On Algebraic Identification of Causal Functionals

C. Hespel, G. Jacob,

LANS, INSA, 20 Avenue Buttes de Coësmes, 35 043 Rennes Cedex, France.
LIFL (URA 369 CNRS) Université Lille I, 59024 Villeneuve d'Ascq Cedex, France.

Abstract

We present here a second step in solving the *algebraic identification problem* for the causal analytic functionals in the sense of Fliess. These functionals are symbolically represented by non commutative formal power series $G = \sum_{w \in Z^\star} < G \mid w > w$, where w is a word on a finite encoding alphabet. The problem consists in computing the coefficients $\langle G \mid w \rangle$ from the choice of a finite set of informations on the input/output behaviour of the functional.

We have already presented a first step, in a previous work: we have shown that one can compute the contribution of G on a family of non commutative polynomials $\mathbf{g}_\mu$ with integer coefficients, indexed by the set of partitions.

These polynomials are linear combinations of the words w. We present here an algorithm, devoted to inverse this relations, by computing the words w as linear combinations of the $\mathbf{g}_\mu$. This very efficient algorithm is implemented in MAPLE. As example we present here a test set covering the identification of 2048 words. Since the algorithm is generic, parametrized by the length of the words, we conjecture that it actually solves the identification at any order.

Résumé

Nous proposons une deuxième étape dans la résolution du *problème de l'identification algébrique* des fonctionnelles causales au sens de Fliess. Une telle fonctionnelle est représentée par une série formelle non commutative $G = \sum_{w \in Z^\star} < G \mid w > w$, où w parcourt l'ensemble des mots sur un alphabet fini de codage. Il s'agit de calculer les coefficients $\langle G \mid w \rangle$ à partir d'un ensemble fini d'informations sur le comportement entrée-sortie de la fonctionnelle.

Nous avons déjà calcul, dans un précédent travail, les contributions de G relatives à une famille $\mathbf{g}_\mu p!$ de polynômes non commutatifs indexée par l'ensemble des partitions.

Ces polynômes sont combinaisons linéaires de mots w. Nous présentons ici un algorithme d'inversion, qui calcule les mots w comme combinaisons linéaires des $\mathbf{g}_\mu$. Cet algorithme très efficace est implémenté en MAPLE. Nous présentons en exemple un jeu d'essais permettant d'identifier 2048 mots. Cet algorithme est générique, paramétré par la longueur des mots. Nous conjecturons qu'il suffit à résoudre l'identification pour tous les mots w de toute longueur.

1 Introduction

Let $\mathbf{a}(t) = (a_j(t))_{j=1..m}$ be an input time function, and let $\mathcal{Z} = \{z_1 \cdots z_m\}$ be a finite encoding alphabet. Following Fliess [2, 3], we call *causal functional* any input/output relation encoded by a noncommutative generating series $\langle G \mid w \rangle$, where w is a word over $\mathcal{Z}$, the evaluation of which is obtained by replacing in G any word w by the corresponding Chen iterated integral over the input $\mathbf{a}(t)$ [1].

A natural question is then to decide if the generating series of a causal functional is a *canonical form.* We know three answers to that problem, given by Fliess (see [4, 6]), Reutenauer [10] and Sonntag and Wang [11]. Namely, they prove that any generating series encoding the null functional is equal to zero. But they do not give any effective computation.

Here we deal with a stronger property, designed as the *Algebraic Identification Problem:* can we compute iteratively the coefficients of a causal functional by some effective computation involving only some information panels on polynomial inputs, and on some corresponding jet values at zero of the output?

In a first work [6], we have computed from such a data panel, the *contributions* of G on a family of noncommutative polynomials $\mathbf{g}_\mu$ indexed by the partitions μ.

Here a give a careful analysis of the polynomial $\mathbf{g}_\mu$, in order to explain our recursive splitting algorithm, implemented in MAPLE.

2 Causal Functionals and Algebraic Identification

2.1 Encoding of causal functionals

Let us consider the n-dimensional dynamical system:

$$(\Sigma) \qquad \left\{ \begin{array}{rcl} \dot{q} & = & f_0(q) + \displaystyle\sum_{j=1..m} f_j(q) a_j(t) \\ y(t) & = & h(q(t)) \end{array} \right.$$

where $\mathbf{a}(t) = (a_j(t))_{j=1..m}$ is the m-dimensional input, $q(t) \in I\!\!R^n$ is the current state, and $y = h(q(t))$ is the output.

The vector field f_0 is called *the drift* of (Σ). It is currently studied by introducing a fictive input $a_0(t) \equiv 1$. A system without drift ($f_0 \equiv 0$) is also called *homogeneous in the inputs.*

We introduce a symbolic alphabet of m colors $\mathcal{Z} = \{z_1, z_2, \cdots, z_m\}$, and the free monoid $\mathcal{Z}^\star$ of words over $\mathcal{Z}$. A *noncommutative power series* S on $\mathcal{Z}$ is any map from $\mathcal{Z}^\star$ to $I\!\!R$, usually written as a formal sum $S = \sum_{w \in \mathcal{Z}^\star} \langle S \mid w \rangle \, w$. Then, starting at the state $q(0) = q_0$, the output of (Σ) can be obtained as an infinite sum

$$y(t) \;=\; \sum_{w \in Z^\star} \langle G_\Sigma \mid w \rangle \, \langle \mathcal{C}_{\mathbf{a}}(t) \mid w \rangle \qquad \text{where}$$

- $G_{\Sigma} = \sum_{w} \langle G_{\Sigma} \mid w \rangle w$ is the *generating series,* or *Fliess* series of the system [2]. It depends only on the geometry of the system, and is defined by the Lie derivative formulas:

$$\langle G_{\Sigma} \mid z_{i_1} z_{i_2} \cdots z_{i_k} \rangle \;=\; f_{i_1} \circ f_{i_2} \circ \cdots \circ f_{i_k} \circ h_{|_{q_0}}$$

- $\mathcal{C}_{\mathbf{a}}(t) = \sum_{w} \langle \mathcal{C}_{\mathbf{a}}(t) \mid w \rangle w$ is the *Chen series* of the input $\mathbf{a}(t)$ and is defined as the solution of the differential equation:

$$\frac{d}{dt}\mathcal{C}_{\mathbf{a}}(t) \;=\; \mathcal{C}_{\mathbf{a}}(t) \sum_{z_j \in Z} a_j(t)\, z_j \qquad \text{with} \quad C_{\mathbf{a}}(0) \;=\; 1$$

Thus the Chen series $\mathcal{C}_{\mathbf{a}}(t)$ appears as a symbolic encoding of the input $\mathbf{a}(t)$, and the Fliess series G_{Σ} plays *the same role as the transfer function* for a linear dynamical system.

We shall put afterwards $\mathcal{L}_{\mathbf{a}}(t) = \sum_{z_j \in Z} a_j(t) z_j$.

2.2 Identification problem

More generally, any noncommutative power series G, up to some convergence condition, can be seen as the symbolic encoding of a causal functional that, from an input $\mathbf{a}(t)$, produces the output given by the same summation:

$$y(t) \;=\; \sum_{w \in Z^{\star}} \langle G \mid w \rangle \, \langle \mathcal{C}_{\mathbf{a}}(t) \mid w \rangle$$

Now we can point out the following two problems:

- The **Identification Problem** consists in computing the coefficients $\langle G \mid w \rangle$ by using the knowledge (or measure) of a panel of input/output correspondences, between inputs and resulting outputs, in a neighbourhood of $t = 0$.
- The **Algebraic Identification Problem** is the same formulation, but we restrict input/output knowledge to panels of finite set of jets of the input, and finite set of jets of the resulting output, at time $t = 0$.

2.3 Our results

Our answer to the *Algebraic Identification Problem* is based on the construction of a family of non commutative polynomials $\mathbf{g}_{\mu}$ indexed by the set of the colored partitions μ. In case of a non input homogeneous system, we have to refine this indices in a family of colored partitions of the special form : $\mu = 1^p \otimes \nu$. The parts 3 and 4 of the paper are devoted to the combinatorial recursive structure of the polynomials $\mathbf{g}_{\mu}$.

In a previous paper, we have done **a first step** [6, 5] of the solution, by computing, from convenient data panel, each system *contribution* $\langle G \mid \mathbf{g}_{\mu} \rangle$ (or $\langle G \mid \mathbf{g}_{1^p \otimes \nu} \rangle$).

It remains in a **second step** to compute, from the contributions $\mathbf{g}_\mu$, all the coefficients $\langle G \mid w\rangle$. That could be reduced to compute the *inverse of the matrix* describing the polynomials $\mathbf{g}_\mu$ (or $\mathbf{g}_{1^p \otimes \nu}$) as linear combination of the words $w \in \mathcal{Z}^\star$.

The part 5 of this paper is devoted to the description of vector and matrix encoding. Then the part 6 presents the Maple package. This package is *generic, not bounded* (except by space and time computation). It has been able to solve the inversion problem for all words of length at most 11 on *two letters* (that is 2048 words), in *non homogeneous case.* **We conjecture that this algorithm actually solves the identification at any order**.

3 The output derivatives

The n^{th} time derivative of the output y is given by:

$$\frac{d^n}{dt^n} y(t) \;=\; \langle G \,||\, \frac{d^n}{dt^n}\mathcal{C}_{\mathbf{a}}(t)\rangle \;=\; \sum_{w\in\mathcal{Z}^\star} \langle G \mid w\rangle \langle \frac{d^n}{dt^n}\mathcal{C}_{\mathbf{a}}(t) \mid w\rangle$$

We obtain, by a straightforward computation:

$$\frac{d^n}{dt^n}\mathcal{C}_{\mathbf{a}}(t) \;=\; \mathcal{C}_{\mathbf{a}}(t)\;\mathbf{A}_n(t)$$

where the $\mathbf{A}_n$ are the noncommutative polynomials recursively defined as follows:

$$\mathbf{A}_{n+1}(t) \;=\; \mathcal{L}_{\mathbf{a}}(t)\mathbf{A}_n(t) + \frac{d}{dt}\mathbf{A}_n(t) \qquad \text{with} \quad \mathbf{A}_0(t) \;=\; 1$$

For instance:

$$\begin{aligned}\mathbf{A}_2 \;&=\; \mathcal{L}_{\mathbf{a}}\mathcal{L}_{\mathbf{a}} + \frac{d}{dt}\mathcal{L}_{\mathbf{a}} \;=\; \sum_{i,j} a_i a_j \,.\, z_i z_j + \sum_j \dot{a}_j \; z_j \\ &=\; \sum_{i<j} a_i a_j \,.\, (z_i z_j + z_j z_i) + \sum_j a_j^2 \,.\, z_j^2 + \sum_j \dot{a}_j \; z_j\end{aligned}$$

3.1 The generic equation

By setting the infinite sum $\mathcal{A} = \sum_{n\in\mathbb{N}} \mathbf{A}_n$, we get the following generic *equation*:

$$\mathcal{A} \;=\; \varepsilon + \mathcal{L}_{\mathbf{a}}\mathcal{A} + \frac{d}{dt}\mathcal{A} \qquad \text{where} \quad \mathcal{L}_{\mathbf{a}} \;=\; \sum_{z_j\in Z} a_j z_j$$

4 Differential algebra and colored partitions

We analyse now these equations in the light of the free differential calculus. Indeed, considering now the inputs a_k specialized in time $t = 0$, as differential letters, it is clear that each $\mathbf{A}_n$ is a noncommutative polynomial in $\mathcal{Z}$, the coefficients of which are (commutative) differential monomials in the a_k.

4.1 Partitions and multiplicities

For any finite sequence of strictly positive integers $\rho = (\rho_1, \rho_2, \cdots \rho_n)$, that we call here *"multi-index"*, we note ρ^+ the sequence obtained by reordering it in increasing order. We call *multiplicity of* ρ the usual notation of ρ^+ as a partition. For example:

$$\text{if} \quad \rho = (3,2,6,2,1,3,2) \qquad \text{then} \quad \rho^+ = 1222336 \qquad \text{and} \quad \mu(\rho) = 1^1 2^3 3^2 6^1$$

On a single letter $a_1 = a$, the *differential monomials* become:

$$a^{\mu} \;=\; (a^{(i_1-1)})^{e_1} (a^{(i_2-1)})^{e_2} \ldots (a^{(i_q-1)})^{e_q}, \qquad 1 \leq i_1 < i_2 < \ldots < i_q$$

Such a monomial μ is indexed by the following partition ([9]):

$$\mu = \left(i_1^{e_1} i_2^{e_2} \ldots i_q^{e_q}\right), \qquad 1 \leq i_1 < i_2 < \ldots < i_q$$

The *weight* and the *size* of μ are defined as usually as follows:

$$\mathrm{wgt}(\mu) = \sum_k e_k . i_k \qquad\qquad \mathrm{size}(\mu) = \sum_k e_k$$

The empty partition is noted ε.

We define now a **derivation law D on partitions**, in order to reflect the effect of time derivative $\dfrac{d}{dt}$ on differential monomials.

- For a differential letter: $\dfrac{d}{dt}(a^{(i_k)}) = a^{(i_{k+1})}$, and then: $D(i_k) = i_{k+1}$
- We extend D to any partition (as for differential monomials) by the Leibnitz derivation rule and commutative reordering (the result being a linear combination of partitions):

$$\begin{aligned} D\left(i_1^{e_1} i_2^{e_2} \ldots i_q^{e_q}\right) &= \sum_{k=1..q} \mathbf{e_k} \star \left(i_1^{e_1} \cdots i_k^{e_k-1} i_{k+1}^{e_{k+1}+1} \ldots i_q^{e_q}\right)^+ \\ D\mu &= \sum_{k=1..q} \mathbf{e_k} \star \left[i_{k+1} \frac{\partial}{\partial i_k}(\mu)\right]^+ \end{aligned}$$

Thus $D\mu$ is a linear combinations of partitions that cover μ in the Young lattice (i.e. for dominance ordering [9]). For example, we have:

$$D(1^1 2^3 3^2 6^1) \;=\; 2^4 3^2 6^1 + \mathbf{3} \star 1^1 2^2 3^3 6^1 + \mathbf{2} \star 1^1 2^3 3^1 4^1 6^1 + 1^1 2^3 3^2 7^1$$

We obtain so a linear equation, with adequate definition of the coefficients $<\!\mu|\tau\!>$:

$$D\mu \;=\; \sum_{\tau \in \text{partitions}} <\!\mu|\tau\!> \tau \qquad\qquad \text{with } <\!\mu|\tau\!> \in I\!N$$

Remark *If* $<\!\mu|\tau\!> \neq 0$*, then* $size(\mu) = size(\tau)$*. In other words, the operator* D *preserve the size.*

4.2 Colored partitions and multiplicities

Let $\mathcal{Z} = \{z_1, z_2, \cdots, z_m\}$ be a finite *alphabet of colors.* We call *colored multi-index* any finite sequence of "colored strictly positive integers" $\rho = (\rho_1 z_{i_1}, \rho_2 z_{i_2}, \cdots, \rho_m z_{i_m})$ (for any k, ρ_k is a strictly positive integer, and z_{i_k} belongs to the alphabet of colors).

The *word of colors* of ρ is the word $w = w(\rho) = z_{i_1} z_{i_2} \cdots z_{i_m}$. The sequence of integers occuring in ρ for the color z_j, after increasing reordering, forms a partition $\pi_j(\rho)$. We define *the multiplicity* of ρ as being the family of the $\pi_j(\rho)$ for all the colors z_j.

Such a family μ of partitions indexed by colors is called a *colored partition.* In order to deal with linear combinations of colored partitions, it is convenient to adopt the tensor product notation:

$$\mu = \mu_1 \otimes \mu_2 \otimes \cdots \otimes \mu_m$$

where the partition μ_j is the component of μ for color z_j. So, on the set of differential letters $\{a_j\}_{j=1..m}$, the colored partition μ will denote the *differential monomial*

$$a^{\mu} = a^{\mu_1 \otimes \mu_2 \otimes \cdots \otimes \mu_m} = a_1^{\mu_1} a_2^{\mu_2} \cdots a_m^{\mu_m}$$

For each color z_j, we denote by a_j the colored partition

$$a_j = \varepsilon^{\otimes(j-1)} \otimes 1 \otimes \varepsilon^{\otimes(m-j)}$$

The **derivation law** D **on colored partitions**, reflecting the time derivative $\dfrac{d}{dt}$ of differential monomials on m differential letters, is given by multilinearly extending the law D defined on each color component:

$$D(\mu_1 \otimes \mu_2 \otimes \cdots \otimes \mu_m) \quad = \quad \sum_{j=1..m} \mu_1 \otimes \cdots \otimes \mu_{j-1} \otimes D(\mu_j) \otimes \mu_{j+1} \otimes \cdots \otimes \mu_m$$

In terms of linear algebra, we get so for any colored partition μ the (sparse) linear equation:

$$D\mu \quad = \sum_{\tau \in \text{colored partitions}} <\mu|\tau> \star \tau \qquad \text{with } <\mu|\tau> \in I\!N$$

4.3 Combinatorial analysis of the generic equation

Let us now interpret combinatorially the series $\mathcal{A}$, by identifying each differential monomial with its colored multiplicity. By factorizing this series according to the colored partitions, it can be rewritten in the form $\mathcal{A} = \sum_{\mu} \mu \,.\, \mathbf{g}_{\mu}$. Then the generic equation becomes:

$$\sum_{\mu} \mu \,.\, \mathbf{g}_{\mu} \quad = \quad \varepsilon + \sum_{j=1}^{m} \sum_{\sigma} a_j \sigma \,.\, z_j \mathbf{g}_{\sigma} + \sum_{\nu} D(\nu) \,.\, \mathbf{g}_{\nu}$$

By identifying in the two members the factors of the same non trivial colored partition $\mu \neq \varepsilon$, we obtain:

$$\mathbf{g}_{\mu} \quad = \quad \sum_{j=1}^{m} z_j \mathbf{g}_{\mu \triangleright a_j} + \sum_{\nu} <\nu|\mu> \star \mathbf{g}_{\nu} \qquad \text{with} <\mu|\tau> \in I\!N$$

where we the symbol $\triangleright$ is used as follows:

$$\mu \triangleright a_j \quad = \quad \begin{cases} \sigma & \text{if } j\sigma = \mu \\ 0 & \text{in other cases} \end{cases}$$

The coefficients $<\nu|\mu>$ are related to the *primitives* of the colored partition μ.

4.4 Non homogeneous colored multiplicities

We specify here the problem in case of *non input homogeneous* sytems with *one controlled input*:

$$(\Sigma) \qquad \begin{cases} \dot{q} & = & f_0(q) + f_1(q)a_1(t) \\ y(t) & = & h(q(t)) \end{cases}$$

Then we must set $a_0(t) \equiv 1$, hence the 2-colored mutiplicity of weight n must be written as:

$$\mu = \mu_0 \otimes \nu \qquad \text{where} \quad \begin{cases} \mu_0 & = & 1^p \\ \text{wgt}(\mu) & = & p + \text{wgt}(\nu) \end{cases}$$

The derivation law must be adapted. Indeed, we have to set $D(\mu_0) = 0$, and then:

$$D(1^p \otimes \nu) \quad = \quad 1^p \otimes D(\nu)$$

The integer p is called *the depth* of the colored partition $1^p \otimes \nu$. In addition we have:

$$\text{wgt}(1^p \otimes \nu) = p + \text{wgt}(\nu) \qquad \text{and} \quad \text{size}(1^p \otimes \nu) = p + \text{size}(\nu)$$

Then the recursive equations become:

$$\mathbf{g}_{1^p \otimes \nu} \quad = \quad z_0 \mathbf{g}_{1^{p-1} \otimes \nu} + z_1 \mathbf{g}_{1^p \otimes \nu \triangleright 1} + \sum_{\sigma} <\sigma|\nu> \star \mathbf{g}_{1^p \otimes \sigma}$$

In the right hand side, the first term vanishes if $p = 0$. The second term vanishes if the partition ν has no part equal to 1. The third sum, extended to the "primitives" σ of ν, is finite.

5 The algorithm

Let us rewrite the recursive equations as follows:

$$\mathbf{g}_{1^p \otimes \nu} - \sum_{\sigma} <\sigma|\nu> \star \mathbf{g}_{1^p \otimes \sigma} \quad = \quad z_0 \mathbf{g}_{1^{p-1} \otimes \nu} + z_1 \mathbf{g}_{1^p \otimes \nu \triangleright 1}$$

Then the left member is a linear combinations of partitions of the same size $n = p + \text{size}(\nu)$. The right member is a linear combinations of partitions of size $n - 1$. We shall use this fact to put the partitions occurring in this left member into a triangular set.

5.1 The matrix encoding

5.1.1 Partitions and words ordering

For any integers m and k, we note $\mathcal{M}_m^k$ the set of partitions that can be written in the form:

$$j_1 j_2 \cdots j_m \quad \text{with} \quad 1 \leq j_1 \leq j_2 \leq j_3 \leq \ldots \leq j_m \leq k$$

ordered by the lexicographical ordering.

We denote by $Z^{m,k}$ the set of words on $\mathcal{Z}$ of length m in z_1 and $k-1$ on z_0, ordered by lexicographical ordering *according to the convention* $z_1 < z_0$. Then the application defined by

$$j_1 j_2 \cdots j_m \quad \mapsto \quad z_0^{j_1-1} z_1 z_0^{j_2-j_1} z_1 z_0^{j_3-j_2} \ldots z_1 z_0^{j_m-j_{m-1}} z_1 z_0^{k-j_m}$$

is a ordered bijection of $\mathcal{M}_m^k$ on $Z^{m,k}$. Their common cardinality is equal to $\binom{m+k-1}{m}$.

5.1.2 Vector encoding

Let k be a fixed positive integer. For any integers m, p we note $\mathbf{G}_{/p}^{m}(k)$ the column vector of all polynomials $\mathbf{g}_{1^p \otimes \nu}$ for $\nu \in \mathcal{M}_m^k$, presented in lexicographical ordering. We obtain so a vector equation:

$$(Id - \mathcal{P}_m^k) \circ \mathbf{G}_{/p}^{m}(k) \quad = \quad z_1 \, ext(\mathbf{G}_{/p}^{m-1}(k), 0) + z_0 \, \mathbf{G}_{/p-1}^{m}(k)$$

where $ext(\mathbf{G}_{/p}^{m-1}(k), 0)$ denotes the column vector $\mathbf{G}_{/p}^{m-1}(k)$ (of dimension $\binom{m+k-2}{m-1}$) completed by 0 polynomials up to dimension $\binom{m+k-1}{m}$. The transformation $(Id - \mathcal{P}_m^k)$ is represented by a matrix $\mathbf{J}_m^k$ sparse and lower triangular, with 1 only on the main diagonal, where appear only some primitive coefficients. We note $\mathbf{T}_m^k$ the inverse matrix of $\mathbf{J}_m^k$.

5.1.3 Expansion of $\mathbf{G}_{/p}^{m}(k)$ on the basis $Z^{m,k}$

We compute here the vector $\mathbf{G}_{/p}^{m}(k)$ on the basis $Z^{m,k}$. In the right member of the previous equation, the splitting sum with the prefixes z_1 and z_0 must then be interpreted as horizontal concatenation of sub-blocks (that we note here by $/\!\!/$). We obtain so:

$$\begin{aligned}
\mathbf{G}_{/p}^{m}(p) &= \mathbf{T}_m^p \left[\begin{pmatrix} \mathbf{G}_{/p}^{m-1}(p) \\ 0 \end{pmatrix} /\!\!/ \mathbf{G}_{/p-1}^{m}(p) \right] \\
\mathbf{G}_{/p-1}^{m}(p) &= \mathbf{T}_m^p \left[\begin{pmatrix} \mathbf{G}_{/p-1}^{m-1}(p) \\ 0 \end{pmatrix} /\!\!/ \mathbf{G}_{/p-2}^{m}(p) \right] \\
\cdots \quad \mathbf{G}_{/1}^{m}(p) &= \mathbf{T}_m^p \left[\begin{pmatrix} \mathbf{G}_{/1}^{m-1}(p) \\ 0 \end{pmatrix} /\!\!/ \mathbf{G}_{/0}^{m}(p) \right] \\
\mathbf{G}_{/0}^{m}(p) &= \mathbf{T}_m^p \begin{pmatrix} \mathbf{G}_{/0}^{m-1}(p) \\ 0 \end{pmatrix} \qquad \cdots \qquad \mathbf{G}_{/0}^{1}(p) = \mathbf{T}_1^p \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}
\end{aligned}$$

5.2 Splitting analysis

5.2.1 The first splitting

The *first splitting* is obtained by computing $\mathbf{J}_m^p \mathbf{G}_{/p}^m(p)$, which has upper block triangular form, the first block of which is $\mathbf{G}_{/p}^{m-1}(p)$.

The remaining second square diagonal block is a restriction $\mathbf{G}_{/p-1}^m(p)$ to its lower part. We shall now describe this lower part, and continue recursively its splitting.

5.2.2 The recursive splitting

For any integers m, and $1 \leq j \leq k$, we note $\mathcal{M}_m^{[j,k]}$ the set of partitions that can be written in the form:

$$i_1^{j_1} i_2^{j_2} \cdots i_m^{j_m} \quad \text{with} \quad j \leq j_1 \leq j_2 \leq j_3 \leq \ldots \leq j_m \leq k$$

The restriction of $\mathbf{J}_m^k$ to the partitions of $\mathcal{M}_m^{[j,k]}$ will be noted $\mathbf{J}_m^k{}_{[j,k]}$. The diagonal block of $\mathbf{G}_{/p}^m(k)$ restricted to the lines of $\mathcal{M}_m^{[j,k]}$ will be noted $\mathbf{G}_{/p}^m{}_{[j,k]}$. This allows to split recursively:

$$\begin{aligned}
\mathbf{J}_m^p{}_{[1,p]}\, \mathbf{G}_{/p-1}^m{}_{[1,p]} &= \left[\begin{pmatrix} \mathbf{G}_{/p-1}^{m-1}{}_{[1,p]} \\ 0 \end{pmatrix} /\!\!/ \mathbf{G}_{/p-2}^m{}_{[1,p]} \right] \\
\mathbf{J}_m^p{}_{[2,p]}\, \mathbf{G}_{/p-2}^m{}_{[2,p]} &= \left[\begin{pmatrix} \mathbf{G}_{/p-2}^{m-1}{}_{[2,p]} \\ 0 \end{pmatrix} /\!\!/ \mathbf{G}_{/p-3}^m{}_{[2,p]} \right] \qquad \cdots
\end{aligned}$$

We deduce, in particular:

Lemma 5.1 *The determinant of $\mathbf{G}_{/p}^m(p)$ is the product of the determinant of $\mathbf{G}_{/p-j}^{m-1}$ and of the determinants of the matrices $\mathbf{G}_{/p-j}^{m-1}{}_{[j,p]}$ for $j = 1 \cdots p$.*

6 The package

The package is very concise. It uses intensively the recursive block structure of the matrices $\mathbf{G}_{/p}^m$, The matrices of primitive coefficients $\mathbf{J}_m^p$ are sparse, with a recursive block triangular form. A combinatorial construction of a sequence extraction procedure allows to an easy programming in MAPLE. For example, the "choice criteria" at *level 3* , are sequences extracted from the sequence $\mathsf{criterium}(4, 0, 3)$ (the / and // are only written to explain the structure, and the $*$ for lacking elements):

$$\begin{aligned}
\mathsf{criterium}(4,0,3) &= [1,2,3,4/5,6,7/8,9/10/\!\!/11,12,13/14,15/16/\!\!/17,18/19/\!\!/20] \\
\mathsf{criterium}(4,1,3) &= [1,2,3,*/5,6,*/8,*/*/\!\!/11,12,*/14,*/*/\!\!/17,*/*/\!\!/*] \\
&= [1,2,3,5,6,8,11,12,14,17] \\
\mathsf{criterium}(4,2,3) &= [1,2,*,*/5,*,*/*,*/*/\!\!/11,*,*/*,*/*/\!\!/*,*/*/\!\!/*] \;=\; [1,2,5,11]
\end{aligned}$$

For computing the matrices $\mathbf{T}_m^p$, we avoid any Gauss elimination. We treat all inversions by a recursive call of products, additions, and concatenation of sub-blocks.

The matrix $\mathbf{G}^{m}_{/p}(p)$ is computed by the call $\mathbf{A}(p,m)$. Its analysis and prime factor decomposition is obtained by the call $\mathsf{analysis}(\mathsf{p},\mathsf{m})$.

We join as examples some results returned by the analysis procedure, with computation time in seconds, on a PowerMac G3 at 266 Mhz. We have been able to complete the analysis for all words of length at most 11 over two letters, that represents the identification of 2048 words, starting from polynomials $\mathbf{g}_{1\otimes\nu}$ with up to 16 decimal digits integers coefficients.

6.1 Conclusion

The package is generic, and it allows to decide for any any word w (except for lack of time or memory size) if it is a linear combination of the polynomials $\mathbf{g}_{\mu}$.

We conjecture that it will never return *zero* to any call of the procedure $\mathsf{analysis}(\mathsf{m},\mathsf{k})$. Hence we conjecture that the Algebraic Identification Problem is completely solved by this algorithm.

References

[1] K.T. Chen, Iterated path integrals , *Bull. Amer. Math. Soc.*, vol.83 (1977) 831-879.

[2] M. Fliess, Fonctionnelles causales non linéaires et indéterminées non commutatives, *Bull. Soc. Math. France* 109 (1981) 3-40.

[3] M. Fliess, M. Lamnabhi, F. Lamnabhi-Lagarrigue, An algebraic approach to nonlinear functional expansions, *IEEE Trans. Circuits and Systems*, vol. CAS-30, n°8 (1983) 554-570.

[4] M. Fliess, On the concept of derivatives and Taylor expansions for nonlinear input/output systems, *Proceedings of the 22nd IEEE Conf. Decision and Control, 1983, San Antonio, Texas, pp 643-646.*

[5] C. Hespel, Iterated derivatives of a non linear dynamic system and Faà di Bruno formula, *Mathematics and Computers in Simulation*, vol. 42, pp 641-657, 1996.

[6] C. Hespel, G. Jacob, First steps towards Exact Algebraic Identification, *Discrete Math.,* vol. 180,pp. 211-210, 1998.

[7] G. Jacob, Algebraic methods and computer algebra for nonlinear systems' study, in: IMACS Symposium MCTS, *Modelling and Control of Technochological systems*, vol.2 (lille, 1991) 599-608.

[8] F. Lamnabhi-Lagarrigue, P.E. Crouch, A formula for iterated derivatives along trajectories of nonlinear systems, *Systems and Control letters* 11 (1988) 1-7.

[9] I.G. Macdonald, *Symmetric Functions and Hall Polynomials*, 2d ed., Oxford Science Publications, 1995.

[10] C. Reutenauer, *private communication.*

[11] Y. Wang, E.D. Sontag, On two definitions of observation spaces, *Systems and Control letters* 13 (1989) 279-289.

Annexe The call **a**$nalysis(m,k)$ gives the prime decompositions of determinant of diagonal blocks excepted the first, which is equal to 1.

> analyse(10,2); # dim(A(10,2))=55.

$$primes, \quad (19)$$
$$primes, \quad (3)\ (17)\ (19)$$
$$primes, \quad (2)^2\ (3)\ (17)(19)$$
$$primes, \quad (2)^4\ (3)\ (13)\ (17)^2\ (19)$$
$$primes, \quad (2)^3\ (11)\ (13)^2\ (17)^2\ (19)$$
$$primes, \quad (2)^2\ (11)\ (13)^3\ (17)^2\ (19)$$
$$primes, \quad (2)^2\ (11)\ (13)^3\ (17)^2\ (19)$$
$$primes, \quad (2)^2\ (11)\ (13)^2\ (17)^2\ (19)$$
$$primes, \quad (2)\ (11)\ (13)\ (17)\ (19)$$
$$temps\ local =,\ 9.000$$

> analyse(10,3)
dim(A(10,3))=220 (maximal decimal size of integers: 12 digits)

$$primes, \quad (2)^{15}\ (3)^8\ (5)^3\ (7)^2\ (11)^6\ (13)^{13}\ (17)^{14}\ (19)^9\ (23)\ (29)$$
$$primes, \quad (2)^{20}\ (3)^8\ (5)^8\ (7)^4\ (11)^8\ (13)^{14}\ (17)^{13}\ (19)^8\ (23)^3\ (29)$$
$$primes \quad (2)^{17}\ (3)^{10}\ (5)^7\ (7)^6\ (11)^{10}\ (13)^{14}\ (17)^{12}\ (19)^8\ (23)^6\ (29)$$
$$primes, \quad (2)^{19}\ (3)^{14}\ (5)^6\ (7)^3\ (11)^{12}\ (13)^{14}\ (17)^{10}\ (19)^8\ (23)^7\ (29)$$
$$primes, \quad (2)^{12}\ (3)^8\ (5)^7\ (7)\ (11)^{13}\ (13)^{13}\ (17)^9\ (19)^8\ (23)^8\ (29)$$
$$primes, \quad (2)^8\ (3)^6\ (5)^7\ (11)^{11}\ (13)^{14}\ (17)^6\ (19)^6\ (23)^7\ (29)$$
$$primes, \quad (2)^3\ (3)^5\ (5)^7\ (11)^8\ (13)^{10}\ (17)^4\ (19)^4\ (23)^6\ (29)$$
$$primes, \quad (2)^3\ (3)^2\ (5)^4\ (7)\ (11)^5\ (13)^6\ (17)^3\ (19)^2\ (23)^3\ (29)$$
$$primes, \quad (2)\ (3)\ (5)\ (7)\ (11)^2\ (13)^2\ (17)\ (19)\ (23)\ (29)$$
$$temps\ local =,\ 236.000$$

analysis(8,4)
# dim(A(10,3))=330: (maximal decimal size of integers: 16 digits).

$$primes,\ (2)^{21}\ (3)^{74}\ (5)^{32}\ (7)^{24}\ (11)^{55}\ (13)^{44}\ (17)^{16}\ (19)^{23}\ (23)^{10}\ (29)^2\ (31)$$
$$primes,\ (2)^{35}\ (3)^{58}\ (5)^{38}\ (7)^{25}\ (11)^{48}\ (13)^{36}\ (17)^{13}\ (19)^{21}\ (23)^{13}\ (29)^3\ (31)$$
$$primes,\ (2)^{16}\ (3)^{54}\ (5)^{26}\ (7)^{24}\ (11)^{39}\ (13)^{30}\ (17)^{11}\ (19)^{18}\ (23)^{14}\ (29)^4\ (31)$$
$$primes,\ (2)^{22}\ (3)^{49}\ (5)^{17}\ (7)^{15}\ (11)^{26}\ (13)^{22}\ (17)^8\ (19)^{13}\ (23)^{11}\ (29)^4\ (31)$$
$$primes,\ (2)^{13}\ (3)^{29}\ (5)^{12}\ (7)^5\ (11)^{17}\ (13)^{14}\ (17)^7\ (19)^8\ (23)^7\ (29)^4\ (31)$$
$$primes,\ (2)^7\ (3)^{17}\ (5)^7\ (11)^7\ (13)^8\ (17)^3\ (19)^4\ (23)^3\ (29)^3\ (31)$$
$$primes,\ (3)^5\ (5)^3\ (11)^2\ (13)^2\ (17)\ (19)\ (23)\ (29)\ (31)$$

The package "IDENTALG" (a shirt MAPLE file) can be obtained by email address: jacob@lifl.fr