

Recurrent Solutions to Dynamics Inverse Problems: A Validation of MP Regression

Vincenzo Manca* and Luca Marchetti

University of Verona, Verona, Italy

Abstract

The paper is focused on a new perspective in solving dynamics inverse problems, by considering recurrent equations based on Metabolic P (MP) grammars. MP grammars are a particular type of multiset rewriting grammars, introduced in 2004 for modeling metabolic systems, which express dynamics in terms of finite difference equations. Regression algorithms, based on MP grammars, were introduced since 2008 for algorithmically solving dynamics inverse problems in the context of biological phenomena. This paper shows that, when MP regression is applied to time series of circular functions (where time is replaced by rotation angle), the dynamics that is found turns out to coincide with recurrent equations derivable from classical analytical definitions of these functions. This validates the MP regression as a general methodology to discover deep logics underlying observed dynamics. At the end of the paper some applications are also discussed, which exploit the regression capabilities of the MP framework for the analysis of periodical signals and for the implementation of sequential circuits providing periodical oscillators.

Keywords: Dynamics; Regression; Problems

Introduction

Let us start by considering the following general Dynamics Inverse Problem (DIP). Given a time series $X = (X[i] \in \mathbb{R}^n \mid i < t \in \mathbb{N})$ of vectors of finite dimension n over the set \mathbb{R} of reals, indexed over a subset of t natural numbers (time points), find a state function $\delta: \mathbb{R}^n \rightarrow \mathbb{R}^n$ such that:

$$(X[i] \in \mathbb{R}^n \mid i < t) = (\delta^i(X[0]) \mid i < t) \quad (1)$$

where δ^i is the i -iteration of δ , with $\delta^0(X) = X$ and $\delta^{i+1}(X) = \delta(\delta^i(X))$ for every $i \in \mathbb{N}$ and $X \in \mathbb{R}^n$.

In other words, we pose the problem of finding an autonomous discrete dynamical system over \mathbb{R}^n with initial state $X[0]$ and a dynamical function δ that is able to provide a given time series, as trajectory originated from a given initial point. When such a solution is found (possibly within an approximation threshold), we can say that a given trajectory in \mathbb{R}^n is represented in terms of a suitable next-state-function defined in the same space. Dynamics inverse problems were the beginning of modern science aimed at discovering the motion laws of planets around the sun. In general, a typical problem of mechanics is the determination of motion laws: from the observed motion to the underlying equations deduced from the knowledge of the forces acting on bodies. The approach we will outline where is similar, but here the forces as "causes of motion changes" are not assumed. Rather, we are interested in inferring a possible (approximate) internal logic regulating how (instead that why) changes of variables are cooperatively organized in a given system. This of course is a solution less precise and less explicative than the classical approach (usually based on ordinary differential equations). However, very often, in very complex systems with poor Information about the causes acting in a system, it is the only possibility that can be realistically investigated. The dynamics inverse problem was investigated in the context of MP theory developed in the last ten years (see, for example, [1-6]). In this paper, when we refer to MP regression, we intend the LGSS (basic) MP regression algorithm, integrating algebraic and statistical regression methods in the MP framework [7-10], that resulted completely satisfactory in all the considered cases. Dynamics inverse problems are crucial in many applicative contexts, where solving a DIP means discovering a

logic relating the variables of a phenomenon, by passing from a time external manifestation to a state internal causation [11-13]. Here, we present an MP solution of a mathematical problem related to circular functions: which are the MP recurrent equations discovered by the MP regression from time series of sine and cosine? We will show that the solutions, expressed by MP grammar for sine and cosine, found by means of LGSS algorithm, correspond to formulae deduced from elementary trigonometric identities or Taylor series. In a sense, this is an evidence of the validity of LGSS algorithm in inferring internal regularities underlying a given dynamics. An MP grammar G is given by a structure [10]:

$$G = (X, R, \Phi)$$

where:

- X is a finite set of real variables;
- R is a finite set of rules and each rule $r \in R$ is expressed by $\alpha_r \rightarrow \beta_r$ with α_r, β_r multisets over X (functions assigning a multiplicity to every variable);

$\Phi = \{\varphi_r \mid r \in R\}$ is the set of regulators, or flux functions

$$\varphi_r: \mathbb{R}^n \rightarrow \mathbb{R}$$

for every $r \in R$, where \mathbb{R}^n is the set of possible states of X , that is, column vectors of \mathbb{R}^n , which we continue ambiguously to denote by X (a state assigns to every variable a real value, in a given order). Any regulator φ_r associates a flux value u to every state of \mathbb{R}^n . A flux u of a rule r establishes an updating of the current state of variables, by decreasing of $u \cdot \alpha_r(x)$ the value of any variable x occurring in α_r and by increasing

*Corresponding author: Vincenzo Manca, University of Verona, Verona, Italy, Tel:390458027981; E-mail: vincenzo.manca@univr.it

Received June 05, 2014; Accepted July 18, 2014; Published July 24, 2014

Citation: Manca V, Marchetti L (2014) Recurrent Solutions to Dynamics Inverse Problems: A Validation of MP Regression. J Appl Computat Math 3: 176 doi:10.4172/2168-9679.1000176

Copyright: © 2014 Manca V, et al. This is an open-access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

of $u \cdot \beta_r(y)$ the value of any variable y occurring in β_r , where $u \cdot \alpha_r(x)$ and $u \cdot \beta_r(y)$ are the multiplicities of x and y in α_r and in β_r , respectively.

When an initial state $X[0]$ is given to an MP grammar G , then, starting from it, we pass from any state to its following state by applying all the rules of the grammar, that is, by summing the decrements and increments acting on each variable, according to all the rules where it occurs. It is easy to show that the corresponding dynamics is expressed with recurrent equations, which are synthetically represented in matricial form [10].

When variables are equipped with measurement units (related to their interpretation), and a time duration is associated to each step, the MP grammar is more properly called an MP system.

The term grammar is due to two reasons. First, rules $\alpha_r \rightarrow \beta_r$ become rewriting rules, of Chomsky grammars considered in Formal Language Theory [14], when α_r and β_r are strings (multisets are strings where the occurrence order is not relevant). Secondly, an MP grammar generates, from its initial state, a sequence of states. Analogously, a Chomsky grammar generates a set of strings from an initial string. The attribute MP comes from the initial application context suggesting MP grammars [1] extending Paun's P systems (multiset rewriting rules distributed in compartments) [15-17]. Applications in modeling biological systems were developed in the last years [3,10,13,18-22].

MP grammars have an intrinsic versatility in describing oscillatory phenomena [13,14]. The schema of MP grammars given in Table 1, called bicatalyticus [10], has an input rule r_1 and an output rule r_3 (incrementing and decrementing the variable x and y , respectively). Both rules are regulated by the same variable that they change (a sort of autocatalysis), while the transformation rule r_2 from x to y is regulated by both variables (bicatalysis). An MP grammar of this type provides a simple model for predator-prey dynamics firstly modeled in differential terms by Lotka and Volterra [23]. The model assumes a simple schema ruling the growth of the two populations x ; y (preys and predators): preys grow by eating nutrients taken from the environment (according to some reproduction factor) and die by predation, while predators grow by eating preys and naturally die (according to some death factor). When predators increase then preys are more abundantly eaten and therefore they decrease. But prey decrease results in a minor food for predators which start to decrease (by providing a consequent increase of preys). This means that the increase of predators produces, after a while, their decrease (and symmetrically, a corresponding inverse oscillation happens for preys). This oscillatory mechanism of the sizes of the two populations is displayed in Figure 1 by the dynamics of the MP grammar of Table 1 with the following regulators:

$$\begin{aligned}\varphi_1(x) &= 0.061 \cdot x + 0.931 \\ \varphi_2(x, y) &= 0.067 \cdot x + 0.15 \cdot y \\ \varphi_3(y) &= 0.154 \cdot y + 0.403,\end{aligned}$$

from the initial state $x[0]=9.5$, $y[0]=6$.

The following matrix:

$$\begin{aligned}r_1: \emptyset &\rightarrow x & \varphi_1(x) \\ r_2: x &\rightarrow y & \varphi_2(x, y) \\ r_3: y &\rightarrow \emptyset & \varphi_3(y)\end{aligned}$$

Table 1: A general schema of MP oscillating grammars

$$A = \begin{pmatrix} 1 & -1 & 0 \\ 0 & 1 & -1 \end{pmatrix}$$

is the stoichiometric matrix of grammar in Table 1, which is constructed by 3 column vectors of integers R_1 , R_2 , R_3 expressing the differences between the coefficients of left and right occurrences of variables (a rule $3x \rightarrow x+y$ would be $(-2, 1)$ because $1x-3x=-2x$ and $y-0y=1y$). Figure 2 is an MP graph illustrating the MP grammar of Table 1, according to a natural representation of rules and regulators, introduced in [24], where: circles are variables, continuous edges with full circles are rules and discontinuous lines with grey circles are regulators (triangles are input and output elements).

An MP grammar G with n variables defines an autonomous discrete dynamical system [3,10] where the space is given by the set R^n of possible states of its variables, and the dynamical function δ is given by the following equation, for any (column) vector $X \in R^n$:

$$\delta(X) = X \oplus (A \times U(X)) \quad (3)$$

where:

\oplus is the vector sum;

A is the stoichiometric matrix, $A = \{(\beta_r(x) - \alpha_r(x))_{x,r} \mid x \in X, r \in R\}$;

$U(X)$ is the column vector of fluxes $(\varphi_r(X) \mid r \in R)$

Any MP grammar has an equivalent grammar (providing the same dynamics) where rules have an input or output form (with the empty multiset \emptyset , on the right or on the left of the rule). In these grammars the reciprocal influence of variables is realized by means of regulators. Moreover, any first-order recurrent systems of equations (where the value of a variable at step $i+1$ may depend only by variables at step i) identifies an MP grammar in input/output form, and vice versa [10].

The LGSS algorithm of MP regression

Let us consider a given DIP, and let us fix a set of m rules over n variables, then the Log-Gain Stoichiometric Stepwise algorithm (LGSS) [7-10] yields an MP grammar, with the given rules and the initial state $X[0]$ of the time series X , generating X (within an approximation threshold). To this end, we chose a set of d primitive functions g_1, \dots, g_d (at most of n variables), called regressors, by means of which regulators are expressed, as linear combinations of some of them. The selection of the regressors corresponding to each regulator is realized by means of suitable statistical tests based on Fisher's F-distribution within a strategy of Stepwise regression [25,26]. The determination of the coefficients of the chosen regressors is computed by combining the method of Least Square Evaluation with a procedure of Stoichiometric Expansion realized by means of suitable algebraic operations on matrices. The name LGSS is due to the use of Log-gain principle [3] for assigning scores in the choice of regressors. In fact, this is a crucial point of the procedure. More technical details on this algorithm can be found in [7-10].

An MP Grammar for Sine and Cosine

Sine and cosine functions can be defined as the values of x and y coordinates, respectively, of a point moving on a circle of unitary radius and center on the origin of xy plane. Circular functions are as old as the Archimedean investigation about the π ratio between circle length and its diameter. As illustrated in [27], Archimedean sine of an angle α is the cord under α , that is, $2 \sin(\alpha/2)$, and Archimedes' investigation about the rectification of circle relies on a kind of bisection formulae (from the cord of an angle to the cord of the half angle) that were

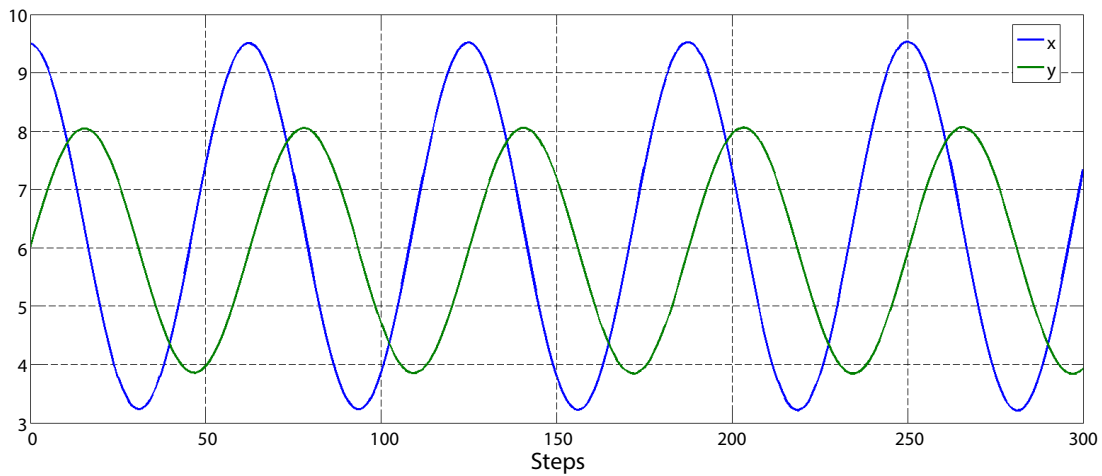


Figure 1: Dynamics of the MP grammar of Table 1 with regulators given in (2).

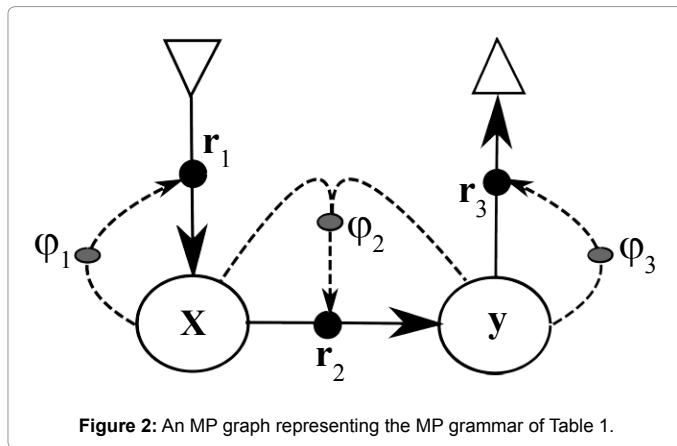


Figure 2: An MP graph representing the MP grammar of Table 1.

the basis for the approximate evaluation of π . Many classical results of trigonometry, such as addition formulae, are due to Arabian and Indian mathematicians (Bhaskara II, 12th century). The analytical representation of sine and cosine was one of Euler's jewels [28]. Now we are interested to know if a simple MP grammar exists that generates sine and cosine tables with a very good approximation. To this end, we start from the MP grammar schema of Table 1, which we know to be a good schema for providing an MP oscillator, and then we use LGSS with the following regressor dictionary $D = \{x^k y^j \mid 0 \leq k \leq 4, 0 \leq j \leq 4\}$ for inferring the right regulator formulae providing a cosine/sine oscillator (sine plays the role of predator and cosine the role of prey). When the time series of sine and cosine, with sampling at 10^{-3} rad, are given as input to the LGSS algorithm, in order to find the best regulators of the MP grammar of Table 1 (with regressor dictionary D), we get the following result (x indicate cosine, y indicates sine):

$$\begin{aligned} r_1 : \emptyset \rightarrow x \quad \varphi_1(x) &= k_1 \cdot x \\ r_2 : x \rightarrow y \quad \varphi_2(x, y) &= k_2 \cdot (x + y) \\ r_3 : y \rightarrow \emptyset \quad \varphi_3(y) &= k_3 \cdot y \end{aligned} \quad (4)$$

where $k_1=0.000999499833375$, $k_2=0.000999999833333$ and $k_3=0.001000499833291$ (the coefficient estimates are cut to the 15th decimal digits, according to the accuracy of the computer architecture used during the computation). MP grammar (4) provides a very precise

sine/cosine oscillator, with maximum absolute error of the order of 10^{-14} (Figure 3).

Let us search for a recurrent definition of sine and cosine by using the trigonometric addition formulae:

$$\begin{aligned} \cos(i+h) &= \cos(i) \cdot \cos(h) - \sin(i) \cdot \sin(h) \\ \sin(i+h) &= \sin(i) \cdot \cos(h) + \cos(i) \cdot \sin(h) \end{aligned} \quad (5)$$

that is:

$$\begin{aligned} \cos(i+h) - \cos(i) &= \cos(i) \cdot \cos(h) - \sin(i) \cdot \sin(h) - \cos(i) \\ \sin(i+h) - \sin(i) &= \sin(i) \cdot \cos(h) + \cos(i) \cdot \sin(h) - \sin(i) \end{aligned} \quad (6)$$

Let us approximate the value of $\sin(h)$ with the value h of the angle. Therefore, the value of $\cos(h)$ is:

$$\cos(h) = \sqrt{1 - \sin^2(h)} = \sqrt{1 - h^2}$$

If we substitute these values to $\sin(h)$ and $\cos(h)$ in equations (6) we obtain:

$$\begin{aligned} \cos(i+h) - \cos(i) &= \sqrt{1-h^2} \cdot \cos(i) - h \cdot \sin(i) - \cos(i) \\ \sin(i+h) - \sin(i) &= \sqrt{1-h^2} \cdot \sin(i) + h \cdot \cos(i) - \sin(i) \end{aligned} \quad (7)$$

which, finally, gives the following MP grammar (x; y are the variables for sine and cosine, respectively):

$$\begin{aligned} r_1 : \emptyset \rightarrow x \quad \varphi_1(x, y) &= \sqrt{1-h^2} \cdot x - h \cdot y - x \\ r_2 : \emptyset \rightarrow y \quad \varphi_2(x, y) &= \sqrt{1-h^2} \cdot y + h \cdot x - y \end{aligned} \quad (8)$$

Proposition 1: The MP grammar (8) coincides with the MP grammar (4) given by LGSS, up to the seventh decimal digit (the corresponding coefficients of the two grammars are the same within an approximation of 10^{-7}).

Proof: Let us modify the stoichiometry of the grammar (8) by reversing the second rule:

$$\begin{aligned} r_1 : \emptyset \rightarrow x \quad \varphi_1(x, y) &= \sqrt{1-h^2} \cdot x - h \cdot y - x \\ r_2 : y \rightarrow \emptyset \quad \varphi_2(x, y) &= y - \sqrt{1-h^2} \cdot y - h \cdot x \end{aligned} \quad (9)$$

Now, let us add a new rule (we write it in second position) with

null flux:

$$\begin{aligned} r_1 : \emptyset \rightarrow x \quad \varphi_1(x, y) &= \sqrt{1-h^2} \cdot x - h \cdot y - x \\ r_2 : x \rightarrow y \quad \varphi_2(x, y) &= 0 \\ r_3 : y \rightarrow \emptyset \quad \varphi_3(x, y) &= y - \sqrt{1-h^2} \cdot y - h \cdot x \end{aligned} \quad (10)$$

and then, let us move the term $h \cdot y$ from the first rule to the second rule and the term $h \cdot y$ from the third rule to the second rule:

$$\begin{aligned} r_1 : \emptyset \rightarrow x \quad \varphi_1(x, y) &= \sqrt{1-h^2} \cdot x - x \\ r_2 : x \rightarrow y \quad \varphi_2(x, y) &= h \cdot y + h \cdot x \\ r_3 : y \rightarrow \emptyset \quad \varphi_3(x, y) &= y - \sqrt{1-h^2} \cdot y \end{aligned} \quad (11)$$

This change introduces an error in variable step variations, with respect to those of the original grammar (9), because the flux $h \cdot y$, which was moved from r_1 to the r_2 , now acts also on the variable y (that is increased of it), and the flux $h \cdot x$, which was moved from r_3 to the r_2 , now acts also on the variable x (that is decreased of it). Therefore, in order to compensate these extra fluxes, we add the term $h \cdot x$ in the flux of r_1 and the term $h \cdot y$ in the flux of r_3 , by obtaining at end the following MP grammar equivalent to (9):

$$\begin{aligned} r_1 : \emptyset \rightarrow x \quad \varphi_1(x) &= [\sqrt{1-h^2} - 1 + h] \cdot x \\ r_2 : x \rightarrow y \quad \varphi_2(x, y) &= h \cdot (x + y) \\ r_3 : y \rightarrow \emptyset \quad \varphi_3(y) &= [1 + h - \sqrt{1-h^2}] \cdot y \end{aligned} \quad (12)$$

We show that this MP grammar is dynamically equivalent to the MP grammar (4). In fact, by construction, it has the same stoichiometry as the grammar (4). Moreover, if we set the step h equal to 10^{-3} , by substituting this value of h in Equation (12) we get the following grammar, where we get coefficients that coincide with the coefficients of the grammar (4) rounded at seventh decimal digit:

$$\begin{aligned} r_1 : \emptyset \rightarrow x \quad \varphi_1(x) &= (0.9999995 - 1 + 0.001) \cdot x = 0.0009995 \cdot x \\ r_2 : x \rightarrow y \quad \varphi_2(x, y) &= 0.001 \cdot x + 0.001 \cdot y \\ r_3 : y \rightarrow \emptyset \quad \varphi_3(y) &= (1 - 0.9999995 + 0.001) \cdot y = 0.0010005 \cdot y \end{aligned} \quad (13)$$

that is, we get for the three rules the constants k_1, k_2, k_3 of (4) (rounded at the seventh decimal digit). MP grammar (13) permits a very accurate computation of the cosine/sine dynamics. Its accuracy surely increases if we use a smaller step, say 0.000001. However, the grammar (4) provides, with the same step (0.001), a more accurate calculation, which is comparable to the precision accuracy of the computer (precision order of 10^{-14}). This means that LGSS, during its calculation, is able to calculate the regulator formulae with a precision that is bounded only by the precision accuracy of the computer which runs the calculations. We will show, in the next Proposition, that what LGSS gives at step 0.001, corresponds exactly to what can be derived by means of a recurrent definition of cosine and sine based on Taylor series:

$$f(x+h) = f(x) + h \cdot f'(x) + \frac{h^2}{2!} \cdot f''(x) + \frac{h^3}{3!} \cdot f'''(x) + \frac{h^4}{4!} \cdot f^{(4)}(x) + \dots$$

In fact, if we apply the formula above to cosine and sine we obtain:

$$\begin{aligned} \cos(i+h) &= \cos(i) - h \cdot \sin(i) - \frac{h^2}{2} \cdot \cos(i) + \frac{h^3}{6} \cdot \sin(i) + \frac{h^4}{24} \cdot \cos(i) - \dots \\ \sin(i+h) &= \sin(i) + h \cdot \cos(i) - \frac{h^2}{2} \cdot \sin(i) - \frac{h^3}{6} \cdot \cos(i) + \frac{h^4}{24} \cdot \sin(i) + \dots \end{aligned} \quad (14)$$

which, finally, leads to the following recurrent formulae:

$$\begin{aligned} \cos(i+h) - \cos(i) &= \left(-\frac{h^2}{2} + \frac{h^4}{24} - \dots\right) \cdot \cos(i) + \left(-h + \frac{h^3}{6} - \dots\right) \cdot \sin(i) \\ \sin(i+h) - \sin(i) &= \left(h - \frac{h^3}{6} + \dots\right) \cdot \cos(i) + \left(\frac{h^2}{2} + \frac{h^4}{24} - \dots\right) \cdot \sin(i) \end{aligned} \quad (15)$$

that, for the sake of simplicity, can be rewritten as follows:

$$\begin{aligned} \cos(i+h) - \cos(i) &= A \cdot \cos(i) - B \cdot \sin(i) \\ \sin(i+h) - \sin(i) &= B \cdot \cos(i) + A \cdot \sin(i) \end{aligned} \quad (16)$$

$$\text{Where } A = \left(-\frac{h^2}{2} + \frac{h^4}{24} - \dots\right) \text{ and } B = \left(-h + \frac{h^3}{6} - \dots\right).$$

In Table 2 are represented the values for the coefficients A and B of equations (16) for different orders of Taylor series and considering a step $h=10^{-3}$, which is the step used for computing the MP grammar (4). This means that, by substituting in recurrent equations (16) the values of A and B taken from Table 2, we can obtain a cosine/sine oscillator which approximates the curves of cosine and sine with sampling 10^{-3} and approximation increasing as the order of terms considered in the Taylor series. The best approximations, with respect to the available computer precision, are computed by using the values in the last row of Table 2.

Recurrent equations (16) can be rewritten in the following MP grammar:

$$\begin{aligned} r_1 : \emptyset \rightarrow x \quad \varphi_1(x, y) &= A \cdot x - B \cdot y \\ r_2 : \emptyset \rightarrow y \quad \varphi_2(x, y) &= B \cdot x + A \cdot y \end{aligned} \quad (17)$$

The following proposition shows that this recurrent definition of sine and cosine coincides with the MP grammar (4) found by means of LGSS algorithm.

Proposition 2: The MP grammar (17) with coefficients A, B of the fourth line of Table 2 is dynamically equivalent to the MP grammar (4) (the two grammars define the same dynamics).

Proof: Let us write the MP grammar (4) by changing its stoichiometry, but keeping the same variations of variables at any step. First, we change the direction of r_1 :

$$\begin{aligned} r_1 : \emptyset \rightarrow x \quad \varphi_1(x, y) &= k_1 \cdot x \\ r_2 : x \rightarrow y \quad \varphi_2(x, y) &= k_2 \cdot (x + y) \\ r_3 : \emptyset \rightarrow y \quad \varphi_3(y) &= -k_3 \cdot y \end{aligned} \quad (18)$$

then, we eliminate the transformation rule in the middle, by rewriting (18) as:

$$\begin{aligned} r_1 : \emptyset \rightarrow x \quad \varphi_1(x, y) &= k_1 \cdot x - k_2 \cdot (x + y) \\ r_2 : \emptyset \rightarrow y \quad \varphi_2(x, y) &= k_2 \cdot (x + y) - k_3 \cdot y \end{aligned} \quad (19)$$

Order	A	B
1	0	0:001
2	-5:00000000 • 10 ⁻⁷	0:001
3	-5:00000000 • 10 ⁻⁷	9:9999833333 • 10 ⁻⁴
4	-4:99999958 • 10 ⁻⁷	7 9:9999833333 • 10 ⁻⁴

Table 2: Values of A and B of Equation (16) for different orders of the Taylor series, by considering a step $h=10^{-3}$ (the estimates are cut to the 15th decimal digits, according to the accuracy of the computer used during the computation). Since, order by order, only one term of the Taylor series is added, which depends only by cosine or by sine, only one column of the table is updated in each row. From the 4th order the coefficients remain stable, that is, the change is reflected only in digits which are not reported in table because too low.

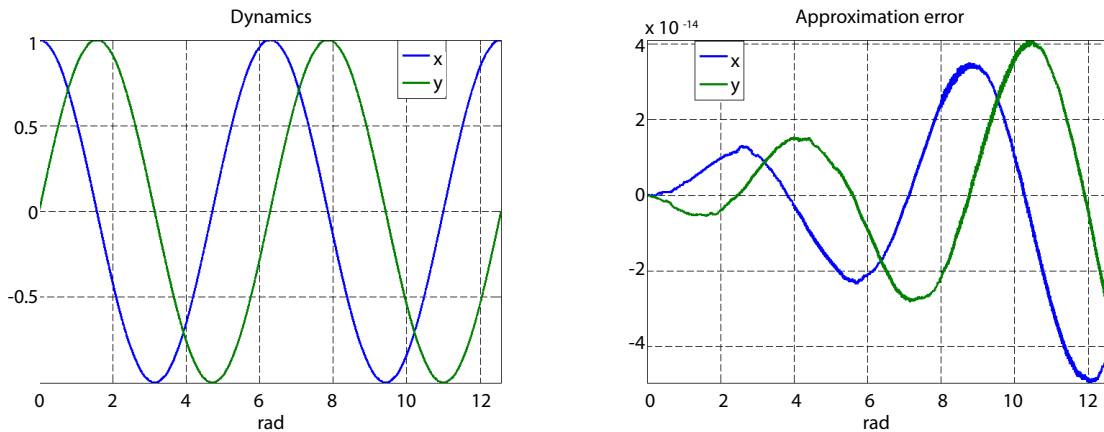


Figure 3: Dynamics and approximation error of the MP grammar (4).

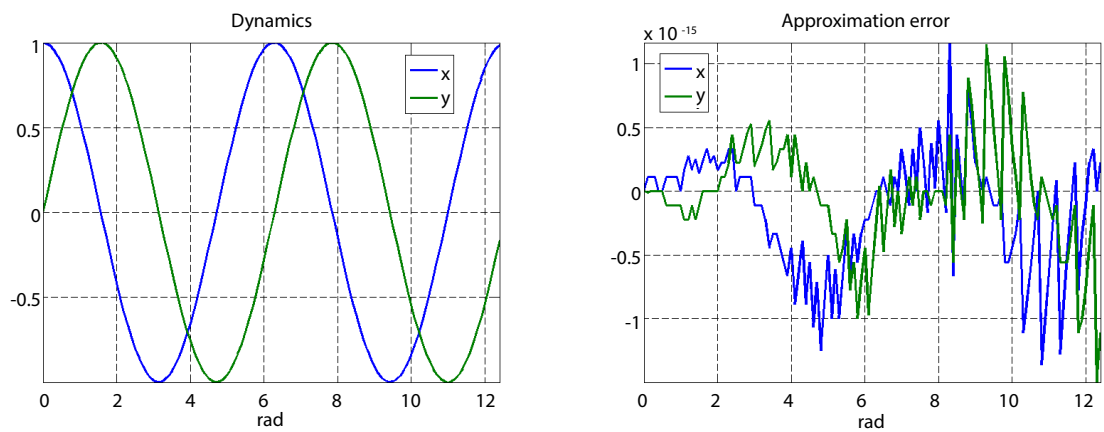


Figure 4: Dynamics and approximation error of the MP grammar (22).

The grammar above can be finally rewritten by grouping terms:

$$\begin{aligned} r_1 : \emptyset \rightarrow x \quad \varphi_1(x, y) &= (k_1 - k_2) \cdot x - k_2 \cdot y \\ r_2 : \emptyset \rightarrow y \quad \varphi_2(x, y) &= k_2 \cdot x + (k_2 - k_3) \cdot y \end{aligned} \quad (20)$$

and this last grammar leads to the conclusion that the MP grammar (17) is equivalent to the grammar (4) when:

$$\begin{aligned} A &= k_1 - k_2 = k_2 - k_3 \\ B &= k_2 \end{aligned} \quad (21)$$

If we use the estimates of k_1 , k_2 and k_3 computed by LGSS for the MP grammar (4), we obtain:

$$\begin{aligned} A &= k_1 - k_2 = k_2 - k_3 = -4.99999958 \cdot 10^{-7} \\ B &= k_2 = 9.9999983333 \cdot 10^{-4} \end{aligned}$$

which corresponds to the coefficients for A and B of 4th order provided in Table 2.

The proposition above shows that LGSS provides formulae and coefficients estimations that are mathematically well grounded and that provides the maximum power of approximation, according to the precision accuracy reachable by the machine performing the computation. LGSS computed coefficients corresponding to Taylor terms of 4th order, because this order provides the precision accuracy of 15 decimal digits. However, now that we are aware of this, we can

better explain the approximation loss of grammar (13) with respect to grammar (4). In fact, if we reconsider the evaluation (21) with the values of (13), we obtain:

$$\begin{aligned} A &= k_1 - k_2 = k_2 - k_3 = -5 \cdot 10^{-7} \\ B &= k_2 = 0.001 \end{aligned}$$

which corresponds to the coefficients for A and B of 2th order provided in Table 2. This means that the approximation power of grammar (4) is increased, with respect to grammar (13), of two orders of Taylor series.

If we apply LGSS directly to the stoichiometry of grammar (17), we will obtain the coefficients of the fourth line of Table 2. The following MP grammar:

$$\begin{aligned} r_1 : \emptyset \rightarrow x \quad \varphi_1(x, y) &= -0.004995834721974 \cdot x - 0.099833416646828 \cdot y \\ r_2 : \emptyset \rightarrow y \quad \varphi_2(x, y) &= 0.099833416646828 \cdot x - 0.004995834721974 \cdot y \end{aligned} \quad (22)$$

has been computed by LGSS, considering the stoichiometry of grammar (17) and a sampling step of 0.1. Even if the sampling step of this grammar is bigger than the one used for calculating grammar (4), this system provides a cosine/sine oscillator that is more precise. In fact, the maximum approximation error in the first two oscillations is of the order of 10^{-15} (Figure 4). It is interesting to see that, thanks to the fact that this time the considered sampling step is bigger, the coefficient estimates computed by LGSS consider more terms of Taylor series. In particular, coefficients of grammar (22) correspond to a Taylor series of

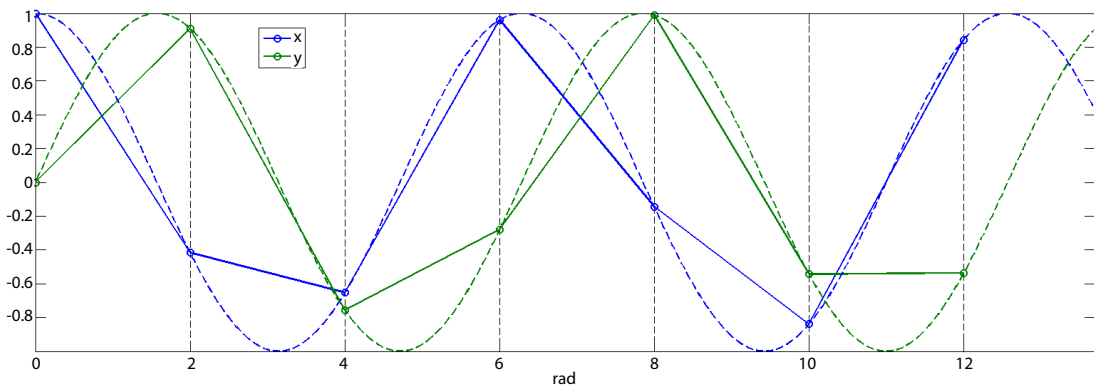


Figure 5: An example of MP dynamics providing sine/cosine dynamics at a very big sampling step (sampling of 2 rad.). For increasing the readability of the dynamics, a sine/cosine oscillatory pattern has been added to the background by dashed lines. Even if the considered sampling step is very big, the dynamics is very precise (the maximum approximation error in the first two oscillations is of the order of 10^{-16} , the coefficients of the corresponding grammar, computed by LGSS, correspond to a Taylor series of 23th degree).

Order	A	B
1	0	0.001
2	$-5.00000000 \cdot 10^{-3}$	0.001
3	$-5.00000000 \cdot 10^{-3}$	$9.9833333333333 \cdot 10^{-2}$
4	$-4.9958333333333 \cdot 10^{-3}$	$9.9833333333333 \cdot 10^{-2}$
5	$-4.9958333333333 \cdot 10^{-3}$	$9.9833416666666 \cdot 10^{-2}$
6	$-4.995834722222 \cdot 10^{-3}$	$9.9833416666666 \cdot 10^{-2}$
7	$-4.995834722222 \cdot 10^{-3}$	$9.9833416646825 \cdot 10^{-2}$
8	$-4.995834721974 \cdot 10^{-3}$	$9.9833416646825 \cdot 10^{-2}$
9	$-4.995834721974 \cdot 10^{-3}$	$9.9833416646828 \cdot 10^{-2}$

Table 3: Values of A and B of grammar (17) for different orders of the Taylor series, with step $h=0.1$.

9th degree (Table 3), the maximum degree that is possible to considerate with a precision of 15 decimal digits.

Applications

The LGSS regression described in MP Grammar for Sine and Cosine part demonstrated to be very reliable even when the discretization step of our analysis is realized at a very macroscopic level, for example by considering sine and cosine at angle increments of 2 rad. In such a case, the values computed by the MP grammar deduced by LGSS were determined with the best possible precision and with a surprising computational speed (Figure 5). In fact, the greater is the angle increment, the higher is the order of terms of Taylor series used by the grammar given by LGSS. And moreover, the greater is the angle increment, the smaller is the number of steps necessary for generating sine and cosine values (with a decreasing of error propagation along the iteration). The charts given in Figure 6 refer to a set of 400 MP models providing sine/cosine dynamics with sampling step ranging from 0.01 rad to 4 rad. For each model is given the mean square error (MSE) and the order of Taylor series reached by the coefficient estimates computed by LGSS. Thanks to the high performance of LGSS, the entire set of models has been computed and simulated in less than a minute by means of a standard laptop with a dual core CPU and 4 GB of memory. An explanation of this very good performance of MP grammars is due to their intrinsic capability of combining a sort of native built-in integro-differential strategy. In fact, in a differential representation of a dynamics, when solutions are found, analytically or by numerical integration, the behavior of variables at a given discrete

time sampling can be obtained by integrating the differential solution in the (macroscopic) time interval of the time step. The MP solutions given by LGSS provide directly this result by means of a very precise evaluation of regulators responsible for the observed dynamical variations between the two extremes of the step interval. The results presented till now apply not only to standard circular functions, but also to any harmonics $\cos(n \cdot i)$ and $\sin(n \cdot i)$, with $n \in \mathbb{R}$. In fact, from Equations (14), (15) and (16), we have:

$$\begin{aligned} \cos(n \cdot i + h) &= \cos(n \cdot i) - n \cdot h \cdot \sin(n \cdot i) - \frac{n^2 \cdot h^2}{2} \cdot \cos(n \cdot i) + \dots \\ \sin(n \cdot i + h) &= \sin(n \cdot i) + n \cdot h \cdot \cos(n \cdot i) - \frac{n^2 \cdot h^2}{2} \cdot \sin(n \cdot i) - \dots \end{aligned} \quad (23)$$

which leads to:

$$\begin{aligned} \cos(n \cdot i + h) - \cos(n \cdot i) &= A \cdot \cos(n \cdot i) - B \cdot \sin(n \cdot i) \\ \sin(n \cdot i + h) - \sin(n \cdot i) &= B \cdot \cos(n \cdot i) + A \cdot \sin(n \cdot i) \end{aligned} \quad (24)$$

$$\text{Where } A = \left(-\frac{n^2 \cdot h^2}{2} + \frac{n^4 \cdot h^4}{24} - \dots \right) \text{ and } B = \left(n \cdot h - \frac{n^3 \cdot h^3}{6} + \dots \right).$$

This means that MP grammar (17) has the capability of generating sine and cosine curves regardless of their frequency and that we can use LGSS to compute the right values for A and B. Now, since we know from Fourier analysis that any periodical signal can be represented by a combination of simple sine waves with different frequencies, this fact leads to important applications in using LGSS for generating recurrent equations that exhibit periodical signals. Moreover, since MP grammar (17) has linear regulators, it is easy to translate it in a combinatorial circuit (Figure 7). Such a circuit can be then incorporated in a sequential one, like that represented in Figure 8, which provides an implementation, in terms of hardware, of the MP system inferred by LGSS. This fact indicates that it is possible to define a methodology, based on LGSS regression that permits to implement sequential circuits of periodical oscillators. A paper is in preparation that investigates along this direction.

Conclusions

The results presented in this paper focus on a new perspective of considering recurrent equations. Usually, their investigation is aimed at finding analytical methods to solve them or at determining properties of the discrete dynamics that they express [29]. On the contrary, here

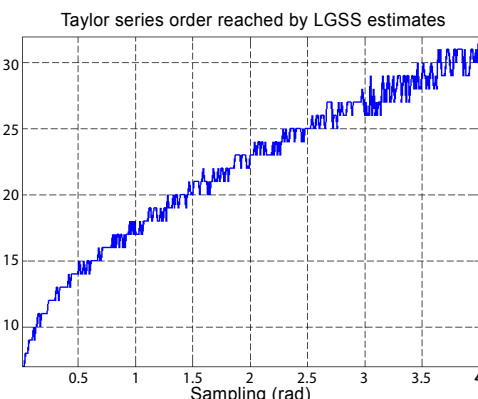
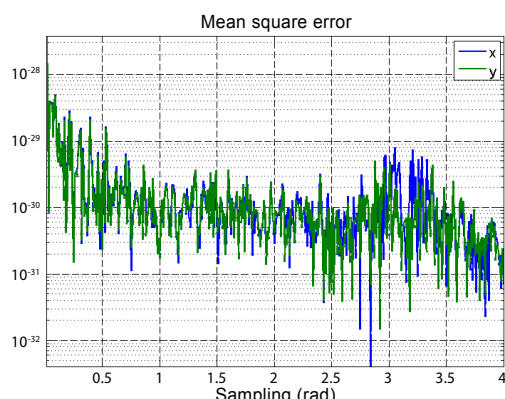


Figure 6: Mean square error (MSE) and order of Taylor series reached by the LGSS estimates for a set of 400 MP models providing sine/cosine dynamics with sampling step ranging from 0:01 rad. to 4 rad.

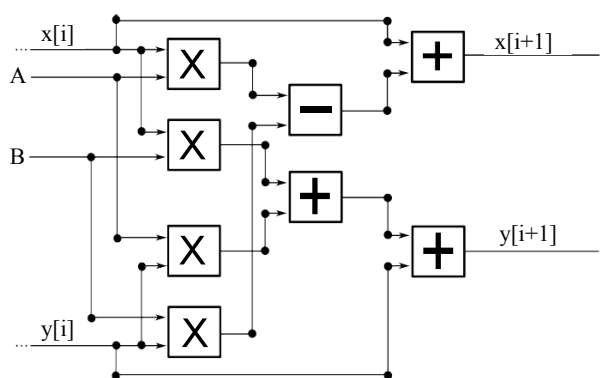


Figure 7: Combinatorial circuit obtained from the translation of MP grammar (17). Square blocks represent n-bit operations (sum, difference and multiplication).

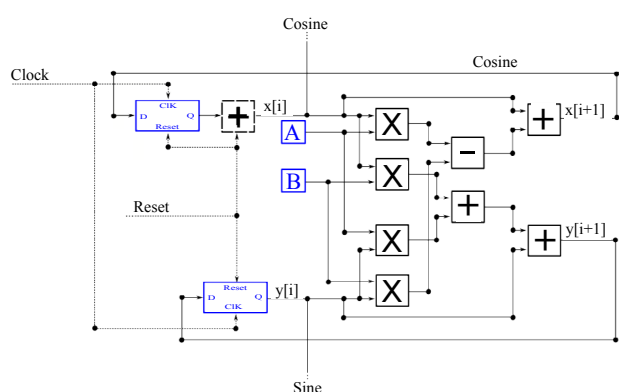


Figure 8: Sequential circuit that integrates the combinatorial circuit of Figure 7 for obtaining a fully functional sine/cosine oscillator. D-dashed lines represent single bit lines; components depicted in blue identify n-bit memories.

we do not cope with their solutions, because their intrinsic algorithmic (iterative) nature provides a direct computation of their dynamics: it is enough to put them in the canonical form $X_{n+1}=F(X_n)$ and iteratively apply the right side of the equation, starting from an initial value (every equational recurrence can be expressed in a first order recurrent form). Of course, in this way the dynamics at step n can be computed only

after computing it in all the steps preceding n , but, in most cases, this is not a real limitation if the computation is performed automatically. Using MP grammars, recurrent equations are constructed by assigning regulators to MP rules, that is, by combining the action of a number of regulators that encode abstract forms of mutual interactions among variables. In this perspective, regulators replace forces that in classical mechanics are the causes of observed motions. In fact, regulators may be related to a big number of unknown forces, very difficult to individuate and to discriminate, therefore, they express abstract entities of rational and compact reconstruction of the internal logic underlying an observed dynamics. When the complexity and the indetermination of systems do not allow us other ways of analysis, this could be an important chance to the comprehension of phenomena. In this paper we deduce a sort of proof of concept of the ability of MP regression to discover the deep logic underlying a phenomenon. Namely, LGSS implicitly applied trigonometric formulae and Taylor series, with the best possible accuracy and a powerful computational efficiency. We would like to stress that LGSS automatically calculates not only the values of the coefficients, but also the form of linear combinations providing regulators. This provides a big advantage with respect to other formalisms, such as GMA-systems [30], which reach a similar power of approximation, for circular functions, only because the form of differential equations, on which they are based, are constrained to fit with the form of Taylor series. Conversely, the exhibility of LGSS allows us to find the best regulators approximating the internal logic of the observed dynamics, without imposing a preliminary form. In papers [10-13,21,22] where MP theory was already successful applied to biomedical problems, very good approximations are obtained even when the regulators have a form that is different from that corresponding to Taylor series. Finally, beside the wide range of applications of the MP approach to the analysis of complex phenomena, in applications section some new applications are introduced that exploit the regression capabilities of LGSS for the analysis of periodical signals. In particular, at the end of the section a new application is suggested, that focus on the definition of a methodology, based on LGSS regression, for implementing sequential circuits of periodical oscillators.

References

1. Manca V, Bianco L, Fontana F (2005) Evolutions and Oscillations of P systems: Application to biological phenomena. LNCS 3365: 63-84.
2. Manca V (2008) The metabolic algorithm for P systems: principles and applications. Theoretical Computer Science 404: 142-157.
3. Manca V (2010) Fundamentals of Metabolic P Systems.

4. Manca V, Marchetti L (2010) Metabolic approximation of real periodical functions. *The Journal of Logic and Algebraic Programming* 79: 363-373.
5. Castellini A, Zucchelli M, Busato M, Manca V (2013) From time series to biological network regulations. *Molecular Biosystems* 9: 225-233.
6. Castellini A, Manca V, Zucchelli M (2014) An evolutionary procedure for inferring MP systems regulation functions of biological networks, *Natural Computing*.
7. Manca V, Marchetti L (2011) Log-Gain Stoichiometric Stepwise regression for MP systems. *Int J of Foundations of Computer Science* 22: 97-106.
8. Manca V, Marchetti L (2012) Solving dynamical inverse problems by means of Metabolic P systems. *BioSystems* 109: 78-86.
9. Manca V, Marchetti L (2013) An algebraic formulation of inverse problems in MP dynamics. *International Journal of Computer Mathematics* 90: 845-856.
10. Manca V (2013) *Infobiotics: Information in biotic systems*. Springer-Verlag.
11. Manca V, Marchetti L (2010) Goldbeter's Mitotic Oscillator Entirely Modeled by MP Systems. *LNCS* 6501: 273-284.
12. Manca V, Marchetti L, Pagliarini R (2011) MP modelling of glucose-insulin interactions in the Intravenous Glucose Tolerance Test. *International Journal of Natural Computing Research* 2: 13-24.
13. Marchetti L, Manca V (2012) A Methodology Based on MP Theory for Gene Expression Analysis. *Membrane Computing, LNCS* 7184: 300-313.
14. Rozenberg G, Salomaa A (1997) (eds): *Handbook of formal languages*. Springer-Verlag.
15. Paun G (2000) Computing with Membranes. *Journal of Computer and Systems Science* 61: 108-143.
16. Paun G (2002) *Membrane Computing-An Introduction*. Springer.
17. Paun G, Rozenberg G, Salomaa A (2010) *Oxford Handbook of Membrane Computing*. Oxford University Press.
18. Manca V (2010) From P to MP Systems. *Springer-Verlag* 5957: 74-94.
19. Manca V (2010) Metabolic P systems. *Scholarpedia*.
20. Manca V (2013) Algorithmic Models of Biochemical Dynamics: MP Grammars Synthesizing Complex Oscillators. *International Journal of Nanotechnology and Molecular Computation* 3: 24-37.
21. Marchetti L, Manca V, Pagliarini R, Bollig-Fischer A (2014) MP Modelling for Systems Biology: Two Case Studies. *Applications of Membrane Computing in Systems and Synthetic Biology* 7: 223-245.
22. Bollig-Fischer A, Marchetti L, Mitrea C, Wu J, Kruger A, et al. Modeling time-dependent transcription effects of HER2 oncogene and discovery of a role for E2F2 in breast cancer cell-matrix adhesion. *Bioinformatics*.
23. Fontana F, Manca V (2008) Predator-prey dynamics in P systems ruled by metabolic algorithm. *Bio Systems* 91: 545-557.
24. Manca V, Bianco L (2008) Biological networks in metabolic P systems. *BioSystems* 91: 489-498.
25. Draper N, Smith H (1981) *Applied Regression Analysis*, 3rd Edition. John Wiley and Sons, New York.
26. Hocking RR (1976) The Analysis and Selection of Variables in Linear Regression. *Biometrics* 32.
27. Toeplitz O (2007) *The Calculus: A Genetic Approach*. The University of Chicago Press.
28. Dunham W (1999) *Euler The Masters of Us All*. The Mathematical Association of America.
29. Luenberger DG (1979) *Introduction to Dynamic Systems*. John Wiley and Sons.
30. Shiraishi F, Takeuchi H, Hasegawa T, Nagasue H (2002) A Taylor series solution in Cartesian space to GMA- system equations and its application to initialvalue problems. *Applied Mathematics and Computation* 127: 103-123.