

Remerciements

C'est pour moi l'occasion de remercier tous mes collègues mathématiciens, enseignants et secrétaires du laboratoire LANS, pour avoir continué à m'accueillir au laboratoire après ma reconversion vers l'informatique.

C'est aussi l'occasion d'exprimer ma reconnaissance aux enseignants, secrétaires, techniciens et ingénieurs du département Informatique de l'INSA et de l'IFSIC, qui m'ont tendu la main malgré les tensions, et à ceux qui, discrètement, patiemment, m'ont appris ce que je sais en informatique.

Je remercie aussi tous les membres du laboratoire LIFL, car malgré l'éloignement, ils m'ont accueillie comme une des leurs, à part entière, et m'ont permis de me déplacer quand le laboratoire LANS ne pouvait plus financer tous mes déplacements.

Dans ce mémoire, se mêlent problèmes de mathématiques, d'informatique, d'automatique. Je suis très honorée que soient réunis dans ce jury des chercheurs de ces disciplines, parmi les plus prestigieux. Ils ont, par leur rayonnement, largement marqué ce mémoire ou orienté mes recherches:

Sophie Tison que je remercie pour avoir accepté de présider le jury

Michel Fliess et Xavier Viennot que je remercie pour avoir accepté la lourde tâche de rapporter sur ce mémoire

Albert Benveniste, Geneviève Dauphin-Tanguy, Philippe Flajolet et Jean Pellaumail que je remercie pour avoir accepté d'être examinateurs.

J'exprime enfin ma profonde gratitude à Gérard Jacob qui a su me communiquer son enthousiasme, m'a permis d'acquérir une parcelle de sa créativité et m'a transmis généreusement les éléments de sa connaissance.

Table des matières

Présentation du mémoire	6
1 Monoïdes et séries formelles	17
1.1 Introduction	17
1.2 Application bilinéaire et produit tensoriel	18
1.2.1 Algèbre d'un ensemble, coupes	18
1.2.2 Lemme 1.2.1, Proposition 1.2.1	19
1.3 Monoïdes et Observations	22
1.3.1 Les présentations	22
1.3.2 Le théorème de Kleene	23
1.4 Séries formelles	24
1.4.1 Définitions et notations	24
1.4.2 Théorème de présentation minimale	26
1.4.3 Quelles structures pour $\mathbb{R}\langle\langle Z \rangle\rangle$?	27
1.4.4 Topologie sur $\mathbb{R}\langle\langle Z \rangle\rangle$	32
2 Séries rationnelles	36
2.1 Introduction	36
2.2 Séries rationnelles en une variable	37
2.3 Séries rationnelles non commutatives	37
2.3.1 Opérations rationnelles	37
2.4 Séries reconnaissables	39
2.4.1 Matrices de Hankel	39
2.4.2 Automates finis	40
2.5 Théorème de Schützenberger	48

2.5.1	Représentations linéaires	48
2.5.2	Systèmes d'équations linéaires à droite	51
2.5.3	Expressions rationnelles	51
2.5.4	Le théorème de Schützenberger	52
2.6	Algèbre syntaxique	53
3	Approximation par une série rationnelle	55
3.1	Introduction	55
3.2	Cas d'une seule variable	56
3.2.1	Par un polynôme	56
3.2.2	Par un approximant de Padé ou de type Padé	56
3.2.3	Notion de fraction continue	59
3.3	Cas de plusieurs variables non commutatives	60
3.3.1	Par une série rationnelle g_k obtenue à partir de la matrice de Hankel de s	60
3.3.2	Par une série rationnelle f_k obtenue à partir de l'automate associé aux résiduelles de s	62
3.3.3	Relation avec les approximants de type Padé et les fractions continues	63
3.3.4	Que se passe-t-il si la série s est rationnelle?	68
3.3.5	Comparaison des deux approximants proposés	69
3.3.6	Application aux systèmes dynamiques	70
4	Systèmes dynamiques et séries formelles	72
4.1	Introduction	72
4.2	Rappels	73
4.2.1	Algèbre de Lie	73
4.2.2	Systèmes dynamiques considérés	74
4.3	Deux séries formelles remarquables	75
4.3.1	La série génératrice	75
4.3.2	La série de Chen	76
4.4	Série produite différentiellement	76
5	Systèmes bilinéaires et séries rationnelles	79

5.1	Introduction	79
5.1.1	Rappels et notations	79
5.2	Représentation minimale et système réduit	80
6	Bilinéarisations de systèmes dynamiques	83
6.1	Introduction	83
6.2	Enoncé du problème	83
6.3	Bilinéarisation de rang minimal	84
6.4	Approximation par automates tronqués	84
6.4.1	Bilinéarisation par l'automate structurel	85
6.4.2	Bilinéarisation par l'automate géométrique	88
6.5	Relation avec les approximants de type Padé	91
6.6	Que se passe-t-il si (Σ) est bilinéaire?	91
6.7	Comparaison des approximants bilinéaires	92
7	Identification algébrique: des voies	95
7.1	Introduction	95
7.2	Tour d'horizon sur les modélisations de boîtes noires	96
7.2.1	Structures des modèles	96
7.2.2	Fondements mathématiques	97
7.3	Identification de paramètres	98
7.3.1	Selon un modèle choisi à priori, dépendant d'un nombre fini de paramètres	98
7.3.2	Selon un modèle adapté, dépendant d'un nombre fini de paramètres	98
7.3.3	Selon une approche neuronale	101
7.4	Enoncé du problème	103
7.4.1	Rappels	103
7.5	Identification par des techniques d'observations	105
7.5.1	Rappels et notations	105
7.5.2	Identification par la méthode du "motion planning"	106
7.5.3	Identification par les fonctionnelles différentiables	108
7.6	Identification par la série génératrice	111
7.6.1	Une solution par les preuves sur l'indiscernabilité?	112

8	Identification algébrique: premier pas	115
8.1	Introduction	115
8.2	Calcul des dérivées itérées de la sortie	115
8.2.1	Dérivées des séries de Chen	115
8.2.2	Calcul explicite des A_i	116
8.2.3	Expression des A_n en les multidérivées des entrées	118
8.3	Un algorithme pour la première étape	120
8.3.1	Cas d'une entrée sans dérive	120
8.3.2	Cas d'une entrée avec dérive	121
8.4	Interprétation combinatoire	122
8.4.1	Les partitions colorées	122
8.4.2	Algèbre différentielle et partitions colorées	126
9	Algorithme d'identifiabilité jusqu'au degré l	130
9.1	Analyse du problème de scindage	131
9.1.1	Identification des coefficients de la série génératrice dans le cas d'une entrée sans dérive	131
9.1.2	Identification des coefficients de la série génératrice dans le cas d'une entrée avec dérive	131
9.2	Algorithme de scindage	152
9.2.1	Calcul sans dérive	152
9.2.2	Une entrée avec dérive	152
9.2.3	L'algorithme	153
9.3	Conclusion et perspectives	154
	Annexe	156

Présentation du mémoire

Introduction

Le point de départ de ce mémoire

Le point de départ est né de la question suivante posée par G.Jacob:

Peut-on proposer une approximation de série formelle non commutative, par des séries rationnelles, qui soit meilleure que celle offerte par des polynômes?

cette question étant soulevée par le théorème de M.Fliess:

Théorème 1 *Toute fonctionnelle causale analytique peut être uniformément approchée par des fonctionnelles régulières (i.e. dont la série génératrice est rationnelle).*

Par “fonctionnelle causale”, on veut dire que la sortie $y(t)$ du système décrit ne dépend que des valeurs antérieures des entrées $a_i(\tau)$ c’est à dire pour $\tau \leq t$.

Avant de décrire le contenu de ce mémoire, nous rappelons dans quel cadre se situe le sujet.

Eléments sur l’histoire des séries formelles

La notion de série formelle en variables non commutatives a été introduite par M.P.Schützenberger [32], en liaison avec les automates et les langages formels.

Deux principales familles de séries formelles ont été étudiées (voir Salomaa-Soittola [31]): **Les séries rationnelles [33], leur sous famille formée des séries reconnaissables [10] et les séries algébriques [8].**

M.P.Schützenberger a introduit les séries rationnelles, et montré que certaines propriétés des séries rationnelles en une variable, trouvent leur bonne généralisation en variables non commutatives. Il généralise d’abord le théorème de Kleene, établissant

l'équivalence des langages reconnaissables et rationnels, en démontrant l'équivalence de la reconnaissabilité et de la rationalité des séries formelles propres. Il montre alors que les représentations matricielles du monoïde libre jouent un rôle fondamental pour les séries formelles.

M.Fliess caractérise les séries rationnelles par leur matrice de Hankel [11]. G.Jacob généralise la notion de boucle dans les automates finis, grâce aux représentations matricielles [21, 20]. C.Reutenauer caractérise les séries rationnelles par leur algèbre syntaxique [27].

La famille des séries algébriques a été elle aussi introduite par M.P.Schützenberger [32]: les séries algébriques sont les composantes de la solution d'un système d'équations algébriques propres. M.Nivat montre que l'étude des séries algébriques dépend de façon fondamentale de l'étude des transductions rationnelles de séries formelles [25]. En établissant la réciproque du théorème de Shamir, G.Jacob fournit une nouvelle caractérisation des séries algébriques [19].

Plus récemment, s'est développée une théorie combinatoire des séries formelles: une représentation combinatoire des séries formelles est proposée sous forme "d'espèces de structures" par Joyal [22] et "d'arbres pondérés" par P.Leroux et X.G.Viennot [23].

Quelques applications des séries formelles

Depuis longtemps, de nombreux problèmes rencontrés en théorie des langages, ont recours aux séries formelles. Citons les problèmes arithmétiques de la théorie des langages [3], l'étude des processus stochastiques [12], l'étude des grammaires hors contexte [8, 7].

Les séries formelles représentent un outil intéressant pour la résolution de problèmes combinatoires. Citons l'énumération des graphes planaires [9], les permutations et réarrangements dans les monoïdes [6].

Enfin, l'étude des séries formelles en variables non commutatives est aussi en lien étroit avec l'algèbre différentielle: une autre application des séries formelles se trouve dans le traitement des systèmes dynamiques. M.Fliess développe l'idée que la série génératrice d'un système peut servir à coder le comportement entrée/sortie du système [13, 14]. Cette idée associée à des travaux sur l'algèbre de Lie [30, 34] donnera naissance à une partie de ce mémoire.

L'objet de ce mémoire

L'objet de ce mémoire est double:

Premièrement, il consiste à apporter une petite contribution à la description d'une série formelle, de manière à introduire des outils mathématiques ou informatiques susceptibles de faciliter son traitement en Calcul Formel. Cette partie est développée dans

les chapitres 1,2,3. Le chapitre 1 propose deux présentations d'une série formelle par sa matrice de Hankel et par l'automate de ses résiduelles. Le chapitre 2 traduit la rationalité d'une série par des propriétés de finitude de ses deux présentations. Le chapitre 3 généralise la notion d'approximant de type Padé et fournit les deux approximants de type Padé associés aux deux présentations.

Deuxièmement, ce mémoire présente deux applications des séries formelles au traitement des systèmes dynamiques.

La première application consiste à proposer des approximants bilinéaires de systèmes dynamiques. Cette application est développée dans le chapitre 6, les chapitres 4 et 5 faisant le lien entre systèmes dynamiques et séries formelles, et entre systèmes bilinéaires et séries rationnelles.

La seconde application consiste à résoudre partiellement le problème de l'Identification Algébrique: *Peut-on identifier la série génératrice d'un système dynamique lorsqu'on connaît les développements de Taylor en $t = 0$ des entrées et des sorties correspondantes?* Dans le chapitre 7, il est présenté plusieurs voies pour résoudre le problème de l'Identification Algébrique. Les chapitres 8 et 9 présentent les deux étapes débouchant sur un algorithme vérifiant si une condition suffisante d'identifiabilité est satisfaite jusqu'à un certain ordre.

Présentation succincte des chapitres

Premier chapitre

Comme on l'a vu dans l'introduction, la notion de série formelle est ancienne. Nous essayons de proposer des définitions les plus fondamentales et les plus intrinsèques possibles, afin de pouvoir travailler dans le cadre le plus général. Certains résultats exposés ici figurent sous une autre forme dans [30].

Nous donnons dans ce chapitre, une approche originale des séries formelles. Nous définissons l'**objet série formelle** comme l'élément de \mathcal{R}^M , ensemble des applications du monoïde M dans \mathcal{R} , M étant le monoïde libre engendré par un alphabet fini Z .

- En préliminaires, nous définissons dans un cadre très général, les notions de produit tensoriel, algèbre d'un ensemble, reconnaissabilité d'une observation f de \mathcal{R}^M , M étant un monoïde quelconque. Nous proposons alors **deux présentations de f : par ses résiduelles, et par sa matrice de Hankel.**
- Pour $M = Z^*$, nous remarquons que l'ensemble \mathcal{R}^M des séries formelles, désormais noté $\mathcal{R}\langle\langle Z \rangle\rangle$, peut être muni d'une **très abondante structure algébrique**: c'est une algèbre de Cauchy grâce au produit de Cauchy (\cdot), une cogèbre de décomposition grâce au coproduit (Γ). Muni des opérations Γ et \cdot , $\mathcal{R}\langle\langle Z \rangle\rangle$ apparaît comme la complétée de l'algèbre de Hopf de décomposition $\mathcal{R}\langle Z \rangle$. Muni

des opérations Φ (coproduit de factorisation), et \smile (produit de mélange), définies par dualité par rapport à \cdot et Γ , $\mathbb{R}\langle\langle Z \rangle\rangle$ apparaît comme la complétée de l'algèbre de Hopf de factorisation $\mathbb{R}\langle Z \rangle$.

- Nous rappelons enfin que $\mathbb{R}\langle\langle Z \rangle\rangle$ est muni de la topologie produit issue de la topologie discrète de \mathbb{R} , cette topologie pouvant être définie par une **valuation ultramétrique**.

Deuxième chapitre

Ce chapitre s'articule autour de trois parties: **série rationnelle, série reconnaissable, théorème de Schützenberger**. Ces notions sont bien connues [28], mais nous avons jugé intéressant de les aborder avec les concepts développés au chapitre précédent, et avec l'idée de disposer d'un outil de présentation pour le Calcul Formel.

- On rappelle que la notion de série rationnelle en variables non commutatives est définie comme une généralisation de la notion de série rationnelle en une variable: Les séries rationnelles sont les éléments de la clôture rationnelle de $\mathbb{R}\langle Z \rangle$, ensemble des polynômes.
- On sait d'après le chapitre précédent que les séries reconnaissables sont caractérisées par la finitude de leur rang (théorème de Schützenberger-Fliess). Notre intention est de proposer une présentation des séries rationnelles, à partir des concepts déjà développés, qui soit adaptée au Calcul Formel. Après une introduction sur les \mathbb{R} -automates (définition, graphe, représentation matricielle), nous proposons d'associer à toute série formelle s , son " \mathbb{R} -automate", dont les états sont les résiduelles de s et les actions sont les actions des lettres sur les résiduelles. Notre présentation d'une série reconnaissable est alors donnée par le \mathbb{R} -automate fini associé.
- Enfin, on rappelle l'énoncé du théorème de Schützenberger établissant l'équivalence de la reconnaissabilité et de la rationalité des séries formelles propres et on en propose une nouvelle démonstration s'appuyant sur des calculs de résiduelles.

Troisième chapitre

Dans ce chapitre, nous poursuivons le double but de **généraliser à plusieurs variables non commutatives, la notion d'approximant de type Padé** d'une série formelle à une variable [4], et de proposer deux mises en œuvre différentes, issues des deux présentations d'une série formelle à plusieurs variables non commutatives, introduites au premier chapitre.

- Après des rappels sur les approximants rationnels classiques de séries à une variable: polynômes, approximants de type Padé, approximants de Padé, on propose

une généralisation de la notion d'approximant de type Padé de série s en variables non commutatives.

- Deux techniques de calcul sont développées pour mettre en œuvre cette notion: L'une est basée sur des calculs à partir de la présentation de s par la matrice de Hankel. Elle fournit un approximant de type Padé de rang minimal. L'autre est obtenue à partir de la présentation de s par l'automate associé à ses résiduelles. Elle consiste à tronquer convenablement cet automate.

Des comparaisons des avantages et des inconvénients respectifs des deux méthodes sont présentées en conclusion. Il est à noter qu'elles fournissent l'une comme l'autre un outil intéressant pour le Calcul Formel.

Quatrième chapitre

Ce chapitre fait seulement **le lien entre séries formelles et systèmes dynamiques**. Nous nous intéressons désormais au traitement de systèmes dynamiques linéaires en la commande. Des séries formelles en variables non commutatives y apparaissent naturellement. Nous rappelons certaines propriétés de ces séries:

- La série génératrice ou série de Fliess [14, 15] sert à coder le comportement entrée/sortie du système. La série de Chen est la série associée à la commande du système.
- Une condition nécessaire et suffisante pour qu'une série formelle quelconque s soit la série génératrice d'un tel système est qu'elle soit "produite différentiellement" [15, 29].

Cinquième chapitre

Ce chapitre fait seulement **le lien entre séries rationnelles et systèmes bilinéaires**. Nous rappelons la définition d'un système bilinéaire, énoncée par l'une des deux propriétés équivalentes suivantes:

- Son équation d'état ne contient que des matrices de dimension finie.
- Sa série génératrice est rationnelle.

Puis nous indiquons comment obtenir un système bilinéaire réduit à partir de sa série génératrice rationnelle, grâce à sa "réalisation minimale" [21].

Sixième chapitre

Le but de ce chapitre est de fournir des **approximants bilinéaires de systèmes dynamiques à l'ordre k** , qui soient des approximants de type Padé et qui vérifient une condition Taylorienne sur la sortie, à l'ordre k :

Les développements de Taylor des sorties du système donné et de son approximant doivent coïncider jusqu'à l'ordre k .

Nous développons deux techniques pour mettre en œuvre ce concept:

- L'une est basée sur des calculs à partir de la présentation de la série génératrice du système, par la matrice de Hankel. Elle fournit un approximant bilinéaire de type Padé de rang minimal.
- L'autre est obtenue en tronquant un automate associé à la série génératrice du système. Nous choisissons de présenter deux automates particuliers que nous appelons "automate géométrique" (méthode de Carleman [5]) et "automate structural" (méthode de l'algèbre différentielle).

Des comparaisons des avantages et des inconvénients respectifs des deux méthodes sont présentés en conclusion. Il est à noter que la méthode par la présentation de la matrice de Hankel et la méthode par l'automate structural fournissent l'une comme l'autre un outil intéressant pour le Calcul Formel.

Septième chapitre

Dans ce chapitre et les suivants, nous poserons le problème de l'**Identification Algébrique**:

Peut-on, à partir d'un jeu d'informations discrètes et exactes sur le comportement entrée/sortie d'une fonctionnelle causale, construire un algorithme calculant jusqu'à un ordre donné, les coefficients (en calcul exact) de sa série génératrice?

Avant d'aborder précisément ce problème, nous rappelons les résultats obtenus pour des problèmes voisins et explorons diverses voies.

Dans un cadre très général, nous rappelons que J.Sjoberg et al. présentent dans [1], un tour d'horizon sur les modélisations de boîtes noires dans l'Identification des systèmes. A.Juditsky et al. développent dans [2] un éventail de fondements mathématiques, qui sont à la base des algorithmes d'Identification de systèmes non linéaires.

Par ailleurs, d'importants résultats ont été obtenus par F.Ollivier [26], E.Walter [35] sur l'Identification de paramètres, et par Narendra et al. [24], sur l'Identification par réseaux neuronaux. Nous les résumons brièvement.

Puis nous proposons plusieurs voies pour résoudre le problème de l'Identification Al-

gébrique:

– **Par des techniques d’observation:**

1) Selon une idée de M.Petitot, il s’agit d’identifier les coefficients de la série génératrice supposée polynomiale, i.e. identifier les coefficients de la fonction d’observation grâce à des mesures de la sortie à divers instants t pour des entrées convenables. Ce problème se ramène à celui du “Motion Planning” [16].

2) On se place sous les mêmes hypothèses qu’en 1). Mais l’identification se fait cette fois par les fonctionnelles différentiables. Il s’agit d’identifier les coefficients de la fonction d’observation grâce à un jeu d’entrées/sorties et leurs dérivées en $t = 0$.

– **Par la série génératrice:**

Le problème posé est de proposer un algorithme déterminant la série génératrice jusqu’au degré k , quand on connaît les développements de Taylor (en $t = 0$) des entrées et les développements de Taylor (en $t = 0$) des sorties correspondantes. Après une analyse du problème [17, 18], nous fournissons un algorithme.

Le traitement de ce problème passe par 3 étapes:

* Ecrire un système d’équations linéaires à partir des dérivées itérées de la sortie.

* Identifier les coefficients des multidérivées des entrées.

* Identifier les mots de même image commutative.

Un début de démonstration de l’identifiabilité algébrique est proposé. En tout état de cause, nous fournissons l’algorithme suivant:

Etant donné un ordre $k \geq 0$, des tests établissent une condition suffisante d’identifiabilité de la série génératrice tronquée aux mots de longueur $l \leq k$.

Il est à noter que l’Identification algébrique s’articule naturellement avec d’autres applications des séries formelles aux systèmes dynamiques: elle peut être utilisée comme point de départ pour de multiples sujets tels que l’approximation bilinéaire, la réalisation minimale, etc...

Huitième chapitre

Dans ce chapitre, nous présentons la première étape vers l’identification Algébrique. Pour cela, nous calculons les dérivées itérées de la sortie du système (Σ) . Elles apparaissent comme des combinaisons linéaires des multidérivées des entrées.

Nous proposons ensuite un algorithme d’identification des coefficients des multidérivées des entrées. Cette identification relève de la théorie du contrôle: elle est basée sur des mesures des valeurs d’entrée/sortie (et de leurs dérivées) et consiste à identifier des coefficients de la représentation algébrique-différentielle du système.

Après quelques rappels de Combinatoire, nous présentons une interprétation combinatoire de certaines entités apparaissant dans les calculs. Cet éclairage par la Combinatoire préparera la seconde étape vers l’identification algébrique développée au chapitre suivant.

Neuvième chapitre

Nous proposons, dans ce chapitre, un algorithme permettant de savoir si l'on peut identifier les coefficients de la série génératrice G jusqu'au degré l , à partir du système d'équations linéaires obtenu précédemment:

$$y^{(k)}(0) = \sum_{\mu} a^{\mu} \langle G | g_{\mu} \rangle$$

Dans une première étape, nous avons calculé les contributions de G , relatives à une famille $\{g_{\mu}\}$ de polynômes associés aux multidérivées des entrées (famille indexée par l'ensemble des partitions).

Dans cette deuxième étape, il reste à extraire les coefficients $\langle G | w \rangle$. Après une analyse de ce problème de scindage, nous proposons un algorithme. Il est à noter que cette seconde étape d'identification n'est pas de même nature que la première: elle est de nature purement algébrique et donc générique. Cette phase de l'identification peut être calculée une fois pour toutes. Cette remarque fait ressortir l'intérêt de pouvoir réaliser le calcul assez loin.

Quelques développements possibles

Les différentes parties de ce mémoire peuvent s'enchaîner pour produire quelques agencements utilitaires possibles tels que:

– **simulation**

Enchaînement de l'identification algébrique, d'approximations bilinéaires et de leurs simulations.

– **compression**

Etant donné un système analytique, enchaînement d'une approximation rationnelle de "petite dimension", de sa simulation, de mesures d'écart entre le système réel et le système approché, et de la validation du système approché.

– **apprentissage**

Etant données des suites redondantes de mesures, enchaînement d'identifications pour chaque suite de tests, comparaisons, approximations (moindres carrés par exemple), et validation en tant que système causal.

Bibliographie

- [1] Sjöberg J., Zhang Q., Ljung L., Benveniste A., Delyon B., Glorennec P.Y., Hjalmarsson H., Juditsky A., *Nonlinear Black-box Modeling in System Identification: a Unified Overview*, Automatica vol. 31, n° 12, pp. 1691-1724, 1995.
- [2] Juditsky A., Hjalmarsson H., Benveniste A., Delyon B., Ljung L., Sjöberg J., Zhang Q., *Nonlinear Black-box Models in System Identification: Mathematical Foundations*, Automatica vol. 31, n° 12, pp. 1725-1750, 1995.
- [3] Berstel J., *Contribution à l'étude des propriétés arithmétiques des langages formels*, Thèse, Université de Paris-VII, 1972.
- [4] Brezinski C., *Padé type approximation and general orthogonal polynomials*, INSM50, Birkhauser, 1980.
- [5] Brockett R.W., *Voterra series and Geometric Control Theory*, Automatica, vol.12, pp. 167-176, 1976.
- [6] Cartier P., Foata D., *Problèmes combinatoires de commutation et réarrangements*, Lect. Notes Math., n° 85, Springer-Verlag, Berlin, 1969.
- [7] Chen W.Y., *Context-free grammars, differential operators and formal power series*, Séries formelles et combinatoire algébrique, pp. 145-159, Bordeaux 2-4 mai 1991.
- [8] Chomsky N., Schützenberger M.P., *The Algebraic Theory of Context-free Languages*, in Computer Programming and Formal Systems, P.Brafford and D.Hirschberg, ed., North Holland, pp. 118-161, 1963.
- [9] Cori R., Richard J., *Enumération des graphes planaires à l'aide des séries formelles en variables non commutatives*, Discrete Math., t. 2, pp. 115-162, 1972.
- [10] Fliess M., *Séries reconnaissables, rationnelles et algébriques*, Bull. Soc. Math. Fr., vol. 94, pp. 231-239, 1970.
- [11] Fliess M., *Sur certaines familles de séries formelles*, Thèse, Université de Paris-VII, 1972.
- [12] Fliess M., *Séries rationnelles positives et processus stochastiques*, Ann. Inst. H. Poincaré, sect. B, vol. XI, n° 1, pp. 1-21, 1975.

- [13] Fliess M., *Un outil algébrique: les séries formelles non commutatives*, in “Mathematical System Theory” (G. Marchesini and S.K. Mitter Eds.), Lecture Notes Econom. Math. Syst., Springer Verlag, vol. 131, pp. 122-148, 1976.
- [14] Fliess M., *Fonctionnelles causales non linéaires et indéterminées non commutatives*, Bull. Soc. Math. France 109, pp.3-40, 1981.
- [15] Fliess M., Lamnabhi M., Lamnabhi-Lagarrigue F., *An algebraic approach to nonlinear functional expansions*, IEEE Trans. Circuits and Systems, vol. CAS-30, n°8, pp. 554-570, 1983.
- [16] Guyon C., *Calcul Symbolique pour la planification de Trajectoire des Systèmes dynamiques nilpotents*, Thèse Lille I, 1995.
- [17] Hespel C., *Iterated derivatives of a non linear dynamic system and Faà di Bruno formula*, Mathematics and Computers in Simulation, vol. 42, pp. 641-657, 1996.
- [18] Hespel C., Jacob G., *First steps towards exact algebraic identification*, Discrete Math., vol. 180, pp. 211-219, 1998.
- [19] Jacob G., *Sur un théorème de shamir*, Inf. Contr., vol. 27, n° 3, pp. 218-261, 1975.
- [20] Jacob G., *Réalisation des systèmes réguliers et séries génératrices non commutatives*, publication interne de l'équipe lilloise d'Informatique théorique, n° IT-26-80, 1980.
- [21] Jacob G., *Réalisation des systèmes réguliers (ou bilinéaires) et séries génératrices non commutatives*, Séminaire d'Aussois, RCP567, Outils et modèles mathématiques pour l'Automatique, l'Analyse des Systèmes, et le traitement du Signal (CNRS Landau), 1980.
- [22] Joyal A., *Une théorie combinatoire des séries formelles*, Advances in Mathematics, 42, pp. 1-82, 1981.
- [23] Leroux P., Viennot X.G., *A combinatorial approach to nonlinear functional expansions: an introduction with example*, Algebraic and Computing Treatment of Noncommutative Power Series, G.Jacob and C.Reutenauer eds., Theoret. Comput. Sciences, 1991.
- [24] Narendra, Parthasarathy K. and K., *Identification and acontrol of Dynamical Systems Using Neural Networks*, IEEE Transactions on Neural Networks, 1 (1), pp. 4-27, 1990.
- [25] Nivat M., *Transductions des langages de Chomsky*, Ann. Inst. Fourier, vol. 18, pp. 339-455, 1968.
- [26] Ollivier F., *Le problème de l'identifiabilité structurelle globale: approche théorique, méthodes effectives et bornes de complexité*, Thèse Ecole Polytechnique, 1990.

- [27] Reutenauer C., *Séries formelles et algèbres syntactiques*, J. Algebra, vol. 66, pp. 448-483, 1980.
- [28] Berstel J., Reutenauer C., *Les séries rationnelles et leurs langages*, Masson, 1984.
- [29] Reutenauer C., *The local realization of generating series of finite Lie rank*, in "Algebraic and Geometric Methods in Nonlinear Control Theory", M.Fliess and M.Hazewinkel eds., D.Reidel Publishing Company, pp. 33-43, 1986.
- [30] Reutenauer C., *Free Lie Algebras*, Oxford Science Publications, 1993.
- [31] Salomaa A., Soittola M., *Automata-Theoretic Aspects of formal power series*, Springer Verlag, 1978.
- [32] Schützenberger M.P., *Un problème de la théorie des automates*, séminaire Dubreil-Pisot (Algèbre et théorie des nombres, 13ème année, n° 3, Inst. H.Poincaré, 1960.
- [33] Schützenberger M.P., *On a theorem of Jung*, Proc. Amer. Math. Soc., vol. 13, pp. 885-890, 1962.
- [34] Viennot G., *Algèbres de Lie libres et Monoïdes libres*, Lecture notes in Mathematics, n° 691, Springer Verlag, 1978.
- [35] Distefano, Lamnabhi-Lagarrigue, Lecourtier, Raksanyi, Vajda, Walter, etc ..., *Identifiability of parametric models*, E.Walter editeur, Pergamon Press, Oxford, 1987.

Chapitre 1

Monoïdes et séries formelles

1.1 Introduction

Comme on l'a vu précédemment, la notion de série formelle est ancienne. Nous essayons de proposer des définitions les plus fondamentales et les plus intrinsèques possibles, afin de pouvoir travailler dans le cadre le plus général. Certains résultats exposés ici figurent sous une autre forme dans [3]

Nous donnons dans ce chapitre, une approche originale des séries formelles. Nous définissons l'**objet série formelle** comme l'élément de \mathbb{R}^M , ensemble des applications du monoïde M dans \mathbb{R} , M étant le monoïde libre engendré par un alphabet fini Z .

- En préliminaires, nous définissons dans un cadre très général, les notions de produit tensoriel, algèbre d'un ensemble, reconnaissabilité d'une observation f de \mathbb{R}^M , M étant un monoïde quelconque. Nous proposons alors **deux présentations de f : par ses résiduelles, et par sa matrice de Hankel.**
- Pour $M = Z^*$, nous observons que l'ensemble \mathbb{R}^M des séries formelles, désormais noté $\mathbb{R}\langle\langle Z \rangle\rangle$, peut être muni d'une **très abondante structure algébrique**: c'est une algèbre de Cauchy grâce au produit de Cauchy (\cdot), une cogèbre de décomposition grâce au coproduit (Γ). Muni des opérations Γ et \cdot , $\mathbb{R}\langle\langle Z \rangle\rangle$ apparaît comme la complétée de l'algèbre de Hopf de décomposition $\mathbb{R}\langle Z \rangle$. Muni des opérations Φ (coproduit de factorisation), et ω (produit de mélange), définies par dualité par rapport à \cdot et Γ , $\mathbb{R}\langle\langle Z \rangle\rangle$ apparaît comme la complétée de l'algèbre de Hopf de factorisation $\mathbb{R}\langle Z \rangle$.
- Nous rappelons enfin que $\mathbb{R}\langle\langle Z \rangle\rangle$ est muni de la topologie produit issue de la topologie discrète de \mathbb{R} , cette topologie pouvant être définie par une **valuation ultramétrique.**

1.2 Application bilinéaire et produit tensoriel

Soient A et B deux espaces vectoriels. On s'intéresse à toutes les applications bilinéaires F de $A \times B$ dans un espace vectoriel C sur un corps K .

Définition 1.2.1 *On appelle produit tensoriel des 2 espaces vectoriels A et B , la donnée d'un espace vectoriel noté $A \otimes B$ et d'une application bilinéaire $\phi : A \times B \rightarrow A \otimes B$ universels dans le sens suivant:*

Pour toute application bilinéaire F de $A \times B$ dans un espace vectoriel C , il existe une et une seule application linéaire θ de $A \otimes B$ dans C qui factorise F à travers ϕ :

$$F = \theta \circ \phi$$

Cette définition par caractérisation universelle peut être traduite en une définition constructive:

Le produit tensoriel de A et B est l'espace vectoriel quotient de $A \times B$ par le sous espace vectoriel J engendré par la famille d'égalités suivantes:

$$\begin{aligned} (ka, b) &= (a, kb) = k(a, b) \quad \forall k \in K \\ (a + a', b) &= (a, b) + (a', b) \\ (a, b + b') &= (a, b) + (a, b') \end{aligned}$$

Tout élément de $A \otimes B$ peut s'écrire comme une somme finie de la forme :

$$\sum_{j \in \{1..n\}} a_j \otimes b_j$$

mais cette écriture n'est pas unique.

1.2.1 Algèbre d'un ensemble, coupes

Soit E un ensemble. On note \mathbb{R}^E l'ensemble des applications de E dans \mathbb{R} . Pour tout f de \mathbb{R}^E , et tout e de E , l'image de e par f est notée

$$\langle f|e \rangle$$

On vérifie que \mathbb{R}^E est un espace vectoriel pour les opérations obtenues en étendant point par point (c'est à dire en tout e de E) les opérations dans \mathbb{R} de somme et de produit.

D'autre part, la multiplication dans \mathbb{R} étant une application bilinéaire, on peut définir une application bilinéaire de $\mathbb{R}^E \times \mathbb{R}^E$ dans \mathbb{R}^E :

$$\begin{aligned} \mathbb{R}^E \times \mathbb{R}^E &\rightarrow \mathbb{R}^E \\ (f, g) &\rightarrow f \odot g \end{aligned}$$

telle que

$$\langle f \odot g | e \rangle = \langle f | e \rangle \langle g | e \rangle$$

$f \odot g$ est appelée alors produit de Hadamard de f et de g . Munie de cette opération, \mathbb{R}^E possède une structure d'algèbre. Nous l'appelons algèbre de l'ensemble E .

Etant donnés deux ensembles E_1 et E_2 , on peut aussi définir une application bilinéaire de $\mathbb{R}^{E_1} \times \mathbb{R}^{E_2}$ dans $\mathbb{R}^{E_1 \times E_2}$. Cette application bilinéaire se factorise de façon unique à travers le produit tensoriel $\mathbb{R}^{E_1} \otimes \mathbb{R}^{E_2}$. Soit $\pi : \mathbb{R}^{E_1} \otimes \mathbb{R}^{E_2} \rightarrow \mathbb{R}^{E_1 \times E_2}$ l'application de factorisation. Elle est encore définie par la propriété:

$$\langle \pi(f \otimes g) | e_1, e_2 \rangle = f(e_1)g(e_2) \quad \forall e_1 \in E_1, \quad \forall e_2 \in E_2$$

D'autre part, pour toute application $h \in \mathbb{R}^{E_1 \times E_2}$, on peut définir la coupe h_{e_1} en un élément $e_1 \in E_1$ (resp. h_{e_2} en un élément $e_2 \in E_2$) comme suit:

$$\langle h_{e_1} | e_2 \rangle = \langle h | e_1, e_2 \rangle = \langle h_{e_2} | e_1 \rangle$$

1.2.2 Lemme 1.2.1, Proposition 1.2.1

► Élimination de Gauss

Lemme 1.2.1 *Un sous espace vectoriel W de \mathbb{R}^E est de dimension finie égale à n ssi il existe conjointement*

- Une suite finie (f_1, f_2, \dots, f_n) d'éléments de \mathbb{R}^E
 - et une suite finie (e_1, e_2, \dots, e_n) d'éléments de E
- vérifiant: $\forall j, k \in [1 \dots n] \quad \langle f_j | e_k \rangle = \delta_j^k$ (δ_j^k étant le symbole de Kronecker).

Une telle famille double $(f_j, e_j)_{j \in [1 \dots n]}$ vérifiant les conditions énoncées, sera encore appelée *présentation découplée* de W .

Preuve

* Si W est réduit au seul vecteur nul, il n'y a rien à montrer.

* Supposons le théorème démontré pour toute dimension inférieure ou égale à n , et supposons W de dimension $n + 1$. Alors W peut s'écrire comme une somme directe :

$$W = W_1 \oplus \mathbb{R}.f$$

W_1 étant un sous espace de dimension n . Choisissons une présentation découplée $(f_j, e_j)_{j \in [1 \dots n]}$ de W_1 . Si l'on pose alors

$$h = f - \sum_{j \in [1 \dots n]} \langle f | e_j \rangle f_j$$

Donc on a : $\langle h | e_k \rangle = \langle f | e_k \rangle - \sum_{j \in [1 \dots n]} \langle f | e_j \rangle \delta_j^k = 0$ pour tout $k \in [1 \dots n]$.

Or W est encore somme directe de W_1 et de $\mathbb{R}.h$ et donc h n'est pas nul. On peut donc choisir $a \in E$ tel que $\langle h | a \rangle \neq 0$. En particulier, a ne peut être l'un des e_j pour $j \leq n$.

On pose donc $e_{n+1} = a$, et on définit f_{n+1} en normalisant h :

$$f_{n+1} = \langle h | e_{n+1} \rangle^{-1} h$$

On en déduit $\langle f_{n+1} | e_{n+1} \rangle = 1$ et la double suite $(f_j, e_j)_{j \in [1 \dots n+1]}$ est une présentation découplée de W .

Fin de preuve

► **Proposition 1.2.1**

Proposition 1.2.1 Soient E_1 et E_2 deux ensembles. L'application linéaire $\pi : \mathbb{R}^{E_1} \otimes \mathbb{R}^{E_2} \rightarrow \mathbb{R}^{E_1} \times \mathbb{R}^{E_2}$ définie par la propriété:

$$\langle \pi(f \otimes g) | e_1, e_2 \rangle = f(e_1)g(e_2) \quad \forall e_1 \in E_1, \quad \forall e_2 \in E_2$$

est injective. En outre, pour une application $h \in \mathbb{R}^{E_1 \times E_2}$, les propriétés suivantes sont équivalentes:

- h appartient à l'image de π
- L'espace vectoriel engendré par les fonctions h_{e_1} est de dimension finie
- L'espace vectoriel engendré par les fonctions h_{e_2} est de dimension finie.

Preuve

1) Montrons que π est injective. Soit

$$F = \sum_{j \in [1 \dots n]} f_j \otimes g_j$$

un élément de $\mathbb{R}^{E_1} \otimes \mathbb{R}^{E_2}$ vérifiant $\pi(F) = 0$. Soit alors W l'espace vectoriel engendré par les g_j .

* Si W est réduit à 0, alors F est nulle.

* Dans le cas contraire, soit $(h_k, b_k)_{k \in [1..n]}$ une présentation découplée de W . On peut

alors réécrire F sous la forme $F = \sum_{j \in [1..n]} \hat{f}_j \otimes h_j$. On a alors

$$0 = \langle \pi(F) | a, b_k \rangle = \sum_{j \in [1..n]} \langle \hat{f}_j | a \rangle \langle h_j | b_k \rangle = \langle \hat{f}_k | a \rangle \quad \forall k$$

Ceci montre que les fonctions \hat{f}_k sont toutes identiquement nulles, et donc que F est nulle.

2) Tout élément de $\mathbb{R}^{E_1} \otimes \mathbb{R}^{E_2}$ peut donc s'identifier avec son image dans $\mathbb{R}^{E_1 \times E_2}$.

* Tout élément h de l'image de π est une somme finie de la forme:

$$h = \sum_{j \in [1..n]} f_j \otimes g_j$$

L'égalité $\langle h | a, b \rangle = \sum_{j \in [1..n]} \langle f_j | a \rangle \langle g_j | b \rangle$ s'écrit encore:

$$h_{a?} = \sum_{j \in [1..n]} \langle f_j | a \rangle g_j$$

Ceci montre que toutes les coupes $h_{a?}$ sont dans l'espace vectoriel de dimension finie engendré par les g_j .

* Inversement, si toutes les coupes $h_{a?}$ sont de la forme

$$h_{a?} = \sum_{j \in [1..n]} \alpha_j(a) g_j$$

pour des coefficients $\alpha_j(a) \in \mathbb{R}$, alors ces coefficients définissent des fonctions $\alpha_j \in \mathbb{R}^{E_1}$, et l'on a:

$$\begin{aligned} \langle h | a, b \rangle &= \sum_{j \in [1..n]} \langle \alpha_j | a \rangle \langle g_j | b \rangle \\ h &= \sum_{j \in [1..n]} \alpha_j \otimes g_j \end{aligned}$$

et donc h est dans l'image de π .

* La démonstration concernant les coupes à droite $h_{?b}$ s'en déduit par symétrie.

Fin de preuve

1.3 Monoïdes et Observations

Soit M un monoïde possédant une loi de composition interne associative notée m ou plus simplement “.” :

Si x et y sont dans M , on note $m(x, y)$ ou $\langle m|x, y \rangle$ ou plus simplement $x.y$.

\mathbb{R}^M se trouve alors muni de la structure d’algèbre définie au paragraphe 1.2.1 . On le note alors $\mathbb{R}\langle\langle M \rangle\rangle$.

Définition 1.3.1 *On appelle observation, toute application f :*

$$\begin{aligned} M &\rightarrow \mathbb{R} \\ e &\rightarrow \langle f|e \rangle \end{aligned}$$

Plusieurs présentations sont intéressantes pour caractériser une observation f .

1.3.1 Les présentations

► Présentation de Hankel, matrice de Hankel

Notons m la loi de composition du monoïde M . On définit alors l’application $m^* : \mathbb{R}^M \rightarrow \mathbb{R}^{M \times M}$ comme la transposée de m , c’est à dire :

$$\begin{aligned} m^*(f) &= f \circ m \\ \langle m^*(f)|x, y \rangle &= f(x.y) \end{aligned}$$

Définition 1.3.2 *On appelle présentation de Hankel de f notée $H(f)$, cette transposée $m^*(f)$. Donc :*

$$\langle m^*(f)|x, y \rangle = \langle f|x.y \rangle$$

Définition 1.3.3 *Pour toute énumération $(x_i)_{i \in \mathbb{N}}$ des éléments de M (on suppose donc M dénombrable), on appelle matrice de Hankel de l’observation f , la description matricielle de $m^*(f)$, pour l’énumération considérée.*

► Présentation par les résiduelles

On peut définir les présentations résiduelles à droite et à gauche de f de la manière suivante [1]:

• A droite .

On note $f \triangleright x$ pour $x \in M$ tel que

$$\langle f \triangleright x_1|x_2 \rangle = \langle f|x_1.x_2 \rangle$$

• **A gauche.**

On note $x \triangleleft f$ pour $x \in M$ tel que

$$\langle x_2 \triangleleft f | x_1 \rangle = \langle f | x_1 . x_2 \rangle$$

• **Remarques :**

1) On observe les propriétés suivantes:

$$\begin{aligned} \forall u, v \in M, f \triangleright u.v &= (f \triangleright u) \triangleright v \\ \forall u, v \in M, u.v \triangleleft f &= u \triangleleft (v \triangleleft f) \end{aligned}$$

2) Si M possède un élément neutre noté ϵ ou 1 , on remarque que:

$$\begin{aligned} \forall u \in M, \langle f | u \rangle &= \langle f \triangleright u | \epsilon \rangle \\ \forall u \in M, \langle f | u \rangle &= \langle u \triangleleft f | \epsilon \rangle \end{aligned}$$

1.3.2 Le théorème de Kleene

Proposition 1.3.1 *Soit f une application d'un monoïde M dans \mathbb{R} . Les propriétés suivantes sont équivalentes :*

- $m^*(f)$ appartient à $\pi(\mathbb{R}^M \otimes \mathbb{R}^M)$
- L'espace vectoriel engendré par les résiduelles à droite $f \triangleright x$ est de dimension finie
- L'espace vectoriel engendré par les résiduelles à gauche $y \triangleleft f$ est de dimension finie

Preuve

C'est une simple relecture de la proposition 1.2.1. Il suffit pour cela de remarquer que $m^*(f)_{x?} = x \triangleleft f$ et $m^*(f)_{?y} = f \triangleright y$

En effet:

$$\begin{aligned} \langle m^*(f)_{x?} | y \rangle &= \langle m^*(f) | x, y \rangle = \langle f | x.y \rangle = \langle f \triangleright x | y \rangle \\ \langle m^*(f)_{?y} | x \rangle &= \langle m^*(f) | x, y \rangle = \langle f | x.y \rangle = \langle y \triangleleft f | x \rangle \end{aligned}$$

Fin de preuve

Définition 1.3.4 *Une fonction f d'un monoïde M à valeurs dans \mathbb{R} est dite reconnaissable si elle vérifie les trois conditions équivalentes de la proposition précédente.*

On note $\mathcal{R}(M)$ l'ensemble des fonctions reconnaissables de \mathbb{R}^M .

On remarque que f est reconnaissable ssi sa présentation de Hankel est de rang fini. En effet, le lemme 1.2.1 permet de dire que les deux espaces engendrés par les résiduelles à gauche (resp. à droite) sont simultanément de dimension finie ou infinie et que les dimensions de ces deux espaces sont égales. Cette dimension commune est alors le rang de $m^*(f)$.

Ceci peut encore s'interpréter comme suit:

Dans l'espace vectoriel engendré par les vecteurs colonnes de la matrice de Hankel de s , il existe une base finie de vecteurs colonnes $\{C_{y_1}, \dots, C_{y_N}\}$ ou encore une base finie de séries résiduelles à gauche $\{y_1 \triangleleft f, \dots, y_N \triangleleft f\}$.

De même, cela revient à dire qu'il existe une base finie de vecteurs lignes $\{L_{x_1}, \dots, L_{x_N}\}$ ou encore une base finie de séries résiduelles à droite $\{f \triangleright x_1, \dots, f \triangleright x_N\}$ dans l'espace vectoriel engendré par les vecteurs lignes de la matrice de Hankel de f .

1.4 Séries formelles

1.4.1 Définitions et notations

► Lettres, mots, séries

- $Z = \{z_0, z_1, \dots, z_m\}$ étant un **alphabet** fini, on suppose maintenant que le monoïde $M = Z^*$, monoïde libre engendré par Z . Les éléments z_i de Z sont appelés **lettres** et la loi de composition interne définie sur $M = Z^*$ est la **concaténation**. Z^* est donc l'ensemble des **mots** formés par concaténation sur les lettres $\{z_0, z_1, \dots, z_m\}$.

- Toute observation $s \in \mathbb{R}^{Z^*}$ est aussi appelée **série formelle** en les variables non commutatives z_0, \dots, z_m . On note cette série formelle comme une somme formelle par

$$s = \sum_{w \in Z^*} \langle s|w \rangle w.$$

On adopte la notation classique $\mathbb{R}\langle\langle Z \rangle\rangle = \mathbb{R}^{Z^*}$.

- Un mot v de Z^* est un **facteur** d'un mot x de Z^* s'il existe deux mots u, w de Z^* tels que

$$x = u.v.w$$

Un mot v est dit **facteur gauche ou préfixe** de $x \in Z^*$ s'il existe un mot $w \in Z^*$ tel que

$$x = v.w$$

Un mot w est dit **facteur droit ou suffixe** de $x \in Z^*$ s'il existe un mot $v \in Z^*$ tel que

$$x = v.w$$

- On peut utiliser les **représentations arborescentes préfixielles et suffixielles** pour déterminer les préfixes et les suffixes des mots.

exemple pour $Z = \{x, y\}$

On lit les préfixes des mots en parcourant l'arbre de droite à gauche (Figure 1.1). On lit les suffixes des mots en parcourant l'arbre de gauche à droite (Figure 1.2).

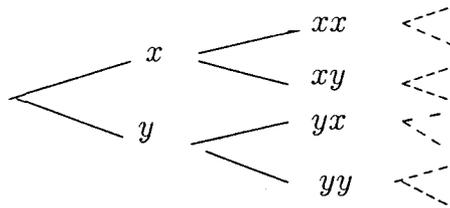


FIG. 1.1 – arbre des préfixes

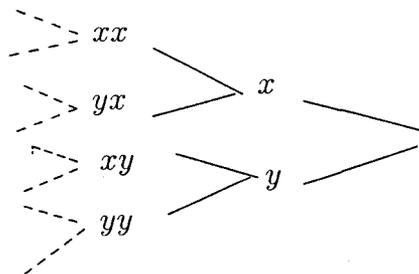


FIG. 1.2 – arbre des suffixes

Un **code préfixe** est une partie C de Z^* telle que

$$\forall u, v \in Z^* \quad u, u.v \in C \Rightarrow v = 1$$

Une **partie préfixielle** est une partie P de Z^* telle que

$$\forall u, v \in Z^* \quad u.v \in P \Rightarrow u \in P$$

► **Support, polynômes**

Un **langage** sur Z est une partie de Z^* .

Le **support** d'une série formelle s est le langage noté $Supp(s)$ formé des mots w de coefficient $\langle s|w \rangle$ non nul.

On appelle **polynôme** sur Z toute série formelle de support fini. On note $\mathbb{R}\langle Z \rangle$ l'ensemble des polynômes sur l'alphabet Z .

► **Résiduelles**

Définition 1.4.1 Conformément à la définition des résiduelles à droite et à gauche données au paragraphe 1.3.1, on définit les résiduelles d'une série s par :

- **A droite.**

$s \triangleright u$ pour $u \in Z^*$ tel que

$$\langle s \triangleright u | v \rangle = \langle s | u.v \rangle$$

• **A gauche.**

$v \triangleleft s$ pour $x \in M$ tel que

$$\langle v \triangleleft s | u \rangle = \langle s | u.v \rangle$$

Donc, on a pour tout mot $u \in Z^*$:

$$\begin{aligned} s \triangleright u &= \sum_{w \in Z^*} \langle s \triangleright u | w \rangle w \\ &= \sum_{w \in Z^*} \langle s | u.w \rangle w \\ &= \sum_{w \in Z^*} \langle s | w \rangle w \triangleright u \end{aligned}$$

$$\begin{aligned} u \triangleleft s &= \sum_{w \in Z^*} \langle u \triangleleft s | w \rangle w \\ &= \sum_{w \in Z^*} \langle s | w.u \rangle w \\ &= \sum_{w \in Z^*} \langle s | w \rangle u \triangleleft w \end{aligned}$$

• **Remarque**

Dans le cas particulier des résiduelles d'un mot par une lettre z de Z , on observe que :

$$\begin{aligned} (y.w) \triangleright z &= \delta_z^y w \\ z \triangleleft (w.y) &= \delta_z^y w \end{aligned}$$

Ceci peut s'interpréter de la manière suivante: La résiduelle à droite (resp. à gauche) d'un mot w par la lettre z est le mot obtenu en supprimant de w le préfixe (resp. le suffixe) z s'il figure, et 0 sinon.

1.4.2 Théorème de présentation minimale

Théorème 2 *Soit s une série reconnaissable telle que l'espace vectoriel engendré par ses résiduelles à droite soit de dimension n . Alors on peut choisir n mots (u_1, \dots, u_n) tels que $(s \triangleright u_1, \dots, s \triangleright u_n)$ en soit un système générateur, pour qu'ils forment un code préfixe.*

Preuve [2]

La démonstration est basée sur une exploration préfixielle de Z^* : Elle consiste à construire successivement des générateurs $s \triangleright \epsilon, s \triangleright z_1, \dots$ etc, en "purchassant" les dépendances linéaires afin de déterminer les (u_1, u_2, \dots) .

Fin de preuve

Remarque

Ce théorème permet de préciser la présentation de Hankel:

Soit $(d_j)_{j \in [1 \dots n]}$ un système générateur de l'espace vectoriel engendré par les résiduelles à droite de s . Soit $(g_j)_{j \in [1 \dots n]}$ un système générateur de l'espace vectoriel engendré par les résiduelles à gauche de s .

En posant

$$m^*(s) = \sum_{i,j} c_{ij} g_i \otimes d_j$$

Alors,

$$\langle s|u.v \rangle = \sum_{i,j} c_{ij} \langle g_i|u \rangle \langle d_j|v \rangle = \langle s \triangleright u|v \rangle$$

C'est à dire

$$s \triangleright u = \sum_{i,j} c_{ij} \langle g_i|u \rangle d_j$$

D'après le théorème, on peut préciser d_j et g_i :

$$\begin{aligned} d_j &= s \triangleright G_j \\ g_i &= D_i \triangleleft s \\ m^*(s) &= \sum_{i,j} c_{ij} (D_i \triangleleft s) \otimes (s \triangleright G_j) \end{aligned}$$

Ceci s'écrit encore :

$$\langle s|u.v \rangle = \sum_{i,j} c_{ij} \langle D_i \triangleleft s|u \rangle \langle s \triangleright G_j|v \rangle$$

Ce théorème permet de disposer d'un procédé effectif pour calculer une représentation linéaire réduite d'une série reconnaissable.

1.4.3 Quelles structures pour $\mathbb{R}\langle\langle Z \rangle\rangle$?

► **Produit de Cauchy**

On définit le **produit de Cauchy** ou **produit de concaténation** de deux séries s_1 et s_2 , noté $s_1.s_2$, en posant :

$$\begin{aligned} s_1.s_2 &= \sum_{w \in Z^*} \langle s_1.s_2|w \rangle w \\ s_1.s_2 &= \sum_{w \in Z^*} \sum_{u_1, u_2 \in Z^*, u_1.u_2=w} (\langle s_1|u_1 \rangle \langle s_2|u_2 \rangle) w \end{aligned}$$

L'addition, la multiplication par un scalaire, le produit de Cauchy munissent $\mathbb{R}\langle\langle Z \rangle\rangle$ d'une nouvelle structure d'algèbre, appelée algèbre de Cauchy.

► **Coproduit de décomposition**

On définit un **coproduit de décomposition**

$$\Gamma : \mathbb{R}\langle\langle Z \rangle\rangle \rightarrow \mathbb{R}\langle\langle Z \rangle\rangle \otimes \mathbb{R}\langle\langle Z \rangle\rangle$$

de la manière suivante :

On définit Γ sur les lettres par :

$$\Gamma(z) = 1 \otimes z + z \otimes 1 \quad \forall z \in Z$$

On le prolonge ensuite en un morphisme pour l'algèbre de Cauchy par :

$$\Gamma(u.v) = \Gamma(u)\Gamma(v) \quad \forall u, v \in Z^*$$

On prolonge aux séries formelles par continuité.

exemple pour deux lettres z_{i_1}, z_{i_2}

$$\begin{aligned} \Gamma(z_{i_1}.z_{i_2}) &= \Gamma(z_{i_1})\Gamma(z_{i_2}) \\ &= (1 \otimes z_{i_1} + z_{i_1} \otimes 1)(1 \otimes z_{i_2} + z_{i_2} \otimes 1) \\ &= 1 \otimes z_{i_1}.z_{i_2} + z_{i_2} \otimes z_{i_1} + z_{i_1} \otimes z_{i_2} + z_{i_1}.z_{i_2} \otimes 1. \end{aligned}$$

Plus généralement, on peut démontrer que :

$$\Gamma(w) = \sum u \otimes v$$

la somme étant étendue à toutes les "bipartitions" en u, v du mot w . Munie de ce coproduit, $\mathbb{R}\langle\langle Z \rangle\rangle$ devient une cogèbre, appelée **cogèbre de décomposition**.

► **Produit de mélange**

On définit le produit de mélange par dualité en posant :

$$\langle u \sqcup v | w \rangle = \langle \Gamma(w) | u \otimes v \rangle \quad \forall u, v \in Z^*$$

Remarques

1) On observe que cette définition par dualité concorde avec la définition récursive du produit de mélange.

exemple pour deux lettres :

$$\begin{aligned} \forall u \in Z^* \quad u \sqcup \epsilon &= \epsilon \sqcup u &= u \\ \forall u, v \in Z^*, \\ \forall z_{i_1}, z_{i_2} \in Z \\ u.z_{i_1} \sqcup v.z_{i_2} &= (u \sqcup v).z_{i_2}.z_{i_1} + (u.z_{i_1} \sqcup v).z_{i_2}. \end{aligned}$$

De plus, si s_1 et s_2 sont des séries formelles,

$$s_1 \sqcup s_2 = \sum_{u,v \in Z^*} \langle s_1|u \rangle \langle s_2|v \rangle u \sqcup v$$

2) On remarque que les résiduelles à droite (resp. à gauche) par une lettre sont **des dérivations pour le produit de mélange** :

$$\forall u, \forall v \in Z^*, \forall z \in Z$$

$$z \triangleleft (u \sqcup v) = (z \triangleleft u) \sqcup v + u \sqcup (z \triangleleft v)$$

$$(\text{resp. } (u \sqcup v) \triangleright z = (u \triangleright z) \sqcup v + u \sqcup (v \triangleright z)).$$

En effet,

$$\begin{aligned} \langle u \sqcup v \triangleright z | w \rangle &= \langle u \sqcup v | z.w \rangle \\ \langle \Gamma(z)\Gamma(w) | u \otimes v \rangle &= \langle (1 \otimes z)\Gamma(w) | u \otimes v \rangle + \langle (z \otimes 1)\Gamma(w) | u \otimes v \rangle \\ &= \langle \Gamma(w) | u \otimes (v \triangleright z) \rangle + \langle \Gamma(w) | (u \triangleright z) \otimes v \rangle \\ &= \langle u \sqcup (v \triangleright z) | w \rangle + \langle (u \triangleright z) \sqcup v | w \rangle \\ (u \sqcup v) \triangleright z &= u \sqcup (v \triangleright z) + (u \triangleright z) \sqcup v \end{aligned}$$

Mais les composées de deux résiduelles à gauche (resp. à droite) par une lettre z_{i_1} puis par une lettre z_{i_2} ne sont pas des dérivations pour le produit de mélange.

► Algèbre de Hopf de décomposition

Muni des opérations Γ et \cdot , $\mathcal{R}\langle\langle Z \rangle\rangle$ apparaît comme la complétée de l'algèbre de Hopf de décomposition $\mathcal{R}\langle Z \rangle$.

Soient $f, g \in \mathcal{L}(\mathcal{R}\langle Z \rangle, \mathcal{R}\langle Z \rangle)$. On définit leur **produit de convolution** en posant :

$$f * g = m \circ (f \otimes g) \circ \Gamma$$

conformément au diagramme suivant :

$$\begin{array}{ccc} & f & \\ \mathcal{R}\langle Z \rangle & \longrightarrow & \mathcal{R}\langle Z \rangle \\ \otimes & & \otimes \\ \mathcal{R}\langle Z \rangle & \longrightarrow & \mathcal{R}\langle Z \rangle \\ & g & \\ \Gamma \uparrow & & \downarrow m \\ \mathcal{R}\langle Z \rangle & \longrightarrow & \mathcal{R}\langle Z \rangle \\ & f * g & \end{array}$$

On définit $f, g, f * g$ par leurs graphes implicites respectifs en posant :

$$\begin{aligned} \text{graphe}(f) &= \sum u \otimes f(u) \\ \text{graphe}(g) &= \sum v \otimes g(v) \\ \text{graphe}(f * g) &= \sum (u \sqcup v) \otimes f(u)g(v) \end{aligned}$$

En particulier,

$$(f * g)(w) = \sum_{uv=w} \langle u \sqcup v | w \rangle f(u)g(v)$$

*L'élément neutre e pour ce produit de convolution vérifie :

$$\forall f, \quad f * e = e * f = f$$

Posons $E = \text{graphe}(e)$ Alors, $E = \epsilon \otimes \epsilon$ et vérifie:

$$\begin{aligned} E(\epsilon) &= \epsilon \\ E(v) &= 0, \quad \forall v \neq \epsilon \end{aligned}$$

*L'identité est définie par :

$$\begin{aligned} id : \mathbb{R}\langle Z \rangle &\longrightarrow \mathbb{R}\langle Z \rangle \\ u &\longrightarrow id(u) = u \end{aligned}$$

de graphe implicite

$$\text{graphe}(id) = \sum u \otimes u$$

*On note par \tilde{u} , l'image miroir de u . Soit α défini par

$$\begin{aligned} \alpha(u) &= (-1)^{|u|} \tilde{u} \\ \alpha(s) &= \sum_u \langle s | u \rangle \alpha(u) \\ &= \sum_u (-1)^{|u|} \langle s | u \rangle \tilde{u} \end{aligned}$$

Alors α est un antipode pour ce produit de convolution i.e. vérifie:

$$\alpha * id = id * \alpha = E$$

La preuve se fait plus facilement en utilisant les notations suivantes:

Notons $I = \text{graphe}(id)$ et $A = \text{graphe}(\alpha)$ les graphes implicites respectifs de id et de α . Alors on remarque que

$$(u \otimes v) \triangleright z = u \otimes (v \triangleright z)$$

et

$$z \triangleleft (\sum u \otimes f(u)) = \sum v \otimes f(vz)$$

c'est à dire

$$(z \triangleleft f)(v) = f(vz)$$

Preuve

Montrons que

$$A * I = \epsilon \otimes \epsilon.$$

On remarque que

$$\begin{aligned} A \triangleright z &= -A(1 \otimes z) \\ I \triangleright z &= (1 \otimes z)I \end{aligned}$$

Donc,

$$\begin{aligned} (A * I) \triangleright z &= (A \triangleright z) * I + A * (I \triangleright z) \\ &= -A(1 \otimes z) * I + A * (1 \otimes z)I \\ &= -A * (1 \otimes z)I + A * (1 \otimes z)I \\ &= 0 \end{aligned}$$

Il en résulte que

$$\begin{aligned} A * I &= \epsilon \otimes \epsilon + \sum_z (A * I) \triangleright z \\ &= \epsilon \otimes \epsilon \end{aligned}$$

Fin de preuve

► Coproduit de factorisation

On définit le coproduit de factorisation Φ par dualité du produit de Cauchy en posant:

$$\langle \Phi(w) | u \otimes v \rangle = \langle u.v | w \rangle$$

ce qui revient à:

$$\Phi(w) = \sum_{u,v \in Z^*, u.v=w} u \otimes v$$

Remarques

1) On prolonge Φ pour le mélange et on montre que Φ est un morphisme pour le mélange c'est à dire [3]:

$$\Phi(u \sqcup v) = \Phi(u) \sqcup \Phi(v)$$

2) On définit une dérivation pour le produit de concaténation en posant

$$\langle \partial_z(u) | w \rangle = \langle u | z \sqcup w \rangle$$

On a bien

$$\partial_z(uv) = \partial_z(u)v + u\partial_z(v)$$

► Algèbre de Hopf de factorisation

Muni des opérations Φ et \sqcup , $\mathbb{R}\langle\langle Z \rangle\rangle$ apparaît comme la complétée de l'algèbre de Hopf de factorisation $\mathbb{R}\langle Z \rangle$.

Comme pour l'algèbre de Hopf de décomposition, on définit:

$$f *_{\Phi} g = \sqcup \circ (f \otimes g) \circ \Phi$$

selon le schéma:

$$\begin{array}{ccc} & f & \\ \mathbb{R}\langle Z \rangle & \longrightarrow & \mathbb{R}\langle Z \rangle \\ & \otimes & \\ \mathbb{R}\langle Z \rangle & \longrightarrow & \mathbb{R}\langle Z \rangle \\ & g & \\ \Phi \uparrow & & \downarrow \sqcup \\ \mathbb{R}\langle Z \rangle & \longrightarrow & \mathbb{R}\langle Z \rangle \\ & f *_{\Phi} g & \end{array}$$

Son graphe implicite est :

$$f *_{\Phi} g = \sum_{u,v} uv \otimes (f(u) \sqcup g(v))$$

On définit l'élément neutre E_{Φ} dont on montre qu'il est égal à E .

On définit de même l'antipode α_{Φ} dont on montre qu'il est égal à α .

En effet, on peut démontrer cela en utilisant les graphes implicites que nous noterons A et I de α et de id définis dans le cadre du produit de convolution:

$$\begin{aligned} I &= \sum_u u \otimes u \\ A &= \sum_u u \otimes \alpha(u) \\ A *_{\Phi} I &= \sum_{u,v} uv \otimes (\alpha(u) \sqcup v) \\ &= \sum_{u,v} \alpha(u)v \otimes (u \sqcup v) \\ &= T(A * I) \end{aligned}$$

en notant $T(u \otimes v) = v \otimes u$.

Par conséquent, $\alpha_{\Phi} = \alpha$.

1.4.4 Topologie sur $\mathbb{R}\langle\langle Z \rangle\rangle$

► Longueur, ordre, degré

longueur d'un mot : Tout élément de Z^* est un mot et la longueur l d'un mot

$w = z_{i_1} \cdots z_{i_l}$ est le nombre de lettres qui le composent. On note cette longueur par $|w|$, et on note le nombre d'occurrences de la lettre z_i dans le mot w par $|w|_{z_i}$. On désigne par $Z^{<l}$, l'ensemble des mots de longueur inférieure strictement à l , et Z^l l'ensemble des mots de longueur l . Ainsi,

$$Z^* = \cup_l Z^l$$

Ordre d'une série : Rappelons que le support d'une série s est le langage

$$\text{Supp}(s) = \{w \in Z^*, \langle s|w \rangle \neq 0\}$$

On définit l'ordre d'une série s comme l'entier $o(s)$ tel que

$$\begin{cases} \text{Supp}(s) \cap Z^{<o(s)} &= 0 \\ \text{Supp}(s) \cap Z^{<o(s)+1} &\neq 0 \end{cases}$$

Degré d'un polynôme : Le degré d'un polynôme est la longueur maximale des mots de son support.

► **Topologie sur $\mathbb{R}\langle\langle Z \rangle\rangle$:**

Des détails figurent dans [2]

\mathbb{R} étant muni de la topologie discrète, on peut munir $\mathbb{R}\langle\langle Z \rangle\rangle$ de la topologie produit. Cette topologie peut être définie par une valuation ultramétrique:

$$\begin{aligned} \omega : \mathbb{R}\langle\langle Z \rangle\rangle \times \mathbb{R}\langle\langle Z \rangle\rangle &\rightarrow \mathbb{N} \cup \infty \\ (s, t) &\rightarrow \omega(s, t) = \inf \{n \in \mathbb{N}, \exists w \in Z^*, \\ &|w| = n, \langle s|w \rangle \neq \langle t|w \rangle\} \end{aligned}$$

Si $\rho \in \mathbb{R}$, $0 < \rho < 1$, alors l'application :

$$\begin{aligned} d : \mathbb{R}\langle\langle Z \rangle\rangle \times \mathbb{R}\langle\langle Z \rangle\rangle &\rightarrow \mathbb{R} \\ \text{satisfaisant } d(s, t) &= \rho^{\omega(s, t)} \end{aligned}$$

est une distance ultramétrique définissant la topologie précédente et $\mathbb{R}\langle\langle Z \rangle\rangle$ est complet pour cette distance.

► **Famille sommable, famille localement finie**

Soit $(s_i)_{i \in I}$ une famille de séries. Elle est **sommable** ssi il existe une série formelle s telle que

$$\begin{aligned} \forall \epsilon > 0, \exists I' \subset I, I' \text{ finie vérifiant} \\ \forall J \subset I, J \text{ finie, } J \supset I', \text{ alors } d\left(\sum_{j \in J} s_j, s\right) \leq \epsilon \end{aligned}$$

s est appelée somme de $(s_i)_{i \in I}$

Une famille $(s_i)_{i \in I}$ est dite **localement finie** ssi $\forall w \in Z^*$, il existe un nombre fini d'indices $i \in I$ tel que $\langle s_i | w \rangle \neq 0$.

Toute famille localement finie est sommable. En particulier, la famille

$$(\langle s | w \rangle w)_{w \in Z^*}$$

est localement finie, de somme s :

$$s = \sum_{w \in Z^*} \langle s | w \rangle w$$

Donc, $\mathbb{R}\langle Z \rangle$ est dense dans $\mathbb{R}\langle\langle Z \rangle\rangle$ qui est son séparé complété.

Une série formelle s est **propre** ssi $\langle s | \epsilon \rangle = 0$. Alors, $(s^n)_{n \in \mathbb{N}}$ est localement finie. En effet, pour tout mot w

$$n > |w| \text{ implique que } \langle s^n | w \rangle = 0$$

Par suite, cette famille est sommable. On désigne par s^* la somme de cette famille

$$s^* = \sum_{n \geq 0} s^n$$

que l'on appelle **étoile** de s .

Bibliographie

- [1] Hochschild G.P., *Basic Theory of Algebraic groups and Lie Algebras*, Springer Verlag, 1981.
- [2] Reutenauer C., *Séries formelles et algèbres syntactiques*, J. Algebra, vol. 66, pp. 448-483, 1980.
- [3] Reutenauer C., *Free Lie Algebras* , Oxford Science Publications, 1993.

Chapitre 2

Séries rationnelles

2.1 Introduction

Ce chapitre s'articule autour de trois parties : **série rationnelle**, **série reconnaissable**, **théorème de Schützenberger**. Ces notions sont bien connues [6], mais nous avons jugé intéressant de les aborder avec les concepts développés au chapitre précédent, et avec l'idée de disposer d'un outil de présentation pour le Calcul Formel.

- On rappelle que la notion de série rationnelle en variables non commutatives est définie comme une généralisation de la notion de série rationnelle en une variable: Les séries rationnelles sont les éléments de la clôture rationnelle de $\mathcal{R}\langle Z \rangle$, ensemble des polynômes.
- On sait d'après le chapitre précédent que les séries reconnaissables sont caractérisées par la finitude de leur rang (théorème de Schützenberger-Fliess). Notre intention est de proposer une présentation des séries rationnelles, à partir des concepts déjà développés, qui soit adaptée au Calcul Formel. Après une introduction sur les \mathcal{R} -automates (définition, graphe, représentation matricielle), nous proposons d'associer à toute série formelle s , son " \mathcal{R} -automate", dont les états sont les résiduelles de s et les actions sont les actions des lettres sur les résiduelles. Notre présentation d'une série reconnaissable est alors donnée par le \mathcal{R} -automate fini associé.
- Enfin, on rappelle l'énoncé du théorème de Schützenberger établissant l'équivalence de la reconnaissabilité et de la rationalité des séries formelles propres et on en propose une nouvelle démonstration s'appuyant sur des calculs de résiduelles.

2.2 Séries rationnelles en une variable

Dans ce cas, l'alphabet Z est réduit à une seule lettre z . Un élément de $\mathbb{R}\langle\langle Z \rangle\rangle$ s'écrit alors

$$s = \sum_{n \geq 0} a_n z^n$$

Proposition 2.2.1 *Une série $s = \sum_{n \geq 0} a_n z^n$ est rationnelle si et seulement si il existe des polynômes P et Q dans $\mathbb{R}\langle Z \rangle$ avec $Q(0) = 1$ tels que s soit égale au développement en série de la fraction rationnelle $P(z)/Q(z)$*

On peut écrire $P(z)/Q(z)$ sous la forme $R(z) + P_1(z)/Q(z)$ avec $\deg(P_1) < \deg(Q)$, $R(z)$ étant la partie entière de $P(z)/Q(z)$.

Citons le résultat classique selon lequel une série s est rationnelle si et seulement si elle vérifie une relation de récurrence linéaire.

Ses coefficients vérifient:

$$a_{n+k} + \alpha_1 a_{n+k-1} + \dots + \alpha_k a_n = 0 \quad \forall n \in \mathbb{N}$$

ce qui traduit une relation sur les résiduelles de la série s .

2.3 Séries rationnelles non commutatives

2.3.1 Opérations rationnelles

Les notations et résultats figurent dans [6].

Rappelons (cf chapitre 1), que l'étoile d'une série propre $s \in \mathbb{R}\langle\langle Z \rangle\rangle$ est la série s^* définie par

$$s^* = \sum_{n \geq 0} s^n$$

Remarquons que $s^* = 1 + s.s^* = 1 + s^*.s$.

Désignons par s^+ la série

$$s^+ = \sum_{n \geq 1} s^n$$

On en déduit que

$$s^* = 1 + s^+ \quad s^+ = s.s^* = s^*.s$$

Alors, s^* apparaît comme l'inverse de $1 - s$ puisque

$$s^*.(1 - s) = s^* - s^*.s = s^* - s^+ = 1$$

$$(1 - s).s^* = s^* - s.s^* = s^* - s^+ = 1$$

Lemme 2.3.1 *Soient t et u deux séries formelles, t étant supposée propre. Alors, l'unique solution s de l'équation*

$$s = u + t.s \quad (\text{resp.} \quad s = u + s.t)$$

est la série

$$s = t^*.u \quad (\text{resp.} \quad s = u.t^*)$$

Preuve :

De $s = u + t.s$, on déduit

$$s = u + t.(u + t.s) = u + t.u + t^2.s$$

et récursivement,

$$s = (1 + t + \dots + t^n).u + t^{n+1}.s$$

Comme t est supposée propre, on a

$$\lim_{n \rightarrow \infty} t^n = 0 \quad \text{et} \quad \lim_{n \rightarrow \infty} \sum_{0 \leq i \leq n} t^i = t^*$$

Donc, $s = t^*.u$ par passage à la limite.

Fin de preuve

Ce lemme permet de résoudre les systèmes d'équations linéaires à gauche (resp. à droite)

Definitions 2.3.1 • *Les opérations rationnelles de $\mathbb{R}\langle\langle Z \rangle\rangle$ sont la somme, le produit de Cauchy, le produit externe de \mathbb{R} sur $\mathbb{R}\langle\langle Z \rangle\rangle$ et l'étoile.*

- *Une partie de $\mathbb{R}\langle\langle Z \rangle\rangle$ est rationnellement close ssi elle est close pour les opérations rationnelles.*
- *On appelle clôture rationnelle de E , partie de $\mathbb{R}\langle\langle Z \rangle\rangle$, la plus petite partie rationnellement close contenant E .*
- *Une série formelle est rationnelle si et seulement si elle appartient à la clôture rationnelle de $\mathbb{R}\langle Z \rangle$.*

2.4 Séries reconnaissables

Rappelons qu'une série $s \in \mathbb{R}\langle\langle Z \rangle\rangle$ est reconnaissable si et seulement si elle vérifie une des propriétés équivalentes du théorème de Kleene (cf chapitre 1, proposition 3.1)

Remarque :

La première propriété signifie qu'il existe deux familles finies $(g_i)_{i \in I}$ et $(d_i)_{i \in I}$ de $\mathbb{R}\langle\langle M \rangle\rangle$ telles que :

$$\langle f | x.y \rangle = \sum_{i \in I} \langle g_i | x \rangle \langle d_i | y \rangle$$

2.4.1 Matrices de Hankel

Rappelons que si m est la multiplication, m^ la transposée de m , alors on appelle présentation de Hankel de s (cf chapitre 1), notée $H(s)$, $m^*(s)$ définie par :

$$\langle m^*(s) | x, y \rangle = \langle s | x.y \rangle$$

Rappelons encore qu'on appelle matrice de Hankel de la série s (cf chapitre 1), la description matricielle de $m^(s)$. Alors, précisons que

- Ses lignes L_x sont indicées par les mots x de Z^* ,
- Ses colonnes C_y sont indicées par les mots y de Z^* ,
- Le terme figurant en (x, y) est :

$$\langle m^*(s) | x, y \rangle$$

*Rappelons enfin qu'on appelle rang d'une série s , le rang de sa matrice de Hankel.

Dans le cadre des séries formelles non commutatives, une relecture du théorème de Kleene est le théorème de Schützenberger-Fliess [1]:

Théorème 3 *Une condition nécessaire et suffisante pour qu'une série $s \in \mathbb{R}\langle\langle Z \rangle\rangle$ soit reconnaissable est qu'elle soit de rang fini N .*

Une présentation intéressante d'une série reconnaissable s est celle donnée par un système de vecteurs lignes (resp. de vecteurs colonnes) formant une base de l'espace vectoriel engendré par les vecteurs lignes (resp. les vecteurs colonnes) de la matrice

de Hankel de s , et les relations de dépendance linéaire satisfaites par les autres vecteurs.

Exemple 1: [3]

Série formelle définie sur l'alphabet $Z = \{z_0, z_1\}$ par

$$s = \sum_{w \in Z^*} (|w|_{z_0} - |w|_{z_1})w$$

Il s'agit d'une série reconnaissable de rang 2. Sa matrice de Hankel est donnée par le tableau suivant:

	ϵ	z_0	z_1	z_0^2	z_0z_1	z_1z_0	$z_1^2 \dots$
ϵ	0	1	-1	2	0	0	-2...
z_0	1	2	0	3	1	1	-1...
z_1	-1	0	-2	1	-1	-1	-3...
z_0^2	2	3	1	4	2	2	0...
z_0z_1	0	1	-1	2	0	0	-2...
z_1z_0	0	1	-1	2	0	0	-2...
z_1^2	-2	-1	-3	0	-2	-2	-4...
...

La matrice $((C_\epsilon, C_{z_0}), (L_\epsilon, L_{z_0}))$ située dans l'angle supérieur gauche de la matrice de Hankel est de déterminant non nul.

Toutes les colonnes C_w et toutes les lignes L_w de cette matrice de Hankel apparaissent comme des combinaisons linéaires de ces deux colonnes, de ces deux lignes. Citons :

$$\begin{aligned} C_{z_1} &= 2C_\epsilon - C_{z_0} \\ C_{z_0^2} &= -C_\epsilon + 2C_{z_0} \\ C_{z_0z_1} &= C_\epsilon \\ C_{z_1z_0} &= C_\epsilon \\ C_{z_1^2} &= 3C_\epsilon - 2C_{z_0} \end{aligned}$$

Les autres relations linéaires relatives aux colonnes indiquées par des mots de longueur supérieure à deux se déduisent de celles-la.

2.4.2 Automates finis

► Définition d'un automate

Définition 2.4.1 Une action à droite d'un monoïde M sur un espace vectoriel Q est la donnée de

$$q * m \in Q, \quad \forall q \in Q, \quad \forall m \in M$$

telle que

- $q \rightarrow q * m$ soit une application linéaire de Q dans Q
- $q * mn = (q * m) * n$

Définition 2.4.2 On appelle automate à droite (à coefficients dans \mathbb{R}) un quintuplet

$$A = \{Z, Q, *, I, h\}$$

Z étant l'alphabet, Q l'espace vectoriel sur \mathbb{R} appelé l'espace des états de l'automate, $*$ une action à droite de Z^* sur Q , h la fonction d'observation.

On définit de même un automate à gauche par la donnée d'une action à gauche. Dans toute la suite, les automates considérés seront à coefficients dans \mathbb{R} .

A toute série formelle s , on peut associer un automate en général infini

$$A_s = \{Z, Q, *, I, h\}$$

tel que

- l'alphabet $Z = \{z_0, \dots, z_m\}$,
- l'espace Q des états est l'espace vectoriel engendré par les résiduelles à droite $s \triangleright w$ de s ,
- l'état $I = s$,
- l'action à droite est définie par les résiduelles à droite

$$(s \triangleright w) * u = (s \triangleright w) \triangleright u = s \triangleright wu$$

- la fonction d'observation h est définie, pour toute série r par

$$h(r) = \langle r | \epsilon \rangle$$

Ainsi,

$$h(s \triangleright w) = \langle s \triangleright w | \epsilon \rangle = \langle s | w \rangle$$

On définit de même l'automate infini à gauche de s en utilisant les résiduelles à gauche.

Exemple 2:

Série génératrice associée à l'équation électrique

$$\dot{v} = -k_1 v - k_2 v^2 + i(t)$$

Cette série a pour expression :

$$s = \sum_{l \geq 0} \sum_{i_j=0}^1 A_{i_1} \cdots A_{i_l} q_l z_{i_1} \cdots z_{i_l}$$

en posant

$$\begin{cases} A_0 &= (-k_1 q - k_2 q^2) \frac{d}{dq} \\ A_1 &= \frac{d}{dq} \end{cases}$$

Avec les notations suivantes (pour $q_0 = q(0)$):

$$\begin{cases} a &= -k_1 q_0 - k_2 q_0^2 \\ b &= -k_1 - 2k_2 q_0 \\ c &= -2k_2 \end{cases}$$

On remarque qu'il existe certaines relations entre les champs de vecteurs A_i :

$$A_1^n A_0 = a A_1^{n+1} + n b A_1^n + \frac{n(n-1)}{2} c A_1^{n-1}$$

Cette relation entre les champs de vecteurs induit une relation sur les résiduelles de s :

$$s \triangleright (z_1^n z_0) = a s \triangleright z_1^{n+1} + n b s \triangleright z_1^n + \frac{n(n-1)}{2} c s \triangleright z_1^{n-1}$$

On peut donc maintenant préciser un automate infini A associé à s :

- L'alphabet $Z = \{z_0, z_1\}$
- L'espace des états $Q = \{s \triangleright z_1^n\}_{n \geq 0}$
- L'état initial $I = s$
- L'action $*$ (à droite) de la lettre z_0 sur l'état $s \triangleright z_1^n$ satisfait :

$$(z_0 * s) \triangleright z_1^n = a s \triangleright z_1^{n+1} + n b s \triangleright z_1^n + \frac{n(n-1)}{2} c s \triangleright z_1^{n-1}$$

L'action $*$ (à droite) de la lettre z_1 sur l'état $s \triangleright z_1^n$ satisfait :

$$(z_1 * s) \triangleright z_1^n = s \triangleright z_1^{n+1}$$

- La fonction d'observation h est définie par

$$\begin{aligned} h(s) &= q_0 \\ h(s \triangleright z_1) &= 1 \end{aligned}$$

Ainsi, on peut calculer aisément tout coefficient de la série s :

$$\langle s | z_{i_1} \cdots z_{i_l} \rangle = h(z_{i_1} \cdots z_{i_l} * s)$$

en particulier, par exemple

$$\langle s | z_0 z_1 z_0^2 \rangle = a(2bc + cb) = 3abc$$

► **Présentation d'un automate par son graphe**

On peut aussi définir un automate par sa représentation sous forme d'un graphe orienté valué. A est alors présenté comme un quintuplet

$$A = \{Z, S, \tau, I, F\}$$

Z étant l'alphabet, S l'ensemble des sommets du graphe, I un ensemble de sommets initiaux valués, τ un ensemble d'arcs de transitions valués, et F un ensemble de sommets terminaux valués.

A toute série formelle s , on peut associer un automate en général infini de présentation graphique

$$A_g = \{Z, S, \tau, I, F\}$$

tel que

- l'alphabet $Z = \{z_0, \dots, z_m\}$,
- l'ensemble des sommets $S = (w \triangleleft s)_{w \in UCZ^*}$ (resp. $Q = (s \triangleright w)_{w \in VCZ^*}$) est un ensemble de résiduelles à gauche (resp. à droite) de s formant un système générateur de l'ensemble $\{w \triangleleft s\}_{w \in Z^*}$, (resp. $\{s \triangleright w\}_{w \in Z^*}$),
- le sommet initial $I = s$,
- tout arc de transition entre sommets, appartenant à τ , est étiqueté par une lettre $z_i \in Z$ et valué par le coefficient figurant dans la relation de dépendance linéaire.
- les sommets terminaux de F sont les $w_{i_j} \triangleleft s$ (resp. $s \triangleright w_{i_j}$) avec la valuation $\langle w_{i_j} \triangleleft s | \epsilon \rangle \neq 0$ (resp. $\langle s \triangleright w_{i_j} | \epsilon \rangle \neq 0$)

Exemple 2:

L'automate infini A associé à s a la présentation graphique suivante:

- L'alphabet $Z = \{z_0, z_1\}$
- L'ensemble des sommets $S = \{s \triangleright z_1^n\}_{n \geq 0}$
- L'état initial $I = s$ de valuation q_0
- Les transitions étiquetées par z_1 sont de la forme :
 $s \triangleright z_1^n \rightarrow s \triangleright z_1^{n+1}$
et sont de valuation 1.
- Les transitions étiquetées par z_0 sont d'une des trois formes :

- $s \triangleright z_1^n \rightarrow s \triangleright z_1^{n+1}$
et sont de valuation a .

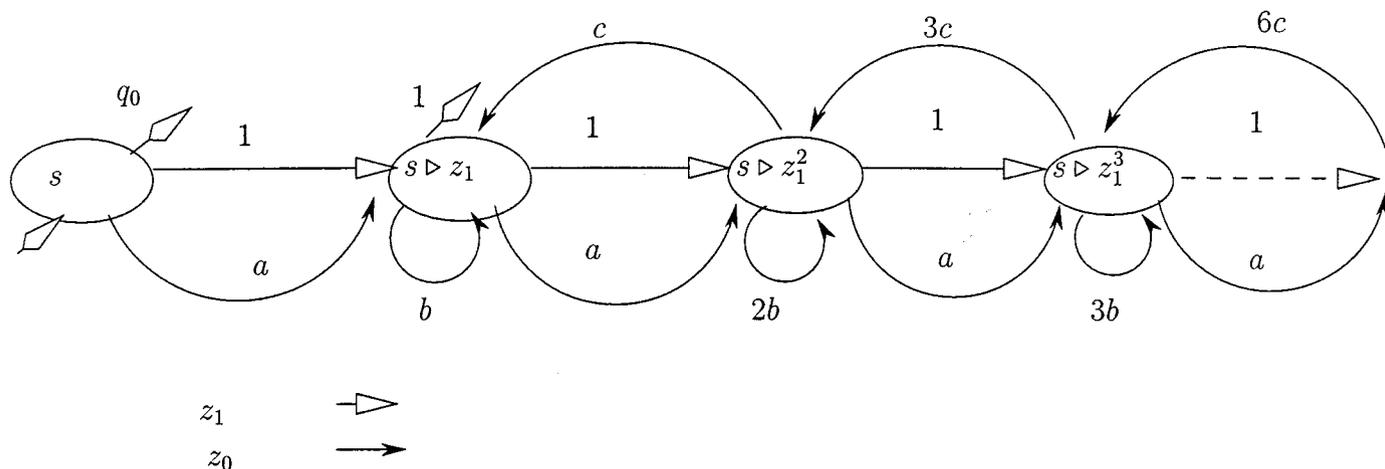


FIG. 2.1 – graphe de l'exemple 2

- $s \triangleright z_1^n \rightarrow s \triangleright z_1^n$
et sont de valuation nb .
- $s \triangleright z_1^{n+1} \rightarrow s \triangleright z_1^n$
et sont de valuation $\frac{n(n-1)}{2}c$.

• Les états terminaux sont les $s \triangleright z_1^n$ tels que $\langle s \triangleright z_1^n | \epsilon \rangle \neq 0$
Son graphe est donné par la figure 2.1.

► **Présentation matricielle d'un automate**

G.Jacob [4] propose une présentation matricielle pour un automate sous la forme :

$$A_m = \{Z, E, \gamma, \mu, \lambda\}$$

tel que Z est l'alphabet, E est l'espace vectoriel engendré par les états, γ est l'état initial, $\mu(z_i)$ est l'application linéaire associée à z_i , λ est la forme linéaire de sortie.

Définition 2.4.3 On appelle automate matriciel à droite (resp. à gauche) un tel automate:

$$A_m = \{Z, E, \gamma, \mu, \lambda\}$$

A toute série s , on peut associer un \mathbb{R} -automate matriciel tel que:

- l'alphabet $Z = \{z_0, \dots, z_m\}$,
- E est l'espace vectoriel engendré par les résiduelles à droite (resp. à gauche) de s . Une base de cet espace vectoriel est

$$\{s \triangleright w_l\}_{w_l}, (\text{resp. } \{w_l \triangleleft s\}_{w_l})$$

- Dans cette base, γ représente l'état initial :

$$\gamma = \begin{pmatrix} 1 \\ 0 \\ \cdot \\ \cdot \\ \cdot \end{pmatrix}$$

- Dans cette base, $\mu(z_i)$ est l'application linéaire associée à l'action de la lettre z_i sur les vecteurs de base.
- Dans cette base, λ est la forme linéaire associée à la fonction d'observation h .

Exemple 2:

L'automate infini A associé à s a la présentation matricielle suivante:

- L'alphabet $Z = \{z_0, z_1\}$
- L'espace vectoriel E engendré par les résiduelles à droite possède la base infinie

$$\{s \triangleright z_1^n\}_{n \geq 0}$$

- Dans cette base, γ représente l'état initial :

$$\gamma = \begin{pmatrix} 1 \\ 0 \\ \cdot \\ \cdot \\ \cdot \end{pmatrix}$$

- Dans cette base, les matrices associées aux actions de z_0 et z_1 sont les suivantes :

Pour $\mu(z_0)$

$$\mu(z_0) = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & \dots \\ a & b & c & 0 & 0 & \dots \\ 0 & a & 2b & 3c & 0 & \dots \\ 0 & 0 & a & 3b & 6c & \dots \\ 0 & 0 & 0 & a & 4b & \dots \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \end{pmatrix}$$

Pour $\mu(z_1)$

$$\mu(z_1) = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & \dots \\ 1 & 0 & 0 & 0 & 0 & \dots \\ 0 & 1 & 0 & 0 & 0 & \dots \\ 0 & 0 & 1 & 0 & 0 & \dots \\ 0 & 0 & 0 & 1 & 0 & \dots \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \end{pmatrix}$$

- La forme linéaire λ associée à la fonction de sortie est :

$$\lambda = (q_0 \quad 1 \quad 0 \quad \dots)$$

► **Définition d'un automate fini**

Définition 2.4.4 *Un automate A est fini si et seulement si l'espace Q des états est de dimension finie.*

D'après le théorème de Fliess, une présentation intéressante d'une série reconnaissable s est celle donnée par un automate A fini associé.

Exemple 1 : [3]

Série formelle définie sur l'alphabet $Z = \{z_0, z_1\}$ par

$$s = \sum_{w \in Z^*} (|w|_{z_0} - |w|_{z_1})w$$

Cet exemple a déjà été traité plus haut. On sait que cette série s est reconnaissable de rang 2.

1) On peut préciser un automate fini associé :

- L'alphabet $Z = \{z_0, z_1\}$
- L'espace des états $Q = \{s, z_0 \triangleleft s\}$
- L'état initial $I = s$
- L'action de z_0 est donnée par

$$\begin{cases} z_0 * s & = z_0 \triangleleft s \\ z_0 * z_0 \triangleleft s & = -s + 2z_0 \triangleleft s \end{cases}$$

Et l'action de z_1 est donnée par

$$\begin{cases} z_1 * s & = 2s - z_0 \triangleleft s \\ z_1 * z_0 \triangleleft s & = s \end{cases}$$

- La fonction d'observation h est la projection sur l'état $z_0 \triangleleft s$.

2) La présentation par son graphe de cet automate à gauche est la suivante :

$$A_g = \{Z, S, \tau, I, F\}$$

- Les transitions étiquetées par z_1 sont de la forme :

- $s \rightarrow s$ de valuation 2
- $s \rightarrow z_0 \triangleleft s$ de valuation -1
- $z_0 \triangleleft s \rightarrow s$ de valuation 1

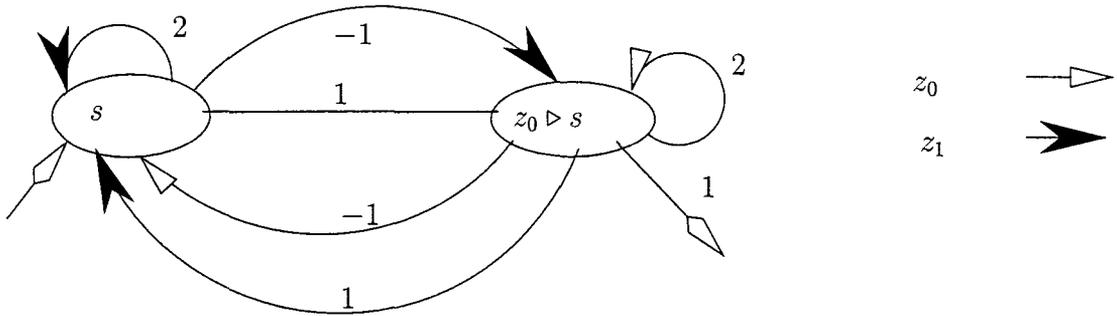


FIG. 2.2 – graphe de l'exemple 1

• Les transitions étiquetées par z_0 sont de la forme :

- $z_0 \triangleleft s \rightarrow z_0 \triangleleft s$ de valuation 2
- $s \rightarrow z_0 \triangleleft s$ de valuation 1
- $z_0 \triangleleft s \rightarrow s$ de valuation -1

• Le sommet terminal est $z_0 \triangleleft s$ avec la valuation 1
Son graphe est donné par la figure 2.2.

► **\mathbb{R} -automate matriciel**

Dans le cas d'une série s reconnaissable, l'espace vectoriel E engendré par les résiduelles de s est de dimension finie N . On sait alors qu'une présentation de s est le \mathbb{R} -automate matriciel A_m défini plus haut.

$$A_m = \{Z, N, \gamma, \lambda, \mu\}$$

avec

$$\begin{cases} N & = \dim(E) \\ \gamma & \in \mathbb{R}^N \\ \lambda & \in {}^t(\mathbb{R}^N) \\ \mu & : Z^* \rightarrow \mathbb{R}^{N \times N} \end{cases}$$

2.5 Théorème de Schützenberger

2.5.1 Représentations linéaires

► Représentation linéaire d'une série reconnaissable

La définition d'une série reconnaissable est habituellement donnée à l'aide d'une représentation linéaire. Cette définition apparaît comme une relecture d'une des propriétés équivalentes du théorème de Kleene.

Définition 2.5.1 Une série formelle $s \in \mathbb{R}\langle\langle Z \rangle\rangle$ est dite reconnaissable ssi il existe un entier $N \geq 1$, un morphisme de monoïdes :

$$\mu : Z^* \rightarrow \mathbb{R}^{N \times N}$$

et deux matrices

$$\lambda \in \mathbb{R}^{1 \times N}, \gamma \in \mathbb{R}^{N \times 1}$$

telles que pour tout mot $w \in Z^*$,

$$\langle s|w \rangle = \lambda(\mu w)\gamma$$

Le triplet (λ, μ, γ) est alors appelé **représentation linéaire de s** et sa dimension est N .

Montrons que cette définition est équivalente à l'une des propriétés équivalentes du théorème de Kleene, par exemple la deuxième, relative à la finitude de l'espace vectoriel engendré par les résiduelles à droite. Pour cela, démontrons le lemme :

► Lemme

Lemme 2.5.1 Si une série formelle $s \in \mathbb{R}\langle\langle Z \rangle\rangle$ est résiduellement finie, alors elle est reconnue par un automate fini.

Preuve :

• Montrons que si s vérifie $\exists(G_1, \dots, G_P) \in Z^*$ tels que $(s \triangleright G_1, \dots, s \triangleright G_P)$ engendrent l'espace vectoriel des résiduelles à droite de s , alors s vérifie la définition donnée plus haut.

Pour établir cela, nous construisons l'automate des résiduelles de s .

- D'après le théorème de la base incomplète, il existe une base de l'espace des résiduelles de s contenant s . Soit $(R_i)_{1 \leq i \leq N}$ cette base. Alors, cette famille définit l'espace d'état et sa dimension N .

- L'état initial est égal à s . Cela définit la matrice γ présentée dans la définition précédente.
- L'action à droite est définie par :

$$R_i \triangleright z = \sum_{k=1}^N \alpha_{i,z}^k R_k \quad \forall z \in Z$$

puisque la famille $(R_k)_{1 \leq k \leq N}$ forme une base de l'espace des résiduelles. Cela définit $\forall z \in Z$, une application linéaire

$$\triangleright z : \mathbb{R}^N \rightarrow \mathbb{R}^N$$

telle que par définition

$$R_i \triangleright uv = (R_i \triangleright u) \triangleright v$$

Cela définit bien le morphisme de monoïde μ de la définition précédente.

- La fonction d'observation $h : \mathbb{R}^N \rightarrow \mathbb{R}$ est telle que

$$h(s \triangleright w) = \langle s \triangleright w | \epsilon \rangle = \langle s | w \rangle$$

Cela définit la matrice λ présentée dans la définition précédente. On a alors

$$\begin{aligned} \langle s | w \rangle &= h(s \triangleright w) \\ \langle s | w \rangle &= h\left(\sum_{i=1}^N \lambda_i R_i\right) \\ \langle s | w \rangle &= \sum_{i=1}^N \lambda_i h(R_i) \\ \langle s | w \rangle &= \sum_{i=1}^N \lambda_i \langle R_i | \epsilon \rangle \end{aligned}$$

Il en résulte que s vérifie bien la définition de la reconnaissabilité par représentation linéaire.

Fin de preuve

- Pour montrer que si la série s possède une représentation linéaire de dimension N , alors l'espace vectoriel des résiduelles à droite de s est de dimension N , il suffit de raisonner sur la matrice de Hankel de s .

► Représentation minimale d'une série reconnaissable

M.Fliess propose une construction effective de la représentation minimale d'une série rationnelle par le théorème [2]:

Théorème 4 Une série formelle $s \in \mathbb{R}\langle\langle Z \rangle\rangle$ est reconnaissable si et seulement si elle est de rang fini N . Elle est alors reconnue par un \mathbb{R} -automate matriciel $M =$

$(N, \gamma, \lambda, \mu)$. On peut alors déterminer deux ensembles de mots $\{g_i\}_{1 \leq i \leq N}$ et $\{d_j\}_{1 \leq j \leq N}$ de longueur inférieure strictement à N , tels que l'application χ de Z^* dans $\mathbb{R}^{N \times N}$ définie par :

$$(\chi(w))_{i,j} = \langle s | g_i . w . d_j \rangle$$

vérifie $\chi(w) = \chi(1)\mu(w)$ et que $\chi(1)$ soit inversible.

Remarques

1) Ce théorème explicite le procédé de calcul d'une représentation linéaire réduite d'une série reconnaissable, proposé par le théorème de présentation minimale.

2) Tout autre \mathbb{R} -automate matriciel de dimension N reconnaissant s , se déduit de M par changement de base.

Ce résultat peut être précisé comme suit [6]:

Proposition 2.5.1 Soit s une série reconnaissable de rang N . Alors, il existe une partie préfixielle P et une partie suffixielle S toutes deux de cardinal N , telles que :

$$\det(\langle s | p\sigma \rangle)_{p \in P, \sigma \in S} \neq 0$$

Exemple 1: [3]

Série formelle définie sur l'alphabet $Z = \{z_0, z_1\}$ par

$$s = \sum_{w \in Z^*} (|w|_{z_0} - |w|_{z_1})w$$

Nous savons que cette série est reconnaissable de rang 2, que les 2 premières lignes et les 2 premières colonnes de sa matrice de Hankel permettent de définir les ensembles de mots $\{g_i\}_{1 \leq i \leq N}$ et $\{d_j\}_{1 \leq j \leq N}$ proposés par le théorème précédent.

En exploitant les relations de dépendances linéaires, concernant les transitions étiquetées par z_0 :

$$\begin{aligned} C_{z_0} &= & +C_{z_0} \\ C_{z_0^2} &= -C_\epsilon & +2C_{z_0} \end{aligned}$$

il vient :

$$\mu(z_0) = \begin{pmatrix} 0 & -1 \\ 1 & 2 \end{pmatrix}$$

En exploitant les relations de dépendances linéaires, concernant les transitions étiquetées par z_1 :

$$\begin{aligned} C_{z_1} &= 2C_\epsilon - C_{z_0} \\ C_{z_1 z_0} &= C_\epsilon \end{aligned}$$

il vient :

$$\mu(z_1) = \begin{pmatrix} 2 & 1 \\ -1 & 0 \end{pmatrix}$$

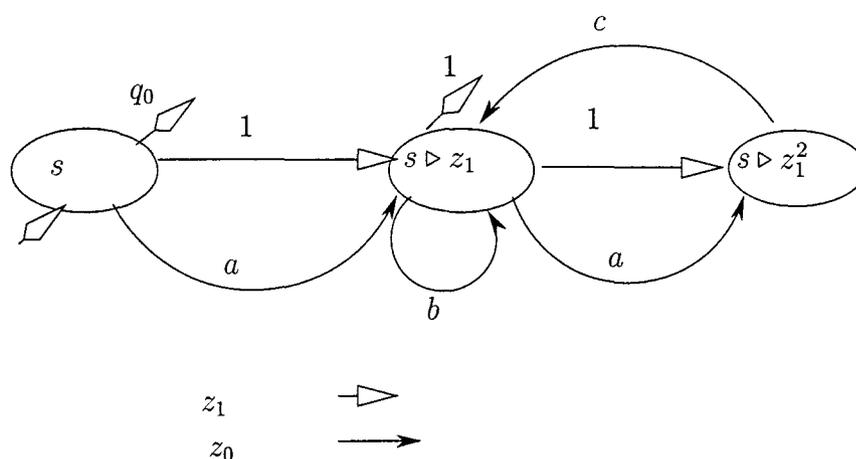


FIG. 2.3 – graphe de l'exemple 2

Le vecteur initial est :

$$\gamma = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

et le covecteur est :

$$\lambda = (\langle s|\epsilon \rangle \quad \langle z_0 \triangleleft s|\epsilon \rangle)$$

2.5.2 Systèmes d'équations linéaires à droite

Lemme 2.5.2 *Toute série $s \in \mathbb{R}\langle\langle Z \rangle\rangle$ reconnue par un automate fini est solution d'un système linéaire à droite.*

Ce lemme étant relativement trivial, nous nous contentons de l'illustrer par un exemple. Reprenons l'exemple 2 de l'équation électrique en tronquant l'automate infini associé aux mots de longueur inférieure ou égale à 3, dont le graphe est donné par la figure 2.3.

Alors il vient le système d'équations linéaires à droite suivant:

$$\begin{cases} s & = q_0 + z_1(s \triangleright z_1) + az_0(s \triangleright z_1) \\ (s \triangleright z_1) & = 1 + z_1(s \triangleright z_1^2) + bz_0(s \triangleright z_1) + az_0(s \triangleright z_1^2) \\ (s \triangleright z_1^2) & = cz_0(s \triangleright z_1) \end{cases}$$

2.5.3 Expressions rationnelles

Lemme 2.5.3 *Toute série $s \in \mathbb{R}\langle\langle Z \rangle\rangle$, solution d'un système linéaire à droite est rationnelle.*

Afin d'illustrer ce lemme, reprenons l'exemple 2 et résolvons le système linéaire à droite posé :

$$\begin{aligned}(s \triangleright z_1) &= 1 + cz_1z_0(s \triangleright z_1) + bz_0(s \triangleright z_1) + acz_0^2(s \triangleright z_1) \\(s \triangleright z_1) &= 1 + (cz_1z_0 + bz_0 + acz_0^2)(s \triangleright z_1) \\(s \triangleright z_1) &= (cz_1z_0 + bz_0 + acz_0^2)^*\end{aligned}$$

Et donc

$$s = q_0 + (z_1 + az_0)(cz_1z_0 + bz_0 + acz_0^2)^*$$

2.5.4 Le théorème de Schützenberger

Théorème 5 [7]

Une série formelle de $\mathbb{R}\langle\langle Z \rangle\rangle$ est rationnelle ssi elle est reconnaissable.

Preuve :

Compte-tenu des 3 lemmes des paragraphes précédents, il suffit de montrer que la classe des séries résiduellement finies contient les lettres et est stable par les opérations rationnelles.

- Les lettres sont résiduellement finies.
- La classe des séries résiduellement finies est stable par combinaison linéaire :

$$\begin{aligned}(\alpha s) \triangleright z &= \alpha(s \triangleright z) \\(s + t) \triangleright z &= s \triangleright z + t \triangleright z\end{aligned}$$

Pour tout mot w ,

$$(s + t) \triangleright w = s \triangleright w + t \triangleright w$$

- La classe des séries résiduellement finies est stable par produit :

$$\begin{aligned}(st) \triangleright z &= \langle s | \epsilon \rangle t \triangleright z + (s \triangleright z)t \\(st) \triangleright zy &= \langle s | \epsilon \rangle t \triangleright zy + \langle s | z \rangle t \triangleright y + (s \triangleright zy)t\end{aligned}$$

- La classe des séries résiduellement finies est stable par l'étoile :

$$\begin{aligned}s^* &= 1 + ss^* \\s^* \triangleright z &= (s \triangleright z)s^* \\s^* \triangleright zy &= \langle s | z \rangle (s^* \triangleright y) + (s \triangleright zy)s^* \\s^* \triangleright zy &= \langle s | z \rangle (s \triangleright y)s^* + (s \triangleright zy)s^*\end{aligned}$$

Et enfin

$$\begin{aligned}(ts^*) \triangleright z &= \langle t | z \rangle s^* \triangleright z + (t \triangleright z)s^* \\(ts^*) \triangleright z &= \langle t | z \rangle (s \triangleright z)s^* + (t \triangleright z)s^*\end{aligned}$$

Fin de preuve

2.6 Algèbre syntaxique

Les notations et résultats suivants figurent dans [6]

Définition 2.6.1 $\forall s \in R\langle\langle Z \rangle\rangle$, on définit une application

$$\begin{aligned} \bar{s} : R\langle Z \rangle &\rightarrow R \\ P &\rightarrow \langle s|P \rangle \\ &= \sum_{w \in Z^*} \langle s|w \rangle \langle P|w \rangle \end{aligned}$$

en prolongeant s de Z^* à $\mathbb{R}\langle Z \rangle$.

Alors, on appelle idéal syntaxique de s , le plus grand idéal bilatère de $R\langle\langle Z \rangle\rangle$, contenu dans le noyau de \bar{s} . On le note I_s .

On définit l'algèbre syntaxique M_s de $s \in R\langle\langle Z \rangle\rangle$ par :

$$M_s = R\langle Z \rangle / I_s$$

On dispose alors d'une nouvelle caractérisation des séries reconnaissables [5]:

Théorème 6 *Une série formelle est reconnaissable si et seulement si son algèbre syntaxique est de rang fini.*

Bibliographie

- [1] Fliess M., *Matrices de Hankel*, J. Maths. Pur. Appl., vol.53, pp. 197-222, 1974.
- [2] Fliess M., *Un outil algébrique: les séries formelles non commutatives*, in "Mathematical System Theory" (G. Marchesini and S.K. Mitter Eds.), Lecture Notes Econom. Math. Syst., Springer Verlag, vol. 131, pp . 122-148, 1976.
- [3] Jacob G., *Réalisation des systèmes réguliers et séries génératrices non commutatives*, publication interne de l'équipe lilloise d'Informatique théorique, n° IT-26-80, 1980.
- [4] Jacob G., *Réalisation des systèmes réguliers (ou bilinéaires) et séries génératrices non commutatives*, Séminaire d'Aussois, RCP567, Outils et modèles mathématiques pour l'Automatique, l'Analyse des Systèmes, et le traitement du Signal (CNRS Landau), 1980.
- [5] Reutenauer C., *Séries formelles et algèbres syntactiques*, J. Algebra, vol. 66, pp. 448-483, 1980.
- [6] Berstel J., Reutenauer C., *Les séries rationnelles et leurs langages*, Masson, 1984.
- [7] Schützenberger M.P., *On the definition of a family of automata*, Inform. and Control, vol. 4, pp 245-270, 1961.

Chapitre 3

Approximation d'une série formelle par une série rationnelle

3.1 Introduction

Dans ce chapitre, nous poursuivons le double but de **généraliser à plusieurs variables non commutatives, la notion d'approximant de type Padé** d'une série formelle à une variable [1], et de proposer deux mises en œuvre différentes, issues des deux présentations d'une série formelle à plusieurs variables non commutatives, introduites au premier chapitre.

- Après des rappels sur les approximants rationnels classiques de séries à une variable: polynômes, approximants de type Padé, approximants de Padé, on propose une généralisation de la notion d'approximant de type Padé de série s en variables non commutatives.
- Deux techniques de calcul sont développées pour mettre en œuvre cette notion: L'une est basée sur des calculs à partir de la présentation de s par la matrice de Hankel. Elle fournit un approximant de type Padé de rang minimal. L'autre est obtenue à partir de la présentation de s par l'automate associé à ses résiduelles. Elle consiste à tronquer convenablement cet automate.

Des comparaisons des avantages et des inconvénients respectifs des deux méthodes sont présentées en conclusion. Il est à noter qu'elles fournissent l'une comme l'autre un outil intéressant pour le Calcul Formel.

3.2 Approximation d'une série formelle par une série rationnelle, dans le cas d'une variable

Les notations et les résultats de ce paragraphe sont extraits de [1, 3]
Le problème consiste à proposer un approximant g de la série

$$s = \sum_{n \geq 0} a_n z^n$$

à l'ordre k vérifiant :

$$\text{ord}(s - g) > k$$

Nous proposons ici pour g un polynôme puis une forme de Padé, un approximant de Padé ou de type Padé.

3.2.1 Par un polynôme

Dans ce cas, g est obtenu en tronquant la série s aux monômes de degré inférieur ou égal à k .

Des outils d'approximation rationnelle plus fins se sont révélés plus intéressants pour résoudre certains problèmes. Citons en physique, le problème des moments de Hausdorff, et en mathématiques le procédé d'accélération de la convergence appelé ϵ -algorithme qui utilisent un autre type d'approximant rationnel : l'approximant de Padé.

3.2.2 Par un approximant de Padé ou de type Padé

► Par un approximant de type Padé

l'idée est de construire une fraction rationnelle, dont le numérateur ait le degré $k - 1$ et le dénominateur le degré k , et qui coïncide avec s jusqu'au degré k [1].

La construction est la suivante:

- On définit une fonctionnelle linéaire ϕ agissant sur l'espace vectoriel des polynômes réels telle que

$$\phi(x^i) = a_i \quad \forall i = 0, 1, \dots$$

Alors $s = \phi((1 - xz)^{-1})$.

- Soit v un polynôme arbitraire de degré k :

$$v(x) = b_0 + b_1 x + \dots + b_k x^k.$$

Posons alors

$$w(z) = \phi\left(\frac{v(x) - v(z)}{x - z}\right)$$

Alors, w est un polynôme de degré $k - 1$. Posons

$$\begin{aligned}\tilde{v}(z) &= z^k v(z^{-1}) \\ \tilde{w}(z) &= z^{k-1} w(z^{-1})\end{aligned}$$

Alors $\tilde{w}(z)/\tilde{v}(z)$ est un approximant de type Padé de s noté $((k-1)/k)_s(z)$ vérifiant en particulier:

$$\tilde{w}(z)/\tilde{v}(z) - s(z) = O(z^k)$$

On peut construire un approximant de type Padé dont le numérateur et le dénominateur ont des degrés quelconques de la manière suivante:

$$s = a_0 + a_1 z + \cdots + a_n z^n + z^{n+1} r_n(z)$$

avec $r_n(z) = a_{n+1} + a_{n+2} z + \cdots$

Alors la série rationnelle

$$a_0 + a_1 z + a_n z^n + z^{n+1} ((k-1)/k)_{r_n}(z)$$

est un approximant de type Padé dont le dénominateur a pour degré k et le numérateur le degré $n + k$. On le note $((n+k)/k)_s(z)$. Cet approximant de type Padé vérifie en particulier:

$$((n+k)/k)_s(z) - s(z) = O(z^{n+k+1})$$

► Par un approximant de Padé

Deux approches différentes sont proposées pour définir un approximant de Padé. La première, est élémentaire et la seconde fait référence aux polynômes orthogonaux.

- **première approche** [3]: On peut définir l'approximant de Padé de s comme suit:

On appelle forme rationnelle, la classe de (P, Q) définie par

$$\text{classe}(P, Q) = \{(aP, aQ) / a \in \mathbb{R}^*\}$$

On appelle forme de Padé de s , une forme rationnelle (U_m, V_m) définie par la condition suivante:

$$\text{ord}(sV_m - U_m) \geq m + n + 1$$

Soit (P_m, Q_n) la forme réduite d'une forme de Padé (U_m, V_n) . Si cette forme réduite est une forme de Padé, alors la fraction rationnelle P_m/Q_n est appelée approximant de Padé.

Définissons le déterminant de Toeplitz de

$$s = \sum_{n \geq 0} a_n z^n$$

de taille n par :

$$C_n^m = \begin{vmatrix} a_m & a_{m-1} & \cdots & a_{m-n+1} \\ a_{m+1} & a_m & \cdots & \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ a_{m+n-1} & a_{m+n-2} & \cdots & a_m \end{vmatrix}$$

avec les conventions $a_k = 0$ si $k < 0$ et $C_0^m = 1$

Et définissons le déterminant de Hankel de

$$s = \sum_{n \geq 0} a_n z^n$$

de taille $n + 1$ par :

$$H_n^m = \begin{vmatrix} a_m & a_{m+1} & \cdots & a_{m+n} \\ a_{m+1} & a_{m+2} & \cdots & \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ a_{m+n} & a_{m+n+1} & \cdots & a_{m+2n} \end{vmatrix}$$

On établit les relations suivantes :

$$\begin{aligned} H_n^m &= -1^{n(n+1)/2} C_{n+1}^{m+n} \\ C_n^m &= -1^{n(n-1)/2} H_{n-1}^{m-(n-1)} \end{aligned}$$

L'approximant de Padé est noté $[m, n]$.

Citons le théorème de Baker :

Théorème 7 Une condition nécessaire et suffisante d'existence d'un approximant de Padé $[m, n]$ est qu'il existe au moins une forme de Padé (U_m, V_n) satisfaisant à la condition :

$$V_n(0) = 1$$

Citons une condition suffisante d'existence d'un approximant de Padé :

Théorème 8 Une condition suffisante d'existence d'un approximant de Padé $[m, n]$ de la série s est :

$$C_n^m(s) \neq 0$$

Donnons enfin l'énoncé du théorème suivant :

Théorème 9 *Soit une série inversible s . S'il existe une fraction rationnelle P_m/Q_n satisfaisant à*

$$\begin{aligned} \text{ord}(s - P_m Q_n^{-1}) &\geq m + n + 1 \\ \text{deg}(P_m) &\leq m, \quad \text{deg}(Q_n) \leq n \end{aligned}$$

Alors, elle s'identifie à l'approximant de Padé et l'on a :

$$s(z) - P_m(z)/Q_n(z) = O(z^{m+n+1})$$

- **seconde approche** [1]: Pour définir l'approximant de Padé, on reprend les notations des approximants de type Padé. L'idée est d'obtenir des approximants de type Padé qui coïncident avec la série s pour un ordre supérieur au degré du numérateur par un choix judicieux de v . On dispose alors du théorème:

Théorème 10 *Si v satisfait $\phi(x^i v(x)) = 0, \quad 0 \leq i \leq m - 1 \leq k - 1$, alors*

$$s(z) - ((k - 1)/k)_s(z) = \frac{z^{m+k}}{\bar{v}(z)} \phi\left(\frac{x^m v(x)}{1 - xz}\right) = O(z^{m+k})$$

Quand $m = k$, v est complètement déterminé car on dispose alors de k équations à k inconnues. Alors, si le déterminant de Hankel de taille k de s est non nul, l'approximant existe et n'est autre que l'approximant de Padé de s .

Notons $\phi^{(n+1)}$ la fonctionnelle telle que $\phi^{(n+1)}(x^i) = c_{n+1+i}$. Alors, le calcul précédent peut s'interpréter comme suit :

Si le polynôme générateur de $((k - 1)/k)_{\tau_n}$ est le polynôme orthogonal de degré k par rapport à la fonctionnelle $\phi^{(n+1)}$, alors $((n + k)/k)_s(z)$ devient l'approximant de Padé $[n + k, k]$.

3.2.3 Notion de fraction continue

Cette notion de fraction continue est extraite de [3].

Soit ψ une application de l'ensemble Σ des couples de suites dans l'ensemble des suites de fractions rationnelles. Les éléments de $\psi(\Sigma)$ sont appelés fractions continues.

Par exemple, étant donné un couple de suites de nombres (a, b) , on appelle fraction continue de Jacobi, la suite des fractions rationnelles $\{\tau_n\}$, définie par les relations de récurrence

$$\begin{aligned} A_0(z) &= 0, & A_1(z) &= 1 \\ B_0(z) &= 1, & B_1(z) &= b_1 + z \\ A_{n+1}(z) &= (b_{n+1} + z) A_n(z) - a_n^2 A_{n-1}(z) \\ B_{n+1}(z) &= (b_{n+1} + z) B_n(z) - a_n^2 B_{n-1}(z) \end{aligned}$$

On note alors

$$\frac{1|}{|b_1 + z} - \frac{a_1^2|}{|b_2 + z} - \dots$$

3.3 Approximation d'une série formelle par une série rationnelle, dans le cas de plusieurs variables non commutatives

Outre le polynôme p obtenu en tronquant la série s aux mots de longueur inférieure ou égale à k , nous proposons deux approximants de la série s à l'ordre k : Le premier est construit à partir de la matrice de Hankel de p , et le second est obtenu à partir de l'automate associé à s .

3.3.1 Par une série rationnelle g_k obtenue à partir de la matrice de Hankel de s .

Nous proposons un développement plus détaillé de cette méthode dans [4]. Soit $H(s)$ la matrice de Hankel de s . L'idée consiste

- à "cacher" dans $H(s)$ les coefficients des mots de longueur supérieure à k . On obtient alors $H(s)_{\leq k}$.
- à sélectionner un système de colonnes libres (resp. de lignes libres) de la gauche vers la droite (resp. de haut en bas) dans $H(s)_{\leq k}$:

$$\{C_1, \dots, C_r\}, \quad (\text{resp. } \{L_1, \dots, L_r\})$$

- à prolonger les colonnes (resp. les lignes) de manière à maintenir les relations de dépendance linéaires entre les colonnes de $H(s)_{\leq k}$ (resp. entre les lignes de $H(s)_{\leq k}$).

On construit ainsi la matrice de Hankel $H(g_k)$ d'une certaine série rationnelle g_k . Par construction, $H(g_k)$ coïncide avec $H(s)_{\leq k}$ pour les mots de longueur inférieure ou égale à k .

Remarques:

- 1) Par construction, le rang de la série approximante g_k est nécessairement minimal.
- 2) Dans [4], on montre que le rang de cet approximant rationnel peut être strictement inférieur à celui du polynôme de degré k , de matrice de Hankel égale à $H(s)_{\leq k}$.
- 3) Enfin, un tel approximant rationnel g_k existe toujours mais il n'est pas unique car il y a indéterminisme
 - sur le choix des colonnes libres
 - sur les relations linéaires qui amènent à la résolution de systèmes d'équations linéaires comportant plus d'indéterminées que d'équations.

Exemple 2 du chapitre précédent:

Série génératrice s associée à l'équation électrique

$$\dot{v} = -k_1 v - k_2 v^2 + i(t)$$

On construit l'approximant rationnel pour $k = 2$ suivant la méthode indiquée plus haut. Pour cela, on calcule $H(s)_{\leq 2}$:

	ϵ	z_0	z_1	z_0^2	z_0z_1	z_1z_0	z_1^2
ϵ	q_0	a	1	ab	0	b	0
z_0	a	ab	0				
z_1	1	b	0				
z_0^2	ab						
z_0z_1	0						
z_1z_0	b						
z_1^2	0						

On peut choisir L_ϵ, L_{z_1} comme lignes libres, pour $q_0 \neq a/b$ en remarquant que :

$$\begin{cases} L_{z_0} &= aL_{z_1} \\ L_{z_0^2} &= abL_{z_1} \\ L_{z_0z_1} &= 0 \\ L_{z_1z_0} &= bL_{z_1} \\ L_{z_1^2} &= 0 \end{cases}$$

Il reste à prolonger les lignes de manière à maintenir les relations de dépendance linéaire existant dans $H(s)_{\leq 2}$. On calcule alors $H(g_2)$:

	ϵ	z_0	z_1	z_0^2	z_0z_1	z_1z_0	$z_1^2 \dots$
ϵ	q_0	a	1	ab	0	b	$0 \dots$
z_0	a	ab	0	ab^2	0	0	$0 \dots$
z_1	1	b	0	b^2	0	0	$0 \dots$
z_0^2	ab	ab^2	0	ab^3	0	0	$0 \dots$
z_0z_1	0	0	0	0	0	0	$0 \dots$
z_1z_0	b	b^2	0	b^3	0	0	$0 \dots$
z_1^2	0	0	0	0	0	0	$0 \dots$
\cdot	\cdot	\cdot	\cdot	\cdot	\cdot	\cdot	\dots

Et les coefficients de g_2 sont donnés par :

$$\langle g|w \rangle = \lambda\mu(w)\gamma$$

avec

$$\mu(z_0) = \begin{pmatrix} 0 & 0 \\ a & b \end{pmatrix}$$

et

$$\mu(z_1) = \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix}$$

Le vecteur initial étant :

$$\gamma = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

et le covecteur étant :

$$\lambda = (q_0 \ 1)$$

On détermine ainsi une série rationnelle g_k approchant s à l'ordre $k = 2$, de rang minimal égal à 2. Remarquons que le rang du polynôme de degré 2 coïncidant avec s pour tout mot de longueur inférieure à 2 serait 3.

3.3.2 Par une série rationnelle f_k obtenue à partir de l'automate associé aux résiduelles de s .

Nous proposons une construction plus détaillée de cet approximant dans [5, 6]

- L'idée consiste à tronquer l'automate associé à la série s , aux mots reconnus de longueur inférieure ou égale à k , en supprimant les états (et les transitions) non atteints (et non parcourues) par ces mots.
- On obtient alors un automate fini (donc associé à une série rationnelle f_k) qui coïncide avec l'automate associé à s , pour tous les mots de longueur inférieure ou égale à k .
- Pour passer de l'ordre k à l'ordre $k + 1$, il suffit d'enrichir l'automate trouvé pour l'ordre k , des états et des transitions éventuelles.

Remarque:

Cet approximant n'est pas nécessairement de rang minimal.

Exemple 2 du chapitre précédent:

Série génératrice s associée à l'équation électrique

$$\dot{v} = -k_1 v - k_2 v^2 + i(t)$$

On construit l'approximant rationnel f_1 pour $k = 1$, et f_2 pour $k = 2$ suivant la méthode indiquée plus haut. On obtient l'automate fini donné par les figures 3.1 et 3.2 en tronquant l'automate infini trouvé au chapitre précédent. Pour $k = 2$, f_2 est exactement le même approximant que celui trouvé par la méthode précédente soit g_2 (figure 3.1). Par contre, pour $k = 1$, on trouve $\text{rang}(f_1) = 2$ alors que $\text{rang}(g_1) = 1$ (figure 3.2). Sur ce même exemple, on voit que les approximants pour des ordres successifs, par la méthode des automates, s'obtiennent aisément (figure 3.3):

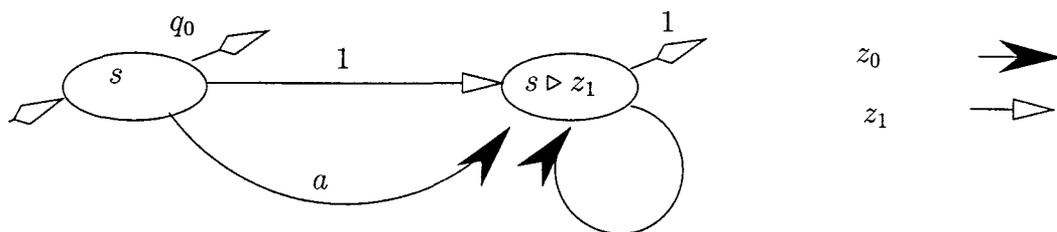


FIG. 3.1 – graphe de f_2 ou de g_2 relatif à l'exemple 2

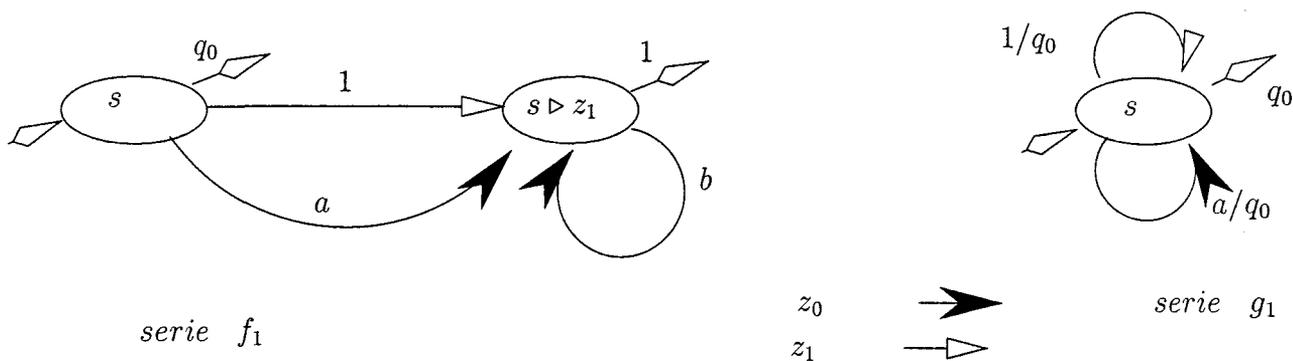


FIG. 3.2 – graphes de f_1 et g_1 relatifs à l'exemple 2

3.3.3 Relation avec les approximants de type Padé et les fractions continues

La notion d'approximant de type Padé définie dans [1], peut se généraliser au cas d'une série formelle en variables non commutatives [7].

Définition 3.3.1 On définit “une famille d'approximants de type Padé d'une série formelle s en variables non commutatives”, comme toute construction itérative, convergente vers s , d'approximants par des séries rationnelles.

► Cas de l'approximant rationnel f_k calculé par l'automate associé à la série s

Selon la définition précédente, l'approximant rationnel calculé par l'automate associé est bien un approximant de type Padé. Il est clair que la suite $(f_k)_{k \in \mathbb{N}}$ des approximants de la série s est convergente de limite s .

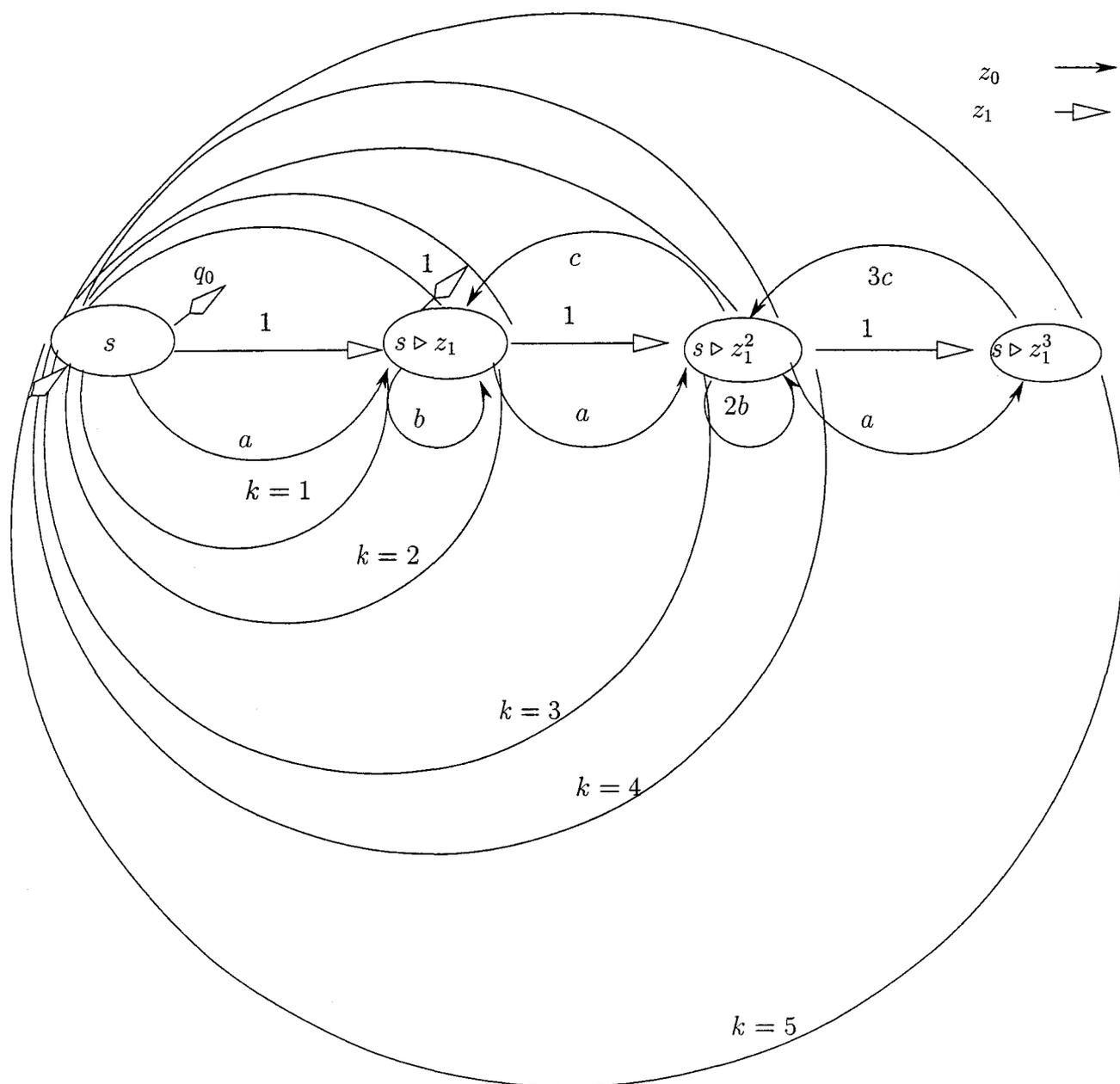


FIG. 3.3 – graphes des f_i pour $1 \leq i \leq 5$ relatifs à l'exemple 2

Reprenons l'exemple 2, étudié préalablement. Alors

$$\left\{ \begin{array}{l} f_1 = (z_1 + az_0) \\ f_2 = (z_1 + az_0)[\epsilon]^*(bz_0)^* \\ f_3 = (z_1 + az_0)[(bz_0)^*(z_1 + az_0)(cz_0)]^*(bz_0)^* \\ f_4 = (z_1 + az_0)[(bz_0)^*(z_1 + az_0)(2bz_0)^*(cz_0)]^*(bz_0)^* \\ \dots = \dots \end{array} \right.$$

que l'on peut encore écrire en notant

$$\frac{u|v}{w} = uw^{-1}v$$

sous la forme :

$$\left\{ \begin{array}{l} f_1 = (z_1 + az_0) \\ f_2 = (z_1 + az_0)(bz_0)^* \\ f_3 = \frac{(z_1 + az_0)|(bz_0)^*}{1 - (bz_0)^*(z_1 + az_0)(cz_0)} \\ f_4 = \frac{(z_1 + az_0)|(bz_0)^*}{1 - (bz_0)^*(z_1 + az_0)(2bz_0)^*(cz_0)} \\ \dots = \dots \end{array} \right.$$

La suite des approximants $(f_k)_{k \in \mathbb{N}}$ est telle que

$$\omega(f_k, f_{k-1}) = \text{ord}(f_k - f_{k-1}) \geq k$$

• pour $k = 2p + 1$, alors $z_1^{p+1}z_0^p$ est reconnu par f_k et non f_{k-1} .

Pour la distance ultramétrique définie au chapitre 1 ($d(s, t) = \rho^{\omega(s, t)}$ $0 < \rho < 1$), la suite $(f_k)_{k \in \mathbb{N}}$ est donc convergente et sa limite est s .

Suivant l'idée de P.Flajolet [2], on peut définir la fraction continue associée à la série caractéristique des "chemins étiquetés positifs", comme la limite d'une suite de séries formelles rationnelles associées aux chemins étiquetés positifs de hauteur $\leq h$.

Définition 3.3.2 On définit la fraction continue associée à un \mathbb{R} -automate, comme la limite d'une suite $(u_h)_{h \in \mathbb{N}}$ de séries formelles rationnelles associées à des chemins réussis passant par des états distants de h au plus de l'état initial et empruntant des transitions autorisées par le \mathbb{R} -automate.

Conformément à cette définition, on peut construire la fraction continue associée à s . Reprenons l'exemple 2. Calculons la fraction continue associée à l'automate :

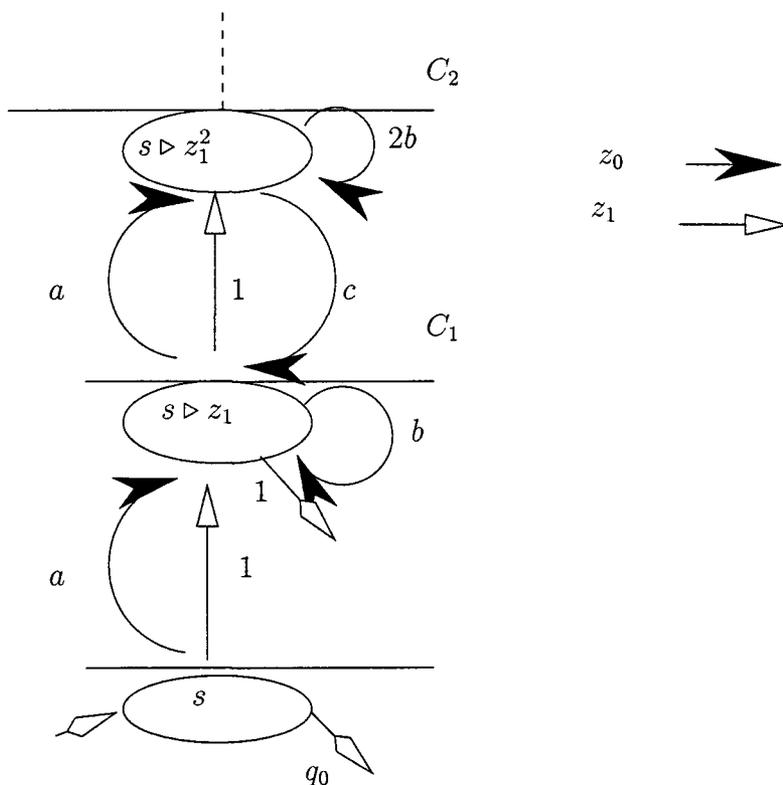


FIG. 3.4 – graphe de la fraction continue associée à l'exemple 2

Posons $u = z_1 + az_0$

Notons C_1 la partie de trajectoire située au dessus du palier 1.

$$uC_1^*(bz_0)^* = \frac{u|(bz_0)^*}{1 - C_1}$$

en notant

$$\frac{u|v}{w} = uw^{-1}v$$

Puis en notant C_2 la partie supérieure au palier 2.

$$uC_1^*(bz_0)^* = \frac{u|(bz_0)^*}{1 - \frac{(bz_0)^*u|(2bz_0)^*cz_0}{1 - C_2}}$$

etc ...

Ceci n'est rien d'autre que la fraction continue associée à la série génératrice de départ (figure 3.4).

Remarquons que l'écriture de la fraction continue est ici possible car l'automate est linéaire. Dans un cas plus général, la fraction continue existe mais son écriture devient illisible.

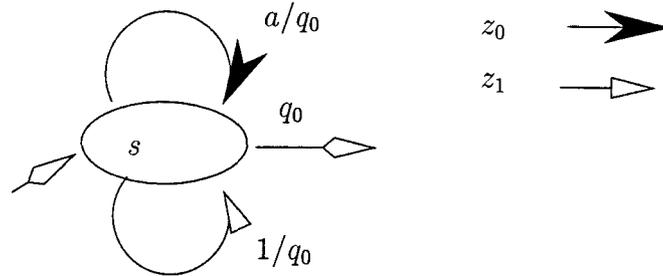


FIG. 3.5 – graphe de g_1 pour l'exemple 2

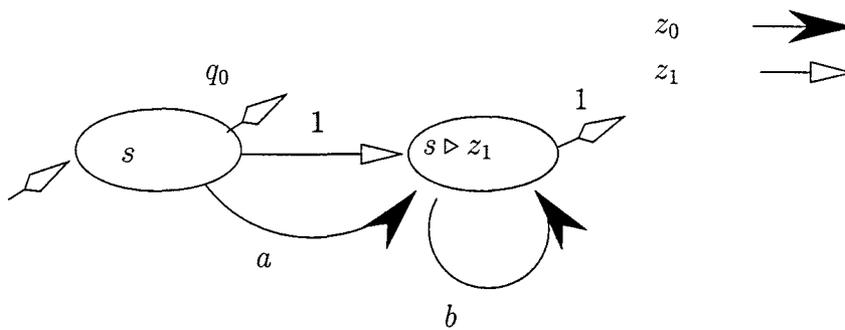


FIG. 3.6 – graphe de g_2 pour l'exemple 2

► Cas de l'approximant rationnel calculé par la matrice de Hankel associé à la série s

Rappelons que l'approximant g_k de la série s construit par la méthode de la matrice de Hankel vérifie :

$$\text{ord}(s - g_k) > k$$

Par conséquent,

$$\text{ord}(g_k - g_{k-1}) \geq k$$

Selon la définition 3.3.1 d'un approximant de type Padé, la famille $(g_k)_{k \in \mathbb{N}}$ apparaît bien comme une famille d'approximants de type Padé. Pour la distance ultramétrique définie au chapitre 1, la suite $(g_k)_{k \in \mathbb{N}}$ est une suite convergente de limite s .

Par contre, on ne peut pas construire une fraction continue conformément à la définition 3.3.2. En effet, les séries rationnelles (g_k) obtenues par cette méthode ne sont pas associées à des automates tronqués à partir du \mathbb{R} -automate associé à s . Pour l'observer, il suffit de remarquer que le passage de g_k à g_{k+1} bouleverse complètement l'automate fini associé :

Reprenons l'exemple 2 de l'équation électrique :

- le calcul de g_1 de rang 1 correspond à un automate de graphe donné par la figure 3.5.
- le calcul de g_2 de rang 2 correspond à un automate de graphe donné par la figure 3.6.

Il est clair que les séries rationnelles g_1 et g_2 ne sont pas associées à des automates obtenus en décrivant des sous automates du \mathbb{R} -automate associé à s .

3.3.4 Que se passe-t-il si la série s est rationnelle ?

• Dans le cas d'un approximant rationnel f_k obtenu par la méthode des automates, il est clair que si l'on part de l'automate associé à s rationnelle, les automates tronqués successifs correspondent à une suite stationnaire d'approximants f_k , égaux à s , à partir d'un certain rang k_0 .

• Le cas d'un approximant rationnel g_k obtenu par la méthode de la matrice de Hankel est plus difficile à résoudre. On montre la proposition suivante:

Proposition 3.3.1 *Si s est une série rationnelle de rang r , alors il existe un ordre k_0 tel que $\forall k \in \mathbb{N}$ vérifiant $k \geq k_0$, l'approximant rationnel g_k calculé par la méthode de la matrice de Hankel, n'est autre que la série s rationnelle de départ. Cette propriété est vraie pour*

$$k_0 = 2r - 1.$$

preuve:

- Montrons d'abord que si k_0 existe, alors $k_0 \geq 2r - 1$.
D'après le théorème de Fliess du chapitre 2, on sait qu'il existe deux ensembles de mots de longueur $< r$:

$$(g_i)_{i=1}^r, \quad (d_j)_{j=1}^r$$

tels que le déterminant de $(\langle s | g_i d_j \rangle)_{1 \leq i, j \leq r}$ soit non nul.

De plus, on sait que la série rationnelle s est entièrement définie par ses r vecteurs colonnes (ou lignes) de base et les actions de lettres $z_h \in Z$ sur ces vecteurs de base. Il en résulte que la connaissance de s est entièrement acquise en se limitant par exemple, à la partie de la matrice de Hankel dont les lignes sont indicées par des mots de longueur $\leq r - 1$, et dont les colonnes sont indicées par des mots de longueur $\leq r$. Cela revient à dire qu'on se limite aux mots de longueur $\leq 2r - 1$.
Donc,

$$k_0 \geq 2r - 1$$

- Montrons que g_k vérifie :

$$\forall k \geq k_0 \quad g_k = s \quad (k_0 = 2r - 1)$$

*Remarquons que g_k et s ont même rang r :

1) en effet, si $\text{rang}(g_k) = r_1$, alors $r_1 \leq r$ puisque g_k est un approximant rationnel de s de rang r_1 minimal.

2) d'autre part, $r \leq r_1$:

en effet, la matrice de Hankel de s restreinte aux lignes et aux colonnes indicées

par des mots de longueur $\leq r - 1$ est de rang r . Soit une base de vecteurs colonnes (ou lignes) de ce tableau obtenue en prenant de la gauche vers la droite, le premier vecteur linéairement indépendant avec les précédents. En prolongeant ces vecteurs colonnes aux mots de longueur $\leq 2r - 1$, on obtient “un système de vecteurs libres” au sens de la construction proposée dans l’approximation par la matrice de Hankel. Donc $r \leq r_1$.

*Il suffit d’observer que s et g_k apparaissent toutes deux comme l’unique série rationnelle de rang r , reconnue par le même automate matriciel.

Fin de preuve

3.3.5 Comparaison des deux approximations proposés

Dans ce paragraphe, nous récapitulons les propriétés de chacun des deux approximations proposés:

- Le rang de l’approximant g_k obtenu par la matrice de Hankel est minimal. Cette propriété le rend préférable à l’approximant f_k obtenu en tronquant l’automate, du point de vue de la place mémoire occupée.
- Les suites $(g_k)_{k \in \mathbb{N}}$ et $(f_k)_{k \in \mathbb{N}}$ des approximations pour des ordres k successifs sont des suites convergent vers la série s .
- La suite $(f_k)_{k \in \mathbb{N}}$ des approximations obtenus en tronquant l’automate associé à s permet de définir une fraction continue au sens de P.Flajolet, alors que ce n’est pas le cas pour la suite $(g_k)_{k \in \mathbb{N}}$ des approximations obtenus par la matrice de Hankel.
- Une autre manière d’exprimer la propriété précédente est d’observer que si le calcul de l’approximant à l’ordre k se révèle insuffisant et qu’il faut passer à l’ordre $k + 1$, on bénéficie des calculs déjà faits, dans le cas de l’approximant par la méthode de l’automate (on enrichit éventuellement l’automate associé à l’ordre k , de nouveaux états et de nouvelles transitions pour obtenir l’automate associé à l’ordre $k + 1$), alors que le calcul de l’approximant à l’ordre $k + 1$ par la méthode de la matrice de Hankel bouleverse les calculs acquis à l’ordre k .
(**exemple 2** de l’équation électrique pour $k=1$ et $k=2$)
- Dans le cas où s est rationnelle, les suites $(f_k)_{k \in \mathbb{N}}$ et $(g_k)_{k \in \mathbb{N}}$ sont stationnaires et égales à s , à partir d’un certain ordre.
- La méthode de l’automate tronqué peut se révéler inimplantable si on ne connaît pas a priori les relations de dépendance linéaire entre les résiduelles. Il se trouve que cette remarque rédhibitoire pour une série quelconque sera repoussée dans le cas d’une série génératrice de système dynamique.

3.3.6 Application aux systèmes dynamiques

Comme nous allons le voir au chapitre suivant, la construction d'approximants rationnels à l'ordre k permettra d'écrire des algorithmes de calcul d'approximants bilinéaires de systèmes dynamiques.

Bibliographie

- [1] Brezenski C., *Padé type approximation and general orthogonal polynomials*, INSM50, Birkhauser, 1980.
- [2] Flajolet P., *Combinatorial aspects of continued fractions*, Discrete Math., vol.32, pp 125-161, 1980.
- [3] Giliewicz J., *Approximants de Padé*, Lecture Notes in Mathematics, vol. 667, Springer Verlag, 1978.
- [4] Hespel C., *Approximation de séries formelles par des séries rationnelles*, RAIRO Inform.Théor., vol.18, n° 3, pp. 241-258, 1984.
- [5] Hespel C., Jacob G., *Approximations bilinéaires de systèmes analytiques: exemples de listings*, CALSYF 7, pp. 181-219, Journées de calcul Formel, Luminy, 1988.
- [6] Hespel C., Jacob G., *Approximation of nonlinear dynamic systems by rational series*, Theoret. Comput. Sciences, 79, pp.151-162, 1991.
- [7] Leroux P., Viennot X.G., *A combinatorial approach to nonlinear functional expansions: an introduction with example*, Algebraic and Computing Treatment of Noncommutative Power Series , G.Jacob and C.Reutenauer eds., Theoret. Comput. Sciences, 1991.

Chapitre 4

Systemes dynamiques et séries formelles

4.1 Introduction

Ce chapitre fait seulement le lien entre séries formelles et systèmes dynamiques. Nous nous intéressons désormais au traitement de systèmes dynamiques linéaires en la commande. Des séries formelles en variables non commutatives y apparaissent naturellement. Nous rappelons certaines propriétés de ces séries:

- La série génératrice ou série de Fliess [1, 2] sert à coder le comportement entrée/sortie du système. La série de Chen est la série associée à la commande du système.
- Une condition nécessaire et suffisante pour qu’une série formelle quelconque s soit la série génératrice d’un tel système est qu’elle soit “produite différentiellement” [2, 5].

4.2 Rappels

4.2.1 Algèbre de Lie

Nous rappelons d'abord quelques définitions. La première définition est relative à l'algèbre de Lie [6].

Définition 4.2.1 On définit le crochet de Lie de deux polynômes puis de deux séries comme suit:

- Crochet de Lie de 2 polynômes P, Q :
C'est une application bilinéaire de

$$\begin{array}{ccc} \mathbb{R}\langle Z \rangle \times \mathbb{R}\langle Z \rangle & \rightarrow & \mathbb{R}\langle Z \rangle \\ (P, Q) & \rightarrow & [P, Q] \end{array}$$

telle que $[P, Q] = PQ - QP$

Soit $\text{Lie}\langle Z \rangle$, l'algèbre de Lie libre sur Z , contenant Z , stable par crochets de Lie. Ses éléments sont les polynômes de Lie sur Z .

Une série formelle s de $\mathbb{R}\langle\langle Z \rangle\rangle$ est une série de Lie si on peut l'écrire sous la forme:

$$s = \sum_{k \geq 1} P_k$$

où P_k est un polynôme de Lie, homogène de degré k .

- Crochet de Lie de 2 séries s_1, s_2 :
Pour

$$s_1 = \sum_{k \geq 1} P_k \quad s_2 = \sum_{l \geq 1} Q_l$$

$$[s_1, s_2] = \sum_{k, l \geq 1} [P_k, Q_l]$$

On définit la notion d'algèbre de Lie:

Définition 4.2.2 L est une algèbre de Lie ssi:

- le crochet $[x, y]$ est bilinéaire
- $\forall x \in L, [x, x] = 0$
- $\forall x, y, z \in L, [x[y, z]] + [y[z, x]] + [z[x, y]] = 0$

L'ensemble des séries de Lie noté $Lie\langle\langle Z \rangle\rangle$ est une algèbre de Lie.

On définit le rang de Lie d'une série formelle:

Définition 4.2.3 *Le rang de Lie d'une série formelle $s \in \mathbb{R}\langle\langle Z \rangle\rangle$ est la dimension de l'espace vectoriel $\{s \circ P \mid P \in Lie\langle Z \rangle\}$.*

4.2.2 Systèmes dynamiques considérés

On peut maintenant définir les systèmes dynamiques considérés:

On s'intéresse aux systèmes analytiques (Σ) affines en la commande $a = \{a_i\}_{0 \leq i \leq m}$ de la forme:

$$(\Sigma) \begin{cases} \dot{q}(t) &= g_0(q) + \sum_{i=1}^m a_i(t)g_i(q) \\ y &= h(q) \end{cases}$$

tels que

- l'état $q = (q^s)_{1 \leq s \leq N} \in Q$, N -variété réelle analytique.
- $\{g_i\}_{0 \leq i \leq m}$ sont des champs de vecteurs définis dans un voisinage de $q(0)$. Ils engendrent la \mathbb{R} -algèbre de Lie notée $L(G)$.
- $h : Q \rightarrow \mathbb{R}^p$ est une fonction d'observation analytique définie dans un voisinage de $q(0)$.
- les entrées $\{a_i\}_{0 \leq i \leq m}$ sont continues par morceaux.

La fonction $y : t \in \mathbb{R} \rightarrow h(q(t)) \in \mathbb{R}^p$ désigne la sortie du système.

Nous associons à (Σ) l'alphabet de codage $\mathcal{Z} = \{z_0, z_1, \dots, z_m\}$.

Nous définissons les opérateurs différentiels comme suit:

- Pour toute lettre z_i

$$\mathcal{Y}(z_i) = \sum_{s=1}^N g_i^s(q) \frac{\partial}{\partial q^s} \quad \forall i \quad 0 \leq i \leq m$$

- Pour tout mot w sur \mathcal{Z} , nous définissons récursivement $\mathcal{Y}(w)$ par

$$\mathcal{Y}(vz_j) = \mathcal{Y}(v) \circ \mathcal{Y}(z_j) \quad \forall z_j \in \mathcal{Z} \quad \forall v \in \mathcal{Z}^*$$

– Pour tout mot $w = z_{i_1} z_{i_2} \cdots z_{i_p}$ et toute fonction analytique h :

$$\mathcal{Y}(w) \circ h = g_{i_1} g_{i_2} \cdots g_{i_p} \circ h$$

où $g_i \circ h$ est la dérivée de Lie de h le long du champ de vecteur g_i .

Posons

$$\begin{cases} a_0(t) \equiv 1 & \forall t, \\ a & = \{a_0, a_1, \dots, a_m\} \end{cases}$$

L'intégrale itérée $\int_0^t \delta_a(w)$

du mot w pour les entrées a est définie par:

$$\begin{cases} \int_0^t \delta_a(\varepsilon) = 1 \\ \int_0^t \delta_a(v z_i) = \int_0^t \left(\int_0^\tau \delta_a(v) \right) a_i(\tau) d\tau \\ \forall z_j \in Z \quad \forall v \in Z^*. \end{cases}$$

(Selon la notation de Chen, l'ordre utilisé ici est l'inverse de celui de Fliess)

Rappelons la **formule de Fliess**:

$$y(t) = \sum_{w \in Z^*} \left[\mathcal{Y}(w) \circ h_{|_{q_0}} \right] \int_0^t \delta_a(w).$$

4.3 Deux séries formelles remarquables

La formule de Fliess fait apparaître deux séries formelles remarquables: la série génératrice et la série de Chen.

4.3.1 La série génératrice

Le comportement entrée/sortie de ces systèmes (Σ) est entièrement déterminé par leur **série génératrice** encore appelée '**série de Fliess**':

$$G = \sum_{w \in Z^*} \left[\mathcal{Y}(w) \circ h_{|_{q_0}} \right] w$$

Remarquons donc que cette série en variables non commutatives est définie sur l'alphabet de codage $Z = \{z_0, z_1, \dots, z_m\}$ correspondant aux entrées du système $a_1(t), \dots, a_m(t)$ et $a_0(t) = 1$, et ses coefficients sont réels.

Le calcul de la sortie $y(t)$ se fait en remplaçant dans l'expression de la série génératrice, tout mot w par la contribution des entrées correspondantes: l'intégrale itérée.

4.3.2 La série de Chen

La série

$$C_a(t) = \sum_{w \in Z^*} \left[\int_0^t \delta_a(w) \right] w$$

est la ‘série de Chen’ de l’entrée a .

Nous remarquons que la formule de Fliess s’écrit encore:

$$\begin{aligned} y(t) &= \sum_{w \in Z^*} \langle G|w \rangle \langle C_a(t)|w \rangle \\ &= \langle G \| C_a(t) \rangle. \end{aligned}$$

(où $\|$ signifie somme infinie).

D’après le théorème de Ree [4], la série de Chen est l’exponentielle d’une série de Lie.

4.4 Série produite différentiellement

Le problème posé est ici de savoir à quelle condition une série formelle est série génératrice d’un système (Σ) .

M.Fliess [2] propose une caractérisation de cette propriété. Avant d’énoncer son théorème, donnons deux nouvelles définitions:

Définition 4.4.1 Une série $s \in \mathbb{R}\langle\langle Z \rangle\rangle$ vérifie l’hypothèse de convergence (C) si Pour tous les polynômes de Lie P_1, \dots, P_d , il existe des constantes α et C telles que

$$\forall i_1, \dots, i_d \in \mathbb{N}, \quad |\langle s | P_1^{i_1} \dots P_d^{i_d} \rangle| \leq \alpha C^{i_1 + \dots + i_d} i_1! \dots i_d!$$

puis

Définition 4.4.2 On dit que (μ, h) est une **représentation différentielle de s de dimension d** si et seulement si il existe un entier d , un homomorphisme μ de Z^* dans le monoïde multiplicatif des endomorphismes de l’ensemble des séries formelles commutatives $\mathbb{R}[[\xi_1, \dots, \xi_d]] = \mathbb{R}[[\xi]]$ tel que

- $\mu(z)$ soit un champ de vecteurs formel, convergent pour tout $z \in Z$
- $h \in \mathbb{R}[[\xi]]$ soit une série formelle convergente tels que

$$\forall w \in Z^* \quad \langle s | w \rangle = h(\mu w)|_0$$

Le théorème s'énonce ainsi [2]:

Théorème 11 *Une série $s \in \mathbb{R}\langle\langle Z \rangle\rangle$ est produite différentiellement ssi son rang de Lie est fini et si elle vérifie l'hypothèse de convergence (C).*

Dans ce cas, son rang de Lie d est égal à la plus petite dimension de ses représentations différentielles.

Pour que s soit la série génératrice d'un système dynamique (Σ) , il faut et il suffit que s soit produite différentiellement.

C.Reutenauer [5] propose une démonstration simplifiée et une autre formulation de ce théorème:

Proposition 4.4.1 *Une série formelle $s \in \mathbb{R}\langle\langle Z \rangle\rangle$ est produite différentiellement s'il existe une représentation différentielle de s de dimension d . Pour que s soit la série génératrice d'un système dynamique (Σ) , il faut et il suffit que s soit produite différentiellement.*

Bibliographie

- [1] Fliess M., *Fonctionnelles causales non linéaires et indéterminées non commutatives*, Bull. Soc. Math. France 109, pp.3-40, 1981.
- [2] Fliess M., *Réalisation locale des systèmes non linéaires, Algèbres de Lie filtrées transitives et séries génératrices non commutatives*, Invent. Math., 71, pp. 521-537, 1983.
- [3] Jacob G., *Algebraic methods and computer algebra for nonlinear systems' study*, IMACS Symposium MCTS, 1991.
- [4] Ree R., *Lie elements and an Algebra associated with shuffle*, Ann. of Math., vol. 68, pp. 210-220, 1958.
- [5] Reutenauer C., *The local realization of generating series of finite Lie rank*, in "Algebraic and Geometric Methods in Nonlinear Control Theory", M.Fliess and M.Hazewinkel eds., D.Reidel Publishing Company, pp. 33-43, 1986.
- [6] Viennot G., *Algèbres de Lie libres et Monoïdes libres*, Lecture notes in Mathematics, n° 691, Springer Verlag, 1978.

Chapitre 5

Systemes bilinéaires et séries rationnelles

5.1 Introduction

Ce chapitre fait seulement le lien entre séries rationnelles et systèmes bilinéaires. Nous rappelons la définition d'un système bilinéaire, énoncée par l'une des deux propriétés équivalentes suivantes:

- Son équation d'état ne contient que des matrices de dimension finie.
- Sa série génératrice est rationnelle.

Les systèmes bilinéaires sont donc aux systèmes dynamiques ce que les séries rationnelles sont aux séries formelles: Leur comportement est entièrement déterminé par la (les) matrice(s) de dimension finie, figurant dans l'équation d'état (B), comme la série rationnelle est caractérisée par sa représentation.

Les outils de présentation ou d'approximation développés précédemment pour les séries rationnelles pourront donc s'appliquer aux systèmes bilinéaires.

Nous indiquons ensuite comment obtenir un système bilinéaire réduit à partir de sa série génératrice rationnelle, grâce à sa "réalisation minimale" [1].

5.1.1 Rappels et notations

Définissons un système bilinéaire par son équation d'état:

$$(B) \begin{cases} \dot{x}(t) &= (M_0 + \sum_{i=1}^m a_i(t)M_i)x(t) \\ \bar{y}(t) &= \lambda.x(t) \end{cases}$$

tels que

- $x(t)$ appartient à un \mathbb{R} -espace vectoriel Q de dimension finie.
- $\{M_i\}_{0 \leq i \leq m} : Q \rightarrow Q$ et $\lambda : Q \rightarrow \mathbb{R}$ sont \mathbb{R} -linéaires.
- les entrées $\{a_i\}_{0 \leq i \leq m}$ sont continues par morceaux ($a_0(t) \equiv 1$)

La série génératrice associée à (B) est:

$$G = \lambda.x(0) + \sum_{\nu \geq 0} \sum_{j_0, \dots, j_\nu=0}^m \lambda.M_{j_0} \cdots M_{j_\nu} x(0) z_{j_0} \cdots z_{j_\nu}$$

C'est donc une série rationnelle.

5.2 Représentation minimale et système réduit

Quelques définitions sont nécessaires. Définissons d'abord l'observabilité et la semi-accessibilité d'un système:

Définition 5.2.1 *Un état $x \in Q$ d'un système bilinéaire est dit observable si, considéré comme nouvel état initial, on peut trouver une entrée pour laquelle la sortie soit nulle. Un système bilinéaire est dit complètement observable ssi tous ses états sont observables.*

D'autre part,

Définition 5.2.2 *Les vecteurs d'état accessibles d'un système bilinéaire sont ceux que l'on peut obtenir comme solution $x(t)$ à un instant t de l'équation d'état.*

On appelle état semi-accessible d'un système bilinéaire, toute combinaison linéaire d'états accessibles.

Un système bilinéaire est dit semi-accessible ssi tous ses états sont semi-accessibles.

Donnons la définition d'un système réduit:

Définition 5.2.3 *Un système bilinéaire est réduit ssi tout état est à la fois observable et semi-accessible.*

Donnons enfin la définition d'une réalisation:

Définition 5.2.4 *On appelle réalisation d'une série formelle s , la donnée d'un système dynamique (Σ) ayant s comme série génératrice.*

G.Jacob [2] montre que la réalisation minimale d'une série rationnelle est en un certain sens la réalisation réduite.

Théorème 12 *Un système bilinéaire de série génératrice s en est une réalisation d'espace d'état de dimension minimale ssi c'est un système réduit.*

Il propose alors une construction explicite de la réalisation réduite passant par la représentation minimale de la série génératrice.

Par exemple, considérons la série prise comme exemple 1 au chapitre 2:

Exemple 1: [2]

Série rationnelle définie sur l'alphabet $Z = \{z_0, z_1\}$ par

$$s = \sum_{w \in Z^*} (|w|_{z_0} - |w|_{z_1})w$$

dont on a trouvé la représentation minimale suivante (μ, γ, λ) :

$$\mu(z_0) = \begin{pmatrix} 0 & -1 \\ 1 & 2 \end{pmatrix}$$

puis

$$\mu(z_1) = \begin{pmatrix} 2 & 1 \\ -1 & 0 \end{pmatrix}$$

le vecteur initial étant :

$$\gamma = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

et le covecteur étant :

$$\lambda = (\langle s | \epsilon \rangle \quad \langle z_0 \triangleleft s | \epsilon \rangle)$$

Alors s apparaît, grâce à cette construction, comme la série génératrice du système bilinéaire réduit suivant:

$$\begin{cases} \dot{x}(t) &= \left\{ \begin{pmatrix} 0 & -1 \\ 1 & 2 \end{pmatrix} + a_1(t) \begin{pmatrix} 2 & 1 \\ -1 & 0 \end{pmatrix} \right\} x(t) \\ \bar{y}(t) &= \begin{pmatrix} 0 & 1 \end{pmatrix} x(t) \\ x(0) &= \begin{pmatrix} 1 \\ 0 \end{pmatrix} \end{cases}$$

Remarque 5.2.1 - *On observe donc que le rang de la série génératrice est aussi la dimension de l'espace d'état du système réduit.*

- *La réalisation minimale est unique à un changement de coordonnées près.*

Bibliographie

- [1] Jacob G., *Réalisation des systèmes réguliers (ou bilinéaires) et séries génératrices non commutatives*, Séminaire d'Aussois, RCP567, Outils et modèles mathématiques pour l'Automatique, l'Analyse des Systèmes, et le traitement du Signal (CNRS Landau), 1980.
- [2] Jacob G., *Réalisation des systèmes réguliers et séries génératrices non commutatives*, publication interne de l'équipe lilloise d'Informatique théorique, n° IT-26-80, 1980.

Chapitre 6

Bilinéarisations de systèmes dynamiques

6.1 Introduction

Le but de ce chapitre est de fournir des approximants bilinéaires de systèmes dynamiques à l'ordre k , qui soient des approximants de type Padé et qui vérifient une condition Taylorienne sur la sortie, à l'ordre k .

Nous développons deux techniques pour mettre en œuvre ce concept:

- L'une est basée sur des calculs à partir de la présentation de la série génératrice du système, par la matrice de Hankel. Elle fournit un approximant bilinéaire de type Padé de rang minimal.
- L'autre est obtenue en tronquant un automate associé à la série génératrice du système. Nous choisissons de présenter deux automates particuliers que nous appelons "automate géométrique" (méthode de Carleman [2]) et "automate structurel" (méthode de l'algèbre différentielle).

Des comparaisons des avantages et des inconvénients respectifs des deux méthodes sont présentées en conclusion. Il est à noter que la méthode par la présentation de la matrice de Hankel et la méthode par l'automate structurel fournissent l'une comme l'autre un outil intéressant pour le Calcul Formel.

6.2 Enoncé du problème

Le problème posé est le suivant: Proposer une construction effective de système bilinéaire (B) approximant un système dynamique donné (Σ), à l'ordre k .

Définition 6.2.1 *On dit qu'un système (B) approxime un système (Σ) à l'ordre k ssi les sorties des deux systèmes (B) et (Σ) ont des développements de Taylor coïncidant jusqu'à l'ordre k .*

Pour résoudre ce problème d'approximation de système dynamique, on se ramène à un problème d'approximation de série. La résolution passe donc par les deux étapes suivantes:

- Montrer que le développement de Taylor jusqu'à l'ordre k de la sortie d'un système ne dépend que des coefficients des mots de longueur $\leq k$, de la série génératrice du système. Ceci a été démontré par C.Hespel et G.Jacob dans [3].
- Construire un système bilinéaire (B) dont la série génératrice rationnelle g approxime la série génératrice s associée au système (Σ) , pour tous les mots de longueur $\leq k$ selon le schéma suivant:

$$\begin{array}{ccc} (\Sigma) & & (B) \\ \downarrow & & \uparrow \\ s & \longrightarrow & g \end{array}$$

Nous présentons plusieurs méthodes. La première est calquée sur la méthode d'approximation par la matrice de Hankel, vue au chapitre 3.

Les autres méthodes sont inspirées de la méthode d'approximation par automate tronqué, décrite aussi au chapitre 3.

- 1) L'une est une illustration par les automates de la méthode bien connue de linéarisation de Carleman. C'est "la méthode de l'automate géométrique".
- 2) L'autre est basée sur des calculs dans l'algèbre des opérateurs différentiels définis par les champs de vecteurs du système (Σ) . C'est "la méthode de l'automate structurel".

6.3 Approximation par un système bilinéaire de rang minimal

Cette méthode consiste à approximer la série génératrice s de (Σ) par la série rationnelle g , à l'ordre k , par la méthode de la matrice de Hankel [5, 6]. Un algorithme en langage de MACSYMA et des exemples d'exécution sont proposés dans [4]. Cet approximant a la propriété d'être de rang minimal.

6.4 Approximation par automates tronqués

L'idée est d'associer au système (Σ) un automate, en général infini, dont les approximants seront obtenus par troncature.

6.4.1 Bilinéarisation par l'automate structurel

Nous développons cette méthode dans [6, 7]. Rappelons que la série génératrice associée au système (Σ) :

$$\begin{cases} \dot{q}(t) &= g_0(q) + \sum_{i=1}^m a_i(t)g_i(q) \\ y &= h(q) \end{cases}$$

a pour expression:

$$s = \sum_{l \geq 0} \sum_{i_j=0}^m g_{i_1} \cdots g_{i_{l-1}} g_{i_l} \circ h|_{q(0)z_{i_1} \cdots z_{i_l}}$$

Or les champs de vecteurs g_r sont analytiques dans un voisinage de l'état initial $q(0)$. Ils peuvent donc s'écrire sous la forme:

$$\begin{aligned} g_r &= \sum_{j=1}^n \theta_r^j(q_1, \dots, q_n) \frac{\partial}{\partial q_j}, \quad 0 \leq r \leq m \\ &= \sum_{j=1}^n \theta_r^j(q_1, \dots, q_n) D_j \end{aligned}$$

Donc la série génératrice s a pour expression:

$$s = \sum_{l \geq 0} \sum_{i_j=0}^m [\theta_{i_1}^{j_1}(q_1, \dots, q_n) \cdots \theta_{i_{l-1}}^{j_{l-1}}(q_1, \dots, q_n) D_{j_1} \cdots D_{j_{l-1}}] g_{i_l} \circ h|_{q(0)z_{i_1} \cdots z_{i_l}}$$

Cette écriture suggère de définir un automate décrivant l'action à droite des champs de vecteurs g_r sur les D_α (α étant un multi-indice). Plus précisément, en notant:

$$\begin{aligned} \alpha &= (i_1, \dots, i_n) \\ D^\alpha &= D_1^{i_1} \cdots D_n^{i_n} \end{aligned}$$

on pose la définition suivante:

Définition 6.4.1 *Le \mathbb{R} -automate structurel associé au système (Σ) est un 5-uple $\{Z, Q, *, I, l\}$ où*

- Z est l'alphabet $\{z_0, \dots, z_m\}$
- Q est l'espace vectoriel engendré par les D^α
- $*$ est l'action à droite des g_r sur les D_α
- I est l'état initial

– l est la forme linéaire définie par le vecteur ligne dont la α ème composante est $D_\alpha \circ h$.

L'action des champs de vecteurs g_r sur les D^α est donnée par la formule de Leibnitz:

$$D^\alpha g_r = \sum_{j=1}^n \sum_{\beta \leq \alpha} \binom{\alpha}{\beta} (D^\beta \circ g_r \circ q_j) D^{\alpha-\beta} D^j$$

avec

$$\alpha = (i_1, \dots, i_n) \quad \beta = (j_1, \dots, j_n)$$

et

$$\beta \leq \alpha \text{ ssi } j_1 \leq i_1, \dots, j_n \leq i_n$$

et si $\beta \leq \alpha$ alors:

$$\binom{\alpha}{\beta} = \binom{i_1}{j_1} \dots \binom{i_n}{j_n}$$

En tronquant ce \mathbb{R} -automate structurel aux états visités par un chemin réussi de longueur $\leq k$, on obtient un automate d'état fini, associé à une série rationnelle, génératrice d'un système bilinéaire approximant.

Cet algorithme est présenté en langage de MACSYMA dans [4].

Exemple 6.4.1 *Equation de Duffing:*

$$y^{(2)} + ay^{(1)} + b^2y + cy^3 = u(t)$$

Cette équation s'écrit encore sous la forme suivante:

$$(\Sigma_1) \begin{cases} \dot{q}(t) &= g_0(q) + u(t)g_1(q) \\ g_0 &= F \frac{\partial}{\partial q_2} + q_2 \frac{\partial}{\partial q_2} = F D_2 + q_2 D_1 \\ g_1 &= \frac{\partial}{\partial q_2} = D_2 \\ y(t) &= q_1(t) \end{cases}$$

en posant $F = -aq_2 - b^2q_1 - cq_1^3$.

Le \mathbb{R} -automate structurel est défini par

- $Z = \{z_0, z_1\}$
- Q est le \mathbb{R} -espace vectoriel engendré par $(D_1^i D_2^j)_{i,j \geq 0}$
- $I = D^0$
- La forme linéaire l est définie par le vecteur ligne

$$(q_1 \ 1 \ 0 \ \dots \ 0)$$

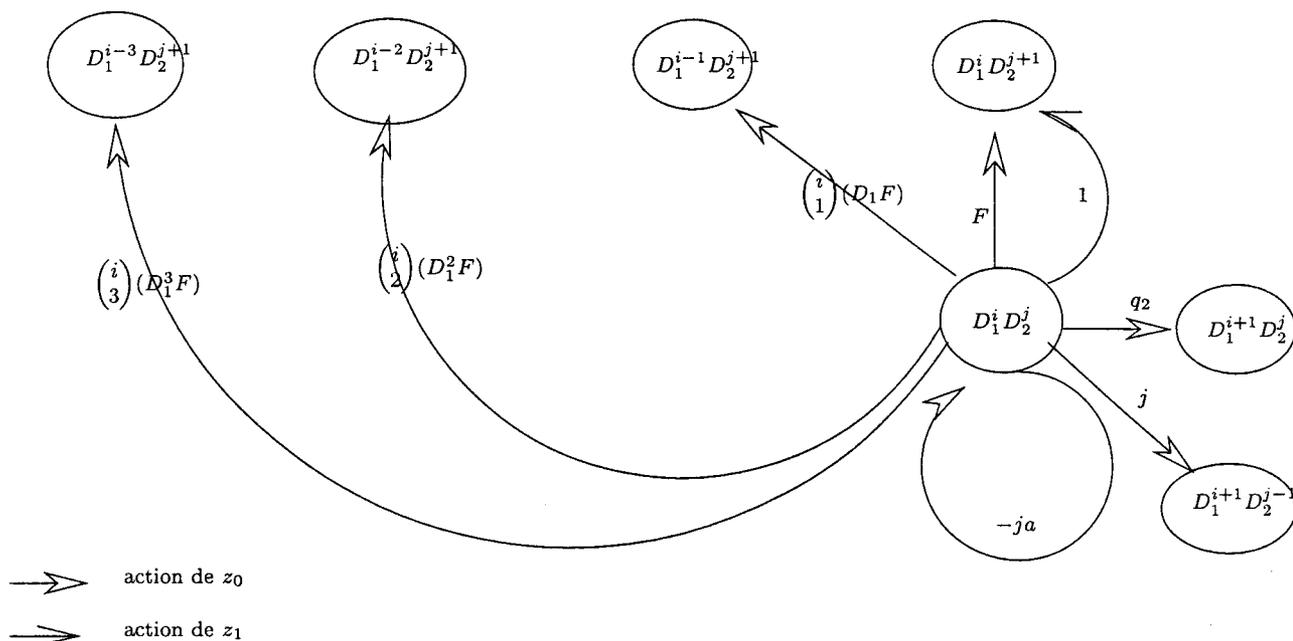


FIG. 6.1 – cellule de l'automate structurel de Duffing

– Les actions de g_0 et g_1 sur les états $D_1^i D_2^j$ sont données par:

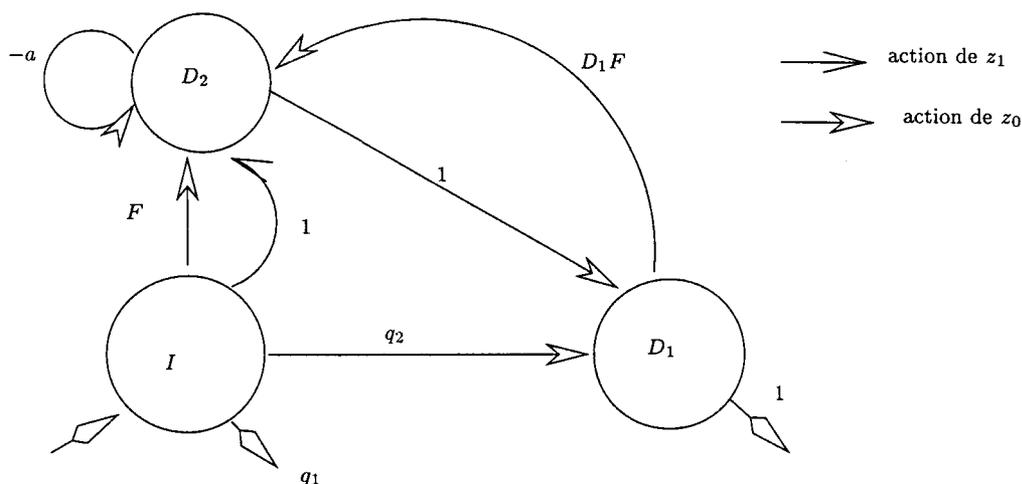
$$\begin{aligned}
 D_1^i D_2^j g_1 &= D_1^i D_2^{j+1} \\
 D_1^i D_2^j g_0 &= F D_1^i D_2^{j+1} \\
 &\quad + \binom{i}{1} (D_1 F) D_1^{i-1} D_2^{j+1} + \binom{i}{2} (D_1^2 F) D_1^{i-2} D_2^{j+1} \\
 &\quad + \binom{i}{3} (D_1^3 F) D_1^{i-3} D_2^{j+1} \\
 &\quad - ja D_1^i D_2^j + q_2 D_1^{i+1} D_2^j + j D_1^{i+1} D_2^{j-1}.
 \end{aligned}$$

conformément au schéma donnant la cellule de l'automate (figure 6.1).

L'automate tronqué est alors obtenu en sélectionnant les états $D_1^i D_2^j$ rencontrés par un chemin réussi de longueur $\leq k$. Pour $k=3$, nous obtenons l'automate tronqué représenté par le schéma correspondant (figure 6.2).

Cet automate contient 3 états. Il est associé à un système bilinéaire (B_1) de dimension 3, donné par les équations:

$$(B_1) \begin{cases} \begin{pmatrix} \bar{q}_1^{(1)} \\ \bar{q}_2^{(1)} \\ \bar{q}_3^{(1)} \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 \\ q_{2|_0} & 0 & 1 \\ F_{|_0} & D_1(F)_{|_0} & a \end{pmatrix} \begin{pmatrix} \bar{q}_1 \\ \bar{q}_2 \\ \bar{q}_3 \end{pmatrix} + \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} \bar{q}_1 \\ \bar{q}_2 \\ \bar{q}_3 \end{pmatrix} u(t) \\ y(t) = \lambda \bar{q}(t) \\ \lambda = (q_{1|_0} \ 1 \ 0) \end{cases}$$


 FIG. 6.2 – automate structural de Duffing pour $k=3$

6.4.2 Bilinéarisation par l'automate géométrique

Cette méthode est inspirée de la méthode de Carleman en remplaçant le produit tensoriel du vecteur d'état par le produit symétrique.

Considérons l'image miroir de la série génératrice:

$$\tilde{s} = \sum_{l \geq 0} \sum_{i_j=0}^m g_{i_l} [g_{i_{l-1}} \cdots g_{i_1} \circ h|_{q(0)}] z_{i_1} \cdots z_{i_l}$$

Cela suggère de définir un automate décrivant l'action à gauche, des champs de vecteurs g_r sur les $(q_1^{i_1} \cdots q_n^{i_n})_{i_1, \dots, i_n \geq 0}$.

On pose alors la définition suivante:

Définition 6.4.2 *Le \mathbb{R} -automate géométrique associé au système (Σ) est un 5-uple $\{Z, E, *, V, l\}$ où*

- Z est l'alphabet $\{z_0, \dots, z_m\}$
- E est l'espace vectoriel engendré par la famille $(q_1^{i_1} \cdots q_n^{i_n})_{i_1, \dots, i_n \geq 0}$
- $*$ est l'action à gauche des g_r sur les $(q_1^{i_1} \cdots q_n^{i_n})_{i_1, \dots, i_n \geq 0}$ définie par les dérivées de Lie:

$$q_1^{i_1} \cdots q_n^{i_n} \rightarrow g_r \circ (q_1^{i_1} \cdots q_n^{i_n})$$

- V est le vecteur initial associé à l'observation:

$$h = \sum_{n \geq 0} \sum_{i_1, \dots, i_n \geq 0} h_{i_1} \cdots h_{i_n} q_1^{i_1} \cdots q_n^{i_n}$$

- l est la forme linéaire définie par le vecteur ligne dont les composantes sont les évaluations des états terminaux en $t = 0$:

$$S = (1 \quad q_{1|_0} \quad \cdots \quad q_{n|_0} \quad q_1 q_{2|_0} \quad \cdots)$$

On obtient ainsi l'automate géométrique. En tronquant cet automate aux états visités par des chemins réussis de longueur $\leq k$, on obtient un automate fini, associé à un système bilinéaire approximant (Σ) à l'ordre k .

Remarquons qu'une "troncature de Volterra" est obtenue en supprimant de l'automate géométrique, les états $q_1^{i_1} \cdots q_n^{i_n}$ pour lesquels $i_1 + \cdots + i_n > k$.

Reprenons le même exemple pour illustrer cette méthode:

Exemple 6.4.2 Equation de Duffing:

$$y^{(2)} + ay^{(1)} + b^2y + cy^3 = u(t)$$

Le \mathbb{R} -automate géométrique est défini par:

- $Z = \{z_0, z_1\}$
- E est le \mathbb{R} -espace vectoriel engendré par $(q_1^i q_2^j)_{i,j \geq 0}$
- La forme linéaire l est définie par le vecteur ligne

$$S = (1 \quad q_{1|_0} \quad q_{2|_0} \quad q_1 q_{2|_0} \quad \cdots)$$

- $V = (0 \quad 1 \quad \cdots)^t$
- Les actions de g_0 et g_1 sur les états $q_1^i q_2^j$ sont données par:

$$\begin{aligned} g_0 \circ (q_1^i q_2^j) &= -j a q_1^i q_2^j - j b^2 q_1^{i+1} q_2^{j-1} - j c q_1^{i+3} q_2^{j-1} + i q_1^{i-1} q_2^{j+1} \\ g_1 \circ (q_1^i q_2^j) &= j q_1^i q_2^{j-1} \end{aligned}$$

conformément au schéma donnant la cellule de l'automate (figure 6.3).

Pour $k = 3$, on obtient un automate tronqué contenant 5 états (figure 6.4).

Cet automate décrit le système bilinéaire (B_2) dont la dimension de l'espace d'état est 5. Ses équations d'état sont les suivantes:

$$(B_2) \left\{ \begin{aligned} \begin{pmatrix} \bar{q}_1^{(1)} \\ \bar{q}_2^{(1)} \\ \bar{q}_3^{(1)} \\ v \bar{q}_4^{(1)} \bar{q}_5^{(1)} \end{pmatrix} &= \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -b^2 & 0 & 0 \\ 0 & 1 & -a & 0 & 0 \\ 0 & 0 & -c & 0 & 0 \\ 0 & 0 & 0 & 3 & 0 \end{pmatrix} \begin{pmatrix} \bar{q}_1 \\ \bar{q}_2 \\ \bar{q}_3 \\ \bar{q}_4 \\ \bar{q}_5 \end{pmatrix} + u(t) \begin{pmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \bar{q}_1 \\ \bar{q}_2 \\ \bar{q}_3 \\ \bar{q}_4 \\ \bar{q}_5 \end{pmatrix} \\ y(t) &= \lambda \bar{q}(t) \\ \lambda &= (1 \quad q_{1|_0} \quad q_{2|_0} \quad q_{1|_0}^3 \quad q_1^2 q_{2|_0}) \end{aligned} \right.$$

Remarque 6.4.1 Dans la méthode de Carleman présentée par Rugh [12], un état initial fictif a été ajouté. Une présentation sur ce même exemple en est faite dans [7].

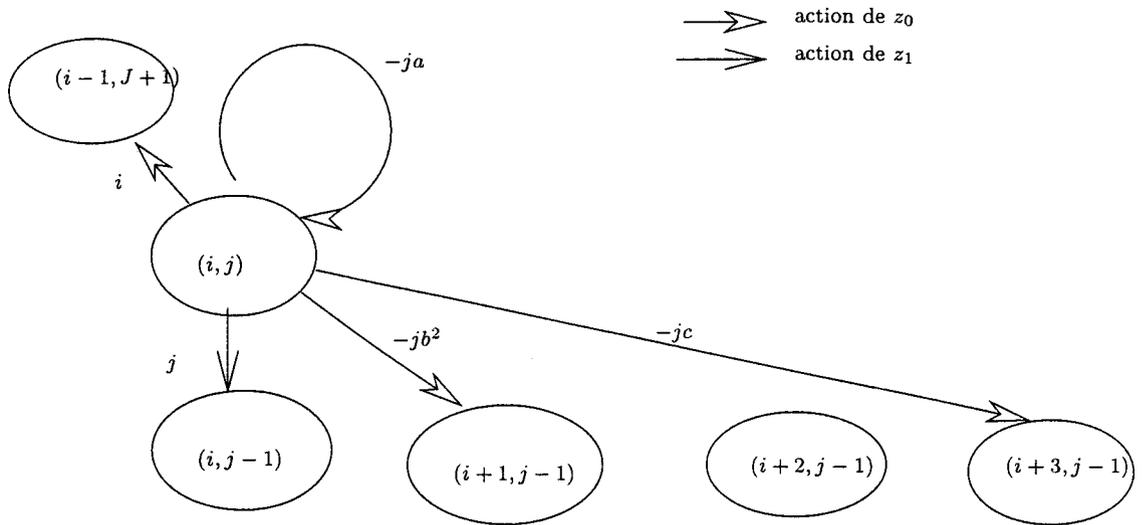


FIG. 6.3 - cellule de l'automate géométrique de Duffing

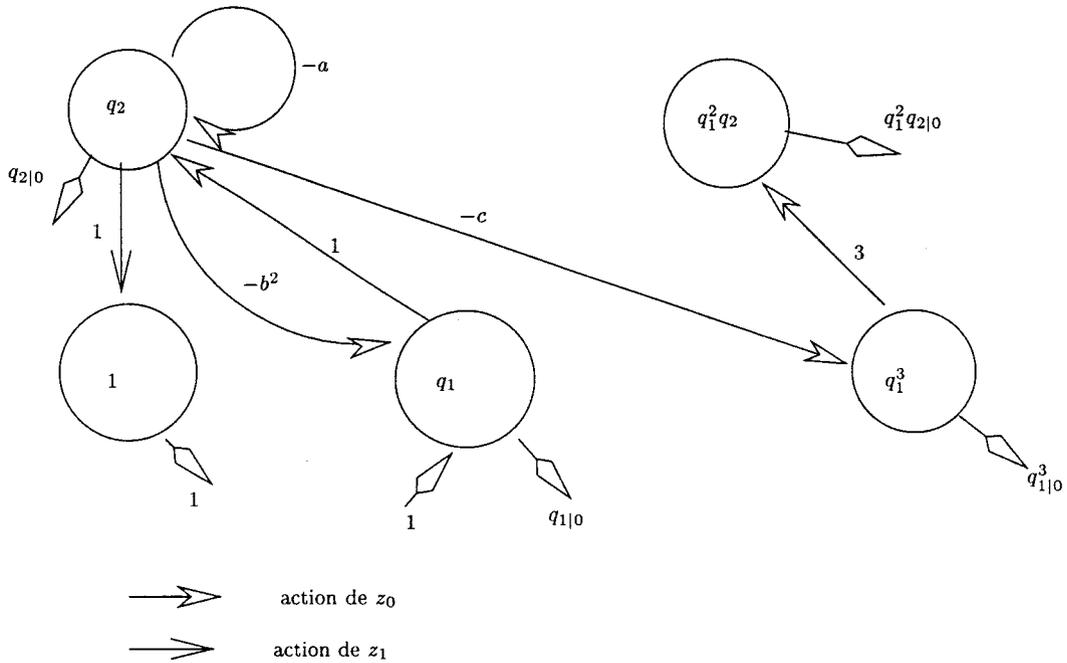


FIG. 6.4 - automate géométrique de Duffing pour $k = 3$

6.5 Relation avec les approximants de type Padé

Lorsqu'un système différentiel à une entrée, n'a pas de terme forcé, une méthode connue pour construire un approximant de type Padé de ce système, consiste à construire un approximant de type Padé de sa série génératrice commutative [1]. Cette méthode peut se généraliser à un système (Σ) . Elle consiste à calculer un approximant rationnel de la série génératrice en variables non commutatives de (Σ) . Nous proposons alors la définition suivante [11]:

Définition 6.5.1 *Une famille d'approximants de type Padé d'un système dynamique (Σ) est une construction itérative convergente de systèmes bilinéaires, obtenus en calculant des approximants rationnels de la série génératrice de (Σ) .*

Comme on l'a vu au chapitre 3, les méthodes par la matrice de Hankel (de la série génératrice) ou par troncature d'automate (associé à la série génératrice), fournissent des approximants de type Padé de la série génératrice de (Σ) . Par conséquent, les approximants de (Σ) obtenus par ces méthodes sont aussi des approximants de type Padé de ce système dynamique. Ils vérifient de plus la condition d'approximation à l'ordre k .

6.6 Que se passe-t-il si le système (Σ) est bilinéaire?

Nous avons constaté au chapitre 3, que les approximants rationnels de la série génératrice appartiennent, dans ce cas, à une suite stationnaire à partir d'un certain ordre k_0 . Par conséquent, à partir de cet ordre, la série génératrice de (Σ) coïncide avec ses approximants rationnels. Le système (Σ) et n'importe lequel de ses approximants ont donc le même système réduit, à partir d'un certain ordre. Ce système réduit est donné par la méthode de la matrice de Hankel.

Remarque 6.6.1 *Comme l'a montré Brockett [2], la méthode de Carleman permet de calculer les noyaux exacts de Volterra si le système est initialisé en un point d'équilibre. Un calcul équivalent, à base de Combinatoire énumérative, a été présenté par X. Viennot [11] et F.Lamnabhi-Lagarrigue [10]. Le même résultat est retrouvé ici en tronquant l'automate géométrique aux états $q_1^{i_1} \cdots q_n^{i_n}$ dont le degré total vérifie:*

$$\sum_{j=1}^n i_j \leq k.$$

On obtient alors un \mathbb{R} -automate fini et donc un nouvel approximant bilinéaire.

6.7 Comparaison des approximants bilinéaires

Nous récapitulons les propriétés des approximants proposés

- La dimension de l'espace d'état du système approximant obtenu par la méthode de la matrice de Hankel est minimale.
- Les approximants obtenus par troncature d'automate (structurel ou géométrique) permettent de passer aisément de l'ordre k à l'ordre $k + 1$, par simple enrichissement de l'automate.
- Il ne semble pas exister de relation d'ordre entre les dimensions des espaces d'état des systèmes approximants obtenus par troncature d'automate structurel ou géométrique, pour un même ordre k , pour un système (Σ) quelconque.
- Pour un système (Σ) dont la fonction d'observation n'est pas polynomiale, l'automate géométrique fournit un approximant qui n'est pas de dimension finie.

Pour toutes ces raisons, les approximants du système (Σ) par la méthode de la matrice de Hankel et par troncature de l'automate structurel nous paraissent les plus intéressants.

Une collaboration avec les laboratoires LSS(Gif sur Yvette), LACIM(Québec à Montréal), LABRI(Bordeaux), met en parallèle Combinatoire et Automates pour le calcul d'approximants bilinéaires [8]

Bibliographie

- [1] Brezenski C., *Padé type approximation and general orthogonal polynomials*, INSM50, Birkhauser, 1980.
- [2] Brockett R.W., *Volterra series and Geometric Control Theory*, Automatica, vol.12, pp. 167-176, 1976.
- [3] Hespel C., Jacob G., *Approximation of nonlinear systems by bilinear ones*, in "Algebraic and Geometric Methods in Nonlinear Control Theory", M.Fliess and M.Hazewinkel eds., D.Reidel Publishing Company, pp.511-520, 1986.
- [4] Hespel C., Jacob G., *Approximations bilinéaires de systèmes analytiques: exemples de listings*, CALSYF 7, pp. 181-219, Journées de calcul Formel, Luminy, 1988.
- [5] Hespel C., Jacob G., *Calcul des approximations locales bilinéaires de systèmes analytiques*, RAIRO APII, vol. 23, pp. 331-349, 1989.
- [6] Hespel C., Jacob G., *Approximation of nonlinear dynamic systems by rational series*, Theoret. Comput. Sciences, 79, pp.151-162, 1991.
- [7] Hespel C., *Truncated bilinear approximants: Carleman, finite Volterra, Padé-type, geometrical and structural automata*, Lecture Notes in Control and Information Sciences, vol. 165, "Algebraic Computing in Control", pp. 265-278, Springer-Verlag, 1991.
- [8] Hespel C., Jacob G., Lamnabhi-Lagarrigue F., Leroux P., Viennot X.G., *Algebraic, Combinatorial and Syntactic Techniques in Nonlinear Control*, Séries formelles et Combinatoire Algébrique, pp. 245-261, Bordeaux, 2-4 mai 1991.
- [9] Lamnabhi-Lagarrigue F., Crouch P.E., *A formula for iterated derivatives along trajectories of nonlinear systems*, Systems and Control letters 11, pp. 1-7, 1988.
- [10] Lamnabhi-Lagarrigue F., Leroux P., Viennot X.G., , *Combinatorial interpretation of Volterra series by bilinear approximations*, in "Analyse des Systèmes dynamiques contrôlés, Progress in Systems and Control Theory", Birkhauser, 1991.
- [11] Leroux P., Viennot X.G., *A combinatorial approach to nonlinear functional expansions: an introduction with example*, Algebraic and Computing Treatment of

Noncommutative Power Series , G.Jacob and C.Reutenauer eds., Theoret. Comput. Sciences, 1991.

- [12] Rugh W.J., *Nonlinear System Theory*, The John Hopkins University press, Baltimore and London.

Chapitre 7

Un tour d'horizon sur l'identification.

Des voies possibles pour l'identification algébrique

7.1 Introduction

Dans ce chapitre et les suivants, nous poserons le problème de l'Identification Algébrique:

Peut-on, à partir d'un jeu d'informations discrètes et exactes sur le comportement entrée/sortie d'une fonctionnelle causale, construire un algorithme calculant jusqu'à un ordre donné, les coefficients (en calcul exact) de sa série génératrice?

Avant d'aborder précisément ce problème, nous rappelons les résultats obtenus pour des problèmes voisins et explorons diverses voies.

Dans un cadre très général, nous rappelons que J.Sjoberg et al. présentent dans [21], un tour d'horizon sur les modélisations de boîtes noires dans l'Identification des systèmes. A.Juditsky et al. développent dans [8] un éventail de fondements mathématiques, qui sont à la base des algorithmes d'Identification de systèmes non linéaires.

Par ailleurs, d'importants résultats ont été obtenus par E.Walter [23], F.Ollivier [16] sur l'Identification de paramètres, et par Narendra et al. [15], sur l'Identification par réseaux neuronaux. Nous les résumons brièvement.

Puis nous proposons plusieurs voies pour résoudre le problème de l'Identification Algébrique:

– **Par des techniques d'observation:**

1) Selon une idée de M.Petitot, il s'agit d'identifier les coefficients de la série gé-

nératrice supposée polynômiale, i.e. identifier les coefficients de la fonction d'observation grâce à des mesures de la sortie à divers instants t pour des entrées convenables. Ce problème se ramène à celui du "Motion Planning" [4].

2) On se place sous les mêmes hypothèses qu'en 1). Mais l'identification se fait cette fois par les fonctionnelles différentiables. Il s'agit d'identifier les coefficients de la fonction d'observation grâce à un jeu d'entrées/sorties et leurs dérivées en $t = 0$.

– **Par la série génératrice:**

Le problème posé est de proposer un algorithme déterminant la série génératrice jusqu'au degré k , quand on connaît les développements de Taylor (en $t = 0$) des entrées et les développements de Taylor (en $t = 0$) des sorties correspondantes. Après une analyse du problème [5, 6], nous fournissons un algorithme.

Le traitement de ce problème passe par 3 étapes:

- * Ecrire un système d'équations linéaires à partir des dérivées itérées de la sortie.
- * Identifier les coefficients des multidérivées des entrées.
- * Identifier les mots de même image commutative.

Un début de démonstration de l'identifiabilité algébrique est proposé. En tout état de cause, nous fournissons l'algorithme suivant:

Etant donné un ordre $k \geq 0$, des tests établissent une condition suffisante d'identifiabilité de la série génératrice tronquée aux mots de longueur $l \leq k$.

Il est à noter que l'Identification algébrique s'articule naturellement avec d'autres applications des séries formelles aux systèmes dynamiques: elle peut être utilisée comme point de départ pour de multiples sujets tels que l'approximation bilinéaire, la réalisation minimale, etc...

Des idées de développement par des enchaînements d'identification, d'approximations bilinéaires, de simulations, etc ... sont proposées en fin du chapitre "Présentation du mémoire".

7.2 Tour d'horizon sur les modélisations de boîtes noires

7.2.1 Structures des modèles

On trouve dans [21], une description des choix et des stratégies pour arriver à un bon modèle, à partir de données observées.

Dans ce cas, le problème d'identification de système est le suivant:

On a observé les entrées $a(t)$ et les sorties correspondantes $y(t)$ de sorte qu'on dispose de

$$a^t = [a(1), \dots, a(t)], \quad y^t = [y(1), \dots, y(t)]$$

Alors,

$$y(t) = g(a^{t-1}, y^{t-1}) + v(t)$$

Le problème est de trouver g . On cherche g dans une famille de fonctions que l'on peut paramétriser par un paramètre θ de dimension finie. On pose:

$$g(a^{t-1}, y^{t-1}, \theta) = g(\phi(t), \theta), \quad \phi(t) = \phi(a^{t-1}, y^{t-1})$$

La recherche d'un bon modèle se ramène à la résolution du double problème suivant

- Comment choisir le vecteur de régression $\phi(t)$?
- Comment choisir l'application non linéaire g de l'espace de régression vers l'espace des sorties?

Plusieurs choix de vecteurs de régression sont proposés, pour des modèles linéaires ou non linéaires. De multiples applications g sont proposées sur le principe suivant:

$$g(\phi, \theta) = \sum_k \alpha_k g_k(\phi)$$

où les fonctions de base g_k sont obtenues en paramétrisant une fonction de base mère:

$$g_k(\phi) = \xi(\phi, \beta_k, \gamma_k)$$

par exemple les séries de Fourier, les ondelettes, ...

Des critères sur la qualité et la flexibilité du modèle sont fournis.

7.2.2 Fondements mathématiques

On trouve dans [8] une discussion sur les différents aspects des fondements mathématiques de l'identification des systèmes non linéaires. En effet, la qualité de l'identification est un compromis entre la puissance d'expression du modèle (dépendant du nombre de paramètres utilisé pour le décrire) et l'erreur stochastique (proportionnelle au nombre de paramètres).

Les fondements mathématiques sont variés: Citons les problèmes de régression non paramétrique avec étude statistique (variables aléatoires), de régression non paramétrique avec étude déterministe (variables non aléatoires), les problèmes d'estimation de l'erreur, les problèmes d'interpolation, extrapolation (à partir d'un nombre fini de données $[x_k, f(x_k)]$, obtenir $f(x)$ en d'autres points que ceux observés afin de calculer la fonction de régression f).

Après ce tour d'horizon, nous présentons deux modes pour aborder l'identification de paramètres.

7.3 Identification de paramètres

De nombreux auteurs ont travaillé à l'identification de paramètres, soit selon un modèle choisi à priori, soit selon un modèle à définir.

7.3.1 Selon un modèle choisi à priori, dépendant d'un nombre fini de paramètres

Un modèle peut être par exemple, un système linéaire [22, 9], bilinéaire et le problème consiste à identifier les coefficients d'une représentation bilinéaire.

Un modèle peut être un système dont la série de Volterra est composée d'un nombre fini de noyaux, le problème consistant à identifier ces noyaux de Volterra.

L'identification des paramètres se fait en ajustant le modèle sur les mesures effectuées, les techniques employées étant de nature statistique.

7.3.2 Selon un modèle adapté, dépendant d'un nombre fini de paramètres

► Des structures de modèles

Dans [24], on trouve un tour d'horizon sur différentes structures de modèles dépendant d'un nombre fini de paramètres.

Il y a lieu d'abord de distinguer les modèles comportementaux (approchant le comportement observé sans connaissance du processus qui a engendré les données) et les modèles basés sur la connaissance (du processus qui a engendré les données). De multiples modèles y sont passés en revue: linéaires, non linéaires, continus, discrets, déterministes, stochastiques, et des critères de choix sont présentés.

► Recherche d'un modèle adapté

Dans [16], l'étude préalable au choix d'un bon modèle est détaillée. **Plusieurs phases préliminaires sont nécessaires:**

- choix d'un protocole expérimental définissant les actions possibles et les mesures effectuelles.
- proposition d'une loi d'évolution du système i.e. choix de grandeurs d'état et de paramètres internes.
- détermination d'un ensemble de valeurs pour les paramètres internes.

Les techniques proposées s'appuient sur l'algèbre différentielle. Plus précisément, le problème consiste à déterminer un ensemble de valeurs pour les paramètres internes,

compatibles avec les mesures effectuées. Cette évaluation définit un modèle, permettant ensuite de prédire l'évolution du système et éventuellement de le commander.

Il importe de pouvoir identifier avec certitude le modèle correspondant au processus réel, i.e. de déterminer un modèle unique en fonction de l'expérience.

• Une classe de modèle est définie:

Un modèle paramétré ou structure sur $k = \mathbb{R}$ ou C est la donnée

- d'entiers n, m, p, r .
- d'un ensemble $D \subset k^r$ de **paramètres admissibles**.
- d'un ensemble $E \subset k^n$ de **vecteurs d'état admissibles**.
- d'une classe d'applications $U \subset \text{Appl}(\mathbb{R}^+, k^m)$.
- de 2 applications

$$\begin{aligned} f &: E \times D \times \mathbb{R}^+ \times k^m \rightarrow E \\ g &: E \times D \rightarrow k^p \end{aligned}$$

- d'une famille d'applications:

$$h_i : D \rightarrow k \quad \forall i \quad 1 \leq i \leq m$$

On appelle système d'équations d'état du modèle paramétré, le système

$$\begin{cases} \frac{dx}{dt} = f(x, \theta, t, a(t)) \\ y = g(x, \theta) \end{cases}$$

Et on appelle conditions initiales, les équations du système:

$$x_i(0) = h_i(\theta)$$

où

- $t \in \mathbb{R}^+$ est le temps
- $x \in E \subset k^n$ est le **vecteur d'état**
- $y \in k^p$ est le **vecteur d'observation**
- $\theta \in D \subset k^r$ est le **vecteur des paramètres internes**.

En donnant une valeur particulière θ_0 , on obtient un **modèle** de la structure noté $M(\theta_0)$.

• Le comportement Entrée/Sortie d'une structure étant trop peu maniable, on lui substitue une fonction des paramètres plus simple, mais en un sens



équivalente:

- On appelle comportement entrée/sortie du modèle $M(\theta)$, l'application

$$C : U \rightarrow \text{Appl}(\mathbb{R}^+, \mathbb{R}^p)$$

Le comportement entrée/sortie de la structure M est $C(\theta)$ qui à θ associe le comportement du modèle $M(\theta)$.

- On appelle résumé (resp. résumé exhaustif), une application $\rho : D \rightarrow E$ telle que:

$$\begin{aligned} \forall(\theta, \theta') \in D^2 \quad C(\theta) = C(\theta') &\implies \rho(\theta) = \rho(\theta') \\ (\text{resp. } \forall(\theta, \theta') \in D^2 \quad C(\theta) = C(\theta') &\iff \rho(\theta) = \rho(\theta')) \end{aligned}$$

Il reste alors à définir la notion d'identifiabilité.

• On définit la notion d'identifiabilité comme propriété de modèle et comme propriété de structure.

- On dit qu'un modèle $M(\theta)$ est localement identifiable s'il existe un voisinage ouvert O de θ dans D tel que:

$$\forall \theta' \in O \quad C(\theta) = C(\theta') \implies \theta = \theta'.$$

- Une structure M est structurellement globalement identifiable s'il existe un sous ensemble S de D de mesure nulle telle que:

$$\forall \theta \in D \setminus S \quad \forall \theta' \in D \quad C(\theta) = C(\theta') \implies \theta = \theta'$$

Il reste à résoudre le problème de l'identifiabilité.

On considère des structures admettant des résumés exhaustifs rationnels. On dispose alors de la proposition et du corollaire suivants:

Proposition 7.3.1 *Soit M une structure admettant un résumé exhaustif rationnel $\rho : D \rightarrow \mathbb{R}^r$. Alors, M est structurellement localement identifiable ssi le rang générique de la matrice jacobienne de ρ est égal à la dimension de l'espace des paramètres.*

Corollaire 7.3.1 *Sous les mêmes hypothèses, M est structurellement localement identifiable ssi le corps des fractions $k(\theta)$ est une extension algébrique de $k(\rho)$.*

Le problème qui demeure est de trouver des résumés exhaustifs rationnels. On sait en trouver dans le cas d'une structure linéaire stationnaire avec conditions initiales nulles.

L'obtention de résumés exhaustifs se fait en appliquant des résultats d'algèbre différentielle. Ainsi dans le cas où f, g, h sont des polynômes ou des fractions rationnelles, satisfaisant à des conditions sur la classe des commandes admissibles, sur l'ensemble des paramètres admissibles, sur la genericité de la solution, sur la non singularité des fonctions y en $t = 0$, des algorithmes sont proposés en SCRATCHPAD II.

Citons aussi l'article de L.Ljung et T.Glad [10], proposant une représentation entrée/sortie des systèmes non linéaires, à l'aide de l'algèbre différentielle et celui de L.Denis-Vidal et G.Joly-Blanchard [3], aménageant l'algorithme des auteurs précédents.

7.3.3 Selon une approche neuronale

Il s'agit de définir une approche sans modèle préconçu, par l'apprentissage.

► En temps discret

Un tour d'horizon est présenté dans [12]. Des applications sont disponibles par exemple dans [13, 14].

Narendra et Parthasarathy [15] ont introduit plusieurs modèles pour représenter les systèmes non linéaires monovariabiles, controllables et observables, à l'aide d'équations aux différences.

Le modèle non linéaire considéré peut être très général, par exemple:

$$y(k) = f(y(k-1), \dots, y(k-na), a(k-1), \dots, a(k-nb))$$

où l'on suppose que

$$f : \mathbb{R}^{na+nb} \rightarrow \mathbb{R}$$

est différentiable et $nb \leq na$.

Un réseau neuronal est instruit pour émuler la fonction non linéaire f et produire la même sortie que le système considéré. C'est :

$$\hat{y}(k) = N(y(k-1), \dots, y(k-na), a(k-1), \dots, a(k-nb))$$

- $\hat{y}(k)$ étant l'estimation de la sortie au temps k
- N étant le réseau émulant la fonction f

Une structure appropriée du réseau est proposée, après quelques tests : nombres de couches et de noeuds.

Le problème de l'identification se réduit alors à présenter au réseau des jeux d'entrée sortie et d'ajuster les paramètres (les poids) de manière à minimiser l'erreur quadratique entre les sorties du réseau et celles du système. Plusieurs algorithmes d'apprentissage basés sur le gradient sont alors proposés tels que "backpropagation-through-time

algorithm” ou “real-time-recurrent-algorithm”.

Par exemple, pour le cas de la colonne à distiller, le réseau neuronal choisi contient une couche cachée de 10 noeuds. Il utilise des fonctions d’activation non linéaires (sigmoïdes) et un facteur d’oubli $\eta = 0.1$. Les données pour cet exemple, sont composées de 2 entrées et de 2 sorties. La procédure d’apprentissage proposée utilise “Neural Network Toolbox” de Matlab 4.2 [13].

► En temps continu

On trouve différentes approches dans la littérature :

- Le modèle considéré dans [11] est un système dynamique du premier ordre de la forme:

$$\dot{x} = f(x)$$

avec $x \in \mathbb{R}$ et tel que ce système ait un nombre fini de points d’équilibre dans un intervalle fermé $I \subset \mathbb{R}$.

Les auteurs recherchent une “représentation” d’un tel système par réseaux neuronaux. Par “représentation d’un système dynamique”, les auteurs entendent

- *Approximation des trajectoires
- *Maintien de la structure de la trajectoire.

Pour cela, un réseau neuronal dynamique essaie de dupliquer les propriétés structurelles du système original. Un tel réseau du premier ordre a pour équation :

$$\dot{x} = N(x) = \sum_{i=0}^k c_i \sigma(\omega_i x + \theta_i)$$

où σ est la fonction sigmoïde donnée par :

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

Les auteurs montrent la faisabilité du problème et annoncent de futurs travaux relatifs aux systèmes dynamiques d’ordres supérieurs.

- T.Chen et H.Chen [1] étendent les capacités des réseaux neuronaux à approximer des fonctions statiques, aux fonctionnelles définies dans $C[a, b]$. Ce résultat peut servir à approximer la sortie d’un système dynamique en n’importe quel instant. Olurotimi [17] propose une méthode dans laquelle les paramètres d’un réseau neuronal récurrent sont déterminés par apprentissage en commande prédictive.

7.4 Enoncé du problème

Le comportement entrée/sortie du système est défini par la fonctionnelle:

$$a \rightarrow y$$

Le problème posé ici est le suivant:

Peut-on, à partir d'un jeu d'informations discrètes et exactes sur le comportement entrée/sortie d'une fonctionnelle causale, construire un algorithme calculant jusqu'à un ordre donné, les coefficients (en calcul exact) de sa série génératrice?

Ce problème peut encore se formuler comme suit:

Existe-t-il un algorithme qui permette de **déterminer explicitement la fonctionnelle entrée/sortie jusqu'à un ordre k donné, quand on connaît soit des mesures de la sortie à divers instants pour des entrées convenablement choisies (cf M.Petitot), soit les développements de Taylor (en $t = 0$) des entrées du système et les développements de Taylor (en $t = 0$) des sorties associées (cf C.Hespel-G.Jacob)?**

Avant de développer plusieurs voies possibles pour résoudre ce problème, nous présentons des rappels.

7.4.1 Rappels

Certaines notations figurant dans ce paragraphe ont été définies au chapitre 4. **On choisit comme modèles, les systèmes analytiques (Σ) affines en la commande**

$a = \{a_i\}_{0 \leq i \leq m}$ de la forme:

$$\begin{cases} \dot{q}(t) &= g_0(q) + \sum_{i=1}^m a_i(t)g_i(q) \\ y &= h(q) \end{cases}$$

tels que

- l'état $q = (q^s)_{1 \leq s \leq N} \in Q$, N -variété réelle analytique.
- $\{g_i\}_{0 \leq i \leq m}$ sont des champs de vecteurs définis dans un voisinage de $q(0)$. Ils engendrent la \mathbb{R} -algèbre de Lie notée $L(G)$.
- $h : Q \rightarrow \mathbb{R}^p$ est une fonction d'observation analytique définie dans un voisinage de $q(0)$.
- les entrées $\{a_i\}_{0 \leq i \leq m}$ sont continues par morceaux.

La fonction $y : t \in \mathbb{R} \rightarrow h(q(t)) \in \mathbb{R}^p$ désigne la sortie du système.

Nous associons à (Σ) l'alphabet de codage $Z = \{z_0, z_1, \dots, z_m\}$.

Les opérateurs différentiels $\mathcal{Y}(w)$ et l'intégrale itérée $\int_0^t \delta_a(w)$ sont définis pour tout mot w de Z^* conformément au chapitre 4.

Rappelons la formule de Fliess:

$$y(t) = \sum_{w \in Z^*} [\mathcal{Y}(w) \circ h_{|q_0}] \int_0^t \delta_a(w).$$

- $G = \sum_{w \in Z^*} [\mathcal{Y}(w) \circ h_{|q_0}] w$ est la série génératrice, ou 'série de Fliess' de (Σ) .

- $C_a(t) = \sum_{w \in Z^*} \left[\int_0^t \delta_a(w) \right] w$ est la 'série de Chen' de l'entrée a .

Nous déduisons

$$\begin{aligned} y(t) &= \sum_{w \in Z^*} \langle G|w \rangle \langle C_a(t)|w \rangle \\ &= \langle G || C_a(t) \rangle. \end{aligned}$$

(où $||$ signifie somme infinie).

Lemme 7.4.1 La dérivée de la série de Chen est

$$\frac{d}{dt} C_a = C_a \cdot \mathcal{L}_a$$

où $\mathcal{L}_a = \sum_{0 \leq i \leq m} a_i \cdot z_i$.

Preuve: Notons " $P \triangleleft$ " la résiduelle à gauche par le polynôme P définie par:

$$\langle P \triangleleft G|w \rangle = \langle G|w.P \rangle$$

Nous obtenons

$$\frac{d}{dt} \langle C_a|w \rangle = \sum_z \left[\int_0^t \delta(z \triangleleft w) \right] a_z(t).$$

et donc

$$\frac{d}{dt}\langle \mathcal{C}_a | w \rangle = \sum_z \langle \mathcal{C}_a | z \triangleleft w \rangle a_z(t).$$

C'est exactement:

$$\frac{d}{dt}(\mathcal{C}_a) = \mathcal{C}_a \cdot \sum_i a_i \cdot z_i$$

Fin de preuve

Alors,

$$y^{(1)}(t) = \langle G | \mathcal{C}_a(t) \mathcal{L}_a(t) \rangle$$

7.5 Identification par des techniques d'observations

Dans ce paragraphe, on se limite au cas où l'algèbre des observations est une algèbre finiment engendrée. Cela revient à dire que le système est supposé nilpotent d'ordre s .

L'idée est d'utiliser la base de Lyndon $(l_i)_{l_i \in \mathcal{L}}$. Pour indiquer les coordonnées locales, on choisit les mots de Lyndon de longueur $\leq s$ et on obtient :

$$\xi = (\xi_1, \dots, \xi_l, \dots, \xi_r)$$

Au lieu d'identifier les coefficients de la série génératrice $G \in \mathbb{R}\langle\langle Z \rangle\rangle$, polynôme en variables non commutatives, on identifie les coefficients de $h \in \mathbb{R}[\xi]$, polynôme en variables commutatives, où

$$\begin{cases} \xi^\alpha &= \frac{1}{\alpha_1! \dots \alpha_k!} \xi_{l_1}^{\alpha_1} \dots \xi_{l_k}^{\alpha_k} \\ h &= \sum_{|\alpha| \leq s} h_\alpha \xi^\alpha \end{cases}$$

7.5.1 Rappels et notations

On définit l'évaluation $\mathcal{E}(w)$ du mot w comme l'intégrale itérée $\int_0^t \delta_a(w)$ donc :

$$\begin{cases} \mathcal{E}(\epsilon)(t) &= 1 \\ \mathcal{E}(wz_i)(t) &= \int_0^t a_i(\tau) \mathcal{E}(w)(\tau) d\tau \end{cases}$$

Par ailleurs, G étant la série génératrice

$$\begin{cases} y(t) &= \sum_{w \in Z^*} [\mathcal{Y}(w) \circ h_{|a_0}] \mathcal{E}(w)(t) \\ &= \mathcal{E}(G) \end{cases}$$

La définition de $\mathcal{Y}(w)$ s'étend aux polynômes p de l'algèbre $\mathbb{R}\langle Z \rangle$. Rappelons que $\mathbb{R}\langle Z \rangle$ est l'algèbre enveloppante de $Lie\langle Z \rangle$.

A toute base B de l'espace vectoriel $\mathbb{R}\langle Z \rangle$, on associe la base duale B^* de $\mathbb{R}\langle\langle Z \rangle\rangle$ définie par:

$$\forall a, b \in B \quad \langle b^* | a \rangle = \delta_a^b$$

Tout polynôme $p \in \mathbb{R}\langle Z \rangle$ s'écrit donc :

$$\left\{ \begin{array}{l} p = \sum_{b \in B} \langle p | b^* \rangle b \\ = \sum_{b \in B} \langle p | b \rangle b^* \end{array} \right.$$

La série génératrice peut alors s'écrire :

$$G = \sum_{b \in B} \left[\mathcal{Y}(b) \circ h_{|_{q_0}} \right] b^*$$

Or, le théorème de Poincaré-Birkhoff-Witt permet de construire des bases de $\mathbb{R}\langle Z \rangle$ à partir d'une base de $Lie\langle Z \rangle$:

Théorème 13 *Si $E = \{e_k \mid k \in \mathbb{N}\}$ est une base de $Lie\langle Z \rangle$, alors*

$$B = \{e_1^{\alpha_1} e_2^{\alpha_2} \cdots e_s^{\alpha_s} \mid s, \alpha_i \in \mathbb{N}\}$$

est une base de l'algèbre enveloppante $\mathbb{R}\langle Z \rangle$.

7.5.2 Identification par la méthode du “motion planning”

Le problème posé par M.Petitot, consiste à identifier les coefficients de la série génératrice supposée polynomiale, autrement dit à **identifier les coefficients de la fonction d'observation h grâce à des mesures de la sortie $y(t)$ à divers instants t pour des entrées $a_i(t)$ choisies convenablement.**

En s'inspirant des travaux de C.Reutenauer, M.Petitot et N.Oussous montrent que toute série polynomiale admet une réalisation analytique minimale polynomiale [18, 19, 20]. **C'est à partir de cette démonstration que M.Petitot propose la voie suivante pour résoudre le problème de l'identification.**

– On suppose donc le système nilpotent libre. On dispose du théorème suivant:

Théorème 14 *Pour un nombre de champs de vecteurs et un ordre de nilpotence fixés, deux systèmes nilpotents libres se déduisent par un changement de coordonnées locales.*

On obtient une réalisation en considérant une base homogène (base de Lyndon), pour l'algèbre $Lie\langle Z \rangle$.

La base $PBWL$ constitue une base de $\mathbb{R}\langle Z \rangle$:

$$Q_w = [l_1]^{\alpha_1} [l_2]^{\alpha_2} \dots [l_k]^{\alpha_k}$$

La base duale $PBWL^*$ constitue une base de $\mathbb{R}\langle\langle Z \rangle\rangle$:

$$S_w = \frac{1}{\alpha_1! \dots \alpha_k!} S_{l_1}^{\cup \alpha_1} \dots S_{l_k}^{\cup \alpha_k}$$

pour $w = l_1^{\alpha_1} l_2^{\alpha_2} \dots l_k^{\alpha_k}$, w étant un produit de mots de Lyndon décroissants, et $[l_i]$ désignant le polynôme de Lie associé à l_i .

– On déduit la réalisation

$$h = \sum_{|\alpha| \leq s} h_\alpha \xi^\alpha \in \mathbb{R}[\xi]$$

où

$$\begin{cases} \alpha &= (\alpha_1, \dots, \alpha_k) \\ \xi &= (\xi_0, \dots, \xi_l, \dots) \end{cases}$$

lorsque l balaie la liste des mots de Lyndon de longueur inférieure ou égale à s . avec

$$\begin{cases} b &= [l_1]^{\alpha_1} [l_2]^{\alpha_2} \dots [l_k]^{\alpha_k} \\ \xi^\alpha &= \frac{1}{\alpha_1! \dots \alpha_k!} \xi_{l_1}^{\alpha_1} \dots \xi_{l_k}^{\alpha_k} \\ h_\alpha &= \langle G | b \rangle \end{cases}$$

– **Identifier le comportement entrée/sortie d'un système dont la série est polynômiale revient à identifier les coefficients h_α de la fonction d'observation h .**

Pour cela, M.Petitot propose d'effectuer des mesures de la sortie $y(t)$ à divers instants t pour des entrées choisies convenablement afin d'identifier ces coefficients, dans le cas où le système est sans dérive.

L'idée de la preuve repose sur les 3 étapes suivantes :

• **Résolution du problème algébrique :**

$h(X_i)$ étant un polynôme de degré s , déterminer un jeu de variables (X_i) tel que la connaissance des X_i détermine complètement h .

• **Par "Motion Planning", intégration de commande pour les systèmes nilpotents libres sans dérive (méthode du "Motion Planning" [4]):**

On utilise la dynamique pour calculer les grandeurs $h(X_i)$. Puis par la méthode du "Motion Planning", on calcule l'entrée a_i permettant d'atteindre la cible $h(X_i)$.

• **Ajustage par tâtonnement.**

Le calcul se termine par la résolution dans l'algèbre commutative du système linéaire obtenu en écrivant :

$$\sum h_\alpha \xi^\alpha = c_\alpha$$

Cette preuve s'appuie sur le lemme :

Lemme 7.5.1 *Deux fonctions polynômiales de $\mathbb{R}^n \rightarrow \mathbb{R}^p$ qui coïncident sur un ouvert de \mathbb{R}^n sont égales.*

7.5.3 Identification par les fonctionnelles différentiables

On se place sous les mêmes hypothèses que dans le paragraphe précédent: On suppose la série génératrice G polynômiale de degré s .

Suivant une idée de G.Jacob, on cherche à identifier les coefficients de la fonction d'observation h grâce à un jeu d'entrées/sorties et de leurs dérivées à l'instant $t = 0$.

Soit

$$\xi = (\xi_1, \dots, \xi_l, \dots, \xi_r)$$

lorsque l balaie la liste des mots de Lyndon de longueur $\leq s$.

Alors,

$$h = \sum_{|\alpha| \leq s} h_\alpha \xi^\alpha$$

pour

$$\begin{cases} b &= [l_1]^{\alpha_1} [l_2]^{\alpha_2} \dots [l_k]^{\alpha_k} \\ \xi^\alpha &= \frac{1}{\alpha_1! \dots \alpha_k!} \xi_{l_1}^{\alpha_1} \dots \xi_{l_k}^{\alpha_k} \\ h_\alpha &= \langle G | b \rangle \end{cases}$$

La méthode consiste à identifier les coefficients h_α de h par des informations de nature différentielle, provenant des correspondances entrées/sorties.

► Expression des dérivées successives de h

Pour présenter la méthode, nous nous limitons à 2 entrées a_1 et a_2 et à $s = 2$. Avec les notations précédentes, on a :

- expression de h

$$h = h(\xi_1, \xi_2, \xi_{12})$$

que l'on notera :

$$h = h(\xi_1, \xi_2, \xi_3)$$

Alors, on peut calculer ses dérivées successives :

– expression de $h^{(1)}$

$$h^{(1)} = \sum_{1 \leq i \leq 3} \frac{\partial h}{\partial \xi_i} \frac{d\xi_i}{dt}$$

Par ailleurs, on a $\frac{d\xi_i}{dt} = a_i$ pour $i = 1, 2$ et on pose

$$a_3 = \frac{d\xi_3}{dt} = \frac{d\xi_{12}}{dt} = \xi_1 a_2$$

Donc

$$h^{(1)} = \sum_{1 \leq i \leq 3} \frac{\partial h}{\partial \xi_i} a_i$$

– expression de $h^{(2)}$

$$h^{(2)} = \sum_{1 \leq i \leq 3} \frac{\partial h}{\partial \xi_i} a_i^{(1)} + \sum_{1 \leq i \leq 3} \frac{\partial^2 h}{\partial \xi_i^2} a_i^2 + 2 \sum_{i \neq j} \frac{\partial^2 h}{\partial \xi_i \partial \xi_j} a_i a_j$$

Par ailleurs, on a $\xi_3^{(2)} = a_3^{(1)} = a_2 a_1 + \xi_1 a_2^{(1)}$

– expression de $h^{(3)}$

$$\begin{aligned} h^{(3)} = & \sum_{1 \leq i \leq 3} \frac{\partial h}{\partial \xi_i} a_i^{(2)} + 3 \sum_{i \neq j} \frac{\partial^2 h}{\partial \xi_i \partial \xi_j} a_i^{(1)} a_j \\ & 3 \sum_{i \neq j} \frac{\partial^3 h}{\partial \xi_i^2 \partial \xi_j} a_i^2 a_j + \sum_{1 \leq i \leq 3} \frac{\partial^3 h}{\partial \xi_i^3} a_i^3 \\ & + 6 a_1 a_2 a_3 \frac{\partial^3 h}{\partial \xi_1 \partial \xi_2 \partial \xi_3} \end{aligned}$$

Par ailleurs, on a $a_3^{(2)} = a_1^{(1)} a_2 + 2 a_2^{(1)} a_1 + \xi_1 a_2^{(2)}$

Remarque:

Le calcul des dérivées successives de h s'exprime de manière générale à l'aide des dérivées partielles de h , des ξ_{l_i} où l_i est un mot de Lyndon de longueur $\leq s$, des entrées et de leurs dérivées. En effet

1. La formule de Dederick [2] généralisant à deux fonctions la formule de Faà di Bruno relative aux dérivées successives d'une fonction composée, permet d'obtenir une formule générale des dérivées de h dans le cas où h est fonction de ξ_1 et de ξ_2 . Cette formule déjà compliquée a naturellement une généralisation à plus de 2 fonctions.
2. La formule d'intégration par parties des intégrales itérées [7] et le théorème de Poincaré-Birkoff-Witt, permettent d'exprimer les dérivées successives des ξ_l en fonction des ξ_{l_i} avec l_i mot de Lyndon.

► **Evaluation des dérivées successives de h à l'instant initial**

A l'instant initial, on a :

$$\xi_1 = \xi_2 = 0, \quad \xi_1^{(1)} = a_1(0), \quad \xi_2^{(1)} = a_2(0)$$

donc

$$\xi_{12}^1 = \xi_3^1 = 0, \quad \xi_{12}^2 = \xi_3^2 = a_1(0)a_2(0)$$

et

$$\xi_{12}^3 = \xi_3^3 = a_1^{(1)}(0)a_2(0) + 2a_2^{(1)}(0)a_1(0)$$

Donc, les dérivées successives de h en $t = 0$ deviennent, en notant

$$h_{i,j,k} = \frac{\partial^{i+j+k} h}{\partial \xi_1^i \partial \xi_2^j \partial \xi_3^k}$$

$$\begin{aligned} h &= h(\xi_1, \xi_2, \xi_3) \\ h^{(1)} &= a_1 h_{100} + a_2 h_{010} \\ h^{(2)} &= a_1^{(1)} h_{100} + a_2^{(1)} h_{010} + a_2 a_1 h_{001} \\ &\quad + a_1^2 h_{200} + a_2^2 h_{020} + 2a_1 a_2 h_{110} \\ h^{(3)} &= a_1^{(2)} h_{100} + a_2^{(2)} h_{010} + (2a_2^{(1)} a_1 + a_2 a_1^{(1)}) h_{001} \\ &\quad + 3a_1^{(1)} a_1 h_{200} + 3a_2^{(1)} a_2 h_{020} \\ &\quad + 3(a_1^{(1)} a_2 + a_2^{(1)} a_1) h_{110} + 3a_1^2 a_2 h_{101} + 3a_1 a_2^2 h_{011} \end{aligned}$$

Et, en se limitant aux dérivées partielles d'ordre ≤ 2 :

$$\begin{aligned} h^{(4)} &= a_1^{(3)} h_{100} + a_2^{(3)} h_{010} + (a_1(2)a_2 + 3a_1(1)a_2(1) + 3a_1 a_2(2)) h_{001} \\ &\quad + 4a_1^{(2)} a_1 h_{200} + 4a_2^{(2)} a_2 h_{020} + 4a_1^{(2)} a_2 h_{110} + 4a_2^{(2)} a_1 h_{110} \\ &\quad + 4(a_1^{(1)} a_2 + 2a_1 a_2^{(1)}) a_1 h_{101} + 4(a_1^{(1)} a_2 + 2a_1 a_2^{(1)}) a_2 h_{011} \\ &\quad + 3a_1^{(1)2} h_{200} + 3a_2^{(1)2} h_{020} + 3(a_1 a_2)^2 h_{002} \\ &\quad + 6a_1^{(1)} a_2^{(1)} h_{110} + 6a_1^{(1)} a_1 a_2 h_{101} + 6a_2^{(1)} a_1 a_2 h_{011} \end{aligned}$$

► **Identification des coefficients h_α de h**

Rappelons que h s'écrit:

$$\begin{aligned} h &= h_0 + h_{100}\xi_1 + h_{100}\xi_2 + h_{100}\xi_3 \\ &\quad + 1/2(h_{200}\xi_1^2 + h_{020}\xi_2^2 + h_{002}\xi_3^2) \\ &\quad + h_{110}\xi_1\xi_2 + h_{101}\xi_1\xi_3 + h_{011}\xi_2\xi_3 \end{aligned}$$

Nous allons identifier les $h_{i,j,k}$ figurant dans la formule précédente par des choix des entrées et de leurs dérivées.

- $h(0)$ permet d'identifier le terme constant de h

- $h^{(1)}(0)$ permet d'identifier h_{100} et h_{010} pour des choix convenables de $a_1(0)$ et de $a_2(0)$.
- $h^{(2)}(0)$ permet les identifications suivantes :
 - 1) Pour $a_1^{(1)}(0) = a_2^{(1)}(0) = a_1(0) = 0$, on identifie h_{020}
 - 2) Pour $a_1^{(1)}(0) = a_2^{(1)}(0) = a_2(0) = 0$, on identifie h_{200}
 - 3) Pour $a_1(0)a_2(0) \neq 0$, on identifie $2h_{110} + h_{001}$
- $h^{(3)}(0)$ permet les identifications suivantes :
 - 1) Pour $a_1^{(1)}(0) = a_2^{(1)}(0) = 0$ et $a_1(0) \neq a_2(0) \neq 0$, on identifie h_{011} et h_{101}
 - 2) Pour $a_1^{(1)}(0) \neq 0$, $a_2(0) \neq 0$, on identifie h_{001} et h_{110}
- $h^{(4)}(0)$ permet d'identifier h_{002} pour $a_1(0) \neq 0$, $a_2(0) \neq 0$.

Tous les coefficients figurant dans l'expression de h (à l'ordre 2), ont ainsi pu être déterminés selon cette méthode. Cette méthode est effective dans le cas où la série génératrice est polynômiale.

Méthode :

1. Obtention des dérivées successives de h par la formule de Dederick généralisée.
2. Calcul des entrées fictives a_i et de leurs dérivées pour $i \geq m$, en $t = 0$, m étant le nombre d'entrées du système.
3. Substitution dans les formules obtenues à l'étape 1 des entrées fictives et de leurs dérivées par les expressions trouvées dans l'étape 2. On obtient alors un système d'équations linéaires dont les indéterminées sont les dérivées partielles multiples de h .
4. Identification des coefficients des multidérivées des entrées.
5. Séparation des dérivées de h figurant comme coefficient de la même multidérivée des entrées. En allant assez loin dans les dérivations successives de h , on isole les indéterminées non encore identifiées, sans en introduire de nouvelles.

7.6 Méthode non commutative: Identification par la série génératrice

Le problème se pose en ces termes :

Existe-t-il un algorithme qui permet de **déterminer explicitement la série génératrice jusqu'à un ordre donné k , quand on connaît les développements de Taylor (en $t = 0$) des entrées du système et les développements de Taylor (en $t = 0$) des sorties associées?**

7.6.1 Une solution par les preuves sur l'indiscernabilité?

Avant d'aborder le problème de l'Identification lui même, nous rappelons [6] que trois preuves (au moins) de la proposition suivante existent :

Proposition 7.6.1 *Toute série indiscernable de 0 (i.e. correspondant à une fonctionnelle entrée/sortie nulle) est identiquement nulle.*

- C.Reutenauer propose une preuve grâce à une technique d'entrées concaténées, en exploitant une propriété d'analyticit  de la sortie $y(t)$.
 - M.Fliess montre, par des passages   la limite, que les r siduelles de la s rie G par n'importe quel mot de Z^* sont identiquement nulles.
 - E.D.Sontag et Y.Wang proposent, par des choix d'entr es continues par morceaux, une preuve bas e sur des calculs de d riv es de la fonctionnelle entr e/sortie.
- Ces preuves n cessitent la connaissance de $y(t)$ dans un voisinage de 0. Elles font intervenir des propri t s topologiques et la th orie des fonctions analytiques. Elles ne fournissent pas une m thode effective pour r soudre le probl me de l'identification.

Dans les chapitres suivants, nous proposons deux  tapes successives pour r soudre le probl me de l'identification alg brique par le calcul des coefficients de la s rie g n ratrice.

Bibliographie

- [1] Chen T., Chen H. *Approximations of continuous functionals by neural networks with applications to dynamic systems*, IEEE Trans. on Neural Networks, Vol.4, n° 6, pp. 910-918, 1993.
- [2] L.S. Dederick, *Successive derivatives of a function of several functions*, Ann. of Math. 27, pp. 385-394, 1925.
- [3] Denis-Vidal L., Joly-Blanchard G., *Identifiability of uncontrolled nonlinear systems: a comparison of methods*, E.J.C., vol. 2, 1998.
- [4] Guyon C., *Calcul Symbolique pour la planification de Trajectoire des Systèmes dynamiques nilpotents*, Thèse Lille I, 1995.
- [5] Hespel C., *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] Hespel C., Jacob G., *First steps towards exact algebraic identification*, Discrete Math., vol. 180, pp. 211-219, 1998.
- [7] Hoang Ngoc Minh, *Contribution au développement d'outils informatiques pour résoudre des problèmes d'automatique non linéaire*, Thèse Lille I, 1990.
- [8] Juditsky A., Hjalmarsson H., Benveniste A., Delyon B., Ljung L., Sjöberg J., Zhang Q., *Nonlinear Black-box Models in System Identification: Mathematical Foundations*, Automatica vol. 31, n° 12, pp. 1725-1750, 1995.
- [9] Landau I.D., *Identification et Commande de Systèmes*, Hermes, 1993.
- [10] Ljung L., Glad T., *On global identifiability for arbitrary model parametrizations*, Automatica vol. 30, n° 2, pp. 265-276, 1994.
- [11] Luzardo J.A., Chassiakos A., Rumbos A., *Representations of First Order Dynamical Systems Using Neural Networks*, Australia ICNN'95.
- [12] Margaglio E., *Identificación de sistemas industriales multivariables usando redes neurales*, Thèse, Universidad Simón Bolívar, Caracas, 1994.

- [13] Lamanna R., Margaglio E., *Neural networks for estimation of product properties in a pilot plant distillation column*, Proceedings of the IASTED international conference, Applied Modelling, Simulation and Optimization, Cancún, Mexico, june 15-17, 1995.
- [14] Margaglio E., Glorennec P.Y., Lamanna R., *Identification of a Binary Distillation Column with Recurrent Neural Networks*, EUFIT'96, pp. 1623-1627, September 2-5, Aachen, Germany, 1996.
- [15] Narendra, Parthasarathy K. and K., *Identification and acontrol of Dynamical Systems Using Neural Networks*, IEEE Transactions on Neural Networks, 1 (1), pp. 4-27, 1990.
- [16] Ollivier F., *Le problème de l'identifiabilité structurelle globale : approche théorique, méthodes effectives et bornes de complexité*, Thèse Ecole Polytechnique, 1990.
- [17] Olurotimi O., *Recurrent neural network training with feedforward complexity*, IEEE Trans. on Neural Networks, Vol.5, n° 2, pp. 910-918, 1994.
- [18] Oussous N., *Etude et traitement des séries formelles non commutatives pour la représentation minimale des systèmes dynamiques non linéaires*, Thèse Lille I, 1988.
- [19] Oussous N.E., Petitot M., *Implantation de la réalisation minimale des systèmes dynamiques en Scratchpad*, ACC Conference, Paris, CNRS, 1991.
- [20] Petitot M., *Algèbre non commutative en SCRATCHPAD : application au problème de la réalisation minimale analytique*, Thèse Lille I, 1992.
- [21] Sjoberg J., Zhang Q., Ljung L., Benveniste A., Delyon B., Glorennec P.Y., Hjalmarsson H., Juditsky A., *Nonlinear Black-box Modeling in System Identification: a Unified Overview*, Automatica vol. 31, n° 12, pp. 1691-1724, 1995.
- [22] Walter E., *Identifiability of State Space Models*, Lect. notes in Biomath., n° 46, Springer, 1982.
- [23] Distefano, Lamnabhi-Lagarrigue, Lecourtier, Raksanyi, Vajda, Walter, etc ..., *Identifiability of parametric models*, E.Walter editeur, Pergamon Press, Oxford, 1987.
- [24] Walter E., Pronzato L., *Identification of Parametric Models*, Springer-Verlag, 1997.

Chapitre 8

Une première étape vers l'Identification algébrique

8.1 Introduction

Dans ce chapitre, nous présentons la première étape vers l'identification algébrique. Pour cela, nous calculons les dérivées itérées de la sortie du système (Σ). Elles apparaissent comme des combinaisons linéaires des multidérivées des entrées.

Nous proposons ensuite un algorithme d'identification des coefficients des multidérivées des entrées. Cette identification relève de la théorie du contrôle: elle est basée sur des mesures des valeurs d'entrée/sortie (et de leurs dérivées) et consiste à identifier des coefficients de la représentation algébrico-différentielle du système.

Après quelques rappels de Combinatoire, nous présentons une interprétation combinatoire de certaines entités apparaissant dans les calculs. Cet éclairage par la Combinatoire préparera la seconde étape vers l'identification algébrique développée au chapitre suivant.

8.2 Obtention d'un système d'équations linéaires à partir des dérivées itérées de la sortie

Les détails de ce paragraphe se trouvent dans [2, 3].

8.2.1 Dérivées des séries de Chen

Reprenons les notations du chapitre précédent. \mathcal{C}_a étant la série de Chen de l'entrée a , et pour $\mathcal{L}_a = \sum_{0 \leq i \leq m} a_i z_i$, on sait que le lemme du chapitre précédent établit le

résultat suivant:

$$\frac{d}{dt}C_a = C_a \cdot \mathcal{L}_a$$

soit

$$y_G^{(1)}(t) = \langle \mathcal{L}_a \triangleleft G | C_a(t) \rangle$$

Il vient alors

$$y_G^{(i)}(t) = \langle G | C_a^{(i)}(t) \rangle \quad \forall i \in \mathbb{N},$$

Il reste alors à calculer les dérivées de Chen : Puisque $\frac{\partial C_a}{\partial t} = C_a \cdot \mathcal{L}_a$, nous obtenons

$$\text{avec } \left. \begin{array}{l} C_a^{(i)} = C_a A_i \\ A_1 = \mathcal{L}_a \\ A_{i+1} = \mathcal{L}_a A_i + D_t A_i \end{array} \right\} (R)$$

où D_t est l'opérateur de dérivation usuel par rapport au temps.

Puisque $C_a(0) = 1$ et $C_a^{(i)}(0) = A_i(0)$ alors, en notant $y^{(i)}$ au lieu de $y_G^{(i)}$,

$$\begin{aligned} y^{(i)}(0) &= \sum_{w \in Z^*} \langle G | w \rangle \langle C_a^{(i)}(0) | w \rangle \\ &= \sum_{w \in Z^*} \langle G | w \rangle \langle A_i(0) | w \rangle \\ &= \langle G | A_i(0) \rangle \end{aligned}$$

où A_i est donné par la formule (R).

8.2.2 Calcul explicite des A_i

Il reste à calculer explicitement les A_i :

Soit $\rho = (\rho_1, \rho_2, \dots, \rho_p)$ un multi-indice de dérivation. On définit son degré et son poids par

$$\deg(\rho) = p, \quad \text{poids}(\rho) = \sum_{j=1}^p (1 + \rho_j) = \deg(\rho) + \sum_{j=1}^p \rho_j.$$

On pose

$$A^{(\rho_k)} = \sum_i a_i^{(\rho_k)} z_i \quad \text{and} \quad A^{[\rho]} = A^{(\rho_1)} \dots A^{(\rho_p)}.$$

Autrement dit:

$$A^{[\rho]} = \sum_{i_1, i_2, \dots, i_p} a_{i_1}^{(\rho_1)} a_{i_2}^{(\rho_2)} \dots a_{i_p}^{(\rho_p)} z_{i_1} z_{i_2} \dots z_{i_p}.$$

Donc:

$$A_i = \sum_{\text{poids}(\rho)=i} \alpha_\rho A^{[\rho]}$$

Notons $\mathcal{A} = \sum_{i \in \mathcal{N}} A_i$. Alors \mathcal{A} satisfait l'équation générique (G):

$$\mathcal{A} = 1 + \mathcal{L}_a \mathcal{A} + D_t \mathcal{A} \quad (G)$$

De plus, on a l'égalité:

$$\mathcal{A} = \sum_{\rho} \alpha_{\rho} A^{[\rho]}$$

Par une méthode de nature combinatoire, on obtient :

$$\alpha_{\rho} = \prod_{i=1}^p \binom{\sum_{j=1}^i \rho_j + i - 1}{\rho_i} = \binom{\rho_1}{\rho_1} \binom{\rho_1 + \rho_2 + 1}{\rho_2} \dots \binom{\rho_1 + \dots + \rho_k + k - 1}{\rho_k} \quad (E)$$

Il vient alors l'expression des dérivées de la sortie

$$\begin{aligned} y^{(n)}(t) &= \sum_{\text{poids}(\rho)=n} \alpha(\rho) \sum_{i_j} a_{i_1}^{(\rho_1)}(t) \dots a_{i_k}^{(\rho_k)}(t) \langle z_{i_1} \dots z_{i_k} \triangleleft G \parallel \mathcal{C}_a(t) \rangle \\ &= \sum_{\text{poids}(\rho)=n} \alpha(\rho) \sum_{i_j} a_{i_1}^{(\rho_1)}(t) \dots a_{i_k}^{(\rho_k)}(t) \langle G \parallel \mathcal{C}_a(t) z_{i_1} \dots z_{i_k} \rangle \end{aligned}$$

Pour un *système analytique* (Σ), cette formule peut s'interpréter de deux manières.

- En initialisant le système (Σ) au temps t (et en posant $\mathcal{C}_a(t) = 0$), on obtient l'*expression locale* de $y^{(n)}(t)$ en l'état $q(t)$:

$$y^{(n)}(t) = \sum \sum \alpha(\rho) a_{i_1}^{(\rho_1)}(t) \dots a_{i_k}^{(\rho_k)}(t) (g_{i_1} \dots g_{i_k} \circ h)_{|_{q(t)}}$$

(C'est la formule de Lamnabhi-Lagarrigue et Crouch [6]).

- En initialisant le système (Σ) au temps 0, on obtient l'*expression globale* de $y^{(n)}(t)$ en l'état initial $q(0)$:

$$y^{(n)}(t) = \sum \sum \alpha(\rho) a_{i_1}^{(\rho_1)} \dots a_{i_k}^{(\rho_k)} y_{i_1 i_2 \dots i_k}(t)$$

où $y_{i_1 i_2 \dots i_k}$ est la sortie du système (Σ), initialisé en $q(0)$, pour la fonction d'observation $g_{i_1} g_{i_2} \dots g_{i_k} \circ h$.

Pour $t = 0$, puisque $\mathcal{C}_a(0) = 1$, on a la formule :

$$\left. \begin{aligned} y^{(n)}(0) &= \sum_{k=0}^n \sum_{i_j} \langle G | z_{i_1} \dots z_{i_k} \rangle \\ &\quad \sum_{\rho_1 + \dots + \rho_k = n - k} \binom{\rho_1}{\rho_1} \binom{\rho_1 + \rho_2 + 1}{\rho_2} \dots \\ &\quad \dots \binom{\rho_1 + \dots + \rho_k + k - 1}{\rho_k} a_{i_1}^{(\rho_1)}(0) \dots a_{i_k}^{(\rho_k)}(0) \end{aligned} \right\} \quad (F_n)$$

Remarque 8.2.1 En généralisant la formule de Faà di Bruno [1] exprimant les dérivées d'une fonction analytique composée, aux fonctionnelles causales, nous prouvons que la formule (F_n) n'est autre que la formule de Faà di Bruno.

8.2.3 Expression des A_n en les multidérivées des entrées

Toute multidérivée $a^{(\rho)}$ peut être considérée comme un monôme différentiel en les variables a_j . On notera ces monômes par a^μ . Ainsi, on a:

$$A_n = \sum_{\text{poids}(\rho)=n} \alpha(\rho) A^{[\rho]} = \sum_{\mu} g_{\mu} a^{\mu},$$

et il reste à calculer les polynômes non commutatifs g_{μ} .

► Calcul sans dérive

– Cas général: Les équations de récurrence

L'équation générique (G) se traduit par:

$$\sum_{\mu} a^{\mu} g_{\mu} = 1 + \sum_{j=1}^m \sum_{\sigma} a_j z_j a^{\sigma} l_{\sigma} + \sum_{\nu} D_t(a^{\nu}) g_{\nu}$$

En introduisant les notations

$$\mu \triangleright a_j = \begin{cases} \sigma & \text{si } a_j a^{\sigma} = a^{\mu} \\ 0 & \text{sinon} \end{cases}$$

et

$$D_t(a^{\nu}) = \sum_{\tau} \langle \nu | \tau \rangle a^{\tau}, \quad \langle \nu | \tau \rangle \in \mathbb{N}$$

On obtient, pour une multiplicité μ non triviale:

$$g_{\mu} = \sum_{j=1}^m z_j g_{\mu \triangleright a_j} + \sum_{\nu} \langle \nu | \mu \rangle g_{\nu}$$

avec $g_{\epsilon} = 1$

– Cas d'une entrée sans dérive:

Dans ce cas,

$$\langle A_n | z_1^{\rho} \rangle = \sum_{\text{poids}(\rho)=n} \alpha(\rho) a_1^{(\rho_1)} a_1^{(\rho_2)} \dots a_1^{(\rho_p)},$$

et

$$a_1^{(\rho_1)} a_1^{(\rho_2)} \dots a_1^{(\rho_p)} = a_1^{\mu_1} \left(a_1^{(1)} \right)^{\mu_2} \dots \left(a_1^{(k-1)} \right)^{\mu_k} = a_1^{\mu}.$$

On appelle alors multiplicité différentielle de a_1^μ , la suite finie d'entiers:

$$\mu = (\mu_1, \mu_2, \dots, \mu_k).$$

On définit le degré et le poids de μ par:

$$p = \deg(\rho) = \deg(\mu) = \sum_{i=1}^k \mu_i$$

$$\sum_{i=1}^k (1 + \rho_i) = \text{poids}(\rho) = \text{poids}(\mu) = \sum_{i=1}^k i\mu_i$$

En appelant développement de μ la séquence:

$$\text{dev}(\mu) = (\underbrace{0, \dots, 0}_{\mu_1 \text{ fois}}, \underbrace{1, \dots, 1}_{\mu_2 \text{ fois}}, \dots, \underbrace{k-1, \dots, k-1}_{\mu_k \text{ fois}}),$$

on remarque que $a^{(\rho)} = a^\mu$ ssi ρ est une permutation de la séquence $\text{dev}(\mu)$, ce que l'on note: $\rho \in \mathcal{S}(\text{dev}(\mu))$.

On obtient alors:

$$\langle g_\mu | z_1^p \rangle = \sum_{\rho \in \mathcal{S}(\text{dev}(\mu))} \alpha(\rho).$$

► Cas d'une entrée avec dérive

– Les équations de récurrence

On devrait écrire $a^\mu = a_0^p a^\nu$. Dans ce cas, on note

$$g_\mu = g_{/p}^\nu$$

On obtient alors

$$A_n = \sum_{p+\text{poids}(\nu)=n} a_0^p a^\nu g_{/p}^\nu$$

Et donc puisque a_0 est identique à 1:

$$A_n = \sum_{p+\text{poids}(\nu)=n} a^\nu g_{/p}^\nu$$

On appelle alors profondeur de $g_{/p}^\nu$, l'entier p .

Le polynôme $g_{/p}^\nu$ est homogène de degré p en la lettre z_0 et homogène de degré $\text{deg}(\nu)$ en la lettre z_1 .

Les équations récurrentes vérifiées par les polynômes $g_{/p}^\nu$ deviennent alors:

$$g_{/p}^\nu = z_0 g_{/p-1}^\nu + z_1 g_{/p}^{\nu \circ a} + \sum_{\sigma} \langle \sigma | \nu \rangle g_{/p}^\sigma$$

– **expression des multidérivées**

On a alors $a_0(t) \equiv 1$. On obtient alors la formule:

$$a_{i_1}^{(\rho_1)} a_{i_2}^{(\rho_2)} \dots a_{i_p}^{(\rho_p)} = a_0^{\mu_1} a_1^{\nu_1} \left(a_1^{(1)}\right)^{\nu_2} \dots \left(a_1^{(k-1)}\right)^{\nu_k} = a^{\mu;\nu} = a_1^\nu.$$

On a les relations:

$$\begin{cases} a^{\mu;\nu} = 0 & \text{excepté pour } \mu = (j), j \in \mathbb{N} \\ a^{(j);\nu} = a_1^\nu. \end{cases}$$

avec

$$\text{dev}((j)) = \underbrace{(0, \dots, 0)}_{j \text{ fois}}.$$

Alors, le coefficient d'un mot w dans $l_{(j);\nu}$ est nul sauf dans le cas $|w|_{z_0}=j$ et $|w|_{z_1}=\text{deg}(\mu)$.

8.3 L'identification algébrique: Première étape

Le détail de ce paragraphe figure dans [3] et un algorithme en MAPLE est développé dans [5]. Nous résumons ici les idées de base d'un algorithme permettant d'identifier les coefficients des multidérivées des entrées a^μ , en résolvant le système d'équations:

$$y^{(k)}(0) = \sum_{\mu} a^\mu \langle G | g_\mu \rangle$$

afin de calculer les coefficients inconnus $\langle G | g_\mu \rangle$.

8.3.1 Cas d'une entrée sans dérive

Nous notons l'entrée par $a(t)$ et nous la fixons polynômiale :

$$a(t) = \sum_{i=0}^k \frac{c_i}{i!} t^i.$$

Alors $A^{(0)}(0) = a(0)z_1 = c_0.z_1$ et $A^{(i)}(0) = c_i.z_1$, et

$$a^{(\rho_1)} \dots a^{(\rho_p)} = a^{\mu_1} \dots (a^{(k-1)})^{\mu_k} = a^\mu$$

nous obtenons :

$$y^{(n)}(0) = \sum_{\text{poids}=n} c_0^{\mu_1} \dots c_{k-1}^{\mu_k} \langle G | z_1^{\mu_1 + \dots + \mu_k} \rangle \lambda_\mu$$

Nous avons donc à résoudre un système triangulaire infini d'équations linéaires :

$$\left\{ \begin{array}{l} y(0) = \langle G|\varepsilon \rangle \\ y^{(1)}(0) = c_0 \langle G|z_1 \rangle \\ y^{(2)}(0) = a_0^{(1)} \langle G|z_1 \rangle + c_0^2 \langle G|z_1^2 \rangle \\ \vdots \\ y^{(n)}(0) = \sum_{\substack{\text{poids}(\mu) = n \\ i = \text{deg}(\mu) < n}} a^\mu(0) \lambda_\mu \langle G|z_1^i \rangle + c_0^n \langle G|z_1^n \rangle \\ \vdots \end{array} \right.$$

Dans ce cas, le calcul des coefficients inconnus $\langle G|z_1^k \rangle$ est évident grâce à un choix de l'entrée satisfaisant :

$$a(0) = c_0 \neq 0,$$

si les dérivées des sorties correspondantes $y^{(p)}(0)$ sont connues pour $0 \leq p \leq k$.

8.3.2 Cas d'une entrée avec dérive

Nous proposons un algorithme permettant d'identifier les coefficients des multi-dérivées des entrées a^p , en résolvant le système d'équations

$$y^{(k)}(0) = \sum_p a^p \langle G|g_p \rangle$$

afin de calculer les coefficients inconnus $\langle G|g_p \rangle$.

Nous choisissons encore l'entrée polynômiale :

$$a_1(t) = \sum_{i=0}^k \frac{c_i}{i!} t^i.$$

Mais dans ce cas, la série génératrice est définie sur l'alphabet $Z = \{z_0, z_1\}$ et

$$\mathcal{L}_a(0) = c_0 \cdot z_1 + z_0, \quad \mathcal{L}_a^{(i)}(0) = c_i \cdot z_1.$$

Nous obtenons alors un système d'équations linéaires qui doit être satisfait pour tout choix des coefficients c_i .

$$y^{(n)}(0) = \sum_{\text{poids}(\mu) \leq n} c_0^{\mu_1} \cdots c_{k-1}^{\mu_{k-1}} \sum_w \langle G|g_\mu \rangle \quad (\text{FD})_n$$

$$\begin{array}{l} |w|_{z_1} = \mu_k + \mu_{k-1} + \cdots + \mu_1 \\ |w|_{z_0} = n - |w|_{z_1} - (k-1)\mu_k - \cdots - \mu_2 \end{array}$$

Plutôt que de résoudre le système complet $(FD)_{n \in \mathbb{N}}$, nous traitons une seule équation $(\text{FD})_n$ et nous montrons que nous pouvons identifier chaque coefficient des multidérivées apparaissant dans cette équation, par des choix convenables d'entrées.

L'idée consiste à définir un ordre sur les multidérivées (indexant les colonnes), et à faire un choix judicieux des valeurs associées des entrées et de leurs dérivées (indexant les lignes), de manière à obtenir un déterminant susceptible de s'écrire sous forme de produits de déterminants de Vandermonde, par des transformations de Gauss. Cette méthode est développée dans [3] et un algorithme effectif en MAPLE est donné dans [5].

Nous montrons de plus que cet algorithme d'identification des multidérivées des entrées s'adapte sans peine au cas de plusieurs entrées.

8.4 Interprétation combinatoire

Des combinatoristes, auxquels nous avons présenté cette première étape d'identification, ont reconnu comme des objets combinatoires certaines entités figurant dans les calculs précédents. Cette remarque, confirmée par les problèmes combinatoires rencontrés au chapitre suivant (deuxième étape vers l'identification algébrique), met en évidence la constatation que le problème de l'identification algébrique peut s'interpréter comme un problème de Combinatoire.

Nous présentons ici des définitions permettant de retrouver \mathcal{A} et g_μ en tant qu'objets combinatoires.

8.4.1 Les partitions colorées

Nous introduisons ici des définitions afin de faire le lien avec la première étape du problème de l'identification.

► Les partitions d'ensembles

On définit une partition d'un ensemble ordonné $\{1, 2, \dots, n\}$ comme une séquence d'entiers $p = p_1 p_2 \dots p_n$ telle que $p_1 = 1$ et $1 \leq p_i \leq \max\{p_1, \dots, p_{i-1}\} + 1$ pour $1 < i \leq n$.

Une telle séquence p détermine une partition contenant $\max_{1 \leq i \leq n} p_i$ blocs, telle que le j ème bloc contienne les $i \in \{1, 2, \dots, n\}$ vérifiant $p_i = j$. Ces blocs sont ordonnés par leurs plus petits éléments.

Soit $P_{n,k}$ les partitions de $\{1, 2, \dots, n\}$ à k blocs.

Soit $P_n = \bigcup_{1 \leq k \leq n} P_{n,k}$ les partitions de $\{1, 2, \dots, n\}$.

Soit $P_{.,k} = \bigcup_{n \geq k} P_{n,k}$, les partitions à k blocs.

Et enfin, soit $\bar{P} = \bigcup_{n \geq 0} P_n = \bigcup_{k \geq 0} P_{.,k}$.

Par convention, $P_{0,0} = \{\epsilon\}$ et $P_{n,0} = \emptyset$ pour $n \geq 1$, où ϵ désigne la partition vide.

► **Les partitions de nombres**

Une partition de nombre (ou “multiplicité”) est un élément du monoïde abélien libre engendré par les entiers positifs \mathbb{N}^+ , i.e., une séquence $\mu = (\mu_1, \mu_2, \mu_3, \dots)$ (aussi écrite $1^{\mu_1} 2^{\mu_2} 3^{\mu_3} \dots$) d’entiers non négatifs, où presque tous les μ_i sont nuls.

La *taille* de la multiplicité μ est $\sum_k k \cdot \mu_k$ et son *poids* est $\sum_k \mu_k$.

Les partitions de nombre forment un treillis, appelé *treillis de Young*. Y_n désigne l’ensemble des partitions de nombre de taille n . C’est le niveau n du treillis de Young. $Y_{n,k}$ désigne l’ensemble des partitions de nombre de taille n et de poids k .

Pour $\mu \in Y_n$ soit $\mu \uparrow$ l’ensemble des $\nu \in Y_{n+1}$ qui couvrent μ , et soit $\mu \downarrow$ l’ensemble des $\nu \in Y_{n-1}$ qui sont couverts par μ .

A chaque partition d’ensemble p , on associe la partition de nombre $\mu(p)$ où $\mu_k(p)$ désigne le nombre de blocs de taille k de p . Ainsi, $P_{n,k}$ est envoyé dans $Y_{n,k}$.

► **Les partitions colorées**

Nous considérons un ensemble C de couleurs et des partitions d’ensemble de n , où chaque bloc est coloré d’une des couleurs de C . Il est commode d’associer cette couleur au plus petit élément du bloc. Prenons $B = \{b_1, b_2, \dots\}$ et C comme alphabets. Nous pouvons décrire l’ensemble CP des partitions colorées comme un langage sur l’alphabet $B \cup C$. Ce langage est défini par l’équation

$$CP = \epsilon + CP \cdot C + \sum_{k \geq 1} CP_{.,k} \cdot (b_1 + \dots + b_k)$$

On note $CP_{.,k}$ l’ensemble des (mots désignant les) partitions colorées avec exactement k blocs. Autrement dit, nous prenons les éléments des alphabets B et C comme “constructeurs” i.e.

- Pour $j \in \mathbb{N}^+$, le symbole b_j désigne “on augmente le j ème bloc d’une partition colorée donnée w d’un nouvel élément” (ce qui est légal seulement si w a au moins j blocs);
- Pour $c \in C$, le symbole c signifie “on augmente une partition colorée donnée d’un nouveau bloc de taille 1 et de couleur c ”. Dans l’équation précédente, ϵ désigne la partition colorée vide. Cette équation peut être écrite sous différentes formes équivalentes, par exemple,

$$CP_{n+1} = CP_n \cdot C + \sum_{k \geq 1} CP_{n,k} \cdot (b_1 + \dots + b_k) \quad (n \geq 0)$$

avec la condition initiale $CP_0 = \epsilon$, ou encore

$$CP_{.,k+1} = CP_{.,k} \cdot C + CP_{n,k+1} \cdot (b_1 + \dots + b_k) \quad (k \geq 0)$$

avec la condition initiale $CP_{.,0} = \epsilon$, ou enfin

$$CP_{n+1,k+1} = CP_{n,k} \cdot C + CP_{n,k+1} \cdot (b_1 + \dots + b_k) \quad (n, k \geq 0)$$

avec des conditions initiales évidentes.

► Les multiplicités colorées

Comme pour les partitions, on peut associer à toute partition colorée $w \in CP$ une “multiplicité” ou “statistique” de ses blocs, en tenant compte des couleurs. C’est une application

$$\mu : CP \rightarrow Y^C : w \mapsto \mu(w) = (\mu_c)_{c \in C} = (\mu_{c,j}(w))_{c \in C, j \in \mathbb{N}^+}$$

où Y^C est l’ensemble des applications de C dans le treillis des partitions de nombres (multiplicités) Y . On écrira les éléments μ de Y^C sous la forme $(\mu_{c,j})_{c \in C, j \in \mathbb{N}^+}$, où on note $\mu_{c,j}$ à la place de $\mu_c(j)$. La signification de $\mu_{c,j}(w) = k$ est que la partition colorée w a exactement k blocs de taille j et de couleur c ($c \in C, j \in \mathbb{N}^+$). L’ensemble Y^C a une structure de produit de treillis. Ainsi, les notations $\mu \uparrow$ et $\mu \downarrow$ ont la même signification que précédemment. En particulier, pour $\mu \in CP, c \in C$ et $k \in \mathbb{N}^+$ soit $\nu = \mu \uparrow_{c,k}$ défini comme suit:

$$\nu_{c'} = \mu_c \quad \forall c' \neq c$$

$$\nu_{c,j} = \begin{cases} \mu_{c,k} - 1 & \text{si } j = k \\ \mu_{c,k+1} + 1 & \text{si } j = k + 1 \\ \mu_{c,k} & \text{sinon} \end{cases}$$

Cette définition n’a de sens que si $\mu_{c,k} > 0$; Si ce n’est pas le cas, alors $\mu \uparrow_{c,k}$ n’est pas défini. De plus, on définit $\nu = \mu \uparrow_{c,0}$ par

$$\nu_{c'} = \mu_c \quad \forall c' \neq c$$

$$\nu_{c,j} = \begin{cases} \mu_{c,1} + 1 & \text{si } j = 1 \\ \mu_{c,j} & \text{sinon} \end{cases}$$

Une manière commode de traiter ces multiplicités est d’introduire un ensemble infini $\{[c, j]; c \in C, j \in \mathbb{N}^+\}$ et de poser

$$\mu(w) = \prod_{c \in C} \prod_{j \in \mathbb{N}^+} [c, j]^{\mu_{c,j}(w)}$$

Ainsi la construction récursive des partitions colorées est donnée par les propriétés

$$\mu(w \cdot c) = [c, 1] \cdot \mu(w) \quad (c \in C)$$

et

$$\mu(w \cdot b_j) = \frac{[c, k + 1]}{[c, k]} \cdot \mu(w)$$

si le j ème bloc de w a la taille k et la couleur c .

En sommant sur toutes les possibilités d’extension de partitions colorées, il vient:

$$\mu(w \cdot C) = \left(\sum_{c \in C} [c, 1] \right) \cdot \mu(w)$$

$$\mu(w \cdot B) = \left(\sum_{c \in C, k \in \mathbb{N}^+} [c, k + 1] \frac{\partial}{\partial [c, k]} \right) \cdot \mu(w)$$

La dernière équation pouvant encore s’écrire:

$$\mu(w \cdot B) = \sum_{c,k} \mu_{c,k}(w) \mu(w) \uparrow_{c,k}$$

► **Les séquences de couleur**

On peut aussi associer à une partition colorée $w \in CP$, son image w_C par projection sur le sous-alphabet C de couleurs, i.e. le mot indiquant la séquence de couleur des blocs. Alors,

$$(w \cdot c)_C = w_C \cdot c \quad (c \in C)$$

ainsi que la condition initiale $\epsilon_C = \epsilon$

► **Les statistiques simultanées**

Il est intéressant de considérer l'application simultanée:

$$CP \rightarrow Y^C \times C^* : w \mapsto (\mu(w), w_C)$$

Dans un anneau de séries formelles à variables commutatives $[c, j]$ ($c \in C, j \in \mathbb{N}^+$), et à variables non commutatives $c \in C$, où tous les $[c, j]$ commutent avec tous les c , on peut considérer l'image de CP , i.e. la série formelle:

$$\mathcal{A} = \sum \{ \mu(w) w_C; w \in CP \}$$

Les images \mathcal{A}_n de CP_n vérifient les équations:

$$\mathcal{A}_{n+1} = \sum_{c \in C} [c, 1] \mathcal{A}_n \cdot c + \sum_{c \in C, k \in \mathbb{N}} [c, k+1] \frac{\partial}{\partial [c, k]} \cdot \mathcal{A}_n$$

avec $\mathcal{A}_0 = \epsilon$.

Ces équations fournissent une méthode récursive pour calculer les \mathcal{A}_n

Par ailleurs, si on fixe la multiplicité et si on recherche toutes les partitions colorées qui y contribuent, il vient:

$$\mathcal{A} = \sum_{\mu \in Y^C} \mu \sum \{ w_C; w \in CP, \mu(w) = \mu \}$$

Notons par g_μ la somme interne, et remarquons que ces séries sur C^* sont en fait des polynômes puisque seuls les mots $w \in CP_n$, apportent une contribution à cette somme si n est la taille de μ (somme des tailles des composantes $\mu_c, c \in C$). La structure récursive des partitions colorées est donnée par les équations:

$$g_\mu = \sum_{c \in C, \uparrow_{c,0} = \mu} g_\nu \cdot c + \sum_{c \in C, j \geq 1, \uparrow_{c,j} = \mu} \nu_{c,j} \cdot g_\nu$$

avec $g_{\mu(\epsilon)} = \epsilon$.

Ces équations fournissent une méthode récursive pour calculer les polynômes g_μ

8.4.2 Algèbre différentielle et partitions colorées

On considère les entrées au temps $t = 0$, comme des lettres différentielles. Alors, les A_n donnés par la formule de récurrence (R) apparaissent comme des polynômes non commutatifs sur Z dont les coefficients sont définis sur les monômes différentiels commutatifs en les a_i (cf [4]).

► **Partitions et multiplicités**

Si $\rho = (\rho_1, \dots, \rho_n)$ est une suite finie d'entiers strictement positifs, appelée multi-*indice*, nous notons ρ^+ la séquence obtenue en réordonnant ρ par ordre croissant: exemple: pour $\rho = (3, 2, 6, 2, 1, 3, 2)$, alors $\rho^+ = 1222336$ et $\mu(\rho) = 1^1 2^3 3^2 6^1$. Sur une seule lettre $a_1 = a$, les monômes différentiels s'écrivent:

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

Alors μ est indexé par la partition suivante:

$$\mu = (i_1^{e_1} i_2^{e_2} \dots i_q^{e_q}), \quad 1 \leq i_1 < i_2 < \dots < i_q$$

Son poids et sa taille sont définis par:

$$poids(\mu) = \sum_k e_k i_k, \quad taille(\mu) = \sum_k e_k$$

Nous définissons une loi de dérivation D sur les partitions pour reproduire l'effet de la dérivation D_t par rapport au temps sur les monômes différentiels:

$$\begin{aligned} D(i_k) &= i_{k+1} \\ D(i_1^{e_1} i_2^{e_2} \dots i_q^{e_q}) &= \sum_{k=1 \dots q} e_k * (i_1^{e_1} i_2^{e_2} \dots i_k^{e_k-1} i_{k+1}^{e_{k+1}+1} \dots i_q^{e_q}) \end{aligned}$$

Ainsi $D(\mu)$ est combinaison linéaire de partitions qui couvrent μ dans le treillis de Young. Cela s'écrit en terme d'algèbre linéaire:

$$D(\mu) = \sum_{\text{partitions } \tau} \langle \mu | \tau \rangle \tau, \quad \langle \mu | \tau \rangle \in \mathbb{N}$$

► **Partitions colorées et multiplicités**

Soient $Z = \{z_1, \dots, z_m\}$, un alphabet fini de couleurs. Nous appelons **multi-*indice* coloré**, toute séquence finie d' "entiers colorés strictement positifs" $\rho = (\rho_1 z_{i_1}, \dots, \rho_m z_{i_m})$ telle que pour tout k , ρ_k est un entier strictement positif, et z_{i_k} appartient à l'alphabet des couleurs. Le mot des couleurs de ρ est le mot $w = w(\rho) = z_{i_1} \dots z_{i_m}$. La séquence des entiers

figurant dans ρ pour la couleur z_j , après avoir réordonné en ordre croissant, forme une partition $\pi_j(\rho)$ pour toutes les couleurs z_j . Nous définissons la multiplicité de ρ comme la famille $\pi_j(\rho)$ pour toutes les couleurs z_j .

Une telle famille μ de partitions indexées par des couleurs, est appelée partition colorée. Nous introduisons ici la notation du produit tensoriel afin de pouvoir manipuler des combinaisons linéaires de partitions colorées.

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

où la partition μ_j est la composante de μ pour la couleur z_j . Aussi, sur l'ensemble des lettres différentielles $\{a_j\}$, la partition colorée μ désignera le monôme différentiel

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

Pour chaque couleur z_j , on désigne par a_j la partition colorée

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

La loi de dérivation D sur les partitions colorées, reflétant la dérivée par rapport au temps D_t des monômes différentiels à m lettres différentielles, est définie en étendant la loi D définie sur les composantes:

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

En terme d'algèbre linéaire, on obtient pour toute partition colorée μ , l'équation linéaire:

$$D(\mu) = \sum_{\substack{\text{partitions} \\ \text{colorées}} \tau} \langle \mu | \tau \rangle * \tau, \quad \langle \mu | \tau \rangle \in \mathbb{N}$$

► Analyse combinatoire de l'équation générique

Il s'agit de donner une interprétation combinatoire de la série \mathcal{A} en identifiant chaque monôme différentiel avec sa multiplicité colorée. En effet, \mathcal{A} peut être réécrit en factorisant selon les multiplicités colorées, sous la forme $\mathcal{A} = \sum_{\mu} \mu \cdot g_{\mu}$. L'équation générique (G) devient alors:

$$\sum_{\mu} \mu \cdot g_{\mu} = \epsilon + \sum_{j=1 \cdots m} \sum_{\sigma} a_j \cdot z_j \cdot \sigma \cdot g_{\sigma} + \sum_{\nu} D(\nu) \cdot g_{\nu}$$

On identifie dans les deux membres les facteurs de la même multiplicité non triviale $\mu \neq \epsilon$, et l'on obtient:

$$g_{\mu} = \sum_{j=1}^m z_j g_{\mu \triangleright a_j} + \sum_{\nu} \langle \nu | \mu \rangle * g_{\nu}, \quad \langle \nu | \mu \rangle \in \mathbb{N}$$

où le symbole \triangleright signifie:

$$\mu \triangleright a_j = \begin{cases} \sigma & \text{si } a_j \sigma = \mu \\ 0 & \text{sinon} \end{cases}$$

Les coefficients $\langle \nu | \mu \rangle$ sont relatifs aux primitives des multiplicités μ .

► **Multiplicités colorées non homogènes**

Nous particularisons ici le cas d'un système à une entrée avec dérive:

$$(\Sigma) \quad \begin{cases} \dot{q}(t) = g_0(q) + a_1(t)g_1(q) \\ y(t) = h(q(t)) \end{cases}$$

Dans ce cas, $a(0)(t)$ est identique à 1, et la multiplicité à deux couleurs de poids n est de la forme:

$$\mu = \mu_0 \otimes \nu \quad \text{avec} \quad \begin{cases} \mu_0 & = 1^p \\ \text{poids}(\mu) & = p + \text{poids}(\nu) = n \end{cases}$$

La loi de dérivation doit s'adapter car $D(\mu_0) = 0$ et donc:

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

L'entier p est appelé **profondeur** de la multiplicité $1^p \otimes \nu$.

Pour simplifier les notations, on pose:

$$g_{/p}^\nu = g_{(1^p \otimes \nu)}$$

Avec cette nouvelle notation, on obtient les équations de récurrence:

$$g_{/p}^\nu = z_0 g_{/p-1}^\nu + z_1 g_{/p}^{\nu \triangleright a} + \sum_{\sigma} \langle \sigma | \nu \rangle * g_{/p}^\sigma$$

Si $p = 0$, le premier terme disparaît. Si ν ne contient pas 1, le second terme disparaît. La dernière somme est relative aux "primitives" de ν . C'est une somme finie.

Bibliographie

- [1] Dederick L.S., *Successive derivatives of a function of several functions*, Ann. of Math. 27, pp. 385-394, 1925.
- [2] Hespel C., *Iterated derivatives of a non linear dynamic system and Faà di Bruno formula*, Mathematics and Computers in Simulation, vol. 42, pp. 641-657, 1996.
- [3] Hespel C., Jacob G., *First steps towards exact algebraic identification*, Discrete Math., vol. 180, pp. 211-219, 1998.
- [4] Hespel C., Jacob G., *On Algebraic Identification of Causal Functionals*, 10th International Conference on Formal Power Series and Algebraic Combinatorics, Toronto, june 14-19, 1998.
- [5] Hespel C., Jacob G., Monnier E., *Premier pas vers l'Identification Algébrique: un algorithme en MAPLE*, publication interne LANS n° 80, INSA Rennes, 1998.
- [6] Lamnabhi-Lagarrigue F., Crouch P.E., *A formula for iterated derivatives along trajectories of nonlinear systems*, Systems and Control letters 11, pp. 1-7, 1988.

Chapitre 9

Une deuxième étape vers l'Identification algébrique: Un algorithme effectif d'identifiabilité de la série génératrice jusqu'au degré l

Nous proposons, dans ce chapitre, un algorithme permettant de savoir si l'on peut identifier les coefficients de la série génératrice G jusqu'au degré l , à partir du système d'équations linéaires obtenu précédemment:

$$y^{(k)}(0) = \sum_{\mu} a^{\mu} \langle G | g_{\mu} \rangle$$

Dans une première étape, nous avons calculé les contributions de G , relatives à une famille $\{g_{\mu}\}$ de polynômes associés aux multidérivées des entrées (famille indexée par l'ensemble des partitions).

Dans cette deuxième étape, il reste à extraire les coefficients $\langle G | w \rangle$. Après une analyse de ce problème de scindage, nous proposons un algorithme. Il est à noter que cette seconde étape d'identification n'est pas de même nature que la première: elle est de nature purement algébrique et donc générique. Cette phase de l'identification peut être calculée une fois pour toutes. Cette remarque fait ressortir l'intérêt de pouvoir réaliser le calcul assez loin.

9.1 Analyse du problème de scindage

9.1.1 Identification des coefficients de la série génératrice dans le cas d'une entrée sans dérive

Ce cas a déjà été traité au chapitre précédent. Nous conservons les mêmes notations. Nous notons l'entrée par $a(t)$ et nous la fixons polynômiale :

$$a(t) = \sum_{i=0}^k \frac{c_i}{i!} t^i.$$

Alors $A^{(0)}(0) = a(0)z_1 = c_0.z_1$ et $A^{(i)}(0) = c_i.z_1$, et

$$a^{(\rho_1)} \dots a^{(\rho_p)} = a^{\mu_1} \dots (a^{(k-1)})^{\mu_k} = a^\mu$$

nous obtenons :

$$y^{(n)}(0) = \sum_{\text{poids}=n} c_0^{\mu_1} \dots c_{k-1}^{\mu_k} \langle G|z_1^{\mu_1+\dots+\mu_k} \rangle \lambda_\mu$$

Nous avons donc à résoudre un système triangulaire infini d'équations linéaires. Dans ce cas, le calcul des coefficients inconnus $\langle G|z_1^k \rangle$ est évident grâce à un choix de l'entrée satisfaisant :

$$a(0) = c_0 \neq 0,$$

si les dérivées des sorties correspondantes $y^{(p)}(0)$ sont connues pour $0 \leq p \leq k$.

9.1.2 Identification des coefficients de la série génératrice dans le cas d'une entrée avec dérive

On se limite au cas d'une entrée avec dérive. Le problème est d'identifier les coefficients, dans la série génératrice, des mots de longueur l , contenant l_1 occurrences de z_1 .

► Quelles équations choisir?

Choisir les équations revient à sélectionner les monômes différentiels dans la famille:

$$\{(a_1^{(i_1)})^{e_1} \dots (a_1^{(i_q)})^{e_q} a_0^{f_0}\}$$

tels que

$$\begin{cases} \sum_j e_j & = l_1 \\ f_0 & = l - l_1 \\ f_0 + \sum_j e_j(1 + i_j) & = n \end{cases}$$

pour $n \in \mathbb{N}$.

Une autre écriture pour la famille est:

$$\left\{ \overbrace{(a_1^{(i_1)} \dots a_1^{(i_1)})}^{e_1} \dots \dots \dots \overbrace{(a_1^{(i_q)} \dots a_1^{(i_q)})}^{e_q} a_0^{f_0} \right\}$$

Considérons les suites $(i_1, \dots, i_1, \dots, i_q, \dots, i_q)$

Le choix proposé consiste à retenir parmi ces suites, les séquences monotones :

Par exemple, si on veut identifier les coefficients des mots de longueur $l = 4$ contenant $l_1 = 2$ occurrences de z_1 , on sélectionne parmi les monômes

$$\{(a_1^{(i_1)})^{e_1} (a_1^{(i_2)})^{e_2} a_0^2\}$$

tels que

$$l_1 = e_1 + e_2 = 2, \quad l - l_1 + e_1(1 + i_1) + e_2(1 + i_2) = n \in \mathbb{N}$$

les monômes suivants:

$$\begin{array}{lll} a_1^2 a_0^2, & & \\ a_1 a_1^{(1)} a_0^2, & a_1^{(1)} a_1^{(1)} a_0^2, & \\ a_1 a_1^{(2)} a_0^2, & a_1^{(1)} a_1^{(2)} a_0^2, & a_1^{(2)} a_1^{(2)} a_0^2 \end{array}$$

On peut interpréter chaque i_j , comme un décalage sur les 1, à partir de la position de départ où les 1 sont calés à gauche dans le mot. Ceci établit une correspondance entre le monôme différentiel et le mot.

pour l'exemple précédent,

$$\begin{array}{lll} 1100 & & \\ (i_2 = 0, i_1 = 0) & & \\ \\ 1010 & 0110 & \\ (i_2 = 1, i_1 = 0) & (i_2 = 1, i_1 = 1) & \\ \\ 1001 & 0101 & 0011 \\ (i_2 = 2, i_1 = 0) & (i_2 = 2, i_1 = 1) & (i_2 = 2, i_1 = 2) \end{array}$$

Notons

$$\{(a_1^{(i_1)})^{e_1} \dots (a_1^{(i_q)})^{e_q} a_0^{f_0}\}_F$$

la famille des monomes différentiels ainsi définie.

Reste à vérifier que le déterminant dont les colonnes sont indicées par les mots de même image commutative et les lignes par les monômes différentiels correspondants ne devra jamais être nul.

Soit $\Delta_l^{l_1}$ ce déterminant. C'est le déterminant du système d'équations linéaires choisies pour identifier les coefficients des mots de longueur l contenant l_1 occurrences de z_1 , de la série génératrice.

► **Jusqu'à quel ordre n_0 aller?**

Admettons, dans ce paragraphe, avoir démontré que les monômes différentiels ainsi sélectionnés permettent l'identification des coefficients des mots correspondants de la série génératrice.

Proposition 9.1.1 *L'identification des coefficients de la série génératrice, à l'ordre l , peut se faire si l'on connaît les dérivées successives de la sortie jusqu'à l'ordre*

$$n_0 = l + E[l/2](l - E[l/2])$$

en notant $E[l/2]$ la partie entière de $l/2$.

Preuve

Pour identifier les coefficients des mots de longueur l contenant l_1 occurrences de z_1 , le monôme de plus grand poids, servant à l'identification, vérifie :

$$n_0 = \text{poids}(\mu) = (l - l_1) + l_1(1 + (l - l_1))$$

car ce monôme est :

$$(a_1^{(l-l_1)})^{l_1} a_0^{l-l_1}$$

(Le décalage maximum correspondant à $i_q = i_{q-1} = \dots = i_1 = l - l_1$, cela se traduit par un ordre de dérivation égal à $l - l_1$ sur les l_1 occurrences de a_1 .)

Donc, $n_0 = l + l_1(l - l_1)$, et n_0 est maximum pour $l_1 = E[l/2]$. L'ordre de dérivation maximum de $y(t)$ intervenant dans le calcul d'identification de la série génératrice à l'ordre l est donc :

$$n_0 = l + E[l/2](l - E[l/2])$$

Fin de preuve

La méthode consiste à mettre chaque $\Delta_l^{l_1}$ sous forme triangulaire par blocs, chaque bloc étant de déterminant $\neq 0$.

► **Mise sous forme triangulaire par blocs du déterminant**

$$\Delta_l^{l_1} = \det(\{w \in Z^l \ / |w|_{z_1} = l_1\} * \{(a_1^{(i_1)})^{e_1} \dots (a_1^{(i_q)})^{e_q} a_0^{f_0}\}_F)$$

◊ Rappelons la relation de récurrence:

$$\begin{cases} A_1 & = \mathcal{L}_a = a_0 z_0 + a_1 z_1 \\ A_{i+1} & = \mathcal{L}_a A_i + D_t A_i \end{cases}$$

pour $C_a^{(i)} = C_a A_i$,

$$y^{(n)}(t) = \langle G \| C_a^{(n)}(t) \rangle$$

Il s'ensuit les remarques suivantes:

1. lorsque l'indice de colonne est un mot commençant par z_1 , les lignes indicées par les monômes:

$$(a_1^{(i_1)})^{e_1} \dots (a_1^{(i_q)})^{e_q} a_0^{f_0}$$

pour $0 < i_1 < i_2 < \dots < i_q$, sont combinaisons linéaires des lignes indicées par les monômes primitives:

$$(a_1^{(i_1-1)})^{e_1} \dots (a_1^{(i_q)})^{e_q} a_0^{f_0}$$

...

$$(a_1^{(i_1)})^{e_1} \dots (a_1^{(i_q-1)})^{e_q} a_0^{f_0}.$$

2. lorsque l'indice de colonne est un mot commençant par z_0 , il faut ajouter à ces combinaisons linéaires des primitives, la contribution due au mot obtenu en enlevant la première lettre z_0 , pour le monome

$$(a_1^{(i_1)})^{e_1} \dots (a_1^{(i_q)})^{e_q} a_0^{f_0-1}$$

En choisissant un ordre judicieux sur les mots, cette remarque permet d'écrire $\Delta_l^{l_1}$ sous forme triangulaire supérieure par blocs.

Cette remarque induira un ordre sur les mots: On les ordonnera par ordre décroissant des lettres en partant du début.

De cette manière, on aura d'abord les mots commençant par l_1 occurrences de z_1 , puis par $l_1 - 1$ occurrences de z_1 , ..., puis par 0 occurrences de z_1 .

Ce classement sera conforme à l'ordre défini sur les monômes différentiels.

Par exemple, pour $l = 4, l_1 = 2$, on classera les mots dans l'ordre suivant:

$$1100; 1010; 1001;$$

$$0110; 0101;$$

$$0011$$

En effectuant les combinaisons de lignes indiquées, Δ_4^2 s'écrit:

	1100	1010	1001	0110	0101	0011
$a_1^2 a_0^2$	1	1	1	1	1	1
$a_1^{(1)} a_1 a_0^2$	$\binom{2}{1} + \binom{1}{1}$	$\binom{3}{1} + \binom{1}{1}$	$\binom{4}{1} + \binom{1}{1}$	$\binom{3}{1} + \binom{2}{1}$	$\binom{4}{1} + \binom{2}{1}$	$\binom{4}{1} + \binom{3}{1}$
$a_1^{(2)} a_1 a_0^2$	$\binom{3}{2} + \binom{2}{2}$	$\binom{4}{2} + \binom{2}{2}$	$\binom{5}{2} + \binom{2}{2}$	$\binom{4}{2} + \binom{3}{2}$	$\binom{5}{2} + \binom{3}{2}$	$\binom{5}{2} + \binom{4}{2}$
$(a_1^{(1)})^2 a_0^2$	0	0	0	$\binom{3}{1}$	$\binom{4}{1}$	8
$-a_1^{(1)} a_1 a_0^2$						
$a_1^{(2)} a_1^{(1)} a_0^2$						
$-2(a_1^{(1)})^2 a_0^2$	0	0	0	$\binom{4}{2} + \binom{4}{1}$	$\binom{5}{2} + \binom{5}{1}$	35
$-a_1^{(2)} a_1 a_0^2$						
$(a_1^{(2)})^2 a_0^2$						
$-a_1^{(2)} a_1^{(1)} a_0^2$						
$-a_1^{(2)} a_1^{(1)} a_0^2$	0	0	0	0	0	$\binom{5}{2}$
$+2(a_1^{(1)})^2 a_0^2$						
$+a_1^{(2)} a_0^2$						

De manière générale,

$$\Delta_l^{l_1} = \prod_{j=0}^{l-l_1} \delta_{l,j}^{l_1}$$

en posant

$$(\delta_{l,j}^{l_1} = \det(\{wz_1 \in Z^{l_j} \mid |w|_{z_1} = l_1 - 1\} * \{M/a_0^j\}_F)$$

où $\{M/a_0^j\}_F$ représente l'ensemble des monômes différentiels, associés à $z_0^j z_1 w$, divisés par a_0^j .

Par exemple,

$$\Delta_4^2 = \delta_{4,0}^2 * \delta_{4,1}^2 * \delta_{4,2}^2$$

avec

$$\begin{aligned} \delta_{4,0}^2 &= \det(\{1100, 1010, 1001\}, \{a_1^2 a_0^2, a_1^{(1)} a_1 a_0^2, a_1^{(2)} a_1 a_0^2\}) \\ \delta_{4,1}^2 &= \det(\{110, 101\}, \{(a_1^{(1)})^2 a_0, a_1^{(2)} a_1^{(1)} a_0\}) \\ \delta_{4,2}^2 &= \det(\{11\}, \{(a_1^{(2)})^2\}) \end{aligned}$$

► **Vérification de la stricte positivité de chaque déterminant $\delta_{l,j}^{l_1}$**

Pour $l_1 = 1$ et $l_1 = 2$, nous présentons une démonstration de la stricte positivité du déterminant $\delta_{l,j}^{l_1}$ et donnons une méthode de calcul de $\delta_{l,j}^{l_1}$.

Pour des valeurs de l_1 supérieures, nous conjecturons que chaque déterminant diagonal $\delta_{l,j}^{l_1}$ est non nul, nous suggérons une éventuelle démonstration, et donnons une méthode de calcul de $\delta_{l,j}^{l_1}$.

Nous proposons un début de récurrence sur le nombre d'occurrences l_1 de z_1 :

- **Pour une occurrence de z_1 :**

Alors, $l_1 = 1$, Δ_6^1 s'écrit

	100000	010000	001000	000100	000010	000001
$a_1 a_0^5$	1	1	1	1	1	1
$a_1^{(1)} a_0^5$	$\binom{1}{1}$	$\binom{2}{1}$	$\binom{3}{1}$	$\binom{4}{1}$	$\binom{5}{1}$	$\binom{6}{1}$
$a_1^{(2)} a_0^5$	$\binom{2}{2}$	$\binom{3}{2}$	$\binom{4}{2}$	$\binom{5}{2}$	$\binom{6}{2}$	$\binom{7}{2}$
$a_1^{(3)} a_0^5$	$\binom{3}{3}$	$\binom{4}{3}$	$\binom{5}{3}$	$\binom{6}{3}$	$\binom{7}{3}$	$\binom{8}{3}$
$a_1^{(4)} a_0^5$	$\binom{4}{4}$	$\binom{5}{4}$	$\binom{6}{4}$	$\binom{7}{4}$	$\binom{8}{4}$	$\binom{9}{4}$
$a_1^{(5)} a_0^5$	$\binom{5}{5}$	$\binom{6}{5}$	$\binom{7}{5}$	$\binom{8}{5}$	$\binom{9}{5}$	$\binom{10}{5}$

On effectue d'abord $\forall k \geq 0$, la transformation θ de la colonne indiquée par le mot w_k commençant par k zéros:

$$w_k = \overbrace{0 \cdots 0}^k 10 \cdots 0$$

en la colonne indiquée par le mot m_k suivant:

$$m_k = w_k - \binom{k}{1} w_{k-1} + \binom{k}{2} w_{k-2} + \cdots + (-1)^k \binom{k}{k} w_0$$

Alors, il est facile de démontrer qu'après cette transformation θ , Δ_l^1 devient le triangle de Pascal:

	m_0	m_1	m_2	m_3	m_4	m_5
$a_1 a_0^5$	$\binom{0}{0}$					
$a_1^{(1)} a_0^5$	$\binom{1}{1}$	$\binom{1}{0}$				
$a_1^{(2)} a_0^5$	$\binom{2}{2}$	$\binom{2}{1}$	$\binom{2}{0}$			
$a_1^{(3)} a_0^5$	$\binom{3}{3}$	$\binom{3}{2}$	$\binom{3}{1}$	$\binom{3}{0}$		
$a_1^{(4)} a_0^5$	$\binom{4}{4}$	$\binom{4}{3}$	$\binom{4}{2}$	$\binom{4}{1}$	$\binom{4}{0}$	
$a_1^{(5)} a_0^5$	$\binom{5}{5}$	$\binom{5}{4}$	$\binom{5}{3}$	$\binom{5}{2}$	$\binom{5}{1}$	$\binom{5}{0}$

En effet, on peut démontrer la formule:

$$\binom{n+k}{n} - \binom{k}{1} \binom{n+k-1}{n} + \binom{k}{2} \binom{n+k-2}{n} + \cdots + (-1)^k \binom{k}{k} \binom{n}{n} = \binom{n}{n-k}$$

en appliquant itérativement l'identité:

$$\binom{n-1}{k-1} = \binom{n}{k} - \binom{n-1}{k}$$

Le déterminant de Δ_l^1 est donc > 0 . De plus, on a la propriété démontrée par X.Viennot: [4]

Théorème 15 Soit $0 \leq a_1 < \cdots < a_k$ et $0 \leq b_1 < \cdots < b_k$ des suites strictement croissantes d'entiers naturels représentant des indices de lignes et de colonnes.

Soit $A_i = (0, a_i)$ et $B_i = (b_i, b_i)$. Alors le déterminant binomial

$$\begin{pmatrix} a_1, \dots, a_k \\ b_1, \dots, b_k \end{pmatrix}$$

est le nombre de k -tuples (w_1, \dots, w_k) de chemins de $\mathbb{N} \times \mathbb{N}$ tels que:

- 1) $\forall i, w_i$ est un chemin de A_i à B_i .
- 2) Les chemins w_i sont deux à deux disjoints.

Ce théorème a pour corollaire :

Corollaire 9.1.1 *Le déterminant binomial*

$$\begin{pmatrix} a_1, \dots, a_k \\ b_1, \dots, b_k \end{pmatrix}$$

est positif ou nul. Il est strictement positif ssi $b_i \leq a_i, \forall i$

Cette propriété nous paraissant particulièrement intéressante, nous proposons la définition suivante:

Définition 9.1.1 *Nous dirons qu'un mineur spécifié par les listes de ses lignes et colonnes*

$$\begin{pmatrix} a_1, \dots, a_k \\ b_1, \dots, b_k \end{pmatrix}$$

extrait d'un déterminant, vérifiant

$$0 \leq a_1 < \dots < a_k, \quad 0 \leq b_1 < \dots < b_k$$

et $\forall i, b_i \leq a_i$, satisfait la propriété de Viennot.

Définissons la propriété (P_1) suivante:

Définition 9.1.2 *On dit qu'un tableau carré D satisfait la propriété (P_1) ssi*

- D est de déterminant strictement positif
- Tout mineur de D satisfaisant la propriété de Viennot est de déterminant strictement positif.

Alors, on observe que Δ_l^1 vérifie la propriété (P_1) .

Remarque 9.1.1 *La propriété (P_1) est probablement une propriété trop forte pour le problème qui nous concerne. Une propriété plus faible telle que la propriété (P'_1) nous suffirait:*

Définition 9.1.3 *On dit qu'un tableau carré D satisfait la propriété (P'_1) ssi tout mineur obtenu en sélectionnant une liste de n lignes d'indices croissants (i_1, \dots, i_n) et la liste des n colonnes consécutives d'indices $1 \dots n$, est de déterminant > 0 .*

– **Pour deux occurrences de z_1 :**

Alors, $l_1 = 2$. Montrons que chaque $\delta_{i,j}^2$ est de déterminant > 0 .

Pour cela, on effectue dans chaque bloc, la même transformation θ sur les colonnes que pour $l_1 = 1$ et on remarque que

$$\begin{aligned}\delta_{i,j}^2 &= T_l^2 \bar{\delta}_{i,j-1}^2, \quad \forall j \geq 1 \\ \delta_{i,0}^2 &= T_l^2 \Delta_i^1\end{aligned}$$

en notant T_l^2 “la matrice de dérivation” c’est à dire la matrice de passage d’une à deux occurrences de z_1 , relative aux mots de longueur l .

$$T_l^2 = \begin{pmatrix} 1 & & & & \\ 2 & 1 & & & \\ 2 & 1 & 1 & & \\ 2 & 1 & 1 & 1 & \\ \vdots & \vdots & \vdots & \vdots & \end{pmatrix}$$

$\bar{\delta}_{i,j-1}^2$ étant le tableau obtenu en rayant la première ligne et la dernière colonne de $\delta_{i,j-1}^2$.

On démontre alors la proposition:

Proposition 9.1.2 *Si un tableau D possède la propriété (P_1) , alors, si on note par \bar{D} , le déterminant obtenu en rayant la première ligne et la dernière colonne de D , $T_{l-1}^2 \bar{D}$ a la même propriété (P_1) .*

Preuve

On remarque que pour $D = \begin{pmatrix} L_1 \\ L_2 \\ L_3 \\ L_4 \\ \vdots \end{pmatrix}$, en notant L_1, L_2, \dots , les lignes de D , on a

$$T_{l-1}^2 \bar{D} = \begin{pmatrix} 1 & & & & \\ 2 & 1 & & & \\ 2 & 1 & 1 & & \\ 2 & 1 & 1 & 1 & \\ \vdots & \vdots & \vdots & \vdots & \end{pmatrix} * \begin{pmatrix} L_2 \\ L_3 \\ L_4 \\ L_5 \\ \vdots \end{pmatrix} = \begin{pmatrix} L_2 \\ 2 * L_2 + L_3 \\ 2 * L_2 + L_3 + L_4 \\ 2 * L_2 + L_3 + L_4 + L_5 \\ \vdots \end{pmatrix}$$

Il est clair que $T_{l-1}^2 \bar{D}$ a un déterminant strictement positif car c’est une combinaison linéaire à coefficients positifs des mineurs de D satisfaisant la propriété de Viennot. Cette remarque reste vraie si, au lieu de considérer $T_{l-1}^2 \bar{D}$, on considère cette fois ses mineurs de Viennot.

Fin de preuve

Définition 9.1.4 Soit (P_2) la propriété suivante:

On dit qu'un tableau carré D satisfait la propriété (P_2) ssi

- D est de déterminant strictement positif et on peut le mettre par un changement de base sous forme diagonal par blocs.
- Les blocs figurant dans D avant sa mise sous forme diagonale sont tels que tout mineur de Viennot extrait d'un de ces blocs est de déterminant strictement positif.

Alors, on observe que Δ_l^2 vérifie la propriété (P_2) . D'autre part, on s'appuie sur la proposition conjecturée suivante:

Proposition 9.1.3 Si $\delta_{i,j}^3$ possède la propriété (P_2) , alors, si on note par $\bar{\delta}_{i,j}^3$ le déterminant obtenu en rayant le premier bloc horizontal et la dernière colonne de chaque bloc vertical de $\delta_{i,j}^3$,

$$\delta_{i,j+1}^3 = T_{l-1}^3 \bar{\delta}_{i,j}^3$$

à la même propriété (P_2) .

Idée de preuve

La preuve pourrait être basée sur les 2 points suivants:

- * T_{l-1}^3 maintient la propriété que tout mineur de Viennot de chaque bloc est de déterminant strictement positif.
- * La mise sous forme diagonale par blocs de $\delta_{i,j+1}^3$ fait apparaître par des pivots de Gauss, un bloc supérieur de déterminant non nul par construction, et des blocs diagonaux de la forme:

$$\tau(\delta_{i,j+1}^2)$$

où τ est une matrice régulière.

Fin d'idée de preuve

Puisque Δ_l^2 vérifie la propriété (P_2) , il en résulte que

$$\det(\delta_{i,j}^3) \neq 0 \quad \forall j.$$

En particulier, Δ_l^3 est de déterminant non nul.

Par exemple, le calcul de $\delta_{5,0}^3$ donne après manipulation sur les colonnes dans

chaque bloc, conformément à la transformation θ de Δ_l^1 en le triangle de Pascal:

	11100	11010 - ...	11001 - ...	10110	10101 - ...	10011
$a_1^3 a_0^2$	1			1		1
$a_1^{(1)} a_1^2 a_0^2$	6	1		8	1	10
$a_1^{(2)} a_1^2 a_0^2$	10	4	1	17	5	26
$a_1^{(1)^2} a_1^2 a_0^2$	15	3		24	4	35
$a_1^{(2)} a_1^{(1)} a_1 a_0^2$	60	19	3	117	31	203
$a_1^{(2)^2} a_1 a_0^2$	70	24	4	162	49	329

puis par mise sous forme diagonale par blocs:

	11100	11010 - ...	11001 - ...	10110	10101 - ...	10011
$a_1^3 a_0^2$	1					
$a_1^{(1)} a_1^2 a_0^2$	6	1				
$a_1^{(2)} a_1^2 a_0^2$	10	4	1			
$a_1^{(1)^2} a_1^2 a_0^2 - \dots$				3	1	
$a_1^{(2)} a_1^{(1)} a_1 a_0^2 - \dots$				22	9	
$a_1^{(2)^2} a_1 a_0^2 - \dots$						10

On constate bien que chaque bloc diagonal est de la forme $\tau(\delta_{4,j}^2)$ donc se calcule à partir des déterminants obtenus pour 2 occurrences.

De même, $\delta_{4,1}^3$ a pour déterminant:

	1110	1101 - ...	1011
$a_1^{(1)^3} a_0^2$	15	3	24
$a_1^{(2)} a_1^{(1)^2} a_0^2$	105	28	189
$a_1^{(2)^2} a_1^{(1)} a_0^2$	280	80	540

puis par mise sous forme diagonale par blocs:

	1110	1101 - ...	1011
$a_1^{(1)^3} a_0^2$	15	3	
$a_1^{(2)} a_1^{(1)^2} a_0^2$	105	28	
$a_1^{(2)^2} a_1^{(1)} a_0^2 - \dots$			20

De plus, $\forall j, \delta_{l,j}^3$ vérifie la propriété (P_2) .

Remarque 9.1.3 Comme dans le cas $l_1 = 2$, on peut définir une propriété (P'_2) plus faible que (P_2) et qui, d'une certaine manière, est une généralisation de (P'_1) .

Définition 9.1.5 Soit (P'_2) la propriété suivante:

Un tableau carré D composé de blocs horizontaux indicés, de taille décroissante et

composé de blocs verticaux indicés, de taille décroissante eux aussi, de sorte que les blocs diagonaux soient carrés, satisfait la propriété (P'_2) ssi tout mineur M_p de D obtenu en sélectionnant p blocs horizontaux d'indices croissants (h_1, \dots, h_p) , complets selon les lignes, et en sélectionnant les p premiers blocs verticaux d'indices consécutifs $1 \dots p$ éventuellement tronqués à droite, pour que les blocs diagonaux du mineur soient carrés, vérifie $\det(M_p) > 0$.

Si notre conjecture est exacte, il s'agit d'établir les propositions suivantes:

- Δ_l^2 vérifie (P'_2)
- Si D a la propriété (P'_2) , alors $T_{l-1}^3 \overline{DI}$ (en notant \overline{DI} , le déterminant obtenu en rayant le premier bloc horizontal et en supprimant la dernière colonne de chaque bloc vertical), a la même propriété (P'_2) .

Il est clair que ceci permettra de montrer que

$\forall j \geq 0 \delta_{l,j}^3$ vérifie (P'_2)
 et Δ_l^4 est de déterminant > 0 .

Donnons d'abord une idée de preuve de la proposition conjecturée suivante:

Proposition 9.1.4 Δ_l^2 vérifie (P'_2)

Idée de preuve

Par construction, Δ_l^2 est composé de blocs. Il a la forme suivante (pour 4 blocs):

$$\begin{pmatrix} A_1 = T_l^2(\Delta_l^1) & B_1 = T_l^2(A_1 I) & C_1 = T_l^2(B_1 I) & D_1 = T_l^2(C_1 I) \\ A_2 = T_l^2(\overline{A_1}) & B_2 = T_l^2(\overline{B_1}) + T_l^2(A_2)I & C_2 = T_l^2(\overline{C_1}) + T_l^2(B_2)I & D_2 = T_l^2(\overline{D_1}) + T_l^2(C_2)I \\ A_3 = T_l^2(\overline{A_2}) & B_3 = T_l^2(\overline{B_2}) + T_l^2(A_3)I & C_3 = T_l^2(\overline{C_2}) + T_l^2(B_3)I & D_3 = T_l^2(\overline{D_2}) + T_l^2(C_3)I \\ A_4 = T_l^2(\overline{A_3}) & B_4 = T_l^2(\overline{B_3}) + T_l^2(A_4)I & C_4 = T_l^2(\overline{C_3}) + T_l^2(B_4)I & D_4 = T_l^2(\overline{D_3}) + T_l^2(C_4)I \end{pmatrix}$$

en notant XI , le bloc X tronqué en enlevant une ou des colonne(s) de droite.

Dans un premier temps, on montre aisément qu'en sélectionnant des blocs horizontaux consécutifs, le déterminant obtenu est toujours > 0 . Il suffit pour le voir, de remarquer qu'en effectuant les combinaisons linéaires de blocs suivantes (en notant H_i le i ème bloc horizontal):

$$H_i \rightarrow H_i - T_l^2(\overline{H_{i-1}})$$

du bas vers le haut plusieurs fois jusqu'à mettre le tableau sous forme triangulaire, les blocs diagonaux sont les $\delta_{l,j}^3$.

Le plus dur reste à faire dans le cas où on sélectionne des blocs horizontaux non consécutifs. Si par exemple, on sélectionne les blocs 1,3,4, il vient le mineur:

$$\begin{pmatrix} A_1 = T_l^2(\Delta_l^1) & (B_1)I = (T_l^2(A_1 I))I & (C_1)I = (T_l^2(B_1 I))I \\ A_3 = T_l^2(\overline{A_2}) & (B_3)I = (T_l^2(\overline{B_2}) + T_l^2(A_3)I)I & (C_3)I = (T_l^2(\overline{C_2}) + T_l^2(B_3)I)I \\ A_4 = T_l^2(\overline{A_3}) & (B_4)I = (T_l^2(\overline{B_3}) + T_l^2(A_4)I)I & (C_4)I = (T_l^2(\overline{C_3}) + T_l^2(B_4)I)I \end{pmatrix}$$

Pour démontrer cela, on peut proposer d'adapter une démonstration de la stricte positivité de tout mineur de Viennot de Δ_l^1 à la stricte positivité de tout bloc de Viennot de Δ_l^2 . On entend par bloc de Viennot, tout mineur obtenu par des choix d'indices croissants de blocs horizontaux, les blocs verticaux étant calés à gauche pour respecter les blocs carrés en diagonale (cf propriété (P_2')).

• Le schéma pourrait être pour Δ_l^1 :

1. On enlève la première ligne. On garde les premières colonnes.
Notons $(\Delta_l^1)_1$, le déterminant obtenu en enlevant la première ligne.

$$(\Delta_l^1)_1 = \begin{pmatrix} 1 & 1 & & & & & \\ 1 & 2 & 1 & & & & \\ 1 & 3 & 3 & 1 & & & \\ 1 & 4 & 6 & 4 & 1 & & \\ 1 & 5 & 10 & 10 & 5 & 1 & \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \end{pmatrix}$$

En effectuant des différences de lignes du bas vers le haut, on obtient:

$$(\Delta_l^1)_1 = \begin{pmatrix} 1 & 1 & & & & & \\ 0 & 1 & 1 & & & & \\ 0 & 1 & 2 & 1 & & & \\ 0 & 1 & 3 & 3 & 1 & & \\ 0 & 1 & 4 & 6 & 4 & 1 & \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \end{pmatrix}$$

Soit

$$(\Delta_l^1)_1 = (\Delta_{l-1}^1)_1$$

Il en résulte que $(\Delta_l^1)_1$ est > 0 .

2. On enlève une ligne quelconque. On garde les premières colonnes.
Notons $(\Delta_l^1)_i$, le déterminant obtenu en enlevant la i ème ligne.

$$(\Delta_l^1)_i = \begin{pmatrix} 1 & & & & & & \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \binom{i-1}{0} & \dots & \binom{i-1}{i-1} & & & & \\ \binom{i+1}{0} & \dots & \binom{i+1}{i-1} & \binom{i+1}{i} & \binom{i+1}{i+1} & & \\ \binom{i+2}{0} & \dots & \binom{i+2}{i-1} & \binom{i+2}{i} & \binom{i+2}{i+1} & \binom{i+2}{i+2} & \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \end{pmatrix}$$

En effectuant des différences de lignes du bas vers le haut, on obtient la relation de récurrence, pour $i \geq 2$:

$$(\Delta_l^1)_i = (\Delta_{l-1}^1)_i + (\Delta_{l-1}^1)_{i-1}$$

On observe encore que $(\Delta_l^1)_i$ est > 0 .

3. On enlève plusieurs lignes. On garde les premières colonnes.

Notons $(\Delta_l^1)_{i,j}$, le déterminant obtenu en enlevant la i ème et la j ème ligne:

$$\begin{pmatrix} \dots & \dots \\ \dots & \begin{pmatrix} i-1 \\ 0 \\ i+1 \\ 0 \end{pmatrix} & \dots & \begin{pmatrix} i-1 \\ i-2 \\ i+1 \\ i-2 \end{pmatrix} & \dots & \begin{pmatrix} i-1 \\ i-1 \\ i+1 \\ i-1 \end{pmatrix} & \dots & \begin{pmatrix} i+1 \\ i+1 \\ i \\ i+1 \end{pmatrix} & \dots & \dots & \dots \\ \dots & \begin{pmatrix} j-1 \\ 0 \\ j+1 \\ 0 \end{pmatrix} & \dots & \begin{pmatrix} j-1 \\ i-2 \\ j+1 \\ i-2 \end{pmatrix} & \dots & \begin{pmatrix} j-1 \\ i-1 \\ j+1 \\ i-1 \end{pmatrix} & \dots & \begin{pmatrix} j-1 \\ i \\ j+1 \\ i+1 \end{pmatrix} & \dots & \begin{pmatrix} j-1 \\ j-1 \\ j+1 \\ j-1 \end{pmatrix} & \dots \\ \dots & \dots \end{pmatrix}$$

En effectuant des différences de lignes du bas vers le haut, on obtient, grâce à la relation:

$$\binom{l+1}{k} = \binom{l-1}{k} + 2 \binom{l-1}{k-1} + \binom{l-1}{k-2}$$

la relation de récurrence, pour $i, j \geq 2$:

$$(\Delta_l^1)_{i,j} = (\Delta_{l-1}^1)_{i,j} + (\Delta_{l-1}^1)_{i,j-1} + (\Delta_{l-1}^1)_{i-1,j} + (\Delta_{l-1}^1)_{i-1,j-1}$$

On observe encore que $(\Delta_l^1)_{i,j}$ est > 0 .

Il semble que cette démonstration puisse se généraliser dans le cas où un nombre quelconque de lignes a été enlevé.

• Le schéma s'adapterait ainsi pour Δ_l^2 :

1. On enlève le bloc supérieur.

Ceci a déjà été traité au début de la preuve, en effectuant des combinaisons linéaires des blocs du bas vers le haut.

2. On enlève un bloc horizontal quelconque.

Enlevons par exemple le second bloc horizontal.

$$(\Delta_l^2)_2 = \left(\begin{array}{cccc|ccc} 1 & & & & 1 & & 1 \\ 3 & 1 & & & 5 & 1 & 7 \\ 4 & 3 & 1 & & 9 & 4 & 16 \\ 5 & 6 & 4 & 1 & 14 & 10 & 30 \\ \hline 10 & 5 & 1 & & 45 & 18 & 126 \\ 35 & 21 & 7 & 1 & 189 & 91 & 616 \\ \hline 35 & 21 & 7 & 1 & 224 & 112 & 840 \end{array} \right)$$

En effectuant des combinaisons linéaires du bas vers le haut, il vient:

$$(\Delta_l^2)_2 = \left(\begin{array}{cccc|ccc} 1 & & & & 1 & & 1 \\ 3 & 1 & & & 5 & 1 & 7 \\ 4 & 3 & 1 & & 9 & 4 & 16 \\ 5 & 6 & 4 & 1 & 14 & 10 & 30 \\ \hline 0 & 0 & 0 & 0 & 26 & 12 & 96 \\ 0 & 0 & 0 & 0 & 118 & 63 & 496 \\ \hline 0 & 0 & 0 & 0 & 153 & 84 & 720 \end{array} \right)$$

Le bloc carré inférieur droit de dimension 3 est sans doute une somme de combinaisons linéaires de $(\Delta_{l-1}^2)_2$ et de $(\Delta_{l-1}^2)_1$, mais la relation suivante ne permet pas d'obtenir clairement une réponse:

$$\begin{pmatrix} 26 & 12 \\ 118 & 63 \end{pmatrix} = T_l^2 \begin{pmatrix} 6 & 2 \\ 12 & 2 \end{pmatrix} * \begin{pmatrix} 3 & 1 \\ 4 & 3 \\ 5 & 6 \end{pmatrix}$$

On démontre en effet que

$$\begin{pmatrix} 26 & 12 \\ 118 & 63 \end{pmatrix} = T_l^2 (\overline{T_l^2 T_l^2} + T_l^2 \overline{T_l^2}) \begin{pmatrix} 3 & 1 & & \\ 4 & 3 & 1 & \\ 5 & 6 & 4 & 1 \end{pmatrix}$$

$$\begin{pmatrix} 26 & 12 \\ 118 & 63 \end{pmatrix} = T_l^2 \left(\begin{pmatrix} 10 & 5 \\ 35 & 21 \end{pmatrix} + \begin{pmatrix} 2 & 1 & \\ & 1 & 1 \end{pmatrix} * \begin{pmatrix} 3 & 1 \\ 10 & 5 \\ 15 & 11 \end{pmatrix} \right)$$

3. On enlève plusieurs blocs horizontaux.
 Cette partie n'a pas été encore traitée.

Fin d'idée de preuve

Proposition 9.1.5 *Si D a la propriété (P'_2) , alors $T_{l-1}^3 \overline{DI}$ (en notant \overline{DI} , le déterminant obtenu en rayant le premier bloc horizontal et en supprimant la dernière colonne de chaque bloc vertical), a la même propriété (P'_2) .*

Idée de preuve

L'idée est de décomposer T_l^3 en produit de matrices $T_{l,k}^3$, chacune de ces matrices n'agissant que sur un bloc horizontal. Par exemple,

$$T_l^3 = \begin{pmatrix} 1 & & \dots & & \dots & & \\ 3 & 1 & & \dots & & & \\ 3 & 1 & 1 & \dots & & & \\ 3 & 1 & 1 & 1 & \dots & & \\ \vdots & \vdots \\ 6 & 2 & & \dots & 1 & & \dots & & & & & & \\ 18 & 6 & 2 & \dots & 2 & 1 & \dots & & & & & & \\ 24 & 8 & 4 & 2 & \dots & 2 & 1 & 1 & \dots & & & & \\ \vdots & \vdots \\ 18 & 6 & 2 & \dots & 2 & 1 & \dots & 1 & & & & & \\ 60 & 20 & 8 & 2 & \dots & 6 & 3 & 1 & \dots & 2 & 1 & & \\ 60 & 20 & 8 & 2 & \dots & 6 & 3 & 1 & \dots & 2 & 1 & 1 & \end{pmatrix}$$

peut se décomposer en:

$$\left(\begin{array}{cccccccc|ccc} 1 & & & & & & & & & & & & \\ & 1 & & & & & & & & & & & \\ & & 1 & & & & & & & & & & \\ & & & 1 & & & & & & & & & \\ & & & & 1 & & & & & & & & \\ & & & & & 1 & & & & & & & \\ & & & & & & 1 & & & & & & \\ & & & & & & & 1 & & & & & \\ & & & & & & & & 1 & & & & \\ \hline & & & & & & & & & 0 & 1 & 1 & \end{array} \right) *$$

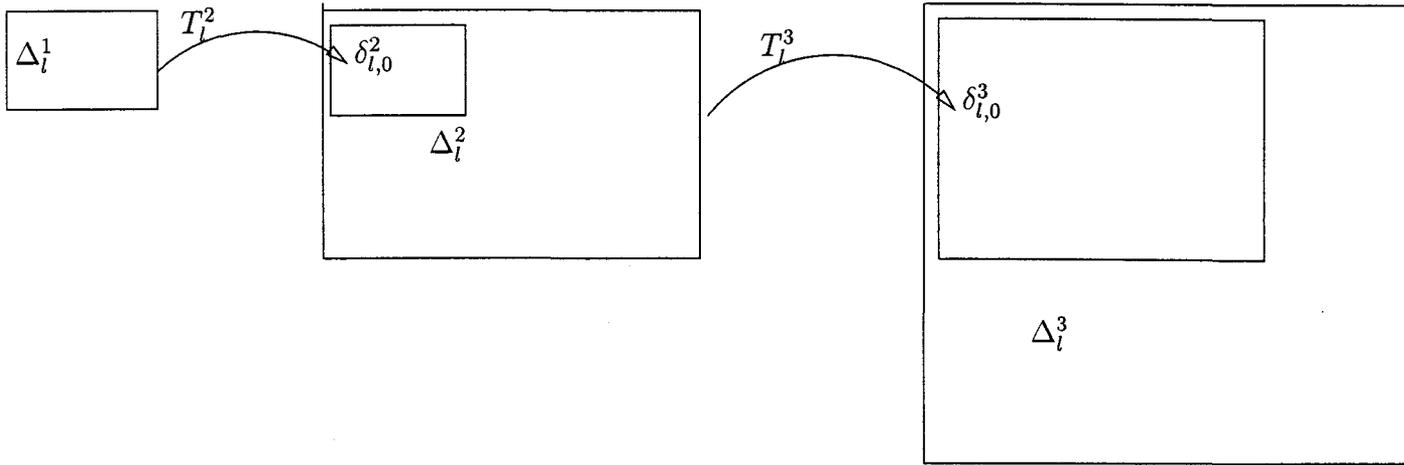


FIG. 9.1 – Calcul des déterminants $\Delta_l^{l_1}$

► **Conclusion: Méthode de calcul des déterminants successifs permettant de séparer les mots de même image commutative:**

Nous résumons les différentes étapes conduisant au calcul de $\Delta_l^{l_1}$:
 Notons t_{l-r-1}^{r+1} , l'opération consistant à tronquer un déterminant relatif aux mots de longueur l en enlevant les colonnes indicées par les mots de longueur $r+1$ et les lignes indicées par les monomes différentiels contenant $a_1^{(l-r-1)}$.

1. Après manipulation des colonnes, le déterminant associé au système d'équations linéaires concernant les coefficients des mots de longueur l contenant une occurrence de z_1 noté Δ_l^1 apparaît comme le déterminant de Pascal.
2. Le déterminant associé au système d'équations linéaires concernant les coefficients des mots de longueur l contenant l_1 occurrences de z_1 , pour $l_1 > 1$, apparaît comme un produit de blocs diagonaux, soit:

$$\Delta_l^{l_1} = \prod_{j=0}^{l-l_1} \delta_{l,j}^{l_1}$$

3. Le premier bloc diagonal a la forme suivante pour $l_1 > 1$:

$$\delta_{l,0}^{l_1} = T_l^{l_1} \Delta_l^{l_1-1}$$

4. En notant par \odot , le produit de composition, les autres blocs diagonaux ont la forme suivante:

$$\delta_{l,j}^{l_1} = \bigodot_{r=l-1}^{r=l-j} (T_r^{l_1} \circ t_{l-r-1}^{r+1}) \delta_{l,0}^{l_1}$$

Ceci peut se résumer par le schéma de la figure 9.1

► Exemple complet de calcul des coefficients de la série génératrice tronquée aux mots de longueur ≤ 3 :

On dispose donc d'un système d'équations linéaires:

$$y^{(k)}(0) = \sum_{\rho} a^{\rho} \langle G | g_{\rho} \rangle$$

- On sait qu'il suffit d'aller jusqu'à l'ordre de dérivation $n_0 = 5$ pour déterminer les coefficients des mots de longueur ≤ 3 .
 - On sait que des choix judicieux des entrées et de leurs dérivées permettent d'identifier les coefficients des multidérivées des entrées.
- *On trouve successivement, pour des ordres de dérivation 0 ou 1 de y :

$$\begin{aligned} \langle G | e \rangle &= y^{(0)} \\ \langle G | z_0 \rangle &= y^{(1)}(0) && \text{pour } a_1(0) = 0 \\ \langle G | z_1 \rangle &= y^{(1)}(0) - \langle G | z_0 \rangle && \text{pour } a_1(0) = 1 \end{aligned}$$

*Pour l'ordre de dérivation 2 de y , on obtient:

$$\langle G | z_0^2 \rangle = y^{(2)}(0) \quad \text{pour } a_1(0) = a_1^{(1)}(0) = 0$$

et,

$$\begin{pmatrix} \langle G | z_1 z_0 \rangle + \langle G | z_0 z_1 \rangle \\ \langle G | z_1^2 \rangle \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ c_0 & c_0^2 \end{pmatrix}^{-1} * \begin{pmatrix} y^{(2)}(0) /_{a_1(0)=1, a_1^{(1)}(0)=0} - \langle G | z_0^2 \rangle \\ y^{(2)}(0) /_{a_1(0)=c_0, a_1^{(1)}(0)=0} - \langle G | z_0^2 \rangle \end{pmatrix}$$

pour $c_0 \neq 0$ et $c_0 \neq 1$.

On obtient donc $\langle G | z_1 z_0 \rangle + \langle G | z_0 z_1 \rangle$ et $\langle G | z_1^2 \rangle$

*Pour l'ordre de dérivation 3 de y , on obtient:

$$\begin{aligned} \langle G | z_0^3 \rangle &= y^{(3)}(0) && \text{pour } a_1(0) = a_1^{(1)}(0) = a_1^{(2)}(0) = 0 \\ \begin{pmatrix} 2 \\ 1 \end{pmatrix} \langle G | z_0 z_1 \rangle + \begin{pmatrix} 1 \\ 1 \end{pmatrix} \langle G | z_1 z_0 \rangle &= y^{(3)}(0) - \langle G | z_0^3 \rangle && \text{pour } a_1^{(1)}(0) = 1, a_1(0) = a_1^{(2)}(0) = 0 \end{aligned}$$

et

$$\begin{aligned} \begin{pmatrix} \langle G | z_0^2 z_1 \rangle + \langle G | z_1 z_0^2 \rangle + \langle G | z_0 z_1 z_0 \rangle \\ \langle G | z_0 z_1^2 \rangle + \langle G | z_1 z_0 z_1 \rangle + \langle G | z_1^2 z_0 \rangle \\ \langle G | z_1^3 \rangle \end{pmatrix} &= \begin{pmatrix} 1 & 1 & 1 \\ c_1 & c_1^2 & c_1^3 \\ c_2 & c_2^2 & c_2^3 \end{pmatrix}^{-1} \\ &* \begin{pmatrix} y^{(3)}(0) /_{a_1(0)=1, a_1^{(1)}(0)=a_1^{(2)}(0)=0} - \langle G | z_0^3 \rangle \\ y^{(3)}(0) /_{a_1(0)=c_1, a_1^{(1)}(0)=a_1^{(2)}(0)=0} - \langle G | z_0^3 \rangle \\ y^{(3)}(0) /_{a_1(0)=c_2, a_1^{(1)}(0)=a_1^{(2)}(0)=0} - \langle G | z_0^3 \rangle \end{pmatrix} \end{aligned}$$

pour $c_1 \neq c_2 \neq 1$ et $c_1 \neq c_2 \neq 0$

On obtient donc:

$$\begin{aligned} \langle G | z_0^2 z_1 \rangle + \langle G | z_1 z_0^2 \rangle + \langle G | z_0 z_1 z_0 \rangle \\ \langle G | z_0 z_1^2 \rangle + \langle G | z_1 z_0 z_1 \rangle + \langle G | z_1^2 z_0 \rangle \\ \langle G | z_1^3 \rangle \end{aligned}$$

*Pour l'ordre de dérivation 4 de y , on obtient par le même procédé:

$$\begin{aligned} \langle G | z_0^4 \rangle &= y^{(4)}(0) \\ \text{pour } a_1(0) = a_1^{(1)}(0) = a_1^{(2)}(0) = a_1^{(3)}(0) = 0 \\ \begin{pmatrix} 1 \\ 1 \end{pmatrix} \langle G | z_1 z_0^3 \rangle + \begin{pmatrix} 2 \\ 1 \end{pmatrix} \langle G | z_0 z_1 z_0 \rangle + \begin{pmatrix} 3 \\ 1 \end{pmatrix} \langle G | z_0^2 z_1 \rangle &= y^{(4)}(0) - \begin{pmatrix} 3 \\ 1 \end{pmatrix} \langle G | z_1^2 \rangle - \langle G | z_0^4 \rangle \\ \text{pour } a_1^{(1)}(0) = 1, a_1(0) = a_1^{(2)}(0) = a_1^{(3)}(0) = 0 \end{aligned}$$

et par inversion de déterminant de Vandermonde, on peut obtenir:

$$\begin{pmatrix} \langle G|z_0^3 z_1 \rangle + \dots + \langle G|z_1 z_0^3 \rangle \\ \langle G|z_0^2 z_1^2 \rangle + \dots + \langle G|z_1^2 z_0^2 \rangle \\ \langle G|z_0 z_1^3 \rangle + \dots + \langle G|z_1^3 z_0 \rangle \\ \langle G|z_1^4 \rangle \end{pmatrix}$$

et on peut calculer:

$$\left(\binom{2}{1} + \binom{1}{1} \right) \langle G|z_1^2 z_0 \rangle + \left(\binom{3}{1} + \binom{1}{1} \right) \langle G|z_1 z_0 z_1 \rangle + \left(\binom{3}{1} + \binom{2}{1} \right) \langle G|z_0 z_1^2 \rangle$$

*Pour l'ordre de dérivation 5 de y , on obtient:

$$\begin{aligned} \langle G|z_0^5 \rangle &= y^{(5)}(0) \\ \text{pour } a_1(0) = a_1^{(1)}(0) = a_1^{(2)}(0) = a_1^{(3)}(0) = a_1^{(4)}(0) = 0 \\ \left(\binom{2}{2} \langle G|z_1 z_0^2 \rangle + \binom{3}{2} \langle G|z_0 z_1 z_0 \rangle + \binom{4}{2} \langle G|z_0^2 z_1 \rangle \right) &= y^{(5)}(0) - \langle G|z_0^5 \rangle \\ \text{pour } a_1^{(2)}(0) = 1, a_1(0) = a_1^{(1)}(0) = a_1^{(3)}(0) = a_1^{(4)}(0) = 0 \end{aligned}$$

et par inversion de déterminant de Vandermonde, on peut calculer

$$\begin{pmatrix} \binom{3}{1} \langle G|z_1^2 z_0 \rangle + \binom{4}{1} \langle G|z_1 z_0 z_1 \rangle + \binom{4}{1} * \binom{2}{1} \langle G|z_0 z_1^2 \rangle \\ \binom{1}{1} \langle G|z_1 z_0^3 \rangle + \binom{2}{1} \langle G|z_0 z_1 z_0^2 \rangle + \binom{3}{1} \langle G|z_0^2 z_1 z_0 \rangle + \binom{4}{1} \langle G|z_0^3 z_1 \rangle \end{pmatrix}$$

• Il reste à identifier les coefficients des mots de même image commutative:

- 1) Les coefficients des mots de longueur ≤ 1 ont déjà été calculés.
- 2) En longueur 2, il manque $\langle G|z_0 z_1 \rangle$ et $\langle G|z_1 z_0 \rangle$. Mais d'après les calculs faits pour les ordres 2 et 3, on connaît:

$$\left(\begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} \right) * \begin{pmatrix} \langle G|z_1 z_0 \rangle \\ \langle G|z_0 z_1 \rangle \end{pmatrix}$$

soit encore

$$\Delta_2^1 * \begin{pmatrix} \langle G|z_1 z_0 \rangle \\ \langle G|z_0 z_1 \rangle \end{pmatrix}$$

Donc, $\langle G|z_0 z_1 \rangle$ et $\langle G|z_1 z_0 \rangle$ sont bien identifiés.

- 3) En longueur 3, il manque $\langle G|w \rangle$, pour $|w| = 3, |w|_{z_1} = 1, 2$. Mais les calculs faits pour les ordres 3, 4, 5 fournissent la connaissance de:

$$\left(\begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 2 & 3 & 4 \\ 2 & 2 & 2 \end{pmatrix} \right) * \begin{pmatrix} \langle G|z_1 z_0^2 \rangle \\ \langle G|z_0 z_1 z_0 \rangle \\ \langle G|z_0^2 z_1 \rangle \end{pmatrix}$$

soit encore

$$\Delta_3^1 * \begin{pmatrix} \langle G|z_1 z_0^2 \rangle \\ \langle G|z_0 z_1 z_0 \rangle \\ \langle G|z_0^2 z_1 \rangle \end{pmatrix}$$

Et donc, $\langle G|z_0^2 z_1 \rangle, \langle G|z_0 z_1 z_0 \rangle, \langle G|z_1 z_0^2 \rangle$ sont identifiés.

Mais les calculs faits pour les ordres 3, 4, 5 fournissent la connaissance de:

$$\left(\begin{pmatrix} 1 & 1 & 1 \\ \binom{2}{1} + \binom{3}{1} & \binom{3}{1} + \binom{1}{1} & \binom{3}{1} + \binom{2}{1} \\ \binom{3}{1} & \binom{4}{1} & \binom{4}{1} * \binom{2}{1} \end{pmatrix} \right) * \begin{pmatrix} \langle G|z_1^2 z_0 \rangle \\ \langle G|z_1 z_0 z_1 \rangle \\ \langle G|z_0 z_1^2 \rangle \end{pmatrix}$$

soit encore

$$\Delta_3^2 * \begin{pmatrix} \langle G|z_1^2 z_0 \rangle \\ \langle G|z_1 z_0 z_1 \rangle \\ \langle G|z_0 z_1^2 \rangle \end{pmatrix}$$

Donc, $\langle G|z_0 z_1^2 \rangle$, $\langle G|z_1 z_0 z_1 \rangle$, $\langle G|z_1^2 z_0 \rangle$ sont bien identifiés. Tous les coefficients de la série génératrice G ont ainsi pu être identifiés jusqu'à l'ordre 3 fixé au départ.

9.2 Élaboration d'un outil de calcul et algorithme de scindage

Un algorithme de scindage, permettant d'analyser récursivement la structure des dépendances linéaires des polynômes g_μ par rapport aux mots w est présenté dans [1]. Il s'appuie sur des opérations matricielles élémentaires (addition, multiplication, concaténation), calquées sur les propriétés de dépendances linéaires des polynômes non commutatifs intervenant dans les équations linéaires. Il permet de tester l'identifiabilité de la série génératrice jusqu'au degré l .

Nous reprenons les notations du chapitre 8.

9.2.1 Calcul sans dérive

L'équation générique:

$$\mathcal{A} = 1 + A_1 \mathcal{A} + D_t \mathcal{A}$$

donne, g_μ étant un polynôme non commutatif,

$$\sum_{\mu} a^{\mu} g_{\mu} = 1 + \sum_{j=1}^m \sum_{\sigma} a_j z_j a^{\sigma} g_{\sigma} + \sum_{\nu} D_t(a^{\nu}) g_{\nu}$$

On obtient, pour une multiplicité μ non triviale:

$$g_{\mu} = \sum_{j=1}^m z_j l_{\mu \triangleright a_j} + \sum_{\nu} \langle \nu | \mu \rangle g_{\nu}$$

avec $g_{\epsilon} = 1$, et avec les notations

$$\mu \triangleright a_j = \begin{cases} \sigma & \text{si } a_j a^{\sigma} = a^{\mu} \\ 0 & \text{sinon} \end{cases}$$

9.2.2 Une entrée avec dérive

On devrait écrire $a^{\mu} = a_0^p a^{\nu}$. Dans ce cas, on note

$$g_{\mu} = g'_{/p}$$

On appelle alors profondeur de $g_{/p}^\nu$, l'entier p .

Le polynôme $g_{/p}^\nu$ est homogène de degré p en la lettre z_0 et homogène de degré $\text{deg}(\nu)$ en la lettre z_1 .

Les équations récurrentes vérifiées par les polynômes $g_{/p}^\nu$ deviennent alors:

$$g_{/p}^\nu = z_0 g_{/p-1}^\nu + z_1 g_{/p}^{\nu \circ a} + \sum_{\sigma} \langle \sigma | \nu \rangle l_{/p}^\sigma$$

9.2.3 L'algorithme

L'algorithme est proposé dans le cas d'une entrée avec dérive. Les polynômes g_μ sont combinaisons linéaires de mots w . L'algorithme proposé est un algorithme d'inversion qui calcule les mots w comme combinaisons linéaires des g_μ

Les équations récurrentes peuvent être réécrites sous la forme:

$$g_{/p}^\nu - \sum_{\sigma} \langle \sigma | \nu \rangle g_{/p}^\sigma = z_0 g_{/p-1}^\nu + z_1 g_{/p}^{\nu \circ a}$$

Puisque chacun des polynômes $g_{/p}^\nu$ est homogène de degré $\text{deg}(\nu) + p$, on obtient pour chaque degré n , une relation reliant récursivement les polynômes $g_{/p}^\nu$ de degré n à ceux de degré $n - 1$. Alors, le membre de gauche apparaît comme une combinaison linéaire de partitions de même taille $n = p + \text{taille}(\nu)$. Le membre de droite est une combinaison linéaire de partitions de taille $n - 1$. Cette remarque nous permet de mettre les partitions apparaissant en membre gauche sous forme triangulaire.

Le codage vectoriel et la traduction matricielle sont détaillés dans [1]. L'algorithme fournit la décomposition en facteurs premiers des déterminants Δ_l^1 .

Des exécutions de l'algorithme en MAPLE donnent les résultats suivants (sur Power-Mac G3, 266 Mhz):

nombre d'occurrences	profondeur	nombre de mots	temps
5	4	2^{10}	460 <i>secondes</i>
5	5	2^{11}	970 <i>secondes</i>

Remarquons que cet algorithme est générique. Il permet de décider pour tout mot w s'il est combinaison linéaire de polynômes g_μ .

Nous conjecturons que l'algorithme ne rend jamais de déterminant nul. Nous conjecturons donc que le problème de l'identification Algébrique est complètement résolu par cet algorithme.

9.3 Conclusion et perspectives

Le problème formel de l'Identification Algébrique a été traité par des méthodes de calcul non commutatif. On peut noter les remarques suivantes:

– **Les performances**

Pour certifier l'Identification, on doit pouvoir aller beaucoup plus loin dans les calculs des déterminants d'identification, en calcul numérique qu'en calcul formel. Le calcul numérique peut suffire pour valider la non nullité de déterminants, ces déterminants étant de gros entiers.

– **Systèmes causaux polynômiaux**

Le schéma d'identification peut être repris pour des systèmes causaux polynômiaux généralisés de la forme:

$$\dot{q} = f(q, a, a^{(1)}, \dots, a^{(n)})$$

– **Valeurs exactes ou non**

Notre identification est basée sur le principe que les valeurs des entrées/sorties et de leurs dérivées sont connues et exactes.

Un autre point de vue serait basé sur l'existence de bruit. Une méthode pourrait consister à faire une identification en boucle fermée, en utilisant un régulateur [3]. On sait que dans ce cas, il y a insensibilité au bruit. Plusieurs approches sont proposées dans le cas linéaire:

1. Trouver le “modèle de procédé” qui minimise l'écart entre le système en boucle fermée “réel” et le système en boucle fermée “simulé” et effectuer des validations statistiques ou temporelles (comparaisons simulation et résultats temps réel).
2. Utiliser des algorithmes d'identification en boucle ouverte manipulant des données filtrées.

On peut penser que certaines de ces approches peuvent trouver une bonne généralisation dans le cas non linéaire.

Les résultats obtenus et quelques perspectives sont développées dans [2].

Bibliographie

- [1] Hespel C., Jacob G., *On Algebraic Identification of Causal Functionals*, 10th International Conference on Formal Power Series and Algebraic Combinatorics, Toronto, June 14-19, 1998.
- [2] Hespel C., Jacob G., *On the Algebraic Identification Problem: results and perspectives*, CASC98, St Petersburg, 20-24 April 1998.
- [3] Landau I.D., *Identification et Commande de Systèmes*, Hermes, 1993.
- [4] Gessel I., Viennot G., *Binomial determinants, paths, and Hook length formulae*, *Advances in Mathematics*, vol.58, pp. 300-321, 1985.

Annexes : Principales publications

Annexe 1

Hespel C., *Approximation de séries formelles par des séries rationnelles*, RAIRO Inform.Théor., vol.18, n° 3, pp. 241-258, 1984.

Annexe 2

Hespel C., Jacob G., *Approximation of nonlinear systems by bilinear ones*, in "Algebraic and Geometric Methods in Nonlinear Control Theory", M.Fliess and M.Hazewinkel eds., D.Reidel Publishing Company, pp.511-520, 1986.

Annexe 3

Hespel C., Jacob G., *Calcul des approximations locales bilinéaires de systèmes analytiques*, RAIRO APII, vol. 23, pp. 331-349, 1989.

Annexe 4

Hespel C., Jacob G., *Approximation of nonlinear dynamic systems by rational series*, Theoret. Comput. Sciences, 79, pp.151-162, 1991.

Annexe 5

Hespel C., *Truncated bilinear approximants: Carleman, finite Volterra, Padé-type, geometrical and structural automata*, Lecture Notes in Control and Information Sciences, vol. 165, "Algebraic Computing in Control", pp. 265-278, Springer-Verlag, 1991.

Annexe 6

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

Annexe 7

Hespel C., Jacob G., *First steps towards exact algebraic identification*, Discrete Math., vol. 180, pp. 211-219, 1998.

Annexe 8

Hespel C., Jacob G., *On Algebraic Identification of Causal Functionals*, 10th International Conference on Formal Power Series and Algebraic Combinatorics, Toronto, june 14-19, 1998.

Annexe 1

Hespel C., *Approximation de séries formelles par des séries rationnelles*, RAIRO Inform.Théor., vol.18, n° 3, pp. 241-258, 1984.

APPROXIMATION DE SÉRIES FORMELLES PAR DES SÉRIES RATIONNELLES (*)

par Christiane HESPEL ⁽¹⁾

Communiqué par J. BERSTEL

Résumé. — *Étant donnée une série formelle en variables non commutatives sur un alphabet fini X , à coefficients dans un corps K , nous proposons une construction de série rationnelle approximante au sens de la valuation X -adique, et qui soit de rang minimal.*

Abstract. — *We consider a formal power series in non commutative variables over a finite alphabet X , with coefficients in a field K . We give a construction of a rational power series approximant in the sense of the X -adic valuation, which is of minimal rank.*

INTRODUCTION

La notion de série formelle en variables non commutatives a été introduite par M. P. Schutzenberger [28] en liaison avec divers sujets d'informatique théorique tels que les automates et les langages formels.

Rappelons qu'on appelle série formelle en variables non commutatives sur X , à coefficients dans un corps K , toute application :

$$s: X^* \rightarrow K,$$

qui associe à tout mot f de X^* , une multiplicité ou coefficient (s, f) dans K .

Dans la théorie des séries formelles, comme en théorie des langages, il est naturel de s'intéresser en priorité aux objets construits par des procédés finis tels que : expression rationnelle, K -automate de type fini, système d'équations algébriques.

Un certain nombre de problèmes issus de la théorie des langages tels que les langages avec coefficients d'ambiguïté, les problèmes arithmétiques de la théorie des langages [2], l'étude des processus stochastiques [12], ont recours à l'utilisation des séries formelles en variables non commutatives.

De même, les séries formelles sont un outil utile pour la résolution de problèmes combinatoires tels que l'énumération des graphes planaires [6], les

(*) Reçu en juin 1982, révisé en juin 1983.

(¹) I.N.S.A. Rennes, 20, avenue des Buttes-de-Coësmes, 35043 Rennes Cedex.

permutations et les réarrangements dans les monoïdes [4] et l'étude des systèmes dynamiques [10].

Deux principales familles de séries formelles ont été étudiées, (*voir* [27]) : les séries rationnelles [30] ainsi que leur sous-famille formée des séries reconnaissables [8] et les séries algébriques [5].

Les séries rationnelles ont été introduites et développées d'abord par M. P. Schutzenberger, montrant ainsi que certaines propriétés des séries rationnelles en une variable, trouvent leur bonne généralisation en variables non commutatives.

Dans un premier temps, M. P. Schutzenberger généralise le théorème de Kleene [20] qui établissait l'équivalence des langages reconnaissables et rationnels, en démontrant l'équivalence de la reconnaissabilité et de la rationalité des séries formelles propres [29]. Il montre ainsi le rôle fondamental joué, pour l'étude des séries formelles, par les représentations matricielles du monoïde libre, notamment les décompositions en représentations irréductibles, obtenant ainsi des résultats fins sur la croissance des coefficients [31].

M. Fliess caractérise, grâce aux matrices de Hankel, les séries rationnelles [9] puis celles à coefficients positifs [12].

D'autre part, G. Jacob généralise, à l'aide des représentations matricielles, la notion de boucle dans les automates finis [19], ce qui lui permet de résoudre des problèmes de décidabilité de la finitude de l'ensemble des coefficients des séries rationnelles.

C. Reutenauer caractérise les séries rationnelles par leur algèbre syntactique : une série est rationnelle si et seulement si son algèbre syntactique est de dimension finie [24]. De plus, il définit la notion de variété de séries rationnelles, généralisant ainsi les variétés de langages au sens de S. Eilenberg [7].

Enfin, certaines propriétés arithmétiques des séries rationnelles déjà étudiées dans le cas d'une variable [26, 1], peuvent être généralisées aux séries formelles à coefficients entiers en variables non commutatives [22, 19, 25].

La famille des séries algébriques a été introduite par M. P. Schutzenberger [28] : les séries algébriques sont définies comme étant les composantes de la solution d'un système d'équations algébriques propres.

Les langages algébriques apparaissent alors comme support de telles séries, qui sont solutions d'un système à coefficients entiers positifs ou nuls [23].

M. Nivat étudie systématiquement la nature de ces solutions [23] mais surtout, il montre que l'étude des séries algébriques dépend de façon fondamentale de l'étude des transductions rationnelles de séries formelles.

A la question : l'image par une transduction rationnelle (resp. algebrico-rationnelle) d'une série rationnelle (resp. algébrique) est-elle une série de même nature, M. Fliess répond partiellement [9] : il montre que si le semi-anneau des coefficients est, soit totalement ordonné de plus petit élément zéro, soit un corps ou un anneau de Krull valués complets, l'image par transduction rationnelle d'une série rationnelle, est rationnelle. Il établit, d'autre part que l'image par transduction algebrico-rationnelle projetée (resp. constructible projetée) d'une série algébrique projetée (resp. constructible projetée) est une série de même nature, et G. Jacob, grâce à la notion de transduction rationnelle régulée, répond à la question posée [16] : il montre que l'image par transduction rationnelle régulée (resp. algébrique régulée) d'une série rationnelle (resp. algébrique) est une série rationnelle (resp. algébrique).

De plus, en établissant la réciproque du théorème de Shamir, G. Jacob fournit une nouvelle caractérisation des séries algébriques [16].

Enfin, W. Wechler observe que ce résultat peut encore s'énoncer ainsi : le langage reconnu par un automate à pile est le support d'une série formelle algébrique et réciproquement [32]. Les résultats obtenus par l'utilisation des transductions rationnelles régulées peuvent être caractérisés tout aussi bien par des notions d'automate à pile : automate à pile « ε -nilpotent » [15], technique reprise et généralisée par W. Kuich dans la notion d'automate à plile « cycle-free » pour l'étude des familles abstraites de séries formelles, généralisant l'étude des « F.A.L. » en théorie des langages [21].

Mais l'étude des séries formelles en variables non commutatives est aussi en lien étroit avec l'algèbre différentielle, ce que montrent notamment les travaux de M. Fliess sur les systèmes analytiques [10, 14].

Le point de départ de cet article est le théorème suivant de M. Fliess [13], qui pose la question d'une approximation par des polynômes ou des séries rationnelles :

THÉORÈME Toute fonctionnelle causale analytique peut être uniformément approchée par des fonctionnelles régulières (i. e. dont la série génératrice est rationnelle).

Le propos de cet article est de rechercher une approximation de séries formelles, par des séries rationnelles au sens de la valuation X -adique, qui soit meilleure que celle offerte par des polynômes.

La trame de l'article est la suivante :

Après l'introduction des notations et des généralités (section 1), où nous rappelons en particulier la caractérisation des séries rationnelles par une propriété de leur rang, nous proposons un majorant du rang minimum d'un polynôme approchant une série formelle donnée à un ordre donné

(section 2-2). Nous montrons (section 3-3) que ce majorant peut être atteint et que, de plus, pour un ordre donné, il existe des séries formelles telles que le rang de toute série rationnelle approximante ne puisse être inférieur à celui des polynômes approchant ces séries.

Toutefois, nous pensons qu'en dépit de l'exemple étudié en section 2-3, il existe de nombreuses séries formelles pour lesquelles on puisse trouver un approximant rationnel de rang inférieur à celui des approximants polynomiaux et nous proposons des classes de telles séries (section 2-5).

De manière générale, nous calculons le rang minimal d'une série rationnelle approchant une série formelle donnée à un ordre donné (section 2-4) et nous en déduisons une construction de cet approximant rationnel (section 2-6).

I. RANG D'UNE SÉRIE RATIONNELLE

Nous rappelons quelques notations et résultats connus sur les séries rationnelles [17].

1. Séries rationnelles

Soit K un corps et X un alphabet fini de cardinal n ; une série formelle sur X est une application s du monoïde libre X^* dans K , et est notée comme une somme formelle :

$$s = \sum \{(s, f) f \mid f \in X^*\}.$$

Le coefficient du mot f dans la série s , est donc l'image (s, f) de f par l'application s , et le langage support de s est l'ensemble des mots de coefficient non nul.

Si nous notons $\text{Supp}(s)$ le support de s , et $X^{<n} = \{f \in X^* \text{ tels que, la longueur de } f, \text{ notée } |f|, \text{ soit inférieure à } n\}$, alors l'ordre de s est l'unique entier $\mathcal{O}(s)$ tel que :

$$\text{et } \begin{cases} \text{Supp}(s) \cap X^{<\mathcal{O}(s)} = \emptyset, \\ \text{Supp}(s) \cap X^{<\mathcal{O}(s)+1} \neq \emptyset. \end{cases}$$

Nous dirons qu'une série formelle est propre si son support ne contient pas le mot vide. L'ensemble des séries propres forme un idéal I de l'algèbre $K \ll X \gg$ des séries formelles, vérifiant :

$$\bigcap_n I^n = \{0\}$$

et la topologie I -adique (ou X -adique) ainsi définie sur $K \ll X \gg$ est séparée [17].

Les séries formelles reconnaissables sont les séries reconnues par un K -automate matriciel de dimension finie, i. e. telles qu'il existe :

N , entier strictement positif;

γ , vecteur colonne de K^N ;

λ , covecteur de dimension N sur K ;

μ , représentation de X^* dans le monoïde des matrices carrées de dimension N sur K ;

vérifiant $s = \sum \{ \lambda(\mu f) \gamma \cdot f / f \in X^* \}$ [8].

Étant donnée une série formelle propre, il y a équivalence entre la reconnaissabilité et la rationalité [29].

Dans le cas particulier où $X = \{t\}$, alors une série rationnelle $s = \sum_{i \in \mathbb{N}} s_i t^i$ peut s'écrire comme le quotient de deux polynômes.

2. Condition sur le rang pour qu'une série formelle soit rationnelle

Notons $K \langle X \rangle$ le sous-ensemble de $K \ll X \gg$ formé des polynômes ou séries de support fini.

En remarquant que toute action à gauche $*$ de X^* sur un K -espace vectoriel E peut être prolongée en une action de $K \langle X \rangle$ à gauche sur E , E se trouve donc muni d'une structure de $K \langle X \rangle$ -module à gauche. On appellera K -module sériel à gauche sur X , la donnée d'un triplet $\mathcal{E} = (E, C, l)$ tel que :

E est un $K \langle X \rangle$ module à gauche;

C est un élément de E appelé vecteur initial;

l est une application linéaire de E dans K .

\mathcal{E} définit alors la série formelle $s_{\mathcal{E}} = \sum \{ l(f \cdot C) / f \in X^* \}$.

Une série formelle reconnaissable est donc une série qui peut être définie par un K -module sériel de dimension finie.

A une série formelle s , associons sa matrice de Hankel $H(s)$, indicée par $X^* \times X^*$, définie par $H(s)_{u,v} = (s, uv)$ pour u et v dans X^* .

Les colonnes de $H(s)$ s'identifient canoniquement aux séries résiduelles à droite de s , et dualement, les lignes de $H(s)$ s'identifient canoniquement aux séries résiduelles à gauche de s .

Nous définissons le rang de s comme égal au rang de $H(s)$, noté $\text{rg}(s)$. Cette notion de rang permet de caractériser les séries rationnelles : [11].

THÉORÈME Une condition nécessaire et suffisante pour qu'une série $s \in K \ll X \gg$ soit rationnelle est qu'elle soit de rang fini N . Elle est alors reconnue par un K -automate matriciel :

$$A = (N, \gamma, \lambda, \mu) \text{ de dimension } N.$$

De plus, on peut trouver des mots $\{g_i\}^N$ et $\{d_j\}^N$ de longueur inférieure à N , tels que l'application χ de X^* dans $K^{N \times N}$ définie par :

$$(\chi^{(f)})_{i,j} = (s, g_i f d_j),$$

vérifie $\chi^{(f)} = \chi^{(1)} \mu(f)$ et que $\chi^{(1)}$ soit inversible.

Une version algorithmique de ce théorème a été présentée par A. Cardon et M. Crochemore [3].

II. RECHERCHE D'UNE SÉRIE RATIONNELLE APPROXIMANTE DE RANG MINIMAL

1. Énoncé du problème

Soient s une série formelle, k et r deux entiers naturels.

Notons $v_k(s) = \{ \text{séries } g \text{ rationnelles telles que } \mathcal{O}(g-s) \geq k \}$ et $\text{App}(k, r, s) = \{ \text{séries } g \text{ de } v_k(s) \text{ telles que } \text{rg}(g) \leq r \}$.

Alors, nous poserons les problèmes du calcul du plus petit entier $r = \rho(s, k)$ tel que $\text{App}(k, r, s)$ soit non vide, de la construction d'une série rationnelle approximante g de rang $\rho(s, k)$ et de la comparaison de $\rho(s, k)$ avec le rang d'un polynôme approximant.

2. Un majorant M du plus petit rang d'un approximant polynomial

Notons $E(l)$ la partie entière d'un nombre réel l .

PROPOSITION 1 : Soit n le cardinal de l'alphabet fini X . Un majorant du rang minimum d'un polynôme g approchant s à l'ordre k est :

$$M = \frac{2 - n^{E((k+1)/2)} - n^{E(k/2)}}{1 - n}.$$

Preuve : Ordonnons les lignes et les colonnes de $H(s)$, par les mots qui les indicent rangés par longueur croissante et, pour une longueur donnée, dans l'ordre lexicographique.

Notons $H_{\leq p, \leq q}(s)$ la matrice de Hankel de s restreinte aux lignes d'indices u tels que $|u| \leq p$ et aux colonnes d'indices v tels que $|v| \leq q$.

Comme g doit approcher s à l'ordre k , ($\mathcal{O}(s-g) \geq k$), alors les matrices de Hankel de s et g restreintes aux mots de longueur inférieure à k , doivent coïncider.

Choisissons g égal au polynôme de degré $k-1$, obtenu en tronquant s . Deux cas sont à distinguer selon la parité de k .

Premier cas : Si $k = 2p + 1$; alors $\text{rg}(g) = \text{rg}(H_{\leq 2p, \leq 2p}(s))$.

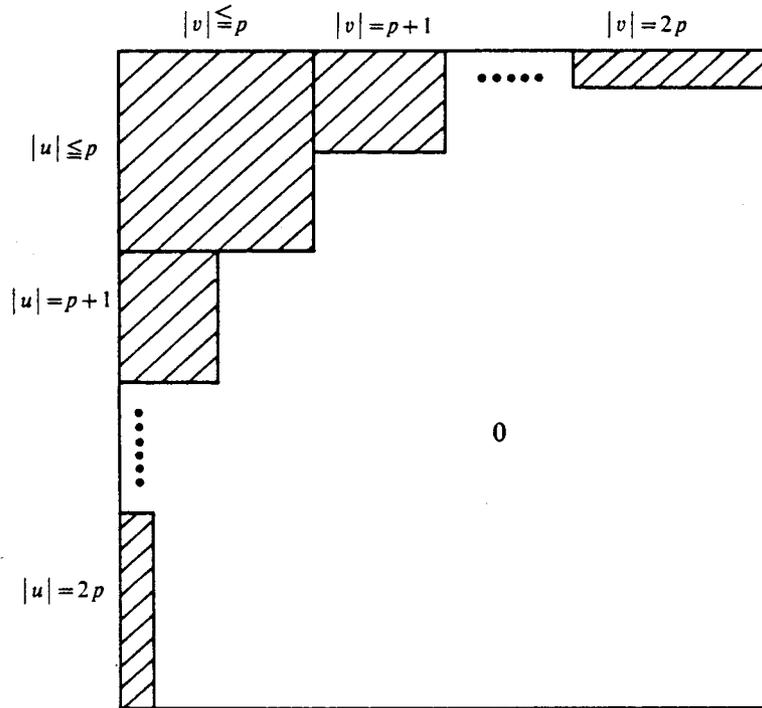


Schéma 1. — $H_{\leq 2p, \leq 2p}(g)$.

Donc, $\text{rg}(g) \leq \text{rg}$ des $(1 + \dots + n^p)$ premières colonnes + rg des $(n^{p+1} + \dots + n^{2p})$ dernières colonnes.

Soit :

$$\text{rg}(g) \leq (1 + \dots + n^p) + (1 + \dots + n^{p-1}).$$

Deuxième cas : Si $k = 2p$; alors $\text{rg}(g) = \text{rg}(H_{\leq 2p-1, \leq 2p-1}(s))$.

Donc, $\text{rg}(g) \leq \text{rg}$ des $(1 + \dots + n^{p-1})$ premières colonnes + rg des $(n^p + \dots + n^{2p-1})$ dernières colonnes.

Soit :

$$\text{rg}(g) \leq (1 + \dots + n^{p-1}) + (1 + \dots + n^{p-1}).$$

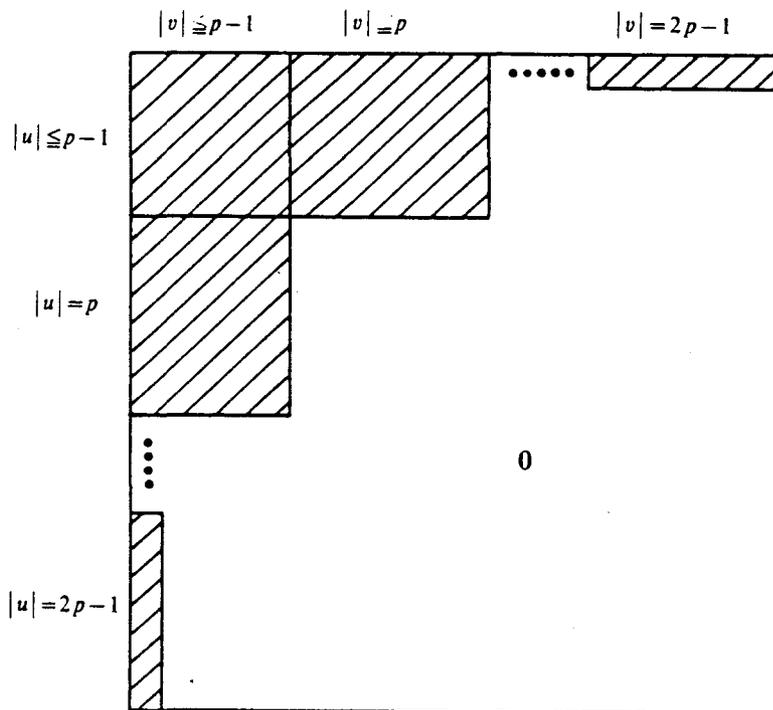


Schéma 2. — $H_{\leq 2p-1, \leq 2p-1}(g)$.

3. Existence de série formelle telle que M soit à la fois le plus petit rang d'un approximant polynomial et rationnel, à l'ordre k

Deux cas sont à distinguer suivant la parité de k .

Premier cas : $k = 2p + 1$.

Soit la série s_k définie par $(s_k, f) = 1$ pour $f = u^2$ et $|f| \geq 2p$, $(s_k, f) = 0$ sinon.

Représentons la matrice de Hankel de s_k , restreinte aux mots f tels que $|f| < k$ (schéma 3).

Il est clair que, quel que soit le prolongement des colonnes de ce tableau, le rang ne pourra être en aucun cas, inférieur à

$$(1 + \dots + n^p) + n^{p-1} + \dots + n + 1 = M.$$

Deuxième cas : $k = 2p$.

Soit la série s_k définie par $(s_k, f) = 1$ pour $|f| \geq 2p - 1$ et $f = xu^2$ (tels que $x \in X$ $u \in X^*$), $(s_k, f) = 0$ sinon.

Représentons la matrice de Hankel de s_k , restreinte aux mots f , tels que $|f| < k$ (schéma 4).

Comme dans le premier cas, on voit que le rang ne pourra être inférieur à $(1 + \dots + n^{p-1}) + n^{p-1} + \dots + n + 1 = M$.

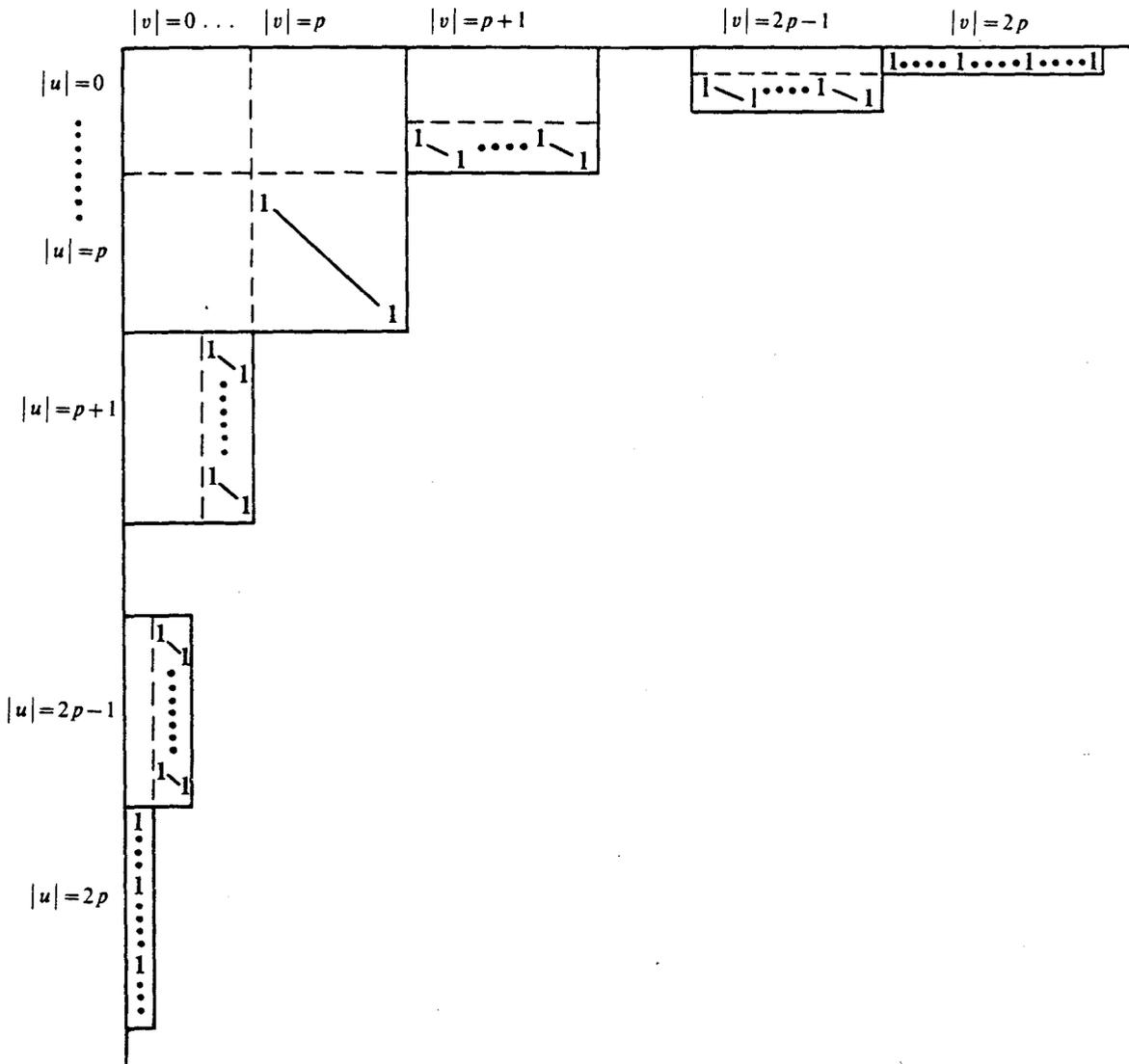


Schéma 3. — $H_{<2p+1, <2p+1}(s_{2p+1})$.

4. Calcul du plus petit rang d'un approximant rationnel

Énonçons quelques remarques utiles pour la recherche d'un approximant rationnel.

PROPOSITION 2 : Soit $p \in \mathbb{N}^*$; si les matrices $H_{\leq p, \leq p}(s)$ et $H_{\leq p, \leq p+1}(s)$ ont même rang r , alors il existe une et une seule série rationnelle g vérifiant :

- (i) $\text{rg}(g) = r$;
- (ii) $\mathcal{O}(s-g) \geq 2p+2$.

De plus, toute série rationnelle g vérifiant $\mathcal{O}(s-g) \geq 2p+1$ est de rang $\geq r$.

Preuve : Soit \mathcal{D} l'espace vectoriel engendré par les colonnes de $H_{\leq p, \leq p}(s)$ et soit D une base de \mathcal{D} .

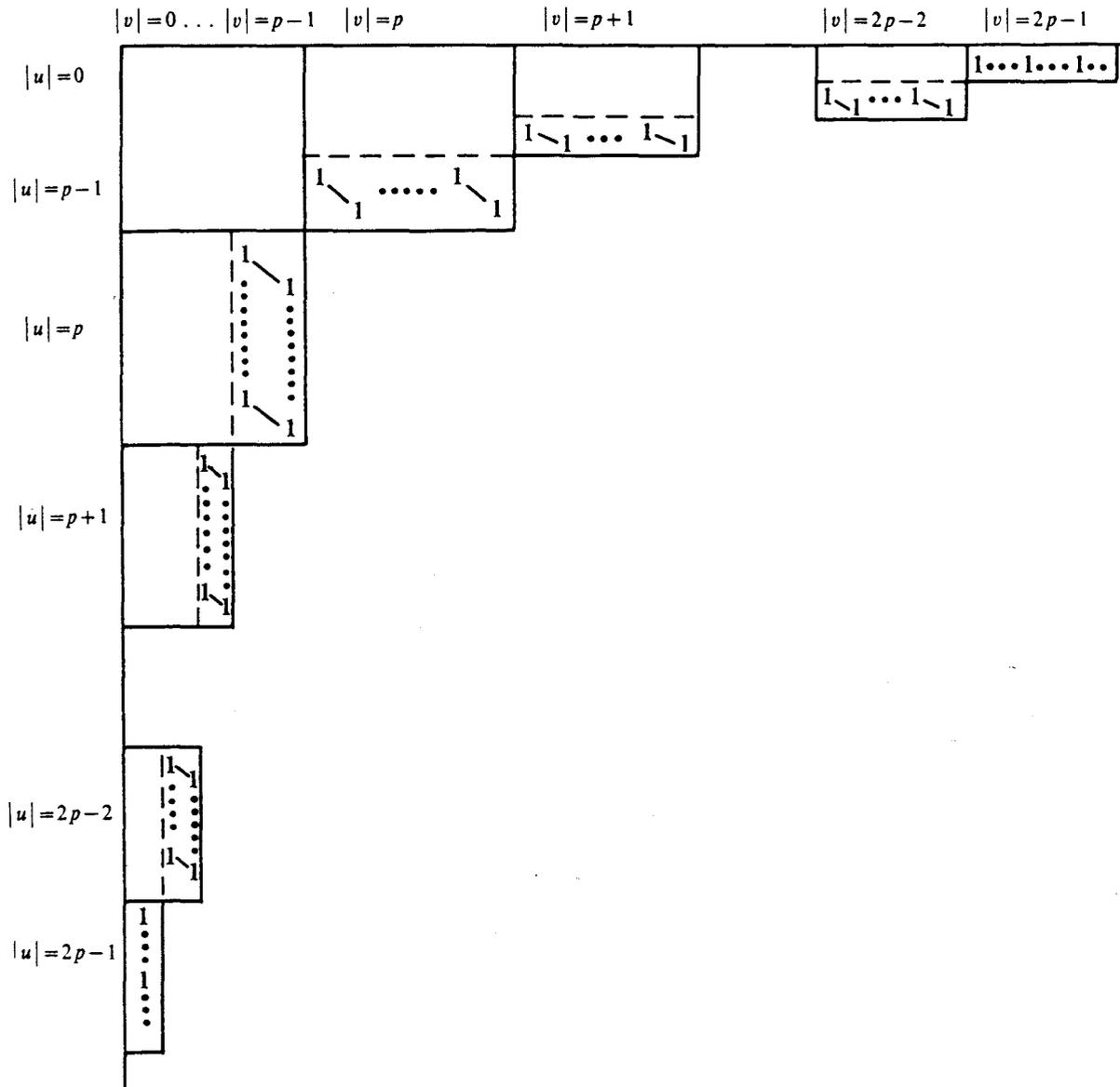


Schéma 4. — $H_{<2p, <2p}(S_{2p})$.

$\forall x \in X$, et $\forall v \in X^p$, la colonne d'indice xv de $H_{\leq p, \leq p+1}(s)$ est combinaison linéaire unique des vecteurs de D .

Cette construction se prolonge en une unique action à gauche de X^* sur \mathcal{D} , donc en un unique X -module sériel à gauche.

La série reconnue par ce X -module sériel vérifie :

$$H_{\leq p, \leq p+1}(s) = H_{\leq p, \leq p+1}(g).$$

COROLLAIRE 1 : Si les colonnes d'une matrice de Hankel, d'indice v_j tel que $|v_j|=j$ sont toutes combinaisons linéaires des colonnes d'indices $v_{<j}$ tel que

$|v_{<j}| < j$, alors les colonnes d'indice $v_{>j}$ tel que $|v_{>j}| > j$, sont aussi combinaisons linéaires des colonnes d'indice $v_{<j}$.

COROLLAIRE 2 : Si certaines colonnes d'une matrice de Hankel, d'indice v_{j_0} tel que $|v_{j_0}| = j$ sont combinaisons linéaires des colonnes d'indice $v_{<j}$ tel que $|v_{<j}| < j$, alors les colonnes d'indice xv_{j_0} sont combinaisons linéaires des colonnes d'indice $xv_{<j}$, quel que soit x dans X .

COROLLAIRE 3 : Si des colonnes d'une matrice de Hankel d'indice v_{j_0} tel que $|v_{j_0}| = j$, sont combinaisons linéaires de certaines colonnes d'indice $v_{\leq j_1}$ tel que $|v_{\leq j_1}| \leq j$, alors les colonnes d'indice xv_{j_0} sont combinaisons linéaires des colonnes d'indice $xv_{\leq j_1}$, quel que soit x dans X .

PROPOSITION 3 : Soit :

$$j_i(s) = \text{rang}(H_{\leq k-1-i, \leq i}(s)) - \text{rang}(H_{\leq k-1-i, \leq i-1}(s))$$

pour $i = 1$ à $k-1$ et soit $j_0(s) = \text{rang}(H_{\leq k-1, \leq 0}(s))$.

Alors, le rang minimal d'une série rationnelle g approchant s à l'ordre k est $\rho(s, k) = j_0(s) + \dots + j_{k-1}(s)$.

Preuve : Explicitons une construction de $H_{\leq \infty, \leq k-1}(g)$ obtenue par prolongement de la matrice de Hankel de s restreinte aux mots de longueur inférieure ou égale à $k-1$.

Choix des colonnes libres

Par analogie aux notations adoptées pour $H_{\leq p, \leq q}(s)$, notons :

$\mathcal{B}_{\leq k-1, \leq 0}$ le système de colonne libre extraite de $H_{\leq k-1, \leq 0}(s)$ et soit $\mathcal{B}_{\leq k-1-i, \leq 0}$ sa restriction à $H_{\leq k-1-i, \leq 0}(s)$;

$\mathcal{B}_{\leq k-2, \leq 1}$ le système de colonnes libres extraites de $H_{\leq k-2, \leq 1}(s)$, qui avec $\mathcal{B}_{\leq k-2, \leq 0}$ engendre $H_{\leq k-2, \leq 1}(s)$ et soit $\mathcal{B}_{\leq k-2-i, \leq 1}$ sa restriction à $H_{\leq k-2-i, \leq 1}(s)$.

$\mathcal{B}_{\leq k-3, \leq 2}$ le système de colonnes libres extraites de $X * \mathcal{B}_{\leq k-3, \leq 1}$, qui avec $\mathcal{B}_{\leq k-3, \leq 0}$ et $\mathcal{B}_{\leq k-3, \leq 1}$ engendrent $H_{\leq k-3, \leq 2}(s)$ ⁽²⁾.

[On note $X * \mathcal{B}_{\leq p, \leq q}$ le système de colonnes, d'indice xv où $x \in X$ et v indice des colonnes $\mathcal{B}_{\leq p, \leq q}$, extrait de $H_{\leq p, \leq q+1}(s)$].

$\mathcal{B}_{\leq 0, \leq k-1}$ le système de colonnes libres extraites de $X * \mathcal{B}_{\leq 0, \leq k-2}$ qui, avec $\mathcal{B}_{\leq 0, \leq 0}, \dots, \mathcal{B}_{\leq 0, \leq k-2}$ engendrent $H_{\leq 0, \leq k-1}(s)$ ⁽²⁾.

⁽²⁾ Ceci est possible d'après le corolaire 3.

Et notons $\mathcal{C}_{\leq k-i-1, \leq i}$ le système des colonnes restantes de chaque bloc $H_{\leq k-1-i, =i}(s)$.

Alors, à l'ordre des colonnes près, on peut représenter la restriction de la matrice de Hankel de s , aux mots de longueur inférieure ou égale à $k-1$, par le schéma suivant :

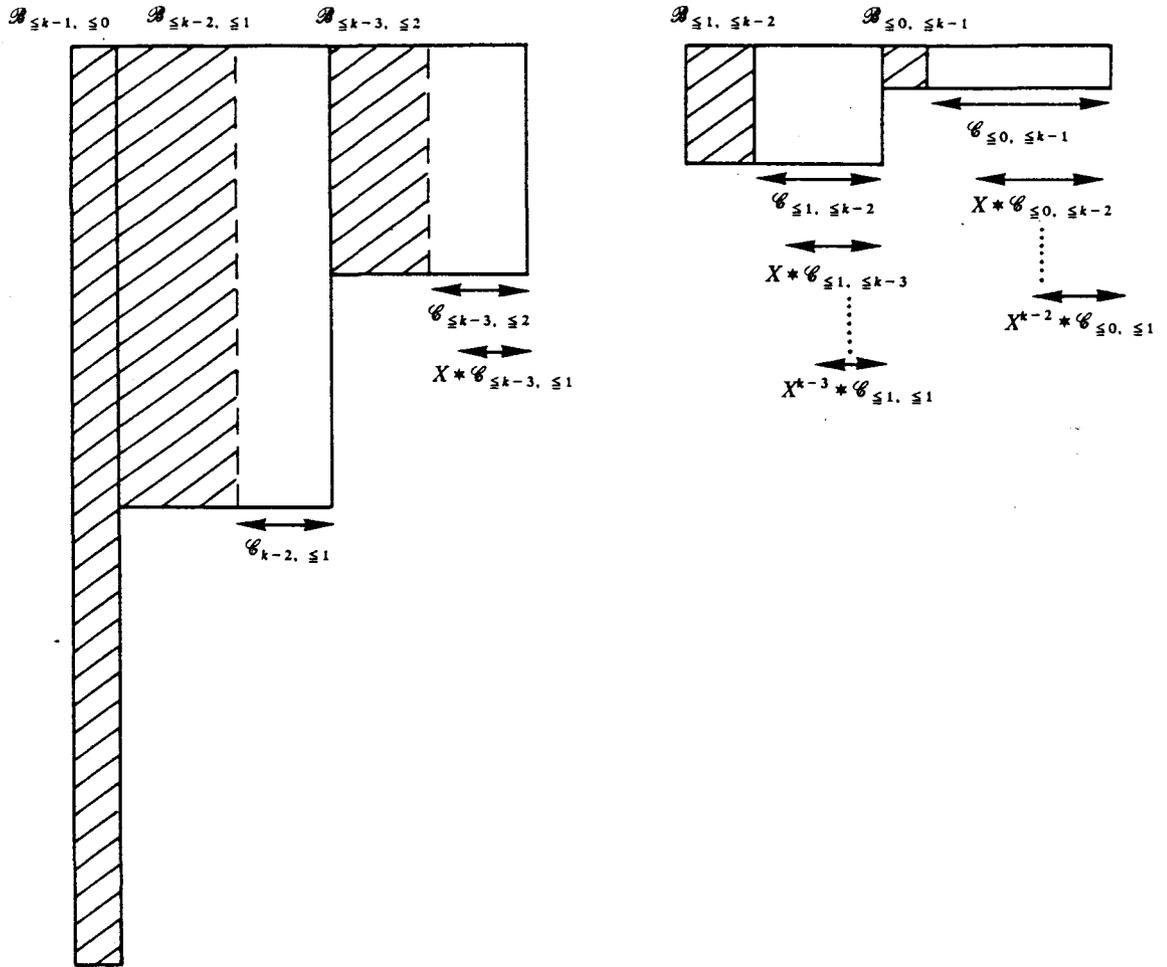


Schéma 5. — Restriction de $H_{\leq k-1, \leq k-1}(s)$ aux mots de longueur $\leq k-1$.

Remarques (cf. corollaires 2 et 3) : Par construction, on a des chaînes d'inclusions :

$$\begin{aligned} \mathcal{B}_{\leq k-2, \leq 1} &\subset X * \mathcal{B}_{\leq k-2, \leq 0}, \\ \mathcal{B}_{\leq k-3, \leq 2} &\subset X * \mathcal{B}_{\leq k-3, \leq 1} \subset X^2 * \mathcal{B}_{\leq k-3, \leq 0}, \\ &\dots \\ \mathcal{B}_{\leq 0, \leq k-1} &\subset X * \mathcal{B}_{\leq 0, \leq k-2} \subset \dots \subset X^{k-1} * \mathcal{B}_{\leq 0, \leq 0}. \end{aligned}$$

De même :

$$\begin{aligned} \mathcal{C}_{\leq k-3, \leq 2} \supset X * \mathcal{C}_{\leq k-3, \leq 1}, \\ \mathcal{C}_{\leq 0, \leq k-1} \supset X * \mathcal{C}_{\leq 0, \leq k-2} \supset \dots \supset X^{k-2} * \mathcal{C}_{\leq 0, \leq 1}. \end{aligned}$$

Par ailleurs, $\mathcal{B}_{\leq k-1, \leq 0}$ est réduit à la première colonne si $\mathcal{O}(s) \leq k-1$. $\mathcal{B}_{\leq 0, \leq k-1}$ est au plus formé d'une colonne et donc $\mathcal{C}_{\leq 0, \leq k-1}$ est formé de $n^{k-1} - 1$ colonnes au moins et $\text{rang}(\mathcal{B}_{\leq k-1-i, \leq i}) = j_i$.

Montrons que l'on peut « border » le tableau représenté par le schéma 5 en prolongeant les colonnes, de manière à préserver les relations de dépendance linéaire entre les colonnes.

Si $E \subset F$ notons par $E - F$, le complémentaire de F dans E .

— Nous supposons d'abord que $\mathcal{B}_{\leq 0, \leq k-1}$ n'est pas vide.

(1) Prolongeons l'unique colonne de $\mathcal{B}_{\leq 0, \leq k-1}$ par des zéros de manière à définir $\mathcal{B}_{\leq 1, \leq k-1}$.

Alors, $\mathcal{C}_{\leq 0, \leq k-1} \div X * \mathcal{C}_{\leq 0, \leq k-2}$ est prolongé dans $H_{\leq 1, \leq k-1}(g)$ par linéarité.

Par conséquent :

$$X * \mathcal{B}_{\leq 0, \leq k-2} = (\mathcal{B}_{\leq 0, \leq k-1}) \cup (\mathcal{C}_{\leq 0, \leq k-1} - X * \mathcal{C}_{\leq 0, \leq k-2})$$

est prolongé lui aussi dans $H_{\leq 1, \leq k-1}(g)$.

Ce qui signifie que $\mathcal{B}_{\leq 1, \leq k-2}$ est prolongé dans $H_{\leq 2, \leq k-2}(g)$ pour former $\mathcal{B}_{\leq 2, \leq k-2}$.

On poursuit le prolongement des colonnes libres : on suppose donc que :

$\mathcal{B}_{\leq 1, \leq k-2}$ a été prolongé pour former $\mathcal{B}_{\leq 2, \leq k-2}, \dots$, etc.;

$\mathcal{B}_{\leq k-3, \leq 2}$ a été prolongé pour former $\mathcal{B}_{\leq k-2, \leq 2}$.

Alors, $\mathcal{C}_{\leq k-3, \leq 2} \div X * \mathcal{C}_{\leq k-3, \leq 1}$ est prolongé dans $H_{\leq k-2, \leq 2}(g)$ par linéarité.

Par conséquent, $X * \mathcal{B}_{\leq k-3, \leq 1}$ est prolongé dans $H_{\leq k-2, \leq 2}(g)$.

Ce qui signifie que $\mathcal{B}_{\leq k-2, \leq 1}$ est prolongé pour former $\mathcal{B}_{\leq k-1, \leq 1}$.

Enfin, puisque $\mathcal{B}_{\leq k-1, \leq 1}$ a été construit, alors on prolonge :

$$\mathcal{C}_{\leq k-2, \leq 1} - X * \mathcal{C}_{\leq k-2, \leq 0} = \mathcal{C}_{\leq k-2, \leq 1}$$

dans $H_{\leq k-1, \leq 1}(g)$ par linéarité.

Par conséquent $X * \mathcal{B}_{\leq k-2, \leq 0}$ est prolongé dans $H_{\leq k-1, \leq 1}(g)$, ce qui signifie que $\mathcal{B}_{\leq k-1, \leq 0}$ est prolongé pour former $\mathcal{B}_{\leq k, \leq 0}$.

(2) Complétons la construction :

Le prolongement de $\mathcal{C}_{\leq k-2, \leq 1}$ détermine les prolongements de $X * \mathcal{C}_{\leq k-3, \leq 1}, \dots, X^{k-2} * \mathcal{C}_{\leq 0, \leq 1}$.

Il en résulte le prolongement de $\mathcal{C}_{\leq k-3, \leq 2}$ ce qui détermine les prolongements de $X * \mathcal{C}_{\leq k-4, \leq 2}, \dots, X^{k-3} * \mathcal{C}_{\leq 0, \leq 2}$, etc.

Il en résulte le prolongement de $\mathcal{C}_{\leq 1, \leq k-2}$ ce qui détermine le prolongement de :

$$X * \mathcal{C}_{\leq 0, \leq k-2}.$$

Il en résulte le prolongement de $\mathcal{C}_{\leq 0, \leq k-1}$.

Cette construction se réitère en prolongeant de nouveau l'unique colonne de $\mathcal{B}_{\leq 1, \leq k-1}$ par des zéros pour former $\mathcal{B}_{\leq 2, \leq k-1}$, puis en reproduisant la construction précédente.

— Supposons qu'il existe i_0 tel que : $\mathcal{B}_{\leq 0, \leq k-1}, \dots, \mathcal{B}_{\leq i_0, \leq k-1-i_0}$ soient vides et $\mathcal{B}_{\leq i_0+1, \leq k-2-i_0}$ non vide.

Alors, on peut prolonger :

$$\mathcal{C}_{\leq i_0, \leq k-1-i_0} \cdot X * \mathcal{C}_{\leq i_0, \leq k-2-i_0} = X * \mathcal{B}_{\leq i_0, \leq k-2-i_0},$$

dans $H_{\leq i_0+1, \leq k-1-i_0}(g)$, par linéarité, ce qui exprime que $\mathcal{B}_{\leq i_0+1, \leq k-2-i_0}$ est prolongé dans $H_{\leq i_0+2, \leq k-2-i_0}(g)$ pour former $\mathcal{B}_{\leq i_0+2, \leq k-2-i_0}$.

La construction se poursuit alors comme dans le premier cas.

En conclusion :

— On peut « border » le tableau représenté par $H_{\leq k-1, \leq k-1}(s)$ restreinte aux mots de longueur inférieure ou égale à $k-1$, en prolongeant les colonnes de manière à préserver les relations de dépendance linéaire, ce prolongement pouvant être réitéré indéfiniment.

— D'autre part, ou bien $j_{k-1} = 0$; donc $\text{rang}(g) = \text{rang}(H_{\leq \infty, \leq k-1}(g))$ (cf. corollaire 1), ou bien $j_{k-1} = 1$; d'après le choix du prolongement de $\mathcal{B}_{\leq 0, \leq k-1}$, alors $\text{rang}(g) = \text{rang}(H_{\leq \infty, \leq k-1}(g))$.

Par conséquent, $\text{rang}(g) = \text{rang}(H_{\leq k-1, \leq k-1}(g)) = j_0 + \dots + j_{k-1}$.

5. Comparaison des approximants polynomiaux et rationnels

On a vu dans le paragraphe 3 que, quel que soit k , on sait construire une série s_k telle que les approximants polynomiaux et rationnels aient le même rang minimal.

Or, pour k donné, la série s_k a la particularité qu'aucun prolongement des colonnes de sa matrice de Hankel, restreinte aux mots f tels que $|f| < k$, ne

fasse « diminuer le rang » (cf. schéma 3); en effet, chaque bloc $H_{\leq k-1-i, =i}(s)$ est « indépendant » : chaque colonne d'un bloc $H_{\leq k-1-i, =i}(s)$ n'est pas combinaison linéaire (excepté la combinaison nulle) de colonnes d'autres blocs $H_{\leq k-1-j, =j}(s)$ (pour $i > j$) restreintes aux indices de lignes dans $X^{\leq k-1-i}$.

Par contre, il semble que, pour un ordre $k' = k + 2$ par exemple, il existe un approximant rationnel de s_k de rang inférieur au rang d'un approximant polynomial.

Par exemple :

Les séries « épaisses » :

ϵ_k est une série épaisse pour l'ordre k si et seulement si k étant donné, ϵ_k vérifie :

$$\text{rg}(H_{\leq E((k-1)/2), \leq E((k-1)/2)}(\epsilon_k)) = 1 + n + \dots + n^{E((k-1)/2)}.$$

Il en résulte que :

$$\begin{aligned} \text{rg}(H_{\leq E((k-1)/2), \leq E((k-1)/2)}(\epsilon_k)) &= \text{rg}(H_{\leq E((k-1)/2), \leq E((k-1)/2)+1}(\epsilon_k)) \\ &= 1 + n + \dots + n^{E((k-1)/2)}. \end{aligned}$$

Alors, d'après la proposition 2, il existe g série rationnelle vérifiant :

$$\text{rg}(g) = \frac{1 - n^{E((k-1)/2)+1}}{1 - n}$$

et approchant ϵ_k à l'ordre k .

Les séries « i -quasi épaisses » :

q_k est une série i -quasi épaisse pour l'ordre k si et seulement si k étant donné, ϵ_k vérifie :

$$\text{rg}(H_{\leq E((k-1)/2)-i, \leq E((k-1)/2)+i}(q_k)) = \text{rg}(H_{\leq E((k-1)/2)-i, \leq E((k-1)/2)+i+1}(q_k)).$$

Alors, d'après le corollaire 1, il existe g rationnelle de rang $\leq \text{rg}(H_{\leq k-1, \leq E((k-1)/2)+i}(q_k))$, approchant q_k à l'ordre k .

6. Construction explicite d'un approximant rationnel de rang minimal

* Dans un premier temps, on fait le choix des « colonnes libres » de $H_{\leq k-1, \leq k-1}(s)$ restreinte aux mots de longueur inférieure ou égale à $k-1$, comme il est dit dans la démonstration de la proposition 3.

Soit $(\mathcal{B}_{\leq k-1-i, \leq i})_{0 \leq i \leq k-1}$ ce système de colonnes libres; les autres colonnes étant notées $(\mathcal{C}_{\leq k-1-i, \leq i})_{0 \leq i \leq k-1}$.

* Dans un second temps, on prolonge éventuellement ces colonnes dans $H_{\leq k-1, \leq k-1}(g)$, comme il est dit dans la démonstration de la proposition 3 :

On cherche si il existe i_0 tel que :

$$i_0 \in [0, k-2] \quad \text{et} \quad \begin{cases} \mathcal{B}_{\leq i_0, \leq k-1-i_0} & \text{soit vide,} \\ \mathcal{B}_{\leq i_0+1, \leq k-2-i_0} & \text{non vide;} \end{cases}$$

→ ou bien j_0 n'existe pas; alors on prolonge $\mathcal{B}_{\leq 0, \leq k-1}$ par des zéros pour former $\mathcal{B}_{\leq 1, \leq k-1}$, puis on prolonge $\mathcal{C}_{\leq 0, \leq k-1} - X * \mathcal{C}_{\leq 0, \leq k-2}$ par linéarité, ce qui fournit un prolongement de $\mathcal{B}_{\leq 1, \leq k-2}$ en $\mathcal{B}_{\leq 2, \leq k-2}$, etc.

$\mathcal{B}_{\leq k-2, \leq 1}$ prolongé en $\mathcal{B}_{\leq k-1, \leq 1}$ puis on prolonge $\mathcal{C}_{\leq k-2, \leq 1} - X * \mathcal{C}_{\leq k-2, \leq 0}$ par linéarité, ce qui fournit un prolongement de $\mathcal{B}_{\leq k-1, \leq 0}$ en $\mathcal{B}_{\leq k, \leq 0}$ et on réitère en prolongeant de nouveau $\mathcal{B}_{\leq 1, \leq k-1}$ par des zéros pour former $\mathcal{B}_{\leq 2, \leq k-1}$.

→ ou bien i_0 existe; alors on prolonge $\mathcal{C}_{\leq i_0, \leq k-1-i_0} - X * \mathcal{C}_{\leq i_0, \leq k-2-i_0}$ par linéarité ce qui donne un prolongement de $\mathcal{B}_{\leq i_0+1, \leq k-2-i_0}$ en $\mathcal{B}_{\leq i_0+2, \leq k-2-i_0}$,

et on poursuit la construction comme dans le premier cas.

On obtient ainsi $j_0 + j_1 + \dots + j_{k-1}$ colonnes libres extraites de $H_{\leq k-1, \leq k-1}(g)$, et engendrant cette matrice.

Ces colonnes libres ont un déterminant non nul, que l'on peut écrire comme un tableau indexé par les lignes g_i et les colonnes d_j , de dimension $j_0 + j_1 + \dots + j_{k-1}$.

D'après les travaux de G. Jacob [18] on sait que cela permet de définir entièrement la série rationnelle g associée.

BIBLIOGRAPHIE

1. J. BERSTEL, *Sur les poles et le quotient de Hadamard de séries N-rationnelles*, C. R. Acad. Sc. Paris, serie A, 272, 1971, p. 1079-1081.
2. J. BERSTEL, *Contribution à l'étude des propriétés arithmétiques des langages formels*, Thèse, Université de Paris-VII, 1972.
3. A. CARDON, M. CROCHEMORE, *Détermination de la représentation standard d'une série reconnaissable*, R.A.I.R.O., Informatique théorique, vol. 14, n° 4, 1980, p. 371-379.
4. P. CARTIER, D. FOATA, *Problèmes combinatoires de commutation et réarrangements*, Lect. Notes Math., n° 85, Springer-Verlag, Berlin, 1969.
5. N. CHOMSKY, M. P. SCHUTZENBERGER, *The Algebraic Theory of Context-free Languages*, in *Computer Programming and Formal Systems*, P. BRAFFORD and D. HIRSCHBERG, éd., North Holland, 1963, p. 118-161.

6. R. CORI, J. RICHARD, *Énumération des graphes planaires à l'aide des séries formelles en variables non commutatives*, Discrete Math., t. 2, 1972, p. 115-162.
7. S. EILENBERG, *Automata, Languages and Machines*, vol. B, New York, Academic Press, 1976.
8. M. FLIESS, *Séries reconnaissables, rationnelles et algébriques*, Bull. Soc. Math. Fr., vol. 94, 1970, p. 231-239.
9. M. FLIESS, *Sur certaines familles de séries formelles*, Thèse, Université de Paris-VII, 1972.
10. M. FLIESS, *Sur la réalisation des systèmes dynamiques bilinéaires*, C. R. Acad. Sc. Paris, série A, 277, 1973, p. 923-926.
11. M. FLIESS, *Matrices de Hankel*, J. Maths. Pur. Appl., vol. 53, 1974, p. 197-222.
12. M. FLIESS, *Série rationnelles positives et processus stochastiques*, Ann. Inst. H. Poincaré, sect. B, vol. XI, n° 1, 1975, p. 1-21.
13. M. FLIESS, *Un outil algébrique : les séries formelles non commutatives*, Mathematical System Theory, G. MARCHESINI et S. K. MITTER, éd., Lect. Notes Économ. Math. Syst., vol. 131, Springer-Verlag, 1976, p. 122-148.
14. M. FLIESS, *Une approche algébrique du développement fonctionnel des solutions d'équations différentielles non linéaires forcées*, Astérisque 75-76, Soc. Math. Fr., 1980, p. 95-103.
15. G. JACOB, *Représentations et substitutions matricielles dans la théorie algébrique des transductions*, Thèse, Université de Paris-VII, 1975.
16. G. JACOB, *Sur un théorème de Shamir*, Inform. Contr., vol. 27, n° 3, 1975, p. 218-261.
17. G. JACOB, *Langages, automates, séries formelles*, Publication 107, Université des Sciences et Techniques de Lille, 1978.
18. G. JACOB, *Réalisation des systèmes réguliers (ou bilinéaires) et séries génératrices non commutatives*, Séminaire d'Aussois, RCP 567, in Outils et modèles mathématiques pour l'automatique, l'analyse des systèmes et le traitement du signal, C.N.R.S. Landau, 1980.
19. G. JACOB, *Un théorème de factorisation des produits d'endomorphismes de K^n* , J. Algebra, 63, 1980, p. 389-412.
20. S. C. KLEENE, *Representation of Events in Nerve Nets and Finite Automata*, in Shannon, C. E., MCCARTHY, J. éd., Automata studies, Princeton, N.J. 1956, p. 3-40.
21. W. KUICH, *An Algebraic Characterization of Some Principal Regulated Rational Cones*, J. Comput. System Sc., vol. 25, 1982, p. 377-401.
22. K. LAMECHE, *Extension d'un théorème de G. Cantor à des séries rationnelles en variables non commutatives*, C. R. Acad. Sc. Paris, série A, 272, 1971, p. 296-298.
23. M. NIVAT, *Transductions des langages de Chomsky*, Ann. Inst. Fourier, vol. 18, 1968, p. 339-455.
24. C. REUTENAUER, *Séries formelles et algèbres syntactiques*, J. Algebra, vol. 66, 1980, p. 448-483.
25. C. REUTENAUER, *Propriétés arithmétiques de séries rationnelles et ensembles denses*, Acta Arith. vol. 39, 1981, p. 133-144.
26. C. REUTENAUER, *Sur les éléments inversibles de l'algèbre de Hadamard des séries rationnelles*, Bull. Soc. Math. Fr., vol. 110, 1982, p. 225-232.
27. A. SALOMAA et M. SOITTOLA, *Automata-Theoretic Aspects of Formal Power Series*, Springer-Verlag, 1977.

28. M. P. SCHUTZENBERGER, *Un problème de la théorie des automates*, Séminaire Dubreil-Pisot (Algèbre et théorie des Nombres), 13^e année, n° 3, Inst. H.-Poincaré, 1960.
29. M. P. SCHUTZENBERGER, *On the Definition of a Family of Automata*, Inform. Contr., vol. 4, 1961, p. 245-270.
30. M. P. SCHUTZENBERGER, *On a Theorem of Jungen*, Proc. Amer. Math. Soc., vol. 13, 1962, p. 885-890.
31. M. P. SCHUTZENBERGER, *Finite Counting Automata*, Inform. Contr., vol. 5, 1962, p. 91-107.
32. W. WECHLER, *An Acceptor for Algebraic Power Series*, Elektronische Informationsverarbeitung und Kybernetik, EIK 15, 1979, p. 611-619.

Annexe 2

Hespel C., Jacob G., *Approximation of nonlinear systems by bilinear ones*, in “Algebraic and Geometric Methods in Nonlinear Control Theory”, M.Fliess and M.Hazewinkel eds., D.Reidel Publishing Company, pp.511-520, 1986.

APPROXIMATION OF NONLINEAR SYSTEMS BY BILINEAR ONES

C. HESPEL*, G. JACOB**

* I.N.S.A 20 avenue des Buttes de Coësmes
35043 RENNES CEDEX

** Laboratoire de Recherche en Informatique Fondamentale
(LA 369) - Université de LILLE I
59655 VILLENEUVE D'ASCQ CEDEX

ABSTRACT

Given an analytic system, we compute a bilinear system of minimal dimension which approximates it up to order k (i.e. the outputs of these two systems have the same Taylor expansion up to order k).

For that purpose, we use a noncommutative formal power series called the generating series of the system. Let s be the series, and y the output of the analytic system : we notice that the Taylor expansion of y up to order k in $t=0$ can be obtained from the coefficients of the words of length not greater than k in the series s . As rational series are characteristic of finite dimensional bilinear systems, the problem is reduced to the following : build a rational series g , which is an approximation of s up to order k (i.e. the coefficients of the words of length not greater than k in g and s are identical), and which is of minimal rank. Then with g we associate a bilinear system, which is a solution to our problem.

INTRODUCTION

Several methods may be used for computing the input-output behaviour of a control system : transfer functions, functional expansions, especially Volterra series, and generating power series.

All these descriptions allow to approach the following question : Is it possible to give an approximation of the input-output behaviour of any system by that of a more elementary system, like for instance a linear system ?

The transfer function of a single input system can be used to build a linear approximation of this system, via the Padé approximants [1,2].

In the same way, the use of Padé approximants in two commutative variables allows to build approximants of a system when its output is an analytic function of several inputs [1]. This method is connected with the continued fraction expansions, and the orthogonal polynomials [1,13]

However, in general, it is impossible to approximate nonlinear systems by linear ones.

Another method based on the Volterra functional expansions can be developed for determining approximations of analytic systems, by computing its first Volterra kernels (References of [6]). The computation of the generating power series of the system can be used as an effective tool in order to compute recursively the Volterra kernels up to great orders [6]. Then we are faced with a new problem : the realization of this truncated Volterra series [12].

The formal power series in several noncommutative variables are the most natural tool when dealing with functional expansions (cf. M. FLIESS). It is impossible to approximate any nonlinear functional by a linear system (i.e. a rational series in one variable). But, on the other hand, it is possible to approximate any nonlinear functional by a bilinear system (i.e. by a polynomial or by a rational series in several noncommutative variables).

Our approach is the following : given an analytic system of generating series s , we compute a reduced bilinear system (i.e. of minimal state-space dimension), whose generating series g is identical with s up to order k . Consequently, the Taylor expansion of the output of the bilinear approximant system up to order k is equal to the Taylor expansion of the output of the given system.

Our approximant is a generalization of Padé approximants for several noncommutative variables.

1. METHOD

1.1. Preliminaries and notations

Consider the analytic system

$$(S) \quad \begin{cases} \dot{q}(t) = A_0(q) + \sum_{i=1}^m u_i(t) A_i(q) \\ y(t) = h(q) \end{cases}$$

State q belongs to a finite dimensional \mathbb{R} -analytic manifold Q . Vector fields A_0, A_1, \dots, A_m and output function h are analytic and defined in a neighbourhood of initial state $q(0)$.

Inputs u_1, \dots, u_m are assumed to be piece-wise continuous.

As M. FLIESS showed [5,6], the input-output behaviour of an analytic system can be coded by a noncommutative formal power series called the generating series of the system. For that purpose, with vector fields A_0, A_1, \dots, A_m he associates an alphabet $X = \{x_0, x_1, \dots, x_m\}$. The generating series is noted

$$s = \sum_w (s, w) w \quad \text{where } w = x_{j_1} \dots x_{j_k} \\ \text{for } k \geq 0 \text{ and } x_{j_i} \in X.$$

(w is called word and its length $|w|$ is equal to k).

The output, y of system (S) may be expressed in the following way :

$$y(t) = \sum_w (s,w) \int_0^t \delta w$$

where

$$\int_0^t \delta w = \int_0^t u_{j_1}(\tau_1) \int_0^{\tau_1} u_{j_2}(\tau_2) \dots \int_0^{\tau_{k-1}} u_{j_k}(\tau_k) d\tau_k \dots d\tau_2 d\tau_1$$

for $w = x_{j_1} \dots x_{j_k}$.

The generating series may be rewritten in the form

$$s = h/q(o) + \sum_{v \geq 0} \sum_{j_0, \dots, j_v=0}^m A_{j_0} \dots A_{j_v} h/q(o) x_{j_v} \dots x_{j_0}$$

(the bar $/q(o)$ indicates evaluation at $q(o)$) which is called the fundamental formula [6].

A generating power series is known to correspond to a finite-dimensional bilinear system if and only if it is rational. On the other hand, we know that a series s is rational if and only if the rank of its Hankel matrix $H(s)$ is finite. (by Hankel matrix $H(s)$, we mean the infinite matrix whose rows and columns are indexed by the words of X^* and whose element, with indexes u and v , is equal to (s, uv)). This rank is also the dimension of the state-space of the reduced system [3,4,10].

1.2. Rational approximation

Using the techniques introduced by M. FLIESS and C. REUTENAUER, in a paper developing the theory of PICARD-VESSIOT [7], we show that the successive derivatives $y(o), y^{(1)}(o), \dots, y^{(k)}(o)$ of the output y of system (S) can be obtained from the coefficients (s,w) of the generating series s , associated with the words w of length $\leq k$. In other words, the Taylor expansion of y up to order k depends only on these coefficients.

The method developed here is the following : given a generating series s , we build a rational series g (therefore associated with a bilinear system) which is identical with s for the words of length $\leq k$, and whose rank is minimal, according to diagram 1

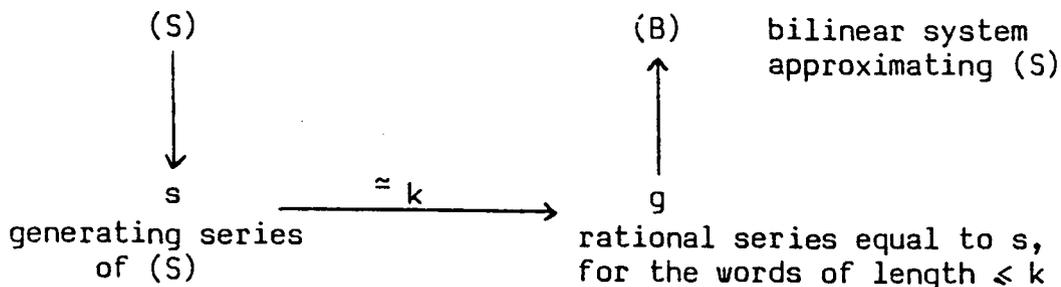


diagram 1 : the method

1.3. Development of the method

We begin by computing the Hankel matrix of s , restricted to the words of length $\leq k$, which we denote

$$H_{\leq k}(s)$$

and we extend it, in order to obtain the Hankel matrix of a rational series g whose rank is minimal. For a detailed explanation the reader should consult [9].

The Hankel matrix of s , restricted to the rows indexed by the words u of length $\leq p$ and to the columns indexed by the words v of length $\leq q$ is noted

$$H_{\leq p, \leq q}(s) .$$

We choose a set of linearly independent column vectors from $H_{\leq k}(s)$, in the following way :

- . we choose the single column vector of $H_{\leq k, \leq 0}(s)$.

- . we choose (if possible) some linearly independent column vectors indexed by words of length = 1 which, joined with the single column of $H_{\leq k-1, \leq 0}(s)$, generates matrix $H_{\leq k-1, \leq 1}(s)$.

- . we choose (if possible) some linearly independent column vectors indexed by words of length = 2 which, joined with the already chosen columns restricted to $H_{\leq k-2, \leq 1}(s)$, generates matrix $H_{\leq k-2, \leq 2}(s)$.

...etc...

The algorithm terminates when there is no more new linearly independent column in some $H_{\leq k-i_0, \leq i_0}(s)$.

Let us call $(C_{v_j})_{j=1 \dots r}$ the family of these column vectors, indexed by words v_j . (diagram 2).

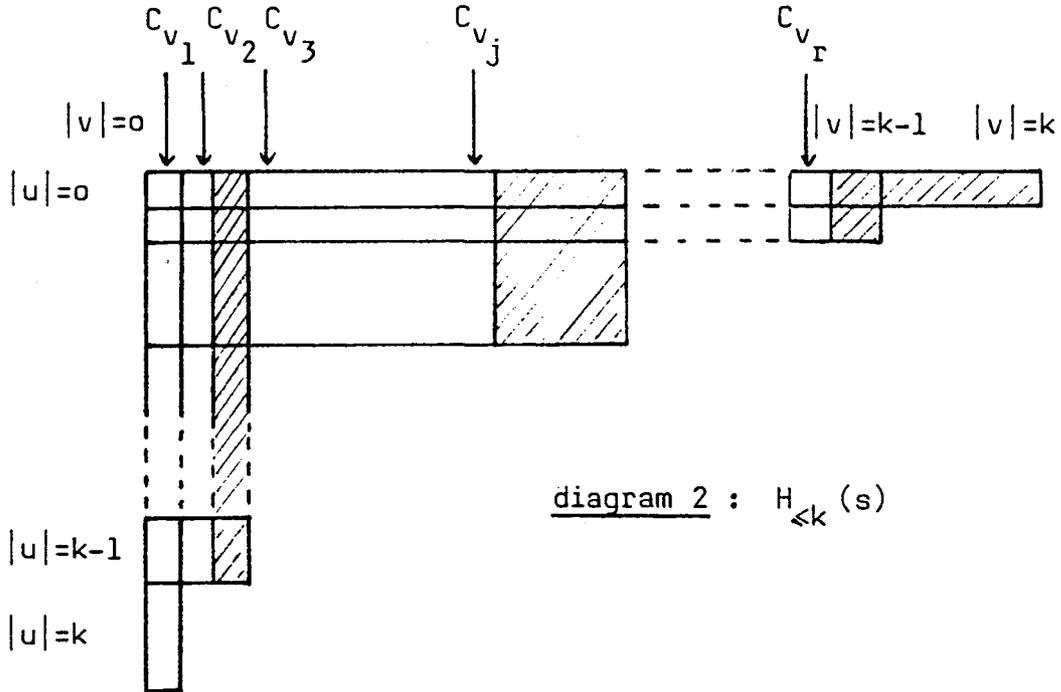
They form a basis of a state-space whose dimension r is minimal.

Then, we get the bilinear system (B) [10] :

$$\dot{q} = \left(\sum_{i=0}^m u_i(t) M_i \right) q(t)$$

by computing the matrices M_i given by :

$$M_i C_{v_j} = C_{x_i v_j}$$



2. PRACTICAL COMPUTATION

2.1. Direct application of the method

Consider the nonlinear differential equation relating the current excitation $i(t)$ and the voltage $v(t)$ across a capacitor [6] :

$$(E_1) \quad \frac{dv}{dt} + k_1 v(t) + k_2 v^2(t) = i(t)$$

i.e.
$$\begin{cases} \dot{q}(t) = A_0(q) + i(t) A_1(q) \\ v(t) = q(t) \end{cases}$$

with

$$\begin{cases} a = -k_1 q(o) - k_2 q(o)^2 \\ b = -k_1 - 2k_2 q(o) \\ c = -2k_2 \end{cases} \quad \text{and} \quad \begin{cases} A_0 = a \frac{d}{dq} \\ A_1 = \frac{d}{dq} \end{cases} .$$

We notice that the fundamental formula

$$s = q(o) + \sum_{v>0} \sum_{j_0, \dots, j_v=0,1} A_{j_0} \dots A_{j_v} q/o \times j_v \dots \times j_0$$

allows us to write the relation

$$\forall n \in \mathbb{N} \quad A_1^n A_0 = a A_1^{n+1} + nb A_1^n + \frac{n(n-1)}{2} c A_1^{n-1}$$

in the following form

$$\forall n \in \mathbb{N} \quad C_{x_0} x_1^n = a C_{x_1^{n+1}} + nb C_{x_1^n} + \frac{n(n-1)}{2} c C_{x_1^{n-1}}$$

where $C_{x_1^n}$ is a column vector of $H(s)$.

Thus, family $(C_{x_1^n})_{n \in \mathbb{N}}$ is a basis of $H(s)$.

If $k = 2p$ then the minimal dimension of the state-space is $p+1$,

$$M_0 = \begin{pmatrix} 0 & 0 & 0 & \dots & 0 & 0 \\ a & b & c & & & 0 \\ 0 & a & 2b & & & \vdots \\ 0 & 0 & a & & & \vdots \\ \vdots & \vdots & \vdots & \frac{(p-1)(p-2)}{2} c & & \vdots \\ \vdots & \vdots & \vdots & (p-1)b & & \frac{p(p-1)}{2} c \\ 0 & 0 & 0 & \dots & a & pb \end{pmatrix} = (C_{x_0} x_1^n) \text{ expressed in basis } (C_{x_1^n})$$

$$M_1 = \begin{pmatrix} 0 & \dots & 0 & 0 \\ 1 & & & \\ 0 & \cdot & & \\ \vdots & \cdot & \cdot & \\ \vdots & & & \\ 0 & & & 1 & 0 \end{pmatrix} = (C_{x_1^{n+1}}) \text{ expressed in basis } (C_{x_1^n})$$

and the bilinear system is given by

$$\begin{cases} \dot{\underline{q}} = (M_0 + i(t) M_1) \underline{q}(t) \\ y(t) = \lambda \underline{q}(t) \end{cases} \quad \text{where} \quad \lambda = (q(0) \ 1 \ 0 \ \dots \ 0) \\ \underline{q}(0) = \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

2.2. The truncated automaton

Let us take local coordinates in a neighbourhood of $q(0)$. The vector fields are given by :

$$A_\ell(q) = \sum_{j=1}^N \theta_\ell^j(q_1, \dots, q_N) \frac{\partial}{\partial q_j} \quad \text{for } \ell = 0, 1, \dots, m.$$

For the sake of simplicity, θ_ℓ^j are assumed to be polynomials in q_1, \dots, q_N . Let $\frac{\partial}{\partial q_j} = D_j$.

As a series is known to be rational if and only if it is recognized by a finite \mathbb{R} -automaton [11], then, with the given system (S), we associate

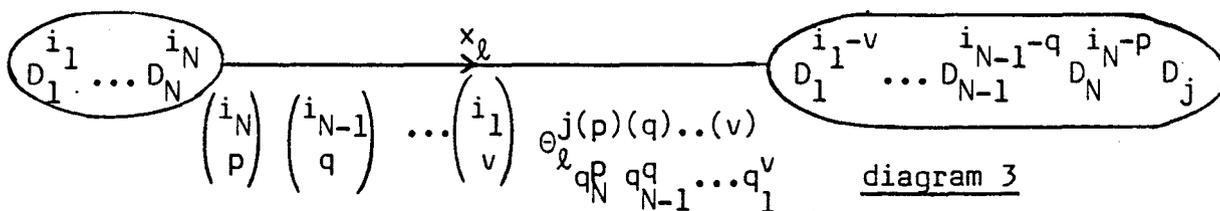
a \mathbb{R} -automaton whose states are the vector fields $D_1^{i_1} D_2^{i_2} \dots D_N^{i_N}$. In order to build this automaton, we compute the product of these vector fields and A_ℓ :

$$D_1^{i_1} D_2^{i_2} \dots D_N^{i_N} A_\ell = \sum_{j=1}^N \sum_{p=0}^{d_{j,N,\ell}} \sum_{q=0}^{d_{j,N-1,\ell}} \dots \sum_{v=0}^{d_{j,1,\ell}} \binom{i_N}{p} \binom{i_{N-1}}{q} \dots \binom{i_1}{v}$$

$$\theta_\ell^{j(p)(q)\dots(v)} D_1^{i_1-v} \dots D_{N-1}^{i_{N-1}-q} D_N^{i_N-p} D_j$$

where $d_{j,k,\ell}$ is the degree of θ_ℓ^j in q_k .

Every term of the sum on the right-hand side of this relation can be connected with one transition of the automaton for letter x_ℓ (diagram 3)



The initial state is vector field I .

Every state $D_1^{i_1} \dots D_N^{i_N}$ is a final state whose weight is equal to $D_1^{i_1} \dots D_N^{i_N} \circ h/q(o)$.

We shall call minimal order index of state $D_1^{i_1} \dots D_N^{i_N}$, the length of the shortest successful path labelled by a word $D_1^{i_1} \dots D_N^{i_N}$ that gets through this state. Then, the number of states whose minimal order index is $\leq k$, gives us a dimension (not necessarily minimal) of the state-space.

Consider the Duffing equation

$$(E_2) \quad y'' + ay' + b^2y + cy^3 = u_1(t)$$

$$\text{i.e.} \quad \begin{cases} \dot{q}(t) = A_0(q) + u_1(t) A_1(q) \\ y(t) = q_1(t) \end{cases}$$

$$\text{with} \quad \begin{cases} F = -aq_2 - b^2q_1 - cq_1^3 \\ A_0 = F \frac{\partial}{\partial q_2} + q_2 \frac{\partial}{\partial q_1} = F D_2 + q_2 D_1 \\ A_1 = \frac{\partial}{\partial q_2} = D_2 \end{cases}$$

We obtain the actions of A_0 and A_1 on states $D_1^j D_2^k$:

$$\begin{aligned}
 D_1^j D_2^k A_1 &= D_1^j D_2^{k+1} \\
 D_1^j D_2^k A_0 &= F D_1^j D_2^{k+1} + \binom{j}{1} F' D_1^{j-1} D_2^{k+1} + \binom{j}{2} F'' D_1^{j-2} D_2^{k+1} \\
 &\quad + \binom{j}{3} F''' D_1^{j-3} D_2^{k+1} - ak D_1^j D_2^k + q_2 D_1^{j+1} D_2^k + k D_1^{j+1} D_2^{k-1} .
 \end{aligned}$$

These actions can be represented by the transitions from states $D_1^j D_2^k$ produced by letters x_0 and x_1 . We thus determine the automaton-cell. (diagram 4).

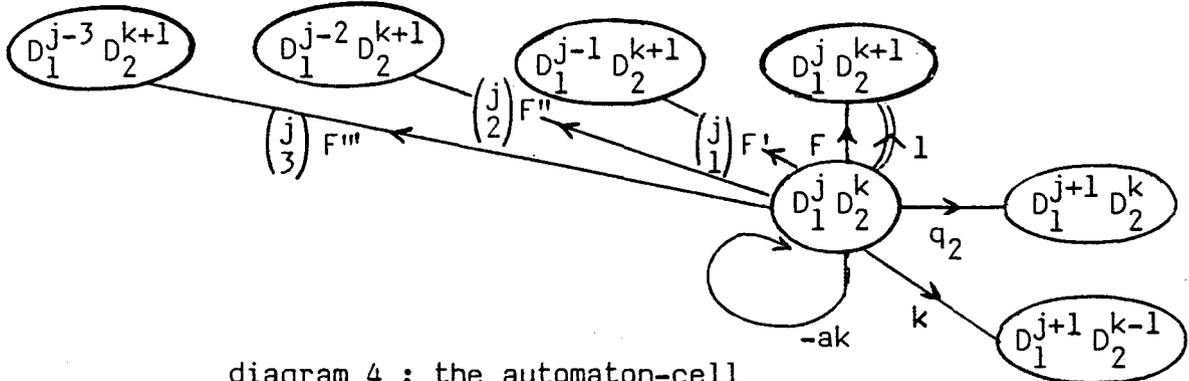


diagram 4 : the automaton-cell

Then we get the truncated automaton for $k = 7$ (diagram 5)

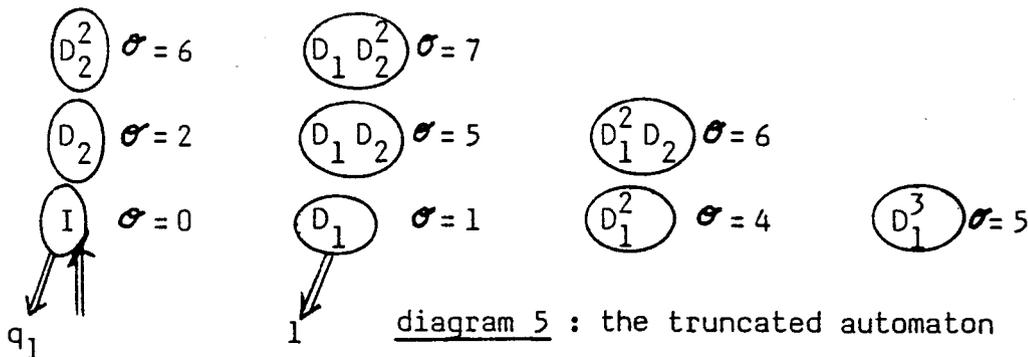


diagram 5 : the truncated automaton

For $k = 7$, 9 states are enough to compute the approximant. (the minimal order index σ is written in the diagram, for every state). We shall see that the dimension, computed by the method of the truncated automaton, is not necessarily minimal.

2.3. A quotient \mathbb{R} -automaton of the truncated automaton (cf. [10])

We use section 2.2., and compute the minimal rank.

Let us go back to the Duffing equation (E_2).

The method developed in 2.1 leads us to compute the rank of the matrix M defined as follows : (for $k = 7$)

- . its rows are indexed by the 9 states selected in 2.2.
- . its columns are indexed by the words of length ≤ 4
- . for state $D_1^j D_2^k$ and word w , the corresponding term is given by

$$D_1^j D_2^k w = D_1^j D_2^k A_{j_1} A_{j_2} \dots A_{j_n} \text{ oh} \quad \text{with } w = x_{j_1} x_{j_2} \dots x_{j_n} .$$

Thus, M can be written :

	1	x_0	$x_1 x_0$	x_0^2	$x_1 x_0^2$	x_0^3	$x_1 x_0^3$	x_0^4
I	q_1	q_2	1	F	-a	$-aF + q_2 F'$	$a^2 + F'$	$F(a^2 + F') + q_2(q_2 F'' - aF')$
D_1	1	0	0	F'	0	$q_2 F'' - aF'$	F''	$F'(a^2 + F') + FF'' + q_2(q_2 F''' - aF'')$
D_2	0	1		-a		$a^2 + F'$	0	$-a(a^2 + 2F') + 2q_2 F''$
D_1^2		0		F''		$q_2 F''' - aF''$	F'''	$F''(a^2 + 3F') + F'''(F - aq_2)$
$D_1 D_2$				0		F''	0	$-2aF'' + 2q_2 F'''$
D_2^2				0		0		$2F''$
D_1^3				F'''		$-aF'''$		$3F''^2 + F'''(a^2 + 4F')$
$D_1^2 D_2$				0		F'''		$-2aF'''$
$D_1 D_2^2$	0	0	0	0	0	0	0	$2F'''$

The rows indexed by states $D_1 D_2^2$ and $D_1^2 D_2$ are linear combinations of the other rows. The matrix, whose rank is equal to 7, is associated with a 7-state-automaton. The same result can be obtained, with a computer, by the direct application of the method (as in 2.1.)

3. CONCLUSION (related to the Duffing equation)

Table 1

Order of approximation k	2	3	4	5	6	7
Minimal dimension of the state-space	2	3	4	5	6	7
calculating time (msec) ⁺ (according to method 2.1) * algorithmic routine * conversational mode	2500	7300	31900	15600	30000	57000
dimension (according to method 2.2)	3	3	4	6	8	9
dimension (by taking a polynomial instead of a rational series)	4	6	10	14	22	30

+ MULTICS-MACSYMA

REFERENCES

- [1] C. BREZINSKI, 'Padé-type approximation and general orthogonal polynomials', *INSM 50*, Birkhäuser.
- [2] J. DELLA DORA, 'Quelques notions sur les approximants de Padé', in *Outils et Modèles Mathématiques pour l'Automatique, l'Analyse des systèmes et le traitement du signal*, vol. 2, ed. du CNRS, 1982, p. 203-224.
- [3] M. FLIESS, 'Matrices de Hankel', *J. Maths. Pur. Appl.*, vol. 53, 1974, p. 197-222.
- [4] M. FLIESS, 'Un outil algébrique : les séries formelles non commutatives', in *Mathematical Systems Theory (G. Marchesini and S.K. Mitter, Eds.)*, Lect. Notes Econom. Math. Syst., vol. 131, Springer-Verlag, 1976, p. 122-148.
- [5] M. FLIESS, 'Fonctionnelles causales non linéaires et indéterminées non commutatives', *Bull. Soc. Math. France*, 109, 1981, P. 3-40.
- [6] 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, p. 554-570.
- [7] M. FLIESS, C. REUTENAUER, 'Théorie de Picard-Vessiot des systèmes réguliers (ou bilinéaires)', *Colloque Nat. CNRS-RCP567, Belle-ile sept. 1982*, in *Outils et Modèles Mathématiques pour l'Automatique l'Analyse des systèmes et le traitement du signal*, CNRS, 1983.
- [8] J. GILEWICZ, 'Approximants de Padé', *Lecture Notes in Mathematics* n° 667, Springer-Verlag.
- [9] C. HESPEL, 'Approximation de séries formelles par des séries rationnelles', *RAIRO, Informatique Théorique*, vol. 18, n° 3, 1984, P. 241-258.
- [10] G. JACOB, 'Réalisation des systèmes réguliers (ou bilinéaires) et séries génératrices non commutatives', *Séminaire d'Aussois, RCP 567*, in *Outils et Modèles Mathématiques pour l'Automatique, l'Analyse des systèmes et le traitement du signal*, CNRS Landau, 1980.
- [11] M.P. SCHUTZENBERGER, 'On the definition of a family of automata', *Inform. Contr.* vol. 4, 1961, p. 245-270.
- [12] SMITH, KUSZTA, BAILEY, 'Mode identification of bilinear systems', *Int. J. of Control*, 37, n° 7, 1983, p. 943-957.
- [13] H. WALL, 'Continued fractions', *Van Nostrand*.

Annexe 3

Hespel C., Jacob G., *Calcul des approximations locales bilinéaires de systèmes analytiques*, RAIRO APII, vol. 23, pp. 331-349, 1989.

Calcul des approximations locales bilinéaires de systèmes analytiques (*)

C. HESPEL ⁽¹⁾, G. JACOB ⁽²⁾

Résumé/Abstract

Etant donné un système analytique, nous proposons un algorithme qui calcule explicitement un système bilinéaire de rang minimal, l'approchant à un ordre k donné. Le système approximant obtenu est fourni sous forme générique, paramétrée par l'état initial.

Cet algorithme s'appuie sur des calculs de séries en variables non commutatives : il consiste à construire une série rationnelle g dont les coefficients coïncident avec ceux de la série s génératrice du système analytique, pour tous les mots de longueur inférieure ou égale à k .

La mise en œuvre est effectuée par un codage des mots de longueur $\leq k$, permettant un traitement aisé des matrices de Hankel associées à s et g . On associe alors à la série rationnelle g , un système bilinéaire tel que les développements de Taylor des sorties du système analytique donné et du système bilinéaire, coïncident jusqu'à l'ordre k . Nous donnons ici une version de cet algorithme dans le langage MACSYMA ainsi qu'un exemple accompagné de l'étude de ses points singuliers.

A new algorithm is presented for the explicit computation of a minimal rank bilinear system which approximates a given analytic system up to order k . We give this approximant system under generic form, with the initial state as a parameter.

The algorithm is based on noncommutative series computation : call s the generating series of the analytic system ; then a rational series g is constructed, whose coefficients are equal to those of s , for words of length not greater than k . These words of length not greater than k are encoded, which allows to deal more easily with the Hankel matrices of s and g . Then a bilinear system is associated to

(*) Reçu en mars 1987.

(¹) INSA 20 avenue des Buttes de Coësmes, 35043 Rennes Cedex.

(²) Laboratoire de Recherche en Informatique Fondamentale (LA 369), Université de Lille I, 59655 Villeneuve d'Ascq Cedex.

g, whose output has the same Taylor expansion as the output of the analytic system up to order *k*. A MACSYMA version of the algorithm is given, followed by an example with a study of its singular points.

Mots clés/Keywords

Approximants de Padé ; Calcul formel ; Réalisation minimale ; Séries génératrices ; Systèmes analytiques ; Systèmes bilinéaires.

Introduction

Plusieurs techniques de représentation ont été proposées pour étudier localement le comportement entrée-sortie des systèmes dynamiques, permettant de caractériser le comportement de ces systèmes indépendamment de toute représentation d'état.

Il s'agit notamment du calcul des fonctions de transfert, des développements fonctionnels (principalement les séries de Volterra) [5], et enfin des séries génératrices non commutatives [4].

Cette diversité de représentations d'une même fonctionnelle entrée-sortie permet de les comparer et, dans certains cas, de calculer les représentations qui satisfont à des critères de choix ou d'optimalité.

Prolongeant notre publication [8], nous proposons dans cet article un algorithme (implémenté dans le langage MACSYMA) qui calcule, pour tout entier positif *k* et tout système analytique (*S*) (linéaire en la commande) de la forme

$$(S) \quad \begin{cases} \dot{q}(t) = A_0(q) + \sum_{i=1}^{m-1} u_i(t) A_i(q) \\ y(t) = f(q), \end{cases}$$

un système bilinéaire (*B*) approchant le système (*S*) à l'ordre *k*, dans le sens suivant : les sorties des 2 systèmes ont le même développement de Taylor jusqu'à l'ordre *k*.

Le système bilinéaire obtenu est donné sous forme générique, admettant en paramètre l'état initial du système (*S*).

Cette approximation est valide en tout état initial *q*(0), hormis en certains points singuliers qui proviennent soit des singularités du système initial, soit de la procédure de résolution des équations.

Si *q*(0) est une valeur singulière pour cette représentation générique, on peut d'ailleurs encore exécuter le même algorithme en spécifiant cette fois la valeur initiale *q*(0), pour obtenir un approximant de rang minimal en ce point.

Sachant que le comportement entrée-sortie de tout système dynamique linéaire en la commande est caractérisé par sa série génératrice, le problème posé peut s'énoncer en termes de série : il s'agit de construire une série rationnelle g (série génératrice du système bilinéaire (B)), qui coïncide avec la série génératrice s de (S), pour tous les mots de longueur $\leq k$, et qui soit de rang minimal.

On notera que l'on ne peut pas en général approcher une fonctionnelle entrée-sortie analytique par la fonctionnelle entrée-sortie d'un système linéaire. Aussi, la classe des systèmes bilinéaires peut apparaître comme la classe la plus simple pouvant approximer les systèmes analytiques.

Le programme proposé à la fin de cet article, dans le langage MACSYMA, effectue le calcul explicite d'un système bilinéaire approximant le système analytique donné, à un ordre k donné, et fournit une représentation réduite. Nous donnons les résultats de l'algorithme sur un exemple.

Plan

I. SYSTÈMES DYNAMIQUES ET SÉRIES FORMELLES

- Systèmes dynamiques
- Séries formelles génératrices

II. MÉTHODE

- Ordre d'approximation
- Approximations linéaires
 - approximations linéaires tangentes
 - approximants de Padé
- Approximations bilinéaires (ou rationnelles)
 - principe
 - mise en œuvre

III. ALGORITHME ET PROGRAMMATION EN MACSYMA

- Codage des mots
- Structure de l'algorithme
- Algorithme en langage MACSYMA

IV. EXEMPLE

- Coefficients
- Résultats de l'algorithme
- Discussion

I. Systèmes dynamiques et séries formelles

I.1. SYSTÈMES DYNAMIQUES

On va s'intéresser aux systèmes dynamiques analytiques, linéaires en la commande, c'est-à-dire aux systèmes pouvant s'écrire sous la forme

$$(S) \quad \begin{cases} \dot{q}(t) = A_0(q) + \sum_{i=1}^{m-1} u_i(t) A_i(q) \\ y(t) = f(q) \end{cases}$$

où

$$\left\{ \begin{array}{l} q \in \text{variété analytique } V \text{ sur } \mathbb{R}. \\ A_0, A_1, \dots, A_{m-1} \text{ sont des champs de vecteurs analytiques définis} \\ \text{dans un voisinage de } q(0). \\ \text{la fonction d'entrée } u(t) = (u_1(t), \dots, u_{m-1}(t)) \text{ est telle que} \\ u_1, \dots, u_{m-1} \text{ soient continues par morceaux.} \\ f \text{ est la fonction d'observation, analytique définie dans un voisinage de} \\ q(0). \end{array} \right.$$

Et on va chercher à approximer de tels systèmes par des systèmes bilinéaires, c'est-à-dire de la forme

$$(B) \quad \begin{cases} \dot{\underline{q}}(t) = \left(M_0 + \sum_{i=1}^{m-1} u_i(t) M_i \right) \underline{q}(t) \\ \underline{y}(t) = \lambda \underline{q}(t) \end{cases}$$

où

$$\left\{ \begin{array}{l} \underline{q} \in \mathbb{R}\text{-espace vectoriel } Q \text{ de dimension finie.} \\ \underline{q}(0) \text{ est l'état initial.} \\ M_0, M_1, \dots, M_{m-1} : Q \rightarrow Q \text{ et } \lambda : Q \rightarrow \mathbb{R} \text{ sont } \mathbb{R}\text{-linéaires.} \\ u(t) = (u_1(t), \dots, u_{m-1}(t)) \text{ est telle que } u_1, \dots, u_{m-1} \text{ soient conti-} \\ \text{nues par morceaux.} \end{array} \right.$$

On verra que le système (B) approxime le système (S) à l'ordre k , si et seulement si, quelle que soit la commande, les sorties des systèmes (S) et (B) ont le même développement de Taylor à l'ordre k .

La construction de l'algorithme d'approximation nécessite le passage par le formalisme des séries génératrices en variables non commutatives.

I.2. SÉRIES FORMELLES GÉNÉRATRICES

Une série formelle s sur un alphabet $X = \{x_0, x_1, \dots, x_{m-1}\}$, à coefficients dans \mathbb{R} , est une application du monoïde libre X^* dans \mathbb{R} , notée

$$s = \sum_{w \in X^*} (s, w) w$$

où X^* est l'ensemble des mots $w = x_{j_v} \dots x_{j_0}$ formés sur l'alphabet X .

On note ε l'élément neutre de X^* , et $|w|$ la longueur du mot w . $\mathbb{R}\langle X \rangle$ et $\mathbb{R}\langle\langle X \rangle\rangle$ sont respectivement les \mathbb{R} -algèbres de polynômes et de séries formelles en variables non commutatives à coefficients dans \mathbb{R} munies de l'addition et du produit de Cauchy.

M. Fliess [4, 5] a montré que l'on peut coder le comportement entrée-sortie d'un système (S) par une série s en variables non commutatives, appelée série génératrice du système. Cette série est construite sur des mots w composés avec les lettres d'un alphabet $X = \{x_0, x_1, \dots, x_{m-1}\}$ associé aux champs de vecteurs A_0, A_1, \dots, A_{m-1} . En notant cette série $s = \sum_{w \in X^*} (s, w) w$, la sortie y du système (S) s'écrit

$$y(t) = \sum_{w \in X^*} (s, w) \int_0^t \delta w$$

où

$$\int_0^t \delta w = \int_0^t u_{j_v}(\tau_v) \int_0^{\tau_v} u_{j_{v-1}}(\tau_{v-1}) \dots \int_0^{\tau_{j_1}} u_{j_0}(\tau_0) d\tau_0 d\tau_1 \dots d\tau_{v-1} d\tau_v.$$

On sait que la série génératrice s a pour expression [5] :

$$s = h_{/q(0)} + \sum_{v \geq 0} \sum_{j_0, \dots, j_v=0}^{m-1} A_{j_0} \dots A_{j_v} f_{/q(0)} x_{j_v} \dots x_{j_0}$$

(la barre $/_{q(0)}$, indiquant la valeur en $q(0)$).

Dans le cas d'un système bilinéaire (B), ce codage par une série peut être calculé comme suit :

$$\begin{aligned} \text{On pose} \quad \mu(x_0) &= M_0 \\ \mu(x_i) &= M_i \quad \text{pour } i = 1, \dots, m-1 \end{aligned}$$

La série génératrice de (B) est alors

$$g = \lambda \underline{q}(0) + \sum_{v \geq 0} \sum_{j_0, \dots, j_v=0}^{m-1} \lambda M_{j_v} \dots M_{j_0} \underline{q}(0) x_{j_v} \dots x_{j_0}.$$

En notant w le mot $x_{j_v} \dots x_{j_0}$ et

$$\mu(w) \text{ l'application linéaire } M_{j_v} \dots M_{j_0},$$

l'expression de g devient :

$$g = \sum_w \lambda \mu(w) \underline{q}(0) w.$$

On remarque que μ peut être considérée comme une représentation du monoïde X^* (des mots sur l'alphabet X) dans le monoïde des endomorphismes de l'espace d'état.

Ceci nous amène à développer les résultats suivants (cf. [1, 2, 3, 9]).

Une série $s \in \mathbb{R} \langle\langle X \rangle\rangle$ est inversible si et seulement si son terme constant est non nul. Une sous-algèbre A de $\mathbb{R} \langle\langle X \rangle\rangle$ est rationnellement close si et seulement si l'inverse de toute série inversible de A , appartient encore à A .

La \mathbb{R} -algèbre $\mathbb{R} \langle(X)\rangle$ des séries rationnelles est la plus petite sous-algèbre rationnellement close de $\mathbb{R} \langle\langle X \rangle\rangle$ qui contienne $\mathbb{R} \langle X \rangle$.

Une caractérisation des séries rationnelles est donnée par le théorème de Kleene-Schutzenger :

Une série de $\mathbb{R} \langle\langle X \rangle\rangle$ est rationnelle si et seulement si elle est reconnue par un \mathbb{R} -automate [10].

Rappelons qu'on appelle \mathbb{R} -automate, le quadruplet $(E, \gamma, \lambda, \mu)$ tel que

$$\begin{cases} E = \mathbb{R}^N & \text{espace d'état de dimension finie} \\ \gamma \in E & \text{vecteur d'état initial} \\ \lambda & \text{forme linéaire de } E \text{ dans } \mathbb{R} \\ \mu & \text{représentation de } X^* \text{ dans } \text{End}(E). \end{cases}$$

La série s reconnue par l'automate est

$$s = \sum_{w \in X^*} (\lambda \mu(w) \gamma) w.$$

Si (λ, μ, γ) sont définis comme dans le système bilinéaire (B) , on obtient un \mathbb{R} -automate \mathcal{A} associé à (B) .

Il est donc clair qu'une série formelle est reconnue par un \mathbb{R} -automate si et seulement si elle est la série génératrice d'un système bilinéaire.

I.3. MATRICES DE HANKEL ET SYSTÈMES RÉDUITS

A toute série s de $\mathbb{R} \langle\langle X \rangle\rangle$, est associée sa matrice de Hankel $H(s)$ indicée par $X^* \times X^*$ et définie par :

$$H(s)_{u,v} = (s, uv) \text{ pour } u, v \text{ dans } X^*.$$

On définit le rang de s , noté $\text{rg}(s)$, comme égal au rang de la matrice $H(s)$. Une autre caractérisation des séries rationnelles est donnée par le théorème suivant : [2]

Une série s de $\mathbb{R}\langle X \rangle$ est rationnelle si et seulement si elle est de rang fini N . Elle est alors reconnue par un \mathbb{R} -automate de dimension N .

Un \mathbb{R} -automate est dit réduit si son espace d'état est de dimension minimale. Un système bilinéaire est dit réduit si et seulement si le \mathbb{R} -automate associé est réduit. De plus, si deux \mathbb{R} -automates reconnaissant la même série formelle rationnelle sont réduits, alors ils sont canoniquement isomorphes.

G. Jacob [9] donne une construction de la représentation réduite d'une série rationnelle s à partir de sa matrice de Hankel. Cette construction fournit explicitement un \mathbb{R} -automate réduit $(\bar{\lambda}, \bar{\mu}, \bar{\gamma})$ tel que

$$(s, w) = \bar{\lambda} \bar{\mu}(w) \bar{\gamma} \quad \forall w \in X^* .$$

II. Méthode

II.1. L'ORDRE D'APPROXIMATION

Étant donné deux systèmes analytiques (S_1) et (S_2) définis pour les mêmes entrées, nous dirons que (S_1) approxime (S_2) à l'ordre k si et seulement si les séries génératrices s_1 de (S_1) et s_2 de (S_2) , coïncident jusqu'à l'ordre k .

$$\text{i.e. } \forall w \in X^* \text{ tel que } |w| \leq k, \text{ alors } (s_1, w) = (s_2, w) .$$

THÉORÈME : *Si le système (S_1) approxime le système (S_2) à l'ordre k , alors, quelles que soient les entrées, les sorties ont même développement de Taylor jusqu'à l'ordre k .*

Il suffit de montrer le lemme suivant :

LEMME : *Les dérivées successives $y^{(i)}(0)$ de la sortie y d'un système analytique en $t = 0$, sont obtenues à partir des coefficients des mots de longueur inférieure ou égale à i , dans l'expression de la série génératrice s du système.*

Preuve : Notons $y(t) = (s, \varepsilon) + \sum_v \sum_j (s, x_j v) \int_0^t \delta(x_j v)$ la sortie du système.

Alors, en utilisant le calcul développé par M. Fliess et C. Reutenauer dans la théorie de Picard-Vessiot [6], on montre que les dérivées successives $y(0)$, $y^{(1)}(0)$, ..., $y^{(k)}(0)$, de la sortie y du système, s'obtiennent à partir des coefficients (s, w) de la série génératrice s , associés aux mots de longueur $\leq k$. En distinguant le mot vide ε , des autres mots $v = x_\ell w$, on calcule ces dérivées successives. En particulier :

$$\begin{aligned} y(0) &= (s, \varepsilon) \\ y^{(1)}(0) &= \sum_j u_j(0)(s, x_j) \\ y^{(2)}(0) &= \sum_j (s, x_j) u_j^{(1)}(0) + \sum_j \sum_k (s, x_j x_k) u_j(0) u_k(0) \\ &\text{etc...} \end{aligned}$$

Il en résulte que le développement de Taylor à l'ordre k de y ne dépend que des coefficients de la série s tronquée à l'ordre k .

II.2. APPROXIMATIONS LINÉAIRES

1. Approximation linéaire tangente

La méthode consiste à considérer la partie linéaire du comportement entrée-sortie d'un système autour d'un point de fonctionnement.

Plus précisément, on remplace le champ de vecteurs

$$A_0 = \sum_{i=1}^N \theta_0^i(q) \frac{\partial}{\partial q_i}$$

par

$$B_0 = \sum_{i=1}^N \left(\theta_0^i(q(0)) + \sum_{j=1}^N (q_j - q_j(0)) \frac{\partial \theta_0^i}{\partial q_j}(q(0)) \right) \frac{\partial}{\partial q_i}$$

et $\forall p = 1, \dots, m-1$

$$A_p = \sum_{i=1}^N \theta_p^i(q) \frac{\partial}{\partial q_i} \quad \text{par} \quad B_p = \sum_{i=1}^N \theta_p^i(q(0)) \frac{\partial}{\partial q_i}.$$

Cette méthode présente l'inconvénient de ne pas être générique : l'approximation qu'elle propose dépend du point de fonctionnement choisi. Par ailleurs, elle peut, dans certains cas, conduire à un système linéaire n'approchant plus le système donné dès l'ordre $k = 2$ (voir exemple).

2. *Approximants de Padé*

La théorie des approximants de Padé permet d'approcher une série en une variable par un quotient de deux polynômes, à n'importe quel ordre.

Dans le cas des systèmes à une entrée, on peut approcher la série génératrice

$$s = \sum_{w \in \{x_0, x_1\}^*} (s, w) w$$

par

$$\sigma = \sum_{i=0}^{\infty} (s, x_0^i) x_0^i + \sum_{i=0}^{\infty} (s, x_0^i x_1) x_0^i x_1 = \sigma_0 + \sigma_1 x_1.$$

Puis on peut approcher à un ordre k arbitraire chacune des séries σ_0 et σ_1 en une seule variable. On peut donc obtenir un approximant de Padé de s à un ordre arbitraire si $\text{supp}(s) \subset x_0^* \cup x_0^* x_1$.

Mais si (s, w) n'est pas nul pour au moins un mot $w \notin x_0^* \cup x_0^* x_1$, alors on ne pourra trouver aucun approximant de Padé de s à un ordre supérieur ou égal à la longueur de w (voir exemple).

Les approximants de Padé sont des quotients de deux polynômes commutatifs, i.e. des séries rationnelles commutatives. Pour l'étude des séries génératrices non commutatives, il convient de les remplacer par des approximants qui soient des séries rationnelles non commutatives. C'est exactement ce que nous proposons dans cet article.

II.3. APPROXIMATIONS BILINÉAIRES (OU RATIONNELLES)

1. *Principe*

D'après le théorème de II.1, le développement de Taylor à l'ordre k de la sortie y du système analytique (S) ne dépend que des coefficients (s, w) des mots de longueur inférieure ou égale à k , de la série génératrice.

La méthode proposée est la suivante :

Étant donné un système analytique (S) , de série génératrice s , on lui associe un système bilinéaire (B) de série génératrice g rationnelle, telle que g coïncide avec s pour les mots de longueur $\leq k$, et soit de rang minimal.

Le système (B) proposé est donc réduit et tel que les sorties des systèmes $S)$ et (B) aient même développement de Taylor jusqu'à l'ordre k .

2. *Mise en œuvre*

La mise en œuvre est faite par une implémentation de l'algorithme proposé dans [7] et [8].

Il s'agit de la démarche suivante :

- a) On considère le polynôme p obtenu en tronquant la série génératrice s du système (S) à l'ordre k .
- b) On calcule la matrice de Hankel exacte de ce polynôme p . Rappelons que cette matrice $H(p)$ est finie puisque p est un polynôme.
- c) On cherche, si possible, à obtenir récursivement une relation de dépendance linéaire de chaque colonne de $H(p)$ par rapport aux colonnes précédentes, tout en préservant la structure de matrice de Hankel, quitte à remplacer certains coefficients nuls par des coefficients non nuls.

Si l'on note C_v la colonne d'indice v d'une matrice de Hankel, indexée par $X^* \times X^*$, il faut rappeler que la colonne C_{vx_i} est déjà entièrement écrite dans la colonne C_v . C'est d'ailleurs ce phénomène qui permet de considérer les colonnes d'une matrice de Hankel $H(s)$, comme les états de l'automate syntaxique à gauche, à valeurs dans \mathbb{R} , qui reconnaît la série s .

Dans la construction précédente, soit $(C_{v_j})_{j=1 \dots \rho}$ une base de la matrice de Hankel modifiée $H(g)$. Cette famille de vecteurs colonnes forme alors une base de l'espace d'état, de dimension ρ minimale.

Notre construction permet d'obtenir le système bilinéaire (B) [9]

$$\begin{aligned} \dot{\underline{q}}(t) &= \left(\sum_{i=0}^{m-1} u_i(t) M_i \right) \underline{q}(t) \\ u_0(t) &= 1, \end{aligned}$$

où les matrices M_i vérifient la relation

$$M_i C_{v_j} = C_{x_i v_j} \quad \begin{array}{l} \forall i \text{ tel que } 0 \leq i \leq m-1 \\ \forall j \text{ tel que } 1 \leq j \leq \rho \end{array}$$

et où l'état initial est $(1 \ 0 \ \dots \ 0)$

et l'observation est définie par $\lambda_j = (g, v_j)$.

III. Algorithme de calcul et programmation d'un système bilinéaire approximant

On suppose

$$m > 1 \quad \text{et} \quad \exists w \in X^* \text{ tel que } |w| \leq k \text{ et } (s, w) \neq 0.$$

III.1. CODAGE DES MOTS

Les lignes et les colonnes de la matrice de Hankel $H(s)$ sont indicées par des mots rangés par longueur croissante, et pour une longueur donnée, dans l'ordre lexicographique.

Le problème est de coder les mots de X^* dans les entiers.

- On peut proposer un codage préliminaire :

$$\text{précode } (\varepsilon) = 1$$

$$\text{précode } (wx_s) = m \times \text{précode } (w) + s \quad \forall w \in X^*, \forall x_s \in X$$

On en déduit que

$$\text{précode } (x_{i_1} \dots x_{i_\ell}) = m^\ell + i_1 m^{\ell-1} + \dots + i_\ell m^0.$$

En d'autres termes, le mot $x_{i_1} \dots x_{i_\ell}$ est codé par l'entier qui s'écrit en base m :

$$1 i_1 \dots i_\ell$$

- Mais pour $m > 2$, ce codage fait apparaître des « trous » : entre le dernier mot de longueur $\ell - 1$ et le premier mot de longueur ℓ , on « saute » tous les entiers dont l'écriture en base m commence par $2, \dots, m - 1$.

On saute donc $(m - 2) m^{\ell-1}$ entiers.

On corrige donc le précode en lui retranchant le nombre de « trous ».

On obtient alors :

$$\text{code } (w) = \text{précode } (w) - (m - 2)(1 + m + \dots + m^{\ell-1})$$

c'est-à-dire

$$\text{code } (x_{i_1} \dots x_{i_\ell}) =$$

$$= m^\ell + i_1 m^{\ell-1} + \dots + i_{\ell-1} m + i_\ell m^0 - (m - 2) \frac{m^\ell - 1}{m - 1}.$$

Remarquons l'expression du code de $x_s w$ en fonction de celle du code de w :

— Pour $w = x_{i_1} \dots x_{i_\ell}$,

$$\text{précode } (w) = 1 i_1 \dots i_\ell \quad \text{en base } m$$

$$\text{précode } (x_s w) = 1 s i_1 \dots i_\ell \quad \text{en base } m$$

— soit $\text{précode}(x_s w) = \text{précode}(w) + (m + s) m^\ell - m^\ell$

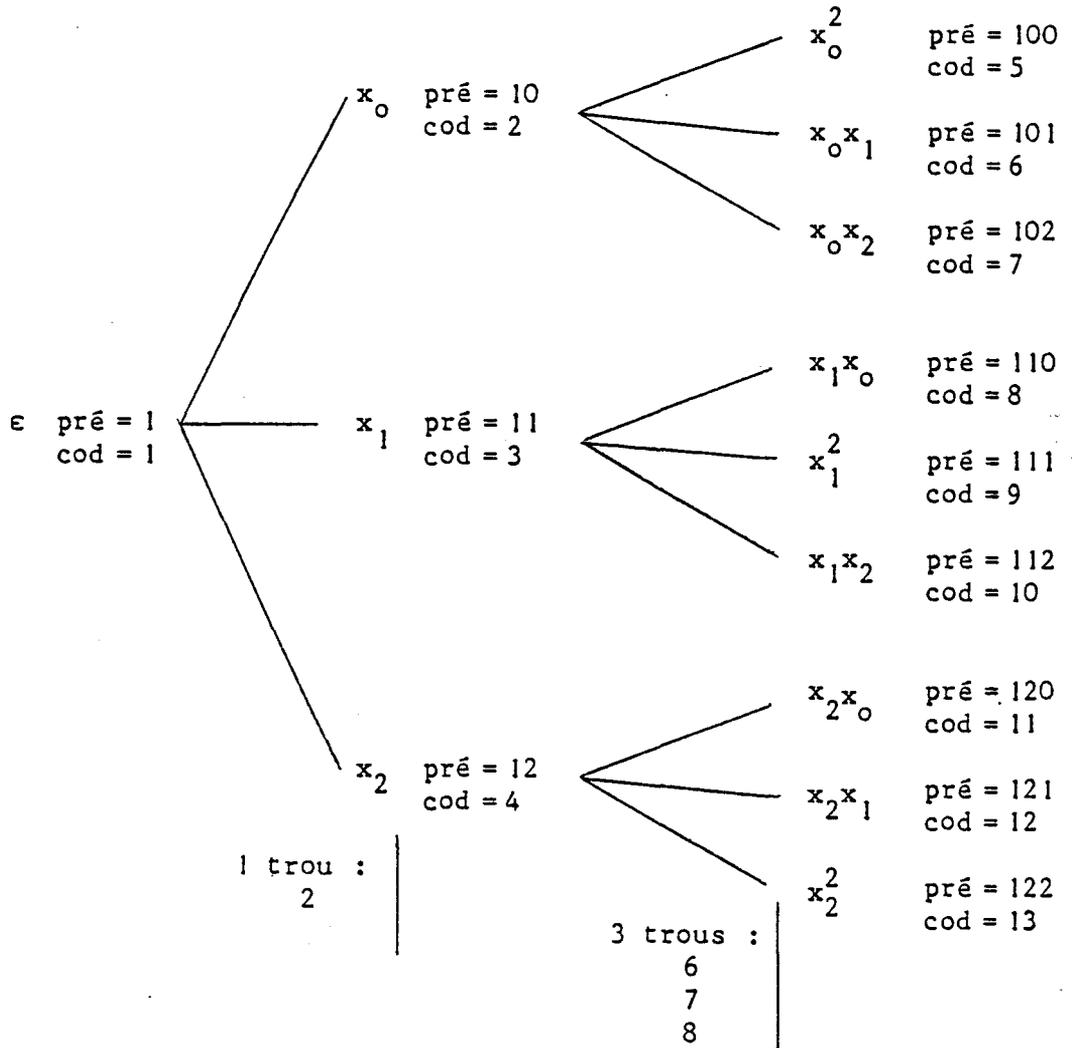
Il en résulte que

$$\text{code } (x_s w) = \text{code } (w) + (m + s - 1) m^\ell - (m - 2) m^\ell$$

puisque $(m - 2) m^\ell$ est la taille du « trou » dans le passage des mots de longueur ℓ , aux mots de longueur $\ell + 1$. Donc finalement

$$\text{code}(x_s w) = \text{code}(w) + (s + 1) m^\ell.$$

Par exemple pour $m = 3$, en notant pré le précode en base m et cod, le code, on a le schéma :



III.2. STRUCTURE DE L'ALGORITHME

L'algorithme est basé sur la structure de l'arbre générateur de mots (cf. schéma précédent).

* Le chargement de la première colonne de la matrice de Hankel, i.e. le calcul des (s, u_ϵ) se fait dans 3 boucles imbriquées :

- La 1^{re} boucle s'effectue relativement à la longueur long, des mots qui seront ensuite concaténés à droite par une lettre x_s .

- La 2^e boucle s'effectue relativement au précode j du mot w considéré, de longueur long.
- La 3^e boucle s'effectue relativement à l'ordre du successeur du mot w considéré.

* Le report de ces coefficients $(s, u\varepsilon)$ aux autres colonnes de la matrice de Hankel, se fait grâce à 4 boucles imbriquées faisant intervenir successivement la longueur du préfixe, la longueur du suffixe, le précode j_1 du préfixe, le précode j_2 du suffixe. Ce report correspond à une concaténation des mots de précodes j_1 et j_2 .

* La recherche des colonnes libres de cette matrice de Hankel, et l'expression des matrices M_0, \dots, M_{m-1} se fait selon la méthode développée en II.2. :

- La 1^{re} boucle s'effectue en comparant le nombre de colonnes libres J dont on a déjà examiné les successeurs et le nombre de colonnes libres déjà sélectionnées T .
- Une colonne libre, indicée par le mot w , ayant été sélectionnée, la 2^e boucle s'effectue relativement à l'ordre du successeur du mot w : plus précisément si la colonne indicée par $x_s w$ est libre, on stocke son indice, on met à jour la matrice M_s , et on prolonge par des zéros, les parties des matrices M_i calculées antérieurement. Si la colonne indicée par $x_s w$ est liée, on résout un système d'équations linéaires afin de mettre à jour M_s .

III.3. ALGORITHME EN LANGAGE MACSYMA

L'algorithme est donné sous forme d'une procédure de nom « bilinéaire », de paramètres

a : tableau de champs de vecteurs

m : nombre de champs de vecteurs

k : ordre d'approximation

f : fonction analytique de V dans R

bilinéaire(a, m, k, f) := block([...], local(...),

(chargement de la 1^{re} colonne de la matrice de Hankel h)

$h[1, 1] : f(q)$,

for long from 0 thru $k - 1$ do

(sont déjà chargés les $h[i, \ell]$ pour longueur de $i \leq \text{long}$)

(for J from m^{long} thru $2m^{\text{long}} - 1$ do

(traitement des successeurs immédiats de i)

(for s from 0 thru $m - 1$ do

$$h \left[mJ + s - \frac{(m-2)(m^{\text{long}+1} - 1)}{m-1}, 1 \right] :$$

$$a[s] \left(h \left[J - \frac{(m-2)(m^{\text{long}} - 1)}{m-1}, 1 \right] \right)$$

),

(report dans les autres colonnes de h)
 for lpre from 0 thru $k - 1$ do
 (les préfixes de longueur k ont déjà été traités)
 (for lsu from 1 thru $k - lpre$ do
 (parcours des préfixes et suffixes de longueur correspondante)
 (for j1 from m^{lpre} thru $2m^{lpre} - 1$ do
 (for j2 from m^{lsu} thru $2m^{lsu} - 1$ do
 (chargement du concaténé de j1 et j2)

$$h \left[j1 - (m - 2) \frac{(m^{lpre} - 1)}{m - 1}, j2 - (m - 2) \frac{(m^{lsu} - 1)}{m - 1} \right] :$$

$$h \left[j1 m^{lsu} - (m - 2) \frac{(m^{lsu + lpre} - 1)}{m - 1} + j2 - m^{lsu}, 1 \right]$$
)
)
)
),

(algorithme de recherche des colonnes libres de h et calcul des matrices $\mu[x_s]$ telles que $\mu[x_s] C_{w_r} = C_{x_s w_r}$)

(dans J on met le nombre de colonnes libres dont on a examiné les successeurs, dans T on met le nombre de colonnes libres déjà créées, dans $b[u]$, resp. $\ell[u]$, on met l'indice de la u -ième colonne libre, resp. la longueur du mot indiquant cette colonne, dans aa on met la partie de h contenant les colonnes libres)

(initialisation)

$J: 0, T: 0, NL1: \frac{m^{k+1} - 1}{m - 1}, r0: 0,$

$b[1]: 1,$

for kk from 1 thru $NL1$ do $aa[kk, 1]: h[kk, 1],$

$g1: \text{genmatrix}(aa, NL1, 1), r1: \text{rank}(g1),$

if $r1 = r0 + 1$ then ($T: 1, \ell[T]: 0$),

(itération)

while $J < T$ do

$\left(J: J + 1, NL1: \frac{m^{k - \ell[J]} - 1}{m - 1}, \right.$

if $NL1 = 0$ then

(for s from 0 thru $m - 1$ do

for J from 1 thru T do $d[s, J, T]: 0$

)else

(if $J = 1$ or $\ell[J] = \ell[J - 1] + 1$ then

$(g0: \text{genmatrix}(aa, NL1, T), r0: \text{rank}(g0)$

),

(traitement des m colonnes $x_s w_{b[J]}$)

for s from 0 thru $m - 1$ do

$(b[T + 1]: b[J] + (s + 1) m^{\ell[J]},$

(on l'amène dans aa)

for kk from 1 thru $NL1$ do

$aa[kk, T + 1]: h[kk, b[T + 1]],$

$g1: \text{genmatrix}(aa, NL1, T + 1), r1: \text{rank}(g1),$

```

        (traitement de cette colonne)
    if r1 = r0 + 1 then (cette colonne est libre)
        (T : T + 1, r0 : r0 + 1, l[T] : l[J] + 1,
         (mise à jour de la matrice d égale à  $\mu$ )
         d[s, T, J] : 1,
         for kk from 1 thru T - 1 do d[s, kk, J] : 0,
         (on complète les colonnes précédentes de d par des 0)
         for r from 0 thru m - 1 do
             (for u from 1 thru J do
                 (if (u < J) or (u = J and r < s)
                     then d[r, T, u] : 0
                 )
             )
         )
    else (cette colonne est liée)
        (création des équations eq en les variables d)
        (for kk from 1 thru NL1 do
            (eq[kk] : aa[kk, T + 1],
             for c from 1 thru T do
                 eq[kk] : eq[kk] - aa[kk, c] d[s, c, J]
             ),
            (création des listes d'équations et de variables et résolution)
            z : makelist (eq[kk], kk, 1, NL1),
            zz : makelist (d[s, kk, J], kk, 1, T),
            linsolve (z, zz), linsolvewarn : false,
            globalsolve : true
        )
    )
),
    (affichage des matrices  $\mu(x_s)$ , de l'état initial et de l'observation)
    for s from 0 thru m - 1 do
        (for J from 1 thru T do
            (print ("voici la ", J, "ème colonne de M[x", s, "]"),
             for i from 1 thru T do print (d[s, i, J])
            )
        )
    )
    print ("voici l'état initial :"),
    print ("1"), for j from 2 thru T do print ("0"),
    print (" t"), print ("voici (observation) :"),
    for j from 1 thru T do print (aa[1, j])
)

```

IV. Exemples

Nous étudions le système (S) suivant :

$$(S) \begin{cases} A_0 = (-k_1 q_2 - k_2 \sin(q_1)) \frac{\partial}{\partial q_2} + q_2 \frac{\partial}{\partial q_1} \\ A_1 = (1 + q_2) \frac{\partial}{\partial q_2} \\ y(t) = q_2 \end{cases}$$

et posons $F = -k_1 q_2 - k_2 \sin(q_1)$

1. Calcul des coefficients de la série génératrice s jusqu'à l'ordre 3 :

$$\begin{aligned} s &= q_2 + Fx_0 + (1 + q_2)x_1 + (-k_1 F + q_2 F'_{q_1})x_0^2 \\ &\quad - k_1(1 + q_2)x_0 x_1 + Fx_1 x_0 + (1 + q_2)x_1^2 \\ &\quad + (F(k_1^2 + F'_{q_1}) + q_2(-k_1 F'_{q_1} + q_2 F''_{q_1}))x_0^3 \\ &\quad + (1 + q_2)(k_1^2 + F'_{q_1})x_0^2 x_1 - k_1 Fx_0 x_1 x_0 - k_1(1 + q_2)x_0 x_1^2 \\ &\quad + (q_2 F'_{q_1} - k_1 F)x_1 x_0^2 - k_1(1 + q_2)x_1 x_0 x_1 \\ &\quad + Fx_1^2 x_0 + (1 + q_2)x_1^3 + \tau(X^3) \end{aligned}$$

2. Résultats de l'algorithme

L'exécution de la procédure bilinéaire à l'ordre 5 donne un rang minimal égal à 6, le système bilinéaire étant donné sous forme matricielle par

$$M_0 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & \frac{q_2 + 1}{q_2} & \beta b & \mu b & yb \\ 0 & 0 & \frac{(2q_2 + 1) \sin(q_1) k_2 + q_2^2 k_1}{q_2^2 + q_2} & \gamma b & \nu b & zb \\ 0 & 1 & 0 & \%r 10 & \%r 14 & \%r 18 \\ 0 & 0 & 1 & \varepsilon b & \pi b & Kb \\ 0 & 0 & 0 & \%r 9 & \%r 13 & \%r 17 \end{pmatrix}$$

$$M_1 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \beta c & \mu c & yc \\ 1 & 0 & 0 & \gamma c & \nu c & zc \\ 0 & 0 & 0 & \%r 12 & \%r 16 & \%r 20 \\ 0 & 1 & 0 & \varepsilon c & \pi c & Kc \\ 0 & 0 & 1 & \%r 11 & \%r 15 & \%r 19 \end{pmatrix}$$

Donc, dès l'ordre 2, cet approximant linéaire tangent n'est plus satisfaisant.

c) Hors des singularités précédentes, le coefficient de x_1^2 étant non nul, tout approximant de Padé devient non satisfaisant dès l'ordre 2.

• Singularités

L'approximant bilinéaire minimal générique de (S) est défini en dehors des singularités

$$\begin{cases} q_2(0) = 0 \\ q_2(0) + 1 = 0 \\ \sin(2q_1(0)) = 0. \end{cases}$$

Les deux premiers cas correspondent à un système (S) dégénéré dans le sens suivant :

— si $q_2(0) = 0$

$$(S) \text{ devient } \begin{cases} A_0 = -k_2 \sin(q_1) \frac{\partial}{\partial q_2} \\ A_1 = \frac{\partial}{\partial q_2} \end{cases}$$

— si $q_2(0) = -1$

$$(S) \text{ devient } \begin{cases} A_0 = (k_1 - k_2 \sin(q_1)) \frac{\partial}{\partial q_2} - \frac{\partial}{\partial q_1} \\ A_1 = 0. \end{cases}$$

La troisième singularité est la conséquence d'un choix dans la résolution des équations permettant de sélectionner des colonnes libres dans la matrice de Hankel.

Dans chacun de ces cas, il y a lieu d'évaluer numériquement la matrice de Hankel en l'état singulier initial, et l'algorithme fournit alors un système numérique, et donc non générique, approximant bilinéaire minimal en ce point singulier.

On obtient alors les rangs minimaux suivants :

Singularité :	$q_2(0) = 0$			rang bilinéaire minimal :	5
—	$q_2(0) = -1$	—	—	—	4
—	$q_1(0) = 0$	—	—	—	5
—	$q_1(0) = \pi$	—	—	—	5
—	$q_1(0) = \pi/2$	—	—	—	6
—	$q_1(0) = 3\pi/2$	—	—	—	6

Un complément à cet article, contenant le traitement complet de 3 exemples avec leur analyse et les listings correspondants, peut être trouvé dans [11].

RÉFÉRENCES

- [1] J. BERSTEL, C. REUTENAUER : *Les séries rationnelles et leurs langages*. Masson, 1984.
- [2] M. FLIESS : *Matrices de Hankel*. J. Math. Pur. Appl. Vol. 53, 1974, p. 197-222.
- [3] M. FLIESS : *Un outil algébrique : les séries formelles non commutatives*. In Mathematical Systems Theory (G. Marchesini and S. K. Mitter, eds) Lect. Notes Econom. Math. Syst., Vol. 131, Springer Verlag, 1976, p. 122-148.
- [4] M. FLIESS : *Fonctionnelles causales non linéaires et indéterminées non commutatives*. Bull. Soc. Math. France, 109, 1981, p. 3-40.
- [5] 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, p. 554-570.
- [6] M. FLIESS, C. REUTENAUER : *Théorie de Picard-Vessiot des systèmes réguliers (ou bilinéaires)*. Colloque Nat. CNRS-RCP 567, Belle-Ile, septembre 1982, in Outils et Modèles Mathématiques pour l'Automatique, l'Analyse des Systèmes et le Traitement du Signal, CNRS, 1983.
- [7] C. HESPEL : *Approximation de séries formelles par des séries rationnelles*. RAIRO, Informatique Théorique, Vol. 18, N° 3, 1984, p. 241-258.
- [8] C. HESPEL, G. JACOB : *Approximation of nonlinear systems by bilinear ones*. In Algebraic and Geometric Methods in Nonlinear Control Theory (M. Fliess and M. Hazewinkel, eds), D. Reidel, 1986, p. 511-520.
- [9] G. JACOB : *Réalisation des systèmes réguliers (ou bilinéaires) et séries génératrices non commutatives*. Séminaire d'Aussois, RCP 567, in Outils et Modèles Mathématiques pour l'Automatique, l'Analyse des Systèmes et le Traitement du Signal, CNRS Landau, 1980.
- [10] M. P. SCHUTZENBERGER : *On the definition of a family of automata*. Inform. Contr. Vol. 4, 1961, p. 245-270.
- [11] C. HESPEL, G. JACOB : *Approximations bilinéaires de systèmes analytiques : exemples de listings*. Publication interne LANS n° 18, INSA de Rennes.

Annexe 4

Hespel C., Jacob G., *Approximation of nonlinear dynamic systems by rational series*, Theoret. Comput. Sciences, 79, pp.151-162, 1991.

Approximation of nonlinear dynamic systems by rational series

Christiane Hespel

INSA, 20, Avenue des Buttes de Coësmes, 35043 Rennes Cedex, France

G rard Jacob

*Laboratoire de Recherche en Informatique Fondamentale (LA 369), Universit  de Lille I,
59655 Villeneuve d'Ascq Cedex, France*

Abstract

Hespel, C., and G. Jacob, Approximation of nonlinear dynamic systems by rational series, Theoretical Computer Science 79 (1991) 151–162.

Given an analytic system, we compute a bilinear system of minimal dimension which approximates it up to order k (i.e. the outputs of these two systems have the same Taylor expansion up to order k). The algorithm is based on noncommutative series computation: let s be the generating series of the analytic system; then a rational series g is constructed, whose coefficients are equal to those of s , for all words of length smaller than or equal to k . These words are digitally encoded, in order to simplify the computations of the Hankel matrices of s and g . We then associate with g , a bilinear system, which is a solution to our problem. Another method may be used for computing a bilinear system which approximates a given analytic system (S). We associate with (S) an \mathbb{R} -automaton of vector fields and build the truncated automaton by cancelling all the states which have the following property: the length of the shortest successful path labelled by a word that gets through this state is strictly greater than k . Then, the number of states of this truncated automaton yields the dimension (not necessarily minimal) of the state-space.

1. Introduction

Several methods may be used for determining the input-output behavior of a dynamic system: transfer functions, functional expansions (Volterra series) [6], and generating power series [5]. With all these descriptions, we have the following problem: is it possible to construct a suitable approximation of the input-output behavior of any dynamic system using a more elementary system?

For single input systems, the transfer function can be used to find a linear approximation by means of Pad  approximants [2, 3]. Nevertheless it is not possible in general to approximate nonlinear systems by linear ones.

Another way for determining approximations of dynamic systems is to compute its first Volterra kernels [6], but in this case one must solve the new problem of the realization of this truncated Volterra series [12].

The formal power series in several noncommutative variables are the most efficient tools for dealing with functional expansions. One cannot approximate any nonlinear functional with a linear system (i.e. a rational series in one variable). But it is possible to approximate any nonlinear functional with a bilinear system (i.e. with a polynomial or a rational series in several noncommutative variables).

In this paper, a new algorithm is presented which enables us to compute a minimal rank bilinear system (B) which approximates a given dynamic system (S)

$$(S) \quad \begin{cases} \dot{q}(t) = A_0(q) + \sum_{i=1}^{m-1} u_i(t) A_i(q), \\ y(t) = f(q), \end{cases}$$

up to order k (i.e. the Taylor expansions of the output of (S) and (B) are equal up to order k). As Fliess showed [5, 6], the input-output behavior of a dynamic system (S), can be coded by a noncommutative formal power series, called the generating series of the system (S).

A generating power series is known to correspond to a finite-dimensional bilinear system if and only if it is rational. Furthermore it is known that a series s is rational if and only if the rank of its Hankel matrix $H(s)$ is finite.

The first method developed here is the following: given a generating series s of (S), one builds a rational series g (associated with bilinear system (B)) which coincides with s for all words of length $\leq k$ and is of minimal rank.

The second method to compute a bilinear system (B) which approximates a given dynamic system (S) is as follows: with the given system (S), we associate an \mathbb{R} -automaton \mathcal{A} . As a series is rational if and only if it is recognized by a finite \mathbb{R} -automaton [11], then we build a truncated automaton τ (associated with the bilinear system (B)), from the \mathbb{R} -automaton \mathcal{A} associated with (S).

2. Notations [1]

Let $X = \{x_1, \dots, x_n\}$ be a finite, nonempty set called alphabet. The set of finite sequences $x_{j_1} \dots x_{j_l}$ of elements of X is the free monoid X^* (with concatenation and neutral element 1).

An element of X^* is a word $w = x_{j_1} \dots x_{j_l}$; the length l of which is denoted by $|w|$.

A formal series s with coefficients in \mathbb{R} is a function

$$X^* \rightarrow \mathbb{R}, \quad \omega \rightarrow (s, \omega)$$

where (s, ω) is called the coefficient of ω in s .

We note by $\mathbb{R}\langle\langle X \rangle\rangle$ the set of formal series over X with coefficients in \mathbb{R} . A structure of a semiring is defined on $\mathbb{R}\langle\langle X \rangle\rangle$ (sum and Cauchy product). The set of polynomials is denoted by $\mathbb{R}\langle X \rangle$.

A formal series $s \in \mathbb{R}\langle\langle X \rangle\rangle$ is proper if the coefficient of the empty word is equal to 0. The sum of the locally finite family $(s^n)_{n \geq 0}$ is denoted by s^* and is called the star of s :

$$s^* = \sum_{n \geq 0} s^n$$

The rational operations in $\mathbb{R}\langle\langle X \rangle\rangle$ are the sum, the Cauchy product, the two external products of \mathbb{R} on $\mathbb{R}\langle\langle X \rangle\rangle$ and the star operation.

The smallest subset containing a subset E of $\mathbb{R}\langle\langle X \rangle\rangle$ and which is rationally closed (i.e. closed for the rational operations) is called the rational closure of E .

A formal series is rational if it belongs to the rational closure of $\mathbb{R}\langle X \rangle$.

A formal series $s \in \mathbb{R}\langle\langle X \rangle\rangle$ is called recognizable if there exists an integer $n \geq 1$, a morphism of monoids

$$\mu : X^* \rightarrow \mathbb{R}^{n \times n}$$

and two matrices $\lambda \in \mathbb{R}^{1 \times n}$ and $\gamma \in \mathbb{R}^{n \times 1}$ such that

$$(s, w) = \lambda \mu \omega \gamma$$

In this case, the triple (λ, μ, γ) is called a linear representation of s , and n is its dimension. According to the Fundamental Theorem of Schutzenberger [11], a formal series is recognizable if and only if it is rational.

The Hankel matrix of a formal series s is the matrix $H(s)$ indexed over $X^* \times X^*$ and defined by

$$H_{u,v}(s) = (s, uv), \quad \forall u, v \in X^*$$

and $\text{rk}(s)$ is the rank of its Hankel matrix.

A series is known to be rational if and only if its rank is finite. In this case, its rank is equal to the minimum of the dimensions of the linear representations of s [4].

3. Approximation by a minimal rank bilinear system

3.1 Approximation of a formal series by a rational series [7]

Let s be a given formal series, we want to build a rational series of minimal rank, which approximates s up to order k .

For that purpose, we compute the Hankel matrix $H(p)$ of the polynome p obtained by truncating s by cancelling the coefficients of the words of length greater than k .

Example 3.1. See fig. 1 for $k = 2$.

Then, we try to find recursively a linearly dependent relation between one column and the precedent columns, by substituting nonzero values for some zeros, while preserving the Hankel matrix structure.

	1	x_0	x_1	x_0^2	x_0x_1	x_1x_0	x_1^2
1	d	a	1	ab	b	0	0
x_0	a	ab	b				
x_1	1	0	0				
x_0^2	ab						
x_0x_1	b						
x_1x_0	0						
x_1^2	0						

Fig. 1.

Let us denote $H_{\leq l, \leq c}$ the Hankel matrix of p , restricted to the rows indexed by the words u of length $\leq l$, as well as to the columns indexed by the words v of length $\leq c$.

In the previous example, the column vector C_1 indexed by 1 and the column vector C_{x_1} indexed by x_1 , are linearly independent in $H_{\leq 2, \leq 0}$, $H_{\leq 1, \leq 1}$, $H_{\leq 0, \leq 2}$, the other column vectors being linearly dependent in these blocks:

$$\begin{aligned}
 C_{x_0x_1} &= bC_{x_1}, \\
 C_{x_1x_0} &= C_{x_1^2} = 0, \\
 C_{x_0^2} &= bC_{x_0}, \\
 C_{x_0} &= aC_{x_1}.
 \end{aligned}$$

In this way, we obtain a minimal rank (=2) rational series even though the rank of $H(p)$ is equal to 3 (Fig. 2).

	1	x_0	x_1	x_0^2	x_0x_1	x_1x_0	x_1^2	x_0^3
1	d	a	1	ab	b	0	0	ab^2
x_0	a	ab	b	ab^2	b^2	0	0	ab^3
x_1	1	0	0	0	0	0	0	.	
x_0^2	ab	ab^2	b^2	ab^3	b^3	0	0	.	
x_0x_1	b	0	0	0	0	0	0	.	
x_1x_0	0	0	0	0	0	0	0	.	
x_1^2	0	0	0	0	0	0	0	.	
x_0^3	ab^2	
.	

Fig. 2.

3.2 Application to the bilinearization of dynamic systems [8]

3.2.1. Method

Consider the dynamic analytic systems, of the form

$$(S) \quad \begin{cases} \dot{q}(t) = A_0(q) + \sum_{i=1}^{m-1} u_i(t) A_i(q), \\ y(t) = f(q). \end{cases}$$

- q is the state which belongs to a finite dimensional \mathbb{R} -analytic manifold V ,
- vector fields A_0, A_1, \dots, A_{m-1} and output function f are analytic and defined in a neighborhood of the initial state $q(0)$.
- $u(t) = (u_1(t) \dots u_{m-1}(t))$ is the input function where u_1, \dots, u_{m-1} are piece-wise continuous.

We build an approximation of these systems by bilinear ones, under the form

$$(B) \quad \begin{cases} \dot{q}(t) = \left(M_0 + \sum_{i=1}^{m-1} u_i(t) M_i \right) q(t), \\ y(t) = \lambda q(t). \end{cases}$$

- q belongs to a finite dimensional \mathbb{R} -vector space Q
- $M_0, \dots, M_{m-1}: Q \rightarrow Q$ and $\lambda: Q \rightarrow \mathbb{R}$ are \mathbb{R} -linear
- Input function $u(t) = (u_1(t) \dots u_{m-1}(t))$ where u_1, \dots, u_{m-1} are piece-wise continuous.

We say that (B) approximates (S) up to order k , if and only if, whatever the input function, the outputs of these two systems have the same Taylor expansion up to order k .

As Fliess showed [5, 6], the input-output behavior of a system (S) can be coded by a noncommutative formal power series s , called the generating series of the system.

For that purpose, with vector fields A_0, \dots, A_{m-1} , he associates an alphabet $X = \{x_0, x_1, \dots, x_{m-1}\}$. The generating series is noted

$$s = \sum_w (s, w) w.$$

The generating power series may be rewritten as [6]

$$s = f|_{q(0)} + \sum_{\nu \geq 0} \sum_{j_0, \dots, j_{\nu} = 0}^{m-1} A_{j_0} \dots A_{j_{\nu}} f|_{q(0)} x_{j_{\nu}} \dots x_{j_0}.$$

A generating power series g corresponds to a finite-dimensional bilinear system if and only if it is rational. The rank of g is also the dimension of the state-space of the reduced system [10].

The approximation of (S) with (B) up to order k amounts to the problem of the approximation of a formal series s with a rational one, g .

3.2.2 Algorithm for the computation of a bilinear system of minimal rank [9]

Consider the following problem. Which is the better coding by successive integers of words $\in X^*$, in order to make a matrix calculation (using arrays) of the Hankel matrix of s ?

Consider for $m = 3$ the word tree shown in Fig. 3; let pre be the precoding (defined below) in the base m notation, and code the coding: We define first the following preliminary coding:

$$\begin{aligned} \text{precoding}(1) &= 1 \\ \text{precoding}(wx_s) &= m \times \text{precoding}(w) + s, \quad \forall w \in X^* \\ &\quad \forall x_s \in X. \end{aligned}$$

Thus,

$$\text{precoding}(x_{i_1} \dots x_{i_l}) = m^l + i_1 m^{l-1} + \dots + i_l m^0.$$

The word $x_{i_1} \dots x_{i_l}$ is then precoded, in the base m notation, by $1i_1 \dots i_l$.

It may be noticed that this coding produces some "holes", for $m > 2$ (between the last word of length $l-1$ and the first word of length l). We obtain the coding by subtracting the number of holes from this precoding:

$$\text{coding}(x_{i_1} \dots x_{i_l}) = m^l + i_1 m^{l-1} + \dots + i_l m^0 - (m-2) \frac{m^l - 1}{m - 1}$$

Moreover, let us emphasize that

$$\text{coding}(x_s w) = \text{coding}(w) + (s+1)m^l.$$

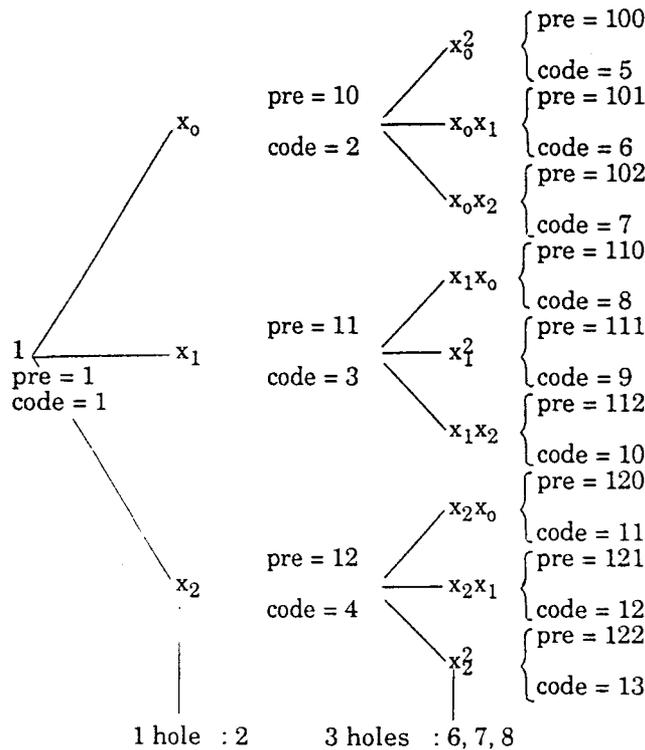


Fig. 3.

This coding of the words allows easy storage of a formal series by the array of its coefficients in $H(s)$.

As the algorithm is written in MACSYMA language, we can make a formal calculation and the generating series s is parameterized by the initial state $q(0)$.

3.2.3 Comparison with the approximation given by the tangent linear system

Example 3.2

$$(S_2) \begin{cases} A_0 = (-k_1 q_2 - k_2 \sin(q_1)) \frac{\partial}{\partial q_2} + q_2 \frac{\partial}{\partial q_1} \\ A_1 = (1 + q_2) \frac{\partial}{\partial q_2} \\ y(t) = q_2 \end{cases}$$

A bilinear system (B_2) , approaching (S_2) up to order $k = 5$, except for the singular points, can be determined using the previous algorithm:

$$(B_2) \begin{cases} \dot{q}(t) = (M_0 + u_1(t)M_1)q(t) \\ y(t) = \lambda q(t) \end{cases}$$

M_0, M_1, λ being parameterized by the initial state:

$$M_0 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & \frac{q_2 + 1}{q_2} & \beta b & \mu b & yb \\ 0 & 0 & \frac{(2q_2 + 1) \sin(q_1) k_2 + q_2^2 k_1}{q_2^2 + q_2} & \gamma b & \nu b & zb \\ 0 & 1 & 0 & \%r10 & \%r14 & \%r18 \\ 0 & 0 & 1 & \varepsilon b & \pi b & Kb \\ 0 & 0 & 0 & \%r9 & \%r13 & \%r17 \end{pmatrix}$$

$$M_1 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \beta c & \mu c & yc \\ 1 & 0 & 0 & \gamma c & \nu c & zc \\ 0 & 0 & 0 & \%r12 & \%r16 & \%r20 \\ 0 & 1 & 0 & \varepsilon c & \pi c & Kc \\ 0 & 0 & 1 & \%r11 & \%r15 & \%r19 \end{pmatrix}$$

$$\lambda = \begin{pmatrix} q_2 \\ -\sin(q_1)k_2 - q_2k_1 \\ q_2 + 1 \\ -k_1(-\sin(q_1)k_2 - q_2k_1) - q_2 \cos(q_1)k_2 \\ -\sin(q_1)k_2 - q_2k_1 \\ q_2 + 1 \end{pmatrix}$$

r_i are parameters which may be chosen arbitrarily, and the other terms, appearing in M_0 and M_1 , are functions of q_1, q_2 . The processing is thus generic for every point except for singular ones. At these points, the Hankel matrix is easy to compute numerically. The algorithm yields a numerical minimal bilinear system, approximating the given system, at these points.

The computation of the tangent linear approximant needs the system equation to be rewritten, at every operating point.

In Example 3.2, the generating series expression s of (S_2) and the generating series expression u of the tangent linear system around $q(0)$, show a difference, from length 2 on, except for singular points:

$$\begin{aligned}(s, x_1 x_0) - (u, x_1 x_0) &= (s, x_1 x_0) = -k_1 q_2(0) - k_2 \sin(q_1(0)), \\(s, x_1^2) - (u, x_1^2) &= (s, x_1^2) \\ &= 1 + q_2(0).\end{aligned}$$

Then, from order 2 on, the tangent linear approximant is invalid.

4. Bilinearization via the vector fields automaton

4.1. The vector fields automaton

Recall that the generating series of a dynamic system (S) may be written as

$$s = f|_{q(0)} + \sum_{\nu \geq 0} \sum_{j_0, \dots, j_\nu=0}^{m-1} A_{j_0} \dots A_{j_\nu} f|_{q(0)} x_{j_\nu} \dots x_{j_0}$$

that is

$$s = \sum_{w \in X^*} A_{\bar{w}} \circ f|_{q(0)} w$$

where \bar{w} is the mirror image of w .

Therefore, the problem reduces to checking the differential operators:

$$A_w = \sum_{\alpha} A_{w,\alpha} D^\alpha$$

with

$$\begin{aligned}D^\alpha &= D_1^{i_1} \dots D_N^{i_N}, \\ D_1 &= \frac{\partial}{\partial q_1}, \dots, D_N = \frac{\partial}{\partial q_N}.\end{aligned}$$

In order to compute these operators, define the following automaton operating on the right whose

states are D^α ,

letter action $D^\alpha * x_i$ is given by (1),

and initial state is I .

Using Leibnitz formula, we can deduce that

$$D^\alpha A_l = \sum_{j=1}^N \sum_{\beta} C_\alpha^\beta D_\alpha^\beta(\theta_l^j) D^{\alpha-\beta} D_j \tag{1}$$

where

$$A_l = \sum_{j=1}^N \theta_l^j(q_1, \dots, q_N) D_j.$$

Moreover, we want to check $A_w \circ f$, that is $\sum_\alpha A_{w,\alpha} D^\alpha(f)$.

Thus, the observation function f enables us to define the final state vector F , determined by its components $F_\alpha = D^\alpha f$.

4.2. The truncated automaton

In order to compute an approximant bilinear system, whose rank is not necessarily minimal, the truncated vector fields automaton can be used. This automaton is constructed by choosing the states D^α that are met along a successful path and the length of which is smaller than or equal to k . Therefore the cut off automaton can be obtained. The number of states appearing in this latter automaton can be easily seen to be the rank of the corresponding bilinear system.

Example 4.1. The Duffing equation is

$$y'' + ay' + b^2y + cy^3 = u_1(t),$$

that is

$$\dot{q}(t) = A_0(q) + u_1(t)A_1(q),$$

$$y(t) = q_1(t),$$

with

$$F = -aq_2 - b^2q_1 - cq_1^3,$$

$$A_0 = F \frac{\partial}{\partial q_2} + q_2 \frac{\partial}{\partial q_1} = FD_2 + q_2D_1,$$

$$A_1 = \frac{\partial}{\partial q_2} = D_2.$$

The actions of A_0 and A_1 produced by the letters x_0, x_1 on states $D_1^i D_2^j$, are given by

$$D_1^i D_2^j A_1 = D_1^i D_2^{j+1},$$

$$D_1^i D_2^j A_0 = FD_1^i D_2^{j+1} + C_1^1 F' D_1^{i-1} D_2^{j+1} + C_1^2 F'' D_1^{i-2} D_2^{j+1} + C_1^3 F''' D_1^{i-3} D_2^{j+1} - ajD_1^i D_2^j + q_2 D_1^{i+1} D_2^j + jD_1^{i+1} D_2^{j-1}.$$

They enable us to obtain the vector fields automaton cell shown in Fig. 4 and the automaton shown in Fig. 5.

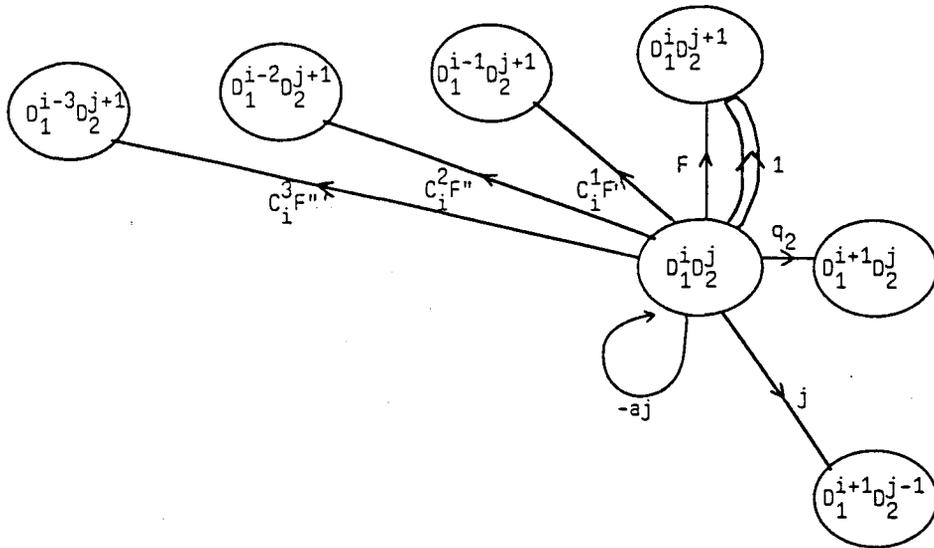


Fig. 4.

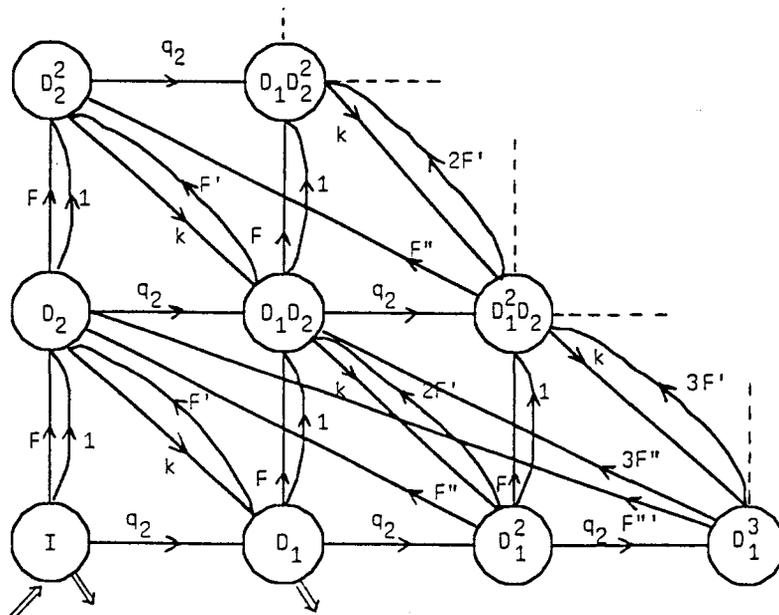


Fig. 5

The truncated automaton can be determined by picking out from these states, the states D^α which are met along a successful path and the length of which is smaller than or equal to k .

Thus we compute

$$j_{acc} = \text{length of the shortest path between } I \text{ and } D^\alpha,$$

$$j_{co-acc} = \text{length of the shortest path between } D^\alpha \text{ and a final state.}$$

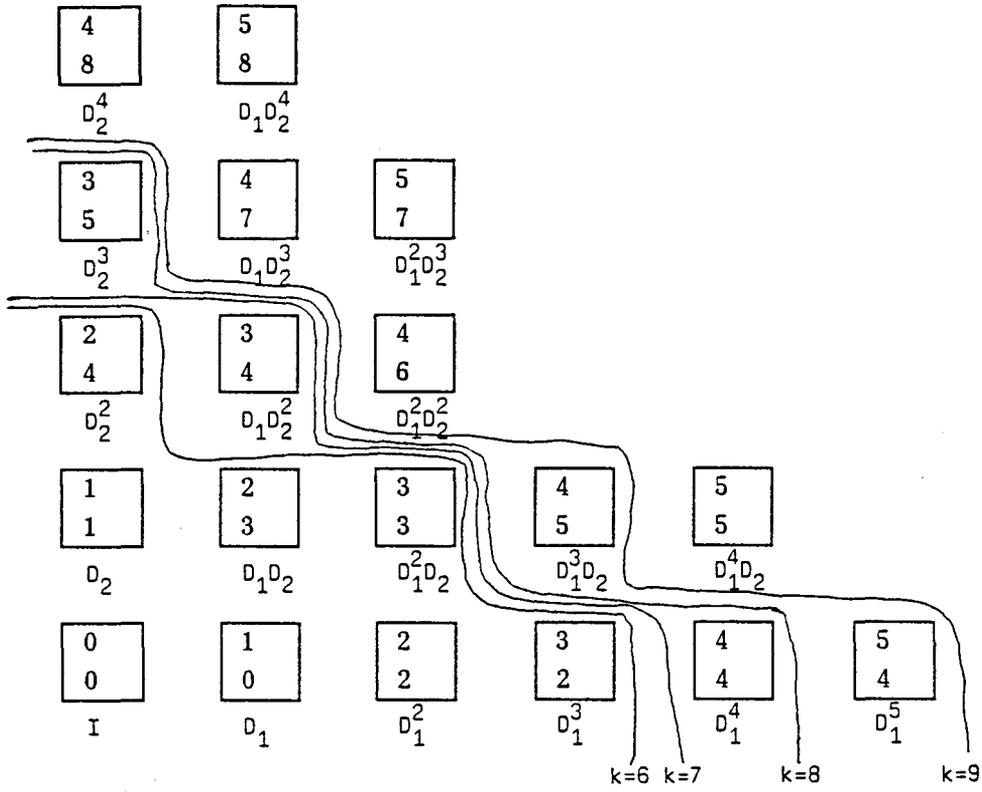


Fig. 6.

	I	D ₁	D ₂	D ₁ ²	D ₁ D ₂	D ₂ ²	D ₁ ³	D ₁ ² D ₂	D ₁ D ₂ ²
1	q ₁	1	0	0
x ₀	q ₂	0	1	0	0
x ₁	0	0
x ₀ ²	F	F'	-a	F''	0	0	F'''	0	0
x ₀ x ₁	0	0
x ₁ x ₀	1	0	0
x ₁ ²	0	0
x ₀ ³	-aF+q ₂ F'	q ₂ F''-aF'	a ² +F'	q ₂ F'''-aF''	F''	0	-aF'''	F'''	0
x ₁ ³	0	0
x ₀ ⁴	F(a ² +F')	F(a ² +F')	-a(a ² +2F')	F''(a ² +3F')	-2aF''	2F''	3F'' ²	-2aF'''	2F'''
	+q ₂ (q ₂ F''-aF') + FF''	+2q ₂ F''	+F'''(F-aq ₂)	+2q ₂ F'''	+F'''(a ² +4F')				
		+q ₂ (q ₂ F'''-aF'')							
x ₁ x ₀ ³	a ² +F'	F''	0	F'''	0				0

Fig. 7.

Let $\boxed{\begin{smallmatrix} j_{\text{acc}} \\ j_{\text{co-acc}} \end{smallmatrix}}$ denote the pair $(j_{\text{acc}}, j_{\text{co-acc}})$ for every state D^α .

For instance, for $k=7$, a truncated automaton containing 9 states, is obtained this way (Fig. 6).

By truncating the vector fields automaton, a rational series realization is obtained, which approximates the given series up to order k .

The truncated automaton is completely accessible but generally, it is not completely observable. Let $M = (D^\alpha * w \circ f)$. M is a matrix whose rows are indexed by words w and whose columns are indexed by the states D^α . The element of row w , column D^α is $D^\alpha A_w f$. In this matrix (Fig. 7), the columns indexed by the states $D_1 D_2^2$ and $D_1^2 D_2$ can be readily seen to be linear combinations of the other columns.

A minimal automaton may be obtained in this way, whose rank, equal to 7, is minimal.

References

- [1] J. Berstel and C. Reutenauer, *Rational Series and their Languages* (Springer, Berlin, 1988).
- [2] C. Brezinski, Padé type approximation and general orthogonal polynomials, INSM50, Birkhäuser.
- [3] J. Della Dora, Quelques notions sur les approximants de Padé, in: *Outils et Modèles Mathématiques pour l'Automatique, l'Analyse des Systèmes et le Traitement du Signal*, vol. 2 (CNRS, 1982) 203-224.
- [4] M. Fliess, Matrices de Hankel, *J. Math. Pure Appl.* **53** (1974) 197-222.
- [5] M. Fliess, Fonctionnelles causales non linéaires et indéterminées non commutatives, *Bull. Soc. Math. France* **109** (1981) 3-40.
- [6] M. Fliess, M. Lamnabhi and F. Lamnabhi-Lagarrigue, An algebraic approach to nonlinear functional expansions, *IEEE Trans. Circuits Systems* **30**(8) (1983) 554-570.
- [7] C. Hespel, Approximation de séries formelles par des séries rationnelles, *RAIRO Inform. Théor.* **18**(3) (1984) 241-258
- [8] C. Hespel and G. Jacob, Approximation of nonlinear systems by bilinear ones, in: M. Fliess and M. Hazewinkel, eds., *Algebraic and Geometric Methods in Nonlinear Control Theory* (D. Reidel, Dordrecht, 1986) 511-520.
- [9] C. Hespel and G. Jacob, Calcul des approximations locales bilinéaires de systèmes analytiques, *RAIRO Automat.-Prod. Inform. Ind.* **23** (1989) 331-349.
- [10] G. Jacob, Réalisation des systèmes réguliers (ou bilinéaires) et séries génératrices non commutatives, in: *Séminaire d'Aussois, RCP567, Outils et modèles mathématiques pour l'Automatique, l'Analyse des Systèmes, et le traitement du Signal* (CNRS, Landau, 1980).
- [11] M.P. Schutzenberger, On the definition of a family of automata, *Inform. and Control* **4** (1961) 245-270.
- [12] W. Smith, Kuszta and Bailey, Mode identification of bilinear systems, *Int. J. Control* **37**(7) (1983) 943-957.

Annexe 5

Hespel C., *Truncated bilinear approximants: Carleman, finite Volterra, Padé-type, geometrical and structural automata*, Lecture Notes in Control and Information Sciences, vol. 165, "Algebraic Computing in Control", pp. 265-278, Springer-Verlag, 1991.

Lecture Notes in Control and Information Sciences

Edited by M. Thoma and A. Wyner

165

G. Jacob,
F. Lamnabhi-Lagarrigue (Eds.)

Algebraic Computing in Control

Proceedings of the First European Conference
Paris, March 13–15, 1991



Springer-Verlag
Berlin Heidelberg New York
London Paris Tokyo
Hong Kong Barcelona Budapest

Truncated bilinear approximants: Carleman, finite Volterra, Padé-type, geometric and structural automata.

C.HESPEL

INSA, 20 av. Buttes de Coësmes, 35043 RENNES Cedex.

15 Octobre 1990

ABSTRACT

In this paper, two original infinite \mathbb{R} -automata representations of an analytic systems are provided, allowing to compute easily a large class of bilinear approximants, including the so called "Carleman linearization method" as well as the "finite Volterra expansions" and "Padé-type".

Each of these \mathbb{R} -automata is an infinite valued graph, and any "truncation" of this graph is an explicit bilinear approximation of the original system, of which the state space description can be obtain *directly, without any computation*.

The "*geometric \mathbb{R} -automaton*" is an improvement of the Carleman description, replacing the tensor powers of the state vector by its symmetrical powers. (This valued graph has also been obtained independently by Viennot, using combinatorial technics.) The "*structural \mathbb{R} -automaton*" describes an action on differential operators.

Any choice of an iterated truncation procedure produces a family of bilinear approximants, that can be viewed as a family of *noncommutative Padé-type approximants*. A particular choice, detailed here, produces the "up to order k " expansion, that gives the exact value for the "*k-order*" Taylor expansion of the output.

Introduction

The aim of this paper is to give two new representation tools for non linear analytic dynamical systems by infinite automata. The first method, by "geometric bilinearization", is based on the differential calculus in the state-space. It is an automaton representation of the Carleman "linearization method", given in a "non redundant form".

The second method, by "structural bilinearization", is based on computations in the algebra of the differential operators defined by the vector fields of the given system.

These two automata are directly obtained from the system equations, by using the Leibnitz derivation formula. These two infinite automata can be truncated according to several criteria, allowing to compute any well-known bilinear approximation like the Carleman method [2,3,16], the bilinear finite Volterra expansions (when there exist) [2].

To be more exact, any iterated truncation procedure provides a family of noncommutative Padé-type approximants [13].

Moreover, a simple reading of such a truncated bilinear automaton furnishes a bilinear dynamic system approximating the given analytic system. This method was partly presented in [8,9,10], and is illustrated here, for the geometrical and structural approaches, according to an approximation criterion "up to order k ". We obtain in that way a bilinear approximation that provides an output Taylor expansion equal to the output Taylor expansion of the given system, up to order k , whatever the input may be.

Let (Σ) be a given nonlinear dynamic analytic system :

$$(\Sigma) \begin{cases} \dot{x}(t) &= A(x) + \sum_{i=1}^m u_i(t)B_i(x) \\ y(t) &= h(x) \end{cases}$$

Recall that several methods are used for approximating the input/output behaviour of this system by a linear or bilinear one.

0.1 The Carleman method

The most famous method is the Carleman method [3], of which are knew two versions: the redundant version (called here *version (R)*, see [16]), and the non redundant version (called here *version (NR)*, [2]). (See also Rotella for a matrix description [15]). It consists in "bilinearizing" the given system, by two successive stages: The first step consists in writing a bilinear differential equation whose dimension is infinite, in the tensor algebra (for the redundant version), or respectively in the symmetrical algebra (for the non redundant version) of the state space. The second step consists in truncating this new system by choosing the tensor powers $x^{\otimes k}$ (resp. the symmetrical power $s^{[k]}$) whose degree is smaller than or equal to $k < N$. We obtain in that way a finite dimensional bilinear system which approximates (Σ) .

For a single system, we get the following truncated state equation:

$$\frac{d}{dt} \begin{pmatrix} x^{\otimes 1}(t) \\ \vdots \\ x^{\otimes N}(t) \end{pmatrix} = \begin{pmatrix} A_{1,1} & \dots & A_{1,N} \\ 0 & A_{2,2} & \dots & A_{2,N} \\ \vdots & \ddots & & \vdots \\ 0 & \dots & 0 & A_{N,1} \end{pmatrix} \begin{pmatrix} x^{\otimes 1}(t) \\ \vdots \\ x^{\otimes N}(t) \end{pmatrix} + \begin{pmatrix} B_{1,1} & B_{1,N-1} & 0 \\ B_{2,0} & B_{2,N-2} & 0 \\ 0 & \ddots & \vdots \\ \vdots & \ddots & \vdots \\ 0 & \dots & 0 & B_{N,0} & 0 \end{pmatrix} \begin{pmatrix} x^{\otimes 1}(t) \\ \vdots \\ x^{\otimes N}(t) \end{pmatrix} u + \begin{pmatrix} B_{1,0} \\ 0 \\ \vdots \\ 0 \end{pmatrix} u + \dots$$

$$y(t) = (c(t) \ 0 \ \dots \ 0) \begin{pmatrix} x^{\otimes 1}(t) \\ \vdots \\ x^{\otimes N}(t) \end{pmatrix},$$

where the state vector $x^{\otimes N}(t)$ has been "truncated" beyond the N^{i-th} tensor power $x^{\otimes N}(t)$. Hence we have set:

$$\begin{aligned} x^{\otimes 1} &= x, \\ \forall i \in \mathbb{N}^*, \quad x^{\otimes i} &= x^{\otimes(i-1)} \otimes x^{\otimes 1}. \end{aligned}$$

The Carleman representation, in the version (R), is redundant in the following sense: given a basis $(e_i)_{1 \leq i \leq n}$ of the state space, we have $x = \sum_{i=1}^n e_i s_i$. Thus $x \otimes x = \sum_{i,j} (e_i \otimes e_j) x_i x_j$ is a redundant expression, since we see that $x_i x_j$ is at once the coefficient of $e_i \otimes e_j$ and $e_j \otimes e_i$.

The non redundant Carleman representation, used by Brockett [2] in the version (NR), corresponds exactly to computations of Taylor series, and to the calculus in the symmetrical algebra of the state space. We illustrate this representation (NR) and this truncation method with the geometrical \mathcal{R} -automaton construction: the states of this geometric \mathcal{R} -automaton are the products $x_1^{i_1} x_2^{i_2} \dots x_N^{i_N}$ (commutative product, i.e. not redundant products). Then, this automaton is truncated by picking out the states whose degree is smaller than or equal to k , i. e. such that

$$\sum_{j=1}^N i_j \leq k,$$

and deleting the others states.

0.2 Finite Volterra expansions in an equilibrium point

As showed by Brockett [2], the Carleman method enables to compute exact Volterra kernels as described, if the system is initialized in an equilibrium point, by a bilinear system. An equivalent computation, using enumerative combinatorics, has recently been presented by Viennot [13], and by Lamnabhi-Lagarrigue [12]. Here we rediscover the same result by constructing the "geometrical \mathcal{R} -automaton": in that case, the autonomous vector field A_0 cannot increase the total degree $i_1 + i_2 + \dots + i_N = k$. And then, one need only to truncate the geometrical automaton by choosing the states $i_1 + i_2 + \dots + i_N$ whose total degree $\sum_{j=1}^N i_j$ is smaller than or equal to k . Thus we obtain a finite \mathcal{R} -automaton, and then a bilinear approximant.

If the system is not initialized in an equilibrium point, we have need change the approximation test : we have to keep any automaton path containing at most k edges labelled by the non autonomous vector fields of the systems. It is possible, but we may obtain an infinite automaton, and then we may obtain Volterra approximants which are no produced by a bilinear system. (The same test can also be used for the structural \mathcal{R} -automaton).

0.3 Padé-type approximants

If a differential system has no forcing term, a well known method consists in constructing the Padé-type approximant of the commutative generating series of the given system [1]. For a nonlinear system (Σ) , the corresponding method consists in computing a rational approximant of the generating series of (Σ) . To be more exact, we define a "Padé-type approximant's family" to be any iterative convergent construction of approximant rational series (in noncommutative variables). (see Leroux-Viennot [12]). We emphasize the generality of the following construction : any iterative method consisting in truncating an \mathcal{R} -automaton provides a family of non commutative Padé-type approximants of the generating series of (Σ) . For any associated bilinear system, a rational expression of its

generating series may be directly written, without any calculus, by reading the automaton. (This algorithm has been written in MACSYMA, see [11]).

We expect empirically that the bilinear systems associated to the Padé-type approximants will converge very rapidly. Unfortunately, we are not able to compute generally the input/output functional coded by the rational series.

0.4 General \mathcal{R} -automaton, and structural \mathcal{R} -automaton

Given an analytic system (Σ) , which is affine in the input variables, the purpose of this paper is to construct two representations of (Σ) , by defining two infinite dimensional bilinear \mathcal{R} -automata, and to truncate these automata in order to compute approximants of (Σ) , for order k .

The first construction illustrates the Carleman method. It consists in defining a "geometrical automaton", whose states $\{x_1^{i_1} x_2^{i_2} \dots x_N^{i_N}\}_{N \geq 0, i_j \geq 0}$ are powers of the state vector components, and whose transitions describe the left action of the system's vector fields.

The second construction consists in defining a "structural automaton", whose states are the partial derivatives $\{D_1^{i_1} D_2^{i_2} \dots D_N^{i_N}\}_{N \geq 0, i_j \geq 0}$, and whose transitions describe the right action of the system's vector fields.

In the two cases, we obtained so the graph description of an infinite bilinear system producing the same generating series G as the original analytical system (Σ) . Then, any finite truncation process gives the graphical description of a finite bilinear system approximating (Σ) , that can be directly written without any computation.

In this paper, as an illustration, we detail the truncation process that gives an approximating bilinear system whose generating series coincides with the generating series of (Σ) up to order k . And consequently, the output Taylor expansion of the analytical system (Σ) and the one of the approximating bilinear system have the same Taylor expansion up to order k .

1 Notations

We are interested in studying the dynamic system (Σ) :

$$(\sigma) \begin{cases} \dot{x}(t) = A_0(x) + \sum_{i=1}^m u_i(t) A_i(x) \\ y(t) = h(x) \end{cases}$$

where

- x belongs to a finite dimensional \mathcal{R} -analytic manifold Q ,
- vector fields A_0, A_1, \dots, A_m and output function h are defined and analytic in a neighbourhood of the initial state $x(0)$,

$$A_l = \sum_{j=1}^n \theta_l^j(x_1, \dots, x_n) \frac{\partial}{\partial x_j} \quad \text{for } 0 \leq l \leq m,$$

- u_1, \dots, u_m are piecewise continuous.

As shown by M. Fliess ([5]), the input/output behaviour of an analytical system can be coded by the following non commutative formal power series s , called "generating series" of the system:

$$s = h_{|x(0)} + \sum_{\nu \geq 0} \sum_{j_1, \dots, j_n=0}^m A_{j_1-1} A_{j_2-2} \cdots A_{j_n-\nu} h_{|x(0)} z_{j_1-1} \cdots z_{j_n-1}$$

with $Z = \{z_0, z_1, \dots, z_m\}$

Our problem consists in building rational series g approximating s up to a given order k , i.e.:

$$\text{ord}(s - g) > k.$$

We recall (cf. Schützenberger theorem [17]), that a power series $g \in \mathbb{R} \ll Z \gg$ is said to be *rational*, (or *regular*) if and only if it can be "recognized" (by a finite \mathbb{R} -automaton), that is there exists a \mathbb{R} -automaton \mathcal{A} over Z :

$$\mathcal{A} = (Q, q_0, l, \tau), \quad \text{where}$$

- Q is a finite dimensional \mathbb{R} -vector space,
- q_0 is the initial state,
- $\tau(z_i)$ is a linear map: $Q \rightarrow Q$,
- the "observation" l is a linear map: $Q \rightarrow \mathbb{R}$,

satisfying:

$$g = \sum_{z_{i_1} z_{i_2} \cdots z_{i_n} \in Z^*} l(\tau(z_{i_1}) \tau(z_{i_2}) \cdots \tau(z_{i_n}) q_0) z_{i_1} z_{i_2} \cdots z_{i_n}.$$

It is convenient to use a more compact notation. We note any finite product $z_{i_1} z_{i_2} \cdots z_{i_n}$, where z_{i_j} are letters in the "alphabet" Z , as a "word", denoted by a single symbol w , belonging to the free monoid Z^* over Z .

Another characterization of recognizable series has been given by Schützenberger, and by M. Fliess [5]: A power series g is recognizable if and only if the rank of its Hankel matrix

$$H(g) = [\langle g | uv \rangle]_{u, v \in Z^* \times Z^*}$$

is finite. These reminders show that the notions of *bilinear system behaviour*, of *rational generating series*, of *finite Hankel rank series* and of *finite \mathbb{R} -automaton recognizable series*, are strictly equivalent. Then, automaton theory is an adequate tool to find rational approximants [9]

To illustrate the connection between these notions, let us consider the following example:

Example 1:

Let $Z = \{z_1, z_2\}$. The following series is recognizable:

$$g = \sum_{w \in Z^*} |w|_{z_1} w.$$

Its Hankel matrix $H(g)$ is given by:

$$H(g) = \begin{pmatrix} & C_\epsilon & C_{z_1} & C_{z_1^2} & C_{z_1^3} & C_{z_1 z_2} & \dots \\ L_\epsilon & 0 & 1 & 0 & 2 & 1 & \dots \\ L_{z_1} & 1 & 2 & 1 & 3 & 2 & \dots \\ L_{z_2} & 0 & 1 & 0 & 2 & 1 & \dots \\ L_{z_1^2} & 2 & 3 & 2 & 4 & 3 & \dots \\ L_{z_1 z_2} & 1 & 2 & 1 & 3 & 2 & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix}$$

Note that $rk(g) = 2$ and that (C_ϵ, C_{z_1}) form a basis of the space of column vectors:

$$(R) = \begin{cases} C_w = C_\epsilon & \text{if } |w|_{z_1} = 0 \\ C_w = |w|_{z_1} \cdot C_{z_1} - (|w|_{z_1} - 1) \cdot C_\epsilon & \text{otherwise.} \end{cases}$$

Hence the automaton transitions are given by:

$$\begin{cases} z_1 \cdot C_\epsilon = C_{z_1} \\ z_1 \cdot C_{z_1} = C_{z_1^2} = 2 \cdot C_{z_1} - C_\epsilon \\ z_2 \cdot C_\epsilon = C_\epsilon \\ z_2 \cdot C_{z_1} = C_{z_2 z_1} = C_{z_1}. \end{cases}$$

By setting $\mu(z_1) = \begin{pmatrix} 0 & -1 \\ 1 & 2 \end{pmatrix}$ and $\mu(z_2) = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$, and $\mu(uv) = \mu(u)\mu(v)$ (for any $u, v \in X^*$). We find again the expression:

$$g = \sum_{w \in Z^*} (0 \ 1) \mu(w) \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

associated with the finite \mathbb{R} -automaton whose valued graph is:

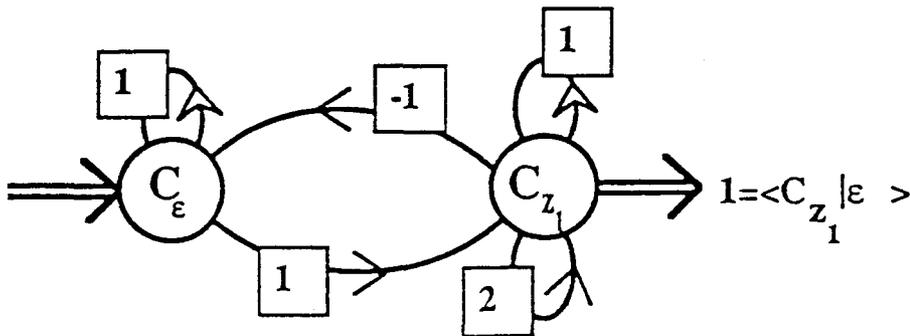


figure 1.
 → action of z_0
 → action of z_1

To illustrate the connection between the generating series of a dynamic system and the associated \mathbb{R} -automaton, let us consider the following example :

Exemple 2:

Consider the nonlinear differential equation relating the current excitation $i(t)$ and the voltage $v(t)$ across a capacitor:

$$\dot{v} = -k_1 v - k_2 v^2 + i(t),$$

$$\text{i.e. } \begin{cases} \dot{x} = A_0(x) + A_1(x)i(t) \\ v(t) = x(t) \end{cases}$$

$$\text{where } \begin{cases} A_0 = (-k_1 x - k_2 x^2) \frac{\partial}{\partial x} \\ A_1 = \frac{\partial}{\partial x} \end{cases}$$

By setting

$$a = -k_1 x - k_2 x^2, \quad b = -k_1 - 2k_2 x, \quad c = -2k_2,$$

and by computing the Hankel matrix $H(s)$, we note that the family $(C_{z_1^n})_{n \in \mathbb{N}}$ forms a basis of the column vector space. Moreover

$$C_{z_0 z_1^n} w = a C_{z_1^{n+1}} + \frac{n(n-1)}{2} c C_{z_1^{n-1}}$$

The graph associated to this infinite \mathbb{R} -automaton is :

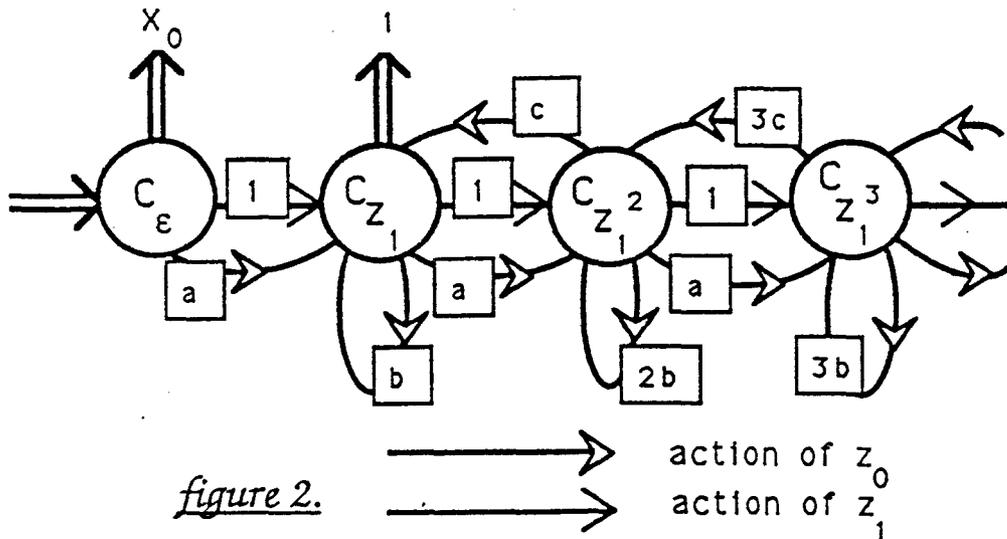


figure 2.

2 Geometrical \mathbb{R} -automaton

This automaton illustrates the Carleman method. Nevertheless, the construction is given in the state space of the symmetrical algebra, instead of the tensor algebra.

Given a dynamic system

$$(S) \quad \begin{cases} \dot{x} = A_0(x) + \sum_{i=1}^m u_i(t) A_i(x) \\ y(t) = h(x) \end{cases}$$

We can associate with it a "geometrical" \mathbb{R} -automaton $G = (E, V, \tau, l)$, where:

- E = the \mathbb{R} -vector space generated by the family $(x_1^{i_1} \cdots x_n^{i_n})_{i_1, \dots, i_n \geq 0}$
- V = the initial vector associated to the observation

$$h = \sum_{n \geq 0} \sum_{i_1, \dots, i_n \geq 0} h_{i_1, \dots, i_n} x_1^{i_1} \cdots x_n^{i_n}$$

- $\tau(z_r)$ is the linear map defined by the Lie derivatives:

$$x_1^{i_1} \cdots x_n^{i_n} \mapsto A_r \circ (x_1^{i_1} \cdots x_n^{i_n})$$

- l = the linear form defined by the row vector whose components are the evaluations of the final states evaluated at $t = 0$:

$$S = (1, x_{1|_0}, x_{2|_0}, \dots, x_{n|_0}).$$

The series which is recognized by this automaton is:

$$s = \sum_{w \in \mathbb{Z}^*} A_{\bar{w}} \circ h_{|_{x(0)}} \cdot w \\ \sum_{p \geq 0} \sum_{\nu_1, \dots, \nu_p \geq 0} l(\tau(z_{\nu_p}) \cdots \tau(z_{\nu_1} V)) z_{\nu_1} \cdots z_{\nu_p}.$$

Here we note \bar{w} the "mirror image" of $w = z_{\nu_1} z_{\nu_2} \cdots z_{\nu_p}$, that is: $\bar{w} = z_{\nu_p} z_{\nu_{p-1}} \cdots z_{\nu_1}$.

Exemple 3: Duffing equation.

$$\ddot{y} + ay + b^2 y + cy_3 = u(t)$$

By a direct translation, we obtain:

$$(S_1) \quad \begin{cases} \dot{x} = A_0(x) + u(t)A_1(x) \\ A_0 = F \frac{\partial}{\partial x_2} = FD_2 + x_2 D_1 \\ A_1 = \frac{\partial}{\partial x_2} = D_2 \\ y(t) = x_1(t), \end{cases}$$

by setting $F = -ax_2 - b_2 x_1 - cx_1^3$. Then we proceed as follows:

- Let E = the \mathbb{R} vector space generated by the family $(x_1^i x_2^j)_{i, j \geq 0}$.
- The initial vector $V = (0 \ 1 \ \dots)^t$.
- By computing the actions of the vector fields A_0 and A_1 on the states $x_1^i x_2^j$, we obtain :

$$A_0 \circ (x_1^i x_2^j) = -j a x_1^i x_2^j - j b^2 x_1^{i+1} x_2^{j-1} - j c x_1^{i+3} x_2^{j-1} + i x_1^{j-1} x_2^{j+1} \\ A_1 \circ (x_1^i x_2^j) = j x_1^i x_2^{j-1}.$$

This is nothing else as the description of the following "automaton cell":

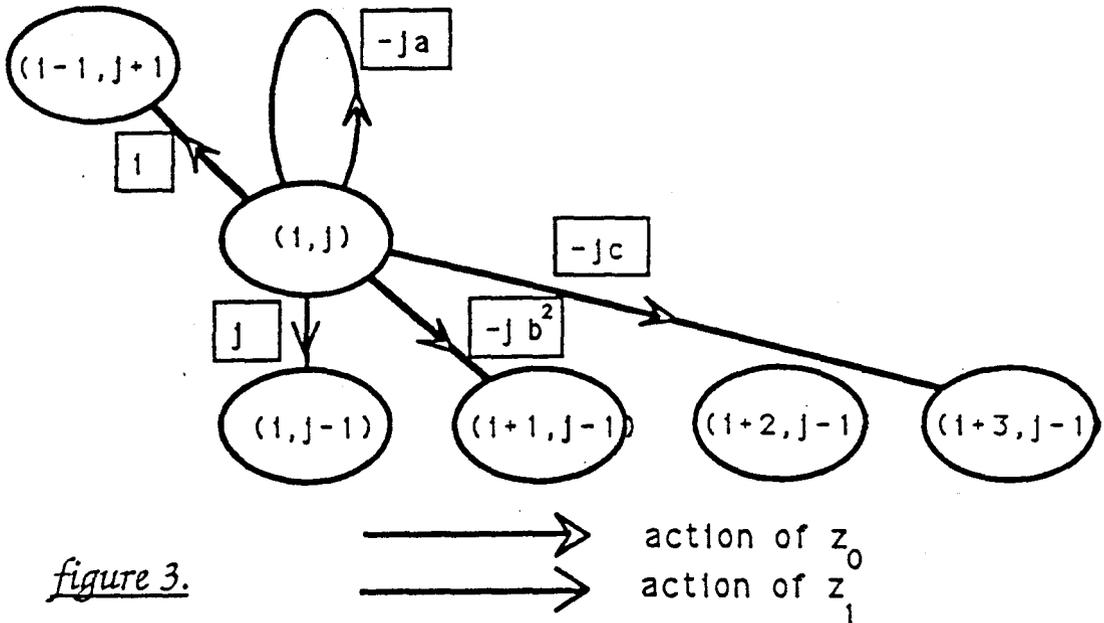


figure 3.

Thus we obtain a "geometrical" \mathbb{R} -automaton, described by a labelled graph having an infinite set of edges. Any finite truncation of this infinite graph furnishes a finite automaton, describing a bilinear dynamic system approximating the given nonlinear analytical system. A "Volterra truncation" is obtain by deleting the states $x_1^i x_2^j$ that satisfy $i + j < k$. In case $k = 3$, we obtain the following automaton, that recognizes exactly the order 3 Volterra approximant (plus higher order terms - a \mathbb{R} -automaton recognizing exactly the order 3 Volterra approximant can easily be obtained, see v.g. [7]).

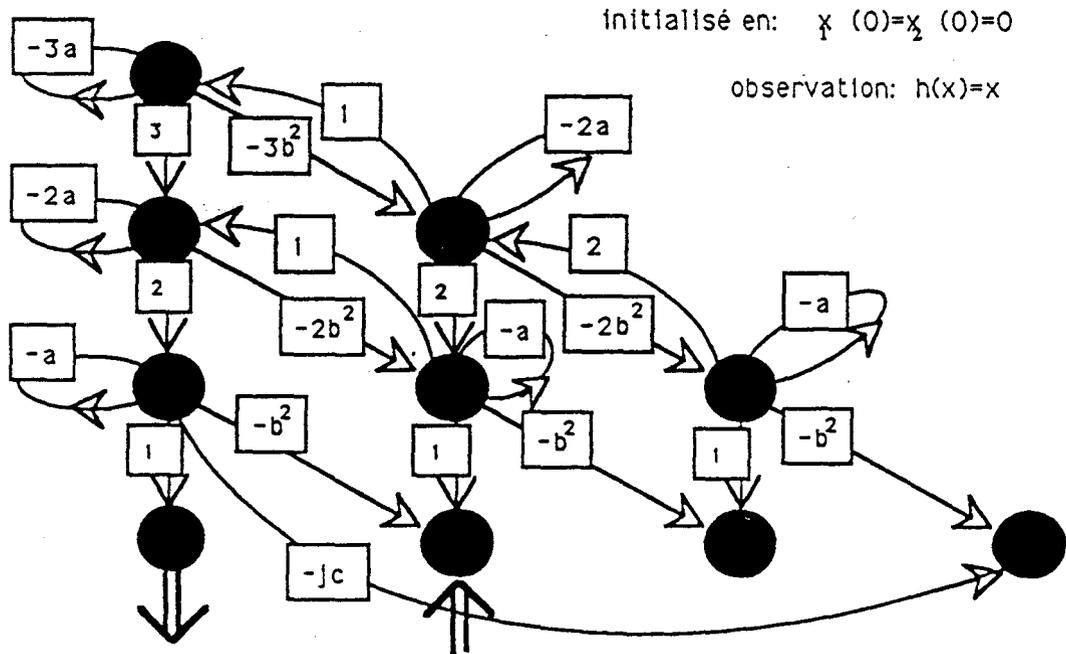


figure 4. Volterra automaton at the order 3.

The valued initial and terminal vertices are given as follows:

$$h = \sum c_{ij} x_1^i x_2^j \quad c_{ij} \Rightarrow \text{circle}(x_1^i x_2^j) \Rightarrow x_1^i x_2^j \Big|_0$$

figure 5. input and output values.

For the construction of a bilinear system A "Taylor truncation" at the order k is obtained as follows: For the order k , we pick out all the states of the form $x_1^i x_2^j$, which are met along some "successful path" of length lek . As illustration, we obtain for $k = 3$ the following truncated automaton:

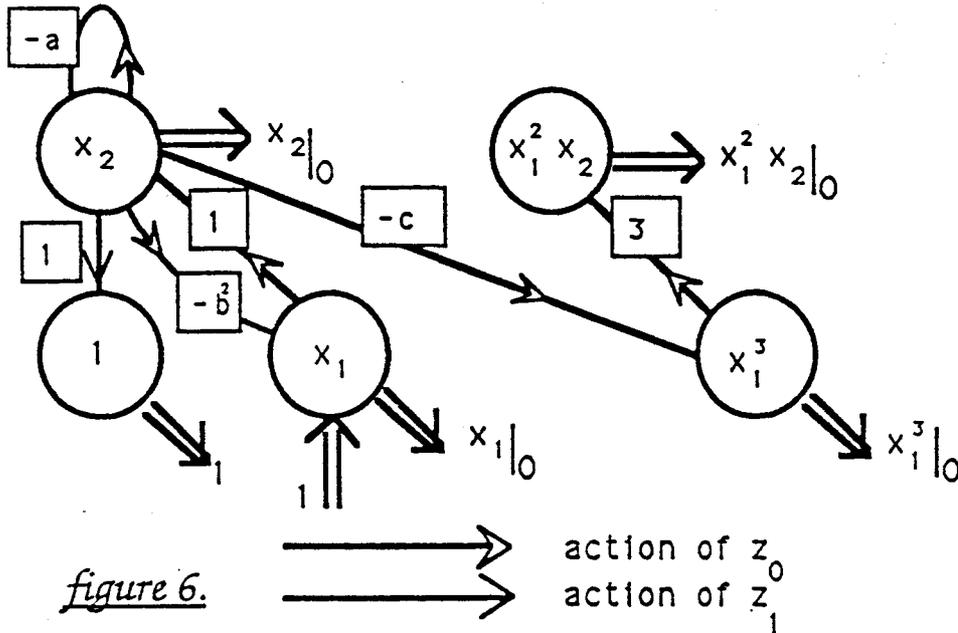


figure 6.

This automaton contains 5 states. It describes the bilinear system (B_1) whose state space dimension is equal to 5. Then (B_1) is a bilinear approximant of order $k = 3$ of the given nonlinear system. Indeed, the state equations of (B_1) are the following:

$$(B_1) \left\{ \begin{array}{l} \begin{pmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \\ v\dot{q}_4 \\ \dot{q}_5 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -b^2 & 0 & 0 \\ 0 & 1 & -a & 0 & 0 \\ 0 & 0 & -c & 0 & 0 \\ 0 & 0 & 0 & 3 & 0 \end{pmatrix} \begin{pmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \\ q_5 \end{pmatrix} + u(t) \begin{pmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \\ q_5 \end{pmatrix} \\ \lambda = (1 \quad x_{1|0} \quad x_{2|0} \quad x_{1^3|0} \quad x_{1^2 x_2|0}) \\ y(t) = \lambda q(t) \end{array} \right.$$

remark

In the Carleman method as presented by Rugh [16], a fictitious initial state is added, as we explain now.

Let us define an augmented state vector by adding a new component x_u subject to the condition:

$$x_u(t) = y(t).$$

Then the augmented system is describe as follows:

$$(S) \begin{cases} \dot{x} = A_0(x) + \sum_{i=1}^m u_i(t)A_i(x) \\ \dot{x}_u = \frac{d}{dt}h(x(t)) \\ \quad = A_0 \circ h + \sum_{i=1}^m u_i(t)(A_i \circ h), \\ \text{output: } s(t) = x_u(t). \end{cases}$$

The new associated $m + 1$ vector fields are:

$$\hat{A}_i = A_i + (A_i \circ h) \frac{\partial}{\partial x_u}.$$

In other words, this method consists in adding a new fictitious initial state to the automaton.

In the example 3 (Duffing equation), the new state equations are deduced from:

$$\begin{cases} A_0 = x_2 D_1 + F D_2 \\ A_1 = D_2 \\ y = h(x_1, x_2) = x_3, \end{cases} \quad \begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = F + u(t) \\ \dot{x}_3 = \frac{\partial h}{\partial x_1} x_2 + \frac{\partial h}{\partial x_2} (F + u(t)), \end{cases}$$

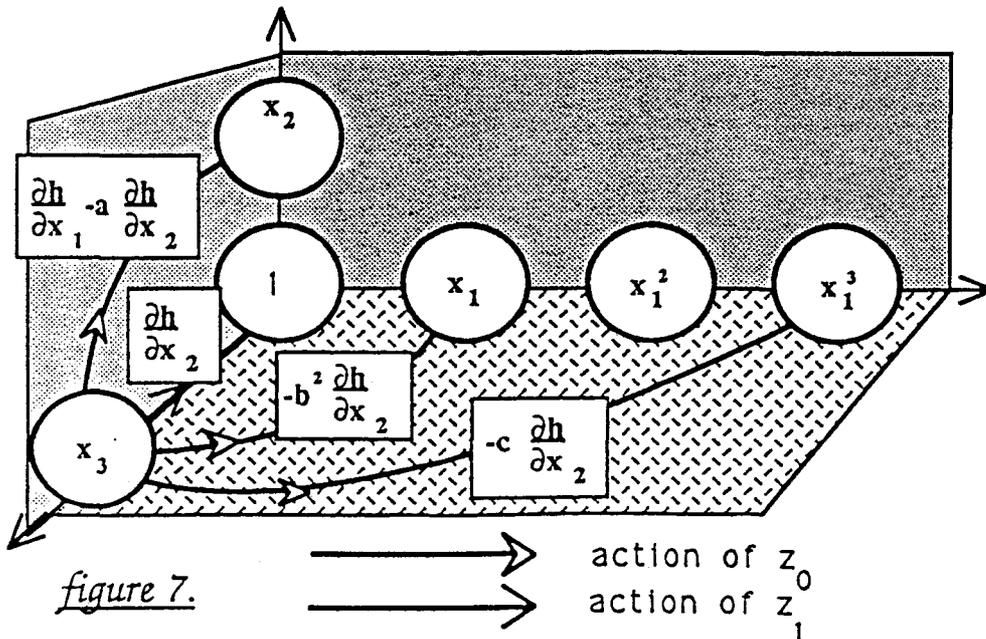
where $F = -ax_2 - b^2x_1 - cx_1^3$. We obtain the new augmented vector fields:

$$\begin{aligned} \hat{A}_0 &= x_2 D_1 + F D_2 + \left(\frac{\partial h}{\partial x_1} x_2 + \frac{\partial h}{\partial x_2} F \right) D_3 \\ \hat{A}_1 &= D_2 + \frac{\partial h}{\partial x_2} D_3. \end{aligned}$$

The new geometric \mathcal{R} -automaton holds the new state x_3 , in addition to the states $(x_1^i x_2^j)_{i,j \geq 0}$. The actions of the vector fields \hat{A}_0 and \hat{A}_1 of this new state are given by:

$$\begin{cases} \hat{A}_0 \circ x_3 = \frac{\partial h}{\partial x_1} + \frac{\partial h}{\partial x_2} F \\ \hat{A}_1 \circ x_3 = \frac{\partial h}{\partial x_2} \end{cases} \quad \text{with } F = -ax_2 - b^2x_1 - cx_1^3;$$

and are represented by the following graph:



3 The structural \mathcal{R} -automaton

With a given dynamic system

$$(\Sigma) \quad \begin{cases} \dot{x}(t) = A_0(x) + \sum_{i=1}^m u_i(t) A_i(x) \\ y(t) = h(x), \end{cases}$$

we associate a "structural \mathcal{R} -automaton" \mathcal{S} for characterizing the structure of the differential algebra defined by the system's vector fields. Let us note $D_i = \frac{\partial}{\partial x_i}$. In order to present the automaton \mathcal{S} , we introduce the multi-indices notation:

$$\begin{aligned} \text{if} \quad \alpha &= (i_1, i_2, \dots, i_n), \\ \text{we set } D^\alpha &= D_1^{i_1} D_2^{i_2} \dots D_n^{i_n}. \end{aligned}$$

Then the structural \mathcal{R} -automaton \mathcal{S} is the 4-uple $(\mathcal{F}, I, \tau, l)$, where

- \mathcal{F} is the \mathcal{R} -vector space generated by the operators D^α ,
- I is the initial state,
- $\tau(z_r)$ is the linear endomorphism of \mathcal{F} describing the right action of A_r on the D^α ,
- l is the linear form defined by the row vector whose α^{th} component is $D_\alpha \circ h$.

The action of the vector field $A_r = \sum_{j=1}^n (A_r \circ x_j) \frac{\partial}{\partial x_j}$ on D^α is given by the Leibniz formula:

$$D^\alpha A_r = \sum_{j=1}^n \sum_{\beta \leq \alpha} \binom{\alpha}{\beta} (D^\beta \circ A_r \circ x_j) D^{\alpha - \beta} D_j$$

where, if $\alpha = (i_1 \dots i_n)$ and $\beta = (j_1 \dots j_n)$, we have set:

$$\begin{aligned} \beta \leq \alpha &\Leftrightarrow j_1 \leq i_1, \quad j_2 \leq i_2, \dots, \quad j_n \leq i_n, \\ \text{and if } \beta \leq \alpha: \quad \binom{\alpha}{\beta} &= \binom{i_1}{j_1} \binom{i_2}{j_2} \dots \binom{i_n}{j_n} \end{aligned}$$

In the example 3 (Duffing equation),

- \mathcal{F} is the \mathcal{R} -vector space generated by $(D_1^i D_2^j)_{i,j \leq 0}$
- the linear form l is defined by the row vector

$$(x_1 \ 1 \ 0 \ \dots \ 0)$$

- the actions of A_0 and A_1 on the states $D_1^i D_2^j$ are given by:

$$\begin{aligned}
 D_1^i D_2^j A_1 &= D_1^i D_2^{j+1} \\
 D_1^i D_2^j A_0 &= F D_1^i D_2^{j+1} \\
 &+ \binom{i}{1} (D_1 F) D_1^{i-1} D_2^{j+1} + \binom{i}{2} (D_1^2 F) D_1^{i-2} D_2^{j+1} + \binom{i}{3} (D_1^3 F) D_1^{i-3} D_2^{j+1} \\
 &- j a D_1^i D_2^j + x_2 D_1^{i+1} D_2^j + j D_1^{i+1} D_2^{j-1}.
 \end{aligned}$$

Thus we obtain the following "automaton cell":

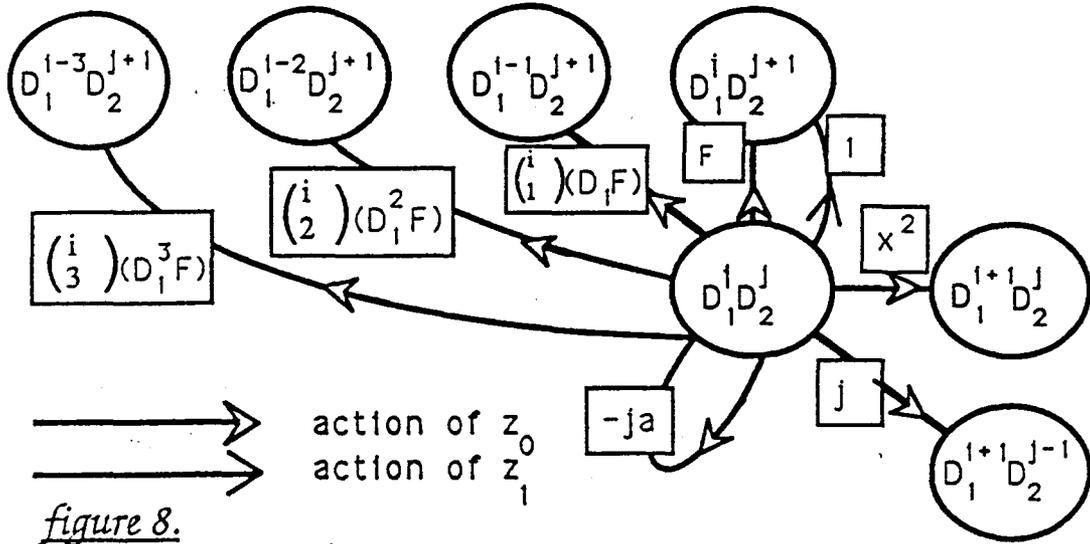


figure 8.

We construct the infinite structural R -automaton, and then we construct the truncated automaton by choosing the states $D_1^i D_2^j$ that are met along a successful path and the length of which is smaller than or equal to k .

For $k = 3$, we get the following truncated automaton:

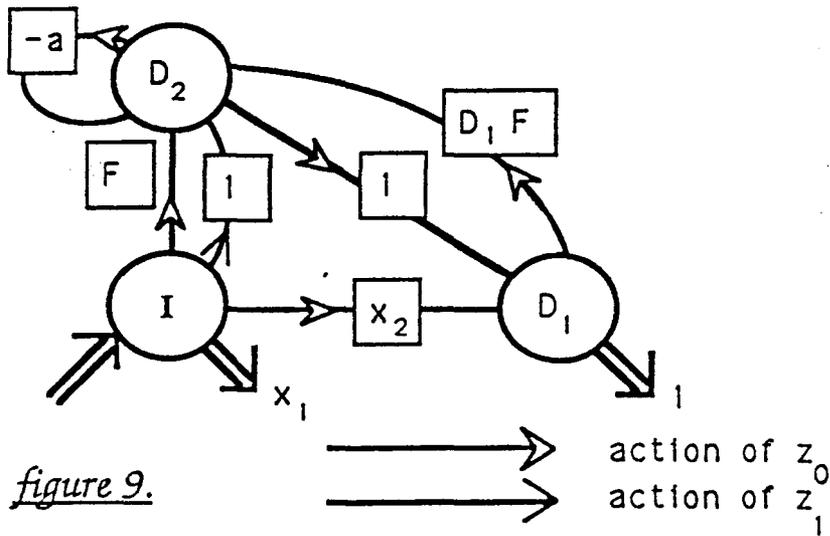


figure 9.

This automaton contains 3 states. It is associated with a bilinear system (\mathcal{B}_2) whose dimension is equal to 3. We deduce the state equation of (\mathcal{B}_2) :

$$(\mathcal{B}_2) \quad \left\{ \begin{array}{l} \begin{pmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 \\ x_{2|_0} & 0 & 1 \\ F_{|_0} & D_1(F)_{|_0} & a \end{pmatrix} \begin{pmatrix} q_1 \\ q_2 \\ q_3 \end{pmatrix} + \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} q_1 \\ q_2 \\ q_3 \end{pmatrix} u(t) \\ y(t) = \lambda q(t) \\ \lambda = \begin{pmatrix} x_{1|_0} & 1 & 0 \end{pmatrix}. \end{array} \right.$$

4 Conclusion

These two methods allow us to compute a bilinear approximant of the given system (S), at the order k . We emphasize that when we wish to calculate a bilinear approximant, for the order $k + 1$, the earlier calculations need not be done again.

These two technics are based on a criterion which is intrinsic and independant of the state representation: the equality of the generating series of the given system and of the approximating bilinear system, up to order k . Because of this remark, these technics defer from the Carleman method.

The geometrical automaton provides a bilinear approximated system, but its dimension is finite if and only if the observation is polynomial.

The structural automaton is a little more difficult to compute. But this method generally provides a bilinear approximated system whose dimension is smaller. But above all, it allows to compute in any case a bilinear approximant system, with an exact representation of the observation.

The algorithm consisting in computing the structural automaton has been implemented in MACSYMA [10].

References

- [1] Brezinski C., *Padé-type approximants and general orthogonal polynomials*, INSM, vol. 50, Birkhäuser Verlag, basel, 1980.
- [2] Brockett R.W., *Volterra series and Geometric Control Theory*, Automatica, vol.12, p. 167-176, 1976.
- [3] Carleman T., *Application de la théorie des équations intégrales singulières aux équations différentielles de la dynamique*, T. Ark. Mat. Astron. Fys. 22B, 1, 1932.
- [4] Flajolet P., *Combinatorial aspects of continued fractions*, Discrete Math., n 32, p. 125-161, 1980.
- [5] Fliess M., *Fonctionnelles causales non linéaires et indéterminées non commutatives*, Bull. Soc. Math. France, 109, p. 3-40, 1981.

- [6] Fliess M., Lamnabhi M., Lamnabhi-Lagarrigue, *An algebraic approach to nonlinear functional expansions*, IEEE Trans. Circuits and Systems, vol. CAS-30, n° 8, p. 554-570, 1983.
- [7] Guyon C., Hoang Ngoc Minh, *Séries de Volterra finies, séries rationnelles*, Technical report, LIFL , Lille, 1991.
- [8] Hespel C., Jacob G., *Calcul des approximations locales bilinéaires de systèmes analytiques*, RAIRO APII, vol.23, p. 331-349, 1989.
- [9] Hespel C., Jacob G., *Rational approximations and automata*, Technical reports LANS n 29, INSA, RENNES, 1990.
- [10] Hespel C., Jacob G., *Algorithme d'approximation bilinéaire de systèmes dynamiques par automate structurel*, Technical report LANS n 31, INSA, RENNES, 1991.
- [11] Hespel C., *Approximation of nonlinear dynamics systems by rational series*, to appear in "Algebraic and Computing Treatment of Noncommutative Power Series" (ed. G. Jacob and C. Reutenauer), Theoretical Computer Science, 1991. Control Theory, Birkhäuser, 1990.
- [12] Lamnabhi-Lagarrigue F., Leroux P., Viennot X.G., *Combinatorial interpretation of Volterra series by bilinear approximations*, to appear in "Analyse des systèmes dynamiques contrôlés", Progress in Systems and Control Theory, Birkhäuser, 1991.
- [13] Leroux P., Viennot X.G., *A combinatorial approach to nonlinear functional expansions: an introduction with example*, to appear in "Algebraic and Computing Treatment of Noncommutative Power Series" (ed. G. Jacob and C. Reuterauer), Theoretical Computer Science, 1991.
- [14] Rotella F., *Méthodes algébriques et analytiques pour la simplification et la commande de systèmes bilinéaires à deux dynamiques*, Thèse, Université des Sciences et Techniques de Lille, 1987.
- [15] Rotella F., *Generating series for analytic systems*, First european conference on "Algebraic Computing in control", Paris, march 13-15,1991.
- [16] Rugh W.J., *Nonlinear system theory*, The John Hopkins university press, Baltimore and London.
- [17] Schützenberger M.P., *On the definition of a family of automata*, Information and Control, vol. 4, p. 245-270, 1961.

Annexe 6

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

Reprinted from

MATHEMATICS AND COMPUTERS IN SIMULATION

Mathematics and Computers in Simulation 42 (1996) 639

Introduction
Dynamical systems

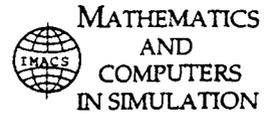


ELSEVIER



ELSEVIER

Mathematics and Computers in Simulation 42 (1996) 641–657



Iterated derivatives of the output of a nonlinear dynamic system and Faà di Bruno formula

Christiane Hespel¹

INSA Laboratoire LANS, 20 avenue des Buttes de Coësmes, 35043 Rennes Cédex, France

Abstract

The Faà di Bruno formula enables us to compute the derivatives of a function of several variables. We show here, in spite of the noncommutativity of causal computations, that the Faà di Bruno grammar makes it possible to compute the derivatives of any causal functional. Using this property and some computations on derivatives of causal functionals, this paper presents the first step towards solving the problem of “Exact Algebraic Identification”. This problem consists in computing the coefficients of a noncommutative generating series when only the Taylor expansion of some inputs (at $t = 0$), and the Taylor expansion (at $t = 0$) of associated outputs are known.

Keywords: Nonlinear systems; Generating series; Algebraic identification; Combinatorics; Syntactic methods

1. Introduction

Consider the nonlinear analytical system affine in the control:

$$(\Sigma) \quad \dot{q} = g_0(q) + \sum_{i=1}^m a_i(t)g_i(q), \quad y(t) = h(q),$$

where the state q belongs to an \mathbb{R} -analytic manifold Q , the dimension of which is finite: N ; the vector fields g_i and scalar ‘observation function’ h are analytical and defined in a neighbourhood of $q(0) = q_0$; and the inputs a_i are real and piecewise continuous.

The input/output behaviour of an analytical system may be obtained by using its generating series (i.e. the Fliess’ series). This formal series in noncommuting variables is defined on the “encoding alphabet” $=\{z_0, z_1, \dots, z_m\}$ corresponding to the system inputs. Then the output is obtained by interpreting words over \mathcal{Z} like iterated integrals of inputs. According to M. Fliess, every noncommutative series defines an input’s causal functional, by using the same method.

We are interested in solving the converse problem: Can we exactly identify the coefficients of a generating series of a system (Σ) , if the Taylor expansions (at $t = 0$) of the system inputs $y(t)$ and the Taylor expansions

¹ E-mail: hespel@nimbus.irisa.fr.

(at $t = 0$) of associated output are known. With this aim in view, we compute the iterated derivatives of $y(t)$. They are obtained as linear combinations of input multiderivatives, the coefficients of which are also linear combinations of the coefficients of the generating series.

After a few preliminary remarks, in the next section we compute the k th derivative of y in terms of input and n th derivatives of inputs, using a combinatorial method. So, the formula of Lamnabhi and Crouch [17] is redemonstrated. (For a combinatorial interpretation and proof, see [18]).

Then we recall the “Faà di Bruno formula”, providing the iterated derivatives of a composite function

$$y = \Psi(\xi_0(t), \xi_1(t), \dots, \xi_m(t)).$$

Using a syntactic method [2], a generalization of the Faà di Bruno formula is proven, providing the iterated derivatives of a causal functional

$$y = \Psi(\xi_0, \xi_1, \dots, \xi_m),$$

ξ_j being primitives of inputs a_j .

In Section 5, we give an algorithm for identifying *the coefficients of input multiderivatives in the generating series*. The problem of algebraic identification will be solved when we are able to identify the coefficients of the generating series, from the coefficients of input multiderivatives. This last problem consists in solving an infinite-dimensional linear equation system, that we hope to triangularize.

The algebraic identification may be used as a starting point for bilinear approximation [11–13], minimal realisation [19], and verifications via simulation methods [1]. It should be noticed that these combinatorial and syntactic methods are suitable for symbolic computation.

2. Preliminaries and notations

For introduction and general setting, see [8,9,16,20].

We associate with the system (Σ) the encoding alphabet $\mathcal{Z} = \{z_0, z_1, \dots, z_m\}$. Let us recall the definition: a *word* w in \mathcal{Z}^* is a finite sequence $z_{i_1} \cdots z_{i_p}$ of letters in \mathcal{Z} . The *empty word* is denoted by ε .

A *formal series* S (in the noncommuting variables z_i) with coefficients in \mathbb{R} is a mapping of \mathcal{Z}^* into \mathbb{R} denoted by

$$S = \sum_{w \in \mathcal{Z}^*} \langle S|w \rangle w.$$

The series 0 and 1 are defined by

$$\langle 0|w \rangle = 0 \text{ and } \langle 1|w \rangle = \delta_\varepsilon^w \quad \forall w \in \mathcal{Z}^*$$

(where δ_ε^w is the Kronecker symbol).

The operations of sum and Cauchy product of two series S, T are defined by

$$S + T = \sum_{w \in \mathcal{Z}^*} (\langle S|w \rangle + \langle T|w \rangle) w, \quad S \cdot T = \sum_{w \in \mathcal{Z}^*} \sum_{uv=w} \langle S|u \rangle \cdot \langle T|v \rangle w$$

(the last series is well defined because the set $\{(u, v) | uv = w\}$ is finite). With these operations, the set of formal series on \mathcal{Z} is an associative algebra denoted by $\mathbb{R}\langle\langle \mathcal{Z} \rangle\rangle$.

– Let us define *differential operators* as follows:

$$\mathcal{Y}(z_i) = \sum_{s=1}^N g_i^s(q) \frac{\partial}{\partial q^s} \quad \forall i, 0 \leq i \leq m,$$

and then for every word w over the encoding alphabet \mathcal{Z} , we define recursively the differential operator $\mathcal{Y}(w)$ as follows (denoting by \circ the composition of operators):

$$\mathcal{Y}(vz_j) = \mathcal{Y}(v) \circ \mathcal{Y}(z_j) \quad \forall z_j \in \mathcal{Z}, \quad \forall v \in \mathcal{Z}^*.$$

Thus for any word $w = z_{i_1} z_{i_2} \cdots z_{i_p}$ and any analytical function h we have

$$\mathcal{Y}(w) \circ h = g_{i_1} g_{i_2} \cdots g_{i_p} \circ h,$$

where $g_i \circ h$ is the Lie derivative of h along the vector field g_i .

– Similarly, by setting $a_0(t) \equiv 1$ for any t , the *iterated integral* $\int_0^t \delta_a(w)$ of the word w for input a is recursively defined as follows:

$$\int_0^t \delta_a(\varepsilon) = 1,$$

$$\int_0^t \delta_a(vz_i) = \int_0^t \left(\int_0^\tau \delta_a(v) \right) a_i(\tau) d\tau \quad \forall z_j \in \mathcal{Z} \quad \forall v \in \mathcal{Z}^*.$$

According to Chen’s notation [3–5], the order used here is the *reverse of Fliess’ one* [10]. Also according to Fliess fundamental formula, we get

$$y(t) = \sum_{w \in \mathcal{Z}^*} \left[\mathcal{Y}(w) \circ h_{|_{q_0}} \right] \int_0^t \delta_a(w).$$

Every term of this series is indexed by a word in \mathcal{Z}^* . Further we introduce naturally the following formal series: Fliess series and Chen series as

$$G = \sum_{w \in \mathcal{Z}^*} \left[\mathcal{Y}(w) \circ h_{|_{q_0}} \right] w \text{ is the } \textit{generating series}, \text{ or } \textit{‘Fliess series’} \text{ of the system } (\Sigma);$$

$$C_a(t) = \sum_{w \in \mathcal{Z}^*} \left[\int_0^t \delta_a(w) \right] w \text{ is the } \textit{‘Chen series’} \text{ of the input } a \text{ (see [3–5]). It is often convenient to set}$$

$$\xi_i(t) = \int_0^t a_i(\tau) d\tau.$$

From the previous definitions, we deduce

$$y(t) = \sum_{w \in \mathcal{Z}^*} \langle G|w \rangle \langle C_a(t)|w \rangle = \langle G||C_a(t) \rangle$$

(where $\|$ means infinite sum).

It should be noticed that $y(t)$ depends not only on the $\xi_i(t)$, but also on the values of the ξ_i in the whole interval $[0, t]$. For that reason, the generating series G defines “a causal functional” of the functions ξ_i [9].

More generally, any noncommuting power series H over \mathcal{Z} , coefficients of which satisfy the following “convergence condition”:

$$\exists K, L \in \mathbb{R}^+, \quad |\langle H|w \rangle| \leq K|w|!L^{|w|} \quad \forall w \in \mathcal{Z}^*,$$

can actually be viewed as causal functional defining the output

$$y_H(t) = \langle H \| C_a(t) \rangle = \sum_{w \in \mathcal{Z}^*} \langle H|w \rangle \langle C_a(t)|w \rangle.$$

This sum is absolutely convergent in some neighbourhood of zero [10].

3. Iterated derivatives of the output

For any formal series G with constant coefficients (and satisfying the convergence condition), the i th time derivative of y_G can be written as

$$y_G^{(i)}(t) = \langle G \| C_a^{(i)}(t) \rangle \quad \forall i \in \mathbb{N},$$

and $y_G^{(i)}(t)$ is absolutely convergent in some neighbourhood of zero.

The first step consists in computing the iterated derivatives $C_a^{(i)}(t)$ of Chen’s series. By using an easy combinatorial method, two formulas for the iterated derivatives of the output of a causal functional are obtained. The first one has already been obtained by F. Lamnabhi and P.E. Crouch, using another proof.

3.1. Derivatives of Chen series (intrinsic computation)

Lemma 1. Let A be $\sum_{0 \leq i \leq m} a_i \cdot z_i$. Then the derivative of the Chen series is $(d/dt)C_a = C_a \cdot A$.

Proof. By recursive definition of iterated integrals, the derivatives of the coefficients of the Chen series satisfy, for any $z_i \in \mathcal{Z}$ and $v \in \mathcal{Z}^*$:

$$\frac{d}{dt} \langle C_a | v z_i \rangle = \left[\int_0^t \delta_a v \right] a_i = \langle C_a | v \rangle a_i.$$

We deduce

$$\sum_{i,v} \frac{d}{dt} \langle C_a | v z_i \rangle v z_i = \sum_{i,v} \langle C_a | v \rangle a_i v z_i.$$

That is exactly

$$\frac{d}{dt} (C_a) = C_a \sum_i a_i z_i. \quad \square$$

Thereby we obtain

$$\begin{aligned} C_a^{(1)} &= C_a A, \\ C_a^{(2)} &= C_a^{(1)} A + C_a A^{(1)} = C_a A A + C_a A^{(1)}, \\ C_a^{(3)} &= C_a^{(1)} A A + C_a A^{(1)} A + C_a A A^{(1)} + C_a^{(1)} A^{(1)} + C_a A^{(2)}, \\ &= C_a A A A + C_a A^{(1)} A + 2C_a A A^{(1)} + C_a A^{(2)}. \\ &\vdots \end{aligned}$$

These relations can also be written as

$$C_a^{(i)} = C_a A_i, \quad \text{where } A_1 = A, \quad A_{i+1} = A A_i + D_t A_i, \quad (R)$$

where D_t is the usual time derivation operator.

It should be noticed that $C_a(0) = 1$ and then $C_a^{(i)}(0) = A_i(0)$. Consequently

$$y^{(i)}(0) = \sum_{w \in Z^*} \langle G|w \rangle \langle C_a^{(i)}(0)|w \rangle = \sum_{w \in Z^*} \langle G|w \rangle \langle A_i(0)|w \rangle = \langle G|A_i(0) \rangle,$$

where A_i is given by the recursive formula (R).

3.2. Explicit computation of A_i

Let $\rho = (\rho_1, \rho_2, \dots, \rho_p)$ be a multi-index of derivation. We define its degree and its weight by

$$\text{deg}(\rho) = p, \quad \text{wgt}(\rho) = \sum_{j=1}^p (1 + \rho_j) = \text{deg}(\rho) + \sum_{j=1}^p \rho_j.$$

We denote

$$A^{(\rho_k)} = \sum_i a_i^{(\rho_k)} z_i \quad \text{and} \quad A^{[\rho]} = A^{(\rho_1)} \dots A^{(\rho_p)}.$$

In other words

$$A^{[\rho]} = \sum_{i_1, i_2, \dots, i_p} a_{i_1}^{(\rho_1)} a_{i_2}^{(\rho_2)} \dots a_{i_p}^{(\rho_p)} z_{i_1} z_{i_2} \dots z_{i_p}.$$

Consequently, we obtain

$$A_i = \sum_{\text{wgt}(\rho)=i} \alpha_\rho A^{[\rho]},$$

where we need the computation of constants α_ρ .

We denote $\mathcal{A} = \sum_{i \in \mathbb{N}} A_i$. Then \mathcal{A} satisfies

$$\mathcal{A} = 1 + A\mathcal{A} + D\mathcal{A} = 1 + \sum_\rho \alpha_\rho A A^{[\rho]} + \sum_\rho \sum_{j=1}^{\text{deg}(\rho)} \alpha_\rho A^{(\rho_1)} \dots A^{(\rho_{j-1})} A^{(\rho_j+1)} A^{(\rho_{j+1})} \dots A^{(\rho_{\text{deg}(\rho)})}.$$

Splitting the computation of ρ into two cases: $\rho_1 \neq 0$, and $\rho_1 = 0$, enables us to write:

Case 1. $\rho_1 \neq 0$ and $\text{deg}(\rho) = p$.

$$\alpha_\rho = \sum_{j=1}^{\text{deg}(\rho)} \alpha_{(\rho_1, \rho_2, \dots, \rho_{j-1}, \rho_j-1, \rho_{j+1}, \dots, \rho_p)} = \sum_{j=1}^{\text{deg}(\rho)-1} \alpha_{(\rho_1, \rho_2, \dots, \rho_{j-1}, \rho_j-1, \rho_{j+1}, \dots, \rho_p)} + \alpha_{(\rho_1, \rho_2, \dots, \rho_p-1)}$$

Case 2. $\rho_1 = 0$ and $\text{deg}(\rho) = p$.

$$\begin{aligned} \alpha_\rho &= \alpha_{(\rho_2, \rho_3, \dots, \rho_p)} + \sum_{j=2}^{\text{deg}(\rho)} \alpha_{(\rho_1, \dots, \rho_{j-1}, \rho_j-1, \rho_{j+1}, \dots, \rho_p)} \\ &= \alpha_{(\rho_2, \rho_3, \dots, \rho_p)} + \sum_{j=2}^{\text{deg}(\rho)-1} \alpha_{(\rho_1, \dots, \rho_{j-1}, \rho_j-1, \rho_{j+1}, \dots, \rho_p)} + \alpha_{(\rho_1, \rho_2, \dots, \rho_p-1)}. \end{aligned}$$

Let us write $\rho = (\check{\rho}, \rho_p)$ with $\check{\rho} = (\rho_1, \dots, \rho_{p-1})$.

By setting

$$\alpha_{(\check{\rho}, \rho_p)} = \alpha_{\check{\rho}} \psi(\text{wgt}(\rho), \rho_p),$$

we prove that ψ must be written in the form

$$\psi(\text{wgt}(\rho), \rho_p) = \psi(\text{wgt}(\rho) - 1, \rho_p) + \psi(\text{wgt}(\rho) - 1, \rho_p - 1) \quad \forall \rho_p > 0.$$

Note particularly that $\psi(2, 1) = 1$. Also we obtain a recursive scheme of binomial coefficients.

If $\rho_p = 0$, we obtain

$$\psi(\text{wgt}(\rho), 0) = \psi(\text{wgt}(\rho) - 1, 0),$$

then

$$\psi(\text{wgt}(\rho), 0) = 1.$$

If $\rho_p = 1$, we get

$$\psi(\text{wgt}(\rho), 1) = \psi(\text{wgt}(\rho) - 1, 1) + \psi(\text{wgt}(\rho) - 1, 0) = \psi(2, 1) + \text{wgt}(\rho) - 2 = \text{wgt}(\rho) - 1,$$

consequently,

$$\psi(\text{wgt}(\rho), \rho_p) = \binom{\text{wgt}(\rho) - 1}{\rho_p}.$$

Moreover since $\text{wgt}(\rho) = p + \rho_1 + \dots + \rho_p$,

$$\alpha_\rho = \prod_{i=1}^p \binom{\sum_{j=1}^i \rho_j + i - 1}{\rho_i} = \binom{\rho_1}{\rho_1} \binom{\rho_1 + \rho_2 + 1}{\rho_2} \dots \binom{\rho_1 + \dots + \rho_k + k - 1}{\rho_k}. \tag{E}$$

3.3. Local and global formulas and Lie derivatives

In this section, we will prove two formulas giving the n th derivative of the output, at instant t : the first one [17] is *local* and defines $y^{(n)}(t)$ over the observation algebra, at state $q(t)$, the other is *global*, and original. It defines $y^{(n)}(t)$ in terms of observation functions at state $q(0)$.

Let us denote by $v \triangleleft H$ the “left remainder” of a series H by a word v by setting

$$\forall u \in \mathcal{Z}^*, \quad \langle v \triangleleft H | u \rangle = \langle H | uv \rangle.$$

If v (respectively w) is a letter (respectively a word), this definition is equivalent to the usual one, expressing $z_i \triangleleft w$ as the left remainder by simplifying w to the right by the suffix z_i :

$$z_i \triangleleft \varepsilon = 0, \quad z_i \triangleleft (vz_j) = \delta_i^j v, \quad (z_j \in \mathcal{Z}, v \in \mathcal{Z}^*).$$

By extending naturally this operation to power series, we obtain the following property:

$$\forall v \in \mathcal{Z}^*, \quad \langle v \triangleleft H | K \rangle = \langle H | Kv \rangle.$$

The geometric interpretation is the following: if H is the generating function of the system (Σ) with observation function h , then the left remainder $w \triangleleft H$ of the series H by the word $w = z_{i_1} z_{i_2} \cdots z_{i_k}$ is the generating series of (Σ) with the new following observation function:

$$g_{i_1} g_{i_2} \cdots g_{i_k} \circ h.$$

Then according to Section 3.1, and the above definition of $A_i(t)$, we can write

$$y^{(n)}(t) = \langle G | C_a^{(n)}(t) \rangle = \langle G | C_a(t) A_n(t) \rangle = \langle A_n(t) \triangleleft G | C_a(t) \rangle.$$

On the other hand,

$$A_n = \sum_{\text{wgt}(\rho)=n} \alpha(\rho) A^{(\rho)},$$

then

$$A_n \triangleleft G = \sum_{\text{wgt}(\rho)=n} \alpha(\rho) \sum_{i_j} a_{i_1}^{(\rho_1)} \cdots a_{i_k}^{(\rho_k)} (z_{i_1} \cdots z_{i_k} \triangleleft G).$$

Consequently,

$$\begin{aligned} y^{(n)}(t) &= \sum_{\text{wgt}(\rho)=n} \alpha(\rho) \sum_{i_j} a_{i_1}^{(\rho_1)}(t) \cdots a_{i_k}^{(\rho_k)}(t) \langle z_{i_1} \cdots z_{i_k} \triangleleft G | C_a(t) \rangle \\ &= \sum_{\text{wgt}(\rho)=n} \alpha(\rho) \sum_{i_j} a_{i_1}^{(\rho_1)}(t) \cdots a_{i_k}^{(\rho_k)}(t) \langle G | C_a(t) z_{i_1} \cdots z_{i_k} \rangle. \end{aligned}$$

For an *analytical system* (Σ) , this formula may be interpreted by two means:

- Initializing the system (Σ) at time t (and then putting $C_a(t) = 1$), we obtain the *local expression* of $y^{(n)}(t)$ in the state $q(t)$:

$$y^{(n)}(t) = \sum \sum \alpha(\rho) a_{i_1}^{(\rho_1)}(t) \cdots a_{i_k}^{(\rho_k)}(t) (g_{i_1} \cdots g_{i_k} \circ h)_{|q(t)}.$$

(This is the formula of Lamnabhi-Lagarrigue and Crouch [17]).

– Initializing the system (Σ) at time 0, we obtain the *global expression* of $y^{(n)}(t)$ in the initial state $q(0)$

$$y^{(n)}(t) = \sum \sum \alpha(\rho) a_{i_1}^{(\rho_1)} \cdots a_{i_k}^{(\rho_k)} y_{i_1 i_2 \dots i_k}(t),$$

where $y_{i_1 i_2 \dots i_k}$ is the output of system (Σ), initialized in $q(0)$, for the observation function $g_{i_1} g_{i_2} \cdots g_{i_k} \circ h$. For $t = 0$, according to $C_a(0) = 1$, we obtain the formula

$$y^{(n)}(0) = \sum_{k=0}^n \sum_{i_j} \langle G | z_{i_1} \cdots z_{i_k} \rangle \times \sum_{\rho_1 + \dots + \rho_k = n-k} \binom{\rho_1}{\rho_1} \binom{\rho_1 + \rho_2 + 1}{\rho_2} \cdots \binom{\rho_1 + \dots + \rho_k + k - 1}{\rho_k} a_{i_1}^{(\rho_1)}(0) \cdots a_{i_k}^{(\rho_k)}(0). \quad (F)$$

3.4. Expression on inputs' differential monomials

Any multiderivative $a^{(\rho)}$ can also be viewed as a differential monomial in the variables a_i . These monomials will be denoted by a^μ . Thus, by expressing each $A_n(t)$ over input components, we get a polynomial over \mathcal{Z} , coefficients of which are differential polynomials over a_i . We shall write

$$A_n = \sum_{\text{wgt}(\rho)=n} \alpha(\rho) A^{[\rho]} = \sum_{\mu} l_{\mu} a^{\mu},$$

and we intend to compute the noncommutative polynomials l_{μ} .

If we consider that input components can be freely chosen (especially if the system is “without drift”) this interpretation preserves the differential weights. It is not the case if some input a_0 is not free, but satisfies $a_0(t) \equiv 1$ (case of a system “with drift”).

3.4.1. Homogeneous case: single input system without drift

In case of a single input without drift, we can write

$$\langle A_n | z_1^p \rangle = \sum_{\text{wgt}(\rho)=n} \alpha(\rho) a_1^{(\rho_1)} a_1^{(\rho_2)} \cdots a_1^{(\rho_p)},$$

and we have

$$a_1^{(\rho_1)} a_1^{(\rho_2)} \cdots a_1^{(\rho_p)} = a_1^{\mu_1} (a_1^{(1)})^{\mu_2} \cdots (a_1^{(k-1)})^{\mu_k} = a_1^{\mu}.$$

So we call differential multiplicity of a_1^{μ} the finite integer sequence

$$\mu = (\mu_1, \mu_2, \dots, \mu_k).$$

We define degree and weight of μ as follows:

$$p = \text{deg}(\rho) = \text{deg}(\mu) = \sum_{i=1}^k \mu_i, \quad \sum_{i=1}^k (1 + \rho_i) = \text{wgt}(\rho) = \text{wgt}(\mu) = \sum_{i=1}^k i \mu_i.$$

If we call *development* of μ the sequence

$$\text{dev}(\mu) = (\underbrace{0, \dots, 0}_{\mu_1 \text{ times}}, \underbrace{1, \dots, 1}_{\mu_2 \text{ times}}, \dots, \underbrace{k-1, \dots, k-1}_{\mu_k \text{ times}}),$$

we observe that $a^{(\rho)} = a^\mu$ iff ρ is a permutation of the sequence $\text{dev}(\mu)$ (notation: $\rho \in \mathcal{S}(\text{dev}(\mu))$). Then we obtain

$$\langle l_\mu | z_1^p \rangle = \sum_{\rho \in \mathcal{S}(\text{dev}(\mu))} \alpha(\rho).$$

Example 1. Let $p = 3$ and $\mu = (1, 0, 2)$. Then we have

$$a^\mu = a_1 \left(a_1^{(2)} \right)^2, \quad \text{dev}(\mu) = (0, 2, 2),$$

and we obtain

$$\langle l_{(1,0,2)} | z_1^3 \rangle = \alpha_{(0,2,2)} + \alpha_{(2,0,2)} + \alpha_{(2,2,0)} = \binom{0}{0} \binom{3}{2} \binom{6}{2} + \binom{2}{2} \binom{3}{0} \binom{6}{2} + \binom{2}{2} \binom{5}{2} \binom{6}{0}.$$

3.4.2. Homogeneous case: two input systems without drift

In that case, we can write

$$\langle A_n | z_{i_1} z_{i_2} \cdots z_{i_p} \rangle = \sum_{\text{wgt}(\rho)=n} \alpha(\rho) a_{i_1}^{(\rho_1)} a_{i_2}^{(\rho_2)} \cdots a_{i_p}^{(\rho_p)}$$

and

$$a_{i_1}^{(\rho_1)} a_{i_2}^{(\rho_2)} \cdots a_{i_p}^{(\rho_p)} = a_1^{\mu_1} \left(a_1^{(1)} \right)^{\mu_2} \cdots \left(a_1^{(k-1)} \right)^{\mu_k} a_2^{v_1} \left(a_2^{(1)} \right)^{v_2} \cdots \left(a_2^{(k-1)} \right)^{v_h} = a^{\mu;v},$$

and we call *differential multiplicity over two variables* the tuple of finite integer sequences

$$\mu; v = (\mu_1, \mu_2, \dots, \mu_k; v_1, v_2, \dots, v_h).$$

The *degree* and the *weight* of $\mu; v$ are defined as follows:

$$p = \text{deg}(\rho) = \text{deg}(\mu; v) = \text{deg}(\mu) + \text{deg}(v), \quad \sum_i (1 + \rho_i) = \text{wgt}(\mu; v) = \text{wgt}(\mu) + \text{wgt}(v).$$

Denote by $|w|_1$ (respectively $|w|_2$) the length of w in the letter ζ_1 (respectively ζ_2). If ρ (respectively θ) is a derivation multi-index of degree $|w|_1$ (respectively $|w|_2$), we can define *the product or 'shuffle' of ρ and θ along w* as it is shown in the following example:

$$\text{for } w = z_2 z_1 z_2 z_1 z_1, \text{ we set } \rho \times_w \theta = (\rho_1, \rho_2, \theta_1, \rho_3, \theta_2).$$

Then the coefficient of a word w in $l_{\mu;v}$ is equal to 0, except in case $\text{deg}(\mu) = |w|_1$ and $\text{deg}(v) = |w|_2$, and we have in this case

$$\langle l_{\mu;v} | w \rangle = \sum_{\substack{\rho_1 \in \mathcal{S}(\text{dev}(\mu)) \\ \rho_2 \in \mathcal{S}(\text{dev}(v))}} \alpha_{\rho_1 \times_w \rho_2}.$$

Example 2. Let $w = z_2 z_1 z_2 z_1 z_1$ and $\mu; v = (1, 0, 2; 1, 1)$. In other words,

$$a^{\mu;v} = a_1 \left(a_1^{(2)} \right)^2 a_2 \left(a_2^{(1)} \right).$$

Then we have

$$\text{dev}(\mu) = (0, 2, 2) \quad \text{and} \quad \text{dev}(\nu) = (0, 1),$$

$$\begin{aligned} \langle l_{(1,0,2)} | z_2 z_1 z_2 z_1 z_1 \rangle &= \alpha_{(0,0,1,2,2)} + \alpha_{(1,0,0,2,2)} + \alpha_{(0,2,1,0,2)} + \alpha_{(1,2,0,0,2)} + \alpha_{(0,2,1,2,0)} + \alpha_{(1,2,0,2,0)} \\ &= \binom{0}{0} \binom{1}{0} \binom{3}{1} \binom{6}{2} \binom{9}{2} + \binom{1}{1} \binom{2}{0} \binom{3}{0} \binom{6}{2} \binom{9}{2} + \binom{0}{0} \binom{3}{2} \binom{5}{1} \binom{6}{0} \binom{9}{2} \\ &\quad + \binom{1}{1} \binom{4}{2} \binom{5}{0} \binom{6}{0} \binom{9}{2} + \binom{0}{0} \binom{3}{2} \binom{5}{1} \binom{8}{2} \binom{9}{0} + \binom{1}{1} \binom{4}{2} \binom{5}{0} \binom{8}{2} \binom{9}{0}. \end{aligned}$$

3.4.3. Nonhomogeneous case: single input system with drift

This case concerns system with a single input a_1 , but with a “drift”, that is an autonomous part: then we must take $a_0(t) \equiv 1$, and consequently $a_0^{(i)}(t) \equiv 0$ for any $i \geq 1$. Then we obtain the degenerated formula:

$$a_{i_1}^{(\rho_1)} a_{i_2}^{(\rho_2)} \dots a_{i_p}^{(\rho_p)} = a_0^{\mu_1} a_1^{\nu_1} (a_1^{(1)})^{\nu_2} \dots (a_1^{(k-1)})^{\nu_k} = a^{\mu;\nu} = a_1^\nu.$$

This formula is a degeneration of the previous one, subject to the relations

$$a^{\mu;\nu} = 0 \quad \text{except for } \mu = (j), \quad j \in \mathbb{N} \quad a^{(j);\nu} = a_1^\nu.$$

We have clearly

$$\text{dev}((j)) = \underbrace{(0, \dots, 0)}_{j \text{ times}}.$$

Then the coefficient of a word w in $l_{(j);\nu}$ is equal to 0 except in case $|w|_{z_0} = j$ and $|w|_{z_1} = \text{deg}(\mu)$. We have in this case

$$\langle l_\nu | w \rangle = \sum_{\substack{\rho_1 \in \mathcal{S}(\text{dev}(j)) \\ \rho_2 \in \mathcal{S}(\text{dev}(\nu))}} \alpha_{\rho_1 \times \rho_2}.$$

Example 3. Let $\nu = (0, 2, 1)$, and $w = z_1 z_1 z_0 z_1 z_0$ (hence $\mu = (2)$). In other words,

$$j = 2 \quad \text{and} \quad a^\nu = (a_1^{(1)})^2 a_1^{(2)}.$$

Then we have

$$\begin{aligned} \text{dev}(\mu) &= (0, 0) \quad \text{and} \quad \text{dev}(\nu) = (1, 1, 2), \\ \langle l_{(0,2,1)} | z_0 z_1 z_0 z_1 z_1 \rangle &= \alpha_{(0,1,0,1,2)} + \alpha_{(0,1,0,2,1)} + \alpha_{(0,2,0,1,1)} \\ &= \binom{0}{0} \binom{2}{1} \binom{3}{0} \binom{5}{1} \binom{8}{2} + \binom{0}{0} \binom{2}{1} \binom{3}{0} \binom{6}{2} \binom{8}{1} + \binom{0}{0} \binom{3}{2} \binom{4}{0} \binom{6}{1} \binom{8}{1} \\ &= \binom{2}{1} \binom{5}{1} \binom{8}{2} + \binom{2}{1} \binom{6}{2} \binom{8}{1} + \binom{3}{2} \binom{6}{1} \binom{8}{1}. \end{aligned}$$

4. A Faà di Bruno’s formula for causal functionals

In this section, we explain the relation between analytical functions of several variables and causal functionals.

4.1. Faà di Bruno's grammar

The analytical functions coincide with the causal functionals, the generating series of which are exchangeable (i.e. two commutatively equivalent words have the same coefficients). Indeed, the behaviour of exchangeable series G can be computed as

$$y_G(t) = \sum \langle G | z_0^{\alpha_0} \dots z_m^{\alpha_m} \rangle \frac{\xi_0^{\alpha_0}(t) \xi_1^{\alpha_1}(t) \dots \xi_m^{\alpha_m}(t)}{k_0! k_1! \dots k_m!}.$$

Recall that the n th derivative of a composite function $F(t) = \varphi(\xi(t))$ may be computed by using Faà di Bruno's formula:

$$F_n = \sum_{k=0}^n \varphi_k \sum_{k_1, \dots, k_n} \frac{n!}{k_1! \dots k_n! 1!^{k_1} \dots n!^{k_n}} \xi_1^{k_1} \dots \xi_n^{k_n},$$

with

$$F_n = D_t^n F(t), \quad \varphi_k = D_u^k [\varphi(u)]_{u=\xi(t)}, \quad \xi_k = D_t^k \xi(t).$$

We present now Chen's combinatorial interpretation [2] of this formula, for the sake of simplicity, for a function of two variables $F(t) = \Phi(\xi_0(t), \xi_1(t))$. The generalization to functions of more than two variables is obvious.

Let us define the following symbols:

$$\begin{aligned} \varphi_\varepsilon &= \Phi, \\ \gamma_0 &= \xi_0, \quad \gamma_k = D_t^k \xi_0(t), \quad \varphi_{w0} = \frac{\partial \varphi_w}{\partial \xi_0}, \\ \delta_0 &= \xi_1, \quad \delta_k = D_t^k \xi_1(t), \quad \varphi_{w1} = \frac{\partial \varphi_w}{\partial \xi_1}. \end{aligned}$$

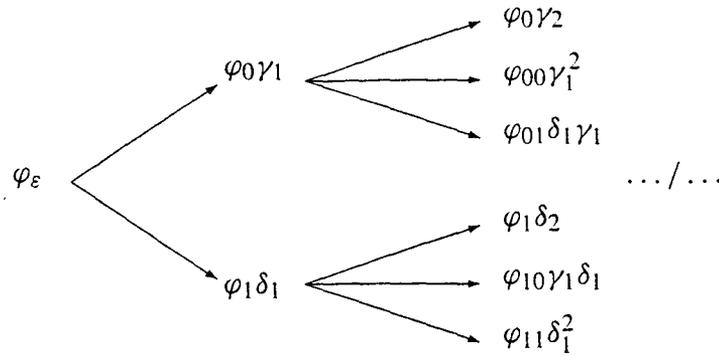
Now, let $f(t)$ be a time function. We obtain the following computation of time derivatives:

$$\frac{d}{dt}(f\Phi) = \frac{df}{dt}\Phi + f \frac{\partial \Phi}{\partial \xi_0} \frac{d\xi_0}{dt} + f \frac{\partial \Phi}{\partial \xi_1} \frac{d\xi_1}{dt}.$$

Chen's grammar is the interpretation over the encoding symbol of this derivation formula. It is defined by the axiom φ_ε , and the following rules:

$$\gamma_i \longrightarrow \gamma_{i+1}, \quad \delta_i \longrightarrow \delta_{i+1}, \quad \varphi_w \longrightarrow \varphi_{w0}\gamma_1, \quad \varphi_w \longrightarrow \varphi_{w1}\delta_1.$$

By using these rules, we obtain a derivation tree for computing the iterated derivatives of $F(t)$:



Example 4.

$$\frac{d^2 F}{dt^2} = \varphi_0 \gamma_2 + \varphi_1 \delta_2 + \varphi_{00} \gamma_1^2 + \varphi_{11} \delta_1^2 + (\varphi_{01} + \varphi_{10}) \gamma_1 \delta_1,$$

$$\frac{d^2 F}{dt^2} = \frac{\partial \Phi}{\partial \xi_0} \xi_0^{(2)}(t) + \frac{\partial \Phi}{\partial \xi_1} \xi_1^{(2)}(t) + \frac{\partial^2 \Phi}{\partial \xi_0^2} \xi_0^{(1)2}(t) + \frac{\partial^2 \Phi}{\partial \xi_1^2} \xi_1^{(1)2}(t) + 2 \frac{\partial^2 \Phi}{\partial \xi_0 \partial \xi_1} \xi_0^{(1)} \xi_1^{(1)}(t).$$

This grammar will be known as *Faà di Bruno's grammar*.

In these computations, we get $\varphi_{10} = \varphi_{01}$, according to the commutativity of the partial derivatives

$$\varphi_{01} = \frac{\partial^2 \varphi}{\partial \xi_0 \partial \xi_1} = \frac{\partial^2 \varphi}{\partial \xi_1 \partial \xi_0} = \varphi_{10}.$$

This property fails in case of causal functionals, essentially on account of the noncommutativity of iterated integrals:

$$\int_0^t \xi_i d\xi_j \neq \int_0^t \xi_j d\xi_i$$

(where ξ_j is the primitive, vanishing at $t = 0$, of the input a_j) or, equivalently, because left remain computations are not commutative. In spite of this remark, the same *Faà di Bruno's grammar* will enable us to compute also the n th derivative of causal functionals, as we shall see now.

4.2. Causal derivative along an input

Let H be some power series. We extend the formula

$$\frac{d}{dt} \langle H \| C_a \rangle = \langle A \triangleleft H \| C_a \rangle$$

to the case of a product of H by some time function $f(t)$. We obtain

$$\begin{aligned} \frac{d}{dt} \langle fH \| C_a \rangle &= \frac{d}{dt} [f \langle H \| C_a \rangle] = \frac{df}{dt} \langle H \| C_a \rangle + f \frac{d}{dt} \langle H \| C_a \rangle \\ &= \langle \frac{df}{dt} H \| C_a \rangle + f \langle A \triangleleft H \| C_a \rangle = \langle \frac{df}{dt} H + f(A \triangleleft H) \| C_a \rangle. \end{aligned}$$

Definition 1. (Causal derivative). We call causal derivative along the input a the operator Δ_a defined over the products of generating series by time functions as follows:

$$\Delta_a(fH) = \frac{df}{dt}H + f(A \triangleleft H) = \frac{df}{dt}H + f \left[\sum_{z_i \in Z} (z_i \triangleleft H) \frac{d\xi_i}{dt} \right].$$

Lemma 2. Let G be a series encoding a causal functional of the ξ_i , and let y_G its output. Then we have clearly

$$\frac{d^n}{dt^n} y_G = \langle \Delta_a^n G \| C_a \rangle.$$

Now let us describe Chen’s grammar computing the n th causal derivative $\Delta_a^n G$ of G along the input a . For the sake of simplicity, we present it in case of a causal functional of two inputs, $y_G = \Psi(\xi_0, \xi_1)$, defined by a formal series over two letters $G \in R\langle\langle z_0, z_1 \rangle\rangle$. (The generalization to more than two input variables is obvious.)

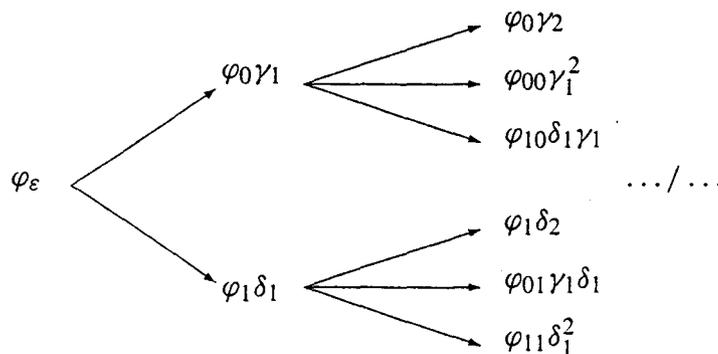
Thus we define the following symbols:

$$\varphi_w = w \triangleleft G \quad (\forall w \in Z^*), \quad \gamma_0 = \xi_0, \quad \gamma_k = D_t^k \xi_0(t), \quad \delta_0 = \xi_1, \quad \delta_k = D_t^k \xi_1(t).$$

The associated Chen’s grammar is defined by the rules, where we write for convenience 0 and 1 in place of z_0 and z_1 :

$$\gamma_i \longrightarrow \gamma_{i+1}, \quad \delta_i \longrightarrow \delta_{i+1}, \quad \varphi_w \longrightarrow \varphi_{0w}\gamma_1, \quad \varphi_w \longrightarrow \varphi_{1w}\delta_1.$$

Using these rules, we obtain the following derivation tree, computing the iterated causal derivatives of G :



Example 5. At the order two, we obtain

$$\Delta_a^2 G = (z_0 \triangleleft G)\xi_0^{(2)} + (z_1 \triangleleft G)\xi_1^{(2)} + (z_0^2 \triangleleft G)\xi_0^{(1)2} + (z_1^2 \triangleleft G)\xi_1^{(1)2} + (z_0z_1 \triangleleft G + z_0z_1 \triangleleft G)\xi_1^{(1)}\xi_0^{(1)}.$$

If we are interested in the successive time derivatives of y_G at time t for an input a , it suffices to compute $\langle \Delta_a^n G \| C_a(t) \rangle$, recalling that

$$\langle w \triangleleft H \| C_a(t) \rangle = \langle H \| C_a(t)w \rangle.$$

Example 5. (continued)

$$\begin{aligned} \frac{d^2 y_G}{dt^2}(t) = & \langle G \| C_a(t) z_0 \rangle \xi_0^{(2)}(t) + \langle G \| C_a(t) z_1 \rangle \xi_1^{(2)}(t) + \langle G \| C_a(t) z_0 z_0 \rangle \xi_0^{(1)2}(t) \\ & + \langle G \| C_a(t) z_1 z_1 \rangle \xi_1^{(1)2}(t) + (\langle G \| C_a(t) z_0 z_1 \rangle + \langle G \| C_a(t) z_1 z_0 \rangle) \xi_0^{(1)}(t) \xi_1^{(1)}(t). \end{aligned}$$

As a particular case, recalling that $C_a(0) = 1$, we have

$$\langle G \| C_a(0) w \rangle = \langle G | 1.w \rangle = \langle G | w \rangle.$$

Thus the second iterated derivative of y_G at time 0 for the input a is

$$\begin{aligned} \frac{d^2 y_G}{dt^2}(0) = & \langle G | z_0 \rangle \xi_0^{(2)}(0) + \langle G | z_1 \rangle \xi_1^{(2)}(0) + \langle G | z_0 z_0 \rangle \xi_0^{(1)2}(0) \\ & + \langle G | z_1 z_1 \rangle \xi_1^{(1)2}(0) + (\langle G | z_0 z_1 \rangle + \langle G | z_1 z_0 \rangle) \xi_0^{(1)}(0) \xi_1^{(1)}(0). \end{aligned}$$

Finally, if we have set $a_0(t) \equiv 1$ (and then $\xi_0(t) = t$, and $a_0^{(1)} = \xi_0^{(2)} = 0$), we obtain

$$\begin{aligned} \frac{d^2 y_G}{dt^2}(0) = & \langle G | z_1 \rangle a_1(0)^{(1)} + \langle G | z_0 z_0 \rangle + \langle G | z_1 z_1 \rangle a_1^2(0) \\ & + (\langle G | z_0 z_1 \rangle + \langle G | z_1 z_0 \rangle) a_1(0). \end{aligned}$$

5. The problem of algebraic identification

This problem consists in computing, by effective and complete formulas, the coefficients of a generating series when we know the Taylor expansions at $t = 0$ of some inputs and the Taylor expansions at $t = 0$ of the corresponding outputs.

This difficult problem remains unsolved. We propose here a first step in this direction: we outline an algorithm for identifying the coefficients of input multiderivatives a^μ , by solving the system of equations

$$y^{(k)}(0) = \sum_{\mu} a^\mu \langle G | l_\mu \rangle,$$

in order to compute the unknown coefficients $\langle G | l_\mu \rangle$.

5.1. Single input without drift

In this case, we denote the single input by $a(t)$ and we determine it polynomial:

$$a(t) = \sum_{i=0}^k \frac{c_i}{i!} t^i.$$

Then $A(0) = a(0)z_1 = c_0 \cdot z_1$ and $A^{(i)}(0) = c_i \cdot z_1$, we obtain

$$y^{(n)}(0) = \sum_{\text{wgt}=n} c_0^{\mu_1} \cdots c_{k-1}^{\mu_k} \langle G | z_1^{\mu_1 + \cdots + \mu_k} \rangle \lambda_\mu.$$

We get a triangular infinite system of linear equations:

$$\begin{aligned}
 y(0) &= \langle G|\varepsilon \rangle \\
 y^{(1)}(0) &= c_0 \langle G|z_1 \rangle \\
 y^{(2)}(0) &= a_1^{(0)} \langle G|z_1 \rangle + c_0^2 \langle G|z_1^2 \rangle \\
 &\vdots \\
 y^{(n)}(0) &= \sum_{\substack{\text{wgt}(\mu) = n \\ \text{deg}(\mu) \leq n}} a^\mu(0) \lambda_\mu \langle G|z_1^\mu \rangle + c_0^n \langle G|z_1^n \rangle \\
 &\vdots
 \end{aligned}$$

In this case, the computation of the unknown coefficients $\langle G|z_1^k \rangle$ is obvious for a selected input that satisfies $c_0 \neq 0$ if the resulting output derivatives $y^{(k)}(0)$ are known.

5.2. Single input with drift

We still choose the single input equal to a polynomial

$$a_1(t) = \sum_{i=0}^k \frac{c_i}{i!} t^i.$$

In this case, the generating series is defined over $\mathcal{Z} = \{z_0, z_1\}$ and

$$A(0) = c_0 \cdot z_1 + z_0, \quad A^{(i)}(0) = c_i \cdot z_1.$$

We then get a linear equation system that must be satisfied for any choice of the coefficients c_i representing an input:

$$y^{(n)}(0) = \sum_{\text{wgt}(\mu) \leq n} c_0^{\mu_1} \dots c_{k-1}^{\mu_{k-1}} \sum_{\substack{w \\ |w|_{z_1} = \text{deg}(\mu) \\ |w|_{z_0} = n - \text{wgt}(\mu)}} \langle G|l_\mu \rangle \quad (\text{FD})_n.$$

Instead of taking the entire system $(\text{FD})_{n \in \mathbb{N}}$, we take only one equation $(\text{FD})_n$ and we prove that we can identify every coefficient of the multiderivatives appearing in this equation, by choosing suitable inputs.

Consider the equation $(\text{FD})_n$; there is a determinant T_n which is not zero, the columns of which are indexed by the multiderivatives of the input and the rows of which are indexed by several choices of the input and its derivatives at $t = 0$.

- (1) The matrix T_n has a *lower triangular block form*. We order the multiderivatives (i.e. the columns) according to the increasing order of the number q of different orders of derivation appearing in the multiderivatives. For the same order q , we put together the multiderivatives containing the common set $\{a^{(i_1)}, a^{(i_2)}, \dots, a^{(i_q)}\}$ of derivatives. Dually, for any set $\{a^{(i_1)}, a^{(i_2)}, \dots, a^{(i_q)}\}$ of derivatives, we select a corresponding set of inputs for which every c_i equals 0, except $\{c_{i_1}, c_{i_2}, \dots, c_{i_q}\}$. This choice gives to T_n a lower triangular form.
- (2) Every diagonal block has a *determinant* $\neq 0$. We prove that by a suitable choice of inputs, every diagonal block may appear in a product form of nonsingular Van der Monde determinants. For example, for the equation $(\text{FD})_5$, we obtain the triangular superior T_5 (see Appendix A). The same method applies to a several input system.

Appendix A

In the matrix T_5 , see Fig. 1, the columns are indexed by the multiderivatives of the input. The lines correspond to choices of values at time 0 of the input and of its derivatives. The matrix T_5 has the block form as shown in Fig. 1.

Further the blocks \mathcal{M}_1 and \mathcal{M}_2 are given in Figs. 2 and 3, respectively.

$$T_5 = \begin{array}{c|ccc} & \text{multiderivatives } \rho & & \\ & q=0 & q=1 & q=2 \\ \text{choice} & 1 & 0 & 0 \\ \text{of} & 0 & \mathcal{M}_1 & 0 \\ \text{input} & 0 & * & \mathcal{M}_2 \end{array}$$

Fig. 1.

$$\mathcal{M}_1 = \begin{array}{c|ccc|cc} a & \dots & a^5 & a^{(1)} & a^{up(1)^2} & a^{(2)} & a^{(3)} & a^{(4)} \\ \hline c_0 & \dots & c_0^5 & & & & & \\ \vdots & \ddots & \vdots & & 0 & & & 0 \\ f_0 & \dots & f_0^5 & & & & & \\ \hline & & & 0 & c_1 & c_1^2 & & 0 \\ & & & & d_1 & d_1^2 & & \\ \hline & & & 0 & & & c_2 & 0 \\ & & & & & & 0 & c_3 & 0 \\ & & & & & & 0 & 0 & c_4 \end{array}$$

Fig. 2.

$$\mathcal{M}_2 = \begin{array}{c|cccc|cc} aa^{(1)} & a^2 a^{(1)} & a^3 a^{(1)} & aa^{(1)^2} & aa^{(2)} & a^2 a^{(2)} & aa^{(3)} & a^{(1)} a^{(1)} \\ \hline c_0 c_1 & c_0^2 c_1 & c_0^3 c_1 & c_0 c_1^2 & & & & \\ d_0 d_1 & d_0^2 d_1 & d_0^3 d_1 & d_0 d_1^2 & & 0 & & 0 \\ e_0 e_1 & e_0^2 e_1 & e_0^3 e_1 & e_0 e_1^2 & & & & \\ f_0 f_1 & f_0^2 f_1 & f_0^3 f_1 & f_0 f_1^2 & & & & \\ \hline & & & 0 & c_0 c_2 & c_0^2 c_2 & & 0 \\ & & & & d_0 d_2 & d_0^2 d_2 & & \\ \hline & & & 0 & & & c_0 c_3 & 0 \\ & & & & & & 0 & c_1^2 \end{array}$$

Fig. 3.

Thus the matrix T_5 has a lower triangular block form. If we choose $c_1 = d_1 = e_1 = f_1 = c_2 = d_2 = 1$, and the other coefficients not equal to 0, then each diagonal block appears as a non-singular Van der Monde matrix, thus T_5 is non-singular.

In the general case, with a suitable choice of the scalar coefficients and some Gauss eliminations, each diagonal block appears as a non-degenerated scalar multiple of some non-singular Van der Monde matrix. Thus, with this choice, the matrix T_n is regular invertible, and allows to solve the system $(FD)_n$.

References

- [1] F. Boussemart, V. Hoang Ngoc Minh, Graphic simulation of nonlinear control systems behaviour, in: SCRATCHPAD, IMACS Symp. MCTS (Lille, 1991).
- [2] W.Y. Chen, Context-free grammars, differential operators and formal power series, *Séries formelles et combinatoire algébrique* (Bordeaux, 2–4 May 1991) 145–159.
- [3] K.T. Chen, Iterated integrals and exponential homomorphisms, *Proc. London Math. Soc.* 4 (1954) 502–512.
- [4] K.T. Chen, Integration of paths, a faithful representation of paths by noncommutative formal power series, *Trans. Amer. Math. Soc.* 89 (1958) 395–407.
- [5] K.T. Chen, Iterated path integrals, *Bull. Amer. Math. Soc.* 83 (1977) 831–879.
- [6] P. Crouch and F. Lamnabhi-Lagarigue, State realizations of nonlinear systems defined by input–output differential equations, *Lecture Notes in Control and Information Sciences*, Vol. 11 (Springer, New York, 1988) 138–149.
- [7] L.S. Dederick, Successive derivatives of a function of several functions, *Ann. of Math.* 27 (1925) 385–394.
- [8] M. Fliess, Un outil algébrique: les séries formelles non commutatives, in: G. Marchesini and S.K. Mitter, eds., *Mathematical System Theory, Lecture Notes in Economics and Mathematical Systems*, vol. 131 (Springer, Berlin, 1976) 122–148.
- [9] M. Fliess, Fonctionnelles causales non linéaires et indéterminées non commutatives, *Bull. Soc. Math. France* 109 (1981) 3–40.
- [10] M. Fliess, M. Lamnabhi and F. Lamnabhi-Lagarigue, An algebraic approach to nonlinear functional expansions, *IEEE Trans. Circuits and Systems CAS-30* (1983) 554–570.
- [11] C. Hespel and G. Jacob, Calcul des approximations locales bilinéaires de systèmes analytiques, *RAIRO APII* 23 (1989) 331–349.
- [12] C. Hespel and G. Jacob, Approximation of nonlinear dynamic systems by rational series, *Theoret. Comput. Sci.* 79 (1991) 151–162.
- [13] C. Hespel, Truncated bilinear approximants: Carleman, finite Volterra, Padé-type, geometrical and structural automata, *Proc. 1st European Conf. on Algebraic Computing in Control* (Paris, 13–15 March 1991).
- [14] V. Hoang Ngoc Minh, Evaluation transform with Kernel function, in: G. Duchamp, D. Krob and G. Jacob, eds. *Algebraic and Computing Treatment of Noncommutative Power Series, Theoretical Computer Science* 79 (1991).
- [15] V. Hoang Ngoc Minh, Input/output behaviour of nonlinear control systems: About exact and approximated computations, *Proc. IMACS Symp. MCTS (Lille, 1991)*.
- [16] G. Jacob, Algebraic methods and computer algebra for nonlinear systems' study, *Proc. IMACS Symp. MCTS* (1991).
- [17] F. Lamnabhi-Lagarigue and P.E. Crouch, A formula for iterated derivatives along trajectories of nonlinear systems, *Systems and Control letters* 11 (1988) 1–7.
- [18] P. Leroux, A. Martin and X.G. Viennot, Computing iterated derivatives along trajectories of nonlinear systems, Preprint, 1992.
- [19] N.E. Oussous and M. Petitot, Implantation de la réalisation minimale des systèmes dynamiques en Scratchpad, *Proc. ACC Conf.* (Paris, CNRS, 1991).
- [20] H.J. Sussmann, A product expansion for the Chen series, in: C. Byrnes and A. Lindquist, eds., *Theory and Applications of Nonlinear Control Systems* (North-Holland, Amsterdam, 1986) 323–335.

Annexe 7

Hespel C., Jacob G., *First steps towards exact algebraic identification*, Discrete Math., vol. 180, pp. 211-219, 1998.

Reprinted from

DISCRETE MATHEMATICS

Discrete Mathematics 180 (1998) 211–219

First steps towards exact algebraic identification

Christiane Hespel^{a,*}, Gérard Jacob^b

^a *I.N.S.A. Laboratoire L.A.N.S., 20, avenue des Buttes de Coësmes, 35043 Rennes Cédex, France*

^b *L.I.F.L. (U.A. 369 C.N.R.S.), Bat. M3, Université Lille I, 59655 Villeneuve d'Ascq Cédex, France*

Received 18 July 1995; received in revised form 29 January 1996; accepted 15 July 1996



ELSEVIER



ELSEVIER

Discrete Mathematics 180 (1998) 211–219

DISCRETE
MATHEMATICS

First steps towards exact algebraic identification

Christiane Hespel^{a,*}, Gérard Jacob^b

^a *I.N.S.A. Laboratoire L.A.N.S., 20, avenue des Buttes de Coësmes, 35043 Rennes Cédex, France*

^b *L.I.F.L. (U.A. 369 C.N.R.S.), Bat.M3, Université Lille I, 59655 Villeneuve d'Ascq Cédex, France*

Received 18 July 1995; received in revised form 29 January 1996; accepted 15 July 1996

Abstract

This paper presents the first step towards solving the problem of the exact algebraic identification of causal functionals. This problem consists in computing the coefficients of a noncommutative generating series when only the Taylor expansion of some inputs (at $t=0$), and the Taylor expansion (at $t=0$) of associated outputs are known.

Resumé

Cet article présente la première étape vers la résolution du problème de l'Identification Algébrique Exacte des fonctionnelles causales. Ce problème consiste à calculer les coefficients d'une série génératrice non commutative, connaissant les développements de Taylor (en $t=0$) des entrées, et les développements de Taylor (en $t=0$) des sorties correspondantes.

1. Introduction

The causal functionals as defined by Fliess [3] are obtained from noncommutative generating series by evaluating words as iterated integrals of the inputs. In this paper, we are interested in solving the 'algebraic identification problem'.

Is there an algorithm for computing the coefficients of a generating series when the Taylor expansions (at $t=0$) of the system inputs and of related outputs are known?

A positive answer would introduce the causal functionals as fully combinatorial objects. Some proofs are actually known of the injectivity of the 'evaluation map' [11,4,9]: if two generating series G and H define the same causal functional, then actually $G=H$. But they do not solve our problem. Here, first we recall and briefly discuss them. Then, we present a computation formula for the iterated derivatives of the output.

* Corresponding author. E-mail: hespel@nimbus.irisa.fr.

Finally, we present a first step of the algebraic identification: we provide an algorithm computing the contribution, in the generating series G , of each inputs multiderivative. It will remain (in a future work) to compute the coefficient in G of each inputs iterated integral.

2. Chen series

2.1. Preliminaries and notations

See [3,5,7,10]. We associate to the input $a = (a_0, \dots, a_m)$ (with $a_0(t) \equiv 1$) an encoding alphabet $\mathcal{Z} = \{z_0, z_1, \dots, z_m\}$. A word w in \mathcal{Z}^* is a finite sequence of letters in \mathcal{Z} . We denote by ε the empty word. A noncommutative formal series S on \mathcal{Z} is a mapping of \mathcal{Z}^* into \mathbb{R}

$$S = \sum_{w \in \mathcal{Z}^*} \langle S | w \rangle w.$$

The operations of sum and Cauchy product of two series S, T are defined by

$$S + T = \sum_{w \in \mathcal{Z}^*} (\langle S | w \rangle + \langle T | w \rangle) w; \quad S.T = \sum_{w \in \mathcal{Z}^*} \sum_{uv=w} \langle S | u \rangle . \langle T | v \rangle w.$$

We define as follows the iterated integral $\langle \mathcal{C}_a(t) | w \rangle = \int_0^t \delta_a(w)$ of a word w for input a :

$$\int_0^t \delta_a(\varepsilon) = 1,$$

$$\int_0^t \delta_a(vz_j) = \int_0^t \left(\int_0^\tau \delta_a(v) \right) a_j(\tau) d\tau, \quad \forall z_j \in \mathcal{Z}, \forall v \in \mathcal{Z}^*.$$

Hence, we define the Chen series [1] of the input a as being $\mathcal{C}_a = \sum_{w \in \mathcal{Z}^*} \langle \mathcal{C}_a | w \rangle w$. The time derivative of \mathcal{C}_a (see [10]) is given as $(d/dt)\mathcal{C}_a = \mathcal{C}_a . \mathcal{L}_a$ where we have set $\mathcal{L}_a = \sum_{0 \leq i \leq m} a_i . z_i$. Otherwise, any noncommuting power series H over \mathcal{Z} , the coefficients of which satisfy the following 'convergence condition': $\exists K, L \in \mathbb{R}^+ | \langle H | w \rangle | < K |w|! L^{|w|}$, $\forall w \in \mathcal{Z}^*$ defines a causal functional [3], absolutely convergent in a neighbourhood of 0, with the output

$$y_H(t) = \sum_{w \in \mathcal{Z}^*} \langle H | w \rangle \langle \mathcal{C}_a(t) | w \rangle = \langle H || \mathcal{C}_a(t) \rangle,$$

where $\|$ means infinite sum. From the derivation formula of Chen series, we deduce

$$\dot{y}_H(t) = \langle H || \mathcal{C}_a . \mathcal{L}_a \rangle.$$

The concatenation of an input a defined on $[0, t_1[$ and b defined on $[0, t_2[$ is the input $a \# b$ defined on $[0, t_1 + t_2[$ by

$$a \# b(t) = \begin{cases} a(t) & \text{if } 0 \leq t < t_1, \\ b(t - t_1) & \text{if } t_1 \leq t < t_1 + t_2. \end{cases}$$

Then we have the relation $\mathcal{G}_{az_c}(t_1 + t) = \mathcal{G}_a(t_1)\mathcal{G}_c(t)$. We have also

$$\begin{aligned} \text{for } 0 < t < t_1, \quad & \dot{y}(t) = \langle G \parallel \mathcal{G}_a(t), \mathcal{L}_a(t) \rangle, \\ \text{for } t_1 \leq t_1 + t < t_1 + t_2, \quad & \dot{y}(t) = \langle G \parallel \mathcal{G}_a(t_1), \mathcal{G}_b(t), \mathcal{L}_b(t) \rangle, \\ \text{and then} \quad & \lim_{t \rightarrow 0^+} [\dot{y}(t_1 + t)] = \langle G \parallel \mathcal{G}_a(t_1), \mathcal{L}_b(0^+) \rangle. \end{aligned}$$

2.2. Sketch of three proofs about injectivity

Proposition 2.1. *The generating series evaluation of the input/output functionals is faithful. In other words, any series indistinguishable from 0 is identically null.*

We cursorily recall three proofs of this proposition, in relation to the identification problem.

2.2.1. The proof of Sontag and Wang [11]

This proof is based on noncommutative parameters partial derivatives. We set

$$\begin{aligned} b &= \mu_1 \# \mu_2 \# \dots \# \mu_k, \\ \text{the input } \mu_i &= (\mu_{i,1}, \mu_{i,2}, \dots, \mu_{i,m}) \text{ is defined on } [0, t_i[. \end{aligned}$$

The Chen series of the input b is $\mathcal{G}_b = \mathcal{G}_{\mu_1} \mathcal{G}_{\mu_2} \dots \mathcal{G}_{\mu_k}$. We consider the output $y(t)$ as a function of the t_j . Then, its iterated partial noncommuting derivatives must be computed as follows:

$$\begin{aligned} \frac{\hat{c}}{\hat{c}t_k} (\langle G \parallel \mathcal{G}_{\mu_1} \dots \mathcal{G}_{\mu_k} \rangle)_{t_i=0^+} &= \langle G \parallel \mathcal{G}_{\mu_1} \dots \mathcal{G}_{\mu_{k-1}} \mathcal{L}_{\mu_k}(0^+) \rangle, \\ \frac{\hat{c}}{\hat{c}t_{k-1}} \frac{\hat{c}}{\hat{c}t_k} (\langle G \parallel \mathcal{G}_{\mu_1} \dots \mathcal{G}_{\mu_k} \rangle)_{t_i=0^+, t_{i-1}=0^+} &= \langle G \parallel \mathcal{G}_{\mu_1} \dots \mathcal{G}_{\mu_{k-2}} \mathcal{L}_{\mu_{k-1}}(0^+) \mathcal{L}_{\mu_k}(0^+) \rangle. \end{aligned}$$

By iterating and assuming G indistinguishable from 0, we obtain

$$\langle G \parallel \mathcal{L}_{\mu_1}(0^+) \dots \mathcal{L}_{\mu_k}(0^+) \rangle = \sum_{l_1, \dots, l_k} \langle G \parallel z_{l_1} \dots z_{l_k} \rangle_{\mu_{1,l_1} \mu_{2,l_2} \dots \mu_{k,l_k}} = 0$$

for any values of μ_{i,l_i} . Then $G = 0$, since its coefficients can be recovered as follows:

$$\langle G \parallel z_{l_1} \dots z_{l_k} \rangle = \frac{\hat{c}^k}{\hat{c}_{\mu_{1,l_1}} \dots \hat{c}_{\mu_{k,l_k}}} \left(\frac{\hat{c}}{\hat{c}t_1} \dots \frac{\hat{c}}{\hat{c}t_k} (y) \right)_{t_i=0^+} = 0.$$

2.2.2. The proof of Fliess [4]

For $\varepsilon > 0$, ε small, there are inputs a, b and a_i ($i \neq 0$), the Chen series of which are

$$\mathcal{G}_a = e^{\varepsilon z_0}; \quad \mathcal{G}_b = e^{2\varepsilon z_0}; \quad \mathcal{G}_{a_i} = e^{\varepsilon z_0 + \varepsilon^2 z_i}.$$

Then we verify,

$$z_0 = \lim_{\varepsilon \rightarrow 0} \frac{1}{\varepsilon} [\mathcal{G}_b - \mathcal{G}_a]; \quad z_i = \lim_{\varepsilon \rightarrow 0} \frac{1}{\varepsilon^2} [\mathcal{G}_{a_i} - \mathcal{G}_a].$$

Now, G being indistinguishable from zero, we deduce for any Chen series \mathcal{C}

$$\langle z_0 \triangleleft G \parallel \mathcal{C} \rangle = \langle G \parallel \mathcal{C} z_0 \rangle = \lim_{\varepsilon \rightarrow 0} \frac{1}{\varepsilon} [\langle G \parallel \mathcal{C} \cdot \mathcal{C}_b \rangle - \langle G \parallel \mathcal{C} \mathcal{C}_a \rangle] = 0,$$

$$\forall i \in [1, m], \quad \langle z_i \triangleleft G \parallel \mathcal{C} \rangle = \langle G \parallel \mathcal{C} z_i \rangle = \lim_{\varepsilon \rightarrow 0} \frac{1}{\varepsilon^2} [\langle G \parallel \mathcal{C} \cdot \mathcal{C}_{a_i} \rangle - \langle G \parallel \mathcal{C} \mathcal{C}_a \rangle] = 0.$$

Then, the power series $z_0 \triangleleft G$ and $z_i \triangleleft G$ are indistinguishable from zero. Recursively, for any word w the series $w \triangleleft \mathcal{C}$ also is indistinguishable from zero. It follows that the series G is equal to zero, because we have $\forall w \in Z^* \langle G \mid w \rangle = \langle w \triangleleft G \mid \varepsilon \rangle = 0$.

2.2.3. The proof of Reutenauer [9].

Shortly, one computes first a concatenated input of the form

$$\mathcal{C} = \mathcal{C}_{a_1}(t_1) \cdots \mathcal{C}_{a_p}(t_p) = e^{t_1 z_0 + z_1 z_1} \cdots e^{t_p z_0 + z_p z_p}.$$

By noting that $y(t) = \langle G \parallel \mathcal{C} \rangle \equiv 0$, by expanding \mathcal{C} as power series in the parameters t_i and z_i , and by using properties of analytical functions of real variables, one finds that

$$\langle G \mid z_0^{j_0} z_1 \cdots z_0^{j_{p-1}} z_{i_{p-1}} z_0^{j_p} \rangle = 0.$$

2.2.4. Remarks on these proofs

All these proofs require the knowledge of $y(t)$ and of its iterated derivatives everywhere in some neighbourhood of 0. None of them provides an exact identification method. This remark justifies our request of an algebraic identification.

3. Iterated derivatives of the output

For any series G , the time derivatives of the output are given by $y_G^{(i)}(t) = \langle G \parallel \mathcal{C}_a^{(i)}(t) \rangle$. Thus, we compute now the iterated time derivatives of the Chen's series (see [6]).

3.1. Derivatives of Chen series

Denoting by D_t the time derivation operator, we obtain recursively

$$(R) \quad \mathcal{C}_a^{(i)} = \mathcal{C}_a A_i, \quad \text{where} \quad \left\{ \begin{array}{l} A_1 = \mathcal{L}_a \\ A_{i+1} = \mathcal{L}_a A_i + D_t A_i \end{array} \right. = \sum a_j z_j \}.$$

We set $A_0 = 1$. Let us note \mathcal{A} the formal sum of the A_i . We have the *combinatorial identity*

$$\sum_{i \in \mathbb{N}} A_i = \mathcal{A} = 1 + \mathcal{L}_a \mathcal{A} + D_t \mathcal{A} = \sum_{\rho} \alpha_{\rho} \mathcal{L}_a^{[\rho]},$$

where the 'derivative' $\mathcal{L}_a^{[\rho]}$ of \mathcal{L}_a by a multiindex $\rho = (\rho_1, \dots, \rho_p)$ is given as follows:

$$\mathcal{L}_a^{(\rho_k)} = \sum_i a_i^{(\rho_k)} z_i, \quad \mathcal{L}_a^{[\rho]} = \mathcal{L}_a^{(\rho_1)} \cdots \mathcal{L}_a^{(\rho_p)} = \sum_{i_1, \dots, i_p} a_{i_1}^{(\rho_1)} \cdots a_{i_p}^{(\rho_p)} z_{i_1} \cdots z_{i_p}.$$

The degree of $\rho = (\rho_1, \dots, \rho_p)$ is p , the weight of ρ is the integer $\sum_{j=1}^p (1 + \rho_j) = \rho_1 + \dots + \rho_k + k$. Then, we have $A_i = \sum_{\text{wgt}(\rho)=i} \mathcal{L}_a^{[\rho]}$. An easy analysis of the combinatorial identity gives as result

$$(E) \quad \alpha_\rho = \prod_{i=1}^p \binom{\sum_{j=1}^i \rho_j + i - 1}{\rho_i} = \binom{\rho_1}{\rho_1} \binom{\rho_1 + \rho_2 + 1}{\rho_2} \dots \binom{\rho_1 + \dots + \rho_k + k - 1}{\rho_k}.$$

3.2. Output derivatives and dynamical systems

With this definition of the coefficients $\alpha(\rho)$, we deduce

$$y^{(n)}(t) = \langle G \parallel \mathcal{C}_a A_n \rangle = \sum_{\text{wgt}(\rho)=n} \alpha(\rho) \sum_{i_j} a_{i_1}^{(\rho_1)}(t) \dots a_{i_k}^{(\rho_k)}(t) \langle z_{i_1} \dots z_{i_k} \triangleleft G \parallel \mathcal{C}_a(t) \rangle,$$

$$y^{(n)}(0) = \langle G \parallel A_n \rangle = \sum_{\text{wgt}(\rho)=n} \alpha(\rho) \sum_{i_j} a_{i_1}^{(\rho_1)}(0) \dots a_{i_k}^{(\rho_k)}(0) \langle G \parallel z_{i_1} \dots z_{i_k} \rangle.$$

In case of a dynamical system given by the state equations on an analytic manifold

$$(\Sigma) \quad \begin{cases} \dot{q} = g_0(q) + \sum_{i=1}^m a_i(t) g_i(q), \\ y(t) = h(q), \end{cases}$$

denoting by $g_i \circ h$ the Lie derivative of h with respect to the vector field g_i , the output of (Σ) is also the output $y_G(t)$ of the causal functional defined by its generating series G [3]

$$G = \sum_{k \geq 0} \sum_{i_j=0}^m g_{i_1} \dots g_{i_k} \circ h|_{q_0} z_{i_1} \dots z_{i_k}.$$

The formula giving the iterated derivative $y^{(n)}(t)$ may be interpreted by two means.

- A local expression in the state $q(t)$ (Lamnabhi-Lagarrigue and Crouch [8])

$$y^{(n)}(t) = \sum \sum \alpha(\rho) a_{i_1}^{(\rho_1)}(t) \dots a_{i_k}^{(\rho_k)}(t) (g_{i_1} \dots g_{i_k} \circ h)|_{q(t)}.$$

- A global expression in the state $q(0)$

$$y^{(n)}(t) = \sum \sum \alpha(\rho) a_{i_1}(0)^{(\rho_1)} \dots a_{i_k}(0)^{(\rho_k)} y_{i_1 i_2 \dots i_k}(t),$$

where $y_{i_1 i_2 \dots i_k}$ is the output of (Σ) initialized in $q(0)$, for the observation $g_{i_1} g_{i_2} \dots g_{i_k} \circ h$.

(Other works on input/output differential equations can be found in [2,12,13].)

4. First step for computing the algebraic identification

In the expansion formula of $y^{(n)}(0)$, each product $a_{i_1}^{(\rho_1)}(0) \dots a_{i_k}^{(\rho_k)}(0)$ can be viewed as the value in 0 of a monomial a^u in the inputs a_i and their derivatives. So we

get an expansion

$$y^{(n)}(0) = \sum_{\text{wgt}(\mu)=n} a^\mu \langle G || l_\mu \rangle.$$

Our aim is here to *effectively compute for each monomial a^μ , the contribution: $\langle G || l_\mu \rangle$.*

4.1. Single input without drift

In this case, the identification problem has a completely straightforward solution as follows. We set $a_0(t) = c_0 \neq 0$ (constant). Then $A_1 = c_0 z_0$, $D_t A_i = 0$, $A_i = c_0^i z_0^i$, and finally,

$$y^{(n)}(0) = c_0^n \langle G | z_0^n \rangle.$$

Such a direct proof cannot be used for more than one input, or when there is a drift part.

4.2. Single input with drift

This case is concerned with dynamic systems having an autonomous part described by a ‘non-controlled vector field’ $g_0 \neq 0$. Equivalently, we set $a_0(t) \equiv 1$. We take $a_1(t) = a(t) = \sum_{i=0}^k \frac{c_i}{i!} t^i$. (In other words $a^{(j)}(0) = c_j, \forall j \leq k$). The differential monomials take the form

$$a^v = (a^{(i_1)})^{e_1} (a^{(i_2)})^{e_2} \dots (a^{(i_q)})^{e_q}.$$

The *weight* of v related to a_1 is the sum $\sum_{j=1}^q e_j(1 + i_j)$. And then we get an expansion

$$y_a^{(n)} = \sum_{\text{sum}_j \{e_j(1+i_j)\} \leq n} (a^{(i_1)})^{e_1} (a^{(i_2)})^{e_2} \dots (a^{(i_q)})^{e_q} G(n)_{i_1, i_2, \dots, i_q}^{e_1, e_2, \dots, e_q} \tag{FD}_n$$

We have to identify the coefficients $G(n)_{i_1, i_2, \dots, i_q}^{e_1, e_2, \dots, e_q}$. Such an equation $(\text{FD})_n$ is provided for any input — that is for any free choice of the input coefficients c_i — and for the resulting output $y_a^{(n)}$.

1. *Identification of the $G(n)_0^0$*

For the trivial input $a(t) \equiv 0$, we obtain directly $y^{(n)}(0) = \langle G | z_0^n \rangle = G(n)_0^0$.

2. *Identification of the $G(n)_{i_1}^{e_1}$*

Define inputs by $c_j = \delta_{i_1, j} b_k$ for some constant b_k . The resulting output is

$$\sum_{e_1(1+i_1) \leq n} b_k^{e_1} G(n)_{i_1}^{e_1}.$$

By taking an adequate number of pairwise distinct k_j , the square-linear system has a Vandermonde determinant. Thus, we can identify the $G(n)_{i_1}^{e_1}$.

3. Recurrence step

We search to identify the $G(n)_{i_1, i_2, \dots, i_q}^{e_1, e_2, \dots, e_q}$ for a fixed set $I = \{i_1, i_2, \dots, i_q\}$ of lower indices. Suppose the already identified monomials for lower index set are *strictly* included in I . We use *specialized inputs* satisfying $c_j = 0$ except in case $j \in I$. Thus, in $(FD)_n$ vanishes any monomial with some lower index j not in I . By eliminating the ‘smaller’ index sets (induction step), the linear system takes the form

$$L_n(a) = \sum_{\text{sum}\{e_j(1+i_j)\} \leq n} c_{i_1}^{e_1} \cdots c_{i_q}^{e_q} G(n)_{i_1, i_2, \dots, i_q}^{e_1, e_2, \dots, e_q},$$

with $L_n(a)$ computed from related output and smaller index set eliminations. We associate to the ‘column index’ $G(n)_{i_1, i_2, \dots, i_q}^{e_1, e_2, \dots, e_q}$ the specialized input $A_{i_1, i_2, \dots, i_q}^{e_1, e_2, \dots, e_q}$ defined by

$$c_{i_1} = b_{i_1, e_1}, \quad c_{i_2} = b_{i_2, e_2}, \dots, \quad c_{i_q} = b_{i_q, e_q}.$$

The square matrix of the obtained linear system has the following properties:

- (a) The row of index $A_{i_1, i_2, \dots, i_q}^{e_1, e_2, \dots, e_q}$ admits as common divisor the product

$$b_{i_1, e_1} b_{i_2, e_2} \cdots b_{i_q, e_q}.$$

- (b) After factorizing these common divisors, straightforward equalities of rows occur when two distinct pairs $c_{i, j}$ and $c_{i', j'}$ are taken equal. Hence, the remaining determinant can be divided by a product of multiples of the differences $b_{i, j} = b_{i', j'}$, for $i \leq i'$, and $j \leq j'$. This last product and the determinant have the same polynomial degree. By checking the diagonal term of the determinant, we deduce that these two polynomials are equal.

By taking pairwise different values for the $c_{i, j}$, that achieves the proof.

Example. In the case $n = 15$, let us study the $G(15)_{1, 2, 3}^{i, j, k}$. We obtain in the restricted expansion the following 9 terms, given below with their weights:

$$\text{wgt}(G_{1, 2, 3}^{1, 1, 1}) = 9, \quad \text{wgt}(G_{1, 2, 3}^{2, 1, 1}) = 11, \quad \text{wgt}(G_{1, 2, 3}^{3, 1, 1}) = 13,$$

$$\text{wgt}(G_{1, 2, 3}^{4, 1, 1}) = 15, \quad \text{wgt}(G_{1, 2, 3}^{1, 2, 1}) = 12, \quad \text{wgt}(G_{1, 2, 3}^{2, 2, 1}) = 14,$$

$$\text{wgt}(G_{1, 2, 3}^{1, 3, 1}) = 15, \quad \text{wgt}(G_{1, 2, 3}^{1, 1, 2}) = 13, \quad \text{wgt}(G_{1, 2, 3}^{2, 1, 1}) = 15.$$

After the first-factorization process (a), the matrix of the *simplified linear system* takes the form

$G_{1,2,3}^{1,1,1}$	$G_{1,2,3}^{2,1,1}$	$G_{1,2,3}^{3,1,1}$	$G_{1,2,3}^{4,1,1}$	$G_{1,2,3}^{1,2,1}$	$G_{1,2,3}^{2,2,1}$	$G_{1,2,3}^{1,3,1}$	$G_{1,2,3}^{1,1,2}$	$G_{1,2,3}^{2,1,2}$
1	$b_{1,1}$	$b_{1,1}^2$	$b_{1,1}^3$	$b_{2,1}$	$b_{1,1}b_{2,1}$	$b_{2,1}^2$	$b_{3,1}$	$b_{1,1}b_{3,1}$
1	$b_{1,2}$	$b_{1,2}^2$	$b_{1,2}^3$	$b_{2,1}$	$b_{1,2}b_{2,1}$	$b_{2,1}^2$	$b_{3,1}$	$b_{1,2}b_{3,1}$
1	$b_{1,3}$	$b_{1,3}^2$	$b_{1,3}^3$	$b_{2,1}$	$b_{1,3}b_{2,1}$	$b_{2,1}^2$	$b_{3,1}$	$b_{1,3}b_{3,1}$
1	$b_{1,4}$	$b_{1,4}^2$	$b_{1,4}^3$	$b_{2,1}$	$b_{1,4}b_{2,1}$	$b_{2,1}^2$	$b_{3,1}$	$b_{1,4}b_{3,1}$
1	$b_{1,1}$	$b_{1,1}^2$	$b_{1,1}^3$	$b_{2,2}$	$b_{1,1}b_{2,2}$	$b_{2,2}^2$	$b_{3,1}$	$b_{1,1}b_{3,1}$
1	$b_{1,2}$	$b_{1,2}^2$	$b_{1,2}^3$	$b_{2,2}$	$b_{1,2}b_{2,2}$	$b_{2,2}^2$	$b_{3,1}$	$b_{1,2}b_{3,1}$
1	$b_{1,1}$	$b_{1,1}^2$	$b_{1,1}^3$	$b_{2,3}$	$b_{1,1}b_{2,3}$	$b_{2,3}^2$	$b_{3,1}$	$b_{1,1}b_{3,1}$
1	$b_{1,1}$	$b_{1,1}^2$	$b_{1,1}^3$	$b_{2,1}$	$b_{1,1}b_{2,1}$	$b_{2,1}^2$	$b_{3,2}$	$b_{1,1}b_{3,2}$
1	$b_{1,2}$	$b_{1,2}^2$	$b_{1,2}^3$	$b_{2,1}$	$b_{1,2}b_{2,1}$	$b_{2,1}^2$	$b_{3,2}$	$b_{1,2}b_{3,2}$

Its determinant is a polynomial of degree 14. By row comparisons (step (b)), we factorize

$$(b_{1,1} - b_{1,2})^3(b_{1,1} - b_{1,3})(b_{1,2} - b_{1,3})(b_{1,1} - b_{1,4})(b_{1,2} - b_{1,4})(b_{1,3} - b_{1,4}) \\ (b_{2,1} - b_{2,2})^2(b_{2,1} - b_{2,3})(b_{2,2} - b_{2,3})(b_{3,1} - b_{3,2})^2$$

that is also a polynomial of degree 14. We check directly that the quotient is equal to 1.

4.3. Several inputs with drift

We consider the case of 2 inputs $a_1(t)$ and $a_2(t)$. We choose them as polynomials

$$a_1(t) = \sum_{i=0}^{l_1} \frac{c_i}{i!} t^i, \quad a_2(t) = \sum_{i=0}^{l_2} \frac{d_i}{i!} t^i, \quad (\text{and } a_0(t) \equiv 1).$$

We obtain the following expansion:

$$y^{(n)} = \sum c_{i_1}^{e_1} \cdots c_{i_p}^{e_p} d_{j_1}^{f_1} \cdots d_{j_q}^{f_q} G(n)_{i_1, \dots, i_p, j_1, \dots, j_q}^{e_1, \dots, e_p, f_1, \dots, f_q}, \quad (\text{FD})_n$$

where the sum is taken for $\text{sum}\{e_k(1 + i_k)\} + \text{sum}\{f_k(1 + j_k)\} \leq n$

The identification of the multiderivative coefficients appearing in the previous equation is solved by the same method as for a single-input system

1. Select some set $\{i_1, i_2, \dots, i_p; j_1, \dots, j_q\}$ of derivative orders, and suppose the identification already done for strict subsets of these derivatives. We use some *specialized inputs* satisfying $c_j = 0$ (resp. $d_k = 0$) except in case $j \in \{i_1, \dots, i_p\}$ and $k \in \{j_1, \dots, j_q\}$. We eliminate the 'shorter monomials' (by preceding identification steps).

To the ‘column index’ $G(n)_{i_1, \dots, i_p; j_1, \dots, j_q}^{e_1, \dots, e_q; f_1, \dots, f_q}$ we associate the specialized input defined by

$$c_{i_1} = b_{i_1, e_1}, \dots, \quad c_{i_q} = b_{i_q, e_q}, \quad \text{and} \quad d_{j_1} = v_{j_1, f_1}, \dots, \quad d_{j_q} = v_{j_q, f_q}.$$

- We can factorize in each row the corresponding product $b_{i_1, e_1} \dots b_{i_p, e_p} v_{j_1, f_1} \dots v_{j_q, f_q}$. And thereafter, by purchasing in the simplified determinant the row equalities appearing each time that one equalizes two distinct indexed coefficients, we can completely compute the determinant, which vanishes only if one of the pairwise distinct coefficients b_{i_k, e_k} or v_{j_k, f_k} is also vanishing.

5. Conclusions

A first step has been taken towards algebraic identification. It remains, by using together the combinatorial definition of the multiderivatives $a^{(\rho)}$ and of the differential monomials a^μ , to separate the contribution $\langle G|w \rangle$ of each noncommutative word in G .

References

- [1] K.T. Chen, Iterated path integrals, *Bull. Amer. Math. Soc.* 83 (1977) 831–879.
- [2] P. Crouch, F. Lamnabhi-Lagarrigue, State realizations of nonlinear systems defined by input–output differential equations, *Lecture Notes in Control and Inform. Sci.* 11 (1988) 138–149.
- [3] M. Fliess, Fonctionnelles causales non linéaires et indéterminées non commutatives, *Bull. Soc. Math. France* 109 (1981) 3–40.
- [4] M. Fliess, On the concept of derivatives and Taylor expansions for nonlinear input/output systems, *Proc. 22nd IEEE Conf. Decision and Control*, San Antonio, Texas, 1983, pp. 643–646.
- [5] M. Fliess, M. Lamnabhi, F. Lamnabhi-Lagarrigue, An algebraic approach to nonlinear functional expansions, *IEEE Trans. Circuits Systems CAS-30* (8) (1983) 554–570.
- [6] C. Hespel, Iterated derivatives of a non linear dynamic system and Faà di Bruno formula, *Math. Comput. Simulation*, to appear.
- [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, pp. 599–608.
- [8] F. Lamnabhi-Lagarrigue, P.E. Crouch, A formula for iterated derivatives along trajectories of nonlinear systems, *Systems Control Lett.* 11 (1988) 1–7.
- [9] C. Reutenauer, private communication.
- [10] H.J. Sussmann, A product expansion for the Chen series, in: C. Byrnes, A. Lindquist (Eds.), *Theory and Applications of Nonlinear Control Systems*, Elsevier, Amsterdam, 1986, pp. 323–335.
- [11] Y. Wang, E.D. Sontag, On two definitions of observation spaces, *Systems Control Lett.* 13 (1989) 279–289.
- [12] Y. Wang, E.D. Sontag, Generating series and non linear systems: analytic aspects, local realizability and i/o representations, *Forum Math.* 4 (1992) 299–322.
- [13] Y. Wang, E.D. Sontag, Algebraic differential equations and rational control systems, *SIAM. J. Control Optim.* 30 (5) (1992) 1126–1149.

Annexe 8

Hespel C., Jacob G., *On Algebraic Identification of Causal Functionals*, 10th International Conference on Formal Power Series and Algebraic Combinatorics, Toronto, June 14-19, 1998.



THE FIELDS INSTITUTE FOR
RESEARCH IN MATHEMATICAL SCIENCES

10th International Conference on Formal Power Series and Algebraic Combinatorics

June 14-19, 1998

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 $G = \sum_{w \in Z^*} \langle G | w \rangle w$, where w is a word on a finite encoding alphabet. The problem consists in computing the coefficients $\langle G | w \rangle$ 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 g_μ 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 g_μ . 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 $G = \sum_{w \in Z^*} \langle G | w \rangle w$, où w parcourt l'ensemble des mots sur un alphabet fini de codage. Il s'agit de calculer les coefficients $\langle G | w \rangle$ 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 g_μ 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 g_μ . 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 $\langle G | w \rangle$, 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}_μ indexed by the partitions μ .

Here we give a careful analysis of the polynomial \mathbf{g}_μ , 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:

$$(\Sigma) \quad \begin{cases} \dot{q} &= f_0(q) + \sum_{j=1..m} f_j(q) a_j(t) \\ y(t) &= h(q(t)) \end{cases}$$

where $\mathbf{a}(t) = (a_j(t))_{j=1..m}$ is the m -dimensional input, $q(t) \in \mathbb{R}^n$ 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 $\mathcal{Z} = \{z_1, z_2, \dots, z_m\}$, and the free monoid \mathcal{Z}^* of words over \mathcal{Z} . A *noncommutative power series* S on \mathcal{Z} is any map from \mathcal{Z}^* to \mathbb{R} , usually written as a formal sum $S = \sum_{w \in \mathcal{Z}^*} \langle S | w \rangle w$. Then, starting at the state $q(0) = q_0$, the output of (Σ) can be obtained as an infinite sum

$$y(t) = \sum_{w \in \mathcal{Z}^*} \langle G_\Sigma | w \rangle \langle \mathcal{C}_\mathbf{a}(t) | w \rangle \quad \text{where}$$

- $G_\Sigma = \sum_w \langle G_\Sigma | w \rangle w$ 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:

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

- $C_a(t) = \sum_w \langle C_a(t) | w \rangle w$ is the *Chen series* of the input $\mathbf{a}(t)$ and is defined as the solution of the differential equation:

$$\frac{d}{dt} C_a(t) = C_a(t) \sum_{z_j \in Z} a_j(t) z_j \quad \text{with } C_a(0) = 1$$

Thus the Chen series $C_a(t)$ 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 $\mathcal{L}_a(t) = \sum_{z_j \in Z} a_j(t) z_j$.

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:

$$y(t) = \sum_{w \in Z^*} \langle G | w \rangle \langle C_a(t) | w \rangle$$

Now we can point out the following two problems:

- The **Identification Problem** consists in computing the coefficients $\langle G | w \rangle$ 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}_μ 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}_μ .

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

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

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:

$$\frac{d^n}{dt^n} y(t) = \langle G | \frac{d^n}{dt^n} \mathcal{C}_a(t) \rangle = \sum_{w \in \mathcal{Z}^*} \langle G | w \rangle \langle \frac{d^n}{dt^n} \mathcal{C}_a(t) | w \rangle$$

We obtain, by a straightforward computation:

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

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

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

For instance:

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

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*:

$$\mathcal{A} = \varepsilon + \mathcal{L}_a \mathcal{A} + \frac{d}{dt} \mathcal{A} \quad \text{where} \quad \mathcal{L}_a = \sum_{z_j \in \mathcal{Z}} a_j z_j$$

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 $\rho = (\rho_1, \rho_2, \dots, \rho_n)$, 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:

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

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

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

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

$$\mu = (i_1^{e_1} i_2^{e_2} \dots i_q^{e_q}), \quad 1 \leq i_1 < i_2 < \dots < i_q$$

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

$$\text{wgt}(\mu) = \sum_k e_k \cdot i_k \quad \text{size}(\mu) = \sum_k e_k$$

The empty partition is noted ε .

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

- For a differential letter: $\frac{d}{dt}(a^{(i_k)}) = a^{(i_{k+1})}$, 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):

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

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:

$$D(1^1 2^3 3^2 6^1) = 2^4 3^2 6^1 + 3 \star 1^1 2^2 3^3 6^1 + 2 \star 1^1 2^3 3^1 4^1 6^1 + 1^1 2^3 3^2 7^1$$

We obtain so a linear equation, with adequate definition of the coefficients $\langle \mu | \tau \rangle$:

$$D\mu = \sum_{\tau \in \text{partitions}} \langle \mu | \tau \rangle \tau \quad \text{with } \langle \mu | \tau \rangle \in \mathbb{N}$$

Remark If $\langle \mu | \tau \rangle \neq 0$, then $\text{size}(\mu) = \text{size}(\tau)$. In other words, the operator D preserve the size.

4.2 Colored partitions and multiplicities

Let $\mathcal{Z} = \{z_1, z_2, \dots, z_m\}$ be a finite *alphabet of colors*. We call *colored multi-index* any finite sequence of “colored strictly positive integers” $\rho = (\rho_1 z_{i_1}, \rho_2 z_{i_2}, \dots, \rho_m z_{i_m})$ (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 $w = w(\rho) = z_{i_1} z_{i_2} \dots z_{i_m}$. The sequence of integers occurring 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:

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

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*

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

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 $\frac{d}{dt}$ of differential monomials on m differential letters, is given by multilinearly extending the law D defined on each color component:

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

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

$$D\mu = \sum_{\tau \in \text{colored partitions}} \langle \mu | \tau \rangle * \tau \quad \text{with } \langle \mu | \tau \rangle \in \mathbb{N}$$

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 $\mathcal{A} = \sum_{\mu} \mu \cdot \mathfrak{g}_{\mu}$. Then the generic equation becomes:

$$\sum_{\mu} \mu \cdot \mathfrak{g}_{\mu} = \varepsilon + \sum_{j=1}^m \sum_{\sigma} a_j \sigma \cdot z_j \mathfrak{g}_{\sigma} + \sum_{\nu} D(\nu) \cdot \mathfrak{g}_{\nu}$$

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

$$\mathfrak{g}_{\mu} = \sum_{j=1}^m z_j \mathfrak{g}_{\mu \triangleright a_j} + \sum_{\nu} \langle \nu | \mu \rangle * \mathfrak{g}_{\nu} \quad \text{with } \langle \nu | \tau \rangle \in \mathbb{N}$$

where we the symbol \triangleright is used as follows:

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

The coefficients $\langle \nu | \mu \rangle$ 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*:

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

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

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

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

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

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

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

Then the recursive equations become:

$$\mathfrak{g}_{1^p \otimes \nu} = z_0 \mathfrak{g}_{1^{p-1} \otimes \nu} + z_1 \mathfrak{g}_{1^p \otimes \nu \triangleright 1} + \sum_{\sigma} \langle \sigma | \nu \rangle * \mathfrak{g}_{1^p \otimes \sigma}$$

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:

$$\mathfrak{g}_{1^p \otimes \nu} - \sum_{\sigma} \langle \sigma | \nu \rangle * \mathfrak{g}_{1^p \otimes \sigma} = z_0 \mathfrak{g}_{1^{p-1} \otimes \nu} + z_1 \mathfrak{g}_{1^p \otimes \nu \triangleright 1}$$

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:

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

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

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

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:

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

where $\text{ext}(\mathbf{G}_{/p}^{m-1}(k), 0)$ 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 \parallel). We obtain so:

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

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:

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

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:

$$\begin{aligned} \mathbf{J}_m^p[1,p] \mathbf{G}_{/p-1}^m[1,p] &= \left[\begin{array}{c} \left(\mathbf{G}_{/p-1}^{m-1}[1,p] \right) // \mathbf{G}_{/p-2}^m[1,p] \\ 0 \end{array} \right] \\ \mathbf{J}_m^p[2,p] \mathbf{G}_{/p-2}^m[2,p] &= \left[\begin{array}{c} \left(\mathbf{G}_{/p-2}^{m-1}[2,p] \right) // \mathbf{G}_{/p-3}^m[2,p] \\ 0 \end{array} \right] \quad \dots \end{aligned}$$

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 criterium(4, 0, 3) (the / and // are only written to explain the structure, and the * for lacking elements):

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

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

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}_μ .

We conjecture that it will never return *zero* to any call of the procedure $\text{analysis}(m, k)$. 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 $analysis(m, k)$ 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, (19)
primes, (3) (17) (19)
primes, (2)² (3) (17)(19)
primes, (2)⁴ (3) (13) (17)² (19)
primes, (2)³ (11) (13)² (17)² (19)
primes, (2)² (11) (13)³ (17)² (19)
primes, (2)² (11) (13)³ (17)² (19)
primes, (2)² (11) (13)² (17)² (19)
primes, (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, (2)¹⁵ (3)⁸ (5)³ (7)² (11)⁶ (13)¹³ (17)¹⁴ (19)⁹ (23) (29)
primes, (2)²⁰ (3)⁸ (5)⁸ (7)⁴ (11)⁸ (13)¹⁴ (17)¹³ (19)⁸ (23)³ (29)
primes, (2)¹⁷ (3)¹⁰ (5)⁷ (7)⁶ (11)¹⁰ (13)¹⁴ (17)¹² (19)⁸ (23)⁶ (29)
primes, (2)¹⁹ (3)¹⁴ (5)⁶ (7)³ (11)¹² (13)¹⁴ (17)¹⁰ (19)⁸ (23)⁷ (29)
primes, (2)¹² (3)⁸ (5)⁷ (7) (11)¹³ (13)¹³ (17)⁹ (19)⁸ (23)⁸ (29)
primes, (2)⁸ (3)⁶ (5)⁷ (11)¹¹ (13)¹⁴ (17)⁶ (19)⁶ (23)⁷ (29)
primes, (2)³ (3)⁵ (5)⁷ (11)⁸ (13)¹⁰ (17)⁴ (19)⁴ (23)⁶ (29)
primes, (2)³ (3)² (5)⁴ (7) (11)⁵ (13)⁶ (17)³ (19)² (23)³ (29)
primes, (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



Abstract

The first aim of this report is to define some mathematical and software tools for dealing with formal series in noncommutative variables, in Algebraic Computing: Two presentations of a formal series are provided; the first one is given by its Hankel matrix, and the other one by its remainder automaton. To each presentation, we associate a Padé-type approximant of the series.

The second aim is to use the formal series as a tool for approximating or identifying the dynamic systems:

We provide 2 techniques for checking some bilinear approximants of dynamic systems, up to a given order k . These techniques are based on a rational approximation of the generating series of the given system. The first one uses its presentation by its Hankel matrix, and gives a minimal rank approximant. The second one is obtained by truncating some automata, which are associated with the generating series of the system. We provide 2 remarkable truncated automata: the structural automaton (method of the differential algebra), and the geometric automaton (Carleman method).

Then, we give an algorithm for identifying the generating series of a dynamic system up to a given order k , when knowing the Taylor expansions for $t = 0$, of the inputs and of the corresponding outputs. Three steps are necessary:

- 1) To get a linear equations system, where the output derivatives are some linear combinations of the input multiderivatives.
- 2) To identify the inputs multiderivatives coefficients by measuring the input/output values, and the input/output derivatives values.
- 3) To extract from the previous coefficients, the coefficients of the words (the degree of which are $\leq k$), of the generating series, by using a generic splitting algorithm.

Finally, we conjecture that the identification problem, as defined previously, is solved at any order, and we propose some combinatoric proof ideas.

Key words

Formal series in noncommutative variables, dynamic systems, bilinear systems, rational approximations, bilinear approximations, identification of causal functional, identifiability, algebraic computation.

Résumé

Le premier objet de ce mémoire est de définir des outils mathématiques ou informatiques utiles au traitement des séries formelles en variables non commutatives par le Calcul Formel: Deux présentations d'une série formelle sont proposées, l'une par sa matrice de Hankel, et l'autre par l'automate de ses résiduelles. A chacune de ces présentations, nous associons un approximant de type Padé de la série.

Le second objet est d'utiliser les séries formelles comme outil d'approximation ou d'identification des systèmes dynamiques:

Nous proposons 2 techniques de construction d'approximants bilinéaires de systèmes dynamiques à un ordre k donné, s'appuyant sur une approximation rationnelle de la série génératrice du système fourni. La première technique utilise sa présentation par la matrice de Hankel. L'approximant obtenu est de rang minimal. La seconde technique s'appuie sur des troncatures d'automates associés à la série génératrice du système. Nous présentons 2 automates tronqués remarquables: l'automate structural (méthode de l'algèbre différentielle) et l'automate géométrique (méthode de Carleman).

Puis nous construisons un algorithme d'identification jusqu'à un degré k donné, de la série génératrice d'un système dynamique, connaissant les développements de Taylor en $t = 0$, des entrées et des sorties associées. Trois étapes sont nécessaires:

- 1) Obtention d'un système d'équations linéaires, où les dérivées de la sortie apparaissent comme des combinaisons linéaires des multidérivées des entrées.
- 2) Identification des coefficients des multidérivées des entrées par des mesures des valeurs d'entrée/sortie et de leurs dérivées.
- 3) Extraction, à partir des coefficients précédents, des coefficients des mots de degré $\leq k$, de la série génératrice, par un algorithme générique de scindage.

Enfin, nous posons la conjecture que l'identification, au sens où nous l'avons définie, est toujours possible, et indiquons des idées de preuve de nature combinatoire.

Mots clés

Séries formelles en variables non commutatives, systèmes dynamiques, systèmes bilinéaires, approximations rationnelles, approximations bilinéaires, identification de fonctionnelle causale, identifiabilité, calcul formel.