denotes the number of memristors in a network. The variable $x(t)$ denotes the trajectory in the state space. This is illustrated in Fig. 3.3. Further, the figure illustrates the operation of the readout layer. The readout layer is assumed to be able to access the values of the individual resistances, which provides the output of the computation

$$y(t) = \psi(x_1(t), x_2(t), \cdots, x_N(t)) \tag{3.3}$$

Now the meaning of the equation $x(t) = \mathcal{R}[u, q](t)$ should be transparent: the network is driven by external voltages, and resistances change according to the specified law. The state depends on the whole history of the drive u and the environmental condition q. The state can be given as input to a readout layer which infers the environmental condition. The simplest form of a readout layer is given, being a weighted linear sum of the network memristances.

Supervised learning and fitness function: How does one optimize the device

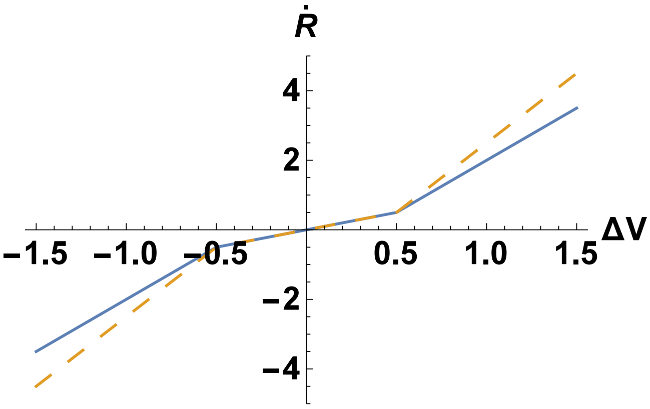

FIGURE 3.2: The rate of memristance change $\dot{R} \equiv dR/dt$ depicted as a function of the voltage drop across the memristor element ΔV. The memristance change is depicted for two cases. All parameters are the same for both cases except for β_M being larger for the model plotted with the solid line than the model plotted with the dashed line.

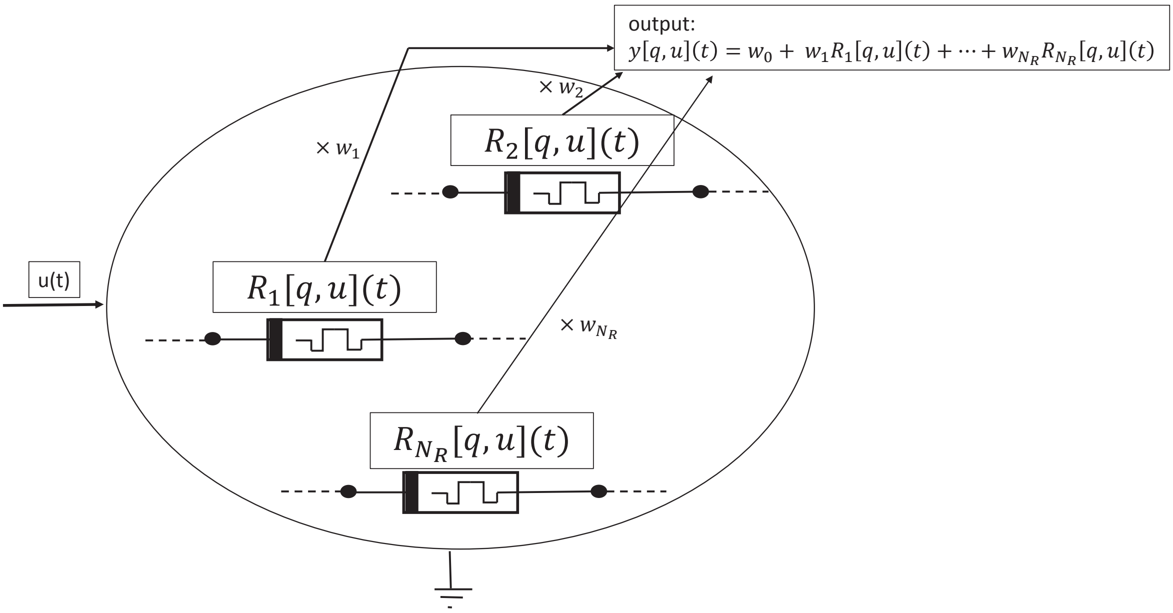

FIGURE 3.3: The state of the memristor network is a vector of all the memristances included in the network. The network is driven by a signal u. The output $y(t)$ is as a linear combination of the memristance values.

then for a particular pattern recognition task? One of the recurring challenges in this thesis is to, for a given network, find a drive signal u_* that maximizes the sensing capacity $SC[u,q]$:

$$u_* = \text{argmax}_u SC[u,q] \tag{3.4}$$

It is hard to define the sensing capacity rigorously. *We posit that the sensing capacity should be related to the state space separability.* In some sense, the sensing capacity should be a function of the differences $\delta[u,q]$. In machine learning one measures the total prediction error,

$$\epsilon = \sum_q \delta[u,q] \tag{3.5}$$

and a viable definition of the sensing capacity measure would be

$$SC[u,q] \sim \frac{1}{\epsilon} \tag{3.6}$$

indicating that a small prediction error should be associated with a large fitness. However, we have often used a slightly different estimate

$$SC[u,q] \sim \sum_q \frac{1}{\delta[u,q]} \tag{3.7}$$

Herein, the sensing capacity $SC[u,q]$ has been introduced to be used as a common metric in both papers I and II.

Papers I and II critically assess which dynamical features of the memristor network could possibly control the sensing capacity. The key insight from these studies is that the trajectory separation in the phase space controls the sensing capacity. Mathematically, this can be formalized as the following requirement. Let $q_1, q_2, \cdots, q_E$ denote the set of distinct environmental conditions we wish to classify. If the drive signal is found such as the state of the memristor network (or any reservoir or interest) occupies different regions $\Omega_1, \Omega_2, \cdots, \Omega_E$, when exposed to the conditions $q_1, q_2, \cdots, q_E$ then a classification is possible. In particular, if the separation of the trajectories is strong, the classification could be achieved with relatively simple readout layer, e.g. a linear classifier might be sufficient.

3.1.1 Sensing with one-memristor network

The simplest possible network, a single-memristor network, has been trained to handle a classification problem with two environments. Since the emphasis is on testing the overall workings of the method, a relatively simply classification problem has been chosen. The goal is to distinguish between two different environments, a stable and a varying one. These were represented by a relatively simple signal pair denoted by q_1 and q_2 and shown Fig. 3.4. The figure has been taken from paper I. Normally, in the supervised learning approach, a class is represented by a group of similar signals, but we have considered only one signal per class. In such a way it is possible to have an intuitive understanding how the optimal drive should look like.

As discussed in chapter 2, this classification problem can be represented as an optimization problem where the goal is to minimize the distances between the classification performed by the system and the desired classification. The goal is to train

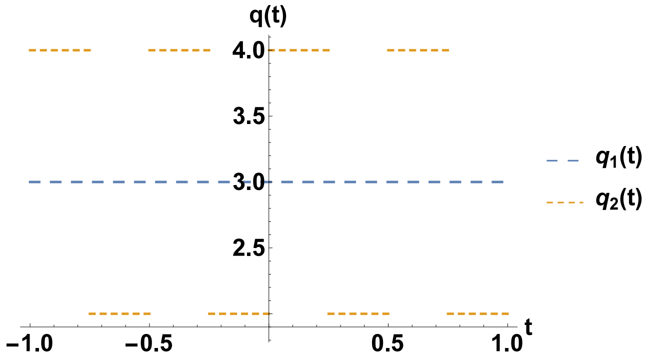

FIGURE 3.4: Figure taken from paper I. The environmental conditions which are denoted as q_1 and q_2.

the system, i.e. to find the drive u_*, so that the sensing is done as

$$S[u_*, q_1] \approx \Phi[q_1] \tag{3.8}$$

$$S[u_*, q_2] \approx \Phi[q_2] \tag{3.9}$$

where the pattern recognition problem that needs to be learned by the memristor is given by $\Phi[q_1](t) = 0$ and $\Phi[q_2](t) = 1$.

The above optimization problem has been solved using genetic algorithms, where we used the following fitness function as the optimization goal:

$$SC_1[u] = \sum_q \frac{1}{\delta^*[u,q]} \tag{3.10}$$

The sum is over environmental conditions. Effectively, the fitness function above, describes the fact that the goal is to find the smallest possible distances for every environmental condition. If both distances are small, SC_1 is large.

A drive signal was found firstly by direct intuitive reasoning. The dynamics of the memristor element were analyzed and we had a good understanding of how to choose a drive signal to lead the memristance to a specific direction under one specific environmental conditions. Therefore, for the two environmental conditions we identified a drive signal for the memristance to occupy two different regions under the two environmental conditions respectively. This intuitive solution was found to be in a good agreement with the drive obtained by running the genetic algorithm. The output for those two drive signals and each of the environmental conditions q_1 and q_2 is shown in Figs. 3.8 and 3.9. Interestingly, under the optimized drive the output is driven faster to the target values $\Phi[q](t)$ but there is a larger variance around $\Phi[q_1]$ and $\Phi[q_2]$.

The figures illustrate nicely the phase space separability idea. The output has been driven to different regions for the two environmental conditions. However, since the drive signal has been optimized only on two environmental conditions, then, it would be expected that this drive signal cannot be used generally to classify stable and varying environmental signals. This expectation has been confirmed by numerically experiments.

3.1.2 Towards collaborative sensing

The key result of paper I is that a one-memristor network can be used for classifying two environment time-series q_1 and q_2. This indicates that a many-memristor network, with many different memristors, could be used to separate between different features of the environmental conditions, and perform a generic classification. This has been investigated in paper II where the classification task is the same as in paper I, i.e. the network should distinguish between a varying or a stable environment. However, to test how well the system generalizes, a larger number of environmental conditions is considered. Every environment q_i is represented as group of signals $q_i \equiv \{q_i^a; a \in E_i\}$ where E_i is the index set that describes the environment q_i.

The simulations were done to investigate a specific idea. One-memristor of the network could contribute to the separation based on different features between a constant environment q_1^a and a varying q_2^a and another memristor could contribute on the separation based on other different features between a constant environment q_1^b and a varying environment q_2^b. For this to be done, all the memristor elements ought to be connected so that the separation of different features can be distributed among the network elements.

Assuming that a collaboration between memristor elements can be exploited as advocated, how should they be connected into a network for the best possible effect? In paper II, we have attempted to answer this very broad question. We have investigated how the number of network elements, their connectivity pattern and the complexity of the elements affect the sensing capacity of the network. Thus the big questions of paper II are as follows:

- How can one quantify the sensing capacity of the device without considering the readout layer? An equivalent question is, how much information about the environment can be stored in the state of the substrate?
- Is the sensing capacity necessarily favored by an increased number of the environment sensitive elements or is there an upper bound of the sensing capacity that cannot not be exceeded?
- Which connectivity pattern are favorable for a larger sensing capacity? In particular, can delay feedback mechanisms be used with an advantage?

The time series of the environmental conditions (the training data) are shown in Fig. 3.5. These conditions will be called the training data since the drive signal has been optimized for these conditions. The training data are labeled with their corresponding class and therefore finding the optimum drive signal is a supervised learning task. In order to test whether the optimum network can be generally used for classifying stable and varying environmental conditions, we created a bigger labeled dataset, the testing data. The testing data consists of thousand randomly created conditions accompanied with the label of their corresponding class.

In paper II, we have not considered a readout layer because we did not want the readout layer to affect the optimization process, or influence the conclusions. If the sensing capacity of the network is relatively large then a simple readout layer would be sufficient for implementing the classification task. An attempt has been made to focus exclusively on the sensing capacity of the reservoir per se. There is a danger that the readout layer actively participates in sensing, and it is hard to control its computational complexity. In papers I and II, the sensing capacity of the network was based on measuring the interclass separability, i.e. when the network is driven

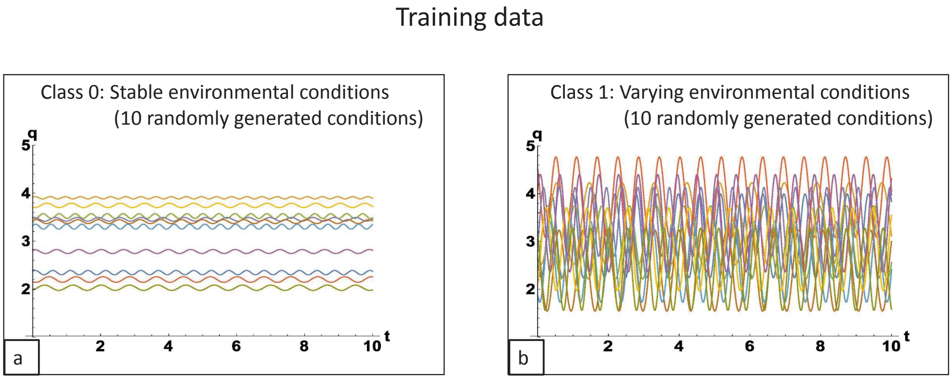

FIGURE 3.5: Figure taken from paper II. The training data for the two different classes. a) The training data for Class 0 b) The training data for class 1.

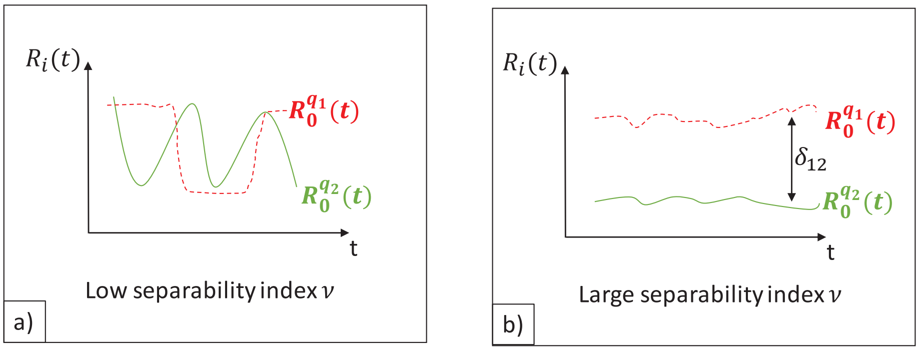

FIGURE 3.6: a) An example of low separability index ν. When the memristance R_0 is driven on average to similar regions under both environmental conditions q_1 and q_2, then this contributes to decreasing the separability index ν. b)When the memristance R_0 is driven on average to different regions then this contributes to increasing ν.

by different environmental conditions, then the state should be driven to different regions. In paper I, due to the fact that the state is one-dimensional, the intraclass separability is a consequence of interclass separability.

In paper II we developed a measure which is indicative of the state separation, the separability index ν. The concept behind the definition of the index ν is illustrated in Fig. 3.6. In Fig. 3.6a, the memristance R_0 is driven on average to similar regions under different environmental conditions and that would make the separability index lower. In Fig. 3.6b, the memristance R_0 is driven on average to two different regions under two different environmental conditions, leading to a larger separability index.

Mathematically, the ideas discussed above were implemented as follows. The mean value of the m^{th} memristance of a network under the environmental condition q_i^a is given as:

$$\bar{R}_m[u, q_i^a] = \frac{1}{T}\int_0^T dt R_m[u, q_j^i](t) \tag{3.11}$$

For a given network with N_R memristors and under a drive signal u, the distance between two environmental conditions belonging to classes i and j, q_i^a and q_j^b, is

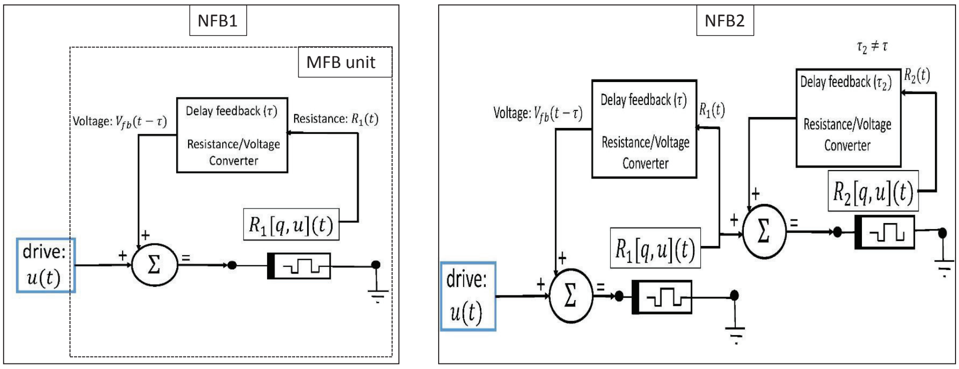

FIGURE 3.7: Figure taken from paper II. In network $NFB1$, the delayed signal is added to the provided drive. In the network $NFB2$ an MFB unit is added. The memristance signal R_1 is converted to voltage and is used to drive the added MFB unit.

given as:

$$d^{i,a}_{j,b} = \sqrt{\frac{1}{N_R}\sum_{m=1}^{N_R}\left(\bar{R}_m[u,q^a_i] - \bar{R}_m[u,q^b_j]\right)^2} \tag{3.12}$$

Totally, for a given network, a drive signal u and a set of training data with classes $c_1, c_2, \cdots, c_k$, the index ν is calculated as the geometric mean of all the possible N_D distances $d^{i,a}_{j,b}$

The structures of the memristor networks were investigated in two ways. Firstly, the complexity of the network topology was increased by adding memristors in parallel and in series. Six networks were considered: $N1$, $N2$, $\cdots$, $N6$ with the number of memristors given by $N_R = 1$, $N_R = 2$, $\cdots$, $N_R = 6$ respectively. Secondly, the complexity of the elements was increased by introducing the MFB element: a memristor with a time-delay feedback loops. We used time-delay feedback loops because the system is expected to gain additional memory properties. The MFB unit at the time t keeps track of the memristance of a previous time with a delay τ, $R(t-\tau)$, and converts it to a voltage signal with a linear mapping. The converted voltage is added to the voltage signal across another memristor element. An example is given in Fig. 3.7 which shows two networks the $NFB1$ and the $NFB2$ with 1 and 2 MFB units respectively. In the network $NFB2$ an MFB unit is added to the network $NFB1$: The memristance signal of $NFB1$ is converted to voltage and is used to drive the added MFB unit. Similarly, the network $NFB3$ was considered by adding an MFB unit to $NFB2$, the network $NFB4$ by adding an MFB unit to $NFB3$. In a similar way, $NFB5$ and $NFB6$ networks were considered. For the networks $NFB2$, $NFB3$, $\cdots$, $NFB6$, the time delays of the MFB units were also considered as free parameters to be optimized.

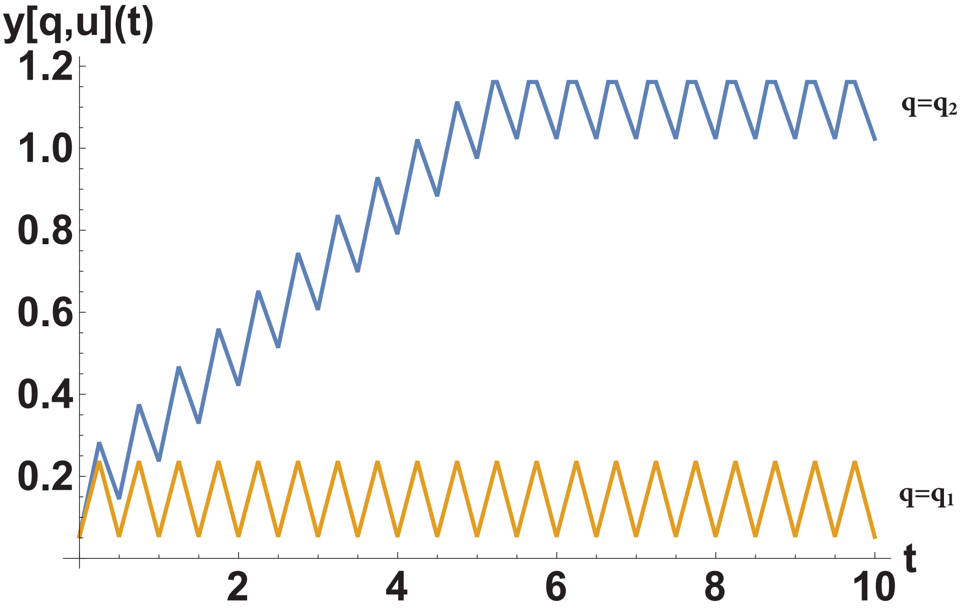

FIGURE 3.8: Figure taken from paper I. The simulated outputs of the one-memristor network under the intuitively found drive signal and the environmental conditions q_1, q_2.

3.2 Results

3.2.1 A simple classification problem with a one-memristor network

The network: the single-memristor network with one node grounded and other node serves as voltage imput. The classification task: two environmental conditions q_1 and q_2 and an one-memristor network.

State space separation was achieved by identifying a drive signal based on intuitive reasoning by understanding the memristor dynamics. The outputs under the intuitive drive signal and the environmental conditions q_1 and q_2 is shown in Fig. 3.8.

The intuitive reasoning was based on the fact that a drive signal can be found such as the memristance increases and decreases with the same rate under the stable environment q_1 resulting in stable memristance around the initial value. Additionally, for the same drive signal, under the environmental condition q_2, the memristance increased with a larger rate than decreased. This resulted in a constantly increasing memristance. Therefore, the memristance occupied different regions under the conditions q_1 and q_2.

For the drive signal u_* found by using the genetic algorithm optimization, the generated outputs are shown in Fig. 3.9. Under the condition q_1 and the drive signal u_*, the dynamics were such so as the memristance decreases on average. However, under the condition q_2 and the drive signal u_*, the memristance increased on average. By comparing the Figs. 3.8 and 3.9 one notices that the optimization algorithm found a drive signal for a faster response, i.e. the output approximated the $\Phi[q_1]$ and $\Phi[q_2]$ faster than when finding the drive signal intuitively. However, there was a larger variation of the outputs around the targets $\Phi[q_1]$ and $\Phi[q_2]$ when using the optimized drive rather than when using the intuitively found drive.

An important result was the following: We implemented several numerical simulations and found that when increasing the amplitude of the drive signal, there was

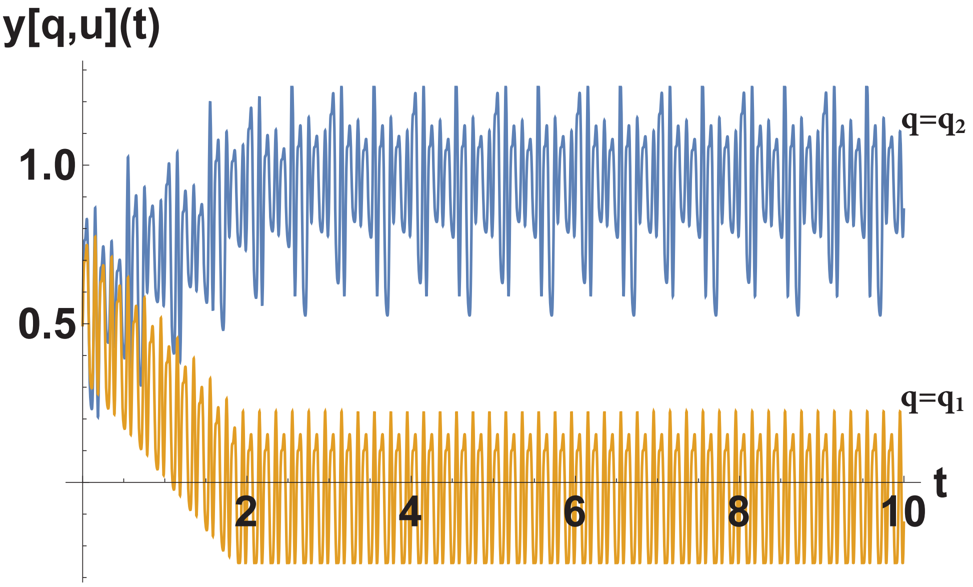

FIGURE 3.9: Figure taken from paper I. The simulated outputs of the one-memristor network under the optimum drive signal (from the genetic algorithm) and the environmental conditions q_1 and q_2.

a faster response and a larger variation around the target $\Phi[q]$. This indicates that there is a trade-off between time response and variation around the target $\Phi[q]$.

3.2.2 Network structures for environment classification

In this section, the optimum separability index ν is shown for the different memristor networks studied. The index ν was maximized by training the memristor networks with the training data. Additionally, to evaluate the performance of the optimized networks, we measured the separability index on the test data. The index ν for both cases is shown in Fig. 3.10, when the networks were constructed by adding in parallel and in series memristors, and in Fig. 3.11, when the *MFB* units were used.

In Fig. 3.10 it is shown that the separability index did not increase when adding memristor elements in parallel and in series. With such network structures there is no adequate collaboration between the memristor elements.

In Fig. 3.11 it was shown that the increase of the elements complexity favored a larger separability index. More specifically, in this figure, as the dimension of the state increases from ($N_R = 1$ for $N1$) towards ($N_R = 4$ for $NFB4$), then, the separability index on both training and test increased. This means that elements collaborate when added into the network. Additionally, the index ν did not improve considerably for both the training and the test data when adding *MFB* units to $NFB4$. Therefore, $NFB4$ can be considered as the network with the minimum amount of resources for achieving the largest index ν. One can also notice in Fig. 3.11 that the index ν on the testing data is favored by an increased number of *MFB* units. Especially, the distance between the index ν on training and testing data tends to be very small when considering the network $NFB6$. This happens because when the dimensionality of the state increases, then, there are more chances for the state to occupy different regions for different environmental conditions.

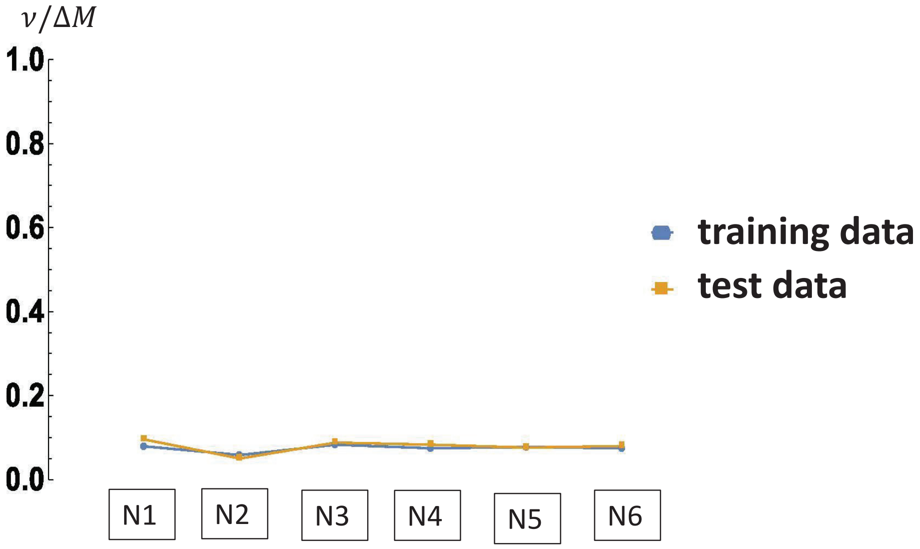

FIGURE 3.10: Figure taken from paper II. The optimum separability index when considering the training data and the measured separability index on the test data for the different network structures $N1$, $N2$, $N3$, $N4$, $N5$ and $N6$.

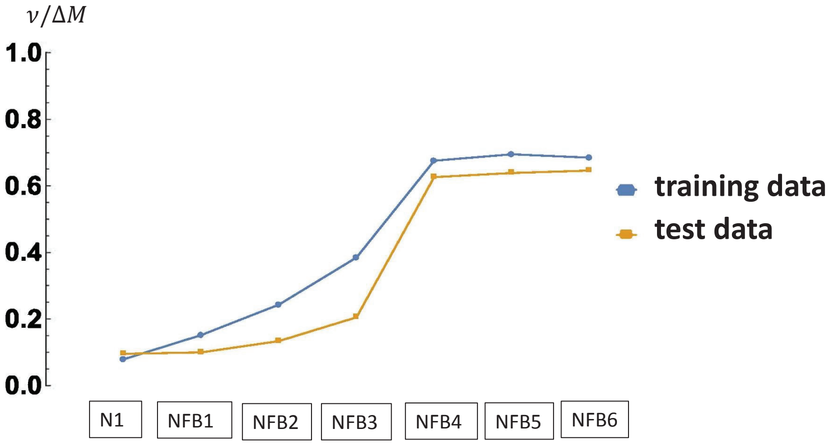

FIGURE 3.11: Figure taken from paper II. The optimum separability index when considering the training data and the measured separability index on the test data for the different network structures $N1$, $NFB1$, $NFB2$, $NFB3$, $NFB4$, $NFB5$ and $NFB6$.

3.3 Discussions

In this part of the thesis, the possibility of using networks of environment sensitive elements for advanced sensing applications has been investigated theoretically, in the context of the SWEET sensing setup. Memristors were considered as an example of environment sensitive elements where a relatively simple model of the interaction between the environment and the memristor has been assumed.

There are important lessons to be learned from the numerical work carried out in paper I. These can be used to speculate on the behavior of larger networks with larger training datasets. *The synchronization between the drive signal and the environmental conditions is important.* A natural question that naturally suggests itself is: Is it still possible to identify a distinct drive signal which synchronizes with a larger dataset of training data? Additionally, the strength of the drive signal was found to regulate the trade-off between the response time and the error. Here it would be very interesting to check whether this finding is true for the larger and more complex networks considered in paper II. If this is not true, then, it might be that this finding is valid only for simple networks of memristors (i.e. connections in parallel and series).

The main question addressed in paper II was: How to design a network to achieve a collaboration between environment sensitive elements? The sensing capacity of the device was found to increase if heterogeneous time delay feedback mechanisms were used (with different time delays). This is an important finding. *It shows that heterogeneity is significant for the performance of the device.* However, we found that after the usage of a specific number of memristors with time delay feedback mechanisms connected to the network, there was no further increase in the sensing capacity, the sensing capacity reached a plateau.

It is also interesting to notice that the memristor networks in paper II constitute an implementation of an artificial neural network primitive: an important part of the integrate and fire concept, a highly non-linear function that takes the weighted sum of the inputs to a neuron and produces an output in a given range. By using delay feedbacks, the memristor elements achieve a similar behavior as neurons: except for adding non-linearity to the network, their memristance can be driven in relatively short times to two states of operation, either their maximum (firing state) or minimum value (non-firing state). However, in contrary to artificial neural networks, we noticed that by adding more and more complex elements to the network, the separability index on the test data did not worsen, i.e. overfitting was not a problem. We argue that this happened because the dimensionality of the state space increased and there were few chances for the state to occupy similar regions of the state under different environmental conditions.

Possibly, if we implemented numerical experiments with larger networks, then, we could see effects of overfitting. This poses a question: for a given classification task, who performs better, the neural network or a memristor network? For choosing whether to use complex memristor networks (with delay feedback units) as suggested in paper II, or software based solution (e.g. artificial neural networks), decisive factors would be the cost (in energy or hardware) to use delay feedback mechanisms. Additionally, memristor networks and artificial neural networks could be used together in hybrid solutions, where the output of memristor networks, which perform hardware computations, could be used as input in software based artificial neural networks.

In this thesis, we constructed a simple way to measure the state separation with a simple number and introduced the separability index ν. Clearly, there could be other ways for estimating the sensing capacity by accounting the temporal behavior of the signals representing the state. For example, a way would be to use a generic measure of the mutual information concept to quantify how much information about the environment is stored in the state of the system. Another would be sensing by identifying specific patterns at the state of the network.

The concepts developed in this part are intended to be used in the RECORD-IT project. Already, an experimentalist group involved in the RECORD-IT project have implemented the ideas of heterogeneity in time delay feedbacks. They have used in their system two different time delay feedbacks and they have noticed sensitivity of the state to the chemical concentrations around their network of sensors.

Chapter 4

Part B - Exploiting algorithms for efficient transient simulations

Part B of this thesis summarizes the work done in papers III and IV. There has been a need to simulate accurately and efficiently several SWEET device prototypes. We developed a generic electronic circuit simulator for simulating the transient behavior of electronic circuits with environment sensitive elements. The ultimate goal was to use the simulator as an optimization tool to identify optimal network designs (e.g. the drive signal and the network parameters). The simulator has been implemented as an integral part of an automatic genetic algorithm optimization procedure where many network designs are tested at random until the one with a desired functionality is found. Since such a numerical optimization process involves an extremely large number of simulations, then, each simulation should be executed in a relatively short time, to make such an optimization approach feasible.

While many electronic component models have been imported to the simulator from the literature, e.g. the widely used models for resistor, memristor, capacitor, inductor, some components were modelled from scratch. In particular, a lot of effort has been put into developing efficient simulation algorithms for the constant phase element (CPE) and the organic electrochemical transistor (OECT). CPEs are models of electronic circuits that are used in equivalent electronic circuits of elements where ionic diffusion is involved. The OECTs are special purpose devices used for analyzing ionic solutions. There is a genuine lack of accurate and algorithmically efficient models for simulating OECT and CPE transient behaviors. Further, for the OECT element of interest, some models are available, but because of their special-purpose nature, they have severe limitations and could not be used directly.

In section 4.1, a new method for simulating the transient response of CPEs is given. In section 4.2, a generic theory of OECT transistors is given which can be used to develop a model for simulating the transient response of OECTs.

4.1 Transient simulation of electronic circuits with Constant Phase elements

The problem regarding the transient simulation of electrical circuits with CPEs is that a repeated numerical evaluation of a computationally expensive convolution integral is needed, shown in Eq. (4.1). This integral relates the instantaneous voltage drop $V_w(t)$ across the element, with the current that has passed through it $I_w(t')$ with $t' \leq t$. To avoid this problem, various methods have been suggested in the literature.

The standard method is to approximate the CPE element by an equivalent RLC circuit. [26, 30, 31, 4] These circuits are easier to simulate. For example, there are commercial packages available that can be used to simulate them efficiently. However, these methods are only accurate in a short range of frequencies due to a finite (often a relatively small) number of resistors, capacitors or inductors. The accuracy in a wide range of frequencies requires the increase in the number of elements, and this increase implies a larger algorithmic complexity cost. It has been pointed out in [4] that the accurate approximation of RLC circuits in a wide range of frequencies is still an open problem.

Another set of methods focuses on expanding the convolution kernel as an infinite series of special functions. [14, 8] The advantage of these approaches is that if the series converges fast then only few terms in the series expansion need to be kept. However, in general it is hard to know how many terms should be kept.

We have developed a novel method for simulating the transient dynamics of CPEs that is both generic, remarkably efficient, and surprisingly accurate. For simplicity reasons this thesis focuses on a specific type of CPE, the Warburg element. The Warburg element is one type of CPE where the applied voltage difference V_w at time instance t, $V_w(t)$, and the current passing through it I_w at time instance t, $I_w(t)$ have a phase difference equal to 45 degrees. The method for simulating transient response of the Warburg element can be easily extended to CPEs.

The need for developing the new method emerges from the fact that the calculation of the voltage across the CPE, $V_w(t)$, requires the calculation of the following convolution integral [15]:

$$V_w(t) = \frac{A_w(\alpha_w)}{\Gamma(\alpha_w)} \int_0^t (t-u)^{\alpha_w - 1} I_w(u)\, du, \quad \alpha_w \in [0,1) \tag{4.1}$$

with Γ being the usual Gamma function, e.g. $\Gamma(1/2) = \sqrt{\pi}$, and $A_w(\alpha_w)$ is a device dependent constant.

Calculating numerically the convolution integral is reasonable for short time intervals. However, a repeated numerical evaluation of the convolution integral can be very expensive for long times. Problems arise regarding the memory usage and the computation time because the time instances of the current $I_w(t)$ have to be stored for a long time interval $[0, t]$. Additionally, the larger this time interval is, the more computationally expensive the calculation of this integral becomes. [15] For example, assuming a grid of n time points $\{t_0 = 0, t_1, \cdots, t_m, \cdots, t_n\}$, the computational cost of evaluating the convolution integral scales as

$$O(n^2) \sim \sum_{m=0}^{n} O(m) \tag{4.2}$$

where $O(m)$ is the algorithmic cost of evaluating the integral for a fixed time instance t_m. Paper III suggests a generic method for decreasing the algorithmic complexity by one order of magnitude.

4.1.1 Updating the convolution integral

For the Warburg element, $\alpha_w = 1/2$ and the voltage is calculated as the following convolution ('*' denotes the convolution operation):

$$V_w(t) = \frac{A_w}{\sqrt{\pi}} \frac{1}{\sqrt{t}} * I_w[t] = \frac{A_w}{\sqrt{\pi}} \Phi_w(t) \tag{4.3}$$

where $A_w \equiv A_w(\alpha_w = 1/2)$.

The computational cost of any standard quadrature algorithm for the approximation of the convolution integral $\Phi_w(t_n)$ in a discrete grid of n time-points $\{t_0, t_1, \cdots, t_m, \cdots, t_n\}$ scales with the size of the time grid n, where $t_0 = 0$ and the distance between two successive time-points is given as $\Delta t_i = t_i - t_{i-1}$:

$$\Phi_w(t_m) \approx \sum_{j=0}^{m} w_j^m \, I_{w,j} \tag{4.4}$$

where for further convenience we use the notation: $I_{w,j} = I_w(t_j)$ and, the weight w_j^m depends on the time-points of the grid t_j and t_m.

While the computational cost of evaluating Eq. (4.4) at a fixed time instance t_m is $O(m)$, the problem is that a repeated evaluation for many time instances leads to a quadratic cost $O(n^2)$, and here we assume that the time grid of the simulation consists of n time points. To deal with this problem, in paper III, we have developed a method to calculate the convolution integral at time t_m, $\Phi_w(t_m)$, by using the already calculated $\Phi_w(t_m - dt)$. To do that, the convolution integral $\Phi_w(t_m)$ is split in two parts:

$$\Phi_w(t_m) = H(t_m, t_\lambda) + \Psi_0(t_m, t_\lambda) \tag{4.5}$$

where,

$$H(t_m, t_\lambda) = \int_{t_\lambda}^{t_m} \frac{I_w(u)\, du}{\sqrt{t-u}} \tag{4.6}$$

and,

$$\Psi_0(t_m, t_\lambda) = \int_{0}^{t_\lambda} \frac{I_w(u)\, du}{\sqrt{t-u}} \tag{4.7}$$

where $H(t_m, t_\lambda)$ is calculated with high precision and $\Psi_0(t_m, t_\lambda)$ is calculated by a simple update of the previously calculated $\Psi_0(t_{m-1}, t_{\lambda-1})$.

The key idea is shown in Fig. 4.1. The One part, $\Psi_0(t_m, t_\lambda)$, is used to approximate the convolution integral between 0 and t_λ and the other part, $H(t_m, t_\lambda)$, for the approximation between t_λ and t_m. At the next time step t_{m+1}, the convolution integral is written again a sum of the two parts:

$$\Phi_w(t_{m+1}) = H(t_{m+1}, t_{\lambda+1}) + \Psi_0(t_{m+1}, t_{\lambda+1}) \tag{4.8}$$

Notice here that the successive distances $t_m - t_\lambda$ and $t_{m+1} - t_{\lambda+1}$ are approximately equal. Therefore, the calculation of $H(t_m, t_\lambda)$ would require similar algorithmic complexity to the calculation of $H(t_{m+1}, t_{\lambda+1})$. However, the calculation of $\Psi_0(t_{m+1}, t_{\lambda+1})$ requires larger algorithmic complexity than the calculation of $\Psi_0(t_m, t_\lambda)$ because $t_{\lambda+1} > t_\lambda$.

Since the algorithmic complexity of calculating $H(t_m, t_\lambda)$ is similar at every time point t_m, we calculate this part with high precision. However, the algorithmic complexity of calculating $\Psi_0(t_m, t_\lambda)$ increases as t_m increases.

In paper III, we developed a method where Ψ_0 at the next time point, $\Psi_0(t_{m+1}, t_{\lambda+1})$, can be approximated by a simple update of Ψ_0 at the previous time point, $\Psi_0(t_m, t_\lambda)$. For this purpose, we suggested a dynamical system with $N+1$ equations and a closure function. This dynamical system is shown in Eqs. (50) and (51) in paper III. The definition of the closure function is given in Eq. (49) in paper III:

$$r(t_{m+1}, t_\lambda) = \frac{\int_0^{t_\lambda} \frac{I_w(u)\, du}{(t_{m+1}-u)^{\frac{2\cdot N+3}{2}}}}{\int_0^{t_\lambda} \frac{I_w(u)\, du}{(t_{m+1}-u)^{\frac{2\cdot N+1}{2}}}} \tag{4.9}$$

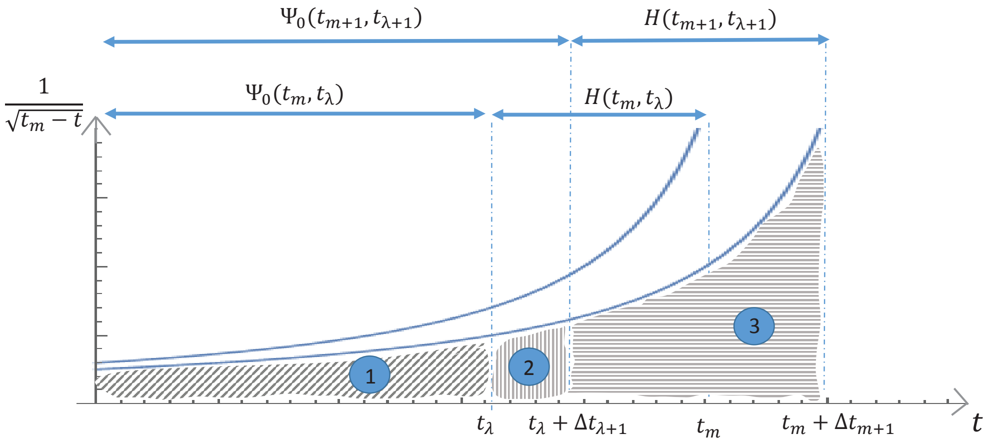
$\Psi_0(t_{m+1}, t_{\lambda+1})$
$H(t_{m+1}, t_{\lambda+1})$
$\frac{1}{\sqrt{t_m - t}}$
$\Psi_0(t_m, t_\lambda)$
$H(t_m, t_\lambda)$
3
2
1
t_λ
$t_\lambda + \Delta t_{\lambda+1}$
t_m
$t_m + \Delta t_{m+1}$
t

FIGURE 4.1: Figure taken from paper III. The concept of calculating the convolution integral at the time-point $t_m + \Delta t_{m+1}$ by using the previous convolution integral at the time-point t_m. The integral $\Phi(t_m)$ ($\Phi(t_m + \Delta t_{m+1})$) is calculated by convolution between the current and the curve reaching the time-point t_m ($t_m + \Delta t_{m+1}$). The integral $\Phi(t_m + \Delta t_{m+1})$ is approximated by three different calculations. Region 1 refers to the tail window calculation by using information from the previous convolution integral. Regions 2 and 3 refer to analytical calculation of the convolution integral by linear interpolating the current $I_w(t)$.

One of the questions of this paper is how to calculate the closure function above. To calculate it analytically, it has been chosen that the current is constant $I_w(u) = I_o$.

Finally, an algorithm is suggested for performing transient simulations of electronic circuits with Warburg elements by using the Modified Nodal Analysis (MNA). The Warburg element is suggested to be used similarly to a Voltage source object with the MNA. The MNA is widely used in electronic circuit simulators for transient simulations and therefore the integration of the developed method with electronic circuit simulators is amenable.

4.1.2 Results

In paper III the designed algorithm was tested on a simple circuit driven a voltage source. Different numerical simulations were performed for two different cases of the voltage source signal: DC and AC signals of different frequencies. The size of the dynamical system was set for all the simulations as $N = 1$.

By performing those numerical simulations, we investigated the effect of the distance $t_m - t_\lambda$ on the algorithmic complexity and the error. We found that there is a trade-off between the error and the algorithmic complexity. By choosing the distance $t_m - t_\lambda$ one can regulate this trade-off. By increasing the distance $t_m - t_\lambda$, then, the error decreases at the cost of a larger execution time of the algorithm.

By comparing our method with a simple RC (resistor-capacitor) circuit, we found that the execution time of our method is similar to a Three-RC circuit (with three capacitors and three resistors). This finding is interesting because one would expect a much larger execution time with our method since our method is heavily dependent on the approximation of $H(t_m, t_\lambda)$ being computed with high precision. However, the execution time with RC methods is relatively large due to the larger number of

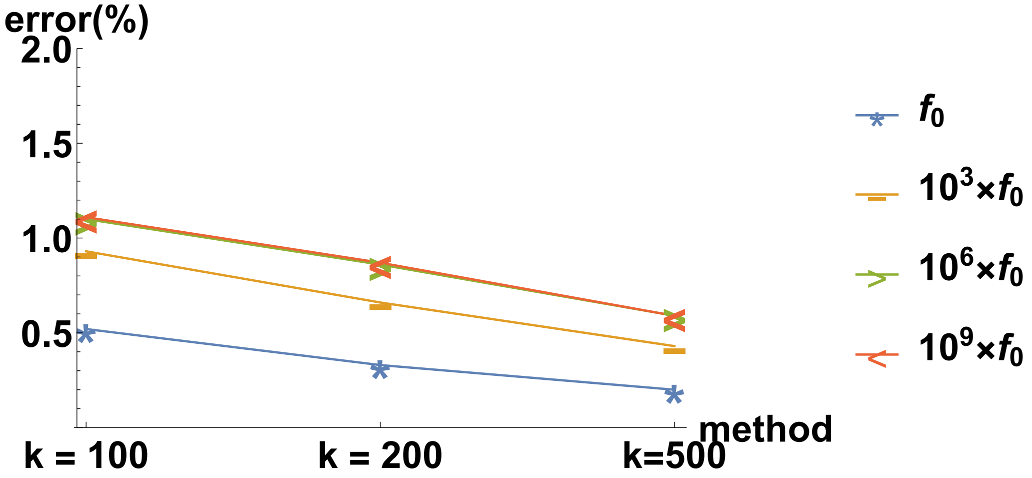

FIGURE 4.2: Figure taken from paper III. The error(%) when the circuit was simulated with an AC voltage source as a sinus signal with amplitude $0.001V$ and four different cases: with period $1/f_0 = 0.628s$, $1/(10^3\ f_0)$, $1/(10^6\ f_0)$ and $1/(10^9\ f_0)$ respectively. The time-step dt was such that 100 time-points were sampled per sinus cycle. The circuit was simulated for five cases: $k = 30$, $k = 100$, $k = 200$, $k = 500$ and $k = \infty$. The parameter k denotes the number of time-points which have been used to calculate $H(t_m, t_\lambda)$ with high precision. The error is calculated as the absolute difference between the ideal case $k = \infty$ and the every other case $k = 30, 100, 200, 500$ divided by the maximum value of the simulation for $k = \infty$. In the time domain, the error is oscillating from 0 to a maximum value. The maximum error is depicted on the vertical axes. The error(%) when $k = 30$ is not shown in this figure for resolution reasons. This error was found as 2.54%, 4.17%, 4.69% and 4.71% at the four frequencies respectively.

nodes in the equivalent circuits (6 nodes were used with the Three-RC circuit and 3 nodes with our method).

A key result is that our method is stable at a large range of frequencies ($1Hz - 1GHz$) as it is shown in Fig. 4.2. This is a great advantage of our method: The RLC circuits have equivalent impedance to constant phase elements in a small range of frequencies while our method is stable in very large range of frequencies. If we wanted to use the "Three-RC circuit" method for achieving a low error in a larger range of frequencies, then, we should design an RC circuit with more components and voltage nodes and the algorithmic complexity would be heavily increased.

4.1.3 Discussions

In paper III, a new method has been designed and tested for performing transient simulations of electrical circuits that contain CPEs. In particular, the problem is that the numerical evaluation of the convolution integral describing the response of the Warburg element is computationally very expensive. By default, the convolution integral has an algorithmic complexity $O(n^2)$ for a grid with n time-points. We identified a way to reduce the algorithmic complexity by one order of magnitude.

One important finding was that the trade-off between error and execution time can be regulated by the distance $t_m - t_\lambda$. It would be interesting to investigate in future how the choice of the total number of equations in the dynamical system affects this trade-off. *This presents an intriguing algorithmic challenge for future work:*

We performed some numerical simulations with $N > 1$ and the error was increased instead of our expectations for a decreased error. Therefore, the question is how is it possible to increase the number of dynamical equations and decrease the error. Such a finding would be of great significance since a better trade-off between error and execution time could be found.

Additionally, regarding the analytical calculation of the closure function, one could argue that it is not reasonable to consider a constant current. A constant current passing through a CPE means that it will charge (or discharge) for the whole time of the simulation which is a rare case in experiments. In the simulations under both AC and DC voltage source signal, the current was not stable. Therefore, the question is if one could use another way to calculate analytically the closure function and achieve less error. *A possibility* is to switch between different update schemes as the simulation progresses.

Finally, the developed methods in paper III can be used to integrate a CPE object in a simulator operating with Modified Nodal Analysis in a large range of frequencies. This has not been achieved before. For example by using RLC methods, e.g. as the one suggested in [26], one should consider a quite large RLC circuit to operate with equivalent impedance to CPEs in the range of frequencies $1Hz - 1GHz$. However, the simulation of quite large circuits requires quite large execution times.

4.2 Transient simulation of electronic circuits with Organic Electrochemical Transistors

Transient simulations of electrical circuits with OECTs can be a useful tool for designing efficient environment sensitive networks for biosensors applications. Such simulations can provide mechanistic understanding of the underlying physical concepts. They can be used to infer circuit parameters by fitting simulations to experimental data, etc.

In the literature, there have been successful theoretical models which unravel the underlying principles of OECTs. [3, 25, 7, 5] However, there are limitations regarding the usage of these models for building simulator primitive that are easily integrated in the electric circuit simulators, e.g. such as SPICE.

For example, Faria et al. [5] have proposed a model for the drain current transient response. Their model is useful for predicting the transient drain current in a range of time when the gate voltage input is known in the whole range before the start of the simulation. However, when connecting OECTs in a network then the gate voltage cannot be known in the whole range. As an example, the gate voltage driving one OECT might be dependent on the chemical concentrations around other OECTs, and therefore this gate voltage cannot be known before the simulation. Sideris *et al* [27] have also suggested a method for simulating the OECT transient. Their method is based on polynomial approximations of the drain current.

In paper IV, we have proposed a theoretical setup for describing the OECT transient response when exposed to arbitrary voltage signals at the electrodes. Our approach is more generic than the one by Sideris *et al*, since it is based on a genuine dynamic ODE paradigm, and allows for more flexible numerical integration techniques.

4.2.1 Equations of motion

In our approach, we generalize the equations of motion previously developed by Bernards and Malliaras [3], and obtain a system of partial differential equations that describe how ionic degrees of freedom are coupled with the electrical degrees of freedom in the material.

The geometry of both the device and the electrolyte are given in Fig. 4.3. The device is a semiconducting substrate with dimensions $\{L, w, y\}$ and is covered by ionic solution above. The device volume is divided in vertical slices with infinitesimal volumes, as shown in Fig. 4.3a. Every infinitesimal volume is described by using the equivalent circuit model, as shown in Fig. 4.3b. The gate voltage $V_g(t)$ is applied on the top of the electrolyte. The voltage on the boundary between the semiconductor and the electrolyte at the position x and time t is denoted as $V_{ch}(x,t)$. $V(x,t)$ denotes the voltage inside the electrolyte. This simple model features a resistance R_e coupled in series to a capacitor C_d. The resistance describes the flow of ions through the slice of the electrolyte above the semiconductor (Fig. 4.3c).

After passing the electrolyte, the ions enter into the semiconductor material. It has been argued that a good model that describes this process is a volume capacitance. The capacitance of the piece of material with volume $v = ywdx$ is given by $C_d = c_d v$ where c_d is the volume capacitance of the device material (Fig. 4.3d).

In every small volume of the device there is an accumulated charge density $Q(x,t)$. In the equivalent circuit, this charge density is the charge of the capacitor C_d. By solving Kirchhoff laws in the equivalent circuits, the following dynamical

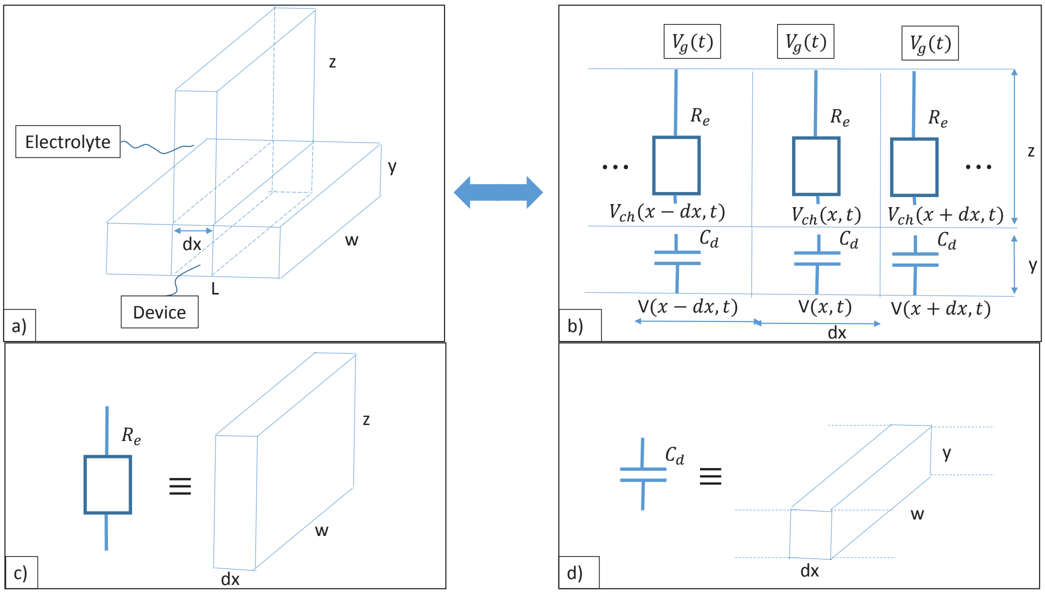

FIGURE 4.3: a) The geometry of the electrolyte and the device is divided in infinitesimal slices with length dx. Each slice contains two parts. The volume $dx\ w\ y$ occupies the region in the OECT material. Above this volume, there is a volume of electrolyte $dx\ w\ z$. b) The equivalent circuits of the device and the electrolyte. The material acts as a volume capacitance where C_d is the capacitance of the device sub-volume. The electrolyte sub-volume has a resistance R_e. Applied voltages are: the gate voltage V_g, $V_{ch}(x,t)$ is the time-dependent voltage at the boundary between electrolyte and device at the position x, and $V(x,t)$ is the local voltage in the device sub-volume. c) The electrolyte is modeled by an equivalent resistance R_e. d) The device is modeled by an equivalent capacitor C_d.

equation is derived:

$$\frac{\partial Q(x,t)}{\partial t} = -\frac{Q(x,t)}{\tau} + \frac{1}{\tau}\, c_d\, y\, w\, [V_g(t) - V(x,t)] \tag{4.10}$$

with the time constant of the equivalent circuit given by $\tau = r_e c_d z y$.

Across the semiconductor material, the Ohm's law relates the current density $J(x,t)$ flowing through the semiconductor device and the voltage $V(x,t)$.

$$J(x,t) = e\mu\rho(x,t)\frac{\partial V(x,t)}{\partial x} \tag{4.11}$$

where $\rho(x,t)$ is the local density of charge carriers, e is the charge of the carrier, and μ is their mobility. The free charge carrier density $\rho(x,t)$ is regulated by the concentration of ions $Q(x,t)$ that are absorbed in the material: an increase in $Q(x,t)$ leads to a decrease in $\rho(x,t)$. An approximate relationship between ρ and Q has been suggested in [3]:

$$\rho(x,t) = \rho_0 \left(1 - \frac{Q(x,t)}{Q_{max}}\right) \tag{4.12}$$

where ρ_0 and Q_{max} are device parameters.

Equations (4.11) and (4.10) have been solved and analyzed by making the assumption that the transient charge in the device $Q(x,t)$ can be approximated by the charge density at the steady state condition $Q_{st}(x,t)$ times a variable T:

$$Q(x,t) \approx T(t) Q_{st}[x, \xi(t)] \tag{4.13}$$

The variable T denotes how far the system is from the steady state condition. If $T = 1$, then the system is at the steady state. Otherwise, the system has less charge ($T < 1$) or more charge ($T > 1$) than in the steady state condition. The externally controlled electrode voltages: gate Voltage $V_g(t)$, drain voltage V_d and source Voltage V_s are collectively denoted as $\xi(t)$. These voltages determine the stationary state charge density profile. If ξ is altered, the charge density profile $Q_{st}(x)$ changes too, and to emphasize this we use $Q_{st}(x, \xi)$

After the straight forward but somewhat tedious algebra, which will not be reproduced here, the solution of the Eq. (4.11) by using the above assumption results in an analytical equation for the drain current I_D:

$$\begin{aligned} I_D(t) &= f_O[T(t), \xi(t)] = \\ &= G\left(1 - \frac{V_g - \frac{V_d}{2}}{V_p}\right) \frac{(V_d\, T)^2}{T\, V_d - \left[V_p\, (1-T)\right]\, Log\left(1 + \frac{T\, V_g}{V_p - T\, V_g}\right)} \end{aligned} \tag{4.14}$$

where G is the conductance of the semiconductor given as $G = e\mu\rho_o W\, \frac{Y}{L}$ and V_p is the pinch-off voltage of the semiconductor.

The analysis of the Eq. (4.10) with the above assumption considering a discrete time grid with time step Δt results in the following update rule for the parameter T:

$$T(t) = \frac{\tau}{\tau + \Delta t}\, \Lambda(t, \Delta t)\, T(t - \Delta t)\; + \; \frac{\Delta t}{\Delta t + \tau}\, \Xi(t) \tag{4.15}$$

where

$$\begin{aligned}\Lambda(t,\Delta t) &\equiv \langle\Lambda(x,t,\Delta t)\rangle_x &(4.16)\\ \Xi(t) &\equiv \langle\Xi(x,t)\rangle_x &(4.17)\end{aligned}$$

with

$$\Lambda(x,t,\Delta t) = \frac{Q_{st}[x,\xi(t-\Delta t)]}{Q_{st}[x,\xi(t)]} = \frac{V_g(t-\Delta t) - V_{st}[x,\xi(t-\Delta t)]}{V_g(t) - V_{st}[x,\xi(t)]} \tag{4.18}$$

and

$$\Xi(x,t) = \frac{V_g(t) - V[x,T(t),\xi(t)]}{V_g(t) - V_{st}[x,\xi(t)]} \tag{4.19}$$

Herein, due to the fact that it is computationally expensive to calculate the integrals in Eqs. (4.16) and (4.17), $\Lambda(t,\Delta t)$ and $\Xi(t)$ are calculated by setting $x = \frac{L}{2}$ in Eqs. (4.18) and (4.19). It is not set $x = 0$ or $x = L$ because there are not transient dynamics at the bounds.

The parameter $\Xi(t)$ indicates how far the transient dynamics is from the steady state conditions. If $V[x,T(t),\xi(t)] = V_{st}[x,\xi(t)]$, then, $T = 1$ and $\Xi(t) = 1$, otherwise, $\Xi(t) \neq 1$ and the update rule has a tendency to move T towards 1.

The parameter $\Lambda(t,\Delta t)$ indicates if the steady state conditions have changed. If $\xi(t-\Delta t) = \xi(t)$, then, $\Lambda(t,\Delta t) = 1$, otherwise, $\Lambda(t,\Delta t) \neq 1$. This means that if $\Lambda(t,\Delta t) \neq 1$ then $\Lambda(t,\Delta t)\ T(t-dt) \neq T(t-dt)$ and the update rule is done from a different point of view.

However, up to here, the contribution of the current coming through the gate, gate current, has not been considered to contribute to the drain current. In previous works, it has been assumed that when the steady state conditions change, then a specific amount of the gate current is driven to the drain and the rest to the source electrode. This gate current is the reason for spikes observed experimentally in the drain current. [6, 5] Therefore, the total current through the drain electrode $I_{D,tot}$ would be calculated by adding a portion of the gate current $\Delta I_g(t)$:

$$I_{D,tot}(t) = I_D(t) + \alpha_1\ \Delta I_g(t) \tag{4.20}$$

where $0 < \alpha_1 < 1$, while $(1-\alpha_1)\ \Delta I_g(t)$ flows into the source electrode. It has been assumed that the gate current is given by

$$\Delta I_g(t) = \frac{Q_{st,tot}(t) - Q_{st,tot}(t-dt)}{dt} \tag{4.21}$$

with the notation $Q_{st,tot}(t) \equiv Q_{st,tot}[\xi(t)]$, where

$$Q_{st,tot}[\xi(t)] \equiv \int_0^L dx Q_{st}[x,\xi(t)] \tag{4.22}$$

Finally, an algorithm is introduced for the transient simulation of OECT models connected to an electronic circuit by using the MNA [11]. The key primitive of the MNA paradigm is the idea of a stamp, as explained in paper III. The stamp of the OECT element is represented by three voltage dependent current sources: one current source at the drain node with the total drain current given by Eq. (4.20), one current source at the gate node with the current given as $-\Delta I_g(t)$ in Eq. (4.21) and one current source at the source node given as $-I_D + \Delta I_g(1-\alpha_1)$.

The four parameters of the model: The OECT model developed in this thesis is

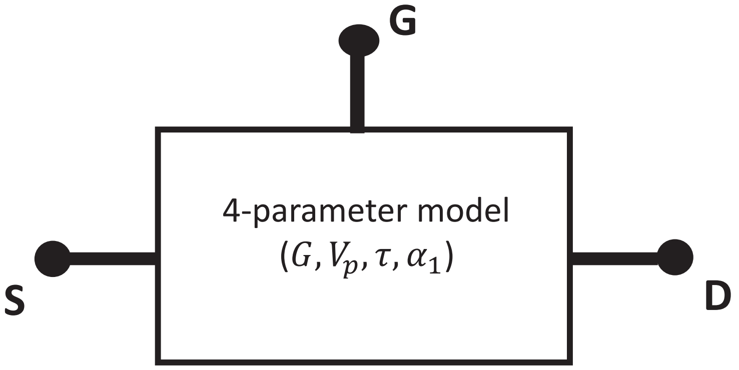

FIGURE 4.4: The OECT four-parameter model suggested in this thesis; S, G and D denote the electrode voltage nodes at the source, gate and drain respectively.

parameterized by the following quantities: the conductance G, the pinch-off voltage V_P, the time constant τ and the parameter α_1 as shown in Fig. 4.4.

4.2.2 Fitting the model to data

An example of fitting the model to experimental data is shown in Fig. 4.5. The experimental drain current was recorded by collaborators in the RECORD-IT project by using the methods given in [23, 22]. The experimental setup: $V_s = 0V$, $V_d = -0.1V$ and the gate voltage was pulsed with a square wave pulse of two levels $0V$ and $0.3V$ with 50% duty cycle. The four parameters of the model have been optimized so as the simulated total drain current $I_{D,tot}$ fits the experimental data.

As one can see in Fig. 4.5, the simulated total drain current agrees with the experimental one. The spike behaviour is correctly reproduced, both the onset and the recovery phases. Additionally, the simulated output relaxes towards the same steady state condition as the one in the experiment. However, there is a behavior that cannot be explained by the current model. In the experiment, the upward spikes are larger than the downward ones. The theoretical model predicts a fully symmetric behavior. For example, when the gate voltage increases from $0V$ to $0.3V$ and when it decreases from $0.3V$ to $0V$, then, the spike currents should have the same absolute value according to the analytical equations derived.

We also noticed in numerical simulations, that when the total current converges towards the steady state solution, then, the variable T converges towards 1. Additionally, when the total current decreased towards the steady state current, the variable T had larger values than one $T > 1$ and when the total current increased, then, it was that $T < 1$. This behavior of T shows that the dynamical system is stable.

4.2.3 Discussions

We developed a theoretical approach for simulating the transients of electrical circuits that contain OECT elements. The dynamical system is stable since the parameter T always converges to $T = 1$ when the external voltages are kept constant. Thus the OECT element always attains the steady state if given enough time, which

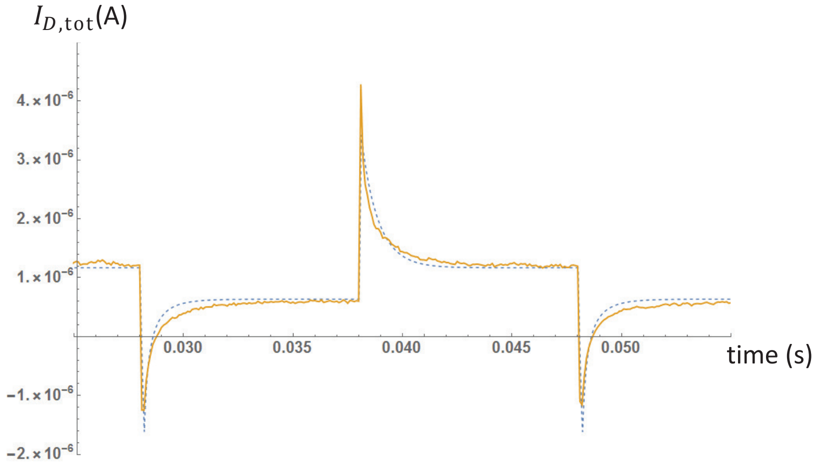

FIGURE 4.5: The simulated and the experimental drain current, with the theoretical parameters fitted to the experimental data. Dashed line: experimental data; The full line: the theoretical prediction.

is an important consistency check. The developed model is relatively easy to integrate into an arbitrary electric circuit simulator, especially the ones that use the MNA method.

The model describes nicely the experimental data, apart from predicting symmetric spike currents. It is possible that the asymmetric spike currents observed in the experiment, are a result of the low sampling frequency of the experimental data, and are in fact symmetric in reality. It would be interesting to test against experimental data collected with a larger sampling frequency. Further, regarding the relaxation to the state condition, we speculate that it might be that there are different time decay constants for the two different phases of increasing and decreasing towards the steady state conditions.

The model that has been developed can be used for reverse engineering of the OECT operation to understand underlying principles of the OECTs. Additionally, the fitting of the model to data can be used to investigate the sensitivity of the four OECT parameters (Fig. 4.4) to environmental conditions. For example, inferring the dependencies of the model parameters on chemical concentrations is a crucial task necessary for designing efficient sensors. These can be extracted by fitting the models to data for different ionic concentrations.

Bibliography

[1] Christopher Bennett et al. "On the inverse pattern recognition problem in the context of the time-series data processing with memristor networks". In: *Advances in Unconventional Computation*. Ed. by Andrew Adamatzky. Vol. Vol 2. Prototypes and algorithms. Springer, 2016.

[2] Erik Bergh and Zoran Konkoli. "On improving the expressive power of chemical computation". In: *Advances in Unconventional Computation*. Ed. by Andrew Adamatzky. Vol. Vol 2. Prototypes and algorithms. Springer, 2016.

[3] D./A. Bernards and G./G. Malliaras. "Steady-State and Transient Behavior of Organic Electrochemical Transistors". In: *Advanced Functional Materials* 17.17 (2007), pp. 3538–3544. ISSN: 1616-3028. DOI: 10.1002/adfm.200601239. URL: http://dx.doi.org/10.1002/adfm.200601239.

[4] O. Enacheanu et al. "Identification of fractional order models for electrical networks". In: *IECON 2006 - 32nd Annual Conference on IEEE Industrial Electronics*. 2006, pp. 5392–5396. DOI: 10.1109/IECON.2006.348151.

[5] Gregório C. Faria, Duc T. Duong, and Alberto Salleo. "On the transient response of organic electrochemical transistors". In: *Organic Electronics* 45 (2017), pp. 215–221. ISSN: 1566-1199. DOI: https://doi.org/10.1016/j.orgel.2017.03.021. URL: http://www.sciencedirect.com/science/article/pii/S1566119917301313.

[6] Jacob T. Friedlein et al. "Microsecond Response in Organic Electrochemical Transistors: Exceeding the Ionic Speed Limit". In: *Advanced Materials* 28.38 (2016), pp. 8398–8404. ISSN: 1521-4095. DOI: 10.1002/adma.201602684. URL: http://dx.doi.org/10.1002/adma.201602684.

[7] Jacob T. Friedlein et al. "Optical Measurements Revealing Nonuniform Hole Mobility in Organic Electrochemical Transistors". In: *Advanced Electronic Materials* 1.11 (2015). 1500189, 1500189–n/a. ISSN: 2199-160X. DOI: 10.1002/aelm.201500189. URL: http://dx.doi.org/10.1002/aelm.201500189.

[8] Francisco Gómez, Juan Rosales, and Manuel Guía. "RLC electrical circuit of non-integer order". In: *Central European Journal of Physics* 11.10 (2013), pp. 1361–1365. ISSN: 1644-3608. DOI: 10.2478/s11534-013-0265-6. URL: https://doi.org/10.2478/s11534-013-0265-6.

[9] M. Hiratsuka, T. Aoki, and T. Higuchi. "Enzyme transistor circuits for reaction-diffusion computing". In: *Ieee Transactions on Circuits and Systems I-Fundamental Theory and Applications* 46.2 (1999), pp. 294–303.

[10] A. Hjelmfelt, E. D. Weinberger, and J. Ross. "Chemical implementation of finite-state machines". In: *Proceedings of the National Academy of Sciences of the United States of America* 89.1 (1992), pp. 383–387.

[11] Chung-Wen HO, ALBERT E. RUEHLI, and PIERCE A. BRENNAN. "The Modified Nodal Approach to Network Analysis". In: *IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS* CAS-22 (1975). DOI: 10.1109/TCS.1975.1084079.

[12] H. Jaeger and H. Haas. “Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication”. In: *Science* 304.5667 (2004), pp. 78–80.

[13] Herbert Jaeger. *The "echo state" approach to analysing and training recurrent neural networks*. Report GDM Report 148 (contains errors). German national research center for information technology, 2001.

[14] Agnieszka Jakubowska and Janusz Walczak. “Analysis of the Transient State in a Series Circuit of the Class $RL_\beta C_\alpha$”. In: *Circuits, Systems, and Signal Processing* 35.6 (2016), pp. 1831–1853. ISSN: 1531-5878. DOI: 10.1007/s00034-016-0270-2. URL: https://doi.org/10.1007/s00034-016-0270-2.

[15] O. Kanoun. *Lecture Notes on Impedance Spectroscopy: Measurement, Modeling and Applications*. CRC Press, 2012. ISBN: 9780203610756. URL: https://books.google.se/books?id=31-dDQAAQBAJ.

[16] Z. Konkoli. “The sweet algorithm: generic theory of using reservoir computing for sensing applications”. In: *International Journal of Parallel, Emergent and Distributed Systems* (2016).

[17] Zoran Konkoli. “On reservoir computing: from mathematical foundations to unconventional applications”. In: *Advances in Unconventional Computation*. Ed. by Andrew Adamatzky. Vol. Vol 1. Theory. Springer, 2016.

[18] Zoran Konkoli. *The state weaving environment-echo tracker (SWEET/RECORD-IT) sensing setup and algorithm (patent pending)*. Patent. 2016.

[19] M. Lukosevicius, H. Jaeger, and B. Schrauwen. “Reservoir Computing Trends”. In: *KI - Könstliche Intelligenz* 26.4 (2012), pp. 365–371.

[20] W. Maass, T. Natschlager, and H. Markram. “Real-time computing without stable states: A new framework for neural computation based on perturbations”. In: *Neural Computation* 14.11 (2002), pp. 2531–2560.

[21] G. E. Moore. “Cramming more components onto integrated circuits (Reprinted from Electronics, pg 114-117, April 19, 1965)”. In: *Proceedings of the Ieee* 86.1 (1998), pp. 82–85.

[22] Sébastien Pecqueur et al. “Cation discrimination in organic electrochemical transistors by dual frequency sensing”. In: *Organic Electronics* 57 (2018), pp. 232–238. ISSN: 1566-1199. DOI: https://doi.org/10.1016/j.orgel.2018.03.020. URL: http://www.sciencedirect.com/science/article/pii/S1566119918301265.

[23] Sébastien Pecqueur et al. “Concentric-Electrode Organic Electrochemical Transistors: Case Study for Selective Hydrazine Sensing”. In: *Sensors* 17.3 (2017), p. 570. ISSN: 1424-8220. URL: http://www.mdpi.com/1424-8220/17/3/570.

[24] Y. V. Pershin and M. Di Ventra. “Experimental demonstration of associative memory with memristive neural networks”. In: *Neural Networks* 23.7 (2010), pp. 881–886.

[25] Jonathan Rivnay et al. “High-performance transistors for bioelectronics through tuning of channel thickness”. In: *Science Advances* 1.4 (2015). DOI: 10.1126/sciadv.1400251. eprint: http://advances.sciencemag.org/content/1/4/e1400251.full.pdf. URL: http://advances.sciencemag.org/content/1/4/e1400251.

[26] N. A. Sekushin. “Equivalent Circuit of Warburg Impedance”. In: *Russian Journal of Electrochemistry* 45.7 (2009).

[27] P. Sideris, S. Siskos, and G. Malliaras. “Verilog-A modeling of Organic Electrochemical Transistors”. In: *2017 6th International Conference on Modern Circuits and Systems Technologies (MOCAST)*. 2017, pp. 1–4. DOI: `10.1109/MOCAST.2017.7937645`.

[28] T. Sienko. *Molecular Computing*. MIT Press, 2003. ISBN: 9780262194877. URL: `https://books.google.se/books?id=RleBQgAACAAJ`.

[29] H. Markram T. Natschläger W. Maass. “The "Liquid Computer": A Novel Strategy for Real-Time Computing on Time Series (Special Issue on Foundations of Information Processing)”. In: *TELEMATIK* 8.1 (2002), pp. 39–43.

[30] Y. Tsividis and J. Milios. “A detailed look at electrical equivalents of uniform electrochemical diffusion using nonuniform resistance-capacitance ladders”. In: *Journal of Electroanalytical Chemistry* 707 (2013).

[31] J. Valsa J. Vlach. “RC models of a constant phase element”. In: *INTERNATIONAL JOURNAL OF CIRCUIT THEORY AND APPLICATIONS* 41.1 (2013), pp. 59–67. ISSN: 0098-9886.