Thesis Proposal:
"MP Grammars, Reactive Systems and Electric Circuits"

Ricardo Henrique Gracini Guiraldelli

November 6, 2013

Abstract

The present work proposes the study of the equivalence between electronic circuits—digital and analog ones—and Metabolic P system. This result provides a new application field for MP system, but also enables the exchange of knowledge between different research field. At last, it may provide the capability to compose hardware that imitates the behaviour of living cell and tissues for a wide range of purposes.

1 Introduction

Science is too broad for just one kind of language. In the past, philosophy was the way of choice for all scientific expression; nowadays, however, the concept of science is so extensive that many "dialects" (i.e., fields of study such as mathematics, physics, biology, etc.) are necessary to formulate questions—as well as answer them—about the diverse patterns observed in nature in a very reasonable and comprehensible way. Each of this fields has its own "set of tools" such as special vocabulary that compress lots of information in concepts, modeling and representations systems that states the object of study in a condensed and clear definition that makes used of common knowledge presented in the field and particular techniques for problem solving based on the accumulated results in the area.

It is not difficult to find, nonetheless, instances of scientific studies that span beyond one particular discipline or has equivalent formulation in others; those, usually, represent general patterns of nature and the rules they derive may be usefully applied in many different kind of situations, with diverse origins. As an illustration, the concept of (ordinary) differential equations

has (one of) its (controversial) origin(s) in physics, with the studies performed by Sir Isaac Newton; today, nevertheless, (ordinary) differential equations extended over a series of different subjects, having applications not only in physics, but also in mathematics, engineering, chemistry, biology, social sciences and many other fields.

But as much as a concept spreads over other domains, it becomes necessary to find lossless procedures to translate it from one domain to another, bringing to the new domain all the intrinsic advantages of the concept in its original representation but excelled by a set of fresh features provided by the new realm. Proceeding with the (ordinary) differential equation example, a mapping of them from the ubiquitous temporal space to the frequency space exists and is needed by a range of applications, specially in electrical circuits analysis. This mapping, called Laplacian transform, not only translate the concept of "variational calculus" to other domains, but also enriches the problem solving methodology transforming the complex equations in a (generally simple) algebraic ones; in this new portrayal, the resolution of those equation is simplified, providing a final solution for the equations in cheaper procedures (in an effort context) than in its original domain. Finally, with the application of an (economical) *inverse mapping*, the result can be ported back to the temporal domain, complete, lossless and accurate.

Following a similar approach to the aforementioned examples, the present work proposes the research for a possible existence of an equivalence between electrical circuits and a particular instance of membrane computing, the metabolic P systems. This equivalence would permit an electrical circuit, let it be a digital or analog one, to be modelled as a metabolic P system under its well defined (discrete) mathematics formalism; also, it would supply the electrical circuits analysis and design with the existing metabolic P systems capability— through the usage of methods such as Log-Gain Stoichiometric Stepwise regression—of inference of rules from a simplified input-output values model of grammar. In the opposite direction, on the other hand, the equivalence would provide the usage of circuit reduction techniques in the model, what could simplify the existing metabolic P system, as well as it would ease the composition of electrical circuits that reproduces the behaviour of already existing metabolic P systems.

Then, the present work is structured to clearly disclose the proposal in the following manner: sections 2 and 3 describe the general theoretical background sustaining the main idea of the thesis proposal; section 4 presents the thesis proposal and the research methodology to be applied. Finally, at section 5, a brief conclusion and trade-offs of the proposed topic are discussed.

2 Metabolic P Systems

The cell, one of the most basics unity of life, is a small dimension but complex and dynamical system limited by a membrane which separates the external world from the internal cellular machinery. This membrane acts as an interface for the cell, selectively collecting molecules from its surroundings and expelling others that were produced inside the cell.

This material exchanging process "feeds" the internal machinery of the cell, activating (or de-activating) a series of interconnected sub-process triggered by the balance between the quantities and types of substances present in cell's cytoplasm—the fluid area inside the cell and bounded by the external membrane. This mechanism, nevertheless, suggests the existence of a computational process inside the cell, where inputted material (external molecules) are transformed, under certain rules, in outputted one (molecules sent beyond the membrane boundaries).

Aware of this mechanism, Gheorge Păun proposed a computational model called P system [23] based on this membrane interaction in cell systems but, at the same time, mathematically formal and consistent, using the concepts of multiset and rewriting systems for the construction of its formal framework.

The Păun's work, since its beginning in 1998 and first publication in 2000 [22], has been expanded by the scientific community in such a way that, nowadays, it has separated from the formal languages fields and is now considered the precursor of the *membrane computing* area, comprising a series of computation models based on the aforementioned cellular mechanism.

Although the membrane computing presented lots of powerful computational models—even with super Turing computational capabilities [4]—and had as one of its goal to bring the mathematical formalism to the biological research, it lacked an easy and direct way to model metabolic and intra-cellular interaction problems commonly arisen in studies supported by bioinformatics, making it difficult to model many real-world studies under the proposed framework.

Attentive to these necessities, Vincenzo Manca proposed a new membrane computing computational model based on the existing Păun's P system which is named Metabolic P system or, for short, MP system [15, 14].

The primary goal of the MP system is to deterministically model metabolic processes, serving as a powerful (discrete) mathematical tool for expressing and supporting biological studies in the cell magnifying level; also, it meant to be a computational "intermediate language" for easy simulation of the formalizes models; at last, it should use promptly understandable notation for potential users unfamiliar with the theoretical computer science jargon commonly found in new computational models.

Strongly influenced by chemical reactions, MP system has a reaction-like notation—that can be seen, also, as a *formal grammar* one—supported by recurrence equations and shifts the focus from pointwise string rewriting to a population transformation through the usage of conventional mole concept (as in chemistry). By the other side, its computational-simulative part is supported by linear algebra and relies on matrix operations for solving the recurrence equation system that characterizes the MP system as a (discrete) dynamical one.

For the sake of abstraction, an MP system can be mathematically represented using the support of a kind of formal grammar (named *MP grammar*) or as a particular type of graph (names *MP graph*). A mathematical description of both representations follows.

Definition 1 (MP grammar)**.** *An MP grammar G is a generative grammar for time series defined as*

$$G = (M, R, I, \Phi)$$

where:

1. $M = \{x_1, x_2, \ldots, x_n\}$ *the finite set of substances (or metabolites), and* $n \in \mathbb{N}$ *the quantities of substances.*
2. $R = \{\alpha_j \to \beta_j \mid 1 \leq j \leq m\}$ *the set of rules (or reactions), with* α_j *and* β_j *multisets over* M*, and* $m \in \mathbb{N}$ *the number of reactions.*
3. $I = (x_1[0], x_2[0], \ldots, x_n[0])$ *is the vector of initial values of substances or the metabolic state at initial step (step 0).*
4. $\Phi = \{\phi_1, \phi_2, \ldots, \phi_m\}$ *is a set of functions (also called regulators), in which every* $\phi_j : \mathbb{R}^n \mapsto \mathbb{M}$*, for* $1 \leq j \leq m$*, is associated with a rule* $r_j \in R$*.*

Conversely, we may can define the MP graph.

Definition 2 (MP graph)**.** *An MP graph is a directed, labeled hypergraph defined as*

$$\mathcal{H} = (V, \mathcal{E})$$

where:

1. $V = M \cup \{\vdash, \dashv\}$
2. $\mathcal{E} = \{e_1, e_2, \ldots, e_n\}$ *in which* $\forall i \in \mathbb{N}, e_i = (T, H)$ *with sets* $T \subseteq 2^{V \setminus \{\dashv\}}, H \subseteq 2^{V \setminus \{\vdash\}}$

3. $\ell : \mathcal{E} \mapsto \Phi$ *is the labeling function for each hyperedge.*

According to definition 1, G generates (a set of) time series, each of them representing the "amount of quantity" of the substances during the time. However, it is important to remind the discrete nature of the MP system and, although possible that the time series cardinality is infinite, it would be, at most, $\aleph_0$ (countable infinite set of values). Hence, the time series are calculated for any time t if and only if $\frac{t}{\tau} \in \mathbb{N}$ for a given constant τ.

Notwithstanding, the rules $\alpha_i \to \beta_i \in R$ depends, as equivalently happens in chemical reactions, on the quantities of the "substances" in the system, which can be expressed with the support of two different concepts: one that maps the multiplicity expressed in the rule for a substance to the actual number of molecules (of the substance) in the system, and; the quantity of mass the unit of the multiplicity represents (for a particular substance).

Hence, if the aforementioned three restrictions are that in consideration along with an MP grammar G, it is formally defines a `MP system`.

Definition 3 (MP system)**.** *A* MP system M *is a discrete dynamical defined as*

$$\mathcal{M} = (G, \tau, \nu, \mu)$$

with

1. G *being an MP grammar following the definition 1;*
2. $\tau \in \mathbb{R}$*, the period (amount of time) of a computational step;*
3. $\nu \in \mathbb{R}$*, the number of molecules that represents the (conventional) mole in the system;*
4. $\mu \in \mathbb{R}^n$ *is the vector of the mole masses of substances.*

It is important to note there is a *parametric* form of the MP system in which there is a set of parameters $P = \{p_0, p_1, \ldots, p_m\}$, with $m \in \mathbb{N}$, that may influence the regulators of $\mathcal{M}$, i.e. $\exists \phi_k \in \Phi \wedge p_q \in P : \phi_k \propto p_q$.

The (computational) step As described in definition 3 and restated many other times in text, MP system is a discrete dynamical system, in which the amounts of the substances are dependent of its previous values and a variational functional that may depend on other substances and parameters (in the case of the parametric MP system). This additional variance

through time is represented as a recurrent equation which the future value of a substance X is represented as $x[i+1] \propto x[i]$.

For the computation of these step values, nonetheless, two mathematical accessories were developed. The first, the *stoichiometric matrix*, is based on the arithmetic executed over chemical reactions to calculate the balance of molecules in a chemical system; the other, *equational metabolic algorithm*, synthesizes the whole computational process specified by the an MP system.

Definition 4 (Stoichiometric matrix)**.** *Let $r_i = \alpha_i \to \beta_i$, where α_i (with an equivalent for β_i) is represented as $\sum k_{i,j}^{+} \times X_j \mid k_{i,j} \in \mathbb{N} \wedge X_j \in M$.*

Let $\text{mult}^{+}(X_j, r_i) = k_{i,j}^{+}$ *be the multiplicity, for the right side (α_i) of the rule r_i, of the substance X_j in the rule. Similarly, there is* $\text{mult}^{-}(X_j, r_i) = k_{i,j}^{-}$ *for the left side (β_i) of the rule.*

A stoichiometric matrix $\mathbb{A}$, of dimension $|M| \times |R|$, has each of its elements defined by

$$a_{l,m} = \text{mult}^{+}(X_l, r_m) - \text{mult}^{-}(X_l, r_m)$$

with $, 1 \leq l \leq |M|$ *and* $1 \leq m \leq |R|$.

Definition 5 (Equational metabolic algorithm—EMA)**.** *Let $U[i] = (\phi_1(i), \phi_2(i), \ldots, \phi_m(i))^T$ be the vector of values, in the time step i, of all regulators, and $\mathbb{A}$ the stoichiometric matrix.*

The vector of substance variation at step i, $\Delta[i]$, *is computed by the equation*

$$\Delta[i] = \mathbb{A} \times U[i]$$

so-called Equational Metabolic Algorithm *whom computes the value of any substance in the time future time step $i+1$ through the recurrent equation*

$$X[i+1] = X[i] + \Delta[i]$$

The above definitions, now, complete specifies the discrete dynamical system and the ways to compute its values. For the fixation of the concept, now, an example ([16, § 2.4.2] and [14, § 3.1.1]) will be developed.

Example 1 (Lotka-Volterra Dynamics)**.** *The Lotka-Volterra dynamics is a simple model of the population dynamics of a prey x and a predator y coexisting in the same environment. It is described by the following ordinary*

differential equations, in which the parameters (A, B, C, D) represents the average mortality and reproduction factors of the two species.

$$\frac{dx}{dt} = (A - By)\, x$$

$$\frac{dy}{dt} = (Cx - D)\, y$$

In the MP system modeling language, the Lotka-Volterra dynamics is defined through four different rules:

1. *reproduction of the prey:* $r_1 : x \to 2x$*;*
2. *death of the prey:* $r_2 : x \to \emptyset$*;*
3. *reproduction of the predator:* $r_3 : y \to 2y$*;*
4. *death of the predator:* $r_4 : y \to \emptyset$*.*

It is important to note that the rules do not express the functional correlation between prey and predator, but the "rewrite of the multiset", i.e., each prey subject x presented in the environment (multiset) is either substituted by two other subjects ($2x$*, in the case of reproduction) or none (represented by the special symbol* $\emptyset$*, expressing the case of death). The same reasoning is valid for predator rules.*

For the expression of the correlation between prey and predator, regulators are associated to each rule, written in a functional and highlighting the interdependence of the populations through the (function) parameters.

Finally, all the elements of the MP grammar for the Lotka-Volterra dynamics are defined and an instance of the it is depicted in table 1.

Rules	**Regulators**
$r_1 : x \to 2x$	$\phi_1 = 3 \times 10^{-2} x$
$r_2 : x \to \emptyset$	$\phi_2 = 9 \times 10^{-4} + 9 \times 10^{-3} xy + 10^{-4} x^2 y$
$r_3 : y \to 2y$	$\phi_3 = 9 \times 10^{-4} + 1.5 \times 10^{-2} xy + 3 \times 10^{-4} x^2 y$
$r_4 : y \to \emptyset$	$\phi_4 = 6.6 \times 10^{-2} y$

Table 1: An instance of an MP grammar for the Lotka-Volterra dynamics.

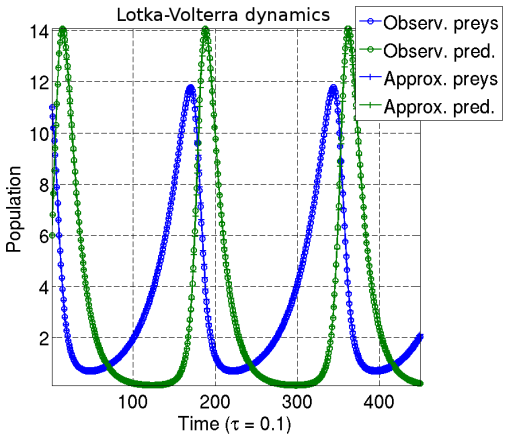


Figure 1: Prey and predators population for $x[0] = 11$ and $y[0] = 6$ for both MP formulation (table 1) and the differential equation one [16, § 2.4.2].

Dynamical Inverse Problem Although the MP system presents itself as a good strategy to describe dynamical systems, it requires the discovery of the correct algebraic form for regulators—what can be found through the usage of heuristics and reasoning over the dependency relations of the data—as well as the proper constant function values, what may require a huge amount of simulations and trials. The above definitions do not provide an algorithmic way for the inference of those values.

Nonetheless, it is important to note that the discovery of the (algebraic) rules behind a dynamical system, given a series of input values and another of observed output ones, is not new, with reports from the 17th century. Hence, it is not unexpected the presence of it in MP modeling instances.

Attentive to the utility of a procedure that could solve the aforementioned problem, Manca and Marchetti developed an inference algorithm which takes two different kinds of time series—correspondent to the inputted and outputted data from a dynamical system—and a dictionary of functions to perform, through the heavy use of linear algebraic operations and statistics, inference of the regulators functions that minimizes the approximation error. For this, they have called it the *Log-Gain Stepwise Regression (LGSS)* algorithm for the reason of some internal procedures used to compute the matching functions and constant for the regulators functional representation. For the sake of brevity and clearly of context, the details of the algorithm will be omitted; nonetheless, a complete description of it can be found in [14].

It is important to note that the application of LGSS has been successful in a series of environments, such as mathematical functions, complex systems

and biomedical studies.

3 Signals, Systems and Circuits

A *signal* is an information carrier [24, 11]. For a reader, a paper is a signal. For an electrical engineer, the time series of a potential difference (voltage) is a signal. For a biochemist, a molecule is a signal.

By the other side, a *system* is a process that generates or modifies a signal [11]. Reconsidering the examples of the last paragraph, a reader is a system for the paper signal. An electrical component is a system for a potential difference signal. And an enzyme is a system for a particular molecule acting as a signal.

Systems, on the other hand, can be composed of other (smaller) systems to create bigger systems. It means that systems may work as mathematical composable abstractions, like arithmetic operations—*multiplication* operator is composable of a series of *sum* operations—or functional ones—let $f(x) = \sin(x)$ and $g(x) = \cos(x)$, then $(f \circ g)(x) = \sin(\cos(x))$. Composability of systems is a very powerful property since it provides a rich set of applications that goes beyond simply transform an input signal by systems, but also the production of complex behaviour such as long range communication, encoding of data and feedback control systems [11, § 1.2].

The representation of a system, on the other hand, must comprise all the aforementioned characteristics and, in addition, provide a direct, elegant and simple description of it, independent of how complex are your building components. Hence, for the purpose of generality, block diagrams are the standard informal way of representation of systems.

It is important to note that although other representations such as functional composition ($\circ$) may be applied for some (simple) systems, there is a large set of systems that the complexity exceeds the reasonable usage of those notations or the visual representation of connected "boxes" are more expressive. For instance, the *feedback control systems* are clearly represented in block diagrams than in standard mathematical notations.

Once the concept of systems is relatively vague, for the sake of clarity they are classified in several categories depending on the signal spaces they are subject to, mostly dependent of the author that defines them.

According to Tom Henzinger [9], nonetheless, the systems may be classified in two big categories, *transducive* and *reactive* systems, with the latter containing five subcategories.

A *transducive system*—also called *combinatorial system*—is the one that acts as function applied to (domain) and producing (range) values; in func-

tion notation, it is defined as $f : X \mapsto Y$. Its behaviour independent of the time, always producing the same output $f(x) = y$ for the input value y. Lending the jargon from functional programming languages, it is said that transducive systems don't produce *side effects*.

A *reactive system*, however, has a broader definition. Firstly, its input data are defined in high level as time-dependent functions—in opposition to values only. Then, it may depend on present but also past information, bringing more general behaviour to the system specification.

There are five classifications for reactive systems depending on its internal definitions: (i) memory-free; (ii) delays; (iii) finite-memory; (iv) infinite-memory; and (v) causality systems.

The *memory-free systems* are those in which the output values are an instantaneous transformation of the input values, i.e., $y(t_i) = f(x(t_i))$, where $y(t_i)$ is the output produced at the time t_i, $x(t_i)$ is the data inputted at the same time t_i and $f(x)$ is the transformation function of the system.

Delay systems, by the other hand, are those systems in which the outputted values are transformations over "delayed inputs", i.e.,

$$y(t_i) = \begin{cases} \kappa & \text{if } t_i < \delta \\ f(x(t_i - \delta)) & \text{if } t_i \geq \delta \end{cases}$$

where κ is a constant value defined by the system.

The delay systems, as defined above, introduce the concept of *memory* to systems' formalism, enabling a system to dependent on current and previous values for computation. Then, the concepts of finite- and infinite-memory systems are, actually, specializations of delay systems. The difference of these systems rely on the cardinality of the sets defining the signals: *finite-memory* ones may be defined as discrete-time delay systems over finite set of values while *infinite-memory* those defined over continuous-time delay systems or infinite set of values.

The last of categorizations of reactive systems is the *causal* (or *implementable*) ones. Causal systems are those in which two equals inputted signals are equally transformed, i.e., let $S = \{\psi \mid \psi : Time \mapsto Values\}$ be the set of signals and $f : S \mapsto S$ be a causal system; then the *if-and-only-if* relation is satisfied: $\forall x', x'' \in S, \forall t \in Time$, if $\forall \tau \in Time, \tau \leq t \Rightarrow x'(\tau) = x''(\tau)$, then $f(x')(t) = f(x'')(z)$.

Additionally, it is useful to include one more class of (delay) systems: the *feedback systems*. In these systems, the input data $x(t_{i+\delta})$ is (composed by) the output $y(t_i)$, as depicted in figure 2.

Feedback systems are always present in control and recurrent systems, and, hence, very common to arise in the modeling of real-world problems.

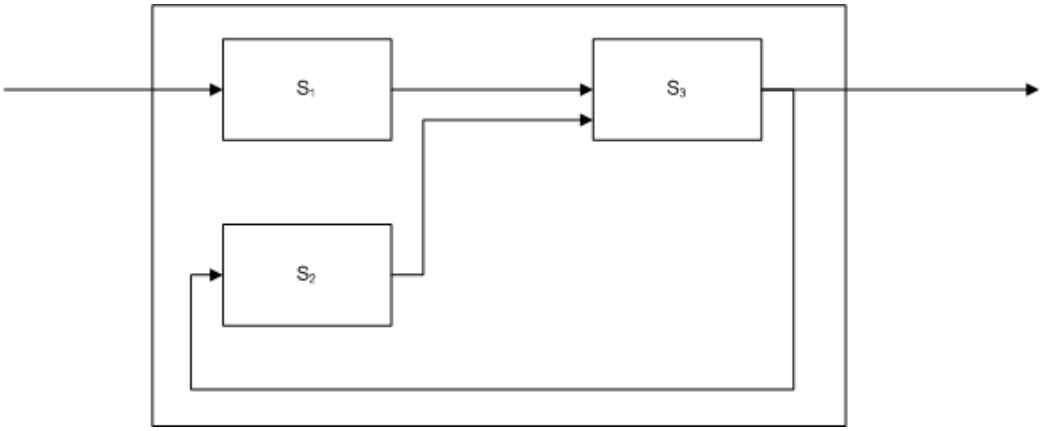


Figure 2: An example of a *feedback system*.

However, they also present an additional complexity to solve their functional representation, requiring a *fixed point* solution.

Electrical Circuits The mathematical formulation of signals and systems give abstract solutions for signal transformation problem, but does not give light on how to manipulate, in reality, the physical signals and its properties: it provides solution without specifying the way an actuator should be constructed to perform its goal in the physical world.

An *electrical circuit*, nonetheless, is a mathematical model of physical electrical circuits in which is easily possible to convert on a real actuator. Based on a series of electrical components, it provides in a graphical representation both the mathematical properties of the system [20, ch. 1, §Circuit Theory] for its detailed analysis—be it through the simplification of electromagnetical theories or graph theory study [3, ch. 2]—as well as the physical components required for the implementation of the system [20, 24].

The electrical circuits are generally classified in *analog* and *digital* depending on the type of input signal they do process. An analog signal is the one that

> is analogous to the physical signal that it represents [...] and exhibits a continuous variation over its range of activity. [24, p. 11]

If a signal is though as a function (as defined in [11, 9]), an analogous signal may be defined as $f_a : \mathbb{R} \mapsto \mathbb{R} \wedge \forall t \in \mathbb{R}^+, \exists \frac{df_a}{dt}$, i.e., the signal is derivable for a time $t \geq 0$.

By the other side, if the representation of a physical signal is done by a sequence of uniformly time-spaced (or periodically) values with finite number of digits characterizing the magnitude of the waveform, it is called a digital signal [24]. A difference between the two signal can be seen in the figure 3.

The discrepancies between analog and digital circuits go beyond the input signals and influences the usages and choices of the electrical components used in the circuit design. Resistors, inductor, capacitor, amplifiers, diodes and many others are usually apply to build analog electrical circuits [20, 24]. Logic gates, flip-flops and counters are typically used in digital circuits [24, 28].

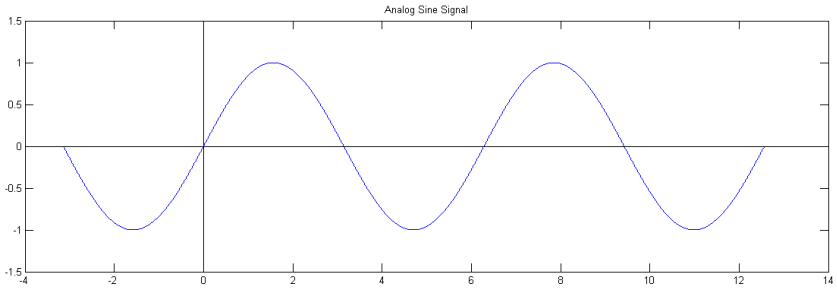
4 Equivalence Relation Between Metabolic P Systems, Reactive Systems and Electrical Circuits

The metabolic P systems have been successfully applied in the modeling of discrete dynamical systems, oscillatory and periodic patterns and, in particular, chemical and biological studies. In all these instances, in particular the latter one, the application of the theory was straightforward given the Nature-inspired model that is based in and the requirements of the projects—finding a mathematical model for observed (experimental) behaviours.

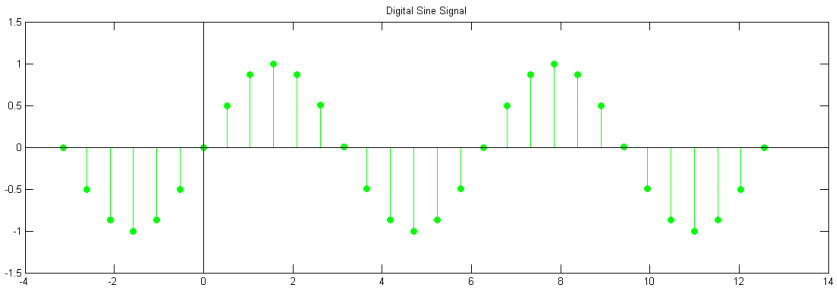
However, much of the produced modelings presented signal behaviours and graphical representations (MP graphs) that resemble electrical circuits and, in particular as noted by Manca and Marchetti, digital circuits. In parallel, inspired by the work on long-term potentiation done by Terje Lømo [13], an application of the dynamical inverse problem through MP system (section 2) to the electrical signals of neurological origins, followed by a synthesis of an electrical circuit that could reproduce them, was being thought by the author.

In fact, the idea of implementing metabolic computing models in hardware is shared among other research groups [21, 6, 8, 19], but with diverse focus: in general, the idea is to implement a metabolic-based computational model in hardware, not to find equivalent instances of problems in electrical circuits; also, they rely strongly on digital composable systems instead of general electrical circuits.

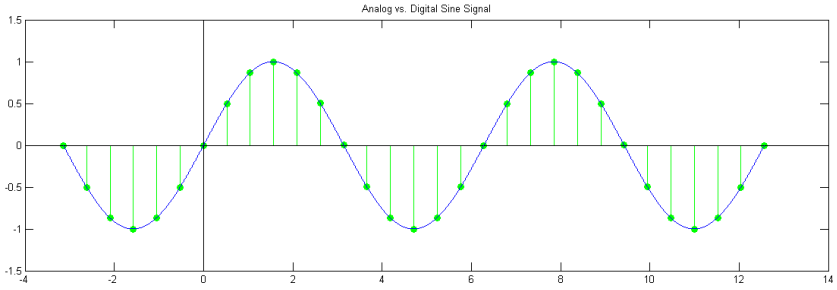
The current proposal, diversely from the aforementioned efforts, seeks for a symmetric equivalence relation [12] between metabolic P systems and reactive systems, more specifically electrical circuits—either analog or digital ones—in a way it is possible to convert a model described in one of these



(a) Analog $\sin(t)$ signal.

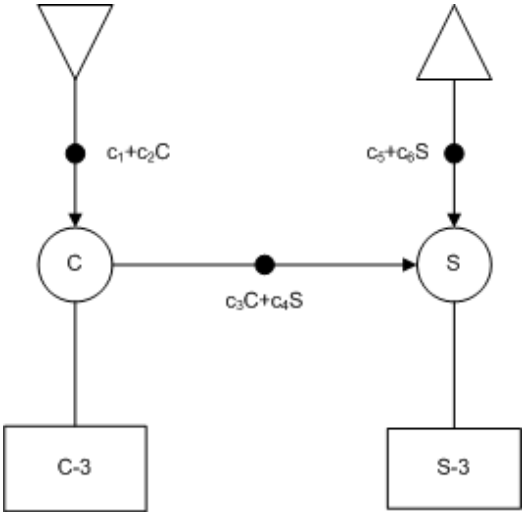


(b) Digital $\sin(t)$ signal with $T = \dfrac{\pi}{6}$.

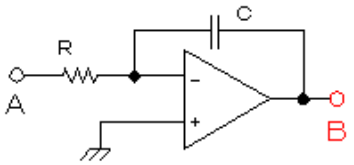


(c) Analog *versus* digital $\sin(t)$ signals.

Figure 3: The differences between the analog and digital signals of $\sin(t)$.



(a) MP graph of the Goniometricus system [14, § 3.3.1].



(b) Electric circuit schematic for a sine-cosine oscilator [10].

languages in another, in a lossless way. A similar to the approach for gene regulatory networks was subject of Marchetti in [16, 18], in which he defines a table of equivalence among elements of this kind of graphs and the MP ones.

Suppose the Goniometricus dynamics [14, § 3.3.1], represented by the MP graph in the figure 4a, where two substances, let us say S and C, oscillates in accordance to the sine and cosine functions, respectively. Its behaviour may be reproduced by an equivalent representation in an analog circuit as depicted in figure 4b, where the A is introduced the sine signal and B outputs the cosine one. The properties of phase difference between the signals is captured and the choice of the correct components (a capacitor and inverter) is chosen to reproduce the correspondent output signal.

Similarly, the same could be applied to digital systems.

For the success of the proposed topic, nonetheless, a structured research methodology should be applied, prioritizing "low-hanging" results and increasingly evolving the complexity and generalization of the equivalence. In this sense, the MP system-to-digital circuit equivalence should take immediate attention given its restricted scope over the analog ones and familiarity

of the involved ones with the subject.

Hence, the study concerning the digital circuits should rely on the Boolean circuits [1], software-hardware equivalence [27, 29] and the reduction of the software simulations from existing tools as MetaPlab [5] and the new MP Theory Java Library [17] to a small set of instructions that can be also implemented in hardware, as happens in hardware description languages such as VHDL.

Conversely, the transformation of MP systems into analogical circuits requires more intensive efforts and studies. Since it is known that systems and control theory models are converted in (also analogical) hardware, it is required to understand those transformations and try to import similar approaches to MP system. Also, deeper studies over analog circuits should be taken in an attempt to map circuitry components in a series of mathematical functions (or *regressors*, in the MP vocabulary), following the successful examples of conversion between MP graph and genetic signaling network [16, 18] and algebraic operations and digital circuitry [20].

Along with the aforementioned efforts, the MP-analog circuits transformation may require an adaptation on some of the MP system internal machinery seeking for a detachment from its common discrete representation — useful for computation performed by the standard digital hardware architecture of nowadays computers — to a broader and general one that may include the analogical operations, approaching the metabolic computing paradigm to the analog computing [25, 26, 2] one and, hence, providing new perspective on results and applications of the technique, moving away from limitations of the system as reduced regresssors dictionaries, fixed-point precision and errors provided by discretization of continuous domain data.

5 Conclusion

Metabolic P systems has been successfully applied in a range of applications, specially on the mathematical, chemical and biological fields. As a result of its effectiveness, partnerships in projects applying the methodology for experimental data analysis have been increasing, general usage software modeling tools have been developed and a solid research group around the subject has been established during.

The study and, later, existence of an equivalence between MP system and hardware systems would, nonetheless, create a symbiosis between the fields similarly to the existing software-hardware one, as well it could produce new applications for both fields. For example, known optimization techniques for design of schematics could be applied to MP models in an attempt to

produce simpler MP graphs and models of existing ones, specially those found by automatic reasoning [17]; in the opposite direction, biological systems could be expressed in MP formalization and then implemented in a hardware board (as a FPGA, for example) for experiments reproductivity reasons or developing "cell-on-a-chip" environments for bioengineering applications [7].

A last but not least important motivation for the search of the transformation lies in the didactic scope. At the moment, the vocabulary and instructive examples of MP systems relies on chemical, biological and theoretical mathematics niches what may work as an obstacle for the realization from new users coming from other fields of the power of metabolic computing when applied in a range of problems. Therefore, the interchangeability of concepts between MP, hardware and control theory models may expose the latter for a broader audience, eradicating the *Babel tower* obstacles among these areas.

References

[1] Sanjeev Arora and Boraz Barak. *Computational Complexity: A Modern Approach*. Cambridge University Press, 2009.

[2] Sunny Bains. Analog computer trumps Turing model, 1998.

[3] Béla Bollobás. *Modern Graph Theory*, volume 184. Springer, 1998.

[4] Cristian Sorin Calude and Gheorghe Păun. Bio-steps beyond Turing. *Bio Systems*, 77(1-3):175–94, November 2004.

[5] Alberto Castellini and Vincenzo Manca. MetaPlab: A Computational Framework for Metabolic P Systems. In DavidWolfe Corne, Pierluigi Frisco, Gheorghe Păun, Grzegorz Rozenberg, and Arto Salomaa, editors, *Membrane Computing SE - 12*, volume 5391 of *Lecture Notes in Computer Science*, pages 157–168. Springer Berlin Heidelberg, 2009.

[6] L. Fernandez, V.J. Martinez, F. Arroyo, and L.F. Mingo. A hardware circuit for selecting active rules in transition P systems. In *Seventh International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC'05)*, page 4 pp. IEEE, 2005.

[7] Lauren Gravitz. Cell on a Chip, 2009.

[8] Abraham Gutiérrez, Luís Fernández, Fernando Arroyo, and Santiago Alonso. Hardware and Software Architecture for Implementing Membrane Systems: A Case of Study to Transition P Systems. In MaxH.

Garzon and Hao Yan, editors, *DNA Computing SE - 22*, volume 4848 of *Lecture Notes in Computer Science*, pages 211–220. Springer Berlin Heidelberg, 2008.

[9] Tom Henzinger and Laurent El Ghaoui. EECS 20 (Signals and Systems) Lecture Notes, 2001.

[10] Seiichi Inoue. Operation and Explanation of the Sine/Cosine Oscilattor.

[11] Edward Ashford Lee and Pravin Varaiya. *Structure and Interpretation of Signals and Systems*. Lulu, 2nd edition, 2011.

[12] Harry Lewis and Christos Papadimitriou. *Elements of the Theory of Computation*. Prentice-Hall, 1997.

[13] Terje Lømo. The discovery of long-term potentiation. *Philosophical transactions of the Royal Society of London. Series B, Biological sciences*, 358(1432):617–20, April 2003.

[14] Vincenzo Manca. *Infobiotics: Information in Biotic Systems*, volume 3 of *Emergence, Complexity and Computation*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.

[15] Vincenzo Manca, Luca Bianco, and Federico Fontana. Evolution and Oscillation in P Systems: Applications to Biological Phenomena. In Giancarlo Mauri, Gheorghe Păun, MarioJ. Pérez-Jiménez, Grzegorz Rozenberg, and Arto Salomaa, editors, *Membrane Computing SE - 4*, volume 3365 of *Lecture Notes in Computer Science*, pages 63–84. Springer Berlin Heidelberg, 2005.

[16] Luca Marchetti. *MP representations of biological structures and dynamics*. PhD thesis, Università degli Studi di Verona, 2012.

[17] Luca Marchetti. MP Theory Java Library, 2013.

[18] Luca Marchetti and Vincenzo Manca. A methodology based on MP theory for gene expression analysis. *Membrane Computing*, 7184:300–313, 2012.

[19] Van Nguyen, David Kearney, and Gianpaolo Gioiosa. An Algorithm for Non-deterministic Object Distribution in P Systems and Its Implementation in Hardware. In DavidWolfe Corne, Pierluigi Frisco, Gheorghe Păun, Grzegorz Rozenberg, and Arto Salomaa, editors, *Membrane Computing SE - 24*, volume 5391 of *Lecture Notes in Computer Science*, pages 325–354. Springer Berlin Heidelberg, 2009.

[20] James William Nilsson and Susan Riedel. *Electric Circuits*. Prentice Hall, 9th edition, 2010.

[21] Biljana Petreska and Christof Teuscher. A Reconfigurable Hardware Membrane System. In Carlos Martín-Vide, Giancarlo Mauri, Gheorghe Păun, Grzegorz Rozenberg, and Arto Salomaa, editors, *Membrane Computing*, pages 269—-285. Springer Berlin Heidelberg, 2004.

[22] Gheorghe Păun. Computing with Membranes. *Journal of Computer and System Sciences*, 61(1):108–143, August 2000.

[23] Gheorghe Păun, Grzegorz Rozenberg, and Arto Salomaa. *The Oxford Handbook of Membrane Computing*. Oxford University Press, 2010.

[24] Adel S. Sedra and Kenneth C. Smith. *Microeletrocnic Circuits*. Oxford University Press, 6th edition, 2009.

[25] Hava T. Siegelmann. Computation beyond the turing limit. *Science (New York, N.Y.)*, 268(5210):545–8, April 1995.

[26] Hava T. Siegelmann. Analog Computational Power. *Science*, 271(5247):373–373, January 1996.

[27] Bernard Tiong Gie Tan. Hardware and software equivalence. *International Journal of Electronics*, 47(6):621–622, December 1979.

[28] Ronald J. Tocci and Neal S. Widmer. *Digital Systems: Principles and Applications*. Prentice Hall, Upper Saddle River, 8th edition, 2001.

[29] Kuo-Pao Yang and Theresa Beaubouef. Equivalence of Hardware and Software: A case study for solving polynomial functions. In *2010 42nd Southeastern Symposium on System Theory (SSST 2010)*, pages 318–322. IEEE, March 2010.