Spring 2004

Generating Series for Interconnected Nonlinear Systems and the Formal Laplace-Borel Transform

Yaqin Li

Old Dominion University

GENERATING SERIES FOR INTERCONNECTED NONLINEAR SYSTEMS AND THE FORMAL LAPLACE-BOREL TRANSFORM

by

Yaqin Li
B.S. July 1998, University of Science and Technology of China
M.E. July 2001, University of Science and Technology of China

A Thesis Submitted to the Faculty of
Old Dominion University in Partial Fulfillment of the
Requirements for the Degree of

DOCTOR OF PHILOSOPHY

ELECTRICAL ENGINEERING

OLD DOMINION UNIVERSITY
May 2004

Approved by:

W. Steven Gray (Director)

Oscar R. González (Member)

Matthias Kawski (Member)

Brett Newman (Member)

John J. Swetits (Member)

ABSTRACT

GENERATING SERIES FOR INTERCONNECTED NONLINEAR SYSTEMS
AND THE FORMAL LAPLACE-BOREL TRANSFORM

Yaqin Li
Old Dominion University, 2004
Director: Dr. W. Steven Gray

Formal power series methods provide effective tools for nonlinear system analysis. For a broad range of analytic nonlinear systems, their input-output mapping can be described by a Fliess operator associated with a formal power series. In this dissertation, the interconnection of two Fliess operators is characterized by the generating series of the composite system. In addition, the formal Laplace-Borel transform of a Fliess operator is defined and its fundamental properties are presented. The formal Laplace-Borel transform produces an elegant description of system interconnections in a purely algebraic context.

Specifically, four basic interconnections of Fliess operators are addressed: the parallel, product, cascade and feedback connections. For each interconnection, the generating series of the overall system is given, and a growth condition is developed, which guarantees the convergence property of the output of the corresponding Fliess operator.

Motivated by the relationship between operations on formal power series and system interconnections, and following the idea of the classical integral Laplace-Borel transform, a new formal Laplace-Borel transform of a Fliess operator is proposed. The properties of this Laplace-Borel transform are provided, and in particular, a fundamental semigroup isomorphism is identified between the set of all locally convergent power series and the set of all well-defined Fliess operators.

A software package was produced in Maple based on the ACE package developed by

the ACE group in Université de Marne-la-Vallée led by Sébastien Veigneau. The ACE package provided the binary operations of addition, concatenation and shuffle product on the free monoid of formal polynomials. In this dissertation, the operations of composition, modified composition, chronological products and the evaluation of Fliess operators are implemented in software. The package was used to demonstrate various aspects of the new interconnection theory.

ACKNOWLEDGMENTS

I would like to express my sincere gratitude to my advisor, Dr. W. Steven Gray, for his constant support and valuable guidance. His warm encouragement and help built the path for me to pursue my Ph.D. study. His persistence and commitment in research provided a role model for me in my future career.

I am also grateful to Dr. Oscar R. González for his guidance throughout the years of my doctoral studies. His working habits, precise attention to detail, as well as his quick grasp of the big picture, have taught me how to be a good researcher as well as a good teacher.

I wish to extend many thanks to my committee members: Dr. Matthias Kawski (Department of Mathematics and Statistics, Arizona State University), Dr. Brett Newman (Department of Aerospace Engineering, Old Dominion University) and Dr. John J. Swetits (Department of Mathematics and Statistics, Old Dominion University) for their great patience and valuable guidance concerning my research.

I also would like to thank the faculty and staff of the Department of Electrical and Computer Engineering at Old Dominion University for giving me the opportunity to continue my graduate study.

Finally, I want to thank all my colleagues in the Systems Research Lab for all the good times we had together.

Yaqin Li

Norfolk, Virginia, U.S.A.

May, 2004

TABLE OF CONTENTS

LIST OF TABLES

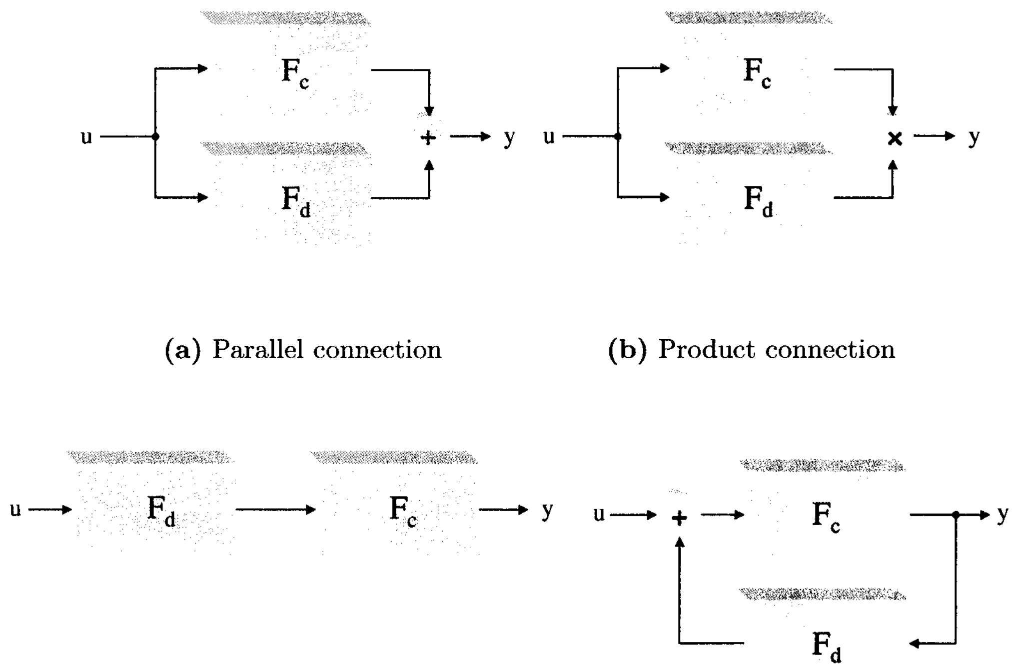
LIST OF FIGURES

CHAPTER I

INTRODUCTION

The connection between algebraic combinatorics and nonlinear control theory has been increasing steadily since the 1960's. One of the uniting branches is the Chen-Fliess series and the associated formal power series methods used in nonlinear system analysis. The formal power series approach was advocated by Fliess in [18–23] and motivated by the iterated path integrals proposed by Chen in [7–15]. The iterated path integral approach possesses rich algebraic structures, thus providing a natural algebraic representation of functional expansions for the outputs of dynamic systems. There are two classical ways to describe systems: the input-output representation and the state space representation. For a broad range of analytic systems, the input-output mapping can be described by a so called *Fliess operator*, which is written in terms of iterated integrals and an associated formal power series. In this dissertation, the main class of systems considered is all analytic systems that can be represented by Fliess operators. To ensure that a Fliess operator represents a well-defined system, its associated formal power series must be locally convergent in the sense that the output of the Fliess operator converges over a finite interval.

In control system applications, systems are interconnected in a variety of ways. Understanding the nature of these interconnections is important for both system analysis and control design. For a large-scale system, it is convenient to first decompose it into subsystems, and then to analyze the whole system by considering the subsystem interconnections. For linear systems, some beautiful and complete results for the interconnections are now standard subjects [35], however, the interconnections of nonlinear systems are not so well understood, see for example, [1, 63, 65, 69, 70]. Our specific interest in this dissertation is



(a) Parallel connection (b) Product connection

(c) Cascade connection (d) Feedback connection

Figure 1.1: Elementary system interconnections.

the interconnection of analytic input-output systems represented as Fliess operators. The four basic interconnections considered are the parallel, product, cascade and feedback connections, as shown in Figure 1.1. A fairly complete algebraic theory can be constructed to describe these interconnections.

The classical Laplace-Borel transform plays an essential role in the analysis and design of linear time-invariant systems. For a linear time-invariant system, the output response can be described by the convolution integral involving the impulse response and the input signal. Applying the classical Laplace-Borel transform maps this convolution integral expressed in the time domain to a purely algebraic expression in the frequency domain. In this way, a linear time-invariant system can be completely characterized by the Laplace transform of its impulse response, normally called the *transfer function* of the system. For proper linear systems, the transfer function can be expanded as a Laurent series. Using this idea of series representation, the classical integral Laplace-Borel transform will be generalized to the class of analytic nonlinear systems that can be described by Fliess operators. By introducing this formal Laplace transform, the iterated integral in the time domain is mapped into an algebraic expression in terms of formal power series, therefore providing an elegant approach to nonlinear system representation. More specifically, in linear system analysis, the Laplace transform of an analytic signal can be represented by its Laurent series in the frequency domain, which is a formal power series. A linear time-invariant system can also be characterized by the Laurent series expansion of its transfer function, which can be viewed as a series representation of the system. Thus, the Laplace-Borel transform of a linear system is really the mapping of a linear convolution operator to a formal power series representation.

Motivated by the series representation and the Laplace-Borel transform for signals and systems in the linear case, the basic idea is to generalize these concepts to the nonlinear setting. The formal Laplace-Borel transform of a signal was generalized to the nonlinear case by Fliess in [20, 23], but a formal transform of a system is not so straightforward. In linear system analysis, a system can be described by its impulse response, which is also a signal. Therefore, the same notion of the Laplace-Borel transform of a signal can be directly applied to produce the transform of a linear system. However, in the nonlinear case, this approach is not possible. Therefore, an appropriate definition for the formal transform of the nonlinear input-output dynamics is needed. Another objective of this dissertation is to apply the formal Laplace-Borel transform to interconnected systems to provide a compact series representation for the composite systems. The well-known isomorphism between time domain and frequency domain in the linear case is also generalized to the nonlinear setting.

1.1 Motivation

A linear system example is considered first to illustrate the basic concepts more clearly. Let the space of inputs $\boldsymbol{u}$ be the set of measurable real-valued functions $u(t) \in \mathbb{R}^m$, and the space of outputs $\boldsymbol{y}$ be a set of measurable functions where $y(t) \in \mathbb{R}^\ell$. A causal linear input-output system is then the mapping $F : \boldsymbol{u} \to \boldsymbol{y}$ defined by the convolution integral involving its impulse response $H(t,\tau)$ and the system input $u(t)$:

$$y(t) = F[u](t) = \int_{t_0}^{t} H(t,\tau)u(\tau)d\tau, \quad t \geq t_0. \tag{1.1}$$

If each component function of H is real analytic, then on some set $D = \{(t,\tau) \in \mathbb{R}^2 : t_0 \leq \tau \leq t \leq t_0 + T\}$, each column of H, H_i can be represented by its Taylor series centered at (τ, t_0),

$$H_i(t,\tau) = \sum_{n_1,n_2=0}^{\infty} c(n_2,i,n_1)\frac{(t-\tau)^{n_2}}{n_2!}\frac{(\tau-t_0)^{n_1}}{n_1!}, \tag{1.2}$$

where each coefficient $c(n_2,i,n_1) \in \mathbb{R}^\ell$. Substituting equation (1.2) into equation (1.1) and using the uniform convergence of the series on D, it follows that

$$y(t) = \sum_{n_1,n_2=0}^{\infty}\sum_{i=1}^{m} c\,(n_2,i,n_1)\int_{t_0}^{t}\frac{(t-\tau)^{n_2}}{n_2!}u_i(\tau)\frac{(\tau-t_0)^{n_1}}{n_1!}d\tau. \tag{1.3}$$

To see the mathematical structure underlying this series representation more clearly, define formally $u_0(t) \equiv 1$ and let

$$E_i[u](t,t_0) = \int_{t_0}^{t} u_i(\tau)d\tau, \quad i = 0,1,\cdots,m.$$

Observe that

$$E_0[u](t,t_0) = t - t_0,$$

and define recursively

$$E_{00}[u](t,t_0) = \int_{t_0}^{t} E_0[u](\tau,t_0)d\tau = \frac{(t-t_0)^2}{2!}.$$

After n_1 iterated integrations,

$$E_{\underbrace{0\cdots 0}_{n_1}}[u](t,t_0) = \frac{(t-t_0)^{n_1}}{n_1!}.$$

Select any $i = 1, 2, \cdots, m$ and define

$$E_{i\underbrace{0\cdots 0}_{n_1}}[u](t,t_0) = \int_{t_0}^{t} u_i(\tau) E_{\underbrace{0\cdots 0}_{n_1}}[u](\tau,t_0)d\tau = \int_{t_0}^{t} u_i(\tau)\frac{(\tau-t_0)^{n_1}}{n_1!}\,d\tau. \tag{1.4}$$

Now integrate expression (1.4) once more and apply integration by parts,

$$\begin{aligned} E_{0i\underbrace{0\cdots 0}_{n_1}}[u](t,t_0) &= \int_{t_0}^{t}\int_{t_0}^{\tau} u_i(\xi)\frac{(\xi-t_0)^{n_1}}{n_1!}\,d\xi d\tau \\ &= \int_{t_0}^{t}(t-\tau)u_i(\tau)\frac{(\tau-t_0)^{n_1}}{n_1!}\,d\tau. \end{aligned}$$

After applying the previous step n_2 times inductively, the iterated integral will be

$$E_{\underbrace{0\cdots 0}_{n_2}i\underbrace{0\cdots 0}_{n_1}}[u](t,t_0) = \int_{t_0}^{t}\frac{(t-\tau)^{n_2}}{n_2\,!}u_i(\tau)\frac{(\tau-t_0)^{n_1}}{n_1!}\,d\tau.$$

Therefore, series (1.3) can be expressed alternatively as

$$y(t) = \sum_{n_1,n_2=0}^{\infty}\sum_{i=1}^{m} c(n_2,i,n_1)E_{\underbrace{0\cdots 0}_{n_2}i\underbrace{0\cdots 0}_{n_1}}[u](t,t_0). \tag{1.5}$$

This example suggests an alternative way to index the summations appearing in (1.5). Define the set of index symbols $I = \{0, 1, \ldots, m\}$ as an alphabet, and any finite sequence over the alphabet I is a word. Let I^k be the set of words $i_k i_{k-1}\cdots i_1$ with length k over alphabet I, where $i_r \in I$ for $1 \le r \le k$. For $k = 0$, I^0 denotes the set whose only element is the empty word $\emptyset$ and $I^* = \bigcup_{k\ge 0} I^k$ denotes the set of all words over I. Let η be an arbitrary word in I^* and define the mapping

$$(c,\eta) = \begin{cases} c(n_2,i,n_1) & : \eta = \underbrace{0\cdots 0}_{n_2}i\underbrace{0\cdots 0}_{n_1} \\ 0 & : \text{otherwise.} \end{cases} \tag{1.6}$$

Then the series (1.5) has a more compact expression

$$y(t) = \sum_{\eta \in I^*} (c,\eta) E_\eta[u](t,t_0). \tag{1.7}$$

In fact, for any mapping of the form $c : I^* \mapsto \mathbb{R}^\ell$, an input-output operator F_c associated with c can be defined using (1.7). Specifically, for an arbitrary formal power series of the form

$$c = \sum_{\eta \in I^*} (c,\eta)\eta,$$

where $\eta = i_k i_{k-1} \cdots i_1 \in I^*$, and $(c,\eta) \in \mathbb{R}^\ell$, the unique input-output operator $F_c : \boldsymbol{u} \to \boldsymbol{y}$ is defined by

$$F_c[u] = \sum_{\eta \in I^*} (c,\eta)\, E_\eta[u], \tag{1.8}$$

which is referred to as a *Fliess operator* [20, 23]. The formal power series is called the *generating series* of the Fliess operator. The set of all formal power series over the alphabet I is denoted by $\mathbb{R}^\ell \langle\langle I \rangle\rangle$. Fliess operators can be regarded as series in a finitely generated free algebra called the Fliess algebra. A Fliess operator can be completely characterized by its generating series. In order for the formal summation in the Fliess operator definition to represent a well-defined system, the coefficients of the series are generally assumed to satisfy the following growth condition [20, 21, 23, 33, 34, 60]

$$|(c,\eta)| \leq K M^{|\eta|} |\eta|!, \;\; k \geq 0, \tag{1.9}$$

for some finite real numbers $K, M > 0$. Here $|\eta|$ denotes the number of symbols in η. It was proven in [28] that if a formal power series has coefficients satisfying the growth condition in (1.9), the output of the Fliess operator converges absolutely and uniformly on a finite time interval when the inputs are restricted to an open ball in L_p space. If a series satisfies (1.9), it is said to be *locally convergent*. The set of all locally convergent formal power series is denoted by $\mathbb{R}^\ell_{LC} \langle\langle I \rangle\rangle$.

In many applications, input-output systems are interconnected in different ways. Given two well-defined Fliess operators F_c and F_d, Figure 1.1 shows four elementary interconnections. One of the general goals of this dissertation is to describe in a unified manner the generating series for each elementary interconnection, and conditions under which they are locally convergent. Some partial results about the local convergence property of interconnected systems have been obtained using a state space approach. If c and d both have finite Lie rank (see [60]), in addition to being locally convergent, then the mappings F_c and F_d each have a finite dimensional analytic state space realization, and therefore so does each interconnected system. The classical literature then provides that the corresponding generating series can be computed by successive Lie derivatives [21] and must be locally convergent [60]. But whether this rank condition is necessary to ensure the local convergence of the interconnected system does not appear in the present literature. For those locally convergent systems that do not have a finite Lie rank, and therefore are not realizable by a finite dimensional, analytic, affine in the control state space system, does there exist a generating series for each possible interconnection? If so, how does one obtain the generating series? Are the new generating series for the composite systems also locally convergent? These problems are fundamental to those who wish to use this model class in applications. The parallel connection is the trivial case, and the product connection was analyzed in [23,66]. But the analysis of these connections can be applied to the study of the cascade and feedback interconnections. In [17], Ferfera showed that for a single-input-single-output (SISO) system (i.e., $\ell = m = 1$), there always exists a generating series $c \circ d$ such that $y = F_c[F_d[u]] = F_{c \circ d}[u]$, but a multi-variable version of this composition product is not available in the literature, nor are any results about local convergence. Generating series for the feedback interconnection is an entirely new problem. For the feedback equation

$$y = F_c[u + F_d[y]], \tag{1.10}$$

whether there exists a generating series $c@d$ such that $y = F_{c@d}[u]$ and y satisfies (1.10) is not known. When F_c is linear, the formal solution to the feedback equation (1.10) can be written as

$$y = F_c[u] + F_c \circ F_d \circ F_c[u] + \cdots$$

It is not immediately clear whether this series converges in any manner, and in particular, converges to another Fliess operator. When F_c is nonlinear, the problem is further complicated by the fact that operators of the form $I + F_d$, where I denotes the identity map, *never* have a Fliess operator representation. One of the main goals in this dissertation is to describe the existence and local convergence of the formal power series for the multi-variable cascade and feedback connections.

The classical Laplace-Borel transform provides a powerful tool for the analysis of signals as well as linear time-invariant systems. The one-sided integral Laplace-Borel transform pair

$$t^k \overset{\mathscr{L}}{\Longleftrightarrow} k!\,(s^{-1})^{k+1}$$

naturally suggests a definition for the *formal Laplace-Borel transform* of a formal power series in one variable:

$$\begin{aligned} \mathscr{L}_f &: \mathbb{R}\langle\langle X \rangle\rangle \rightarrow \mathbb{R}\langle\langle X \rangle\rangle \\ &: c \mapsto \tilde{c} \\ \mathscr{B}_f &: \mathbb{R}\langle\langle X \rangle\rangle \rightarrow \mathbb{R}\langle\langle X \rangle\rangle \\ &: \tilde{c} \mapsto c, \end{aligned}$$

where the alphabet is $X = \{x_0\}$, $\tilde{c}$ is a series with coefficients $(\tilde{c}, \emptyset) = 0$, and

$$(\tilde{c}, x_0^{k+1}) = k!\,(c, x_0^k), \quad \forall\, k \geq 0.$$

This formal Laplace-Borel transform naturally suggests a transform pair for analytic signals. Any analytic signal has a power series expansion in the time domain. By applying the Laplace transform to every term in the power series, a Laurent series representation of the signal in the frequency domain can be obtained. The transform, therefore, provides a transformation between different series representation of analytic signals, one in the time domain and one in the frequency domain. As discussed earlier, the idea can be applied to analytic linear systems. As a linear time-invariant input-output system can be completely characterized by its impulse response, which is also a signal itself, the Laplace transform of a system is simply its transfer function. For nonlinear systems, however, the situation is more complicated as the usual procedure for determining the Laplace transform of a signal can not be directly applied to such a system. In this setting, the formal Laplace-Borel transform for analytic signals was first used by Fliess in [20, 23] and later by Minh in [44] to represent the input and output of a Volterra operator, which in turn produced a type of symbolic calculus for computing the output response of a nonlinear system given various inputs. What is absent in this framework, however, is the explicit notion of computing the formal transform of the input-output system represented by a Fliess operator. Thus, another general goal of the dissertation is to define this type of transform. In this way, the formal Laplace-Borel transform provides an alternative interpretation of the symbolic calculus proposed by Fliess [23] when combined with the notion of the composition product.

1.2 Literature Survey

The idea of representing functional expansions by formal power series of non-commutative variables comes from the interaction of two areas: algebraic combinatorics and iterated integral theory. The formal power series method in computer algebra dates back to the early 1960's when introduced by Schützenberger as a generalization of automata and formal languages [16, 55–57]. The theory of the iterated integral was proposed by Chen in the late 1950's. In [7], Chen defined a formal power series in m non-commutative indeterminates associated with each path in the m-dimensional Euclidean space, $\mathbb{R}^m$, for an iterated integral, and then generalized the definition to an arbitrary m-dimensional differentiable manifold in [8]. The theory of iterated integrals was later applied by Chen to discover new relationships in the algebraic structure of loop spaces [13, 14]. Motivated by the theory of iterated integrals and their rich algebraic structures, Fliess first applied the formal power series representation in 1973 to give a theory of realization for bilinear systems [18] and later in 1974 a more general realization theory for nonlinear systems [19]. In [23], the representation of a system output by a type of symbolic calculus involving iterated integrals was introduced. Specifically, a formula for computing the solution of a differential equation with a forcing function was given in terms of a functional expansion. The input-output operator defined in this particular fashion was further developed in [20, 21, 60, 66, 67]. In [28] it was shown that if the growth condition on the coefficients of the formal power series (1.9) is satisfied, the input-output mapping constitutes a well-defined operator whose domain lies in a ball in L^p space while the range lies in a ball in L^q space, where $1/p + 1/q = 1$. In addition, Sklyar and Ignatovich expressed the input-output mapping of an affine system as a series of nonlinear power moments, which corresponds to selecting different basis for the Fliess algebra [58].

Formal power series methods are also connected with other techniques in nonlinear system analysis. A few examples are the free Lie-algebraic techniques, combinatorics on words, and the differential geometric methods. The free Lie algebra dates back to the beginning of the twentieth century in the work of Campbell, Baker and Hausdorff on the exponential mapping in a Lie group described by the Campbell-Baker-Hausdorff formula [2, 5, 29]. Subsequently, extensive work has been done in free Lie algebras and their application to nonlinear control theory [4, 8, 9, 20, 40, 52, 61]. Sussmann provides an expansion of Chen series as a product of exponentials in a P. Hall basis [62]. The exponential product expansion dramatically simplifies some hard analytic results, as shown by Kawski and Sussmann in [40].

The algebraic structure of formal power series also involves combinatorics on words, a field that has grown separately within several branches of mathematics, such as group theory, and the areas of automata and formal languages [3, 16, 43, 55–57]. Much literature is also devoted to the shuffle algebra [3, 43, 49, 52], and its connection to the multiplication of two systems [21, 66]. The set of formal power series with the operations of concatenation and shuffle forms a Hopf algebra. The duality of the concatenation and shuffle implies two bialgebra structures on the set of formal power series [43, 52]. The algebraic nature of the composition product in this setting has not been explored.

Differential geometry has been used extensively in nonlinear control since the 1970's [31, 32, 46], a brief overview of which is given by Respondek in [51]. Since the natural state space of many engineering systems is a differentiable manifold, the differential geometric methods has proven to be very elegant and powerful. There is extensive research involving the state space realization of nonlinear systems over a differentiable manifold [6, 19, 21, 31, 33, 46, 60, 67]. The different notions of controllability and observability of nonlinear systems are also addressed in this setting. In [60], the existence and uniqueness of minimal realizations of a nonlinear system was studied. In [66], a precise correspondence was established between realizability of input-output operators and the existence of high order differential equations involving the derivatives of inputs and outputs. It was shown in [68] that the order of the input-output equation satisfied by a nonlinear system is no less than the minimal dimension of any observable realization of the system.

System interconnections have interested control theorists for decades. Willems studied interconnections in the context of system behaviors, which does not distinguish explicitly the direction of signal flow [63, 69, 70]. An explicit direction of signal flow, however, must be predetermined for cascades. In [22], Fliess provides an interpretation of the cascade decomposition in a state space setting using the joint notions of foliation and ideals of transitive Lie algebras. The approach Rugh applied in [53] is to consider the Volterra-Wiener type nonlinear systems as compositions of feedback-free interconnections of linear dynamics and static nonlinear elements. In [39], Kawski proposed a possible approach to nonlinear state space system feedback interconnection using the chronological algebra. In [17], Ferfera produced explicitly the generating series for cascaded interconnection of two Fliess operators in the SISO case, and showed that there always exists a series $c \circ d$ such that $y = F_c[F_d[u]] = F_{c\circ d}[u]$.

The classical integral Laplace-Borel transform provides an elegant way to analyze linear systems. The generalization of this transform also provides the possibility of algebraic representation of nonlinear systems, which, in turn, enables the solution of nonlinear problems by recursive computer algebraic procedures. Since the 1950's, much effort has been devoted to the extension of the linear system techniques to nonlinear system analysis, and some notable success has been achieved. The formal Laplace-Borel transform was used by Fliess in [20] to represent the input and output signals of a nonlinear analytic system, which in turn produced a type of symbolic calculus for computing the output response of a nonlinear system given various control inputs. Later, Lamnabhi proposed a way to compute the output response of a nonlinear system by a generalization of the Heaviside operational calculus in [41]. In [21, 23], the formal Laplace-Borel transform of the input and output signals was used to solve a nonlinear differential equation with forcing functions, and the non-commutative indeterminates in a formal power series representation were compared to the association of variables developed by Mitzel and Rugh [45], and the Fliess representation was compared to the nonlinear high order transfer function representation proposed in [53]. In [44] Minh viewed the system output signal as a function parameterized by the indeterminates of input signals and introduced an evaluation transformation to compute the temporal output response of the system given different inputs. In [59], Sternin and Shatalov described a formal Laplace-Borel transformation over the single-variable alphabet and used it to reconstruct resurgent functions. What is absent in all of these approaches, is an explicit notion of the transform of the input-output operator to characterize the nonlinear input-output system.

1.3 Problem Statement

The main goal of this dissertation is to address the following problems:

1. To describe in a unified manner the generating series for the four elementary system interconnections: the parallel, product, cascade and feedback connections. In each case, the generating series for the composite system is to be produced, and a growth condition on the coefficients of the generating series is also provided, when one exists.

2. To provide a definition of the formal Laplace-Borel transform of a Fliess operator, to characterize its basic properties, and to apply the formal Laplace-Borel transform in the analysis of system interconnections.

3. To characterize the algebraic structures of the set of formal power series and the set of Fliess operators with respect to system interconnection and the formal Laplace-Borel transform.

1.4 Dissertation Outline

The dissertation is organized as follows. In Chapter II, the basic terminologies regarding formal power series are reviewed, as well as some useful operations over the set of formal power series. In this chapter, the definition of the composition product in the multi-variable setting is given by generalizing an existing definition for a single variable composition product, and its set of known properties is expanded. The algebraic structure of the formal power series is studied in the presence of the concatenation, shuffle and composition products.

Chapter III is devoted to the interconnections of Fliess operators. Specifically, the formal definition of a Fliess operator is introduced, and the four basic interconnections of two Fliess operators are described in a unified manner: the parallel connection, product connection, cascade connection and feedback connection. In each case, the corresponding generating series is produced, and a growth condition is provided, when one exists. The analysis starts with the three non-recursive system interconnections and the corresponding binary operations on formal power series. Based on the analysis of the non-recursive connections, the feedback connection is then addressed. A new binary operation, the feedback product is introduced and characterized. A *modified* composition product is also defined in the process.

Motivated by the correspondence between operations on formal power series and system interconnections, in Chapter IV, a nonlinear extension of the classical integral Laplace-Borel transform is proposed: the formal Laplace-Borel transform of a Fliess operator. The properties of this transform are further explored, and its applications to system analysis are illustrated. The formal Laplace-Borel transform provides an alternative interpretation of the symbolic calculus introduced by Fliess et al. in [23] to compute the output response of analytic nonlinear systems. In particular, using the concept of the formal Laplace-Borel transform and the composition product, an explicit relationship is derived between the transforms of the input and output signals of a nonlinear system. Finally, it is shown that the formal Laplace-Borel transform provides an isomorphism between the semigroup of all convergent Fliess operators under composition, and the semigroup of all locally convergent formal power series under the composition product.

The main purpose of Chapter V is to provide a software implementation of the main operations described in the previous chapters. An implementation package in Maple is presented based on the ACE package developed by the ACE group in Université de Marne-la-Vallée led by Sébastien Veigneau. The ACE package provides building blocks for the binary operations on the free monoid of formal polynomials, such as the concatenation product and the shuffle product. In this software package, the following binary operations are implemented: the chronological product, composition product and modified composition product, as well as the left and right shift operators, the degree, order, and metric function in the space of formal polynomials. The results in previous chapters are illustrated by command line examples using the software.

Chapter VI summarizes the main conclusions of this dissertation and gives some ideas for future research.

CHAPTER II

FORMAL POWER SERIES

2.1 Introduction

In this chapter, the basic terminology regarding formal power series is introduced, and the necessary tools are developed for the analysis in subsequent chapters. Specifically, a set of definitions concerning formal power series are introduced; four operations over the set of formal power series are given: the concatenation, shuffle, chronological and composition products, and their properties are characterized. The basic definitions for formal power series and the shuffle product are mainly from the classical literature, e.g., [43]. The definition and properties regarding the chronological product are from [37,38]. Much research has been done on the shuffle algebra [3,43,52], while the algebraic properties of the composition product are unavailable in the literature.

2.2 Definition of Formal Power Series

It is customary in combinatorics to refer to a set of indeterminates as an *alphabet*. Its elements are called *letters*. A word over the alphabet $X = \{x_0, x_1, \ldots, x_m\}$ is a finite sequence of letters $x_{i_1}x_{i_2}\cdots x_{i_n}$ where $x_{i_r} \in X, \forall\, 1 \leq r \leq n$. The number of letters contained in a word is called the *length* of the word and is denoted by $|\cdot|$. The word with zero length is the empty word and is denoted by $\emptyset$. The set of words $x_{i_n}x_{i_{n-1}}\cdots x_{i_1}$ with length n, is denoted by X^n. When $n = 0$, $X^0 = \{\emptyset\}$. The set of all words over the alphabet X is $X^* := \bigcup_{k\geq 0} X^k$. A *formal language* is any subset of X^*.

Definition 2.2.1. *A **formal power series** over an alphabet X is any mapping of the form*

$$c : X^* \to \mathbb{R}^\ell,$$

and the set of all such mappings is denoted by $\mathbb{R}^\ell\langle\langle X\rangle\rangle$.

Definition 2.2.2. *For a formal power series, c, the image of a word $\eta \in X^*$ under c, denoted by (c, η), is called the **coefficient** of η in c. The coefficient $(c, \emptyset)$ is called the **constant term** of c. If the constant term is 0, the formal power series is called **proper**.*

In this dissertation, a formal power series $c \in \mathbb{R}^\ell\langle\langle X\rangle\rangle$, is represented by the natural formal summation in the following:

$$c = \sum_{\eta \in X^*} (c, \eta)\eta.$$

Definition 2.2.3. *The **support** of c is the set of words*

$$supp(c) := \{\eta \in X^* : (c, \eta) \neq 0\}.$$

*The **order** of c is defined by*

$$ord(c) := \begin{cases} \inf\{|\eta| : \eta \in supp(c)\} & : \quad c \neq 0 \\ \infty & : \quad c = 0. \end{cases}$$

From the definition, the order of an improper series is always 0. The set of all formal power series with finite support is called the set of all formal polynomials, and will be denoted by $\mathbb{R}\langle X\rangle$.

Definition 2.2.4. *Given an arbitrary set S, for any $c, d \in S$, define a mapping $f : S \times S \to \mathbb{R}^+ \cup \{0\}$. The function f is called an ultrametric if it satisfies the following properties:*

1. $f(c, d) \geq 0$

2. $f(c, d) = 0 \iff c = d$

3. $f(c,d) = dist(d,c)$

4. $f(c,d) \leq \max\{\, f(c,e), f(d,e)\,\}$.

Definition 2.2.5. [3] *The function $dist(\cdot,\cdot)$ over the set $\mathbb{R}^\ell\langle\langle X\rangle\rangle$ is defined by*

$$\begin{aligned} dist \;&:\; \mathbb{R}^\ell\langle\langle X\rangle\rangle \times \mathbb{R}^\ell\langle\langle X\rangle\rangle \to \mathbb{R}^+ \cup \{0\} \\ &:\; (c,d) \mapsto \sigma^{ord(c-d)}, \end{aligned}$$

where $\sigma \in (0,1)$ is an arbitrary constant.

The function $dist(\cdot,\cdot)$ can be verified to have all the four properties in Definition 2.2.4, therefore $dist(\cdot,\cdot)$ forms an ultrametric over the set $\mathbb{R}^\ell\langle\langle X\rangle\rangle$. Clearly the last property is stronger than the triangle inequality, therefore an ultrametric is always a metric.

Theorem 2.2.1. *[3] The set of all formal power series $\mathbb{R}^\ell\langle\langle X\rangle\rangle$ forms a complete metric space under the ultrametric $dist(\cdot,\cdot)$.*

Proof: To prove completeness, one needs to prove that every Cauchy sequence in $\mathbb{R}^\ell\langle\langle X\rangle\rangle$ is convergent. Let $\{c_n\}$ be a Cauchy sequence in $\mathbb{R}^\ell\langle\langle X\rangle\rangle$, then

$$\forall\, \epsilon > 0,\ \exists \text{ an integer } N \text{ such that for any } n, m > N,\ dist(c_n, c_m) < \epsilon.$$

From the definition of $dist(\cdot,\cdot)$, then, $ord(c_n - c_m) > \log_\sigma \epsilon$. Therefore,

$$\forall\, \eta \in X^* \text{ such that } |\eta| < \log_\sigma \epsilon,\ \ (c_n, \eta) = (c_m, \eta) \text{ for sufficiently large } n \text{ and } m.$$

Thus for any word η, $\{(c_n,\eta)\}$ is a Cauchy sequence in $\mathbb{R}^\ell$. Since $\mathbb{R}^\ell$ is complete, $\{(c_n,\eta)\}$ converges to a vector in $\mathbb{R}^\ell$, which will be denoted as (c,η). Let $c := \sum_{\eta\in X^*}(c,\eta)\eta$, clearly $dist(c_n, c) \to 0$ and $c \in \mathbb{R}^\ell\langle\langle X\rangle\rangle$. Therefore $\mathbb{R}^\ell\langle\langle X\rangle\rangle$ is complete. ■

$\mathbb{R}^\ell\langle\langle X\rangle\rangle$ is a vector space. With the ultrametric defined over the set $\mathbb{R}^\ell\langle\langle X\rangle\rangle$ in Definition 2.2.5, the space $(\mathbb{R}^\ell\langle\langle X\rangle\rangle, dist)$ is a bounded metric space, as the metric

$dist(c,d) \leq 1$ for all c, $d \in \mathbb{R}^{\ell}\langle\langle X \rangle\rangle$. From the definition, it is easy to see that $dist(c,d) = 1$ if and only if $(c, \emptyset) \neq (d, \emptyset)$.

2.3 Operations on Formal Power Series

In this section, four operations over the set of formal power series will be introduced: the concatenation, shuffle, chronological and composition products. The four operations provide a natural way to characterize the algebraic structures of the formal power series. The operations will also facilitate future analysis of the system interconnections. For each operation, the definition used in this dissertation is presented, in addition to some other existing alternative definitions in the literature. The properties of each operation are described, along with various relationships between the four operations.

2.3.1 Concatenation Product

The concatenation of two words is defined as the following.

Definition 2.3.1. *[3] The* ***concatenation*** *of two words η, $\xi \in X^*$ is the mapping*

$$\begin{aligned} \mathcal{C} &: X^* \times X^* \to X^* \\ &: (\eta, \xi) \mapsto \eta\xi. \end{aligned}$$

For any word $\eta \in X^*$,

$$\eta\emptyset = \emptyset\eta = \eta.$$

Hence, the empty word, $\emptyset$, is the neutral element for the concatenation operation. Intuitively, any inverse operator should involve removing one word from the other. The following shift operators can be used for this purpose.

Definition 2.3.2. *[3] For any $x_i \in X$, the **left shift**, $x_i^{-1}(\cdot)$, of a word η is defined as*

$$x_i^{-1}(\eta) := \begin{cases} \eta' & : \text{ if } \eta = x_i\eta' \\ 0 & : \text{ otherwise.} \end{cases}$$

The left concatenation inverse of a word can be written in terms of the left shift, specifically $\eta^{-1}(\eta\xi) = \xi$, $\forall \eta, \xi \in X^*$. Symmetrically, the right shift operator can be defined in the following way.

Definition 2.3.3. *[3] For any $x_i \in X$, the **right shift**, $(\cdot)\, x_i^{-1}$, of a word is defined as*

$$(\eta)x_i^{-1} := \begin{cases} \eta' & : \text{ if } \eta = \eta' x_i \\ 0 & : \text{ otherwise.} \end{cases}$$

Straightforwardly, the right inverse of the concatenation of two words can be written using the right shift operator, $(\xi\eta)\eta^{-1} = \xi$, $\forall \eta,\, \xi \in X^*$. The shift operations can be extended naturally to the formal power series as follows.

Definition 2.3.4. *For any $c \in \mathbb{R}^{\ell}\langle\langle X \rangle\rangle$, $\forall x_i \in X$, the left shift operator $x_i^{-1}(\cdot)$ of a formal power series c is*

$$x_i^{-1}(c) := \sum_{\eta \in X^*} (c, \eta)\, x_i^{-1}(\eta),$$

and the right shift operator is

$$(c)\, x_i^{-1} := \sum_{\eta \in X^*} (c, \eta)\, (\eta) x_i^{-1}.$$

The concatenation product of two formal power series can be obtained by extending Definition 2.3.1 in the following way. This operation is frequently called the Cauchy product of two series.

Definition 2.3.5. *[43] The **concatenation product** of two formal power series $c, d \in$*

$\mathbb{R}\langle\langle X\rangle\rangle$ *is the mapping*

$$\mathcal{C} \;:\; \mathbb{R}\langle\langle X\rangle\rangle \times \mathbb{R}\langle\langle X\rangle\rangle \to \mathbb{R}\langle\langle X\rangle\rangle$$
$$:\; cd \mapsto \textstyle\sum_{\eta\in X^*} \sum_{\eta=\,\xi\psi} (c,\xi)(d,\psi)\eta.$$

The unit formal power series, 1, the support of which consists of only the empty word, $\emptyset$, is the identity element for the concatenation product. In the case of formal power series and formal polynomials, the following operation is usually viewed as a type of concatenation inverse.

Definition 2.3.6. *[3] The* ***star operator*** *applied to a formal power series* $c \in \mathbb{R}\langle\langle X\rangle\rangle$ *is defined as*

$$c^\star := \sum_{n\geq 0} c^n := (1-c)^{-1},$$

where c^n *denotes the concatenation power.*

If the formal power series c is not proper, that is, the constant term $(c,\emptyset) \neq 0$, it is always possible to write $c = (c,\emptyset)(1-c')$. Then it follows that there exists a $c^{-1} \in \mathbb{R}\langle\langle X\rangle\rangle$ such that under concatenation product $cc^{-1} = 1$ and $c^{-1}c = 1$. A formal power series c is invertible if and only if it is not proper [3]. Specifically, the concatenation inverse of c can be written as

$$c^{-1} = \frac{1}{(c,\emptyset)}(1-c')^{-1} = \frac{1}{(c,\emptyset)}(c')^\star. \tag{2.1}$$

The fundamental properties of the catenation product are summarized in the following theorem.

Theorem 2.3.1. *Let* $X = \{x_0, x_1, \cdots, x_m\}$. *For all* c, d, $e \in \mathbb{R}\langle\langle X\rangle\rangle$, *and* α, $\beta \in \mathbb{R}$, *the following identities hold:*

1. *Bilinearity*
$$(\alpha c + \beta d)e = \alpha(ce) + \beta(de)$$
$$c(\alpha d + \beta e) = \alpha(cd) + \beta(ce)$$

2. *Associativity* $(cd)e = c(de) = cde$

3. *Identity* *The identity element for concatenation product C is the unit series* 1.

4. *Left shift* $(\xi\nu)^{-1}c = \nu^{-1}\left(\xi^{-1}(c)\right)$

$$x_i^{-1}(cd) = x_i^{-1}(c)d + (c, \emptyset)x_i^{-1}d$$

$$x_i^{-1}(c^\star) = x_i^{-1}(c)c^\star$$

5. *Invertibility* *A formal power series c is invertible if and only if it is not proper. The inverse of c is given in (2.1). For a mononomial, i.e., a word, the left (right) inverse can be determined by the left (right) shift.*

Proof: The properties 1,2, and 3 are straightforward. The proofs for the left shift and invertibility can be found in [3, p. 13]. ■

2.3.2 Shuffle Product

The shuffle product of two words is defined recursively using the concatenation product.

Definition 2.3.7. *[23, 49, 50] For two arbitrary letters x_j, $x_k \in X$ and two words η, $\xi \in X^*$, the* ***shuffle product*** *is defined recursively by*

$$(x_j\eta) \sqcup\!\sqcup (x_k\xi) = x_j[\eta \sqcup\!\sqcup x_k\xi] + x_k[x_j\eta \sqcup\!\sqcup \xi],$$

with $\emptyset \sqcup\!\sqcup \emptyset = \emptyset$ *and* $\xi \sqcup\!\sqcup \emptyset = \emptyset \sqcup\!\sqcup \xi = \xi$.

In this dissertation, Definition 2.3.7 is used for all the proofs and analysis. An alternative definition is given in [43], where the recursion is done from the right. It can be easily verified that the shuffle product of two words $\eta \sqcup\!\sqcup \xi$ is a formal polynomial composed of words each having length of $|\eta| + |\xi|$. Therefore, for a fixed $\nu \in X^*$, the coefficient

$$(\eta \sqcup\!\sqcup \xi, \nu) = 0 \text{ if } |\eta| + |\xi| \neq \nu.$$

The shuffle product of two words can be viewed as a mixing of the letters of the two words which preserves the order of the letters in each word. In [43], the shuffle product of two words is defined as a summation over a subset of X^* in the following.

Definition 2.3.8. *[43] For two arbitrary words η, $\xi \in X^*$, the* ***shuffle set of two words*** *is the subset of X^* defined by*

$$\begin{aligned} \mathcal{S}_{\eta \,\sqcup\!\sqcup\, \xi} = \{\, h \,|\, h = a_1 b_1 a_2 b_2 \cdots a_n b_n,\ n \geq 0, \\ a_i, b_i \in X^*,\ \eta = a_1 a_2 \cdots a_n,\ \xi = b_1 b_2 \cdots b_n \}. \end{aligned}$$

The ***shuffle product*** *is the summation over the shuffle set of the two words, which is also called the characteristic polynomial of the shuffle set.*

The definition of the shuffle product of two words can be extended to two formal power series in the following manner.

Definition 2.3.9. *[23] For any two series $c, d \in \mathbb{R}\langle\langle X \rangle\rangle$, the* ***shuffle product*** *of c and d is defined by*

$$c \,\sqcup\!\sqcup\, d = \sum_{\eta, \xi \in X^*} (c, \eta)(d, \xi) \eta \,\sqcup\!\sqcup\, \xi.$$

For vector valued series, the shuffle product is defined in a componentwise fashion. Specifically, for any two series $c, d \in \mathbb{R}^{\ell}\langle\langle X \rangle\rangle$, the ***shuffle product*** *of c and d is defined by*

$$(c \,\sqcup\!\sqcup\, d)_i = \sum_{\eta, \xi \in X^*} (c_i, \eta)(d_i, \xi) \eta \,\sqcup\!\sqcup\, \xi,$$

where $(c \,\sqcup\!\sqcup\, d)_i$, c_i and d_i are the i-th components of $c \,\sqcup\!\sqcup\, d$, c and d, respectively.

The Hopf algebra structure of the set of all formal polynomials $\mathbb{R}\langle X \rangle$ also suggests a definition of the shuffle product as the adjoint of the diagonal map [43,52]. The set $\mathbb{R}\langle X \rangle$

has an associated scalar product given by

$$\begin{aligned}(\cdot,\cdot) \;&:\; \mathbb{R}\langle X\rangle \times \mathbb{R}\langle X\rangle \to \mathbb{R} \\ &:\; (p,q) \mapsto \sum_{\eta\in X^*} (p,\eta)(q,\eta).\end{aligned}$$

The shuffle product of two formal polynomials can be viewed as a bilinear mapping

$$\begin{aligned}shuffle \;&:\; \mathbb{R}\langle X\rangle \otimes \mathbb{R}\langle X\rangle \to \mathbb{R}\langle X\rangle \\ &:\; p\otimes q \mapsto p \sqcup\!\sqcup\, q.\end{aligned}$$

The diagonal mapping is the co-product associated with the algebra $\mathbb{R}\langle\langle X\rangle\rangle$

$$\begin{aligned}\Delta \;&:\; \mathbb{R}\langle\langle X\rangle\rangle \to \mathbb{R}\langle X\rangle \otimes \mathbb{R}\langle X\rangle \\ &:\; c \mapsto \sum_{u,v\in X^*} (c, u \sqcup\!\sqcup\, v)\, u\otimes v.\end{aligned}$$

For any letter x_i, the co-product $\Delta(x_i) = x_i \otimes \emptyset + \emptyset \otimes x_i$.

Definition 2.3.10. *[43, 52] For any $c \in \mathbb{R}\langle\langle X\rangle\rangle$, the* ***shuffle product*** *of two formal polynomials p, $q \in \mathbb{R}\langle X\rangle$ is the adjoint of the diagonal map defined by*

$$(c, p \sqcup\!\sqcup\, q) = (\Delta(c), p\otimes q).$$

The concatenation product of two formal polynomials can also be viewed as a bilinear mapping

$$\begin{aligned}concatenation \;&:\; \mathbb{R}\langle X\rangle \otimes \mathbb{R}\langle X\rangle \to \mathbb{R}\langle X\rangle \\ &:\; p\otimes q \mapsto pq.\end{aligned}$$

Let Δ' denote the adjoint of the concatenation product, then $\forall\, r \in \mathbb{R}\langle\langle X\rangle\rangle$,

$$(p\,q, r) = (p\otimes q, \Delta'(r)).$$

This implies that for any series, the co-product associated with the algebra $\mathbb{R}\langle\langle X\rangle\rangle$ is

$$\begin{aligned} \Delta' &: \mathbb{R}\langle\langle X\rangle\rangle \to \mathbb{R}\langle X\rangle \otimes \mathbb{R}\langle X\rangle \\ &: c \mapsto \sum_{u,v\in X^*} (c, uv)\, u \otimes v. \end{aligned}$$

It is shown in [52] that there is a duality between the two bialgebra structures on $\mathbb{R}\langle\langle X\rangle\rangle$ via the following theorem.

Theorem 2.3.2. *[52] The adjoint of the shuffle product Δ is a homomorphism for the concatenation product. The adjoint of the concatenation product Δ' is a homomorphism for the shuffle product. That is, $\forall\, c,\, d \in \mathbb{R}\langle\langle X\rangle\rangle$,*

$$\begin{aligned} \Delta(c\,d) &= \Delta(c)\,\Delta(d) \\ \Delta'(c \sqcup\!\sqcup\, d) &= \Delta'(c) \sqcup\!\sqcup\, \Delta'(d). \end{aligned}$$

Some basic properties of the shuffle product are given below. From the duality of the concatenation and shuffle products, there exists a symmetric relationship between the properties of the shuffle product and those of the concatenation product.

Theorem 2.3.3. *Let $X = \{x_0, x_1, \cdots, x_m\}$. For all $c,\, d,\, e \in \mathbb{R}\langle\langle X\rangle\rangle$, and $\alpha,\, \beta \in \mathbb{R}$, the following identities hold:*

1. *Bilinearity*
$$(\alpha c + \beta d) \sqcup\!\sqcup\, e = \alpha(c \sqcup\!\sqcup\, e) + \beta(d \sqcup\!\sqcup\, e)$$
$$c \sqcup\!\sqcup\, (\alpha d + \beta e) = \alpha(c \sqcup\!\sqcup\, d) + \beta(c \sqcup\!\sqcup\, e)$$

2. *Commutativity* $c \sqcup\!\sqcup\, d = d \sqcup\!\sqcup\, c$

3. *Associativity* $(c \sqcup\!\sqcup\, d) \sqcup\!\sqcup\, e = c \sqcup\!\sqcup\, (d \sqcup\!\sqcup\, e)$

4. *Identity* *The identity element for the shuffle product is the unit series* 1.

5. *Left shift* $\quad x_i^{-1}(c \sqcup\!\sqcup\, d) = (x_i^{-1}c) \sqcup\!\sqcup\, d + c \sqcup\!\sqcup\, (x_i^{-1}d)$

Right shift $\quad (c \sqcup\!\sqcup\, d)x_i^{-1} = (cx_i^{-1}) \sqcup\!\sqcup\, d + c \sqcup\!\sqcup\, (dx_i^{-1})$.

Proof: Properties 1-4 can be verified by straightforward (although tedious) combinatorial calculations using Definition 2.3.7.

5. The left shift property of the shuffle product can also be proven using Definition 2.3.7 (see [66]). The first step is to prove the identity for two words. Let $\eta, \xi \in X^*$. If either $\eta = \emptyset$ or $\xi = \emptyset$, the identity is straightforward. Otherwise, when η and ξ are nonempty, then

$$x_i^{-1}(\eta \sqcup\!\sqcup\, \xi) = x_i^{-1}(x_j\eta' \sqcup\!\sqcup\, x_k\xi') = x_i^{-1}x_j(\eta' \sqcup\!\sqcup\, \xi) + x_i^{-1}x_k(\eta \sqcup\!\sqcup\, \xi').$$

Applying the left shift operator gives

$$x_i^{-1}x_j = \begin{cases} 1 & : \; if \;\; i = j \\ 0 & : \; if \;\; i \neq j, \end{cases}$$

and therefore,

$$x_i^{-1}(\eta \sqcup\!\sqcup\, \xi) = (x_i^{-1}\eta) \sqcup\!\sqcup\, \xi + \eta \sqcup\!\sqcup\, (x_i^{-1}\xi).$$

Next, for any $c, d \in \mathbb{R}\langle\langle X \rangle\rangle$,

$$\begin{aligned} x_i^{-1}(c \sqcup\!\sqcup\, d) &= \sum_{\eta,\xi \in X^*} (c,\eta)(d,\xi) x_i^{-1}(\eta \sqcup\!\sqcup\, \xi) \\ &= \sum_{\eta,\xi \in X^*} (c,\eta)(d,\xi) \left((x_i^{-1}\eta) \sqcup\!\sqcup\, \xi + \eta \sqcup\!\sqcup\, (x_i^{-1}\xi)\right) \\ &= \sum_{\eta,\xi \in X^*} (c,\eta)(d,\xi) \left((x_i^{-1}\eta) \sqcup\!\sqcup\, \xi\right) + \sum_{\eta,\xi \in X^*} (c,\eta)(d,\xi) \left(\eta \sqcup\!\sqcup\, (x_i^{-1}\xi)\right) \\ &= \sum_{\eta,\xi \in X^*} (c,x_i\eta)(d,\xi) \left(\eta \sqcup\!\sqcup\, \xi\right) + \sum_{\eta,\xi \in X^*} (c,\eta)(d,x_i\xi) \left(\eta \sqcup\!\sqcup\, \xi\right) \\ &= (x_i^{-1}c) \sqcup\!\sqcup\, d + c \sqcup\!\sqcup\, (x_i^{-1}d). \end{aligned}$$

The right shift property can be justified by an analogous procedure. ■

2.3.3 Chronological Product

A chronological product on a vector space C (over the field $\mathbb{R}$) is a bilinear operator $* : C \times C \mapsto C$ which satisfies the chronological identity

$$a * (b * c) = (a * b) * c + (b * a) * c \text{ for all } a,\, b,\, c \in C. \tag{2.2}$$

A chronological product on $\mathbb{R}\langle\langle X \rangle\rangle$ is defined recursively using the concatenation product [36–38, 40]. The right chronological product is described below.

Definition 2.3.11. *[36] The* ***right chronological product*** *of a non-empty word $\eta \in X^*\backslash\{\emptyset\}$ with a letter $x_i \in X$ is defined as*

$$\eta * x_i = \eta x_i.$$

For two non-empty words η and ξx_j, the right chronological product is defined recursively using the chronological identity

$$\begin{aligned} \eta * \xi x_j &= (\eta * \xi) * x_j + (\xi * \eta) * x_j \\ &= (\eta * \xi + \xi * \eta) x_j \end{aligned}$$

with $\eta * \emptyset = 0$ and $\emptyset * \eta = \eta$.

The definition of the right chronological product of two formal power series can be obtained in the following manner.

Definition 2.3.12. *[36] For any two series $c, d \in \mathbb{R}\langle\langle X \rangle\rangle$ with either $(c, \emptyset) \neq 0$ or $(d, \emptyset) \neq 0$, the* ***right chronological product*** *is defined as*

$$c * d = \sum_{\eta,\, \xi \in X^*} (c, \eta)(d, \xi)\, \eta * \xi.$$

It can be verified that Definition 2.3.12 satisfies the chronological identity (2.2). Similarly, one can define the left chronological product in the following manner.

Definition 2.3.13. *[37] The* ***left chronological product*** *of a non-empty word* $\eta \in X^*\backslash\{\emptyset\}$ *with a letter* $x_i \in X$ *is defined as*

$$\eta * x_i = x_i\eta.$$

For two non-empty words η *and* $x_j\xi$*, the left chronological product is defined recursively using the chronological identity*

$$\begin{aligned}\eta * x_j\xi &= (\eta * \xi) * x_j + (\xi * \eta) * x_j \\ &= x_j(\eta * \xi + \xi * \eta)\end{aligned}$$

with $\eta * \emptyset = \eta$ *and* $\emptyset * \eta = 0$.

From the definition, it is immediate that for a word with a single letter x_i, the chronological power is equal to the concatenation power, i.e.,

$$x_i^{*n} = \underbrace{(\cdots((x_i * x_i) * x_i) * \cdots) * x_i}_{n \text{ copies of } x_i} = \underbrace{x_i x_i \cdots x_i}_{n} = x_i^n.$$

The fundamental properties of the chronological product are given in the following theorem.

Theorem 2.3.4. *For all* c, d, $e \in \mathbb{R}\langle\langle X \rangle\rangle$, *and* α, $\beta \in \mathbb{R}$, *the following identities hold for both the left and the right chronological products:*

1. *Bilinearity* $$(\alpha c + \beta d) * e = \alpha(c * e) + \beta(d * e)$$ $$c * (\alpha d + \beta e) = \alpha(c * d) + \beta(c * e)$$

2. *Symmetrization [36]* $c * d + d * c = c \sqcup\!\sqcup d$

3. *Variation of the chronological identity* $c * (d * e) = (c \sqcup\!\sqcup d) * e$.

Proof: Here only the explicit proofs for left chronological product are given. The properties of the right chronological product can be proven in an analogous fashion.

1. The bilinearity property follows directly from the definition of the chronological product.

2. To prove the symmetrization property of the left chronological product of two formal power series, the symmetrization of two words is first proven. Let $\eta,\ \xi \in X^*$ with $|\eta|+|\xi| \geq 1$, that is, η and ξ can not both be empty words. Then it will be shown by induction that $\eta*\xi+\xi*\eta = \eta \sqcup\!\sqcup\, \xi$. When $|\eta|+|\xi| = 1$, there are two cases: $\eta = x_i,\ \xi = \emptyset$ and $\eta = \emptyset,\ \xi = x_i$. By the definition of the left chronological product, $x_i * \emptyset + \emptyset * x_i = x_i + 0 = x_i = x_i \sqcup\!\sqcup\, \emptyset$ and $\emptyset * x_i + x_i * \emptyset = 0 + x_i = x_i = \emptyset \sqcup\!\sqcup\, x_i$. Now suppose the symmetrization property holds for any η and ξ such that $|\eta| + |\xi| \leq k$. Consider two arbitrary words $\bar{\eta}$ and $\bar{\xi}$ such that $|\bar{\eta}| + |\bar{\xi}| = k + 1$. For the nontrivial case when $\bar{\eta}$ and $\bar{\xi}$ are both nonempty, applying the definition of left chronological product gives

$$\begin{aligned}
\bar{\eta} * \bar{\xi} + \bar{\xi} * \bar{\eta} &= \bar{\eta} * (x_j\bar{\xi}') + \bar{\xi} * (x_k\bar{\eta}') \\
&= \bar{\eta} * (\bar{\xi}' * x_j) + \bar{\xi} * (\bar{\eta}' * x_k) \\
&= (\bar{\eta} * \bar{\xi}' + \bar{\xi}' * \bar{\eta}) * x_j + (\bar{\xi} * \bar{\eta}' + \bar{\eta}' * \bar{\xi}) * x_k \\
&= x_j(\bar{\eta} \sqcup\!\sqcup\, \bar{\xi}') + x_k(\bar{\xi} \sqcup\!\sqcup\, \bar{\eta}') = (x_j\bar{\xi}') \sqcup\!\sqcup\, (x_k\bar{\eta}') \\
&= \bar{\eta} \sqcup\!\sqcup\, \bar{\xi}.
\end{aligned}$$

Therefore, for any formal power series c and d it follows that

$$\begin{aligned}
c * d + d * c &= \sum_{\eta,\xi \in X^*} (c,\eta)(d,\xi)\eta * \xi + \sum_{\eta,\xi \in X^*} (d,\xi)(c,\eta)\xi * \eta \\
&= \sum_{\eta,\xi \in X^*} (c,\eta)(d,\xi)(\eta * \xi + \xi * \eta) \\
&= \sum_{\eta,\xi \in X^*} (c,\eta)(d,\xi)(\eta \sqcup\!\sqcup\, \xi) \\
&= c \sqcup\!\sqcup\, d.
\end{aligned}$$

3. Applying the symmetrization property and the chronological identity gives the identity in part 3. ■

The chronological power of a formal power series $c \in \mathbb{R}\langle\langle X \rangle\rangle$ is defined as $c^{*n} = \underbrace{(\cdots((c * c) * c) * \cdots) * c}_{n \text{ copies of } c}$. An interesting observation by Kawski is that the shuffle power of a formal polynomial can be represented in terms of the chronological power [38]. The result is also true for formal power series.

Lemma 2.3.1. *For any $c \in \mathbb{R}\langle\langle X \rangle\rangle$, it follows that $c^{\sqcup\!\sqcup\, n} = n!\, c^{*n}$.*

Proof: The proof is adapted from the inductive procedure in [38]. The equality is trivially true when $n = 0, 1$. For $n \geq 2$, one must first show that $c * c^{*(n-1)} = (n-1) \cdot c^{*n}$. This, too, is trivially true when $n = 1, 2$. Now suppose this identity holds up to $n - 2$. Then

$$
\begin{aligned}
c * c^{*(n-1)} &= c * (c^{*(n-2)} * c) = (c * c^{*(n-2)}) * c + (c^{*(n-2)} * c) * c \\
&= (n-2) \cdot c^{*(n-1)} * c + c^{*n} = (n-1) \cdot c^{*n}.
\end{aligned}
$$

Next, it is shown inductively that $c^{\sqcup\!\sqcup\, n} = n!\, c^{*n}$ for $n \geq 2$. When $n = 2$, by the symmetrization property, $c^{\sqcup\!\sqcup\, 2} = c * c + c * c = 2\, c^{*2}$. So suppose the identity holds up to some $n - 1$. Then employing the previous identity

$$
\begin{aligned}
c^{\sqcup\!\sqcup\, n} &= c \sqcup\!\sqcup \left(c^{\sqcup\!\sqcup\, (n-1)}\right) = c \sqcup\!\sqcup \left((n-1)!\, c^{*(n-1)}\right) \\
&= (n-1)!\, c \sqcup\!\sqcup \left(c^{*(n-1)}\right) = (n-1)! \left(c * \left(c^{*(n-1)}\right) + \left(c^{*(n-1)}\right) * c\right) \\
&= (n-1)! \left((n-1) \cdot c^{*n} + c^{*n}\right) \\
&= n!\, c^{*n}.
\end{aligned}
$$

■

2.3.4 Composition Product

The composition product of two formal power series is defined recursively in terms of the shuffle product and concatenation. A definition of the composition product over an alphabet $X = \{x_0, x_1\}$ first appeared in [17]. The definition is expanded here to formal power series over an arbitrary finite alphabet.

Definition 2.3.14. *[24–27] The* ***composition product*** *of a word $\eta \in X^*$ with a formal power series $d \in \mathbb{R}^m \langle\langle X \rangle\rangle$ is*

$$\eta \circ d = \begin{cases} \eta & : |\eta|_{x_i} = 0, \ \forall i \neq 0 \\ x_0^{n+1}[d_i \sqcup\!\sqcup (\eta' \circ d)] & : \eta = x_0^n x_i \eta', \ n \geq 0, \ i \neq 0 \end{cases}$$

where $|\eta|_{x_i}$ denotes the number of symbols in η equivalent to x_i, and $d_i : \xi \mapsto (d, \xi)_i$ is the i-th component of d.

Consequently, if

$$\eta = x_0^{n_k} x_{i_k} x_0^{n_{k-1}} x_{i_{k-1}} \cdots x_0^{n_1} x_{i_1} x_0^{n_0}, \tag{2.3}$$

where $i_j \neq 0$ for $j = 1, \ldots, k$, it follows that

$$\eta \circ d = x_0^{n_k+1}[d_{i_k} \sqcup\!\sqcup x_0^{n_{k-1}+1}[d_{i_{k-1}} \sqcup\!\sqcup \cdots x_0^{n_1+1}[d_{i_1} \sqcup\!\sqcup x_0^{n_0}] \cdots]].$$

Alternatively, for any $\eta \in X^*$ of the form (2.3) one can uniquely define a set of right factors $\{\eta_0, \eta_1, \ldots, \eta_k\}$ of η by the iteration

$$\eta_{j+1} = x_0^{n_{j+1}} x_{i_{j+1}} \eta_j, \ \ \eta_0 = x_0^{n_0}, \ \ i_{j+1} \neq 0, \tag{2.4}$$

so that $\eta = \eta_k$ with $k = |\eta| - |\eta|_{x_0}$. In this setting $\eta \circ d = \eta_k \circ d$ where $\eta_{j+1} \circ d = x_0^{n_{j+1}}[d_{i_{j+1}} \sqcup\!\sqcup (\eta_j \circ d)]$ and $\eta_0 = x_0^{n_0}$.

Theorem 2.3.5. *[24–27] Given a fixed $d \in \mathbb{R}^m \langle\langle X \rangle\rangle$, the family of series $\{\eta \circ d : \eta \in X^*\}$ is locally finite, and therefore summable.*

Proof: Given an arbitrary $\eta \in X^*$ expressed in the form (2.3), it follows directly that

$$\begin{aligned} ord(\eta \circ d) &= n_0 + k + \sum_{j=1}^{k} n_j + ord(d_{i_j}) \\ &= |\eta| + \sum_{j=1}^{|\eta|-|\eta|_{x_0}} ord(d_{i_j}). \end{aligned} \tag{2.5}$$

Hence, for any $\xi \in X^*$,

$$\begin{aligned} I_d(\xi) &:= \{\eta \in X^* : (\eta \circ d, \xi) \neq 0\} \\ &\subset \{\eta \in X^* : ord(\eta \circ d) \leq |\xi|\} \\ &= \{\eta \in X^* : |\eta| + \sum_{j=1}^{|\eta|-|\eta|_{x_0}} ord(d_{i_j}) \leq |\xi|\}. \end{aligned}$$

Clearly this latter set is finite, and thus $I_d(\xi)$ is finite for all $\xi \in X^*$. This fact implies summability [3]. ∎

Definition 2.3.15. *[24–27] The* ***composition product*** *of two formal power series $c \in \mathbb{R}^{\ell}\langle\langle X \rangle\rangle$ and $d \in \mathbb{R}^{m}\langle\langle X \rangle\rangle$ is*

$$c \circ d = \sum_{\eta \in X^*} (c, \eta)\, \eta \circ d. \tag{2.6}$$

The locally finite property ensures that the composition product of two series is well-defined. The summation can also be written using the set of all right factors as described in equation (2.4). Equation (2.4) suggests a way to decompose a formal power series c, which leads to the definition of homogenous series.

Definition 2.3.16. *[24, 25, 27] Any $c \in \mathbb{R}^{\ell}\langle\langle X \rangle\rangle$ can be written unambiguously in the form*

$$c = c_0 + c_1 + c_2 + \cdots,$$

where $c_k \in \mathbb{R}^{\ell}\langle\langle X \rangle\rangle$ has the defining property that $\eta \in supp(c_k)$ only if $|\eta| - |\eta|_{x_0} = k$. Some *of the series c_k may be the zero series. When $c_0 = 0$, c is referred to as being* ***homogeneous.*** *When $c_k = 0$ for $k = 0, 1, \ldots, l-1$ and $c_l \neq 0$ then c is called* ***homogeneous of order*** *l.*

Let X^i be the set of all words of length i in X^*. For each word $\eta \in X^i$, the j-th right factor, η_j, has exactly j letters not equal to x_0. Therefore, given any $\nu \in X^*$:

$$(c \circ d, \nu) = \sum_{i,j=0}^{|\nu|} \sum_{\substack{\eta_j \in X^i \\ i \geq j}} (c, \eta_j)(\eta_j \circ d, \nu). \tag{2.7}$$

The second summation is understood to be the sum over the set of all possible j-th right factors of words of length i. This set has a familiar combinatoric interpretation. A *composition* of a positive integer N is an ordered set of positive integers $\{a_1, a_2, \ldots, a_K\}$ such that $N = a_1 + a_2 + \cdots + a_K$. (For example, 3 has the compositions $1+1+1$, $1+2$, $2+1$ and 3). For a given N and K, it is well known that there are $C_K(N) = \binom{N-1}{K-1}$ possible compositions. Now each factor $\eta_j \in X^i$, when written in the form

$$\eta_j = x_0^{n_j} x_{i_j} x_0^{n_{j-1}} x_{i_{j-1}} \cdots x_0^{n_1} x_{i_1} x_0^{n_0},$$

maps to a unique composition of $i+1$ with $j+1$ elements:

$$i + 1 = (n_0 + 1) + (n_1 + 1) + \cdots + (n_j + 1).$$

Thus, there are exactly $C_{j+1}(i+1)m^j = \binom{i}{j}m^j$ possible factors η_j in X^i, and the total number of terms in the summations of equation (2.7) is $((m+1)^{|\nu|+1} - 1)/m \approx (m+1)^{|\nu|}$.

Other elementary properties concerning the composition product are summarized in the following lemma.

Lemma 2.3.2. *For any $c \in \mathbb{R}^\ell\langle\langle X \rangle\rangle$ and $d \in \mathbb{R}^m\langle\langle X \rangle\rangle$, the following identities hold:*

1. $c \circ 0 = c_0 := \sum_{n \geq 0} (c, x_0^n)\, x_0^n$.

2. $c_0 \circ d = c_0$. *(In particular, $1 \circ d = 1$.)*

3. $c \circ 1 = c_{\not 1} := \sum_{\eta \in X^*} (c, \eta)\, x_0^{|\eta|}$. *(Therefore,* $c \circ 1 = c$ *if and only if* $c_0 = c$.*)*

4. $(x_0^n c) \circ d = x_0^n (c \circ d)$.

5. If c *is homogeneous, then* $\lim_{n \to \infty} c^{\circ n} = 0$ *in the ultrametric sense of Definition 2.2.5.*

Proof: Item 1-4 follows directly from the definition of the composition product. Only item *5* needs to be justified.

5. From equation (2.5), it is easy to see that

$$ord(\eta_k \circ d) \geq |\eta_k| + k \cdot ord(d).$$

Therefore,

$$ord(c_k \circ d) \geq \min_{\eta \in supp(c_k)} |\eta| + k \cdot ord(d) \geq ord(c_k) + k \cdot ord(d).$$

If c is homogenous, then $c = c_1 + c_2 + \cdots$. Therefore,

$$\begin{aligned} ord(c \circ c) &\geq \min\{ord(c_1 \circ c), ord(c_2 \circ c), \cdots, ord(c_n \circ c), \cdots\} \\ &\geq \min\{ord(c_1) + ord(c), ord(c_2) + 2ord(c), \cdots, ord(c_n) + n \cdot ord(c), \cdots\} \\ &\geq \min\{ord(c_1), ord(c_2), \cdots, ord(c_n), \cdots\} + ord(c) \\ &= ord(c) + ord(c) \\ &= 2\, ord(c). \end{aligned}$$

Now suppose $ord(c^{\circ k}) \geq k\, ord(c)$ holds up to some fixed k. Observe that

$$ord(c^{\circ(k+1)}) \geq ord(c) + ord(c^{\circ k}) \geq (k+1) \cdot ord(c).$$

Therefore, $ord(c^{\circ n}) \geq n \cdot ord(c)$ holds for any $n \geq 0$. Since c is homogenous, $ord(c)$ is at least greater than 1. Thus, $ord(c^{\circ n} - 0) \geq n$ or equivalently, $dist(c^{\circ n}, 0) \leq \sigma^n$. Finally,

$$0 \leq \lim_{n \to \infty} dist(c^{\circ n}, 0) \leq \lim_{n \to \infty} \sigma^n = 0,$$

that is to say, $\lim_{n \to \infty} c^{\circ n} = 0$ in the ultrametric sense of Definition 2.2.5. ■

Example 2.3.1. Suppose $c = \frac{1}{2}x_0x_1x_0 - \frac{1}{3}x_1x_0^2$ and $d = x_0$ then $c \circ d = 0$. That is, it is possible to have $c \circ d = 0$, where both c and d are *not* zero. □

The composition product can be written in terms of the left chronological product. If $\eta = x_0^n x_i \eta'$, then by Definitions 2.3.13 and 2.3.14,

$$
\begin{aligned}
\eta \circ d &= x_0^{n+1}[d_i \sqcup\!\sqcup (\eta' \circ d)] \\
&= x_0^n x_0 [d_i * (\eta' \circ d) + (\eta' \circ d) * d_i] \\
&= x_0^n \left[d_i * \left((\eta' \circ d) * x_0\right)\right] \\
&= \underbrace{\left(\cdots \left(d_i * \left((\eta' \circ d) * x_0\right) * x_0\right) * \cdots\right) * x_0}_{n+1 \text{ copies of } x_0} \\
&= x_0^n [d_i * x_0(\eta' \circ d)]. \qquad (2.8)
\end{aligned}
$$

Therefore, the composition product of a word $\eta \in X^*$ with a formal power series $d \in \mathbb{R}^m \langle\langle X \rangle\rangle$ can be written as

$$
\eta \circ d = \begin{cases} \eta & : \ |\eta|_{x_i} = 0, \ \ \forall i \neq 0 \\ x_0^n [d_i * x_0(\eta' \circ d)] & : \ \eta = x_0^n x_i \eta', \ \ n \geq 0, \ \ i \neq 0. \end{cases}
$$

The following theorem states that the composition product on $\mathbb{R}^\ell \langle\langle X \rangle\rangle \times \mathbb{R}^m \langle\langle X \rangle\rangle$ is continuous in its left argument. (Right argument continuity will be addressed later.)

Theorem 2.3.6. *[24–27] Let $\{c_i\}_{i \geq 1}$ be a sequence in $\mathbb{R}^\ell \langle\langle X \rangle\rangle$ with $\lim_{i \to \infty} c_i = c$. Then $\lim_{i \to \infty}(c_i \circ d) = c \circ d$ for any $d \in \mathbb{R}^m \langle\langle X \rangle\rangle$ in the ultrametric sense of Definition 2.2.5.*

Proof: Define the sequence of non-negative integers $k_i = ord(c_i - c)$ for $i \geq 1$. Since c is the limit of the sequence $\{c_i\}_{i \geq 1}$, $\{k_i\}_{i \geq 1}$ must have an increasing subsequence $\{k_{i_j}\}$. Now observe that

$$
dist(c_i \circ d, c \circ d) = \sigma^{ord((c_i - c) \circ d)}
$$

and

$$\begin{aligned}
ord((c_{i_j} - c) \circ d) &= ord\left(\sum_{\eta \in supp(c_{i_j} - c)} (c_{i_j} - c, \eta)\, \eta \circ d \right) \\
&\geq \inf_{\eta \in supp(c_{i_j} - c)} ord(\eta \circ d) \\
&= \inf_{\eta \in supp(c_{i_j} - c)} |\eta| + \sum_{j=1}^{|\eta| - |\eta|_{x_0}} ord(d_{i_j}) \\
&\geq k_{i_j}.
\end{aligned}$$

Thus, $dist(c_{i_j} \circ d, c \circ d) \leq \sigma^{k_{i_j}}$ for all $j \geq 1$, and $\lim_{i \to \infty} c_i \circ d = c \circ d$. ■

Some algebraic properties of the composition product are summarized in the following theorem.

Theorem 2.3.7. *For any $c, d, e \in \mathbb{R}^m \langle\langle X \rangle\rangle$ and $\alpha, \beta \in \mathbb{R}$, the following identities hold:*

1. *Linearity* $\quad (\alpha\, c + \beta\, d) \circ e = \alpha\, (c \circ e) + \beta\, (d \circ e)$
2. *Distributivity over shuffle* $\quad (c \sqcup\!\sqcup\, d) \circ e = (c \circ e) \sqcup\!\sqcup\, (d \circ e)$
3. *Associativity* $\quad (c \circ d) \circ e = c \circ (d \circ e)$.

Proof: Only the associativity property is nontrivial.

3. The first step is to prove that for all $\eta \in X^*$, $(\eta \circ d) \circ e = \eta \circ (d \circ e)$. The proof is by induction. For $\eta = \eta_0 = x_0^{n_0}$ and using the definition of the composition product

$$\begin{aligned}
(\eta_0 \circ d) \circ e &= (x_0^{n_0} \circ d) \circ e = x_0^{n_0} \circ e = x_0^{n_0}, \\
\eta_0 \circ (d \circ e) &= x_0^{n_0}.
\end{aligned}$$

Therefore, $(\eta_0 \circ d) \circ e = \eta_0 \circ (d \circ e)$. Now suppose associativity holds up to η_k as defined in (2.4). For $\eta_{k+1} = x_0^{n_{k+1}} x_{i_{k+1}} \eta_k$,

$$\begin{aligned}(\eta_{k+1} \circ d) \circ e &= (x_0^{n_{k+1}} x_{i_{k+1}} \eta_k \circ d) \circ e \\ &= \left(x_0^{n_{k+1}+1} \left[d_{i_{k+1}} \sqcup\!\sqcup (\eta_k \circ d) \right] \right) \circ e \\ &= x_0^{n_{k+1}+1} \left(\left[d_{i_{k+1}} \sqcup\!\sqcup (\eta_k \circ d) \right] \circ e \right) \\ &= x_0^{n_{k+1}+1} \left[d_{i_{k+1}} \circ e \sqcup\!\sqcup ((\eta_k \circ d) \circ e) \right] \\ &= x_0^{n_{k+1}+1} \left[(d \circ e)_{i_{k+1}} \sqcup\!\sqcup (\eta_k \circ (d \circ e)) \right] \\ &= x_0^{n_{k+1}} x_{i_{k+1}} \eta_k \circ (d \circ e) \\ &= \eta_{k+1} \circ (d \circ e).\end{aligned}$$

Therefore,

$$\begin{aligned}(c \circ d) \circ e &= \left(\sum_{\eta \in X^*} (c, \eta) \eta \circ d \right) \circ e = \sum_{\eta \in X^*} (c, \eta)(\eta \circ d) \circ e \\ &= \sum_{\eta \in X^*} (c, \eta) \eta \circ (d \circ e) = c \circ (d \circ e).\end{aligned}$$

■

In general, $c \circ (d + e) \neq c \circ d + c \circ e$, but for series whose support is a subset of $L = \{x_0^{n_1} x_i x_0^{n_0} | n_0, n_1 \in \mathbb{N}\}$, which are called ***linear series***, this is a valid identity. This is illustrated in the following example.

Example 2.3.2. Consider the alphabet $X = \{x_0, x_1\}$, series $c = x_1^2$ and $d = e$. Then

$$\begin{aligned}c \circ (d + e) &= x_1^2 \circ (2d) = x_0 \left[2d \sqcup\!\sqcup x_0(2d) \right] \\ &= 4x_0 \left[d \sqcup\!\sqcup x_0 d \right] = 2(x_0 d)^{\sqcup\!\sqcup 2}.\end{aligned}$$

However,

$$c \circ d + c \circ e = 2(c \circ d) = 2(x_1^2 \circ d)$$
$$= 2x_0 \left[d \sqcup\!\sqcup x_0 d \right] = (x_0 d)^{\sqcup\!\sqcup 2}.$$

Now for the linear series $c_1 = \sum_{n_1, n_0 \geq 0} (c, x_0^{n_1} x_i x_0^{n_0}) x_0^{n_1} x_i x_0^{n_0} = \sum_{\eta_1 \in X^*} (c, \eta_1) \eta_1$, observe that

$$\eta_1 \circ (d + e) = x_0^{n_1} x_i x_0^{n_0} \circ (d + e) = x_0^{n_1 + 1} \left[(d + e)_i \sqcup\!\sqcup x_0^{n_0} \right]$$
$$= x_0^{n_1+1} \left[(d_i \sqcup\!\sqcup x_0^{n_0}) + (e_i \sqcup\!\sqcup x_0^{n_0}) \right] = x_0^{n_1+1} (d_i \sqcup\!\sqcup x_0^{n_0}) + x_0^{n_1+1} (e_i \sqcup\!\sqcup x_0^{n_0})$$
$$= \eta_1 \circ d + \eta_1 \circ e.$$

Therefore,

$$c_1 \circ (d + e) = \sum_{\eta_1 \in X^*} (c_1, \eta_1) \left(\eta_1 \circ (d + e) \right)$$
$$= \sum_{\eta_1 \in X^*} (c_1, \eta_1)(\eta_1 \circ d) + \sum_{\eta_1 \in X^*} (c_1, \eta_1)(\eta_1 \circ e)$$
$$= c_1 \circ d + c_1 \circ e.$$

□

The composition product is not commutative in general. But in the next example, the linear series $c \in \mathbb{R} \langle\langle X \rangle\rangle$ of the form $c = \sum_{n_1 \geq 0} (c, x_0^{n_1} x_1) x_0^{n_1} x_1$ is shown to commute under composition with another such series.

Example 2.3.3. Let $X = \{x_0, x_1\}$, $c = \sum_{n \geq 0} (c, x_0^n x_1) x_0^n x_1$ and $d = \sum_{m \geq 0} (d, x_0^m x_1) x_0^m x_1$. Then the compositions of the two linear series c and d are

$$c \circ d = \sum_{\eta \in X^*} (c, \eta) \eta \circ d = \sum_{n \geq 0} (c, x_0^n x_1) x_0^{n+1} d$$
$$= \sum_{n, m \geq 0} (c, x_0^n x_1)(d, x_0^m x_1) x_0^{n+m+1} x_1,$$

and

$$d \circ c = \sum_{\eta \in X^*} (d, \eta)\eta \circ c = \sum_{m \geq 0} (d, x_0^m x_1) x_0^{m+1} c$$

$$= \sum_{n,m \geq 0} (d, x_0^m x_1)(c, x_0^n x_1) x_0^{n+m+1} x_1.$$

Clearly $c \circ d = d \circ c$. Also note that the composition product of the linear series produces the convolution sum [35]. □

Given a fixed $c \in \mathbb{R}^m \langle\langle X \rangle\rangle$, consider the mapping $\mathbb{R}^m \langle\langle X \rangle\rangle \to \mathbb{R}^m \langle\langle X \rangle\rangle : d \mapsto c \circ d$. The goal is to show that this mapping is always a contraction on $\mathbb{R}^m \langle\langle X \rangle\rangle$, i.e., that

$$dist(c \circ d, c \circ e) < dist(d, e), \quad \forall\, d, e \in \mathbb{R}^m \langle\langle X \rangle\rangle,$$

so that fixed point theorems can be applied in later analysis [30, 47, 48, 54]. Consider the following lemma.

Lemma 2.3.3. *[24–27] For any $c_k \in \mathbb{R}^m \langle\langle X \rangle\rangle$,*

$$dist(c_k \circ d, c_k \circ e) \leq \sigma^k \cdot dist(d, e), \quad \forall\, d, e \in \mathbb{R}^m \langle\langle X \rangle\rangle,$$

where c_k is a formal power series with the defining property that $\eta \in supp(c_k)$ only if $|\eta| - |\eta|_{x_0} = k$ as in Definition 2.3.16.

Proof: The proof is by induction for the nontrivial case where $c_k \neq 0$. First suppose $k = 0$. From the definition of the composition product it follows directly that $\eta \circ d = \eta$ for all $\eta \in supp(c_0)$. Therefore,

$$c_0 \circ d = \sum_{\eta \in supp(c_0)} (c_0, \eta)\, \eta \circ d = \sum_{\eta \in supp(c_0)} (c_0, \eta)\, \eta = c_0,$$

and

$$\begin{aligned} dist(c_0 \circ d, c_0 \circ e) &= dist(c_0, c_0) = 0 \\ &\leq \sigma^0 \cdot dist(d, e). \end{aligned}$$

Now fix any $k \geq 0$ and assume the claim is true for all $c_0, c_1, \ldots, c_k$. In particular, this implies that

$$ord(c_k \circ d - c_k \circ e) \geq k + ord(d - e). \tag{2.9}$$

For any $j \geq 0$, words in $supp(c_j)$ have the form η_j as defined in (2.4). Observe then that

$$\begin{aligned}
c_{k+1} \circ d - c_{k+1} \circ e &= \sum_{\eta_{k+1} \in X^*} (c_{k+1}, \eta_{k+1})\ \eta_{k+1} \circ d - (c_{k+1}, \eta_{k+1})\ \eta_{k+1} \circ e \\
&= \sum_{\eta_k, \eta_{k+1} \in X^*} (c_{k+1}, \eta_{k+1})\ \left[x_0{}^{n_{k+1}+1}[d_{i_{k+1}} ⧢ [\eta_k \circ d]] - x_0{}^{n_{k+1}+1}[e_{i_{k+1}} ⧢ [\eta_k \circ e]] \right] \\
&= \sum_{\eta_k, \eta_{k+1} \in X^*} (c_{k+1}, \eta_{k+1})\ \left[x_0{}^{n_{k+1}+1}[d_{i_{k+1}} ⧢ [\eta_k \circ d]] - x_0{}^{n_{k+1}+1}[d_{i_{k+1}} ⧢ [\eta_k \circ e]] + \right. \\
&\quad \left. x_0{}^{n_{k+1}+1}[d_{i_{k+1}} ⧢ [\eta_k \circ e]] - x_0{}^{n_{k+1}+1}[e_{i_{k+1}} ⧢ [\eta_k \circ e]] \right] \\
&= \sum_{\eta_k, \eta_{k+1} \in X^*} (c_{k+1}, \eta_{k+1})\ \left[x_0{}^{n_{k+1}+1}[d_{i_{k+1}} ⧢ [\eta_k \circ d - \eta_k \circ e]] + \right. \\
&\quad \left. x_0{}^{n_{k+1}+1}[(d_{i_{k+1}} - e_{i_{k+1}}) ⧢ [\eta_k \circ e]] \right]
\end{aligned}$$

using the fact that the shuffle product distributes over addition (componentwise). Next, applying the identity (2.5) and the inequality (2.9) with $c_k = \eta_k$, it follows that

$$\begin{aligned}
ord(c_{k+1} \circ d - c_{k+1} \circ e) &\geq \min \left\{ \inf_{\eta_{k+1} \in supp(c_{k+1})} n_{k+1} + 1 + ord(d) + k + ord(d - e), \right. \\
&\quad \left. \inf_{\eta_{k+1} \in supp(c_{k+1})} n_{k+1} + 1 + ord(d - e) + |\eta_k| + k \cdot ord(e) \right\} \\
&\geq k + 1 + ord(d - e),
\end{aligned}$$

thus,

$$dist(c_{k+1} \circ d, c_{k+1} \circ e) \leq \sigma^{k+1} \cdot dist(d, e).$$

Hence, $dist(c_k \circ d, c_k \circ e) \leq \sigma^k \cdot dist(d, e)$ holds for any $k \geq 0$. ■

Applying the above lemma leads to following result.

Lemma 2.3.4. *[24–27] If $c \in \mathbb{R}^m \langle\langle X \rangle\rangle$ then for any series $c_0' \in \mathbb{R}^m \langle\langle X_0 \rangle\rangle$,*

$$dist((c_0' + c) \circ d, (c_0' + c) \circ e) = dist(c \circ d, c \circ e), \ \ \forall\, d, e \in \mathbb{R}^m \langle\langle X \rangle\rangle. \tag{2.10}$$

If c is homogeneous of order $l \geq 1$ then

$$dist(c \circ d, c \circ e) \leq \sigma^l \cdot dist(d, e), \ \ \forall\, d, e \in \mathbb{R}^m \langle\langle X \rangle\rangle. \tag{2.11}$$

Proof: The equality is proven first. Since the ultrametric *dist* is shift-invariant:

$$\begin{aligned} dist((c_0' + c) \circ d, (c_0' + c) \circ e) &= dist\left(c_0' \circ d + c \circ d, c_0' \circ e + c \circ e\right) \\ &= dist\left(c_0' + c \circ d, c_0' + c \circ e\right) \\ &= dist\left(c \circ d, c \circ e\right). \end{aligned}$$

The inequality is proven next by first selecting any fixed $l \geq 1$ and showing inductively that it holds for any partial sum $\sum_{i=l}^{l+k} c_i$ where $k \geq 0$. When $k = 0$, Lemma 2.3.3 implies that

$$dist(c_l \circ d, c_l \circ e) \leq \sigma^l \cdot dist(d, e).$$

If the result is true for partial sums up to any fixed k then using the ultrametric property

$$dist(d, e) \leq \max\{dist(d, f), dist(f, e)\}, \ \ \forall\ d, e, f \in \mathbb{R}^m \langle\langle X \rangle\rangle,$$

it follows that

$$\begin{aligned} & dist\left(\left(\sum_{i=l}^{l+k+1} c_i\right) \circ d, \left(\sum_{i=l}^{l+k+1} c_i\right) \circ e\right) \\ &= dist\left(\left(\sum_{i=l}^{l+k} c_i\right) \circ d + c_{l+k+1} \circ d, \left(\sum_{i=l}^{l+k} c_i\right) \circ e + c_{l+k+1} \circ e\right) \\ &\leq \max\left\{ dist\left(\left(\sum_{i=l}^{l+k} c_i\right) \circ d + c_{l+k+1} \circ d, \left(\sum_{i=l}^{l+k} c_i\right) \circ d + c_{l+k+1} \circ e\right), \right. \\ & \qquad \left. dist\left(\left(\sum_{i=l}^{l+k} c_i\right) \circ d + c_{l+k+1} \circ e, \left(\sum_{i=l}^{l+k} c_i\right) \circ e + c_{l+k+1} \circ e\right) \right\} \end{aligned}$$

$$= \max\left\{dist(c_{l+k+1} \circ d, c_{l+k+1} \circ e), dist\left(\left(\sum_{i=l}^{l+k} c_i\right) \circ d, \left(\sum_{i=l}^{l+k} c_i\right) \circ e\right)\right\}$$

$$\leq \max\left\{\sigma^{l+k+1} \cdot dist(d,e), \sigma^l \cdot dist(d,e)\right\}$$

$$\leq \sigma^l \cdot dist(d,e).$$

Hence, the result holds for all $k \geq 0$. Finally the lemma is proven by noting that $c = lim_{k\to\infty} \sum_{i=l}^{l+k} c_i$ and using the left argument continuity of the composition product proven in Theorem 2.3.6 and the continuity of the ultrametric $dist(\cdot,\cdot)$.

■

The main result regarding the contractive mapping is the following.

Theorem 2.3.8. *[24–27] For any $c \in \mathbb{R}^m\langle\langle X\rangle\rangle$ the mapping $d \mapsto c \circ d$ is a contraction on $\mathbb{R}^m\langle\langle X\rangle\rangle$.*

Proof: Choose any series $d, e \in \mathbb{R}^m\langle\langle X\rangle\rangle$. If c is homogeneous of order $l \geq 1$ then the result follows directly from equation (2.11). Otherwise, observe that via equation (2.10):

$$dist(c \circ d, c \circ e) = dist\left(\left(\sum_{l=1}^{\infty} c_i\right) \circ d, \left(\sum_{l=1}^{\infty} c_i\right) \circ e\right)$$

$$\leq \sigma \cdot dist(d,e)$$

$$< dist(d,e).$$

■

An immediate consequence of the contractive mapping property is the right continuity of the composition product in the ultrametric sense.

Theorem 2.3.9. *[24–27] Let $\{d_i\}_{i\geq 1}$ be a sequence in $\mathbb{R}^m\langle\langle X\rangle\rangle$ with $\lim_{i\to\infty} d_i = d$. Then $\lim_{i\to\infty}(c \circ d_i) = c \circ d$ for all $c \in \mathbb{R}^m\langle\langle X\rangle\rangle$.*

Proof: The proof follows directly from the property of the contractive mapping.

$$\lim_{i\to\infty} dist(c \circ d_i, c \circ d) \leq \lim_{i\to\infty} dist(d_i, d) = 0.$$

■

2.4 Algebraic Structures of Formal Power Series

Using the properties of various operations over the set of formal power series $\mathbb{R}^m \langle\langle X \rangle\rangle$, a summary of the algebraic structures of $\mathbb{R}^m \langle\langle X \rangle\rangle$ is given in the following theorem.

Theorem 2.4.1. Let $\mathbb{R}^m \langle\langle X \rangle\rangle$ be the set of formal power series with the four operations defined over $\mathbb{R}^m \langle\langle X \rangle\rangle$: addition, concatenation, the shuffle product and the composition product. The following statements are true:

1. $(\mathbb{R}^m \langle\langle X \rangle\rangle, +)$ is a vector space.

2. $(\mathbb{R}^m \langle\langle X \rangle\rangle, +)$ is a commutative group.

3. $(\mathbb{R}^m \langle\langle X \rangle\rangle, \mathcal{C})$ is a monoid with the identity $c_{\mathcal{I}} = 1 = \emptyset$.

4. $(\mathbb{R}^m \langle\langle X \rangle\rangle, ⧢)$ is a commutative monoid with the identity $c_{\mathcal{I}} = 1 = \emptyset$.

5. $(\mathbb{R}^m \langle\langle X \rangle\rangle, \circ)$ is a semigroup.

6. $(\mathbb{R}^m \langle\langle X \rangle\rangle, +, ⧢)$ is a commutative ring.

7. $(\mathbb{R}^m \langle\langle X \rangle\rangle, +, ⧢)$ is an $\mathbb{R}$-algebra.

8. $(\mathbb{R}^m \langle\langle X \rangle\rangle, +, ⧢)$ is an integral domain.

Proof: Statements 1, 2 and 3 are straightforward. Statement 4 can be justified by the commutativity property in Theorem 2.3.3, and statement 5 can be proven by the associativity in Theorem 2.3.7. Statement 6 follows statement 2 and 4. Statement 7 can be justified from statement 6 and the bilinearity property of the shuffle product in Theorem 2.3.3. The proof for 8 was provided in [66, Lemma 2.1.1]. It is included here for completeness.

The following is to show that given any c, $d \in \mathbb{R}^m\langle\langle X\rangle\rangle$, one has $c \sqcup\!\sqcup d \neq 0$ whenever $c \neq 0$ and $d \neq 0$. To prove this, first order all the words $\eta = x_{i_1}x_{i_2}\cdots x_{i_k} \in X^*$ lexicographically, then take two nonzero series c, d and let

$$z_1 = x_{i_1}x_{i_2}\cdots x_{i_m}$$

and

$$z_2 = x_{j_1}x_{j_2}\cdots x_{j_n}$$

be the smallest words in the support of c and d, respectively. Let

$$\xi = x_{l_1}x_{l_2}\cdots x_{l_{m+n}}$$

be the smallest word in the support of $z_1 \sqcup\!\sqcup z_2$. Then the coefficient of ξ in $c \sqcup\!\sqcup d$ is:

$$(c \sqcup\!\sqcup d, \xi) = \sum_{\eta_k,\, \eta_l \in X^*} (c, \eta_k)(c, \eta_l)(\eta_k \sqcup\!\sqcup \eta_l, \xi).$$

As ξ, z_1 and z_2 are the smallest words on the support of $c \sqcup\!\sqcup d$, c and d respectively, one obtains

$$(c \sqcup\!\sqcup d, \xi) = (c, z_1)(d, z_2)(z_1 \sqcup\!\sqcup z_2, \xi),$$

which is nonzero since (c, z_1), (d, z_2) and $(z_1 \sqcup\!\sqcup z_2, \xi)$ are all nonzero. ■

In general, $(c, +, \circ)$ can NOT form a ring, as the composition product is not right distributive over addition, i.e., $c \circ (d + e) \neq c \circ d + c \circ e$. Nevertheless, for the subspace of linear series, the right distributivity holds, therefore, $(c_{linear}, +, \circ)$ forms a ring.

CHAPTER III

INTERCONNECTION OF ANALYTIC NONLINEAR SYSTEMS

3.1 Introduction

In this chapter, a class of nonlinear input-output operators known as *Fliess operators* is introduced. Each Fliess operator can be characterized by a formal power series. The formal power series must be locally convergent to ensure that the Fliess operator represents a well-defined system. The main problem of interest in this chapter is the interconnection of two Fliess operators. Four fundamental interconnections are presented in a unified fashion: the parallel, product, cascade and feedback connections. In each interconnection, the key issues considered are: What is the generating series for the composite system? Do the interconnected systems still have well-defined Fliess operator representations? In particular, is the local convergence property preserved under system interconnections?

The chapter is organized as follows. First, the definition of a Fliess operator is given, and the local convergence property of its generating series is introduced. In Section 3 the local convergence property of a formal power series is addressed under the shuffle and composition products. In Section 4, the four fundamental system interconnections are analyzed primarily by applying the results of Section 3. The system interconnections are divided into two groups: three *nonrecursive* connections: the parallel, product and cascade connections; and the feedback connection, which is recursive in nature. Specifically, the cascade connection of two Fliess operators is shown to be always locally convergent, and the feedback connection is always input-output locally convergent.

3.2 Definition of Fliess Operator and Generating Series

Let $X = \{x_0, x_1, \ldots, x_m\}$ denote an alphabet of $m + 1$ letters, and X^* the set of all words over X. For each $c \in \mathbb{R}^\ell \langle\langle X \rangle\rangle$, one can formally associate a corresponding m-input, ℓ-output operator F_c in the following manner. Let $p \geq 1$ and $a < b$ be given. For a measurable function $u : [a,b] \longrightarrow \mathbb{R}^m$, define $\|u\|_p = \max\{\|u_i\|_p : 1 \leq i \leq m\}$, where $\|u_i\|_p$ is the usual L_p-norm for a measurable real-valued function, u_i, defined on $[a,b]$. Let $L_p^m[a,b]$ denote the set of all measurable functions defined on $[a,b]$ having a finite $\|\cdot\|_p$ norm and $B_p^m(R)[a,b] := \{u \in L_p^m[a,b] : \|u\|_p \leq R\}$. With $t_0, T \in \mathbb{R}$ fixed and $T > 0$, define inductively for each $\eta \in X^*$ the mapping $E_\eta : L_1^m[t_0, t_0 + T] \rightarrow C[t_0, t_0 + T]$ with $E_\emptyset = 1$, and

$$E_{x_{i_k} x_{i_{k-1}} \cdots x_{i_1}}[u](t, t_0) = \int_{t_0}^{t} u_{i_k}(\tau) E_{x_{i_{k-1}} \cdots x_{i_1}}[u](\tau, t_0)\, d\tau,$$

where $u_0(t) \equiv 1$.

Definition 3.2.1. *The* ***Fliess operator*** *corresponding to a formal power series $c \in \mathbb{R}^\ell \langle\langle X \rangle\rangle$ is an input-output operator*

$$F_c[u](t) = \sum_{\eta \in X^*} (c, \eta)\, E_\eta[u](t, t_0).$$

The formal power series c is called the ***generating series*** *of the Fliess operator.*

All Volterra operators with analytic kernels, for example, are Fliess operators. In the classical literature where these operators first appeared [20, 21, 23, 33, 34, 60], it is generally assumed that the coefficients of the generating series c satisfies the following growth condition

$$|(c, \eta)| \leq K M^{|\eta|} |\eta|!, \qquad \forall \eta \in X^*, \tag{3.1}$$

where $|z| = \max\{|z_1|, |z_2|, \ldots, |z_\ell|\}$ for $z \in \mathbb{R}^\ell$, and $|\eta|$ denotes the length of η.

The following theorem states that the growth condition on the coefficients of c in (3.1) ensures that the output $F_c[u]$ converges uniformly and absolutely on a finite interval.

Theorem 3.2.1. *[28] Suppose $c \in \mathbb{R}^{\ell}\langle\langle X\rangle\rangle$ satisfies the growth condition in (3.1). Then there exists $R > 0$ and $T > 0$ such that for each $u \in B_1^m(R)[t_0, t_0 + T]$, the output*

$$y(t) = F_c[u](t) = \sum_{\eta \in X^*} (c, \eta) E_\eta[u](t, t_0)$$

is absolutely and uniformly convergent on $[t_0, t_0 + T]$. Furthermore, the function $y(t)$ is absolutely continuous on $[t_0, t_0 + T]$.

In light of the above convergence theorem, the following definition is given.

Definition 3.2.2. *A formal power series is said to be **locally convergent** if its coefficients satisfy the growth condition in (3.1).*

The set of all locally convergence series in $\mathbb{R}^{\ell}\langle\langle X\rangle\rangle$ is denoted by $\mathbb{R}^{\ell}_{LC}\langle\langle X\rangle\rangle$, and the set of Fliess operators with locally convergent generating series is denoted by $\mathcal{F}$. In [28] it is shown that any $F_c \in \mathcal{F}$ constitutes a mapping from $B_p^m(R)[t_0,\ t_0+T]$ into $B_q^{\ell}(S)[t_0,\ t_0+T]$ for sufficiently small $R, S, T > 0$, where the numbers $p, q \in \mathbb{N}^+$ are conjugate exponents, i.e., $1/p + 1/q = 1$ with $(1, \infty)$ being a conjugate pair by convention.

3.3 Local Convergence of Formal Power Series under Composition

In this section, the local convergence property of formal power series is considered under the composition product. As the composition product is defined recursively using shuffle product and concatenation, it is necessary to start from the local convergence property of the shuffle product. It is a point of reference and provides some important insight. The following theorem was proven in [66].

Theorem 3.3.1. *[66] Suppose $c, d \in \mathbb{R}_{LC}\langle\langle X \rangle\rangle$ with growth constants K_c, M_c and K_d, M_d, respectively. Then $c \sqcup\!\sqcup\, d \in \mathbb{R}_{LC}\langle\langle X \rangle\rangle$ with*

$$|(c \sqcup\!\sqcup\, d, \nu)| \leq K_c K_d M^{|\nu|}(|\nu|+1)!, \quad \forall \nu \in X^*, \tag{3.2}$$

where $M = \max\{M_c, M_d\}$.

Noting that $n+1 \leq 2^n$, $n \geq 0$, equation (3.2) can be written more conventionally as

$$|(c \sqcup\!\sqcup\, d, \nu)| \leq K_c K_d (2M)^{|\nu|}|\nu|!, \quad \forall \nu \in X^*.$$

The result is easily extended to the multi-variable case by the componentwise definition of the shuffle product on $\mathbb{R}^\ell\langle\langle X \rangle\rangle$, i.e., for all $c, d \in \mathbb{R}^\ell\langle\langle X \rangle\rangle$, define the i^{th} component $(c \sqcup\!\sqcup\, d, \nu)_i = (c_i \sqcup\!\sqcup\, d_i, \nu)$, $\forall \nu \in X^*$, $i = 1, 2, \ldots, \ell$, with c_i and d_i denote the i^{th} components of c and d, respectively. The corresponding growth constants are $K_c = \max_i\{K_{c_i}\}$, $M_c = \max_i\{M_{c_i}\}$, etc. The specific goal here is to show that $c \circ d$ is also locally convergent when the series c and d are locally convergent, and to produce an inequality analogous to (3.2). The basic properties of the shuffle product given below are essential.

Lemma 3.3.1. *[66] For $c, d \in \mathbb{R}\langle\langle X \rangle\rangle$ and any $\nu \in X^*$:*

1. $$(c \sqcup\!\sqcup\, d, \nu) = \sum_{\xi, \bar{\xi} \in X^*} (c, \xi)(d, \bar{\xi})(\xi \sqcup\!\sqcup\, \bar{\xi}, \nu) = \sum_{i=0}^{|\nu|} \sum_{\substack{\xi \in X^i \\ \bar{\xi} \in X^{|\nu|-i}}} (c, \xi)(d, \bar{\xi})(\xi \sqcup\!\sqcup\, \bar{\xi}, \nu)$$

2. $$\sum_{\substack{\xi \in X^i \\ \bar{\xi} \in X^{|\nu|-i}}} (\xi \sqcup\!\sqcup\, \bar{\xi}, \nu) = \binom{|\nu|}{i}, \quad 0 \leq i \leq |\nu|.$$

Identity 2 in this theorem is actually an inequality ($\leq$) in [66]. It also appears (unproven) in the context of counting subwords in [43, p. 139]. So a proof of this slightly stronger result is in order.

Proof of Identity 2: For any fixed $\nu \in X^*$ and $i = 0$

$$\sum_{\substack{\xi \in X^0 \\ \bar{\xi} \in X^{|\nu|}}} (\xi \sqcup\!\sqcup \bar{\xi}, \nu) = \sum_{\bar{\xi} \in X^{|\nu|}} (\bar{\xi}, \nu) = 1 = \binom{|\nu|}{0}.$$

A similar analysis hold when $i = |\nu|$. For the case where $0 < i < |\nu|$, an inductive proof on the length of $|\nu|$ will work. When $|\nu| = 0$ or $|\nu| = 1$, the claim is trivial. When $|\nu| = 2$ and $i = 1$, i.e., $0 < i < |\nu| = 2$, note that

$$\sum_{\xi, \bar{\xi} \in X} (\xi \sqcup\!\sqcup \bar{\xi}, \nu) = \sum_{x, \bar{x} \in X} (x\bar{x} + \bar{x}x, \nu) = 2 = \binom{2}{1}.$$

Next suppose the identity *2.* holds up to some fixed $|\nu| = n - 1 > 0$. Clearly when $|\nu| = n$ and either $i = 0$ or $i = n$, the identity is true. So assume that $0 < i < n$. Define $\nu = z\nu'$, $z \in X$, $\nu' \in X^{n-1}$, and $\delta_{xz} = 1$ for $x = z \in X$ and zero otherwise. Then observe that

$$\begin{aligned}
\sum_{\substack{\xi \in X^i \\ \bar{\xi} \in X^{n-i}}} (\xi \sqcup\!\sqcup \bar{\xi}, \nu) &= \sum_{\substack{\xi \in X^i \\ \bar{\xi} \in X^{n-i}}} (z^{-1}(\xi \sqcup\!\sqcup \bar{\xi}), \nu') \\
&= \sum_{\substack{x \in X \\ \xi' \in X^{i-1} \\ \bar{\xi} \in X^{n-i}}} (z^{-1}(x\xi') \sqcup\!\sqcup \bar{\xi}, \nu') + \sum_{\substack{x \in X \\ \xi \in X^i \\ \bar{\xi}' \in X^{n-i-1}}} (\xi \sqcup\!\sqcup z^{-1}(x\bar{\xi}'), \nu') \\
&= \sum_{\substack{x \in X \\ \xi' \in X^{i-1} \\ \bar{\xi} \in X^{n-i}}} \delta_{xz}(\xi' \sqcup\!\sqcup \bar{\xi}, \nu') + \sum_{\substack{x \in X \\ \xi \in X^i \\ \bar{\xi}' \in X^{n-i-1}}} \delta_{xz}(\xi \sqcup\!\sqcup \bar{\xi}', \nu') \\
&= \sum_{\substack{\xi' \in X^{i-1} \\ \bar{\xi} \in X^{n-i}}} (\xi' \sqcup\!\sqcup \bar{\xi}, \nu') + \sum_{\substack{\xi \in X^i \\ \bar{\xi}' \in X^{n-i-1}}} (\xi \sqcup\!\sqcup \bar{\xi}', \nu') \\
&= \binom{n-1}{i-1} + \binom{n-1}{i} = \binom{n}{i},
\end{aligned}$$

where in general $\xi^{-1}(\cdot)$ denotes the left shift operator in X^*. Therefore, the identity is true for all $|\nu| \geq 0$. ■

Now given any $\eta \in X^*$ of the form $\eta = x_0^{n_k} x_{i_k} x_0^{n_{k-1}} x_{i_{k-1}} \cdots x_0^{n_1} x_{i_1} x_0^{n_0}$, the set of right factors $\{\eta_0, \eta_1, \ldots, \eta_k\}$ defined by (2.4) produces a corresponding family of real-valued functions:

$$
\begin{aligned}
S_{\eta_0}(n) &= \frac{1}{|\eta_0|!}, \quad n \geq 0 \\
S_{\eta_1}(n) &= \frac{1}{(n)_{n_1+1}} S_{\eta_0}(n), \quad n \geq |\eta_1| \geq 1 \\
S_{\eta_j}(n) &= \frac{1}{(n)_{n_j+1}} \sum_{i=0}^{n-|\eta_j|} S_{\eta_{j-1}}(n-(n_j+1)-i), \quad n \geq |\eta_j| \geq j, \ \ 2 \leq j \leq k,
\end{aligned}
$$

where $(n)_i = n!/(n-i)!$ denotes the falling factorial. The next two lemmas form the core of the local convergence proof for the composition product.

Lemma 3.3.2. *[24] Suppose $c \in \mathbb{R}^{\ell}_{LC}\langle\langle X\rangle\rangle$ and $d \in \mathbb{R}^{m}_{LC}\langle\langle X\rangle\rangle$ with constants K_c, M_c and K_d, M_d, respectively. Then*

$$|(c \circ d, \nu)| \leq K_c \, \psi_{|\nu|}(K_d) \, M^{|\nu|} |\nu|!, \quad \forall \nu \in X^*, \tag{3.3}$$

where $M = \max\{M_c, M_d\}$, and $\{\psi_n(K_d)\}_{n\geq 0}$ is the set of degree n polynomials in K_d

$$\psi_n(K_d) = \sum_{i,j=0}^{n} \sum_{\substack{\eta_j \in X^i \\ i \geq j}} K_d^j \, S_{\eta_j}(n) \, |\eta_j|!, \quad n \geq 0.$$

Proof: The proof has two main steps. It is first shown that for any integer $l > 0$ and any $\eta \in X^*$ with $|\eta| \leq l$ and right factors $\{\eta_0, \eta_1, \ldots, \eta_k\}$ as defined in equation (2.4), that

$$|(\eta_j \circ d, \nu)| \leq K_d^j M_d^{-|\eta_j|} M_d^{|\nu|} \, |\nu|! \, S_{\eta_j}(|\nu|) \tag{3.4}$$

for all $0 \leq j \leq k$ and $|\eta_j| \leq |\nu| \leq l$. (Note that $(\eta_j \circ d, \nu) = 0$ and $S_{\eta_j}(|\nu|)$ is simply not defined when $|\nu| < |\eta_j|$.) This is shown by induction on j. The case $j = 0 < l$ is trivial. When $j = 1 \leq l$, the left shift operator $x_0^{-(n_1+1)} := (x_0^{n_1+1})^{-1}$ is employed. Specifically, for any ν with $|\eta_1| \leq |\nu| \leq l$ and containing the left factor $x_0^{n_1+1}$ (otherwise the claim is trivial):

$$\begin{aligned}
|(\eta_1 \circ d, \nu)| &= \left|(x_0^{n_1+1}(d_{i_1} ⧢ x_0^{n_0}), \nu)\right| \\
&= \left|(d_{i_1} ⧢ x_0^{n_0}, \underbrace{x_0^{-(n_1+1)}(\nu)}_{\nu'})\right| \\
&= \left|\sum_{\xi \in X^{|\nu'|-n_0}} (d_{i_1}, \xi)(\xi ⧢ x_0^{n_0}, \nu')\right| \\
&\leq \sum_{\xi \in X^{|\nu'|-n_0}} (K_d M_d^{|\xi|} |\xi|!)\,(\xi ⧢ x_0^{n_0}, \nu') \quad (\text{since } 0 \leq |\xi| < l) \\
&\leq K_d M_d^{|\nu'|-n_0} (|\nu'| - n_0)! \binom{|\nu'|}{n_0} \\
&= K_d M_d^{-|\eta_1|} M_d^{|\nu|} |\nu|!\, S_{\eta_1}(|\nu|).
\end{aligned}$$

Now assume that the result holds up to some fixed j where $1 \leq j \leq k-1$. Then in a similar fashion for $|\eta_{j+1}| \leq |\nu| \leq l$:

$$\begin{aligned}
|(\eta_{j+1} \circ d, \nu)| &= \left|(d_{i_{j+1}} ⧢ (\eta_j \circ d), \underbrace{x_0^{-(n_{j+1}+1)}(\nu)}_{\nu'})\right| \\
&= \left|\sum_{i=0}^{|\nu'|} \sum_{\substack{\xi \in X^i \\ \bar{\xi} \in X^{|\nu'|-i}}} (d_{i_{j+1}}, \xi)(\eta_j \circ d, \bar{\xi})(\xi ⧢ \bar{\xi}, \nu)\right|.
\end{aligned}$$

Since $(\eta_j \circ d, \bar{\xi}) = 0$ for $|\bar{\xi}| < |\eta_j|$, it follows using the coefficient bounds for d (because $0 \leq |\xi| \leq l - (j+1)$) and Lemma 3.3.1 (since $|\eta_j| \leq |\bar{\xi}| < l - (n_{j+1}+1)$) that

$$\begin{aligned}
|(\eta_{j+1} \circ d, \nu)| &\leq \sum_{i=0}^{|\nu'|-|\eta_j|} \sum_{\substack{\xi \in X^i \\ \bar{\xi} \in X^{|\nu'|-i}}} (K_d M_d^{|\xi|} |\xi|!) \cdot \left(K_d^j M_d^{-|\eta_j|} M_d^{|\bar{\xi}|} |\bar{\xi}|!\, S_{\eta_j}(|\bar{\xi}|)\right) (\xi ⧢ \bar{\xi}, \nu') \\
&= K_d^{j+1} M_d^{-|\eta_{j+1}|} M_d^{|\nu|} \sum_{i=0}^{|\nu'|-|\eta_j|} i!\,(|\nu'| - i)!\, S_{\eta_j}(|\nu'| - i) \binom{|\nu'|}{i} \\
&= K_d^{j+1} M_d^{-|\eta_{j+1}|} M_d^{|\nu|} |\nu|!\, \frac{1}{(|\nu|)_{n_{j+1}+1}} \sum_{i=0}^{|\nu'|-|\eta_j|} S_{\eta_j}(|\nu| - (n_{j+1}+1) - i) \\
&= K_d^{j+1} M_d^{-|\eta_{j+1}|} M_d^{|\nu|} |\nu|!\, S_{\eta_{j+1}}(|\nu|).
\end{aligned}$$

Hence, the claim is true for all $0 \leq j \leq k$.

In the second step of the proof, the claimed upper bound on $(c \circ d, \nu)$ is produced in terms of the polynomials $\psi_n(K_d)$. Since $\eta \in I_d(\nu)$ only if $|\eta| \leq |\nu|$, and using the inequality (3.4), it follows that

$$\begin{aligned}
|(c \circ d, \nu)| &= \left| \sum_{\substack{i,j=0 \\ i \geq j}}^{|\nu|} \sum_{\eta_j \in X^i} (c, \eta_j)(\eta_j \circ d, \nu) \right| \\
&\leq \sum_{\substack{i,j=0 \\ i \geq j}}^{|\nu|} \sum_{\eta_j \in X^i} (K_c M^{|\eta_j|} |\eta_j|!) \cdot (K_d^j M^{-|\eta_j|} M^{|\nu|} |\nu|! \, S_{\eta_j}(|\nu|)) \\
&= K_c \, \psi_{|\nu|}(K_d) \, M^{|\nu|} |\nu|!.
\end{aligned}$$

■

Lemma 3.3.3. *[24] For each right factor η_j of a given word $\eta \in X^*$, the following bound apply:*

$$0 < S_{\eta_j}(n) \leq \frac{(1+\alpha)^{n-|\eta_j|+j}}{\alpha^j |\eta_j|!}$$

for any $\alpha > 0$ and all $n \geq |\eta_j|$.

Proof: The proof is again by induction. The $j = 0$ case is trivial. When $j = 1$ observe that

$$\begin{aligned}
S_{\eta_1}(n) &= \frac{1}{(n)_{n_1+1} |\eta_0|!} \\
&\leq \frac{1}{(|\eta_1|)_{n_1+1} |\eta_0|!}, \quad n \geq |\eta_1| \\
&= \frac{1}{|\eta_1|!} \\
&\leq \left(\frac{1+\alpha}{\alpha} \right) \frac{(1+\alpha)^{n-|\eta_1|}}{|\eta_1|!}, \quad n \geq |\eta_1|.
\end{aligned}$$

Now suppose the lemma is true up to some fixed $j \geq 1$, then

$$\begin{aligned} S_{\eta_{j+1}}(n) &= \frac{1}{(n)_{n_{j+1}+1}} \sum_{i=0}^{n-|\eta_{j+1}|} S_{\eta_j}(n-(n_{j+1}+1)-i) \\ &\leq \frac{1}{(n)_{n_{j+1}+1}} \sum_{i=0}^{n-|\eta_{j+1}|} \frac{(1+\alpha)^{(n-(n_{j+1}+1)-i)-|\eta_j|+j}}{\alpha^j |\eta_j|!} \\ &\leq \frac{(1+\alpha)^j}{\alpha^j |\eta_{j+1}|!} \sum_{i=0}^{n-|\eta_{j+1}|} (1+\alpha)^{n-|\eta_{j+1}|-i}, \quad n \geq |\eta_{j+1}| \\ &\leq \frac{(1+\alpha)^{n-|\eta_{j+1}|+j+1}}{\alpha^{j+1}|\eta_{j+1}|!}. \end{aligned}$$

So the result holds for all $j \geq 0$. ■

The main local convergence theorem for the composition product is below.

Theorem 3.3.2. *[24] Suppose $c \in \mathbb{R}^{\ell}_{LC}\langle\langle X \rangle\rangle$ and $d \in \mathbb{R}^{m}_{LC}\langle\langle X \rangle\rangle$ with growth constants K_c, M_c and K_d, M_d, respectively. Then $c \circ d \in \mathbb{R}^{\ell}_{LC}\langle\langle X \rangle\rangle$ with*

$$|(c \circ d, \nu)| \leq K_c((\phi(mK_d)+1)M)^{|\nu|}(|\nu|+1)!, \quad \forall \nu \in X^*,$$

where $\phi(x) := x/2 + \sqrt{x^2/4+x}$ and $M = \max\{M_c, M_d\}$. If $mK_d \gg 1$, the growth condition approaches $K_c(mK_dM)^{|\nu|}(|\nu|+1)!$, that is

$$|(c \circ d, \nu)| \lesssim K_c(mK_dM)^{|\nu|}(|\nu|+1)!, \quad \forall \nu \in X^*.$$

Proof: In light of Lemma 3.3.2, the goal is to show that for all $n \geq 0$: $\psi_n(K_d) \leq (\phi(mK_d)+1)^n(n+1)$. Observe that applying Lemma 3.3.3 gives for any $\alpha > 0$:

$$\begin{aligned} \psi_n(K_d) &\leq \sum_{i,j=0}^{n} \sum_{\substack{\eta_j \in X^i \\ i \geq j}} K_d^j \frac{(1+\alpha)^{n-|\eta_j|+j}}{\alpha^j} \\ &\leq (1+\alpha)^n \sum_{i=0}^{n} \sum_{j=0}^{i} \binom{i}{j} \left(\frac{mK_d}{\alpha}\right)^j \left(\frac{1}{1+\alpha}\right)^{i-j} \\ &= (1+\alpha)^n \sum_{i=0}^{n} \beta^i, \end{aligned}$$

where $\beta := mK_d/\alpha + 1/(1+\alpha)$. Setting $\beta = 1$ corresponds to letting $\alpha = \phi(mK_d)$, and the first inequality of the theorem is proven. To produce the second inequality, simply note that in general $mK_d < \phi(mK_d) + 1$ with $mK_d \approx \phi(mK_d) + 1$ when $mK_d \gg 1$. (Some notable values of $\phi(mK_d) + 1$ are given in Table 3.1.) Here ϕ_g denotes the Golden Ratio $(1+\sqrt{5})/2 = 1.618\ldots\ldots$ ■

Table 3.1: Specific values of $\phi(mK_d) + 1$.

mK_d	$\phi(mK_d) + 1$
0	1
$\ll 1$	$\simeq \sqrt{mK_d} + 1$
1/2	2
1	$\phi_g + 1 = \phi_g^2 = \frac{3+\sqrt{5}}{2}$
$\gg 1$	$\approx mK_d$
$+\infty$	$+\infty$

Example 3.3.1. In some cases, the coefficient boundaries given in Theorem 3.3.2 are conservative, i.e., smaller growth constants can be produced by exploiting particular features of the series under consideration. For example, when $c \in \mathbb{R}_{LC}\langle\langle X \rangle\rangle$ is a linear series of the form $c = \sum_{n\geq 0}(c, x_0^n x_1)\, x_0^n x_1$, it can be shown directly, by writing the composition product as a convolution sum and using the fact that $\sum_{k=0}^{n} \binom{n}{k}^{-1} < 3$ for any $n \geq 0$, that

$$|(c \circ d, \nu)| < K_c K_d M^{|\nu|} |\nu|!, \quad \forall \nu \in X^*.$$

□

Example 3.3.2. Let $X = \{x_0, x_1\}$, and suppose $c = \sum_{k>0}(k!)^2\, x_1^k$. Then according to Lemma 2.3.2, $c \circ 0 = 0$ and $1 \circ c = 1$. That is, it is possible that $c \circ d$ can be locally convergent even when c or d is not. □

This last example of the composition product motivates the following definition.

Definition 3.3.1. *A series $c \in \mathbb{R}^\ell\langle\langle X\rangle\rangle$ is* **input-output locally convergent** *if for every $c_u \in \mathbb{R}^m_{LC}\langle\langle X_0\rangle\rangle$ it follows that $c \circ c_u \in \mathbb{R}^\ell_{LC}\langle\langle X_0\rangle\rangle$, where $X_0 = \{x_0\}$.*

It is immediate that every locally convergent series is input-output locally convergent, but the converse claim is only known to hold at present in certain special cases.

Lemma 3.3.4. *[24] Let $c \in \mathbb{R}^\ell\langle\langle X\rangle\rangle$ be an input-output locally convergent formal power series with non-negative coefficients. Then c is locally convergent.*

Proof: Set $c_u = 1$ and let K, M be the growth constants for the series $c \circ 1$. Then from Lemma 2.3.2, property *3*,

$$|(c \circ 1, x_0^n)| = \max_i \sum_{\eta \in X^n} (c_i, \eta) \leq KM^n\, n!, \quad \forall\, n \geq 0.$$

Thus, $|(c,\eta)| = \max_i (c_i, \eta) \leq KM^n\, n!$ for all $n \geq 0$. ■

Lemma 3.3.5. *[24] Let $c \in \mathbb{R}^\ell\langle\langle X\rangle\rangle$ be an input-output locally convergent linear series of the form $c = \sum_{j \geq 0}(c, x_0^j x_{i_j})\, x_0^j x_{i_j}$, where $i_j \in \{1, 2, \ldots, m\}$ for all $j \geq 0$. Then c is locally convergent.*

Proof: Again set $c_u = 1$ and let K, M be the growth constants for the series $c \circ 1$. Then

$$|(c \circ 1, x_0^n)| = \max_i |(c_i, x_0^{n-1} x_{i_n})| \leq KM^n\, n!,$$

and the conclusion follows. ■

3.4 System Interconnections

In this section, the interconnections of two Fliess operators are considered. The four elementary interconnections of interest are parallel, product, cascade and feedback connections, as shown in Figure 1.1. In the case of the cascade and feedback connections it is assumed that $\ell = m$. Given two locally convergent Fliess operators F_c and F_d, the general goal of this section is to describe in a unified manner the generating series for each elementary interconnection and the conditions under which each series is locally convergent. The analysis starts with the three nonrecursive interconnections: the parallel, product and cascade connections, then the feedback connection is characterized with the aid of the nonrecursive results.

3.4.1 The Nonrecursive Connections

In this section, the generating series are produced for the three nonrecursive interconnections, and the growth condition for each generating series is derived. The main results concerning the three nonrecursive interconnections are given in the following theorem.

Theorem 3.4.1. *[24] If $c, d \in \mathbb{R}^{\ell}_{LC}\langle\langle X\rangle\rangle$ then each nonrecursive interconnected input-output system shown in Figure 1.1 (a)-(c) has a Fliess operator representation generated by a locally convergent series as indicated:*

1. $F_c + F_d = F_{c+d}$
2. $F_c \cdot F_d = F_{c \,\sqcup\!\sqcup\, d}$
3. $F_c \circ F_d = F_{c \circ d}$, *where* $\ell = m$.

Proof: 1. Observe that

$$F_c[u](t) + F_d[u](t) = \sum_{\eta\in X^*} [(c,\eta) + (d,\eta)]\, E_\eta[u](t,t_0)$$
$$= F_{c+d}[u](t).$$

Since c and d are locally convergent, define $M = \max\{M_c, M_d\}$. Then for any $\eta \in X^*$ it follows that

$$|(c+d,\eta)| = |(c,\eta) + (d,\eta)|$$
$$\leq (K_c + K_d) M^{|\eta|} |\eta|!,$$

or $c + d$ is locally convergent.

2. In light of the componentwise definition of the shuffle product, it can be assumed without loss of generality that $\ell = 1$. Therefore,

$$F_c[u](t)F_d[u](t) = \sum_{\eta\in X^*} (c,\eta)E_\eta[u](t,t_0) \sum_{\xi\in X^*} (c,\xi)E_\xi[u](t,t_0)$$
$$= \sum_{\eta,\xi\in X^*} (c,\eta)(d,\xi)\, E_\eta[u](t,t_0)E_\xi[u](t,t_0)$$
$$= \sum_{\eta,\xi\in X^*} (c,\eta)(d,\xi)\, E_{\eta \sqcup\!\sqcup \xi}[u](t,t_0)$$
$$= F_{c \sqcup\!\sqcup d}[u](t).$$

Local convergence of $c \sqcup\!\sqcup d$ is provided by Theorem 3.3.1.

3. It is first shown by induction that $F_\eta \circ F_d = F_{\eta\circ d}$ for any $\eta \in X^*$ and $d \in \mathbb{R}^m\langle\langle X\rangle\rangle$. Choose any $\eta \in X^*$, and let $\{\eta_j\}$ be the corresponding set of right factors defined in (2.4). Clearly,

$$F_{\eta_j}[u](t) = E_{\eta_j}[u](t,t_0)$$
$$F_d[u](t) = \sum_{\xi\in X^*} (d,\xi)E_\xi[u](t,t_0),$$

and therefore,

$$(F_{\eta_j} \circ F_d[u])(t) = E_{\eta_j}\left[F_d[u]\right](t, t_0).$$

The following result follows directly from the definition of the composition product:

$$\begin{aligned}(F_{\eta_0} \circ F_d[u])(t) &= E_{\eta_0}[u](t, t_0) = F_{\eta_0}[u](t) \\ &= F_{\eta_0 \circ d}[u](t).\end{aligned}$$

Now assume the claim holds up to some fixed factor η_j. Then

$$\begin{aligned}(F_{\eta_{j+1}} \circ F_d[u])(t) &= E_{\underbrace{x_0 \cdots x_0}_{n_{j+1} \text{ times}} x_{i_{j+1}} \eta_j}[F_d[u]](t, t_0) \\ &= \underbrace{\int_{t_0}^{t} \cdots \int_{t_0}^{\tau_2}}_{n_{j+1}+1 \text{ times}} F_{d_{i_{j+1}}}[u](\tau_1) E_{\eta_j}[F_d[u]](\tau_1, t_0)\, d\tau_1 \ldots d\tau_{n_{j+1}+1} \\ &= \underbrace{\int_{t_0}^{t} \cdots \int_{t_0}^{\tau_2}}_{n_{j+1}+1 \text{ times}} F_{d_{i_{j+1}} \sqcup\!\sqcup (\eta_j \circ d)}[u](\tau_1)\, d\tau_1 \ldots d\tau_{n_{j+1}+1} \\ &= F_{\underbrace{x_0 \cdots x_0}_{n_{j+1}+1 \text{ times}} [d_{i_{j+1}} \sqcup\!\sqcup (\eta_j \circ d)]}[u](t) \\ &= F_{\eta_{j+1} \circ d}[u](t).\end{aligned}$$

Thus, the claim holds for $\eta = \eta_{j+1}$ and, by induction, for $\eta = \eta_k$, $\forall\, k \geq 0$. Finally,

$$\begin{aligned}(F_c \circ F_d[u])(t) &= \sum_{\eta \in X^*} (c, \eta) E_\eta[F_d[u]](t, t_0) \\ &= \sum_{\eta \in X^*} (c, \eta) F_{\eta \circ d}[u](t) \\ &= \sum_{\eta \in X^*} (c, \eta) \left[\sum_{\nu \in X^*} (\eta \circ d, \nu) E_\nu[u](t, t_0) \right] \\ &= \sum_{\nu \in X^*} \left[\sum_{\eta \in X^*} (c, \eta)(\eta \circ d, \nu) \right] E_\nu[u](t, t_0) \\ &= \sum_{\nu \in X^*} (c \circ d, \nu)\, E_\nu(t, t_0) \\ &= F_{c \circ d}[u](t).\end{aligned}$$

Local convergence of $c \circ d$ was proven in Theorem 3.3.2.

■

Two examples are considered in the following. The first involves a series which is locally convergent but does not have a finite Lie rank. In the second example, both series have a finite Lie rank. Using their corresponding state space realizations, it is possible to determine by simulation the finite escape-time for the cascade realization. This number can be compared against the known lower bound derived from the growth constants of $c \circ d$ computed in Theorem 3.3.2.

Example 3.4.1. [24] Let $X = \{x_0, x_1\}$, and c be the formal power series with coefficients

$$(c, \eta) = \begin{cases} 1 & : \ \eta = \xi\xi, \ \xi \in X^* \\ 0 & : \ \textit{otherwise.} \end{cases}$$

It is trivially locally convergent. The claim is that the Lie rank of c is not finite. Let $\mathcal{L}(X)$ denote the usual Lie algebra defined in terms of the Lie bracket on X^*: $[\eta, \xi] = \eta\xi - \xi\eta$. For any $\eta = x_{i_k} \cdots x_{i_1} \in X^*$ define the iterated Lie bracket

$$[\eta] = [x_{i_k}, [x_{i_{k-1}}, \cdots [x_{i_2}, x_{i_1}] \cdots]].$$

It is shown in [46, p. 79] that $\{[\eta] : \eta \in X^*\}$ is a spanning set for $\mathcal{L}(X)$ when viewed as a linear space. The Hankel matrix for c is $\mathcal{H}_c = diag(1, 1, \ldots)$ when its components are indexed by $X^* \times X^*$ [19]. The elements of X^* are assumed to be ordered lexicographically. Now the support of any given polynomial $[\eta] \in \mathcal{L}(X)$ is contained in $X^{|\eta|}$. Likewise, the series $\mathcal{H}_c([\eta])$ also has its support in $X^{|\eta|}$. Hence the dimension of $\mathcal{H}_c(\mathcal{L}(X))$ can not be finite, and F_c has no finite dimensional state space realization. Never the less, $c \circ c$ is well-defined and locally convergent. □

Example 3.4.2. [24] Let $X = \{x_0, x_1\}$, $c = \sum_{k\geq 0} K_c M_c^k k!\, x_1^k$ and $d = \sum_{k\geq 0} K_d M_d^k k!\, x_1^k$, where $K_c, M_c > 0$ and $K_d, M_d > 0$ are arbitrary constants. It is easily verified that the state space systems

$$\begin{aligned} \dot{z}_c &= M_c z_c^2 u_c,\ \ z_c(0) = 1 & \qquad \dot{z}_d &= M_d z_d^2 u_d,\ \ z_d(0) = 1 \\ y_c &= K_c z_c & y_d &= K_d z_d \end{aligned}$$

realize the operators $F_c : u_c \mapsto y_c$ and $F_c : u_d \mapsto y_d$, respectively, for sufficiently small inputs and intervals of time. Letting $z = [z_c^T\ \ z_d^T]^T$ it follows directly that $F_{c\circ d}$ is realized by

$$\dot{z} = f(z) + g(z)\, u, \qquad z(0) = [1\ \ 1]^T \tag{3.5}$$

$$y = h(z), \tag{3.6}$$

where

$$f(z) = \begin{pmatrix} K_d M_c z_c^2 z_d \\ 0 \end{pmatrix},\ \ g(z) = \begin{pmatrix} 0 \\ M_d z_d^2 \end{pmatrix},\ \ h(z) = K_c z_c.$$

The first few coefficients of c, d and $c \circ d$ are given in Table 3.2 along with the upper bounds on the coefficients of $c \circ d$ predicted by Theorem 3.3.2. Since these upper bounds hold for *any* series c and d with the given growth constants, they can be conservative in specific cases.

In [28] it is shown that given any series $c \in \mathbb{R}^{\ell}_{LC}\langle\langle X\rangle\rangle$, where $X = \{x_0, x_1, \ldots, x_m\}$ and $|(c,\nu)| \leq K_c M_c^{|\nu|}\, |\nu|!$, $\forall\, \nu \in X^*$, if

$$\max\{||u||_1, T\} \leq \frac{1}{(m+1)^2 M_c},$$

then $F_c[u]$ converges absolutely and uniformly on $[0, T]$. The result still holds if one has the slightly more generous growth condition $|(c,\nu)| \leq K_c M_c^{|\nu|}\, (|\nu|+1)!$. For a constant input

Table 3.2: Some coefficients (c,ν), (d,ν), $(c\circ d,\nu)$ and upper bounds for $(c\circ d,\nu)$ in Example 3.4.2 assuming $K_d \gg 1$.

ν	(c,ν)	(d,ν)	$(c\circ d,\nu)$	upper bounds for $(c\circ d,\nu)$
$\emptyset$	K_c	K_d	K_c	K_c
x_0	0	0	$K_c(K_dM_c)$	$K_c(K_dM)$
x_1	K_cM_c	K_dM_d	0	$K_c(K_dM)$
x_0^2	0	0	$K_c(K_dM_c)^2\,2!$	$K_c(K_dM)^2\,2!$
x_0x_1	0	0	$K_c(K_dM_c)M_d$	$K_c(K_dM)^2\,2!$
x_1x_0	0	0	0	$K_c(K_dM)^2\,2!$
x_1^2	$K_cM_c^2\,2!$	$K_dM_d^2\,2!$	0	$K_c(K_dM)^2\,2!$
x_0^3	0	0	$K_c(K_dM_c)^3\,3!$	$K_c(K_dM)^3\,3!$
$x_0^2x_1$	0	0	$K_c(K_dM_c)^2M_d\,2^2$	$K_c(K_dM)^3\,3!$
$x_0x_1x_0$	0	0	$K_c(K_dM_c)^2M_d\,2$	$K_c(K_dM)^3\,3!$
$x_0x_1^2$	0	0	$K_c(K_dM_c)M_d^2\,2$	$K_c(K_dM)^3\,3!$
$x_1x_0^2$	0	0	0	$K_c(K_dM)^3\,3!$
$x_1x_0x_1$	0	0	0	$K_c(K_dM)^3\,3!$
$x_1^2x_0$	0	0	0	$K_c(K_dM)^3\,3!$
x_1^3	$K_cM_c^3\,3!$	$K_dM_d^3\,3!$	0	$K_c(K_dM)^3\,3!$

$u(t)=\bar{u}$ where $|\bar{u}|\geq 1$, define

$$T_{\max}=\frac{1}{(m+1)^2M_c|\bar{u}|}. \tag{3.7}$$

Then it follows from Theorem 3.3.2 that $F_{cod}[\bar{u}]$ will always be well-defined on at least the interval $[0,T_{\max})$, where

$$T_{\max}=\frac{1}{4M_{cod}|\bar{u}|}$$

and $M_{cod}=(\phi(K_d)+1)\max\{M_c,M_d\}$. Four specific cases are described in Table 3.3. Here each $T_{\max}$ is compared against the finite escape time, t_{esc}, of the state space system (3.5)-(3.6) with $u(t) = \bar{u} = 1$, which is determined numerically (see Figure 3.1). In each case, the value of $T_{\max} < t_{esc}$, but as expected $T_{\max}$ is conservative in these examples since the upper bounds for the coefficients $(c \circ d, \nu)$ are conservative. □

Table 3.3: $T_{\max}$ and t_{esc} for specific examples of $c \circ d$ with $\bar{u} = 1$.

Case	K_c	M_c	K_d	M_d	M_{cod}	$T_{\max}$	t_{esc}	$t_{esc}/T_{\max}$
1	10	5	5	5	34.3	0.0073	0.0362	4.96
2	5	10	5	5	68.5	0.0036	0.0190	5.21
3	5	5	10	5	59.6	0.0042	0.0190	4.53
4	5	5	5	10	68.5	0.0036	0.0329	9.02

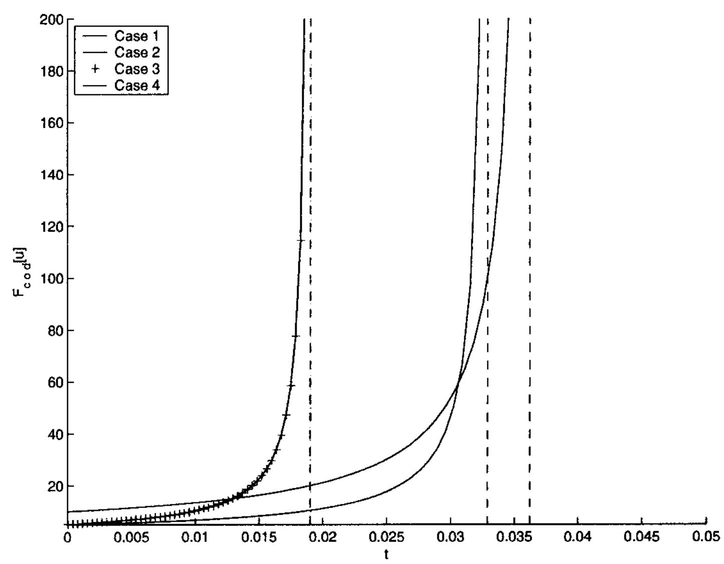


Figure 3.1: The output of $F_{cod}[u]$ when $u(t) = \bar{u} = 1$ for Cases 1-4.

3.4.2 The Feedback Connection

Given any $c, d \in \mathbb{R}^m_{LC}\langle\langle X\rangle\rangle$, the general goal of this section is to determine when there exists a y which satisfies the feedback equation (1.10), and in particular, when does there exist a generating series e so that $y = F_e[u]$ for all admissible inputs u. In the latter case, the feedback equation becomes equivalent to

$$F_e[u] = F_c[u + F_{d \circ e}[u]], \tag{3.8}$$

and the *feedback product* of c and d is defined by $c@d = e$. It is assumed throughout that $m > 0$, otherwise the feedback connection is degenerate. An initial obstacle in this analysis is that F_e is required to be the composition of two operators, F_c and $I + F_{d \circ e}$, where the second operator can *never* be represented by a Fliess operator due to the direct feed term I. This does not prevent the composition from being a Fliess operator, but to compensate for the presence of this term a *modified* composition product is needed.

Definition 3.4.1. *[24] For any $\eta \in X^*$ and $d \in \mathbb{R}^m\langle\langle X\rangle\rangle$ the* **modified composition product** *is defined as*

$$\eta \,\tilde{\circ}\, d = \begin{cases} \eta & : \ |\eta|_{x_i} = 0, \ \ \forall\, i \neq 0 \\ x_0^n x_i(\eta' \,\tilde{\circ}\, d) + x_0^{n+1}[d_i \sqcup\!\sqcup (\eta' \,\tilde{\circ}\, d)] & : \ \eta = x_0^n x_i \eta', \ \ n \geq 0, \ \ i \neq 0. \end{cases}$$

For $c \in \mathbb{R}^\ell\langle\langle X\rangle\rangle$ and $d \in \mathbb{R}^m\langle\langle X\rangle\rangle$, the definition is extended as

$$c \,\tilde{\circ}\, d = \sum_{\eta \in X^*} (c, \eta)\, \eta \,\tilde{\circ}\, d.$$

Analogous to the composition product, the following theorem ensures that the modified composition product of two series is always well-defined.

Theorem 3.4.2. *[24] Given a fixed $d \in \mathbb{R}^m\langle\langle X\rangle\rangle$, the family of series $\{\eta \,\tilde{\circ}\, d : \eta \in X^*\}$ is locally finite, and therefore summable.*

Proof: Given an arbitrary $\eta \in X^*$ expressed in the form (2.3), it follows directly that

$$\begin{aligned} ord(\eta \,\tilde{\circ}\, d) \geq ord(\eta \circ d) &= n_0 + k + \sum_{j=1}^{k} n_j + ord(d_{i_j}) \\ &= |\eta| + \sum_{j=1}^{|\eta|-|\eta|_{x_0}} ord(d_{i_j}). \end{aligned} \tag{3.9}$$

Hence, for any $\xi \in X^*$,

$$\begin{aligned} I_d(\xi) &:= \{\eta \in X^* : (\eta \,\tilde{\circ}\, d, \xi) \neq 0\} \\ &\subset \{\eta \in X^* : (\eta \circ d, \xi) \neq 0\} \\ &\subset \{\eta \in X^* : ord(\eta \circ d) \leq |\xi|\} \\ &= \{\eta \in X^* : |\eta| + \sum_{j=1}^{|\eta|-|\eta|_{x_0}} ord(d_{i_j}) \leq |\xi|\}. \end{aligned}$$

Clearly this latter set is finite, and thus $I_d(\xi)$ is finite for all $\xi \in X^*$. This fact implies summability [3]. ■

From Definition 3.4.1, the modified composition product is left distributive. Nevertheless, it is not right distributive, i.e.,

$$(c + d) \,\tilde{\circ}\, e = c \,\tilde{\circ}\, e + d \,\tilde{\circ}\, e$$

$$c \,\tilde{\circ}\, (d + e) \neq c \,\tilde{\circ}\, d + c \,\tilde{\circ}\, e.$$

The modified composition product is not associative, either. That is, $(c \,\tilde{\circ}\, d) \,\tilde{\circ}\, e \neq c \,\tilde{\circ}\, (d \,\tilde{\circ}\, e)$. This can be illustrated by the following counterexample.

Example 3.4.3. For a linear series $c_1 = \sum_{n_1, n_0 \geq 0} (c_1, x_0^{n_1} x_i x_0^{n_0}) x_0^{n_1} x_i x_0^{n_0}$, a series $d_0 = \sum_{m_0 \geq 0} (d_0, x_0^{m}) x_0^{m}$, and an arbitrary series e, the modified composition product

$$c_1 \,\tilde{\circ}\, (d_0 \,\tilde{\circ}\, e) = c_1 \,\tilde{\circ}\, d_0 = c_1 + c_1 \circ d_0,$$

while

$$(c_1 \,\tilde{\circ}\, d_0) \,\tilde{\circ}\, e = (c_1 + c_1 \circ d_0) \,\tilde{\circ}\, e$$

$$= c_1 \,\tilde{\circ}\, e + (c_1 \circ d_0) \,\tilde{\circ}\, e$$

$$= c_1 + c_1 \circ e + c_1 \circ d_0.$$

Therefore, the modified composition product is not associative. □

Theorem 3.4.3. *Let $\{c_i\}_{i\geq 1}$ be a sequence in $\mathbb{R}^m\langle\langle X\rangle\rangle$ with $\lim_{i\to\infty} c_i = c$. Then $\lim_{i\to\infty}(c_i \,\tilde{\circ}\, d) = c \,\tilde{\circ}\, d$ for all $d \in \mathbb{R}^m\langle\langle X\rangle\rangle$ in the ultrametric sense.*

Proof: Define the sequence of non-negative integers $k_i = ord(c_i - c)$ for $i \geq 1$. Since c is the limit of the sequence $\{c_i\}_{i\geq 1}$, $\{k_i\}_{i\geq 1}$ must have an increasing subsequence $\{k_{i_j}\}$. Now observe that

$$dist(c_i \,\tilde{\circ}\, d, c \,\tilde{\circ}\, d) = \sigma^{ord((c_i - c)\,\tilde{\circ}\, d)}$$

and

$$ord((c_{i_j} - c) \,\tilde{\circ}\, d) = ord\left(\sum_{\eta\in supp(c_{i_j}-c)} (c_{i_j} - c, \eta)\, \eta \,\tilde{\circ}\, d\right)$$

$$\geq \inf_{\eta\in supp(c_{i_j}-c)} ord(\eta \,\tilde{\circ}\, d)$$

$$\geq \inf_{\eta\in supp(c_{i_j}-c)} ord(\eta \circ d)$$

$$= \inf_{\eta\in supp(c_{i_j}-c)} |\eta| + \sum_{j=1}^{|\eta|-|\eta|_{x_0}} ord(d_{i_j})$$

$$\geq k_{i_j}.$$

Thus, $dist(c_{i_j} \,\tilde{\circ}\, d, c \,\tilde{\circ}\, d) \leq \sigma^{k_{i_j}}$ for all $j \geq 1$, and $\lim_{i\to\infty} c_i \,\tilde{\circ}\, d = c \,\tilde{\circ}\, d$. ■

Similar to the contractive mapping property of the composition product, it can be verified that the mapping $\mathbb{R}^m\langle\langle X\rangle\rangle \to \mathbb{R}^m\langle\langle X\rangle\rangle : d \mapsto c \,\tilde{\circ}\, d$ is always a contraction on

$\mathbb{R}^m\langle\langle X\rangle\rangle$ for a given $c \in \mathbb{R}^m\langle\langle X\rangle\rangle$, i.e.,

$$dist(c\,\tilde{\circ}\,d, c\,\tilde{\circ}\,e) < dist(d,e), \quad \forall\, d,e \in \mathbb{R}^m\langle\langle X\rangle\rangle.$$

The procedure to prove the contractive mapping of the modified composition product is a minor variation of previous results concerning the composition product, in particular, Lemma 2.3.3, Lemma 2.3.4 and Theorem 2.3.8.

Lemma 3.4.1. *For any $c_k \in \mathbb{R}^m\langle\langle X\rangle\rangle$, where c_k is a formal power series with the defining property that $\eta \in supp(c_k)$ only if $|\eta| - |\eta|_{x_0} = k$ as in Definition 2.3.16,*

$$dist(c_k\,\tilde{\circ}\,d, c_k\,\tilde{\circ}\,e) \leq \sigma^k \cdot dist(d,e), \quad \forall\, d,e \in \mathbb{R}^m\langle\langle X\rangle\rangle.$$

Proof: The proof is by induction for the nontrivial case where $c_k \neq 0$. First suppose $k = 0$. From the definition of the modified composition product it follows directly that $\eta\,\tilde{\circ}\,d = \eta$ for all $\eta \in supp(c_0)$. Therefore,

$$c_0\,\tilde{\circ}\,d = \sum_{\eta \in supp(c_0)} (c_0,\eta)\,\eta\,\tilde{\circ}\,d = \sum_{\eta \in supp(c_0)} (c_0,\eta)\,\eta = c_0,$$

and

$$\begin{aligned} dist(c_0\,\tilde{\circ}\,d, c_0\,\tilde{\circ}\,e) &= dist(c_0,c_0) = 0 \\ &\leq \sigma^0 \cdot dist(d,e). \end{aligned}$$

Now fix any $k \geq 0$ and assume the claim is true for all $c_0, c_1, \ldots, c_k$. In particular, this implies that

$$ord(c_k\,\tilde{\circ}\,d - c_k\,\tilde{\circ}\,e) \geq k + ord(d-e). \tag{3.10}$$

For any $j \geq 0$, words in $supp(c_j)$ have the form η_j as defined in (2.4). Similar to the composition product, observe then that

$$
\begin{aligned}
c_{k+1} \,\tilde{\circ}\, d - c_{k+1} \,\tilde{\circ}\, e &= \sum_{\eta_{k+1} \in X^*} (c_{k+1}, \eta_{k+1})\; \eta_{k+1} \,\tilde{\circ}\, d - (c_{k+1}, \eta_{k+1})\; \eta_{k+1} \,\tilde{\circ}\, e \\
&= \sum_{\eta_k, \eta_{k+1} \in X^*} (c_{k+1}, \eta_{k+1})\; \big[x_0{}^{n_{k+1}} x_{i_{k+1}} (\eta_k \,\tilde{\circ}\, d - \eta_k \,\tilde{\circ}\, e) + \\
&\qquad x_0{}^{n_{k+1}+1} [d_{i_{k+1}} \sqcup\!\sqcup (\eta_k \,\tilde{\circ}\, d) - e_{i_{k+1}} \sqcup\!\sqcup (\eta_k \,\tilde{\circ}\, e)]\big] \\
&= \sum_{\eta_k, \eta_{k+1} \in X^*} (c_{k+1}, \eta_{k+1})\; \big[x_0{}^{n_{k+1}} x_{i_{k+1}} (\eta_k \,\tilde{\circ}\, d - \eta_k \,\tilde{\circ}\, e) + x_0{}^{n_{k+1}+1} \\
&\qquad [d_{i_{k+1}} \sqcup\!\sqcup (\eta_k \,\tilde{\circ}\, d) - d_{i_{k+1}} \sqcup\!\sqcup (\eta_k \,\tilde{\circ}\, e)] + [d_{i_{k+1}} \sqcup\!\sqcup (\eta_k \,\tilde{\circ}\, e) - e_{i_{k+1}} \sqcup\!\sqcup (\eta_k \,\tilde{\circ}\, e)]\big] \\
&= \sum_{\eta_k, \eta_{k+1} \in X^*} (c_{k+1}, \eta_{k+1})\; \big[x_0{}^{n_{k+1}} x_{i_{k+1}} (\eta_k \,\tilde{\circ}\, d - \eta_k \,\tilde{\circ}\, e) + x_0{}^{n_{k+1}+1} \\
&\qquad [d_{i_{k+1}} \sqcup\!\sqcup (\eta_k \,\tilde{\circ}\, d - \eta_k \,\tilde{\circ}\, e) + x_0{}^{n_{k+1}+1} [(d_{i_{k+1}} - e_{i_{k+1}}) \sqcup\!\sqcup (\eta_k \,\tilde{\circ}\, e)]\big]
\end{aligned}
$$

using the fact that the shuffle product distributes over addition (componentwise). Next, applying the inequalities (3.9) and (3.10) with $c_k = \eta_k$, it follows that

$$
\begin{aligned}
ord(c_{k+1} \,\tilde{\circ}\, d - c_{k+1} \,\tilde{\circ}\, e) \geq \min \Big\{ &\inf_{\eta_{k+1} \in supp(c_{k+1})} n_{k+1} + 1 + k + ord(d - e), \\
&\inf_{\eta_{k+1} \in supp(c_{k+1})} n_{k+1} + 1 + ord(d_{i_{k+1}}) + k + ord(d - e), \\
&\inf_{\eta_{k+1} \in supp(c_{k+1})} n_{k+1} + 1 + ord(d_{i_{k+1}} - e_{i_{k+1}}) + ord(\eta_k \,\tilde{\circ}\, e) \Big\} \\
\geq \min \Big\{ &\inf_{\eta_{k+1} \in supp(c_{k+1})} n_{k+1} + 1 + k + ord(d - e), \\
&\inf_{\eta_{k+1} \in supp(c_{k+1})} n_{k+1} + 1 + ord(d_{i_{k+1}}) + k + ord(d - e), \\
&\inf_{\eta_{k+1} \in supp(c_{k+1})} n_{k+1} + 1 + ord(d_{i_{k+1}} - e_{i_{k+1}}) + |\eta_k| + k \cdot ord(e) \Big\} \\
\geq k + 1 + ord(d - e)&,
\end{aligned}
$$

thus,

$$
dist(c_{k+1} \,\tilde{\circ}\, d, c_{k+1} \,\tilde{\circ}\, e) \leq \sigma^{k+1} \cdot dist(d, e).
$$

Hence, $dist(c_k \tilde{\circ}\, d, c_k \tilde{\circ}\, e) \leq \sigma^k \cdot dist(d,e)$ holds for any $k \geq 0$. ■

Applying the above lemma leads to the following result.

Lemma 3.4.2. *If $c \in \mathbb{R}^m \langle\langle X \rangle\rangle$ then for any series $c_0' \in \mathbb{R}^m \langle\langle X_0 \rangle\rangle$*

$$dist((c_0' + c) \tilde{\circ}\, d, (c_0' + c) \tilde{\circ}\, e) = dist(c \tilde{\circ}\, d, c \tilde{\circ}\, e), \;\; \forall\, d, e \in \mathbb{R}^m \langle\langle X \rangle\rangle. \tag{3.11}$$

If c is homogeneous of order $l \geq 1$ then

$$dist(c \tilde{\circ}\, d, c \tilde{\circ}\, e) \leq \sigma^l \cdot dist(d,e), \;\; \forall\, d, e \in \mathbb{R}^m \langle\langle X \rangle\rangle. \tag{3.12}$$

Proof: The equality is proven first. Since the metric *dist* is shift-invariant:

$$\begin{aligned} dist((c_0' + c) \tilde{\circ}\, d, (c_0' + c) \tilde{\circ}\, e) &= dist\left(c_0' \tilde{\circ}\, d + c \tilde{\circ}\, d, c_0' \tilde{\circ}\, e + c \circ e\right) \\ &= dist\left(c_0' + c \tilde{\circ}\, d, c_0' + c \tilde{\circ}\, e\right) \\ &= dist\left(c \tilde{\circ}\, d, c \tilde{\circ}\, e\right). \end{aligned}$$

The inequality is proven next by first selecting any fixed $l \geq 1$ and showing inductively that it holds for any partial sum $\sum_{i=l}^{l+k} c_i$ where $k \geq 0$. When $k = 0$ Lemma 3.4.1 implies that

$$dist(c_l \tilde{\circ}\, d, c_l \tilde{\circ}\, e) \leq \sigma^l \cdot dist(d,e).$$

If the result is true for partial sums up to any fixed k then using the ultrametric property

$$dist(d,e) \leq \max\{dist(d,f), dist(f,e)\}, \;\; \forall\, d, e, f \in \mathbb{R}^m \langle\langle X \rangle\rangle,$$

it follows that

$$\begin{aligned} & dist\left(\left(\sum_{i=l}^{l+k+1} c_i\right) \tilde{\circ}\, d, \;\left(\sum_{i=l}^{l+k+1} c_i\right) \tilde{\circ}\, e\right) \\ &= dist\left(\left(\sum_{i=l}^{l+k} c_i\right) \tilde{\circ}\, d + c_{l+k+1} \tilde{\circ}\, d, \;\left(\sum_{i=l}^{l+k} c_i\right) \tilde{\circ}\, e + c_{l+k+1} \tilde{\circ}\, e\right) \end{aligned}$$

$$\leq \max\left\{ dist\left(\left(\sum_{i=l}^{l+k} c_i\right) \,\tilde{\circ}\, d + c_{l+k+1} \,\tilde{\circ}\, d,\ \left(\sum_{i=l}^{l+k} c_i\right) \,\tilde{\circ}\, d + c_{l+k+1} \,\tilde{\circ}\, e\right),\right.$$
$$\left. dist\left(\left(\sum_{i=l}^{l+k} c_i\right) \,\tilde{\circ}\, d + c_{l+k+1} \,\tilde{\circ}\, e,\ \left(\sum_{i=l}^{l+k} c_i\right) \,\tilde{\circ}\, e + c_{l+k+1} \,\tilde{\circ}\, e\right)\right\}$$
$$= \max\left\{ dist(c_{l+k+1} \,\tilde{\circ}\, d, c_{l+k+1} \,\tilde{\circ}\, e),\ dist\left(\left(\sum_{i=l}^{l+k} c_i\right) \,\tilde{\circ}\, d,\ \left(\sum_{i=l}^{l+k} c_i\right) \,\tilde{\circ}\, e\right)\right\}$$
$$\leq \max\left\{\sigma^{l+k+1} \cdot dist(d,e),\ \sigma^l \cdot dist(d,e)\right\}$$
$$\leq \sigma^l \cdot dist(d,e).$$

Hence, the result holds for all $k \geq 0$. Finally the lemma is proven by noting that $c = lim_{k\to\infty} \sum_{i=l}^{l+k} c_i$ and using the left argument continuity of the modified composition product proven in Theorem 3.4.3 and the continuity of the metric $dist(\cdot,\cdot)$. ■

Now, the result regarding the contractive mapping is readily given from Lemma 3.4.1 and Lemma 3.4.2.

Theorem 3.4.4. *[24] For any $c \in \mathbb{R}^m\langle\langle X\rangle\rangle$ the mapping $d \mapsto c \,\tilde{\circ}\, d$ is a contraction on $\mathbb{R}^m\langle\langle X\rangle\rangle$.*

Proof: Choose any series $d, e \in \mathbb{R}^m\langle\langle X\rangle\rangle$. If c is homogeneous of order $l \geq 1$ then the result follows directly from equation (3.12). Otherwise, observe that via equation (3.11):

$$dist(c \,\tilde{\circ}\, d, c \,\tilde{\circ}\, e) = dist\left(\left(\sum_{l=1}^{\infty} c_i\right) \,\tilde{\circ}\, d,\ \left(\sum_{l=1}^{\infty} c_i\right) \,\tilde{\circ}\, e\right)$$
$$\leq \sigma \cdot dist(d,e)$$
$$< dist(d,e).$$

■

Following the identical procedure as for the composition product, the right argument continuity of the modified composition product can be readily obtained from the contractive mapping property.

Theorem 3.4.5. *Let* $\{d_i\}_{i\geq 1}$ *be a sequence in* $\mathbb{R}^m\langle\langle X\rangle\rangle$ *with* $\lim_{i\to\infty} d_i = d$. *Then* $\lim_{i\to\infty}(c\,\tilde{\circ}\,d_i) = c\,\tilde{\circ}\,d$ *for all* $c \in \mathbb{R}^m\langle\langle X\rangle\rangle$.

Proof: From the contractive mapping property of the modified composition product,

$$\lim_{i\to\infty} dist(c\,\tilde{\circ}\,d_i, c\,\tilde{\circ}\,d) \leq \lim_{i\to\infty} dist(d_i, d) = 0.$$

■

The following theorem states that there is a system interconnection corresponding to the modified composition product, as shown in Figure 3.2.

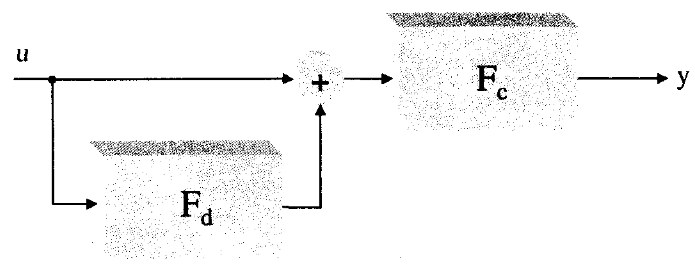


Figure 3.2: The modified composition interconnection.

Theorem 3.4.6. *[24] For any* $c \in \mathbb{R}^{\ell}_{LC}\langle\langle X\rangle\rangle$ *and* $d \in \mathbb{R}^{m}_{LC}\langle\langle X\rangle\rangle$*, it follows that*

$$F_{c\,\tilde{\circ}\,d}[u] = F_c[u + F_d[u]]$$

for all admissible u.

Proof: The result is verified directly by inserting the direct feed term into the proof of Theorem 3.4.1, part *3*. ■

The first main result for the analysis of the feedback interconnection is given next.

Theorem 3.4.7. *[24] Let* c, d *be fixed series in* $\mathbb{R}^m\langle\langle X\rangle\rangle$. *Then:*

1. *The mapping*

$$S \;:\; \mathbb{R}^m\!\ll\! X\!\gg \to \mathbb{R}^m\!\ll\! X\!\gg$$
$$\;:\; e_i \mapsto e_{i+1} = c\,\tilde{\circ}\,(d \circ e_i) \qquad (3.13)$$

has a unique fixed point in $\mathbb{R}^m\!\ll\! X\!\gg$, $c@d = \lim_{i\to\infty} e_i$, *which is independent of* e_0.

2. *If* c, d *and* $c@d$ *are locally convergent then* $F_{c@d}$ *satisfies the feedback equation (3.8).*

Proof:

1. The mapping S is a contraction since by Theorems 2.3.8 and 3.4.4,

$$\begin{aligned} dist(S(e_i), S(e_j)) &< dist(d \circ e_i, d \circ e_j) \\ &< dist(e_i, e_j). \end{aligned}$$

Therefore, the mapping S has a unique fixed point $c@d$ that is independent of e_0, i.e.,

$$c@d = c\,\tilde{\circ}\,(d \circ (c@d)). \qquad (3.14)$$

2. From the stated assumptions concerning c, d and $c@d$ it follows that

$$\begin{aligned} F_{c@d}[u] &= F_{c\,\tilde{\circ}\,(d\circ(c@d))}[u] \\ &= F_c[u + F_d[F_{c@d}[u]]] \end{aligned}$$

for any admissible u. ■

Example 3.4.4. If either one of the subsystems in a feedback connection is a system with a generating series over the alphabet $X_0 = \{x_0\}$, that is, the output $y(t)$ is a function independent of the input $u(t)$. Then the feedback connection degenerates to a modified composition connection.

1. $c_0@d = c_0\,\tilde{\circ}\,d = c_0$

2. $c@d_0 = c\,\tilde{\circ}\,d_0$.

In this case, there is no information actually transmitted in the feedback path. □

Example 3.4.5. Suppose c is a linear series and d is arbitrary. Then $c@d = \lim_{i\to\infty} e_i$, where

$$\begin{aligned} e_{i+1} &= c\,\tilde{\circ}\,(d \circ e_i) \\ &= c + (c \circ d) \circ e_i. \end{aligned}$$

If c and d are both linear, setting $e_0 = c$ gives

$$c@d = c + \sum_{k=1}^{\infty} (c \circ d)^{\circ k} \circ c, \tag{3.15}$$

Using the associativity property of the composition product, the feedback product

$$c@d = \sum_{n\geq 0} c \circ (d \circ c)^{\circ n} = c \circ \sum_{n\geq 0} (d \circ c)^{\circ n},$$

where $c^{\circ k}$ denotes k copies of c composed $k-1$ times. Since $(c, \emptyset) = 0$, applying Lemma 2.3.2 part *5* gives

$$((c \circ d)^{\circ k}, \nu) = 0, \;\; \forall\, k > |\nu|.$$

Hence,

$$(c@d, \nu) = (c, \nu) + \sum_{k=1}^{|\nu|-1} ((c \circ d)^{\circ k} \circ c, \nu).$$

Now,

$$\begin{aligned} c@d &= \sum_{k\geq 0} c \circ (d \circ c)^{\circ k} = c \circ \sum_{k\geq 0} (d \circ c)^{\circ k} \\ &= \sum_{k\geq 0} (c \circ d)^{\circ k} \circ c = \left\{ \sum_{k\geq 0} (c \circ d)^{\circ k} \right\} \circ c. \end{aligned}$$

This is analogous to the transfer function representation of the feedback connection in the linear time-invariant case

$$\begin{aligned} G@H &= G(1-HG)^{-1} = G\sum_{k\geq 0}(HG)^k \\ &= (1-GH)^{-1}G = \sum_{k\geq 0}(GH)^k G. \end{aligned}$$

□

Example 3.4.6. A special class of feedback system is the unity feedback system. If the external input signal $u(t) = 0$, the self-excited loop is described by $e = lim_{k\to\infty} c^{\circ k} \circ 0$, where c is the generating series for the open-loop system. Consider the system as shown in Figure 3.3 with the initial condition $y(0) = 1$.

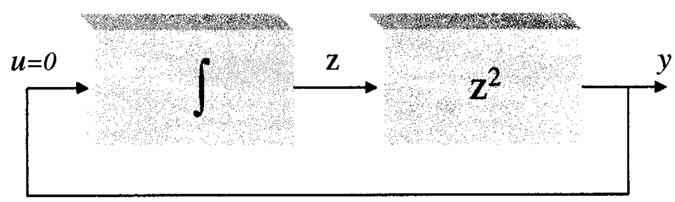


Figure 3.3: Self-excited loop for Example 3.4.6.

The open-loop system has the following state space realization

$$\begin{cases} \dot{z} = u, & z(0) = 1 \\ y = u^2. \end{cases}$$

Compare to the standard form of state space representation

$$\begin{cases} \dot{z} = f(z) + g(z)u, & z(0) = z_0 \\ y = h(z), \end{cases}$$

one obtains $f = 0$, $g = 1$, $h = z^2$ and $z_0 = 1$. The generating series c of the open-loop system can be obtained by recursive computation of the Lie derivatives from the state space realization by $(c,\eta) = L_{g_\eta}h(z(0))$ for $\eta \in X^*$. In this specific case, the support of c is $\{\emptyset, x_1, x_1^2\}$, and the corresponding coefficients are

$$\begin{aligned} \eta &= \emptyset, & L_{g_\emptyset}h(z(0)) &= z(0) = 1 \\ \eta &= x_1, & L_{g_{x_1}}h(z(0)) &= 2z(0) = 2 \\ \eta &= x_1^2, & L_{g_{x_1^2}}h(z(0)) &= L_g L_g h(z(0)) = 2. \end{aligned}$$

Therefore $c = 1 + 2x_1 + 2x_1^2 = (1+x_1) \sqcup\!\sqcup (1+x_1)$. It can be verified by induction that the generating series for the self-excited loop is

$$e = \lim_{k\to\infty} c^{\circ k} \circ 0 = \sum_{i\geq 0} (i+1)! x_0^i.$$

Therefore, the output

$$y(t) = F_e[u](t) = \sum_{i\geq 0} (i+1)! \frac{t^i}{i!} = \frac{1}{(1-t)^2}.$$

□

An obvious question is whether $c@d$ is always locally convergent, or at least input-output locally convergent, when both c and d are locally convergent. The local convergence of c and d guarantees that the feedback system in Figure 1.1(d) is at least *well-posed* in the sense described in [1,65] since F_c and F_d are well-defined causal analytic operators. That is, there is a sufficiently small $T > 0$ and $R > 0$ such that for any $u \in B_p^m(R)[t_0, t_0+T]$, there exists a $y \in B_q^m(R)[t_0, t_0+T]$ which satisfies the feedback equation (1.10). But whether $y = F_{c@d}[u]$ on some ball of input functions of nonzero radius over a nonzero interval of time is not immediate. The following example shows that $\mathbb{R}^m_{LC}\langle\langle X\rangle\rangle$ is not a closed subset of $\mathbb{R}^m\langle\langle X\rangle\rangle$ in the ultrametric topology.

Example 3.4.7. [24] Let $X = \{x_0, x_1\}$ and consider the sequence of polynomials in $\mathbb{R}^m_{LC}\langle\langle X\rangle\rangle$:

$$e_i = x_1 + 2^2 2!\, x_1^2 + 3^3 3!\, x_1^3 + \cdots + i^i i!\, x_1^i, \quad i \geq 1.$$

Clearly,

$$e := \lim_{i\to\infty} e_i = \sum_{k>0} k^k k!\, x_1^k$$

is not locally convergent. □

The central issue is whether such an example can be produced by repeated compositions of a locally convergent series. It will be first shown that the answer to this question is *no*. Then the more general case described by equation (3.13) is examined. This leads to the main conclusion that the feedback product of two locally convergent series is always input-output locally convergent.

Observe first that if $e = c \circ e$ then it follows using the definition of the composition product that e must have the form $e = \sum_{n\geq 0}(e, x_0^n)\, x_0^n$. Furthermore, since e appears on both sides of the expression $e = c \circ e$, it is possible to express by repeated substitution each coefficient (e, x_0^n) in terms of the coefficients $\{(c,\nu) : |\nu| \leq n\}$. For example, if $X = \{x_0, x_1\}$, the first few coefficients of e are:

$$(e,\emptyset) = (c,\emptyset)$$

$$(e,x_0) = (c,x_0) + (c,\emptyset)(c,x_1)$$

$$\begin{aligned}(e,x_0^2) = {} & (c,x_0^2) + (c,x_0)(c,x_1) + (c,\emptyset)(c,x_1)^2 + (c,\emptyset)(c,x_0x_1) + (c,\emptyset)(c,x_1x_0) \\ & + (c,\emptyset)^2(c,x_1^2)\end{aligned}$$

$$\begin{aligned}(e,x_0^3) = {} & (c,x_0^2)(c,x_1) + (c,x_0)(c,x_1)^2 + (c,\emptyset)(c,x_1^3) + (c,\emptyset)(c,x_1)(c,x_0x_1) + \\ & (c,\emptyset)(c,x_1)(c,x_1x_0) + (c,\emptyset)^2(c,x_1)(c,x_1^2) + (c,x_0)(c,x_0x_1) + \\ & (c,\emptyset)(c,x_1)(c,x_0x_1) + 2(c,x_0)(c,x_1x_0) + 2(c,\emptyset)(c,x_1)(c,x_1x_0) + \\ & 3(c,\emptyset)(c,x_0)(c,x_1^2) + 3(c,\emptyset)^2(c,x_1)(c,x_1^2) + (c,x_0^3) + (c,\emptyset)(c,x_0^2x_1) +\end{aligned}$$

$$(c,\emptyset)(c,x_0x_1x_0)+(c,\emptyset)^2(c,x_0x_1^2)+(c,\emptyset)(c,x_1x_0^2)+(c,\emptyset)^2(c,x_1x_0x_1)+$$
$$(c,\emptyset)^2(c,x_1^2x_0)+(c,\emptyset)^3(c,x_1^3)$$
$$\vdots$$

If c is locally convergent with growth constants K_c, M_c then

$$|(e,\emptyset)| \le K_c$$
$$|(e,x_0)| \le K_c(K_c+1)M_c$$
$$|(e,x_0^2)| \le K_c\left(\frac{3}{2}K_c^2+\frac{5}{2}K_c+1\right)M_c^2\,2!$$
$$|(e,x_0^3)| \le K_c\left(\frac{7}{3}K_c^3+\frac{35}{6}K_c^2+\frac{16}{3}K_c+1\right)M_c^3\,3!$$
$$\vdots$$

This suggests that a variation of the inequality (3.3) is possible, namely that

$$|(e,x_0^n)| \le K_c\,\tilde{\psi}_n(K_c)\,M_c^n\,n!,\ \ \forall\, n\ge 0,$$

where each $\tilde{\psi}_n(K_c)$ is a polynomial in K_c of degree n. The following lemma establishes the claim using a family of polynomials of the form

$$\tilde{\psi}_n(K_c)=\sum_{i,j=0}^{n}\sum_{\substack{\eta_j\in X^i\\ i\ge j}} K_c^j\tilde{S}_{\eta_j}(K_c,n)|\eta_j|!,\ \ n\ge 0. \tag{3.16}$$

Given a fixed n, any word η_j with $|\eta_j|\le n$ in the innermost summation has a corresponding set of right factors $\{\eta_0,\eta_1,\ldots,\eta_j\}$. Each function $\tilde{S}_{\eta_j}(K_c,n)$ is a polynomial in K_c. When $j>0$, $\tilde{S}_{\eta_j}(K_c,n)$ is computed iteratively using its right factors and the previously computed polynomials $\{\tilde{\psi}_0(K_c),\tilde{\psi}_1(K_c),\ldots,\tilde{\psi}_{n-1}(K_c)\}$:

$$\tilde{S}_{\eta_0}(K_c,n) = \frac{1}{|\eta_0|!}, \quad n \geq |\eta_0| \geq 0$$

$$\tilde{S}_{\eta_1}(K_c,n) = \frac{1}{(n)_{n_1+1}}\, \tilde{\psi}_{n-|\eta_1|}(K_c)\, \tilde{S}_{\eta_0}(K_c,n), \quad n \geq |\eta_1| \geq 1$$

$$\tilde{S}_{\eta_2}(K_c,n) = \frac{1}{(n)_{n_2+1}} \sum_{i=0}^{n-|\eta_2|} \tilde{\psi}_i(K_c)\, \tilde{S}_{\eta_1}(K_c, n-(n_2+1)-i), \quad n \geq |\eta_2| \geq 2$$

$$\vdots$$

$$\tilde{S}_{\eta_j}(K_c,n) = \frac{1}{(n)_{n_j+1}} \sum_{i=0}^{n-|\eta_j|} \tilde{\psi}_i(K_c)\, \tilde{S}_{\eta_{j-1}}(K_c, n-(n_j+1)-i), \quad n \geq |\eta_j| \geq j, \quad 2 \leq j \leq n.$$

It is easily verified that

$$\deg(\tilde{S}_{\eta_j}(K_c,n)) = \begin{cases} 0 & : \ j = 0 \\ n - |\eta_j| & : \ j > 0, \end{cases}$$

so that as expected using equation (3.16)

$$\deg(\tilde{\psi}_n(K_c)) = \max_{0<j\leq i\leq n} j + (n-i) = n.$$

See Table 3.4 for the case where $m = 1$.

Table 3.4: The first few polynomials $\tilde{S}_{\eta_j}(K_c,n)$ and $\tilde{\psi}_n(K_c)$ when $m = 1$.

n	η_j	$\tilde{S}_{\eta_0}(K_c,n),\ldots,\tilde{S}_{\eta_j}(K_c,n)$	$\tilde{\psi}_n(K_c)$
0	$\emptyset$	$\tilde{S}_{\emptyset}(K_c,0) = 1$	1
1	x_0	$\tilde{S}_{x_0}(K_c,1) = 1$	$K_c + 2$
	x_1	$\tilde{S}_{\emptyset}(K_c,1) = 1$, $\tilde{S}_{x_1}(K_c,1) = 1$	
2	x_0^2	$\tilde{S}_{x_0^2}(K_c,2) = \frac{1}{2}$	$\frac{3}{2}K_c^2 + 3K_c + 3$
	x_0x_1	$\tilde{S}_{\emptyset}(K_c,2) = 1$, $\tilde{S}_{x_0x_1}(K_c,2) = \frac{1}{2}$	
	x_1x_0	$\tilde{S}_{x_0}(K_c,2) = 1$, $\tilde{S}_{x_1x_0}(K_c,2) = \frac{1}{2}$	
	x_1^2	$\tilde{S}_{\emptyset}(K_c,2) = 1$, $\tilde{S}_{x_1}(K_c,2) = \frac{1}{2}K_c + 1$, $\tilde{S}_{x_1^2}(K_c,2) = \frac{1}{2}$	

Lemma 3.4.3. *[24] Let* $c \in \mathbb{R}^m_{LC}\langle\langle X\rangle\rangle$ *with the growth constants* K_c, M_c*, and* $e \in \mathbb{R}^m\langle\langle X\rangle\rangle$ *such that* $e = c \circ e$*. Then*

$$|(e, x_0^n)| \leq K_c\, \tilde{\psi}_n(K_c)\, M_c^n\, n!, \ \ \forall n \geq 0. \tag{3.17}$$

Proof: The proof has some elements in common with that of Lemma 3.3.2, except here it is not assumed a priori that e is locally convergent. The basic approach employs *nested inductions*. The outer induction is on n. It is clear from the discussion above that the claim holds when $n = 0$, $n = 1$ and $n = 2$ for $m = 1$. A similar calculation can be done for arbitrary $m \geq 1$. Now suppose equation (3.17) holds up to some fixed $n - 1 \geq 1$. Given any η_j, where $|\eta_j| \leq n$, it will first be shown by induction on j (the inner induction) that

$$|(\eta_j \circ e, x_0^n)| \leq K_c^j M_c^{-|\eta_j|} M_c^n\, n!\, \tilde{S}_{\eta_j}(K_c, n), \ \ 0 \leq j \leq n. \tag{3.18}$$

The $j = 0$ case is straightforward. Suppose $j = 1$. Then $0 \leq n - |\eta_1| \leq n - 1$ and

$$\begin{aligned}
|(\eta_1 \circ e, x_0^n)| &= \left|\left(x_0^{n_1+1}(e_{i_1} \sqcup\!\sqcup\, x_0^{n_0}), x_0^n\right)\right| \\
&= \left|\left(e_{i_1} \sqcup\!\sqcup\, x_0^{n_0}, x_0^{n-(n_1+1)}\right)\right| \\
&= \left|\left(e_{i_1}, x_0^{n-|\eta_1|}\right)\left(x_0^{n-|\eta_1|} \sqcup\!\sqcup\, x_0^{n_0}, x_0^{n-(n_1+1)}\right)\right| \\
&\leq \left(K_c\, \tilde{\psi}_{n-|\eta_1|}(K_c)\, M_c^{n-|\eta_1|}(n - |\eta_1|)!\right)\binom{n-(n_1+1)}{n-|\eta_1|} \\
&= K_c M_c^{-|\eta_1|} M_c^n\, n!\, \tilde{S}_{\eta_1}(K_c, n).
\end{aligned}$$

Now assume the inequality (3.18) holds up to some fixed j, where $1 \leq j \leq n - 1$. Then $0 \leq n - |\eta_{j+1}| \leq n - (j + 1)$ and

$$\begin{aligned}
|(\eta_{j+1} \circ e, x_0^n)| &= \left|\left(e_{i_{j+1}} \sqcup\!\sqcup\, (\eta_j \circ e), x_0^{n-(n_{j+1}+1)}\right)\right| \\
&= \left|\sum_{i=0}^{n-(n_{j+1}+1)} \left(e_{i_{j+1}}, x_0^i\right)\left(\eta_j \circ e, x_0^{n-(n_{j+1}+1)-i}\right)\binom{n-(n_{j+1}+1)}{n-(n_{j+1}+1)-i}\right|.
\end{aligned}$$

Since $\left(\eta_j \circ e, x_0^{n-(n_{j+1}+1)-i}\right) = 0$ when $n-(n_{j+1}+1)-i < |\eta_j|$ or equivalently $i > n-|\eta_{j+1}|$, it follows using the coefficient bound (3.17) for e (because $0 \le i \le n-1$) and the bound (3.18) for $\eta_j \circ e$ that

$$\begin{aligned}
|(\eta_{j+1} \circ e, \nu)| &\le \sum_{i=0}^{n-|\eta_{j+1}|} \left(K_c \tilde{\psi}_i(K_c) M_c^i\, i!\right) \left(K_c^j M_c^{-|\eta_j|} M_c^{n-(n_{j+1}+1)-i}\, (n-(n_{j+1}+1)-i)! \cdot \right.\\
&\quad \left. \tilde{S}_{\eta_j}(K_c, n-(n_{j+1}+1)-i)\right) \binom{n-(n_{j+1}+1)}{n-(n_{j+1}+1)-i}\\
&= K_c^{j+1} M_c^{-|\eta_{j+1}|} M_c^n\, n!\, \frac{1}{(n)_{n_{j+1}+1}} \sum_{i=0}^{n-|\eta_{j+1}|} \tilde{\psi}_i(K_c)\, \tilde{S}_{\eta_j}(K_c, n-(n_{j+1}+1)-i)\\
&= K_c^{j+1} M_c^{-|\eta_{j+1}|} M_c^n\, n!\, \tilde{S}_{\eta_{j+1}}(K_c, n).
\end{aligned}$$

Hence, the claim is true for all $0 \le j \le n$.

To complete the outer level induction, observe that

$$\begin{aligned}
|(e, x_0^n)| &= |(c \circ e, x_0^n)|\\
&= \left| \sum_{i,j=0}^{n} \sum_{\substack{\eta_j \in X^i\\ i \ge j}} (c, \eta_j)(\eta_j \circ e, x_0^n) \right|\\
&\le \sum_{i,j=0}^{n} \sum_{\substack{\eta_j \in X^i\\ i \ge j}} \left(K_c M_c^{|\eta_j|} |\eta_j|!\right) \left(K_c^j M_c^{-|\eta_j|} M_c^n\, n!\, \tilde{S}_{\eta_j}(K_c, n)\right)\\
&\le K_c\, \tilde{\psi}_n(K_c)\, M_c^n\, n!.
\end{aligned}$$

Therefore, the inequality (3.17) holds for all $n \ge 0$. ■

This result now makes the following theorem concerning the local convergence of e possible.

Theorem 3.4.8. *[24] If $c \in \mathbb{R}^m_{LC}\langle\langle X \rangle\rangle$ with growth constants K_c, M_c and $e = c \circ e$ then $e \in \mathbb{R}^m_{LC}\langle\langle X \rangle\rangle$. Specifically, when $K_c \gg 1$ then*

$$|(e, x_0^n)| \le K_c (2mK_cM_c)^n\, n!, \quad \forall\, n \ge 0.$$

Proof: To prove the local convergence of e, note there is no loss of generality in assuming that $K_c \gg 1$. The claim then follows from Lemma 3.4.3 by showing that when $K_c \gg 1$:

$$\tilde{\psi}_n(K_c) \leq (2mK_c)^n, \ \ \forall n \geq 0.$$

The $n = 0$ case is trivial. When $n \geq 1$, $\tilde{\psi}_n(K_c)$ can be approximated by its highest order term, specifically, its degree n term. This corresponds to those terms in equation (3.16) where $1 \leq i = j \leq n$:

$$\begin{aligned}
\tilde{\psi}_n(K_c) &\approx \sum_{j=1}^{n} K_c^j \sum_{\eta_j \in X^j} \tilde{S}_{\eta_j}(K_c, n)\, j! \\
&= \sum_{j=1}^{n} (mK_c)^j\, \tilde{S}_{x_1^j}(K_c, n)\, j! \\
&\approx \left(\sum_{j=1}^{n} m^j \gamma_{j,n}\, j! \right) K_c^n,
\end{aligned}$$

where each $\tilde{S}_{x_1^j}(K_c, n)$ has been approximated by its highest order term $\gamma_{j,n} K_c^{n-j}$. It is easily verified using the definition of $\tilde{S}_{x_1^j}(K_c, n)$ that the coefficients $\gamma_{j,n}$ can be computed successively by

$$\begin{aligned}
\gamma_{1,n+1} &= \frac{1}{n+1} \left(\sum_{j=1}^{n} m^j \gamma_{j,n}\, j! \right) \\
\gamma_{j,n+1} &= \frac{1}{n+1} \left(\gamma_{j-1,n} + \sum_{i=1}^{n+1-j} \left(\sum_{k=1}^{i} m^k \gamma_{k,i}\, k! \right) \gamma_{j-1,n-i} \right), \quad 2 \leq j \leq n \\
\gamma_{n+1,n+1} &= \frac{1}{n+1}\, \gamma_{n,n},
\end{aligned}$$

with $\gamma_{1,1} = 1$. In which case,

$$\tilde{\psi}_n(K_c) \approx (n+1)\gamma_{1,n+1}\, K_c^n, \ n \geq 0.$$

Another inductive argument shows that $(n+1)\gamma_{1,n+1} \leq (2m)^n$, $n \geq 0$. (See also Table 3.5 and Figure 3.4.) Thus, the theorem is proven. ■

Table 3.5: Some leading coefficients of $\tilde{\psi}_n(K_c)$.

m	$(n+1)\gamma_{1,n+1}$, $n \geq 0$
1	1,1,1.5,2.5,4.375,7.875,14.4375,26.8125,50.2734,94.9609,...
2	1,2,6,20,70,252,924,3432,12870,48620,184756,705432,...
3	1,3,13.5,67.5,354.375,1913.625,10524.9375,58638.9378,...

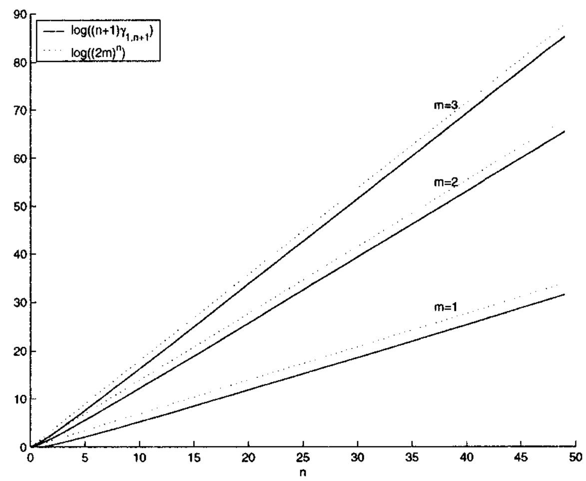


Figure 3.4: The growth of $\log((n+1)\gamma_{1,n+1})$ and $\log((2m)^n)$ for $m = 1, 2, 3$.

The final step of the analysis is to use Theorem 3.4.8 to address the input-output local convergence of the feedback product.

Theorem 3.4.9. *[24] If* $c, d \in \mathbb{R}^m_{LC}\langle\langle X\rangle\rangle$ *then* $c@d$ *is input-output locally convergent. Specifically, when* $K_c, K_d \gg 1$ *then*

$$((c@d) \circ b, x_0^n) \leq K_c((2m)^2 K_c(K_b + K_d)M)^n \, n!$$

for any $b \in \mathbb{R}^m_{LC}\langle\langle X_0\rangle\rangle$ *and where* $M = \max\{M_b, M_c, M_d\}$.

Proof: Select any series $b \in \mathbb{R}^m_{LC}\langle\langle X_0 \rangle\rangle$. If follows from equation (3.14) that

$$\begin{aligned}(c@d) \circ b &= (c \,\tilde{\circ}\, (d \circ (c@d))) \circ b \\ &= c \circ (b + d) \circ ((c@d) \circ b).\end{aligned}$$

Since b, c and d are all locally convergent, Theorem 3.4.8 implies that $(c@d) \circ b$ is always locally convergent, and therefore $c@d$ must be input-output locally convergent. To produce the given growth condition for the output series, first replace c in Theorem 3.4.8 with $c \circ (b + d)$ and note that $K_{c\circ(b+d)} = K_c$. The assumption that $K_c \gg 1$ ensures that the growth estimate in this theorem applies. Next, since $K_d \gg 1$, Theorem 3.3.2 provides that

$$M_{c\circ(b+d)} = 2m(K_b + K_d) \max\{M_b, M_c, M_d\},$$

using the fact that $|\nu| + 1 \leq 2^{|\nu|}$ for all $\nu \geq 0$. This produces the desired result. ■

Example 3.4.8. [24] For any $c, d \in \mathbb{R}^m_{LC}\langle\langle X \rangle\rangle$, a self-excited feedback loop can be described by $F_{c@d}[0] = F_{(c@d)\circ 0}[u] = F_{(c@d)_0}[u]$ (c.f. Lemma 2.3.2, property *2.*). In this case $(c@d)_0 = \lim_{i\to\infty} e_i$, where $e_{i+1} = (c \circ d) \circ e_i$. Using the $m = 0$ version of equation (3.7) (since the closed-loop system has in effect no external input) and Theorem 3.4.8, $F_{c@d}[u]$ will converge at least on the interval $[0, T_{\max})$, where

$$T_{\max} = \frac{1}{M_{(c@d)_0}} = \frac{1}{K_{c\circ d} M_{c\circ d}}. \tag{3.19}$$

For example, when $c\circ d = 1 + x_1$ it is easy verified that $(c@d)_0 = \sum_{k\geq 0} x_0^k$ so that $F_{c@d}[0](t) = e^t$ for $t \geq 0$. In this case, $T_{\max} = 1$ is very conservative. When $c \circ d = 1 + 2x_1 + 2x_1^2$ it follows that $(c@d)_0 = \sum_{k\geq 0} (k+1)!\, x_0^k$ and $F_{c@d}[0](t) = 1/(1-t)^2$ for $0 \leq t < 1$. Here $T_{\max} = 0.5$ is less conservative. □

Example 3.4.9. [24] Reconsider the state space systems in Example 3.4.2. The operator $F_{c@d}[u]$ then has the analytic state space realization:

$$f(z) = \begin{pmatrix} K_d M_c z_c^2 z_d \\ K_c M_d z_c z_d^2 \end{pmatrix}, \quad g(z) = \begin{pmatrix} M_c z_c^2 \\ 0 \end{pmatrix}, \quad h(z) = K_c z_c$$

near $z(0) = [1 \;\; 1]^T$. The first few coefficients of $c@d$ are given in Table 3.6.

Table 3.6: Some coefficients (c, ν), (d, ν) and $(c@d, \nu)$ in Example 3.4.9.

ν	(c,ν)	(d,ν)	$(c@d,\nu)$
$\emptyset$	K_c	K_d	K_c
x_0	0	0	$K_c K_d M_c$
x_1	$K_c M_c$	$K_d M_d$	$K_c M_c$
x_0^2	0	0	$K_c((K_d M_c)^2\, 2! + K_c K_d M_c M_d)$
$x_0 x_1$	0	0	$K_c K_d M_c^2\, 2!$
$x_1 x_0$	0	0	$K_c K_d M_c^2\, 2!$
x_1^2	$K_c M_c^2\, 2!$	$K_d M_d^2\, 2!$	$K_c M_c^2\, 2!$
x_0^3	0	0	$K_c((K_d M_c)^3\, 3! + K_c (K_d M_c)^2 M_d\, 7 + K_c^2 K_d M_c M_d^2\, 2!)$
$x_0^2 x_1$	0	0	$K_c((K_d M_c)^2 M_c\, 3! + K_c K_d M_c^2 M_d\, 3)$
$x_0 x_1 x_0$	0	0	$K_c((K_d M_c)^2 M_c\, 3! + K_c K_d M_c^2 M_d\, 2)$
$x_0 x_1^2$	0	0	$K_c K_d M_c^3\, 3!$
$x_1 x_0^2$	0	0	$K_c((K_d M_c)^2 M_c\, 3! + K_c K_d M_c^2 M_d)$
$x_1 x_0 x_1$	0	0	$K_c K_d M_c^3\, 3!$
$x_1^2 x_0$	0	0	$K_c K_d M_c^3\, 3!$
x_1^3	$K_c M_c^3\, 3!$	$K_d M_d^3\, 3!$	$K_c M_c^3\, 3!$

Since $c@d$ is a non-negative series in this case, local convergent and input-output local convergence are equivalent. Setting $u(t) = \bar{u} = 1$ is equivalent to letting $b = 1$ in Theorem 3.4.9. Therefore, using equation (3.7) (again with m=0) and the growth condition from

Theorem 3.4.9, a lower bound on the finite escape time for this system is

$$T_{\max} = \frac{1}{M_{(c@d)\circ 1}} = \frac{1}{4K_c(K_d+1)M}.$$

Four specific cases of $T_{\max}$ are given in Table 3.7 and compared against the numerically determined escape times shown in Figure 3.5. The conservativeness in this estimate has in some sense accumulated when compared to the cascade connection in Example 3.4.2. □

Table 3.7: $T_{\max}$ and t_{esc} for specific examples of $(c@d) \circ 1$.

Case	K_c	M_c	K_d	M_d	$M_{(c@d)\circ 1}$	$T_{\max}$	t_{esc}	$t_{esc}/T_{\max}$
1	10	5	5	5	1200	0.000833	0.01281	15.37
2	5	10	5	5	1200	0.000833	0.01205	14.46
3	5	5	10	5	1100	0.000909	0.01266	13.93
4	5	5	5	10	1200	0.000833	0.01281	15.37

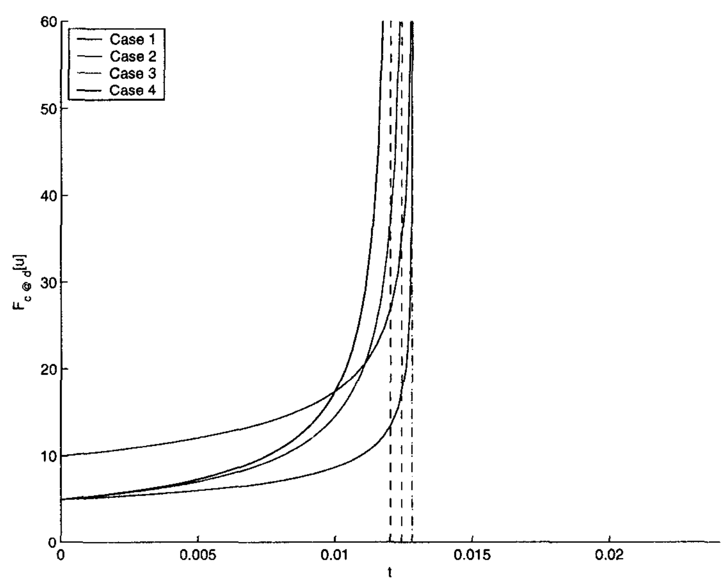


Figure 3.5: The output of $F_{c@d}[u]$ when $u(t) = \bar{u} = 1$ for Cases 1-4.

CHAPTER IV

FORMAL LAPLACE-BOREL TRANSFORM

4.1 Introduction

In this chapter, the formal Laplace-Borel transform pair for a Fliess operator is defined and related to existing notions of the classical Laplace-Borel transform. Then, using the composition product, the formal Laplace-Borel transform is applied to characterize the dynamics of input-output systems and the cascade interconnection of analytic nonlinear systems. An explicit relationship is derived between the formal Laplace-Borel transforms of the input and output signals of a Fliess operator. This result provides an efficient alternative interpretation of the symbolic calculus introduced by Fliess to compute the output response of nonlinear analytic systems [23]. Finally, the formal Laplace-Borel transform is shown to be an isomorphic mapping between various algebraic structures on the set of all convergent Fliess operators and the set of all locally convergent formal power series under the addition, the shuffle and the composition products.

The chapter is organized as follows. In Section 2, the notion of a formal Laplace-Borel transform of a Fliess operator is defined. Then its basic properties are explored and illustrated by a set of examples. In the section that follows, the relationship between the composition product and the formal Laplace-Borel transform is developed, and the idea is applied to the cascade interconnection of systems. In Section 4, combined with the composition product, the formal Laplace-Borel transform is applied to characterize the input-output dynamics of analytic systems. This theory provides a compact interpretation of the symbolic calculus proposed by Fliess [23]. In the last section, the formal Laplace-Borel transform is presented as an isomorphic mapping between the sets $\mathbb{R}_{LC}\ll X\gg$ and $\mathcal{F}$, which provides a generalization of the time domain and frequency domain duality in linear system analysis.

4.2 Definition and Properties of the Formal Laplace-Borel Transform

In this section, the definition of the formal Laplace-Borel transform of a Fliess operator is presented, and some basic properties are characterized. Many of the properties of the formal Laplace-Borel transform in the present nonlinear context have counterparts in the classical Laplace-Borel transform widely used in linear analysis. In linear time-invariant system analysis, a causal homogeneous input-output mapping is expressed in terms of a convolution of the system impulse response $h(t)$ with the input signal

$$y(t) = \int_{t_0}^{t} h(t-\tau)u(\tau)d\tau.$$

This mapping can also be uniquely characterized by the Laurent series of its system transfer function $H(s) = \mathscr{L}\{h(t)\} = \sum_{k>0} h_k s^{-k}$. Given an input-output Fliess operator defined on a set of admissible inputs, the following lemma ensures that its associated generating series is unique.

Lemma 4.2.1. *[66, Corollary 2.2.4] Suppose c and d are both locally convergent power series. Let $\boldsymbol{u}_T$ be the set of essentially bounded measurable functions $u : [0,T] \to \mathbb{R}^m$. If $F_c = F_d$ on some $\nu_T = \{u \in \boldsymbol{u}_T : \|u\|_\infty < 1\}$, $T > 0$, then $c = d$.*

So throughout this chapter it is always assumed that the set of admissible inputs are at least within the set ν_T, therefore, the following definition is well-posed.

Definition 4.2.1. *[42] Let $X = \{x_0, x_1, \ldots, x_m\}$. The **formal Laplace transform on** $\mathscr{F}$ is defined as*

$$\begin{aligned}\mathscr{L}_f &: \mathscr{F} \to \mathbb{R}^{\ell}_{LC}\langle\langle X \rangle\rangle \\ &: F_c \mapsto c.\end{aligned}$$

*The corresponding **formal Borel transform** is*

$$\begin{aligned}\mathscr{B}_f &: \mathbb{R}^{\ell}_{LC}\langle\langle X \rangle\rangle \to \mathscr{F} \\ &: c \mapsto F_c.\end{aligned}$$

To present the basic properties of the formal Laplace-Borel transform, a generalized series, the Dirac series is artificially introduced in the following definition.

Definition 4.2.2. *A **Dirac series**, δ, is a generalized series with the defining property that $F_\delta[u] = u$ for all bounded measurable inputs $u \in \nu_T$. Each component of the Dirac series, δ_i has the property that $F_{\delta_i}[u] = u_i$ for any $1 \leq i \leq m$.*

It is easy to see that the Dirac series, δ, is the identity element for the composition product, that is $c \circ \delta = \delta \circ c = c$. Each component δ_i has the property that $(\delta_i \circ c, \eta) = (c, \eta)_i$ for any $1 \leq i \leq m$, where $(c, \eta)_i$ is the i^{th} component of $(c, \eta) \in \mathbb{R}^m$. Similar to the generalized Dirac function, the Dirac series has the property that $x_0(\delta_i \sqcup\!\sqcup\, c) = x_i c$. Some of the basic properties of Laplace-Borel transform are stated in the following theorem.

Theorem 4.2.1. *[42] Let $X = \{x_0, x_1, \ldots, x_m\}$. Given any $c, d \in \mathbb{R}^m_{LC}\langle\langle X \rangle\rangle$ and scalars $\alpha, \beta \in \mathbb{R}$, the following identities hold:*

1. *Linearity*

$$\mathscr{L}_f\left[\alpha F_c + \beta F_d\right] = \alpha c + \beta d$$

$$\mathscr{B}_f\left[\alpha c + \beta d\right] = \alpha F_c + \beta F_d$$

2. Multiplication

$$\mathscr{L}_f\left[F_c \cdot F_d\right] = c \sqcup\!\sqcup\, d$$

$$\mathscr{B}_f\left[c \sqcup\!\sqcup\, d\right] = F_c \cdot F_d$$

3. Scalability of input

$$\mathscr{L}_f\left[F_c[\alpha(\cdot)]\right] = \sum_{k\geq 0} \alpha^k c_k$$

$$\mathscr{B}_f\left[\sum_{k\geq 0} \alpha^k c_k\right] = F_c[\alpha(\cdot)],$$

where $c = c_0 + c_1 + c_2 + \cdots$ *as in Definition 2.3.16.*

4. Integration

$$\mathscr{L}_f\left[I^n F_c\right] = x_0^n c$$

$$\mathscr{B}_f\left[x_0^n c\right] = I^n F_c$$

where $I^n(\cdot)$ *is the* n^{th} *integration operator and*

$$I^n F_c[u] := \int_0^t \int_0^{\tau_1} \cdots \int_0^{\tau_{n-1}} F_c[u](\tau_n)\, d\tau_n \cdots d\tau_2\, d\tau_1.$$

5. Differentiation

$$\mathscr{L}_f\left[DF_c\right] = x_0^{-1}(c) + \sum_{i=1}^{m} \delta_i \sqcup\!\sqcup\, (x_i^{-1}(c))$$

$$\mathscr{B}_f\left[x_0^{-1}(c) + \sum_{i=1}^{m} \delta_i \sqcup\!\sqcup\, (x_i^{-1}(c))\right] = DF_c$$

If x_0^n *is a left factor of* c *then*

$$\mathscr{L}_f\left[D^n F_c\right] = x_0^{-n}(c)$$

$$\mathscr{B}_f\left[x_0^{-n}(c)\right] = D^n F_c$$

where $D(\cdot)$ is the differentiation operator with $DF_c[u] := \frac{d}{dt}F_c[u]$, while $D^n(\cdot)$ is the n^{th} differentiation operator and $D^nF_c[u] := \frac{d^n}{dt^n}F_c[u]$.

6. *Concatenation of inputs*

$$\mathscr{L}_f\left[F_c\left[\#^2(\cdot)\right](2(\cdot))\right] = \sum_{\eta\in X^*}(c,\eta)\sum_{\rho\nu=\eta}\rho \sqcup\!\sqcup \nu$$

$$\mathscr{B}_f\left[\sum_{\eta\in X^*}(c,\eta)\sum_{\rho\nu=\eta}\rho \sqcup\!\sqcup \nu\right] = \left[F_c\left[\#^2(\cdot)\right](2(\cdot))\right],$$

where $\#^2(\cdot)$ is the concatenation of two copies of control signal of the following form

$$\left(\#^2(u)\right)(\tau) = (u\#u)(\tau) = \begin{cases} u(\tau) & \text{if } 0 \leq \tau \leq t \\ u(\tau - t) & \text{if } t < \tau \leq 2t. \end{cases}$$

The result can be generalized to the concatenated control signal consisting of n repeated inputs as follows

$$\mathscr{L}_f\left[F_c\left[\#^n(\cdot)\right](n(\cdot))\right] = \sum_{\eta\in X^*}(c,\eta)\sum_{\nu_1\nu_2\cdots\nu_n=\eta}\nu_1 \sqcup\!\sqcup \nu_2 \sqcup\!\sqcup \cdots \sqcup\!\sqcup \nu_n$$

$$\mathscr{B}_f\left[\sum_{\eta\in X^*}(c,\eta)\sum_{\nu_1\nu_2\cdots\nu_n=\eta}\nu_1 \sqcup\!\sqcup \nu_2 \sqcup\!\sqcup \cdots \sqcup\!\sqcup \nu_n\right] = F_c\left[\#^n(\cdot)\right](n(\cdot)).$$

Proof: The three properties of linearity, integration and scalability of inputs are straightforward. The multiplication property follows from results in the literature [17, 44, 66]. The properties that need to be justified are the differentiation property and the concatenation of inputs.

5. *Proof of the differentiation property*

It was shown in [66] that the first derivative of a Fliess operator is

$$\frac{d}{dt}F_c[u](t) = F_{x_0^{-1}(c)}[u](t) + \sum_{i=1}^{m} u_i F_{x_i^{-1}(c)}[u](t).$$

Applying the formal Laplace-Borel transform to this equality gives the first pair of equations. Now if x_0 is a left factor of c, then $F_{x_i^{-1}(c)}[u](t) = 0$ for $i = 1, 2, \cdots, m$. In this case $\frac{d}{dt}F_c[u](t) = F_{x_0^{-1}(c)}[u](t)$. Proceeding inductively, the second pair of equations follows.

6. Proof of the concatenation of inputs

It was proven in [66] that for the concatenated input signal

$$(u\#v)\,(\tau) = \begin{cases} u(\tau) & \text{if } 0 \le \tau \le t \\ v(\tau - t) & \text{if } t < \tau \le T, \end{cases}$$

the Fliess operator

$$F_c\,[u\#v]\,(\tau + t) = \sum_{\eta \in X^*} (c, \eta) \sum_{\rho\nu = \eta} E_\rho[v](\tau) E_\nu[u](t). \tag{4.1}$$

In (4.1), let $u = v$ and $t = \tau$, then

$$F_c\,[u\#u]\,(2t) = F_c\left[\#^2(u)\right](2t) = \sum_{\eta \in X^*} (c, \eta) \sum_{\rho\nu = \eta} E_\rho[u](t) E_\nu[u](t).$$

Next apply the formal Laplace-Borel transform, the identity is proven. The more general case of n concatenated inputs can be derived by induction. ■

Example 4.2.1. [42] Let $X = \{x_0, x_1, x_2\}$ and $F_c[u](t) = \exp\left[\int_0^t u_1(t) + u_2(t) dt\right]$. Observe that F_c can be expanded as

$$\begin{aligned} F_c[u](t) &= \sum_{n \ge 0} \frac{1}{n!} \left(\int_0^t u_1(t) + u_2(t)\, dt \right)^n \\ &= \sum_{n \ge 0} \int_0^t [(u_1(\tau_1) + u_2(\tau_1))] \int_0^{\tau_1} [u_1(\tau_2) + u_2(\tau_2)] \cdots \\ &\qquad \int_0^{\tau_{n-1}} [u_1(\tau_n) + u_2(\tau_n)]\; d\tau_n \cdots d\tau_2\, d\tau_1. \end{aligned}$$

Therefore,

$$\mathscr{L}_f[F_c] = \sum_{n \ge 0} (x_1 + x_2)^n = (x_1 + x_2)^*.$$

(See also [66, Example 2.3.9] for discussion related to this example.) □

Other formal Laplace-Borel transform pairs are given in Table 4.1.

Table 4.1: Some formal Laplace-Borel transform pairs.

F_c	$\mathscr{L}_f[F_c]$
$F_c : u \mapsto 1$	1
$F_c : u \mapsto t^n$	$n!\, x_0^n$
$F_c : u \mapsto \left(\sum_{i=0}^{n-1} \frac{\binom{i}{n-1}}{i!} a^i t^i \right) e^{at}$	$(1 - a x_0)^{-n}$
$F_c : u \mapsto \frac{1}{n!} \left(\int_{t_0}^{t} \sum_{j=1}^{k} u_{i_j}(\tau) d\tau \right)^n$	$(x_{i_1} + x_{i_2} + \cdots + x_{i_k})^n$
$F_c : u \mapsto \sum_{n \geq 0} \frac{a_n}{n!} \int_{t_0}^{t} \sum_{j=1}^{k} u_{i_j}(\tau) d\tau$	$\sum_{n \geq 0} a_n (x_{i_1} + x_{i_2} + \cdots + x_{i_k})^n$
$F_c : u \mapsto e^{\int_{t_0}^{t} \sum_{j=1}^{k} u_{i_j}(\tau) d\tau}$	$(x_{i_1} + x_{i_2} + \cdots + x_{i_k})^*$
$F_c : u \mapsto \int_{t_0}^{t} \sum_{j=1}^{k} u_{i_j}(\tau) d\tau \, e^{\int_{t_0}^{t} \sum_{j=1}^{k} u_{i_j}(\tau) d\tau}$	$\frac{x_{i_1} + x_{i_2} + \cdots + x_{i_k}}{[1 - (x_{i_1} + x_{i_2} + \cdots + x_{i_k})]^2}$
$F_c : u \mapsto \cos \left(\int_{t_0}^{t} \sum_{j=1}^{k} u_{i_j}(\tau) d\tau \right)$	$\frac{1}{1 + (x_{i_1} + x_{i_2} + \cdots + x_{i_k})^2}$
$F_c : u \mapsto \sin \left(\int_{t_0}^{t} \sum_{j=1}^{k} u_{i_j}(\tau) d\tau \right)$	$\frac{x_{i_1} + x_{i_2} + \cdots + x_{i_k}}{1 + (x_{i_1} + x_{i_2} + \cdots + x_{i_k})^2}$

Example 4.2.2. [42] Let $X = \{x_0, x_1, \ldots, x_m\}$. Suppose F_c has the generating series $c = \sum_{\eta \in X^*} \eta$, and F_ξ is given for some fixed word $\xi \in X^*$. Then

$$
\begin{aligned}
\mathscr{L}_f[F_c \cdot F_\xi] &= \mathscr{L}_f[F_c] \sqcup\!\sqcup \mathscr{L}_f[F_\xi] \\
&= c \sqcup\!\sqcup \xi \\
&= \sum_{\nu \in X^*} \binom{\nu}{\xi} \nu,
\end{aligned}
$$

where $\binom{\nu}{\xi}$ denotes the binomial coefficients over words in X^* (see [43, p. 127]). □

Example 4.2.3. From the linearity and multiplication property of the formal Laplace-Borel transform, analytic functions of Fliess operator can be put in direct correspondence with analytic functions of formal power series, specifically,

$$\mathscr{L}_f\left[\sum_{n\geq 0} a_n\{F_c\}^n\right] = \sum_{n\geq 0} a_n c^{\sqcup\!\sqcup\, n} = \sum_{n\geq 0} a_n\, n!\, c^{*n}.$$

Some examples are given below:

$$\mathscr{L}_f\left\{e^{F_c}\right\} = \sum_{n\geq 0} \frac{c^{\sqcup\!\sqcup\, n}}{n!}$$

$$\mathscr{L}_f\left\{\cos F_c\right\} = \sum_{n\geq 0} \frac{(-1)^n}{(2n)!} c^{\sqcup\!\sqcup\, 2n}$$

$$\mathscr{L}_f\left\{\sin F_c\right\} = \sum_{n\geq 0} \frac{(-1)^n}{(2n+1)!} c^{\sqcup\!\sqcup\, (2n+1)}.$$

□

4.3 The Formal Laplace-Borel Transform and the Composition Product

The composition product of two series $c \in \mathbb{R}^{\ell}\langle\langle X\rangle\rangle$ and $d \in \mathbb{R}^{m}\langle\langle X\rangle\rangle$ over an alphabet $X = \{x_0, x_1, \ldots, x_m\}$ is defined recursively in terms of the concatenation and the shuffle product. The composition product is associative, i.e., $(c \circ d) \circ e = c \circ (d \circ e)$, hence $(\mathbb{R}^m\langle\langle X\rangle\rangle, \circ)$ forms a semigroup (Without loss of generality, it is assumed $\ell = m$ in this section). It was shown in Theorem 3.3.2 that the composition of two locally convergent formal power series is always locally convergent, therefore the set $\mathbb{R}^m_{LC}\langle\langle X\rangle\rangle$ is closed under composition, and $(\mathbb{R}^m_{LC}\langle\langle X\rangle\rangle, \circ)$ forms a semigroup. Similarly, over the set of convergent

Fliess operators $\mathcal{F}$, the composition of two convergent Fliess operators is still a well-defined Fliess operator. Applying the formal Laplace transform produces the generating series for the cascaded Fliess operators, which is the composition product. Specifically, for any $c \in \mathbb{R}^{\ell}_{LC}\langle\langle X \rangle\rangle$ and $d \in \mathbb{R}^{m}_{LC}\langle\langle X \rangle\rangle$,

$$F_c \circ F_d = F_{c \circ d}. \tag{4.2}$$

In the next theorem it is shown that the formal Laplace-Borel transform provides an isomorphism between the two semigroups $(\mathbb{R}^{m}_{LC}\langle\langle X \rangle\rangle, \circ)$ and $(\mathscr{F}, \circ)$.

Theorem 4.3.1. *[42] Let* $X = \{x_0, x_1, \ldots, x_m\}$*. For any* $c \in \mathbb{R}^{\ell}_{LC}\langle\langle X \rangle\rangle$ *and* $d \in \mathbb{R}^{m}_{LC}\langle\langle X \rangle\rangle$*:*

$$\mathscr{L}_f(F_c \circ F_d) = \mathscr{L}_f(F_c) \circ \mathscr{L}_f(F_d)$$

$$\mathscr{B}_f(c \circ d) = \mathscr{B}_f(c) \circ \mathscr{B}_f(d).$$

Proof: The proof is straightforward. For any well-defined F_c and F_d,

$$\begin{aligned} \mathscr{L}_f(F_c \circ F_d) &= \mathscr{L}_f(F_{c \circ d}) = c \circ d \\ &= \mathscr{L}_f(F_c) \circ \mathscr{L}_f(F_d). \end{aligned}$$

Conversely, for any locally convergent c and d, the composition product is still locally convergent. Applying the formal Borel transform gives

$$\begin{aligned} \mathscr{B}_f(c \circ d) &= F_{c \circ d} = F_c \circ F_d \\ &= \mathscr{B}_f(c) \circ \mathscr{B}_f(d). \end{aligned}$$

■

The isomorphism between the two semigroups $(\mathcal{F}, \circ)$ and $(\mathbb{R}^{m}_{LC}\langle\langle X \rangle\rangle, \circ)$ is illustrated in Figure 4.1.

$$\begin{array}{ccc} (F_c, F_d) & \xrightarrow{\;\circ\;} & F_c \circ F_d = F_{c \circ d} \\ \mathscr{B}_f \uparrow \;\; \downarrow \mathscr{L}_f & & \mathscr{L}_f \downarrow \;\; \uparrow \mathscr{B}_f \\ (c, d) & \xrightarrow{\;\circ\;} & c \circ d \end{array}$$

Figure 4.1: The isomorphism between the semigroups $(\mathcal{F}, \circ)$ and $(\mathbb{R}^m_{LC}\langle\langle X \rangle\rangle, \circ)$.

Example 4.3.1. Consider the linear time-invariant system $y(t) = \int_0^t h(t-\tau)u(\tau)\,d\tau$, where h is analytic at $t = 0$. Then $y = F_c[u]$ with $(c, x_0^k x_1) = h^{(k)}(0)$, $k \geq 0$ and zero otherwise. Letting $u(t) = \sum_{k\geq 0}(c_u, x_0^k)\, t^k/k!$ then it follows that $y(t) = \sum_{n\geq 0}(c_y, x_0^n)\, t^n/n!$, where

$$\begin{aligned} c_y &= c \circ c_u \\ &= \sum_{k\geq 0}(c, x_0^k x_1)\, x_0^k x_1 \circ c_u \\ &= \sum_{k\geq 0}(c, x_0^k x_1)\, x_0^{k+1} c_u. \end{aligned}$$

Therefore,

$$(c_y, x_0^n) = \sum_{k=0}^{n-1}(c, x_0^k x_1)\,(c_u, x_0^{n-1-k}), \quad n \geq 1,$$

which is just the conventional convolution sum. □

Example 4.3.2. [42] Let $X = \{x_0, x_1, x_2\}$, $F_c[u](t) = \cos\left(\int_0^t u_1(t) + u_2(t)dt\right)$ and $d \in \mathbb{R}^2_{LC}\langle\langle X \rangle\rangle$. Defining

$$\begin{aligned} F_e[u](t) &= (F_c \circ F_d)[u](t) \\ &= \cos\left(\int_0^t F_{d_1}[u](t) + F_{d_2}[u](t)dt\right), \end{aligned}$$

the formal Laplace-Borel transform of F_e is then

$$\mathscr{L}_f[F_e] = c \circ d = \sum_{i\geq 0}(-1)^i (x_1 + x_2)^{2i} \circ d.$$

□

Example 4.3.3. [42] Let $X = \{x_0, x_1, \ldots, x_m\}$ and $c \in \mathbb{R}^m_{LC}\langle\langle X \rangle\rangle$. It is easily verified by induction that for $n \geq 1$,

$$x_i^n \circ c = \frac{1}{n!}(x_0 c_i)^{\sqcup\!\sqcup\, n}, \quad i = 1, 2, \cdots, m. \tag{4.3}$$

Applying the formal Laplace-Borel transform to both sides gives

$$\begin{aligned}
\mathscr{B}_f\left[x_i^n \circ c\right] &= \mathscr{B}_f\left[\frac{1}{n!}(x_0 c_i)^{\sqcup\!\sqcup\, n}\right] \\
&= \frac{1}{n!}\left[\mathscr{B}_f[x_0 c_i]\right]^n \\
&= \frac{1}{n!}\left[\int_0^t F_{c_i}[u](\tau)\, d\tau\right]^n.
\end{aligned}$$

□

Example 4.3.4. [42] First consider the linear ordinary differential equation

$$\frac{d^n y(t)}{dt^n} + \sum_{i=0}^{n-1} a_i \frac{d^i y(t)}{dt^i} = \sum_{i=0}^{n-1} b_i \frac{d^i u(t)}{dt^i}$$

with zero initial conditions. Integrate both sides of the equation n times and assume there exists a $c \in \mathbb{R}_{LC}\langle\langle X \rangle\rangle$ such that $y(t) = F_c[u](t)$. Then after applying the formal Laplace-Borel transform, the equation becomes

$$\begin{aligned}
\left(\delta + \sum_{i=0}^{n-1} a_i\, x_0^{n-1-i} x_1\right) \circ c &= \sum_{i=0}^{n-1} b_i\, x_0^{n-1-i} x_1 \\
\left(1 + \sum_{i=0}^{n-1} a_i\, x_0^{n-i}\right) c &= \sum_{i=0}^{n-1} b_i\, x_0^{n-1-i} x_1.
\end{aligned}$$

Therefore,

$$c = \left(1 + \sum_{i=0}^{n-1} a_i\, x_0^{n-i}\right)^{-1} \sum_{i=0}^{n-1} b_i\, x_0^{n-1-i} x_1.$$

Rephrased in the language of the integral Laplace transform, this is equivalent to

$$\begin{aligned} Y(s) &= \left(1+\sum_{i=0}^{n-1} a_i \frac{1}{s^{n-i}}\right)^{-1} \left(\sum_{i=0}^{n-1} b_i \frac{1}{s^{n-i}}\right) U(s) \\ &= \left(s^n + \sum_{i=0}^{n-1} a_i s^i\right)^{-1} \left(\sum_{i=0}^{n-1} b_i s^i\right) U(s). \end{aligned}$$

Now consider the nonlinear differential equation

$$\frac{d^n y(t)}{dt^n} + \sum_{i=0}^{n-1} a_i \frac{d^i y(t)}{dt^i} + \sum_{i=2}^{k} p_i\, u(t) y^i(t) = \sum_{i=0}^{n-1} b_i \frac{d^i u(t)}{dt^i}$$

with zero initial conditions. Again integrate both side of the equation n times and assume $y(t) = F_c[u](t)$. Applying the formal Laplace-Borel transform gives

$$\begin{aligned} \left(\delta + \sum_{i=0}^{n-1} a_i\, x_0^{n-1-i} x_1\right) \circ c + \sum_{i=2}^{k} p_i\, x_0^{n-1} x_1 (c^{\sqcup\!\sqcup\, i}) &= \sum_{i=0}^{n-1} b_i\, x_0^{n-1-i} x_1 \\ \left(1 + \sum_{i=0}^{n-1} a_i\, x_0^{n-i}\right) c + \sum_{i=2}^{k} p_i\, x_0^{n-1} x_1 (c^{\sqcup\!\sqcup\, i}) &= \sum_{i=0}^{n-1} b_i\, x_0^{n-1-i} x_1. \end{aligned}$$

As in [23], a recursive procedure can be applied to solve the algebraic equation iteratively so that

$$c = c_1 + c_2 + \cdots$$

with

$$c_1 = \left(1 + \sum_{i=0}^{n-1} a_i\, x_0^{n-i}\right)^{-1} \sum_{i=0}^{n-1} b_i\, x_0^{n-1-i} x_1$$

and for $n \geq 2$

$$c_n = \left(1 + \sum_{i=0}^{n-1} a_i\, x_0^{n-i}\right)^{-1} x_0^{n-1} x_1 \sum_{j=2}^{k} p_j \sum_{\substack{\nu_1 \geq 1, \ldots, \nu_j \geq 1 \\ \nu_1 + \nu_2 + \cdots + \nu_j = n}} c_{\nu_1} \sqcup\!\sqcup\, c_{\nu_2} \sqcup\!\sqcup \cdots \sqcup\!\sqcup\, c_{\nu_j}.$$

□

4.4 Symbolic Calculus for the Output Response of Fliess Operators

In [20, 23, 41], a symbolic calculus was developed to compute the output response of an analytic nonlinear system represented by a Volterra operator. It is known that all Volterra operators with analytic kernels are Fliess operators. the formal Laplace-Borel transform pair between the composition product of formal power series and the composition of Fliess operators in Theorem 4.3.1 provides the link to the symbolic calculus.

Theorem 4.4.1. *[42] Let F_c be a Fliess operator with $c \in \mathbb{R}^{\ell}_{LC}\langle\langle X \rangle\rangle$, and $y = F_c[u]$ with u analytic. If c_u denotes the formal Laplace transform of the input u then y is analytic with Laplace transform $c_y = c \circ c_u$.*

Proof: The analyticity of y follows from [66, Lemma 2.3.8]. The identity follows from equation (4.2). Specifically, for any admissible input v:

$$\begin{aligned} F_{c_y}[v] = y &= F_c[F_{c_u}[v]] \\ &= F_{c \circ c_u}[v]. \end{aligned}$$

Then by [66, Corollary 2.2.4] it follows that $c_y = c \circ c_u$. ■

Theorem 4.4.1 provides a compact interpretation of the symbolic calculus of Fliess by applying the relationship between the composition product and the formal Laplace-Borel transform in Theorem 4.3.1. Lemma 2.3.2 suggests some formulae for computing certain system output responses using the formal Laplace-Borel transform. The zero input response of a system is always its natural response. For a system with a generating series in only one letter x_0, the output response is independent of the system input, that is, the formal Laplace transform of the output response y is always identical to the generating series of the input-output system. Lemma 2.3.2 part *3* provides a formal Laplace-Borel transform approach to compute the unit step response of nonlinear systems. As the formal Laplace transform of the unit step signal is $c_u = 1$, therefore the formal Laplace transform of the output y is given by replacing each variable $x_i (i \neq 0)$ in the generating series of c by x_0, that is, $c_y = c \circ 1 = c_{\mathbb{I}} := \sum_{\eta \in X^*} (c, \eta)\, x_0^{|\eta|}$. The following examples further illustrate the application of the formal Laplace-Borel transform in computing the output response of analytic nonlinear systems.

Example 4.4.1. [42] Consider a simple Wiener system as shown in Figure 4.2 where $z(0) = 0$.

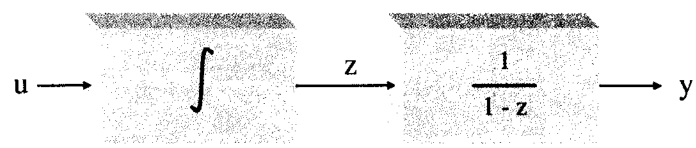


Figure 4.2: A simple Wiener system.

The mapping $u \mapsto y$ can be written as

$$y(t) = \sum_{n=0}^{\infty} (E_{x_1}[u](t))^n = \sum_{n=0}^{\infty} E_{x_1^{\sqcup\!\sqcup\, n}}[u](t) = \sum_{n=0}^{\infty} n!\; E_{x_1^n}[u](t).$$

Therefore $y = F_c[u]$ where $c = \sum_{n \geq 0} n!\, x_1^n$. When $u(t) = t^m/m!$, for example, the formal Laplace transform of u is $c_u = x_0^m$. From Theorem 4.4.1 and equation (4.3) it follows that

$$c_y = \sum_{n=0}^{\infty} n!\, x_1^n \circ x_0^m = \sum_{n=0}^{\infty} (x_0^{m+1})^{\sqcup\!\sqcup\, n} = \sum_{n=0}^{\infty} \frac{((m+1)n)!}{(m+1)!^n} x_0^{(m+1)n}.$$

Consequently,

$$y(t) = \sum_{n=0}^{\infty} \frac{((m+1)n)!}{(m+1)!^n} \frac{t^{(m+1)n}}{((m+1)n)!} = \sum_{n=0}^{\infty} \frac{t^{(m+1)n}}{(m+1)!^n} = \frac{1}{1 - \frac{t^{m+1}}{(m+1)!}}.$$

□

Example 4.4.2. Consider a linear time-invariant system with the generating series $c = \sum_{n\geq 0}(c, x_0^n x_1)x_0^n x_1$. Suppose the input to the system is $u(t) = \cos(\omega t)$. Then the output response of the system can be computed by the formal Laplace-Borel transform as follows. The generating series for the input cosine signal is $c_u = \mathscr{L}_f\left[\cos(\omega t)\right] = \frac{1}{1+(\omega x_0)^2}$, therefore, by Theorem 4.4.1, the generating series for the output response is

$$\begin{aligned} c_y = c \circ c_u &= \sum_{n\geq 0}(c, x_0^n x_1)x_0^n x_1 \circ \frac{1}{1+(\omega x_0)^2} \\ &= \sum_{n\geq 0}(c, x_0^n x_1)\frac{x_0^{n+1}}{1+(\omega x_0)^2}. \end{aligned}$$

Applying the Borel transform, the output response $y(t)$ is

$$\begin{aligned} y(t) &= \mathscr{B}_f\left[c_y\right](t) \\ &= \mathscr{B}_f\left[\sum_{n\geq 0}(c, x_0^n x_1)\frac{x_0^{n+1}}{1+(\omega x_0)^2}\right] \\ &= \sum_{n\geq 0}\frac{(c, x_0^n x_1)}{\omega^{n+1}}\cos\left(\omega t - \frac{(n+1)\pi}{2}\right) \\ &\quad + \sum_{n\geq 0,\, n\ even}(c, x_0^n x_1)\sum_{i=0}^{\frac{n}{2}-1}\frac{(-1)^{\frac{n}{2}-i-1}}{(2i+1)!\,\omega^{n-2i}}t^{2i+1} \\ &\quad + \sum_{n\geq 0,\, n\ odd}(c, x_0^n x_1)\sum_{i=0}^{\frac{n-1}{2}}\frac{(-1)^{\frac{n-1}{2}-i}}{(2i)!\,\omega^{n+1-2i}}t^{2i}. \end{aligned} \tag{4.4}$$

□

4.5 The Isomorphisms Induced by the Formal Laplace-Borel Transform

For all $c \in \mathbb{R}_{LC}\langle\langle X\rangle\rangle$, the associated operator F_c is a well-defined Fliess operator. As has been discussed in earlier chapters, many binary operations can be defined over the set

$\mathcal{F}$: addition, multiplication, the composition and modified composition. The collection of Fliess operators $\mathcal{F}$, with these basic operations, has the algebraic structures described in the following theorem.

Theorem 4.5.1. *The following statements are true:*

1. $(\mathcal{F}, +)$ *is a commutative group.*
2. $(\mathcal{F}, \cdot)$ *is a commutative monoid, the identity of which is* $F_\emptyset = 1$.
3. $(\mathcal{F}, \circ)$ *is a semigroup.*
4. $(\mathcal{F}, +, \cdot)$ *is a commutative ring.*
5. $(\mathcal{F}, +, \cdot)$ *is an* $\mathbb{R}$*-algebra.*
6. $(\mathcal{F}, +, \cdot)$ *is an integral domain.*

Proof: Statement 1,2 and 3 are straightforward. The proofs for statement 4, 5 and 6 follow analogously to the proofs for Theorem 2.4.1. ∎

Observe that $(\mathcal{F}, +, \circ)$ can NOT form a ring, as the composition product is not right distributive over addition, i.e., $F_c \circ (F_d + F_e) \neq F_c \circ F_d + F_c \circ F_e$.

Theorem 4.5.2. *Under the formal Laplace-Borel transform, the following statements are true:*

1. $(\mathbb{R}\langle\langle X \rangle\rangle, +)$ *and* $(\mathcal{F}, +)$ *are isomorphic commutative groups.*
2. $(\mathbb{R}\langle\langle X \rangle\rangle, ⧢)$ *and* $(\mathcal{F}, \cdot)$ *are isomorphic commutative monoids, the identity of which is* $c_{\mathcal{I}} = 1 = \emptyset$ *and* $F_\emptyset = 1$ *respectively.*
3. $(\mathbb{R}\langle\langle X \rangle\rangle, \circ)$ *and* $(\mathcal{F}, \circ)$ *are isomorphic semigroups.*

4. $(\mathbb{R}\ll X\gg, +, ⧢)$ *and* $(\mathcal{F}, +, \cdot)$ *are isomorphic commutative rings.*

5. $(\mathbb{R}\ll X\gg, +, ⧢)$ *and* $(\mathcal{F}, +, \cdot)$ *are isomorphic* $\mathbb{R}$*-algebras.*

6. $(\mathbb{R}\ll X\gg, +, ⧢)$ *and* $(\mathcal{F}, +, \cdot)$ *are isomorphic integral domains.*

Proof:

The proofs follow directly from Theorem 2.4.1 and Theorem 4.5.1. ■

CHAPTER V

IMPLEMENTATION PACKAGE IN MAPLE

5.1 Introduction

The main purpose of this chapter is to provide a software implementation of the main tools described in the previous chapters. Based on the ACE package developed by the ACE group in Université de Marne-la-Vallée led by Sébastien Veigneau [64], an implementation package in Maple is presented. The ACE package provides some binary operations on the free monoid of formal polynomials, such as the concatenation product and the shuffle product. The general purpose of this chapter is to demonstrate the implementation of the basic operations involved in the previous chapters over the set of formal polynomials: the left and right chronological product, the composition product, the modified composition product, and also some other operations such as the degree and order of a formal polynomial, the ultrametric distance between two formal polynomials, etc. Examples are provided to illustrate the usage of the commands, and also to demonstrate some of the properties related to these operations. A user guide as well as the source code is provided in the appendices.

5.2 Operations on Formal Polynomials

The main binary operations involved in the analysis are addition, concatenation, the shuffle, chronological, composition and modified composition products. Three fundamental operations: addition, concatenation and the shuffle product are available in the ACE package. Those fundamental operations provide the building blocks for other operations. The following examples demonstrate the basic operations over the set of formal polynomials.

Example 5.2.1. The following two commands are applied to determine the degree and order of a formal polynomial.

```
> FreeDegree(2*w[1]+2*w[2,3,4]);
```

gives the degree of the formal polynomial

$$3$$

```
> FreeOrder(2*w[1]+2*w[2,3,4]);
```

computes the order of the formal polynomial

$$1$$

The command 'FreeDist' is used to compute the ultrametric distance between two formal polynomials. For example,

```
> FreeDist(w[0,1], w[0,1]+2*w[2,3,4,5], sigma);
```

$$\sigma^4$$

Examples involving the left shift and right shift operators are given below.

```
> FreeLShift(w[1,2], w[1,2,3,4,5]+w[0,1]+w[1,2]);
```

$$w_{3,4,5} + w_{[]}$$

```
> FreeRShift(w[1,2,3,4,5]+w[0,1]+3*w[4,5], w[4,5]);
```

$$w_{1,2,3} + 3w_{[]}$$

The composition and the modified composition products are illustrated next. For the single-output case, use `FreeCompose(c, [d])` for the composition product and `FreeModCompose(c, [d])` for the modified composition product. The return value is a scalar formal power series.

```
> FreeCompose(w[1,2]+2*w[0,1], [w[1],w[2,0,3]]);
```

$$w_{0,1,0,2,0,3} + w_{0,0,1,2,0,3} + w_{0,0,2,0,1,3} + w_{0,0,2,1,0,3} + w_{0,0,2,0,3,1} + 2w_{0,0,1}$$

```
> FreeModCompose(w[1,2]+2*w[0,1], [w[1],w[2,0,3]]);
```

$$w_{1,2} + w_{1,0,2,0,3} + w_{0,1,0,2,0,3} + w_{0,1,2} + w_{0,0,1,2,0,3} + w_{0,0,2,0,1,3} + w_{0,2,1}$$
$$+ w_{0,0,2,1,0,3} + w_{0,0,2,0,3,1} + 2w_{0,1} + 2w_{0,0,1}$$

For the multi-output case, the commands FreeComposeMIMO and FreeModComposeMIMO are used, respectively. For the multi-output case, the return value is a row vector. For example,

```
> FreeComposeMIMO([w[1,2]+2*w[0,1], w[1], [w[1],w[2,0,3]]);
```

$$\left[w_{0,1,0,2,0,3} + w_{0,0,1,2,0,3} + w_{0,0,2,0,1,3} + w_{0,0,2,1,0,3} + w_{0,0,2,0,3,1} + 2w_{0,0,1}, \quad w_{0,0,1} + w_{0,1,0} \right]$$

```
> FreeModComposeMIMO([w[1,2], w[1,0]], [w[1],w[2,3]]);
```

$$\begin{bmatrix} w_{1,2} + w_{1,0,2,3} + w_{0,0,2,3,1} + w_{0,1,2} + w_{0,0,2,1,3} + w_{0,1,0,2,3} + w_{0,0,1,2,3} + w_{0,2,1} \\ w_{1,0} + w_{0,0,1} + w_{0,1,0} \end{bmatrix}^T$$

□

Example 5.2.2. The following examples illustrate the properties of the chronological, composition, and modified composition products.

A verification of the symmetrization of the left and right chronological products is considered first.

```
> aa1:=FreeLChro(w[1,2,3],w[4,5])+FreeLChro(w[4,5],w[1,2,3]);
> bb1:=FreeShuffle(w[1,2,3],w[4,5]);
> evalb(aa1=bb1);
```

The left chronological product would be

$$
\begin{aligned}
aa1 &:= w_{4,5,1,2,3} + w_{4,1,5,2,3} + w_{4,1,2,3,5} + w_{4,1,2,5,3} + w_{1,4,2,5,3} \\
&\quad + w_{1,2,4,3,5} + w_{1,4,2,3,5} + w_{1,4,5,2,3} + w_{1,2,3,4,5} + w_{1,2,4,5,3} \\
bb1 &:= w_{4,5,1,2,3} + w_{4,1,5,2,3} + w_{4,1,2,3,5} + w_{4,1,2,5,3} + w_{1,4,2,5,3} \\
&\quad + w_{1,2,4,3,5} + w_{1,4,2,3,5} + w_{1,4,5,2,3} + w_{1,2,3,4,5} + w_{1,2,4,5,3}
\end{aligned}
$$

true.

```
> aa2:=FreeRChro(w[1,2,3],w[4,5])+FreeRChro(w[4,5],w[1,2,3]);
> bb2:=FreeShuffle(w[1,2,3],w[4,5]);
> evalb(aa2=bb2);
```

The right chronological product would be

$$
\begin{aligned}
aa2 &:= w_{4,5,1,2,3} + w_{4,1,5,2,3} + w_{4,1,2,3,5} + w_{4,1,2,5,3} + w_{1,4,2,5,3} \\
&\quad + w_{1,2,4,3,5} + w_{1,4,2,3,5} + w_{1,4,5,2,3} + w_{1,2,3,4,5} + w_{1,2,4,5,3} \\
bb2 &:= w_{4,5,1,2,3} + w_{4,1,5,2,3} + w_{4,1,2,3,5} + w_{4,1,2,5,3} + w_{1,4,2,5,3} \\
&\quad + w_{1,2,4,3,5} + w_{1,4,2,3,5} + w_{1,4,5,2,3} + w_{1,2,3,4,5} + w_{1,2,4,5,3}
\end{aligned}
$$

true.

The next example illustrates the associativity of the composition product.

```
> aa3:=FreeCompose(FreeCompose(w[1,0,2]+2*w[0,1],[w[1],w[2,0,3]]),
                              [w[1],w[2],w[3]]):
> bb3:=FreeCompose(w[1,0,2]+2*w[0,1],[FreeCompose(w[1],[w[1],w[2],w[3]]),
       FreeCompose(w[2,0,3], [w[1],w[2],w[3]]) ]):
> evalb(aa1=bb1);
```

true

The modified composition product is not associative. The following is an example.

```
> aa4:=FreeModCompose(FreeModCompose(w[1,0,2]+2*w[0,1],
        [ w[1], w[2,0,3]]), [w[1],w[2],w[3] ]):
> bb4:=FreeModCompose( w[1,0,2]+2*w[0,1], [ FreeModCompose(w[1],
     [w[1],w[2],w[3]]), FreeModCompose(w[2,0,3], [w[1],w[2],w[3]]) ] ):
> evalb(aa4=bb4);
```

false.

□

5.3 Operations on Fliess Operators

Operations on $\mathcal{F}$ include computing the output of a Fliess operator given a formal polynomial c and an input vector of time functions $u(t)$. In the following examples, the basic properties of the formal Laplace-Borel transform are illustrated. The first example illustrates how to compute the output response of a Fliess operator for a given input. For a multi-input-single-output Fliess operator, use `Fliess(c, ut)`. For multi-input-multi-output case, use `FliessMIMO(c, ut)`.

Example 5.3.1. Consider a 3-input, 1-output nonlinear system with the generating series $c := x_1x_2x_3 + 2x_0x_1x_0x_2x_0x_3$. Let the input signal applied to the system be $ut :=$ $[\cos(t), \cos(2t), \cos(3t)]$. Then the following commands implement this model:

```
> ut := [cos(t), cos(2 t), cos(3 t)];
> Fliess(w[1,2,3]+2*w[0,1,0,2,0,3], ut);
```

The corresponding output is

$$-1/240\sin(4t) - 1/360\sin(6t) - 1/24\sin(2t) - 1/12t + 1/5\sin(t) - 1/72\cos(2t) + 1/36t^2$$

$$-1/7200\cos(4t) - 1/16200\cos(6t) + 79/900\cos(t) + 1/324\cos(3t) - 199/2592.$$

For multi-output case, the command FliessMIMO is applied. For example, the command

```
> c:=[w[1,1], w[1,2], w[1,3]];
> FliessMIMO(c, ut);
```

gives the output vector

$$\begin{bmatrix} -1/2\cos(t) + 1/2, \\ -1/12\cos(3t) - 1/4\cos(t) + 1/3, \\ -1/24\cos(4t) - 1/12\cos(2t) + 1/8 \end{bmatrix}$$

□

In the following examples, the basic properties of the formal Laplace-Borel transform are demonstrated.

Example 5.3.2. First consider the isomorphism between $(\mathbb{R}_{LC}\langle\langle X\rangle\rangle, ⧢)$ and $(\mathcal{F}, \cdot)$ in Theorem 4.5.2. Applying the following commands

```
> gt := [t, t^2, t^3];
> a1:= Fliess(FreeShuffle(w[1,0]+2*w[1,2,3],w[1,2,1]),gt);
> b1:= Fliess(w[1,0]+2*w[1,2,3],gt)*Fliess(w[1,2,1],gt);
> evalb(a1=expand(b1));
```

gives

$$a1 := \frac{1}{210}t^{10} + \frac{1}{8820}t^{16}$$

$$b1 := \frac{1}{70}(\frac{1}{3}t^3 + \frac{1}{126}t^9)t^7$$

true.

Next, an illustration of the correspondence between the derivative of the output of a Fliess operator and the left shift operator is provided, as well as the correspondence between the integral and the concatenation with x_0 in Theorem 4.2.1.

```
> a2:=diff(diff(Fliess(w[0,0,1,0]+2*w[0,0,2,3], ut),t),t);

> b2:= Fliess(FreeLShift(w[0,0],w[0,0,1,0]+2*w[0,0,2,3]),ut);

> evalb(a2=b2);
```

The following illustrates the equivalence:

$$a2 := 2/3\cos(t) + t\sin(t) - 3/5 - 1/15\cos(5t)$$

$$b2 := 2/3\cos(t) + t\sin(t) - 3/5 - 1/15\cos(5t)$$

true

For the integration property of the formal Laplace-Borel transform, the commands

```
> a3:=int(Fliess(w[1,0]+2*w[2,3], ut),t)-
        eval(int(Fliess(w[1,0]+2*w[2,3],ut),t),t=0);

> b3:= Fliess(FreeConcat(w[0],w[1,0]+2*w[2,3]),ut);

> evalb(a3=b3);
```

produce

$$a3 := 5/3\sin(t) - t\cos(t) - 3/5t - 1/75\sin(5t)$$

$$b3 := 5/3\sin(t) - t\cos(t) - 3/5t - 1/75\sin(5t)$$

true

The isomorphism between the two semigroups $(\mathbb{R}\ll X\gg, \circ)$ and $(\mathcal{F}, \circ)$ is illustrated by the following.

```
> a4:=Fliess(FreeCompose(w[1,2,3], [w[1],w[2,2],w[3,3,3]]),ut);

> b4:=Fliess(w[1,2,3], FliessMIMO( [ w[1], w[2,2], w[3,3,3]] ,ut)) ;

> evalb(a4=b4);
```

The result below demonstrates that the Fliess operator of the composition product is the composition of Fliess operators.

$$\begin{aligned} a4 \; := \; & \frac{13}{19595520}\sin(6t) - \frac{1}{11664}t\cos(t) + \frac{1}{41472}t - \frac{1}{58226688}\sin(12t) + \frac{1}{466560}\sin(5t) \\ & - \frac{67}{188116992}\sin(8t) + \frac{1}{11664}\sin(t) - \frac{5}{248832}\sin(2t) - \frac{1}{279936}\sin(3t) \\ & + \frac{29}{7464960}\sin(4t) + \frac{1}{67931136}\sin(14t) - \frac{1}{16796160}\sin(10t) \\ b4 \; := \; & \frac{13}{19595520}\sin(6t) - \frac{1}{11664}t\cos(t) + \frac{1}{41472}t - \frac{1}{58226688}\sin(12t) + \frac{1}{466560}\sin(5t) \\ & - \frac{67}{188116992}\sin(8t) + \frac{1}{11664}\sin(t) - \frac{5}{248832}\sin(2t) - \frac{1}{279936}\sin(3t) \\ & + \frac{29}{7464960}\sin(4t) + \frac{1}{67931136}\sin(14t) - \frac{1}{16796160}\sin(10t) \end{aligned}$$

true.

The following sequence of demands demonstrate that the modified composition product is corresponding to the modified composition connection, as shown in Theorem 3.4.6.

```
> a6:=Fliess(FreeModCompose(w[1,2,3], [w[1],w[2,2],w[3,3,3]]), ut);

> b6:=Fliess(w[1,2,3],  ut+FliessMIMO( [ w[1], w[2,2], w[3,3,3]], ut));
```

```
> evalb(a6=b6);
```

The outcome of this procedure is

$$
\begin{aligned}
a6 \quad := \quad & \frac{1}{466560}\cos(5t) + \frac{1}{67931136}\cos(14t) - \frac{17239}{6531840}\sin(6t) - \frac{1}{11664}t\cos(t) - \frac{1229}{13824}t \\
& + \frac{114169}{39191040}\cos(6t) - \frac{24449}{7464960}\cos(4t) - \frac{197}{640493568}\cos(12t) - \frac{73}{213497856}\sin(12t) \\
& + \frac{1}{466560}\sin(5t) - 17735/188116992\cos(8t) - \frac{83}{184757760}\cos(10t) + \frac{1}{11664}t\sin(t) \\
& + \frac{6060581}{34836480} + \frac{17285}{188116992}\sin(8t) + \frac{88303}{408240}\sin(t) - \frac{11405}{248832}\sin(2t) - \frac{11}{93312}\sin(3t) \\
& - \frac{37699}{7464960}\sin(4t) - \frac{29411}{136080}\cos(t) + \frac{10631}{248832}\cos(2t) - \frac{31}{279936}\cos(3t) \\
& + \frac{1}{67931136}\sin(14t) + \frac{61}{184757760}\sin(10t) \\
b6 \quad := \quad & \frac{1}{466560}\cos(5t) + \frac{1}{67931136}\cos(14t) - \frac{17239}{6531840}\sin(6t) - \frac{1}{11664}t\cos(t) - \frac{1229}{13824}t \\
& + \frac{114169}{39191040}\cos(6t) - \frac{24449}{7464960}\cos(4t) - \frac{197}{640493568}\cos(12t) - \frac{73}{213497856}\sin(12t) \\
& + \frac{1}{466560}\sin(5t) - 17735/188116992\cos(8t) - \frac{83}{184757760}\cos(10t) + \frac{1}{11664}t\sin(t) \\
& + \frac{6060581}{34836480} + \frac{17285}{188116992}\sin(8t) + \frac{88303}{408240}\sin(t) - \frac{11405}{248832}\sin(2t) - \frac{11}{93312}\sin(3t) \\
& - \frac{37699}{7464960}\sin(4t) - \frac{29411}{136080}\cos(t) + \frac{10631}{248832}\cos(2t) - \frac{31}{279936}\cos(3t) \\
& + \frac{1}{67931136}\sin(14t) + \frac{61}{184757760}\sin(10t)
\end{aligned}
$$

true.

$\square$

CHAPTER VI

CONCLUSIONS AND FUTURE RESEARCH

6.1 Main Conclusions

The main contributions of this dissertation are: the development of a growth condition for the local convergence property of interconnected Fliess operators in the cascade and feedback interconnections, the definition of the formal Laplace-Borel transform of a Fliess operator, and the description of the algebraic structures of the set of formal power series and the set of Fliess operators behind the formal Laplace-Borel transform.

The four basic interconnections of analytic nonlinear systems represented by Fliess operators are described in a unified manner. The corresponding generating series for cascaded Fliess operators in the multi-variable case are given in Definition 2.3.14. The composition product of two locally convergent formal power series is shown to still be locally convergent, and a growth condition for the coefficients is given in Theorem 3.3.2. The generating series for the feedback connection of two Fliess operators is shown in Theorem 3.4.7 to be always well defined and in Theorem 3.4.9 it is proven to be at least input-output locally convergent.

The definition of the formal Laplace-Borel transform of a Fliess operator is given in Definition 4.2.1, and its basic properties are presented in Theorem 4.2.1. By combining the idea of the formal Laplace-Borel transform with the composition product, it is shown in Theorem 4.3.1 that the formal Laplace-Borel transform provides an isomorphism between the semigroup of all convergent Fliess operators under composition, and the semigroup of all locally convergent formal power series under the composition product. This result provides a generalization of the time domain and frequency domain isomorphism in the linear case. Specifically, an explicit relationship is derived between the formal Laplace-Borel transforms of the input and output signals of a Fliess operator in Theorem 4.4.1. This result provides a compact interpretation of the symbolic calculus introduced by Fliess et al. [23] to compute the output response of nonlinear systems.

Finally, a set of isomorphic algebraic structures for the set of formal power series and the set of Fliess operators is described in Theorem 4.5.2 with the aid of the system interconnection theory and the formal Laplace-Borel transform theory developed in this dissertation.

6.2 Future Research

Among the many ideas for future research, a logical next step would include a deeper understanding of the algebraic structure of $\mathbb{R}_{LC}\langle\langle X\rangle\rangle$, the properties of Laplace-Borel transform, and how they are related in the framework of system interconnections. For example, it is already known that the system interconnections corresponding to addition, the shuffle, composition and feedback products are the four elementary interconnections shown in Figure 1.1. However, the corresponding interconnection for the concatenation product is still not clear. There exists certain duality between the shuffle algebra and concatenation when viewed as the linear mappings on a tensor product space [52]. In addition, the property of the formal Laplace-Borel transform concerning the concatenation of inputs in Theorem 4.2.1 also suggests some connection between the shuffle and the concatenation [62, 66].

Another interesting idea would be to further develop the algebraic properties of the composition product. There has been considerable research results on the structure of the shuffle algebra [3, 43, 52]. From Definition 2.3.14, the composition product is clearly connected to the shuffle and concatenation. Equation (2.8) also suggests an alternative definition of the composition product in terms of the left chronological product. The algebraic properties of the chronological product has been studied in [36, 38, 40]. Therefore, the relationships between the composition and chronological products as well as the shuffle product, can provide some insight into the algebraic properties of the composition product.

In the definition of the formal Laplace-Borel transform of a Fliess operator in Chapter IV, the set of admissible inputs is assumed to lie within the L_∞ space. Can this admissible set of inputs be expanded to the L_1 space? The expansion seems possible in current setting.

The properties of the formal Laplace-Borel transform and their applications in system analysis is another important topic. Most of the properties of linear integral Laplace-Borel transform have some counterparts in the nonlinear setting. For example, the linearity property is identical to that for the linear system case, and the property concerning the scalability of inputs is analogous to the time scaling property in the linear integral Laplace-Borel transform. Therefore, it is natural to explore other possible corresponding properties in the context of Fliess operators and formal power series. For example, what are the properties corresponding to the time-shift and frequency-shift properties in the integral Laplace-Borel transform? Furthermore, what new properties can be identified using the formal Laplace-Borel transform that only arise in the nonlinear setting?

The local convergence property of the feedback product is not yet perfectly characterized. In Theorem 3.4.9, the feedback product is proven to be always input-output locally convergent. In some special cases, input-output local convergence guarantees local convergence, as shown in Lemma 3.3.4 and Lemma 3.3.5. Whether this input-output local convergence *always* implies local convergence is still not completely understood.

The formal Laplace-Borel transform is a tool to analyze nonlinear systems using their generating series. In the linear case, the Laplace-Borel transform corresponds to frequency domain analysis of linear systems. In [53], the frequency response of a nonlinear system is characterized by association of variables. In the current setting, the generating series of a Fliess operator plays a similar role to the transfer function. Therefore, a very interesting future topic in nonlinear system analysis would be to show how this generating series approach can provide an alternative interpretation to the frequency response analysis of the nonlinear systems represented by Fliess operators.

REFERENCES

[1] M. Araki and M. Saeki, *A quantitative condition for the well-posedness of interconnected dynamical systems*, IEEE Trans. Automatic Control **AC-28** (1983), no. 5, 569–577.

[2] H. F. Baker, *Alternants and continuous groups*, Proc. of the London Mathematical Society **2** (1905), no. 3, 24–47.

[3] J. Berstel and C. Reutenauer, *Les Séries Rationelles et Leurs Langages*, Springer-Verlag, Paris, 1984.

[4] R. W. Brockett, R. S. Millman, and H. J. Sussmann, *Nonlinear Control Systems*, Birkhäuser, Boston, 1983.

[5] J. E. Campbell, *On a law of combination of operators bearing on the theory of continuous transformation groups*, Proc. of the London Mathematical Society **1** (1898), no. 28, 381–390.

[6] R. Chabour and A. Ferfera, *Singularity for static-state feedback linearizable bilinear systems*, IEEE Trans. Automatic Control **AC-44** (1999), no. 8, 1559–1564.

[7] K. T. Chen, *Iterated integrals and exponential homomorphisms*, Proc. London Math. Soc. **3** (1954), no. 4, 502–512.

[8] ——, *Integration of paths, geometric invariants and a generalized Baker-Hausdorff formula*, Ann. of Math. **65** (1957), no. 2, 163–178.

[9] ——, *On the composition functions of nilpotent Lie groups*, Proc. Amer. Math. Soc. **8** (1957), no. 6, 1158–1159.

[10] ——, *Integration of paths—a faithful representation of paths by non-commutative formal power series*, Trans. Amer. Math. Soc. **89** (1958), no. 2, 395–407.

[11] ——, *Algebraization of iterated integration along paths*, Bull. Amer. Math. Soc. **73** (1967), 975–978.

[12] ——, *Iterated path integrals and generalized paths*, Bull. Amer. Math. Soc. **73** (1967), 935–938.

[13] ——, *Differential forms and homotopy groups*, J. Differential Geometry **6** (1971/72), 231–246.

[14] ——, *Iterated integrals of differential forms and loop space homology*, Ann. of Math. **97** (1973), no. 2, 217–246.

[15] ——, *Iterated path integrals*, Bull. Amer. Math. Soc. **83** (1977), no. 5, 831–879.

[16] N. Chomsky and M. P. Schützenberger, *The algebraic theory of contextfree languages*, Computer Programming and Formal Systems **6** (1963), 118–161.

[17] A. Ferfera, *Combinatoire du monoïde libre et composition de certains systèmes non linéaires*, Analyse des Systèms, Soc. Math. France **75–76** (1980), 87–93.

[18] M. Fliess, *Sur la réalisation des systèmes dynamiques bilinéaires*, C. R. Acad. Sci., Paris, France **A-277** (1973), 923–926.

[19] ——, *Matrices de Hankel*, J. Math. Pures Appl. **53** (1974), 197–224 (erratum, **54** (1975)).

[20] ——, *Fonctionnelles causales non linéaires et indéterminées non commutatives*, Bull. Soc. Math. France **109** (1981), 3–40.

[21] ——, *Réalisation locale des systèmes non linéaires, algèbres de Lie filtrées transitives et séries génératrices non commutatives*, Invent. Math. **71** (1983), 521–537.

[22] ——, *Cascade decomposition of nonlinear systems, foliation and ideals of transitive Lie algebras*, Systems and Control Letters **5** (1985), no. 4, 263–265.

[23] M. Fliess, M. Lamnabhi, and F. Lamnabhi-Lagarrigue, *An algebraic approach to nonlinear functional expansions*, IEEE Trans. Circuits Syst. **30** (1983), no. 8, 554–570.

[24] W. S. Gray and Y. Li, *Generating series for interconnected analytic nonlinear systems*, SIAM J. Control and Optimization, to appear.

[25] ——, *Fliess operators in cascade and feedback systems*, Proc. 2002 Conference on Information Sciences and Systems(CISS'02), Princeton, New Jersey, U.S.A., March 2002, pp. 173–178.

[26] ——, *Generating series for nonlinear cascade and feedback systems*, Proc. 41$^{\text{st}}$ IEEE Conference on Decision and Control, Las Vegas, Nevada, U.S.A., December 2002, pp. 2720–2725.

[27] ——, *Interconnected systems of Fliess operators*, Proc. 15$^{\text{th}}$ International Symposium on Mathematical Theory of Networks and Systems, South Bend, Indiana, U.S.A., August 2002.

[28] W. S. Gray and Y. Wang, *Fliess operators on L_p spaces: convergence and continuity*, Systems and Control Letters **46** (2002), no. 2, 67–74.

[29] F. Hausdorff, *Die symbolische Exponential formel in der Gruppentheorie*, Leipziger Berichte **58** (1906), 19–48.

[30] U. Heckmanns, *Aspects of Ultrametric Spaces*, Queen's Papers in Pure and Applied Math., vol. 109, Queen's University, Kingston, Ontario, 1998.

[31] A. Isidori, *Nonlinear Control Systems*, 3rd ed., Springer-Verlag, London, 1995.

[32] A. Isidori, A. Krener, C. Gori-Giorgi, and S. Monaco, *Nonlinear decoupling via feedback: a differential geometric approach*, IEEE Trans. Automatic Control **AC-26** (1981), no. 2, 331–345.

[33] B. Jakubczyk, *Existence and uniqueness of realizations of nonlinear systems*, SIAM J. Control and Optimization **18** (1980), 445–471.

[34] ——, *Local realization of nonlinear causal operators*, SIAM J. Control and Optimization **24** (1986), 230–242.

[35] T. Kailath, *Linear Systems*, Prentice-Hall, Englewood Cliffs, NJ, 1971.

[36] M. Kawski, *Chronological algebras and nonlinear control*, Proc. Asian Control Conference, Tokyo, Japan, March 1994.

[37] ——, *Nonlinear control and combinatorics of words*, Geometry of Feedback and Optimal Control, Eds. B. Jakubczyk and W. Respondek (1998), 305–346.

[38] ——, *Chronological algebras: combinatorics and control*, Proc. Pontryagin Conf., vol. 64, Moscow, 1999, pp. 144–178.

[39] ——, *Calculus of nonlinear interconnections with applications*, Proc. 39$^{\text{th}}$ Conference on Decision and Control, Sydney, Australia, December 2000, pp. 1661–1666.

[40] M. Kawski and H. J. Sussmann, *Noncommutative power series and formal Lie-algebraic techniques in nonlinear control theory*, Operators, Systems, and Linear Algebra (Kaiserslautern, 1997), European Consort. Math. Indust., Teubner, Stuttgart, 1997, pp. 111–128.

[41] M. Lamnabhi, *A new symbolic calculus for the response of nonlinear systems*, Systems and Control Letters **2** (1982), no. 3, 154–161.

[42] Y. Li and W. S. Gray, *The formal Laplace-Borel transform, Fliess operators and the composition product*, Proc. 36^{th} IEEE Southeastern Symposium on System Theory, Atlanta, Georgia, U.S.A., March 2004, pp. 333–337.

[43] M. Lothaire, *Combinatorics on Words*, Cambridge University Press, Cambridge, 1983.

[44] V. H. N. Minh, *Evaluation transform*, Theoretical Computer Science **79** (1991), no. 1, 163–177.

[45] G. E. Mitzel and W. J. Rugh, *On a multi-dimensional S transform and the realization problem for homogeneous nonlinear systems*, IEEE Trans. on Automatic Control **AC-22** (1977), 825–830.

[46] H. Nijmeijer and A. J. van der Schaft, *Nonlinear Dynamical Control Systems*, Springer Verlag, New York, 1990.

[47] S. Priess-Crampe and P. Ribenboim, *Fixed points, combs and generalized power series*, Abh. Math. Sem. Univ. Hamburg **63** (1993), 227–244.

[48] ——, *Fixed point and attractor theorems for ultrametric spaces*, Forum Math. **12** (2000), 53–64.

[49] R. Ree, *Lie elements and an algebra associated with shuffles*, Annals of Math. **68** (1958), 210–220.

[50] ——, *On generalized Lie elements*, Canadian J. Math. **12** (1960), 493–502.

[51] W. Respondek, *Geometric method in nonlinear control theory: a survey*, Neural Adaptive Control Technology, Eds. R. Zbikowski and K. J. Hunt, World Scientific, Singapore, 1996, pp. 111–148.

[52] C. Reutenauer, *Free Lie Algebras*, Clarendon Press, Oxford Science Publications, Oxford University Press Inc., New York, 1993.

[53] W. J. Rugh, *Nonlinear System Theory - the Volterra/Wiener Approach*, The John Hopkins University Press, Baltimore, Maryland, 1981.

[54] E. Schörner, *Ultrametric fixed point theorems and applications*, Proc. International Conference and Workshop on Valuation Theory, July 1999.

[55] M. P. Schützenberger, *On the definition of a family of automata*, Inf. and Control **4** (1961), 245–270.

[56] ——, *Finite counting automata*, Inf. and Control **5** (1962), 91–107.

[57] ——, *On context-free languages and push-down automata*, Inf. and Control **6** (1963), 246–264.

[58] G. M. Sklyar and S. Yu. Ignatovich, *Representation of control systems in the Fliess algebra and in the algebra of nonlinear power moments*, Systems and Control Letters **47** (2002), no. 3, 27–235.

[59] B. Y. Sternin and V. E. Shatalov, *Borel-Laplace Transform and Asymptotic Theory*, CRC Press, Boca Raton, Florida, 1996.

[60] H. J. Sussmann, *Existence and uniqueness of minimal realizations of nonlinear systems*, Math. System Theory **12** (1977), 263–284.

[61] ——, *Lie brackets and local controllability: A sufficient condition for scalar-input systems*, SIAM J. Control and Optimization **21** (1983), 686–713.

[62] ——, *A product expansion of the Chen series*, SIAM J. Control and Optimization **21** (1983), 686–713.

[63] H. L. Trentelman and J. C. Willems, *The behavioral approach as a paradigm for modelling interconnected systems*, European Journal on Control **9** (2003), 326–339.

[64] `http://phalanstere.univ-mlv.fr/~ace/`, An Algebraic Combinatorics Environment(ACE) for the Computer Algebra system in Maple.

[65] M. Vidyasagar, *On the well-posedness of large-scale interconnected systems*, IEEE Trans. Automatic Control **AC-25** (1980), 413–421.

[66] Y. Wang, *Algebraic Differential Equations and Nonlinear Control Systems*, Ph.D. thesis, Rutgers University, Department of Mathematics, 1990.

[67] Y. Wang and E.D. Sontag, *Generating series and nonlinear systems: analytic aspects, local realizability and i/o representations*, Forum Mathematicum **4** (1992), 299–322.

[68] ——, *Orders of input/output differential equations and state-space dimensions*, SIAM J. Control and Optimization **33** (1995), no. 4, 1102–1126.

[69] J. C. Willems, *Paradigms and puzzles in the theory of dynamical systems*, IEEE Trans. Automatic Control **AC-36** (1991), no. 3, 569–577.

[70] ——, *On interconnections, control and feedback*, IEEE Trans. Automatic Control **AC-42** (1997), no. 3, 326–339.

APPENDIX

SOFTWARE IMPLEMENTATION IN MAPLE

A.1 Overview

This appendix provides a user guide to the software implementation package presented in this dissertation. To set up the software implementation, first follow these two steps:

Installation of Maple and ACE package : After installation of Maple, download the ACE package from `http://phalanstere.univ-mlv.fr/~ace/`, follow the instructions to install ACE package;

Load the FreePoly.mws file : Open in the workspace the FreePoly.wms developed in this dissertation, and load the procedures.

The procedures developed in this dissertation can be divided in two categories:

Operations over $\mathbb{R}<X>$: This category includes a set of Maple procedures to calculate and display operations over the set of formal polynomials. The binary operations include the left and right shift operators, the left and right chronological products, the composition product, the modified composition product, and also the ultrametric distance between two formal polynomials. The unary operations include the length of a word, the degree and order of a formal polynomial.

Operations over $\mathcal{F}$: The operations over the set of Fliess operators mainly involve the calculation of the output response of a Fliess operator given different inputs.

A.2 Syntax Description

The syntax of the procedures are listed as follows.

"**FreeLength**: to calculate the length of a word"

Syntax FreeLength(η::indexed) where

- η is a word of the form $w[i_1 i_2 \cdots i_k]$

"**FreeDegree**: to calculate the degree of a formal polynomial"

Syntax FreeDegree(c) where

- c is a formal polynomial over the alphabet $w[0], w[1], \cdots, w[m]$

"**FreeOrder**: to calculate the order of a formal polynomial"

Syntax FreeOrder(c) where

- c is a formal polynomial over the alphabet $w[0], w[1], \cdots, w[m]$

"**FreeDist**: to calculate the ultrametric distance between two formal polynomials"

Syntax FreeDist(c, d, σ) where

- c, d are two formal polynomials over the alphabet $w[0], w[1], \cdots, w[m]$
- σ is the parameter used in the ultrametric. It can be a symbol or a numeric value.

"**FreeLShift**: Left shift operation of a formal polynomial by a word"

Syntax FreeLShift(η::indexed, c) where

- η is a word
- c is a formal polynomial over the alphabet $w[0], w[1], \cdots, w[m]$

"**FreeRShift**: Right shift operation of a formal polynomial by a word"

Syntax FreeRShift(c, η::indexed) where

- c is a formal polynomial over the alphabet $w[0], w[1], \cdots, w[m]$
- η is a word

"**FreeChro**: Left chronological product of two formal polynomials"

Syntax FreeChro(c, d) where

- c, d are two formal polynomials over the alphabet $w[0], w[1], \cdots, w[m]$

"**FreeRChro**: Right chronological product of two formal polynomials"

Syntax FreeRChro(c, d) where

- c, d are two formal polynomials over the alphabet $w[0], w[1], \cdots, w[m]$

"**FreeLChro**: Left chronological product of two formal polynomials"

Syntax FreeLChro(c, d) where

- c, d are two formal polynomials over the alphabet $w[0], w[1], \cdots, w[m]$

"**FreeCompose**: Composition product of a formal polynomial with an array of formal polynomials"

Syntax FreeCompose(c, d::list) where

- c is a scalar formal polynomial over the alphabet $w[0], w[1], \cdots, w[m]$
- d is an array of formal polynomials of the form [*op 1*,*op 2*, $\cdots$, *op m*]

"**FreeComposeMIMO**: Composition product in MIMO case: composition of two arrays of formal polynomials "

Syntax FreeComposeMIMO(c::list,d::list) where

- c and d are both arrays of formal polynomials of the form [*op 1*,*op 2*, $\cdots$, *op m*]

"**FreeModCompose**: Modified composition product of a formal polynomial with an array of formal polynomials"

Syntax FreeModCompose($c, d ::$ list) where

- c is a scalar formal polynomial over the alphabet $w[0], w[1], \cdots, w[m]$
- d is a list of formal polynomials of the form [*op 1*,*op 2*, $\cdots$, *op m*]

"**FreeModComposeMIMO**: Modified composition product in the MIMO case"

Syntax FreeModComposeMIMO(c::list,d::list) where

- c and d are both arrays of formal polynomials of the form [*op 1*,*op 2*, $\cdots$, *op m*]

"**Fliess**: Output of a Fliess operator associated with a formal polynomial supplied with an input as an array of time domain functions $[u_1(t), \cdots, u_m(t)]$";

Syntax Fliess(c, $u ::$ list) where

- c is a scalar formal polynomial over the alphabet $w[0], w[1], \cdots, w[m]$
- u is an array of time domain functions of the form $[u_1(t), u_2(t), \cdots, u_m(t)]$

"**FliessMIMO**: Output of a Fliess operator associated with a generating polynomial with input as an array of time domain functions $[u_1(t), \cdots, u_m(t)]$ in the MIMO case";

Syntax FliessMIMO($c ::$ list, $u ::$ list) where

- c is an array of formal polynomials over the alphabet $w[0], w[1], \cdots, w[m]$
- u is an array of time domain functions of the form $[u_1(t), u_2(t), \cdots, u_m(t)]$

A.3 Source Code for Operations on Formal Polynomials

```
>
> with(FREE):
> with(combinat):
> ## The FREE (ACE) and COMBINATORICS algebraic package over the free monoid
>
> ###########################
> ###  FreeLength(eta::indexed)
> ###########################
> FreeLength := proc(eta::indexed)
> description "Length of a word in the free monoid";
>    local length;
>    length:= nops(eta);
>
>  return length;
>
> end proc:
>
> ##########################
> ###  FreeDegree(c)
> ##########################
> FreeDegree := proc(c)
>  description "Degree of a polynomial in the free monoid";
>  local degree, tmpdeg, i;
>
>  degree:=0;
>
>  if (c=0) then
>      return degree;
>  end if;
>
>  if type(c,indexed) then
>       return FreeLength(c);
>  else
>     if type(c,'+') then
>          for i to nops(c) do
>                   if type(op(i,c),indexed) then
>                    tmpdeg:=FreeLength(op(i,c));
>                        if degree< tmpdeg  then
>                         degree:=tmpdeg;
>                    end if;
>              else
>                    if type(op(i,c),'*') then
>                        tmpdeg:=FreeLength(op(2,op(i,c)));
>                            if degree< tmpdeg  then
>                             degree:=tmpdeg;
>                        end if;
>                        end if
>               end if;
>         end do;
>         return degree;
>     end if;
>  end if;
>
>  if type(c,'*') then
>       degree :=FreeLength(op(2,c));
>       return degree;
>  end if;
>
> end proc:
>
> ########################
> ###  FreeOrder(c)
> ########################
> FreeOrder  := proc(c)
>  description "Order of a polynomial/series in the free monoid";
>  local order, tmpord, i;
```

```
>
> order:=infinity;
>
> if (c=0) then
>     return order;
> end if;
>
> if type(c,indexed) then
>        return FreeLength(c);
> else
>     if type(c,'+') then
>           for i to nops(c) do
>                if type(op(i,c),indexed) then
>                     tmpord:=FreeLength(op(i,c));
>                           if order > tmpord  then
>                            order :=tmpord;
>                       end if;
>                else
>                      if type(op(i,c),'*') then
>                           tmpord:=FreeLength(op(2,op(i,c)));
>                               if order > tmpord  then
>                                   order:=tmpord;
>                               end if;
>                      end if;
>                end if;
>          end do;
>          return order;
>     end if;
> end if;
>
> if type(c,'*') then
>       order :=FreeLength(op(2,c));
>       return order;
> end if;
>
> end proc:
>
> ########################
> ###  FreeDist(c,d, sigma)
> ########################
> FreeDist  := proc(c, d, sigma)
>  description "The distance between two formal polynomials in the ultrametric sense";
>  local dist;
>       dist:=sigma^(FreeOrder(c-d));
>    return dist;
>
> end proc:
>
> ############################
> ###  FreeLShiftWord(eta::indexed, xi::indexed)
> ############################
> FreeLShiftWord := proc(eta::indexed, xi::indexed)
>  description "Left shift operation of a word by another word";
>
>  local i, tmplshiftword;
>  tmplshiftword := 0;
>
>     if nops(eta) > nops(xi) then
>       return 0;
>     else
>        for i from 1 to nops(eta) do
>            if  op(i, eta) <> op(i, xi) then
>                return 0;
>            else
>                tmplshiftword := w[op(i+1..nops(xi), xi)];
>            end if;
>
>        end do;
```

```
>         return tmplshiftword;
>      end if;
>  end proc:
>
> ############################
> ###  FreeLshift(eta::indexed, c)
> ############################
> FreeLShift := proc(eta::indexed, c)
>  description "Left shift operation of a formal polynomial by a word";
>  local i, tmplshift;
>  tmplshift := 0;
>
>     if type(c,indexed) then
>            return FreeLShiftWord(eta, c) ;
>     else
>
>           if type(c,'*') then
>               return op(1,c)*FreeLShiftWord(eta, op(2,c));
>           else
>                if type(c,'+') then
>                       for i to nops(c) do
>                              if type(op(i,c),indexed) then
>                                tmplshift := tmplshift +
                                    FreeLShiftWord(eta, op(i,c)) ;
>                          else
>                             if type(op(i,c),'*') then
>                               tmplshift := tmplshift+op(1,op(i,c))
                                 *FreeLShiftWord(eta,op(2,op(i,c)));
>                             end if;
>                          end if;
>                      end do;
>                end if;
>                return tmplshift;
>           end if;
>
>     end if;
>
> end proc:
>
> ############################
> ###  FreeRshiftWord(xi::indexed, eta::indexed)
> ############################
> FreeRShiftWord := proc(xi::indexed, eta::indexed)
>  description "Right shift operation of a word by another word";
>
>  local i, tmprshiftword;
>     if nops(xi) < nops(eta) then
>        return 0;
>     else
>         for i from 1 to nops(eta) do
>             if  op(nops(xi)+1-i, xi) <> op(nops(eta)+1-i, eta) then
>                 return 0;
>             else
>                 tmprshiftword := w[op(1..(nops(xi)-i), xi)];
>             end if;
>
>         end do;
>         return tmprshiftword;
>      end if;
>  end proc:
>
> ############################
> ###  FreeRshift(c, eta::indexed)
> ############################
> FreeRShift := proc(c, eta::indexed)
>  description "Right shift operation of a formal polynomial by a word";
>  local i, tmprshift;
>  tmprshift := 0;
```

```
>
>     if type(c,indexed) then
>           return FreeRShiftWord(c, eta );
>     else
>
>          if type(c,'*') then
>              return op(1,c)*FreeRShiftWord(op(2,c), eta);
>          else
>               if type(c,'+') then
>                      for i to nops(c) do
>                             if type(op(i,c),indexed) then
>                               tmprshift := tmprshift +
>                                      FreeRShiftWord(op(i,c),eta) ;
>                         else
>                            if type(op(i,c),'*') then
>                               tmprshift := tmprshift+op(1,op(i,c))
>                                  *FreeRShiftWord(op(2,op(i,c)),eta);
>                            end if;
>                         end if;
>                     end do;
>               end if;
>               return tmprshift;
>          end if;
>
>     end if;
>
> end proc:
>
> ########################
> ###  FreeRChroWord(eta::indexed,xi::indexed)
> ########################
> FreeRChroWord := proc (eta::indexed, xi::indexed) option remember;
> description "Right chronological product of two words";
> local i, prefix, affix, result, xiprime, tmp;
>
>     if nops(eta) = 0 then
>         return xi
>     end if;
>     if nops(xi) = 0 then
>         return 0
>     else
>         i := 1;
>         for i to nops(xi) do
>             xiprime := w[op(1 .. nops(xi)-i,xi)];
>             affix := w[op(nops(xi),xi)];
>             tmp := FreeRChroWord(eta,xiprime)+FreeRChroWord(xiprime,eta);
>             result := FreeConcat(tmp,affix);
>            return result;
>         end do
>     end if
>
> end proc:
>
> ########################
> ###  FreeLChroWord(eta::indexed,xi::indexed)
> ########################
> FreeLChroWord := proc (eta::indexed, xi::indexed)
> local i, prefix, affix, result, xiprime, tmp;
> option remember;
> description "Left chronological product of two words";
>
>     if nops(eta) = 0 then
>          return 0
>     end if;
>
>     if nops(xi) = 0 then
>          return eta
>     else
```

```
>         i := 1;
>         for i to nops(xi) do
>              xiprime := w[op(i+1 .. nops(xi),xi)];
>              prefix := w[op(1,xi)];
>              tmp := FreeLChroWord(eta,xiprime)+FreeLChroWord(xiprime,eta);
>              result := FreeConcat(prefix,tmp);
>             return result;
>         end do
>     end if
>
> end proc:
>
> ########################
> ###  FreeChroWord(eta::indexed,xi::indexed)
> ########################
> FreeChroWord := proc (eta::indexed, xi::indexed)
> option remember;
> description "Left chronological product of two words";
>
>    printf(" ***   Left Chronological product by default. \n");
>    printf(" ***   For right Chronological product, use FreeRChroWord(eta::indexed, xi::indexed)");
>
>    return FreeLChroWord(eta, xi)
>
> end proc:
>
> ########################
> ###  FreeRChroWord2Poly(eta::indexed, d)
> ########################
> FreeRChroWord2Poly := proc (eta::indexed, d)
> option remember;
> description "Right chronological product of a word with a polynomial";
> local i, chroprod;
>    chroprod := 0;
>
>    if type(d,indexed) then return FreeRChroWord(eta,d);
>    else
>        if type(d,'+') then
>             for i to nops(d) do
>                  if type(op(i,d),indexed) then
>                        chroprod := chroprod+FreeRChroWord(eta, op(i,d))
>                  else
>                        if type(op(i,d),'*') then
>                             chroprod := chroprod+op(1,op(i,d))*FreeRChroWord(eta, op(2,op(i,d)))
>                        end if;
>                  end if;
>             end do;
>             return(chroprod);
>        end if;
>    end if;
>
>    if type(d,'*') then
>       chroprod:=op(1,d)*FreeRChroWord(eta, op(2,d));
>       return chroprod;
>    end if;
>
> end proc:
>
> ########################
> ###  FreeLChroWord2Poly(eta::indexed, d)
> ########################
> FreeLChroWord2Poly := proc (eta::indexed, d)
> option remember;
> description "Left chronological product of a word and a polynomial";
> local i, chroprod;
>    chroprod := 0;
>
>    if type(d,indexed) then return FreeLChroWord(eta,d);
```

```
>    else
>        if type(d,'+') then
>            for i to nops(d) do
>                if type(op(i,d),indexed) then
>                    chroprod := chroprod+FreeLChroWord(eta, op(i,d))
>                else
>                    if type(op(i,d),'*') then
>                        chroprod := chroprod+op(1,op(i,d))*FreeLChroWord(eta, op(2,op(i,d)))
>                    end if;
>                end if;
>            end do;
>            return(chroprod);
>        end if;
>    end if;
>
>    if type(d,'*') then
>       chroprod:=op(1,d)*FreeLChroWord(eta, op(2,d));
>       return chroprod;
>    end if;
>
> end proc:
>
> ########################
> ###  FreeChroWord2Poly(eta::indexed, d)
> ########################
> FreeChroWord2Poly:= proc (eta::indexed, d)
> option remember;
> description "Left chronological product of a word and a polynomial";
>
>    printf(" ***   Left Chronological product by default. \n");
>    printf(" ***   For the Right Chronological product, use FreeRChroWord2Poly(c, d)");
>
>    return FreeLChroWord2Poly(eta, d)
>
> end proc:
>
> ########################
> ###  FreeRChro(c, d)
> ########################
> FreeRChro := proc (c, d)
> option remember;
> description "Right chronological product of two polynomials";
> local i, chroprod;
>    chroprod := 0;
>
>    if type(c,indexed) then return FreeRChroWord2Poly(c,d);
>    else
>        if type(c,'+') then
>            for i to nops(c) do
>                if type(op(i,c),indexed) then
>                    chroprod := chroprod+FreeRChroWord2Poly(op(i,c),d)
>                else
>                    if type(op(i,c),'*') then
>                        chroprod := chroprod+op(1,op(i,c))*FreeRChroWord2Poly(op(2,op(i,c)),d)
>                    end if;
>                end if;
>            end do;
>            return(chroprod);
>        end if;
>    end if;
>
>    if type(c,'*') then
>       chroprod:=op(1,c)*FreeRChroWord2Poly(op(2,c),d);
>       return chroprod;
>    end if;
>
> end proc:
>
```

```
> ########################
> ###  FreeLChro(c, d)
> ########################
> FreeLChro := proc (c, d)
> option remember;
> description "Left chronological product of two polynomials";
> local i, chroprod;
>    chroprod := 0;
>
>    if type(c,indexed) then return FreeLChroWord2Poly(c,d);
>    else
>        if type(c,'+') then
>            for i to nops(c) do
>                if type(op(i,c),indexed) then
>                      chroprod := chroprod+FreeLChroWord2Poly(op(i,c),d)
>                else
>                      if type(op(i,c),'*') then
>                           chroprod := chroprod+op(1,op(i,c))*FreeLChroWord2Poly(op(2,op(i,c)),d)
>                      end if;
>                end if;
>            end do;
>            return(chroprod);
>        end if;
>    end if;
>
>    if type(c,'*') then
>       chroprod:=op(1,c)*FreeLChroWord2Poly(op(2,c),d);
>       return chroprod;
>    end if;
>
> end proc:
>
> ########################
> ###  FreeChro(c, d)
> ########################
> FreeChro := proc (c, d)
> option remember;
> description "Left chronological product of two polynomials";
>
>    printf(" ***   Left Chronological product by default. \n");
>    printf(" ***   For the Right Chronological product, use FreeRChro(c, d)");
>
>    return FreeLChro(c, d)
>
> end proc:
>
> ########################
> ###  FreeComposeWord2Poly(eta::indexed,d::list)
> ########################
> FreeComposeWord2Poly := proc(eta::indexed,d::list)
> option remember;
> local i, prefix, result, etaprime, tmp;
> description "Composition product of a word with a formal polynomial";
>
>    if nops(eta) = 0 then return eta; end if;
>    for i from 1 to nops(eta) do
>        if op(i,eta) = 0 then result:=eta;
>        else
>            prefix:=FreeConcat(w[0],w[op(1..(i-1),eta)]);
>            etaprime:= w[op(i+1..nops(eta), eta)];
>            tmp:=FreeShuffle(op(op(i,eta),d),
                                  FreeComposeWord2Poly(etaprime,d));
>            result:=FreeConcat(prefix,tmp);
>        return result;
>        end if;
>    end do;
>
> end proc:
```

```
>
> ########################
> ###  FreeCompose(c,d::list)
> ########################
> FreeCompose := proc (c, d::list)
> option remember;
> description "Composition product of a scalar formal polynomial and
>           an array of formal polynomials in the free algebra";
> local comprod, i;
>
>    comprod := 0;
>
>    if type(c,indexed) then return FreeComposeWord2Poly(c,d);
>    else
>        if type(c,'+') then
>             for i to nops(c) do
>               if type(op(i,c),indexed) then
>                 comprod := comprod+FreeComposeWord2Poly(op(i,c),d);
>               else
>                   if type(op(i,c),'*') then
>                      comprod := comprod+op(1,op(i,c))*
                                FreeComposeWord2Poly(op(2,op(i,c)),d);
>                   end if;
>               end if;
>             end do;
>             return(comprod);
>        end if;
>    end if;
>
>    if type(c,'*') then
>       comprod:=op(1,c)*FreeComposeWord2Poly(op(2,c),d);
>       return comprod;
>    end if;
>
> end proc:
>
> ########################
> ###  FreeComposeMIMO(c::list, d::list)
> ########################
> FreeComposeMIMO := proc (c::list, d::list)
> # option remember;
> description "Composition product for MIMO case";
> local tmp, i, dim;  dim:=nops(c);
>     tmp := [seq(0^i,i=1..dim)];
>     for i from 1 to nops(c) do
>         tmp[i] := FreeCompose(op(i,c), d);
>     end do;
>  return tmp;
>
> end proc:
>
> ########################
> ###  FreeModComposeWord2Poly(eta::indexed,d::list)
> ########################
> FreeModComposeWord2Poly := proc(eta::indexed,d::list)
> option remember;
> description "Modified composition product of a word with an array of formal polynomials";
> local i, prefix1, prefix2, result, etaprime, tmp;
>
>    if nops(eta) = 0 then return eta; end if;
>    for i from 1 to nops(eta) do
>        if op(i,eta) = 0 then
>            result:=eta;
>        else
>            prefix1:=w[op(1..i,eta)];
>            prefix2:=FreeConcat(w[0],w[op(1..(i-1),eta)]);
>            etaprime:= w[op(i+1..nops(eta), eta)];
>            tmp:=FreeShuffle(op(op(i,eta),d),
```

```
                            FreeModComposeWord2Poly(etaprime,d));
>           result:=FreeConcat(prefix1,FreeModComposeWord2Poly
                            (etaprime,d))+FreeConcat(prefix2,tmp);
>           return result;
>        end if;
>    end do;
>
> end proc:
>
> ########################
> ###  FreeModCompose(c,d::list)
> ########################
> FreeModCompose := proc (c, d::list)
> option remember;
> description "Modified composition product of two polynomials in the free algebra";
> local comprod, i; comprod := 0;
>
>     if type(c,indexed) then
>        return FreeModComposeWord2Poly(c,d);
>     else
>        if type(c,'*') then
>           comprod:=op(1,c)*FreeModComposeWord2Poly(op(2,c),d);
>           return comprod;
>        end if;
>
>        if type(c,'+') then
>            for i to nops(c) do
>                if type (op(i,c),indexed) then
>                    comprod := comprod+FreeModComposeWord2Poly(op(i,c),d);
>                else
>                    if type(op(i,c),'*') then
>                       comprod := comprod+op(1,op(i,c))*
                             FreeModComposeWord2Poly(op(2,op(i,c)),d);
>                    end if;
>                end if;
>            end do;
>            return(comprod);
>        end if;
>     end if;
>
> end proc:
>
> ########################
> ###  FreeModComposeMIMO(c::list, d::list)
> ########################
> FreeModComposeMIMO := proc (c::list, d::list)
> # option remember;
> description "Modified Composition product for MIMO case";
> local tmp, i, dim;  dim:=nops(c);
>     tmp := [seq(0^i,i=1..dim)];
>     for i from 1 to nops(c) do
>         tmp[i] := FreeModCompose(op(i,c), d);
>     end do;
>  return tmp;
>
> end proc:
>
```

A.4 Source Code to Calculate Output Response of a Fliess Operator

```
> ########################
> ###  FliessWord(eta::indexed,u::list)
> ########################
> FliessWord := proc(eta::indexed, u::list)
> option remember;
> description "Output of a Fliess operator associated with a generating word
            applied to a time domain input vector u(t)";
>
> local i, etaprime, tmp, result;
>
>     if nops(eta) = 0 then
>         return 1;
>     end if;
>     for i from 1 to nops(eta) do
>            etaprime:= w[op(i+1..nops(eta), eta)];
>      if op(i,eta) = 0 then
>              result:=int( FliessWord(etaprime,u), t) -
>                         eval(int( FliessWord(etaprime,u), t),t=0);
>              return result;
>      else
>         result:=int( u[op(i,eta)]*FliessWord(etaprime,u),t)-
>           eval(int(u[op(i,eta)]*FliessWord(etaprime,u), t),t=0);
>         return result;
>      end if;
>     end do;
>
> end proc:
>
########################
###  Fliess(c, u::list)
########################
Fliess := proc (c, u::list)
>  option remember;
> description "Output of a Fliess operator associated with a generating polynomial
              supplied with input as a time domain function vector u(t)";
>
> local tmpsum, i; tmpsum := 0;
>
>     if type(c,indexed) then
>         return FliessWord(c,u);
>     else
>          if type(c,'*') then
>               tmpsum:=op(1,c)*FliessWord(op(2,c),u);
>               return tmpsum;
>          end if;
>
>          if type(c,'+') then
>               for i to nops(c) do
>                    if type(op(i,c),indexed) then
>                          tmpsum := tmpsum+FliessWord(op(i,c),u)
>                    else
>                         if type(op(i,c),'*') then
>                               tmpsum := tmpsum+op(1,op(i,c)) *
>                                       FliessWord(op(2,op(i,c)),u);
>                         end if;
>                    end if;
>               end do;
>               return(tmpsum);
>          end if;
>     end if;
>
> end proc:
>
>
```

```
> ########################
> ###  FliessMIMO(c::list, u::list)
> ########################
> FliessMIMO := proc (c::list, u::list)
> description "Output of a Fliess operator in MIMO case";
>
> local tmp, i, dim;  dim:=nops(c); dim;
>     tmp := [seq(0^i,i=1..dim)]; tmp;
>     for i from 1 to nops(c) do
>         tmp[i] := Fliess(op(i,c), u);
>     end do;
>  return tmp;
>
> end proc:
>
```

CURRICULUM VITAE
YAQIN LI

DEGREES:

University of Science and Technology of China (USTC), Hefei, China

- Master of Engineering (Control Theory and Control Engineering), July 2001
- Bachelor of Science (Automatic Control), July 1998
- Bachelor of Science (Economics and Management Science), July 1998

ACADEMIC AWARDS:

- Best Paper of the Session Award, International Conference on Computer, Communication and Control Technologies (CCCT'03), Orlando, FL, 2003
- Outstanding Ph.D. Student Research Award, ECE Department, ODU, 2002
- Microsoft Research Institute President Fellowship, USTC, 2001
- Shing-Tung Yau Fellowship, USTC, 2001
- HUAWEI Scholarship, USTC, 2000
- HUAXIN Scholarship, USTC, 1996
- Star River Scholarship, USTC, 1995

PUBLICATIONS:

Conference Papers

1. Yaqin Li and W. Steven Gray, The Formal Laplace-Borel Transform, Fliess Operators and the Composition Product, Proc. 36th IEEE Southeastern Symposium on System Theory (SSST'04), Atlanta, Georgia, U.S.A., March 14-16, 2004, pp. 333–337.
2. Yaqin Li, Min Song and W. Steven Gray, A Self-Adaptive Predictive Congestion Control Model for Extreme Networks, Proc. 2003 International Conference on Computer, Communication and Control Technologies (CCCT'03), Orlando, Florida, U.S.A., July 31–August 2, 2003, pp.327-332.
3. W. Steven Gray and Yaqin Li, Generating Series for Nonlinear Cascade and Feedback Systems, Proc. 41st IEEE Conference on Decision and Control (CDC'02), Las Vegas, Nevada, U.S.A., December 10-13, 2002, pp. 2720-2725.
4. W. Steven Gray and Yaqin Li, Fliess Operators in Cascade and Feedback Systems, Proc. 2002 Conference on Information Sciences and Systems (CISS'02), Princeton, New Jersey, U.S.A, March 20-22, 2002, pp. 173-178.
5. W. Steven Gray and Yaqin Li, Interconnected Systems of Fliess Operators, 2002 International Symposium on the Mathematical Theory of Networks and Systems (MTNS'02), South Bend, Indiana, U.S.A., August 12-16, 2002.

Journal Papers

1. W. Steven Gray and Yaqin Li, Generating Series for Interconnected Analytic Nonlinear Systems, SIAM Journal on Control and Optimization, to appear.

2. Sifu Li, Zifu Xu and Yaqin Li, A Predictive Control with Wiener Model Nonlinear System Based on the Laguerre-RBF Combined Model (in Chinese), Information and Control, vol. 30, no. 4, August 2001, pp. 305-308.

3. Sifu Li, Yaqin Li and Zifu Xu, An Extension to Laguerre Model Adaptive Predictive Control Algorithm (in Chinese), Journal of USTC, vol. 31, no. 1, January 2001, pp. 92-98.

4. Sifu Li, Jun Qian and Yaqin Li, The Predictive Control Based on Laguerre Function for a Class of Nonlinear Systems (in Chinese), Journal of USTC, vol. 30, no. 5, October 2000, pp. 548-553.