Automatica 119 (2020) 109085

Contents lists available at ScienceDirect

Automatica

journal homepage: www.elsevier.com/locate/automatica

Brief paper

Nonlinear system identification for multivariable control via discrete-time Chen–Fliess series☆

W. Steven Gray [a,*], G.S. Venkatesh [a], Luis A. Duffaut Espinosa [b]

[a] Department of Electrical and Computer Engineering, Old Dominion University, Norfolk, VA 23529, USA
[b] Department of Electrical and Biomedical Engineering, University of Vermont, Burlington, VT 05405, USA

ARTICLE INFO

Article history:
Received 21 June 2019
Received in revised form 11 January 2020
Accepted 3 May 2020
Available online 1 July 2020

Keywords:
Nonlinear system identification
Chen–Fliess series
Adaptive control

ABSTRACT

The nonlinear system identification problem is solved for a multivariable nonlinear input–output system that can be represented in terms of a Chen–Fliess functional expansion. The problem is formulated in terms of a regression which is linear in the parameters and has a nonlinear regressor. An inductive implementation of the nonlinear regressor is developed using the underlying noncommutative algebraic and combinatorial structures. The method is demonstrated in an adaptive control application involving a two-input, two-output Lotka–Volterra dynamical system.

1. Introduction

The system identification problem for nonlinear dynamical systems is a vast subject intersecting many different disciplines as described in the survey papers by Ljung (2010) and Schoukens & Ljung (2019). The *best* approach and solution are highly dependent on the particular nature of the application. In this paper, the application is adaptive control of nonlinear input–output systems that can be represented in terms of *Chen–Fliess functional series* or *Fliess operators* (Fliess, 1981, 1983; Isidori, 1995). For example, any analytic control affine state space system in continuous-time has an input–output map with a Fliess operator representation. Its form is independent of any coordinate system and is uniquely specified by the coefficients of a noncommutative formal power series. Perhaps the most closely related existing methods in the literature are those related to the identification of Volterra series, for example, Batselier et al. (2017a, 2017b), Birpoutsoukis et al. (2017), Doyle III et al. (2002), Hu et al. (2019) and Pillonetto et al. (2014). (See Schoukens & Ljung, 2019 for a more extensive list.) This is due to the fact that any continuous-time Volterra series with analytic kernels can be written in terms of a Fliess operator (Fliess, 1981). The primary advantage of the proposed method is that the identification problem for kernel functions is replaced with a much simpler linear parametric identification problem. In addition, the approach can easily accommodate terms from kernel functions beyond the second order, which is the standard in most of the existing literature. The method described in Padoan & Astolfi (2016) should also be mentioned as it employs formal power series concepts, but it is restricted to the autonomous case.

As most implementations are in discrete-time, the specific model class will be the set of discrete-time Fliess operators, which are known to be capable of approximating their continuous-time counterparts to arbitrary precision (Gray et al., 2017a). The problem is formulated in terms of a nonlinear regression which is linear in the parameters. Thus, the parameter estimation problem is solved by a standard recursive least-squares algorithm. What makes the problem nontrivial is finding an inductive implementation of the nonlinear regressor. For single-input systems, the problem can be bypassed by applying brute force methods (Gray et al., 2017b, 2019b). But in the multivariable setting addressed here, this is no longer feasible as the complexity becomes unmanageable. Instead, it is necessary to exploit the underlying noncommutative algebraic and combinatorial structures to produce a suitable induction. Finally, the method is demonstrated using a two-input, two-output Lotka–Volterra dynamical system in a deterministic setting. Here the identified Chen–Fliess series is employed in an indirect adaptive control scheme where the

☆ This research was supported by the National Science Foundation under grants CMMI-1839378 and CMMI-1839387. The material in this paper was partially presented at: the 21st International Conference on System Theory, Control and Computing (ICSTCC 2017), October 19-21, 2017, Sinaia, Romania. The 2019 American Control Conference (ACC), July 10-12, 2019, Philadelphia, PA, USA. The 53rd Annual Conference on Information Sciences and Systems (CISS 2019), March 20-22, 2019, Baltimore, MD, USA. The 58th IEEE Conference on Decision and Control, December 11-13, 2019, Nice, France. This paper was recommended for publication in revised form by Associate Editor Kyriakos G. Vamvoudakis under the direction of Editor Miroslav Krstic.

* Corresponding author.
E-mail addresses: sgray@odu.edu (W.S. Gray), gsvenky89@gmail.com (G.S. Venkatesh), lduffaut@uvm.edu (L.A. Duffaut Espinosa).

https://doi.org/10.1016/j.automatica.2020.109085
0005-1098/

control signal is computed using one-step ahead prediction. Some of these results have appeared in preliminary form in Gray et al. (2019a) and Venkatesh et al. (2019).

The paper is organized as follows. In the next section, a brief summary of the key concepts concerning Chen–Fliess series is given. The main results are developed in Section 3. The simulation example is presented in the subsequent section. The conclusions and directions for future research are given in the last section.

2. Chen–Fliess series

In this section, a brief review of Chen–Fliess series (Fliess, 1981, 1983; Gray & Wang, 2002) and their discrete-time counterparts (Duffaut Espinosa et al., 2018; Gray et al., 2017a) is provided.

An *alphabet* $X = \{x_0, x_1, \ldots, x_m\}$ is any nonempty and finite set of noncommuting symbols referred to as *letters*. A *word* $\eta = x_{i_1} \cdots x_{i_k}$ is a finite sequence of letters from X. The number of letters in a word η, written as $|\eta|$, is called its *length*. The empty word, $\emptyset$, is taken to have length zero. The collection of all words having length k is denoted by X^k, and $X^* = \bigcup_{k\geq 0} X^k$. The latter is a monoid under the concatenation product. Any mapping $c : X^* \to \mathbb{R}^\ell$ is called a *formal power series*. Often c is written as the formal sum $c = \sum_{\eta\in X^*}(c,\eta)\eta$, where the *coefficient* (c,η) is the image of $\eta \in X^*$ under c. The set of all noncommutative formal power series over the alphabet X is denoted by $\mathbb{R}^\ell\langle\langle X\rangle\rangle$. It forms an associative $\mathbb{R}$-algebra under the Cauchy product.

2.1. Continuous-time case

Given any $c \in \mathbb{R}^\ell\langle\langle X\rangle\rangle$ one can associate a causal m-input, ℓ-output operator, F_c, in the following manner. Let $\mathfrak{p} \geq 1$ and $t_0 < t_1$ be given. For a Lebesgue measurable function $u : [t_0, t_1] \to \mathbb{R}^m$, define $\|u\|_\mathfrak{p} = \max\{\|u_i\|_\mathfrak{p} : 1 \leq i \leq m\}$, where $\|u_i\|_\mathfrak{p}$ is the usual $L_\mathfrak{p}$-norm for a measurable real-valued function, u_i, defined on $[t_0, t_1]$. Let $L^m_\mathfrak{p}[t_0, t_1]$ denote the set of all measurable functions defined on $[t_0, t_1]$ having a finite $\|\cdot\|_\mathfrak{p}$ norm and $B^m_\mathfrak{p}(R)[t_0, t_1] := \{u \in L^m_\mathfrak{p}[t_0, t_1] : \|u\|_\mathfrak{p} \leq R\}$. Assume $C[t_0, t_1]$ is the subset of continuous functions in $L^m_1[t_0, t_1]$. Define inductively for each $\eta \in X^*$ the map $E_\eta : L^m_1[t_0, t_1] \to C[t_0, t_1]$ by setting $E_\emptyset[u] = 1$ and letting

$$E_{x_i\bar{\eta}}[u](t, t_0) = \int_{t_0}^{t} u_i(\tau)E_{\bar{\eta}}[u](\tau, t_0)\,d\tau,$$

where $x_i \in X$, $\bar{\eta} \in X^*$, and $u_0 = 1$. The *Chen–Fliess series* corresponding to c is

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

(Fliess, 1981, 1983). It can be shown that if there exist real numbers $K, M \geq 0$ such that

$$|(c,\eta)| \leq KM^{|\eta|}\,|\eta|!, \quad \forall\eta \in X^*$$

($|z| := \max_i |z_i|$ when $z \in \mathbb{R}^\ell$) then the series defining F_c converges absolutely and uniformly for sufficiently small $R, T > 0$ and constitutes a well defined mapping from $B^m_\mathfrak{p}(R)[t_0, t_0+T]$ into $B^\ell_\mathfrak{q}(S)[t_0,\ t_0+T]$, where the numbers $\mathfrak{p}, \mathfrak{q} \in [1, +\infty]$ are conjugate exponents, i.e., $1/\mathfrak{p} + 1/\mathfrak{q} = 1$ (Gray & Wang, 2002). Any such mapping is called a *Fliess operator*. A Fliess operator F_c defined on $B^m_\mathfrak{p}(R)[t_0, t_0 + T]$ is said to be *realizable* when there exists a state space model

$$\dot{z}(t) = g_0(z(t)) + \sum_{i=1}^{m} g_i(z(t))\,u_i(t), \quad z(t_0) = z_0 \tag{1a}$$

$$y_j(t) = h_j(z(t)), \quad j = 1, 2, \ldots, \ell, \tag{1b}$$

where each g_i is an analytic vector field expressed in local coordinates on some neighborhood $\mathcal{W}$ of z_0, and each output function h_j is an analytic function on $\mathcal{W}$ such that (1a) has a well defined solution $z(t)$, $t \in [t_0, t_0+T]$ for any given input $u \in B^m_\mathfrak{p}(R)[t_0, t_0 + T]$, and $y_j(t) = F_{c_j}[u](t) = h_j(z(t))$, $t \in [t_0, t_0 + T]$, $j = 1, 2, \ldots, \ell$. It can be shown that for any word $\eta = x_{i_k} \cdots x_{i_1} \in X^*$

$$(c_j, \eta) = L_{g_\eta} h_j(z_0) := L_{g_{i_1}} \cdots L_{g_{i_k}} h_j(z_0), \tag{2}$$

where $L_{g_i} h_j$ is the *Lie derivative* of h_j with respect to g_i.

2.2. Discrete-time case

Inputs in the discrete-time setting are assumed to be sequences of vectors from the normed linear space

$$l^{m+1}_\infty(N_0) := \{\hat{u} = (\hat{u}(N_0), \hat{u}(N_0 + 1), \ldots) : \|\hat{u}\|_\infty < \infty\},$$

where $\hat{u}(N) := [\hat{u}_0(N), \hat{u}_1(N), \ldots, \hat{u}_m(N)]^T$, $N \geq N_0$ with $\hat{u}_i(N) \in \mathbb{R}$, $|\hat{u}(N)| := \max_{i\in\{0,1,\ldots,m\}} |\hat{u}_i(N)|$, and $\|\hat{u}\|_\infty := \sup_{N\geq N_0} |\hat{u}(N)|$. The subspace of finite sequences over $[N_0, N_f]$ is denoted by $l^{m+1}_\infty[N_0, N_f]$. Given a generating series $c \in \mathbb{R}^\ell\langle\langle X\rangle\rangle$, the corresponding *discrete-time Fliess operator* is defined as

$$\hat{F}_c[\hat{u}](N) = \sum_{\eta\in X^*}(c,\eta)S_\eta[\hat{u}](N)$$

for any $N \geq N_0$, where

$$S_{x_i\eta}[\hat{u}](N) = \sum_{k=1}^{N} \hat{u}_i(k)S_\eta[\hat{u}](k)$$

with $x_i \in X$, $\eta \in X^*$, and $\hat{u} \in l^{m+1}_\infty[N_0]$. By assumption, $S_\emptyset[\hat{u}](N) := 1$.

Following Grüne & Kloeden (2001), select some fixed $u \in L^m_1[0, T]$ with $T > 0$ finite. Choose an integer $L \geq 1$, let $\Delta := T/L$, and define the sequence of real numbers

$$\hat{u}_i(N) = \int_{(N-1)\Delta}^{N\Delta} u_i(t)\,dt, \quad i = 0, 1, \ldots, m,$$

where $N \in \{1, 2, \ldots, L\}$. Assume $u_0 = 1$ so that $\hat{u}_0(N) = \Delta$. Since numerically only finite sums can be computed, a truncated version of $\hat{F}_c$ will be useful,

$$\hat{y}(N) = \hat{F}^J_c[\hat{u}](N) := \sum_{\eta\in X^{\leq J}}(c,\eta)S_\eta[\hat{u}](N), \tag{3}$$

where $X^{\leq J} := \bigcup_{k=0}^{J} X^k$. The main assertion proved in Gray et al. (2017a) (Theorems 6 and 7) is that the class of truncated, discrete-time Fliess operators acts as a set of universal approximators with computable error bounds for their continuous-time counterparts. In which case, they can be used to approximate any input–output system corresponding to state space realization (1) with increasing accuracy as L and J increase.

3. System identification for discrete-time Fliess operators

The main objective of this section is to solve the system identification problem for the model class of truncated discrete-time Fliess operators. This is done one output channel at a time, so without loss of generality it is assumed that $\ell = 1$. First the problem is posed in terms of a nonlinear regression. Then an inductive implementation of the regressor is developed.

3.1. Nonlinear regression problem

The first step is to write (3) as the nonlinear regression

$$\hat{y}(N) = \phi^T(N)\theta_0, \quad N \geq 1, \tag{4}$$

where

$$\phi(N) = [S_{\eta_1}[\hat{u}](N)\, S_{\eta_2}[\hat{u}](N) \cdots S_{\eta_l}[\hat{u}](N)]^T$$
$$\theta_0 = [(c, \eta_1)\,(c, \eta_2) \cdots (c, \eta_l)]^T$$

with $l = \mathrm{card}(X^{\leq J}) = \sum_{k=0}^{J}(m+1)^k = ((m+1)^{J+1} - 1)/m$ and assuming some ordering $(\eta_1, \eta_2, \ldots)$ has been imposed on the words in X^*. If an estimate of θ_0 is available at time $N-1$, say $\hat{\theta}(N-1)$,[1] then (4) gives a corresponding prediction of $\hat{y}(N)$:

$$\hat{y}_p(N) := \phi^T(N)\hat{\theta}(N-1). \tag{5}$$

Since the parameters appear linearly in (5), the following recursive least-squares algorithm is used to update the series coefficients:

$$\hat{\theta}(N) = \hat{\theta}(N-1) + g(N-1)e(N) \tag{6a}$$
$$e(N) = y(N\Delta) - \phi^T(N)\hat{\theta}(N-1) \tag{6b}$$
$$g(N-1) = \frac{P(N-2)\phi(N)}{1 + \phi^T(N)P(N-2)\phi(N)} \tag{6c}$$
$$P(N-1) = P(N-2) - \frac{P(N-2)\phi(N)\phi^T(N)P(N-2)}{1 + \phi^T(N)P(N-2)\phi(N)} \tag{6d}$$

for any $N \geq 1$ with the initial estimate $\hat{\theta}(0)$ given, and $P(-1)$ being any positive definite matrix P_0 (Goodwin & Sin, 2009, p. 65). Covariance resetting is done periodically to enhance convergence. The corresponding one-step ahead predictor is shown in Fig. 1. Here input–output data (u, y) from some unknown continuous-time plant (or the error system between the plant and an assumed model) is fed into the unit. The only assumption is that the data came from a system which has a Fliess operator representation, for example, any system modeled by (1). In general, the predictor has no a priori knowledge of the system, so $\hat{\theta}(0)$ is initialized to zero. Setting $P_0 = I$, it is known that this algorithm minimizes the performance index

$$\mathbf{J}_{\bar{N}}(\theta) := \sum_{N=1}^{\bar{N}}[y(N\Delta) - \phi^T(N)\theta]^2 + \frac{1}{2}\left\|\theta - \hat{\theta}(0)\right\|^2$$

with respect to the parameter θ. It should be stated that since the model class consists of truncated versions of $\hat{F}_c$, there is no reason to expect the parameter vector $\hat{\theta}(N)$ to converge to c in any fashion as N increases. But this is not a problem since the only objective is to ensure that the underlying continuous-time input–output map F_c is well approximated by $\hat{F}^J_{\hat{\theta}(N)}$. On the other hand, the approximation theory presented in Gray et al. (2017a) guarantees that if the underlying system has such a Fliess operator representation then the true generating series c similarly truncated is a feasible limit point for the sequence $\hat{\theta}(N)$, $N \geq 0$. What is not so clear is how to efficiently compute the update of the regressor $\phi(N+1)$ given the next input–output pair $(u(N+1), y(N+1))$. This issue is addressed next.

[1] The hat notation is used throughout to distinguish discrete-time signals from continuous-time signals, for example, $\hat{y}(N) = y(N\Delta)$. In this one instance, however, it is also used to indicate an *estimate* of the parameter θ.

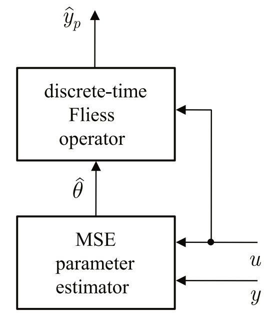


Fig. 1. One-step ahead predictor based on a discrete-time Fliess operator.

3.2. Inductive implementation of the regressor

To devise an inductive implementation of the regressor, it is necessary to identify the algebraic structure underlying the iterated sums in the definition of the discrete-time Fliess operator. The starting point for this is the following concept.

Definition 1 (*Gray et al., 2017a*). Given any $N \geq N_0$ and $\hat{u} \in l_\infty^{m+1}(N_0)$, a **discrete-time Chen series** is defined as

$$S[\hat{u}](N, N_0) = \sum_{\eta \in X^*} \eta S_\eta[\hat{u}](N, N_0),$$

where

$$S_{x_i\eta}[\hat{u}](N, N_0) = \sum_{k=N_0}^{N} \hat{u}_i(k) S_\eta[\hat{u}](k, N_0) \tag{7}$$

with $x_i \in X$, $\eta \in X^*$, and $S_\emptyset[\hat{u}](N, N_0) := 1$. If $N_0 = 0$ then $S[\hat{u}](N, 0)$ is abbreviated as $S[\hat{u}](N)$.

Given the bijection between the letters of $X = \{x_0, x_1 \ldots, x_m\}$ and components of $\hat{u} = [\hat{u}_0, \hat{u}_1, \ldots, \hat{u}_m]^T$, define the monomial $\hat{u}_\eta(N) = \hat{u}_{i_k}(N) \cdots \hat{u}_{i_1}(N)$ for any $\eta = x_{i_k} \cdots x_{i_1} \in X^*$ and $N \geq N_0$ with $\hat{u}_\emptyset(N) := 1$. Hence, for fixed any $\hat{u} \in l_\infty^{m+1}(N_0)$ and $N \geq N_0$, there is a corresponding formal power series in $\mathbb{R}\langle\langle X \rangle\rangle$ whose coefficients are given by these monomials, namely, $c_u(N) := \sum_{\eta \in X^*} \hat{u}_\eta(N)\eta$. Observe that

$$S_{x_i\eta}[\hat{u}](N_0, N_0) = \hat{u}_{x_i}(N_0) S_\eta[\hat{u}](N_0, N_0)$$

so that $S_\eta[\hat{u}](N_0, N_0) = \hat{u}_\eta(N_0)$, and thus, $S[\hat{u}](N_0, N_0) = c_u(N_0)$.

Example 2. If $X = \{x_1\}$ and $\hat{u}_{x_1}(N_0) = \hat{u}_1(N_0)$, then

$$S[\hat{u}](N_0, N_0) = \sum_{k=0}^{\infty}(\hat{u}_1(N_0)x_1)^k =: (1 - \hat{u}_1(N_0)x_1)^{-1}.$$

A key observation is that a discrete-time Chen series $S[\hat{u}](N, N_0)$ satisfies a difference equation as described next and proved in Appendix A.

Theorem 3 (*Gray et al., 2019a*). *For any $\hat{u} \in l_\infty^{m+1}(N_0)$ and $N \geq N_0$*

$$S[\hat{u}](N+1, N_0) = c_u(N+1) S[\hat{u}](N, N_0)$$

with $S[\hat{u}](N_0, N_0) = c_u(N_0)$ so that

$$S[\hat{u}](N, N_0) = \overleftarrow{\prod_{i=N_0}^{N}} c_u(i), \tag{8}$$

where $\overleftarrow{\prod}$ denotes a directed product from right to left.

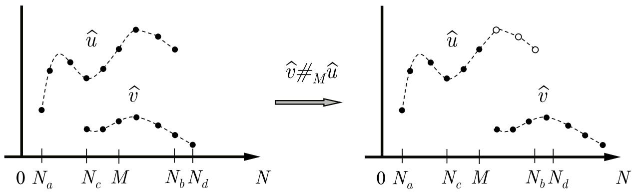


Fig. 2. Concatenation of inputs $\hat{u}$ and $\hat{v}$ at time $N = M$.

Example 4. Consider the case in Example 2 where $X = \{x_1\}$ and $\hat{u}_{x_1}(i) = \hat{u}_1(i)$ for all $i \geq N_0$. Then $c_u(i) = \sum_{k\geq 0}(\hat{u}_1(i)x_1)^k = (1-\hat{u}_1(i)x_1)^{-1}$ and

$$S[\hat{u}](N,N_0) = (1-\hat{u}_1(N)x_1)^{-1}\cdots(1-\hat{u}_1(N_0)x_1)^{-1}.$$

For instance,

$$\begin{aligned} S[\hat{u}](1,0) &= c_u(1)c_u(0) \\ &= 1 + (\hat{u}_1(1)+\hat{u}_1(0))x_1 + (\hat{u}_1^2(1)+ \\ &\quad \hat{u}_1(1)\hat{u}_1(0)+\hat{u}_1^2(0))x_1^2 + (\hat{u}_1^3(1)+ \\ &\quad \hat{u}_1^2(1)\hat{u}_1(0)+\hat{u}_1(1)\hat{u}_1^2(0)+\hat{u}_1^3(0))x_1^3 + \cdots \end{aligned}$$

In this case, $S[\hat{u}](N,N_0)$ is always a rational series (Berstel & Reutenauer, 1988).

Consider next two input sequences $(\hat{u},\hat{v}) \in l_\infty^{m+1}[N_a,N_b] \times l_\infty^{m+1}[N_c,N_d]$ with $N_b > N_a$ and $N_d > N_c$. The *concatenation* of $\hat{u}$ and $\hat{v}$ at $M \in [N_a,N_b]$ is taken to be

$$(\hat{v}\#_M\hat{u})(N) = \begin{cases} \hat{u}(N) & : \quad N_a \leq N \leq M \\ \hat{v}((N-M)+N_c) & : \quad M < N \leq M+(N_d-N_c) \end{cases}$$

as shown in Fig. 2. Define the set of sequences

$$l_{\infty,e}^{m+1}(0) := l_\infty^{m+1}(0) \cup \{\hat{\mathbf{0}}\},$$

where $\hat{\mathbf{0}}$ denotes the empty sequence with duration zero so that formally $\hat{v}\#_M\hat{\mathbf{0}} = \hat{\mathbf{0}}\#_M\hat{v} := \hat{v}$ for all $\hat{v} \in l_{\infty,e}^{m+1}(0)$. In which case, $l_{\infty,e}^{m+1}(0)$ is a monoid under this input concatenation operator. Define $S[\hat{\mathbf{0}}] = 1$. The following is a straightforward generalization of Theorem 3.

Theorem 5 (Gray et al., 2019a, *Discrete-Time Chen's Identity*). *Given $(\hat{u},\hat{v}) \in l_\infty^{m+1}[N_a,N_b] \times l_\infty^{m+1}[N_c,N_d]$, $M \in [N_a,N_b]$, and $N \in [M, M+(N_d-N_c)]$ it follows that*

$$S[\hat{v}]((N-M)+N_c,N_c)S[\hat{u}](M,N_a) = S[\hat{v}\#_M\hat{u}](N,N_a).$$

In particular, when $N_a = N_c = 0$ then

$$S[\hat{v}](N-M)S[\hat{u}](M) = S[\hat{v}\#_M\hat{u}](N). \tag{9}$$

Define the set of discrete-time Chen series

$$\mathcal{M}_C(X) = \{S[\hat{u}](N) \in \mathbb{R}\langle\langle X\rangle\rangle : \hat{u} \in l_\infty^{m+1}[0,N_f], \\ 0 \leq N \leq N_f < \infty\}.$$

The next theorem follows directly from (9).

Theorem 6 (Gray et al., 2019a). *$\mathcal{M}_C(X)$ is a monoid under the Cauchy product. In addition, $S : l_{\infty,e}^{m+1}(0) \to \mathcal{M}_C(X)$ is a monoid homomorphism.*

Let $\mathrm{End}(\mathbb{R}^\infty)$ be the set of endomorphisms on the $\mathbb{R}$-vector space of real right-sided infinite sequences. This set can be viewed as the monoid of doubly infinite matrices with well defined matrix products and unit $I = \mathrm{diag}(1,1,\ldots)$. A monoid M is said to have an *infinite dimensional real representation*, Π, if the mapping $\Pi : M \to \mathrm{End}(\mathbb{R}^\infty)$ is a monoid homomorphism. The representation is *faithful* if Π is injective.

Theorem 7 (Gray et al., 2019a). *The monoid $\mathcal{M}_C(X)$ has a faithful infinite dimensional real representation Π given by $\Pi(S[\hat{u}](N)) = \overleftarrow{\prod}_{i=0}^{N}\mathcal{S}(i)$, where $\mathcal{S}(i)$ is any matrix representation of the $\mathbb{R}$-linear map on $\mathbb{R}\langle\langle X\rangle\rangle$ given by the left concatenation map $\mathcal{C} : d \mapsto c_u(i)d$.*

Proof. The representation claim follows from (8). To see that Π is injective, assume a fixed ordering of the words in X^*, say $\{\eta_1,\eta_2,\ldots\}$. Then define the matrix $[\mathcal{S}(i)]_{jk} = (c_u(i)\eta_k,\eta_j) = \hat{u}_\xi(i)$, where $\xi\eta_k = \eta_j$. Thus, $\mathcal{S}(i)$ is a lower triangular matrix with ones along the diagonal since $u_\emptyset(i) = 1$, $i \geq 0$. The first column is comprised of the coefficients of $c_u(i)$ in the order given to X^*. Hence, the map Π on the monoid $\mathcal{M}_C(X)$ is injective since $c_u(i)$ can be uniquely identified from $\mathcal{S}(i) = \Pi(S[\hat{u}](i,i))$.

Note that the above theorem implies that (5) can be written in the form

$$\begin{aligned} \hat{y}_p(N+1) &= \hat{\theta}^T(N)\Pi(S[\hat{u}](N+1))e_1 \\ &= \hat{\theta}^T(N)\mathcal{S}(N+1)\Pi(S[\hat{u}](N))e_1 \end{aligned} \tag{10}$$

for $N \geq N_0$, where $\mathcal{S}(N+1)$ and $S[\hat{u}](N)$ have been suitable truncated, and $e_1 := [1\,0\,0\cdots 0]^T \in \mathbb{R}^l$. Equation (10) can also be written in the form $\hat{y}_p(N+1) = Q(\hat{u}(N+1))$, where Q is a polynomial in the components of $\hat{u}(N+1)$ with maximum degree $l-1$.

Example 8. Suppose $X = \{x_1\}$ as in Example 4. Assuming the ordering on X^* to be $\{\emptyset, x_1, x_1^2,\ldots\}$. Then for all $i \geq 0$

$$\mathcal{S}(i) = \begin{bmatrix} 1 & 0 & 0 & 0 & \cdots \\ \hat{u}_1(i) & 1 & 0 & 0 & \cdots \\ \hat{u}_1^2(i) & \hat{u}_1(i) & 1 & 0 & \cdots \\ \hat{u}_1^3(i) & \hat{u}_1^2(i) & \hat{u}_1(i) & 1 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}$$

and $c_u(i) = \sum_{k\geq 0}\hat{u}_1^k(i)x_1^k$. In addition,

$$\begin{aligned} &\Pi(S[\hat{u}](1)) \\ &= \mathcal{S}(1)\mathcal{S}(0) \\ &= \left[\begin{matrix} 1 \\ \hat{u}_1(1)+\hat{u}_1(0) \\ \hat{u}_1^2(1)+\hat{u}_1(1)\hat{u}_1(0)+\hat{u}_1^2(0) \\ \hat{u}_1^3(1)+\hat{u}_1^2(1)\hat{u}_1(0)+\hat{u}_1(1)\hat{u}_1^2(0)+\hat{u}_1^3(0) \\ \vdots \end{matrix}\right. \\ &\qquad \left.\begin{matrix} 0 & 0 & 0 & \cdots \\ 1 & 0 & 0 & \cdots \\ \hat{u}_1(1)+\hat{u}_1(0) & 1 & 0 & \cdots \\ \hat{u}_1^2(1)+\hat{u}_1(1)\hat{u}_1(0)+\hat{u}_1^2(0) & \hat{u}_1(1)+\hat{u}_1(0) & 1 & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{matrix}\right]. \end{aligned}$$

As expected, the first column coincides with the coefficients of $S[\hat{u}](1)$ in Example 4. Setting $J = 3$ so that $l = \mathrm{card}(X^{\leq J}) = 4$ gives the truncated versions

$$\hat{\theta}^T(N) = \begin{bmatrix} (c,\emptyset) & (c,x_1) & (c,x_1^2) & (c,x_1^3) \end{bmatrix} \tag{11a}$$

$$\mathcal{S}(N) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ \hat{u}_1(N) & 1 & 0 & 0 \\ \hat{u}_1^2(N) & \hat{u}_1(N) & 1 & 0 \\ \hat{u}_1^3(N) & \hat{u}_1^2(N) & \hat{u}_1(N) & 1 \end{bmatrix} \tag{11b}$$

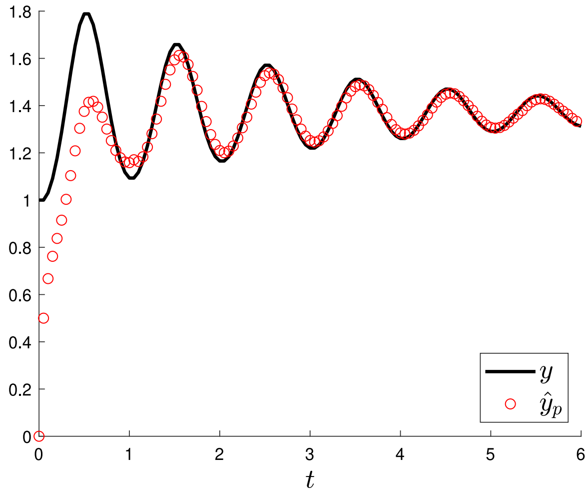


Fig. 3. Predictor output $\hat{y}_p$ versus true output y in Example 8.

$$\Pi(S[\hat{u}](N)) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ S_{x_1}(N) & 1 & 0 & 0 \\ S_{x_1^2}(N) & S_{x_1}(N) & 1 & 0 \\ S_{x_1^3}(N) & S_{x_1^2}(N) & S_{x_1}(N) & 1 \end{bmatrix}, \quad (11c)$$

where $S_{x_1^k}(N) := (S[\hat{u}](N), x_1^k)$. Therefore, the output

$$\hat{y}_p(N+1) = Q(\hat{u}(N+1)) = \sum_{i=0}^{3} q_i(N)\hat{u}_1^i(N+1),$$

where the coefficients $q_i(N)$ are functions of (c, x_1^k) and $S_{x_1^k}(N)$, $k = 0, 1, 2, 3$.

As a specific example, consider a plant modeled by the Fliess operator

$$y = F_c[u] = \sum_{k=0}^{\infty} (c, x_1^k) E_{x_1^k}[u](t, 0),$$

where the generating series is $c = \sum_{k \geq 0} x_1^k$. The system has the state space realization

$$\dot{z}(t) = u(t), \quad z(0) = 0, \quad y(t) = e^{z(t)} \quad (12)$$

since for all $t \geq 0$

$$y(t) = \sum_{k=0}^{\infty} E_{x_1}^k[u](t, 0)\frac{1}{k!} = \sum_{k=0}^{\infty} E_{x_1^k}[u](t, 0) = F_c[u](t).$$

The output y shown in Fig. 3 is computed from a numerical simulation of the state space model (12) when the input $u(t) = 2e^{-t/3}\sin(2\pi t)$ is applied. The output of the predictor $\hat{y}_p(N)$, $N \geq 0$ as implemented using (6), (10), and (11) is also shown in the figure. As the predictor processes more data, its estimate of the output y improves asymptotically.

The more challenging problem is systematically building a real representation of $\mathcal{M}_C(X)$ when X has more than one letter, as in the multivariable case or when the *drift letter* x_0 is present. A partial ordering $\preceq$ is first defined on all words in X^*. For all $\zeta, \eta \in X^*$, let $\zeta \preceq \eta$ if and only if there exists a $\gamma \in X^*$ such that $\gamma^{-1}(\eta) = \zeta$, where γ^{-1} denotes the left-shift operator. The following theorem is proved in Appendix B.

Theorem 9 (Venkatesh et al., 2019). *The pair $(X^*, \preceq)$ is a partially ordered set.*

The partial order $(X^*, \preceq)$ can be graphically represented by a Hasse diagram. Starting with $\emptyset$ at the root, the Hasse diagram of $(X^*, \preceq)$ when $X = \{x_0, x_1, \ldots, x_m\}$ forms an $(m+1)$-ary infinitely branching tree. Define an injective map $R : X \longrightarrow \mathcal{C}$, where $\mathcal{C}$ is a set of colors. Color the edge between the nodes η and $x_i\eta$ in the tree with the color $R(x_i)$. In (10), the underlying discrete-time Fliess operator $\hat{F}_c[\hat{u}]$ has been truncated up to words of length J. Therefore, the tree is pruned at the Jth level. Next, a depth-first search (DFS) algorithm is employed to traverse the graph and generate words. The corresponding vector of words, $\chi^J(X)$, is called the *order vector* of degree J and is given by $\chi^0(X) = [\emptyset]$ and

$$\chi^{J+1}(X) = \begin{bmatrix} \emptyset & \chi^J(X)x_0 & \chi^J(X)x_1 & \cdots & \chi^J(X)x_m \end{bmatrix}^T \quad (13)$$

for $J \geq 0$.

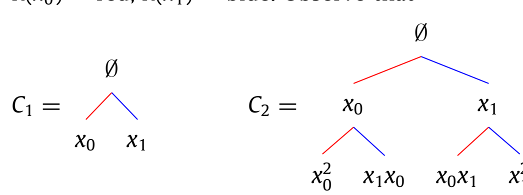
Let $\mathcal{S}^J(N+1)$ denote the matrix $\mathcal{S}(N+1)$ truncated for words up to length J, i.e., $\mathcal{S}^J(N+1) \in \mathbb{R}^{l \times l}$ with $l = \text{card}(X^{\leq J})$. An inductive algorithm to build such matrices is developed next.

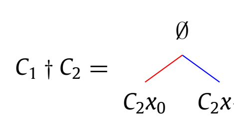
Definition 10. Define C_i as the colored tree of the Hasse diagram of $(X^*, \preceq)$ up to the ith level, that is, the $(m+1)$-ary tree with $\emptyset$ as the root and $\eta \in X^i$ as leaves of the Hasse diagram. Let $C := \{C_i : i \in \mathbb{N}_0\}$ be the set of colored trees given by $(X^*, \preceq)$ of all levels.

It is useful to define a product $\dagger$ on C as follows: $C_i \dagger C_j :=$\{tree with each leaf node $\beta \in X^i$ replaced by the tree C_j, where all the nodes of C_j are right concatenated with β\}. The following theorem is proved in Appendix C.

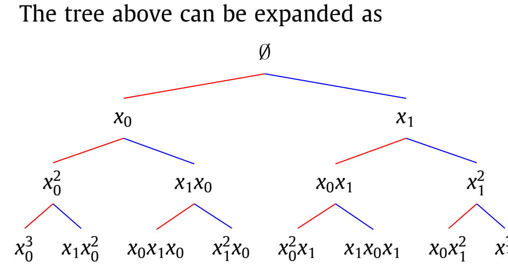
Theorem 11. *$(C, \dagger)$ is a commutative monoid isomorphic to the additive monoid $(\mathbb{N}_0, +)$. Specifically, $C_i \dagger C_j = C_{i+j}$ for all $C_i, C_j \in C$.*

Example 12. Let $X = \{x_0, x_1\}$ and define the color map R as: $R(x_0) = $ red, $R(x_1) = $ blue. Observe that





The tree above can be expanded as



This final tree is identified as C_3 so that $C_1 \dagger C_2 = C_3$.

Now assume that each color $R(x_i)$, $x_i \in X$, is given the weight $\hat{u}_i(N+1)$ at the discrete time instant $N+1$. Then it follows for any $\eta_j, \eta_k \in X^*$ that

$$[S(N+1)]_{jk} = (c_u(N+1)\eta_k, \eta_j) = \begin{cases} \text{weight of the path from } \eta_k \text{ to } \eta_j \text{ in } C_n, \\ \text{where } \; n \geq |\eta_j|. \end{cases}$$

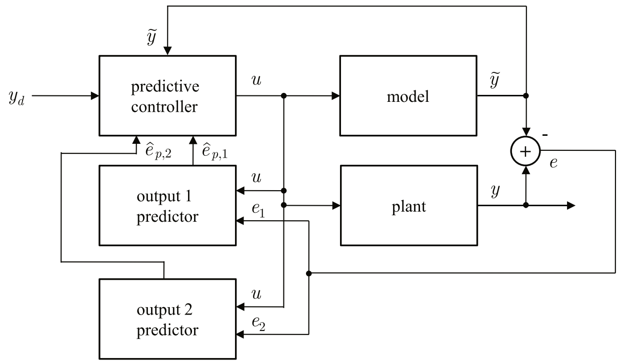


Fig. 4. Closed-loop system with a two-input, two-output predictive controller.

By Theorem 11, in the case where $X = \{x_0, x_1\}$ with color map R defined as in Example 12, $C_{J+1} = C_1 \dagger C_J$. That is,

$$C_{J+1} = \begin{array}{c} \emptyset \\ \swarrow \quad \searrow \\ C_J x_0 \quad C_J x_1 \end{array}$$

Hence, from the structure of the order vector in (13) and the above tree recursion, one can deduce for any $m \geq 1$ that the block structure of the matrix $\mathcal{S}^{J+1}(N+1)$ can be written inductively in terms of $\mathcal{S}^J(N+1)$ as:

$$\mathcal{S}^{J+1}(N+1) = \left[\begin{array}{c|c} 1 & 0\cdots 0 \\ \hline \hat{u}(N+1)\otimes(\mathcal{S}^J(N+1)e_1) & \begin{array}{c}\text{block diag}(\mathcal{S}^J(N+1), \\ \ldots, \mathcal{S}^J(N+1))\end{array} \end{array}\right], \quad (14)$$

where '⊗' denotes the Kronecker matrix product, and the block diagonal matrix is comprised on $m+1$ blocks.

Example 13. Let $X = \{x_0, x_1\}$. For $J = 2$, the words are indexed by the order vector $\chi^2(X) = [\emptyset\ x_0\ x_0^2\ x_1x_0\ x_1\ x_0x_1\ x_1^2]$. $\mathcal{S}^2(N+1)$ can be computed directly from C_2 to be

$$\mathcal{S}^2(N+1) = \left[\begin{array}{c|ccc|ccc} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline \hat{u}_0 & 1 & 0 & 0 & 0 & 0 & 0 \\ \hat{u}_0^2 & \hat{u}_0 & 1 & 0 & 0 & 0 & 0 \\ \hat{u}_1\hat{u}_0 & \hat{u}_1 & 0 & 1 & 0 & 0 & 0 \\ \hline \hat{u}_1 & 0 & 0 & 0 & 1 & 0 & 0 \\ \hat{u}_0\hat{u}_1 & 0 & 0 & 0 & \hat{u}_0 & 1 & 0 \\ \hat{u}_1^2 & 0 & 0 & 0 & \hat{u}_1 & 0 & 1 \end{array}\right]. \quad (15)$$

(For brevity, the argument $(N+1)$ is suppressed in the entries of the matrix.) But the same matrix can also be computed inductively from (14). For the base case, $\mathcal{S}^0(N+1) = 1$ so that

$$\mathcal{S}^1(N+1) = \left[\begin{array}{c|c|c} 1 & 0 & 0 \\ \hline \hat{u}_0 & 1 & 0 \\ \hat{u}_1 & 0 & 1 \end{array}\right].$$

Applying (14) once more gives (15).

4. Application to adaptive control

Suppose y_d is a desired output known to be in the range of a given plant with an underlying but unknown Fliess operator representation F_c. It is most likely in applications that y_d was designed using an assumed model (1). When both the plant and model are given the same input, a modeling error $e_i = y_i - \tilde{y}_i$ is generated for the ith channel of the plant's output as shown in Fig. 4. This signal and the applied input are then fed to a predictor of the type presented in the previous section in order to learn the input–output behavior of each error map $u \mapsto e_i$, $i = 1, \ldots, m$, which in this case must also have a Fliess operator representation. At any sample time the output of the plant is approximated by $\tilde{y}(N\Delta) + \hat{e}_p(N) = \tilde{y}(N\Delta) + Q(\hat{u}(N))$, where Q was defined using (10). The input u generated by the predictive controller to track y_d is a piecewise constant function taking values for $N \in \{1, 2, \ldots, L\}$ equivalent to

$$\hat{u}(N) := \arg\min_{|\hat{u}(N)|\leq\bar{u}} \hat{y}_e^T(N) W \hat{y}_e(N) \quad (16)$$

for some fixed bound $\bar{u} > 0$ and where

$$\hat{y}_e(N) := y_d(N\Delta) - [\tilde{y}(N\Delta) + Q(\hat{u}(N))]$$

with $W \in \mathbb{R}^{\ell\times\ell}$ being a fixed symmetric positive semi-definite weighting matrix. The MatLab command `fmincon` is used to compute local minima over the interval $[-\bar{u}, \bar{u}]$. In summary then equations (6), (10), and (16) provide a fully inductive implementation of a one-step ahead predictive controller. If the model is omitted from this set up, the resulting controller is still viable and can be viewed as a type of data-driven/model free closed-loop system as first proposed for SISO systems in Gray et al. (2017b).

As an example, consider the Lotka–Volterra model

$$\dot{z}_i = \beta_i z_i + \sum_{j=1}^{n} \alpha_{ij} z_i z_j, \quad i = 1, \ldots, n, \quad (17)$$

used to describe the population dynamics of n species in competition (Chauvet et al., 2002; May & Leonard, 1975; Smale, 1976). Here z_i is the biomass of the ith species, β_i represents the growth rate of the ith species, and the parameter α_{ij} describes the influence of the jth species on the ith species. Assume a subset of system parameters β_{i_j}, $j = 1, \ldots, m$ in (17) can be actuated and thus viewed as inputs u_i, $i = 1, \ldots, m$, and a set of output functions is given

$$y_j = h_j(z), \quad j = 1, \ldots, \ell.$$

Since the inputs enter the dynamics linearly, the input–output map $u \mapsto y$ has an underlying Fliess operator representation F_c with generating series $c \in \mathbb{R}^\ell\langle\langle X\rangle\rangle$ computable directly from (2) for a given initial condition z_0. Of particular interest here is the two dimensional predator–prey system

$$\dot{z}_1 = \beta_1 z_1 - \alpha_{12} z_1 z_2$$
$$\dot{z}_2 = -\beta_2 z_2 + \alpha_{21} z_1 z_2,$$

where $y_1 = z_1$ and $y_2 = z_2$ are taken to be the population of prey and predator species respectively, and (17) has been re-parameterized so that $\beta_i, \alpha_{ij} > 0$. This positive system has precisely two equilibria when all the parameters are fixed, namely, a saddle point equilibrium at the origin and a center at $z_e = (\beta_2/\alpha_{21}, \beta_1/\alpha_{12})$ corresponding to periodic solutions. Taking the system inputs to be $u_1 = \beta_1$ and $u_2 = \beta_2$, the *orbit transfer* problem is to determine an input to drive the system from some initial orbit to within an ϵ neighborhood of a final orbit using a given orbit transfer trajectory.

The proposed controller was tested in simulation assuming all the plant's parameters are set to unity. The discretization parameters are given in Table 1. The tracking performance for various choices of model parameter errors and input bounds $\|\hat{u}\|_\infty$ (the

Table 1
Discretization parameters for simulations.

L	J	T	Δ	ϵ
100	3	6	0.06	0.05

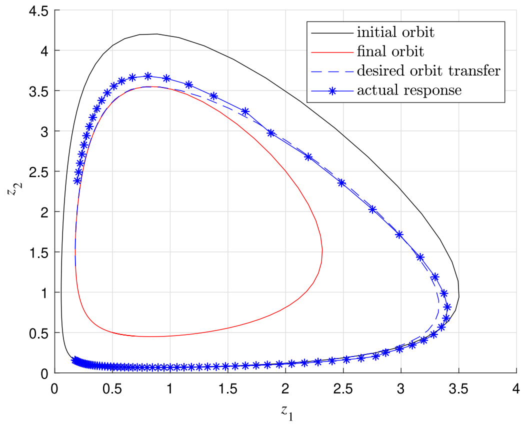


Fig. 5. Orbit transfer with +20% model error in α_{21}.

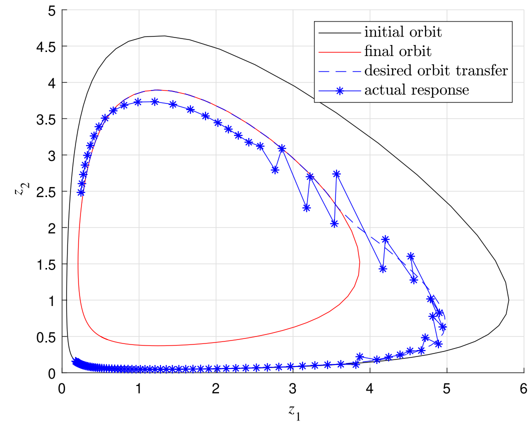


Fig. 6. Orbit transfer with -20% model error in α_{21}.

Table 2
Normalized RMS tracking errors for MIMO system.

$\Delta\alpha_{12}$	$\Delta\alpha_{21}$	δy_1	δy_2	$\|\hat{u}\|_\infty$
Exact model		8.66×10^{-9}	1.25×10^{-8}	2
−5	0	0.012	0.007	2
0	−5	0.020	0.016	2
5	0	0.004	0.006	2
0	5	0.018	0.015	2
−10	0	0.016	0.012	1.5
0	−10	0.056	0.041	1.5
10	0	0.010	0.009	1.5
0	10	0.037	0.025	1.5
−20	0	0.023	0.024	0.5
0	−20	0.144	0.113	0.5
20	0	0.012	0.016	0.5
0	20	0.071	0.047	0.5
−50	0	0.092	0.096	0.5
50	0	0.010	0.028	0.5
0	50	0.062	0.095	0.5
Model free		0.191	0.897	1

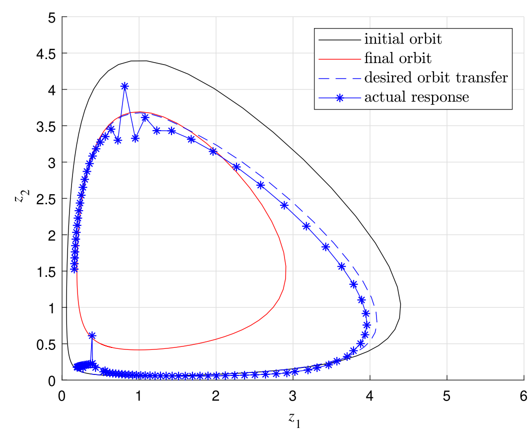


Fig. 7. Orbit transfer with no model.

positivity constraint on the input was not enforced) is shown in Table 2. For each plant parameter λ, $\Delta\lambda := (\lambda_{\text{model}} - \lambda_{\text{plant}}) \times 100\%$. In addition, δy_i for $i = 1, 2$ is the RMS error per sample normalized by the sample value of the desired trajectory for the given output channel. First the control system was tested assuming the exact plant model is available. In which case, the modeling error is zero and the predictors are inactive. The closed-loop performance is therefore determined solely by the predictive controller, which is quite accurate. Next a variety of parametric errors were introduced in the model. For all such cases, the weighting matrix W was set to the identity matrix. As an example, the simulation results for the case of $+20\%$ error in α_{21} are shown in Fig. 5. The case where $\Delta\alpha_{21} = -20\%$ is an extreme scenario as decreasing α_{21} any further resulted in the plant's response being oscillatory as shown in Fig. 6. Finally, the model free case was also simulated as shown in Fig. 7. Here it was necessary to select a nontrivial weighting matrix, in this case $W = [1\ 0.25;\ 0.25\ 1]$, in order for the optimizer to compensate for the cross coupling between the input–output channels, something that was done automatically when a model was present. Note that tracking was achieved, but the performance was about an order of magnitude worse than most cases employing a model. While in practice input bounds are dictated by the physical application, it was observed here that the larger the modeling error, the more conservative the input bounds needed to be in order to avoid instabilities. On the other hand, if the bounds were too conservative, then there was not enough actuation energy available to follow the desired trajectory.

5. Conclusions and future research

The nonlinear system identification problem was solved for multivariable nonlinear input–output systems that can be represented in terms of Chen–Fliess functional expansions. The problem was formulated in terms of a regression which is linear in the parameters and has a nonlinear regressor. An inductive implementation of the nonlinear regressor was developed using the underlying noncommutative algebraic and combinatorial structures. The method was demonstrated in an adaptive control application involving a two-input, two-output Lotka–Volterra dynamical system.

Future work will include the introduction of measurement noise in the identification problem, considering the persistency of excitation problem in this setting, exercising the method on more complex engineering plants, and identifying conditions under which closed-loop stability can be guaranteed.

Appendix A. Proof of Theorem 3

The first identity is addressed by proving that

$$S_\eta[\hat{u}](N+1,N_0) = (c_u(N+1)S[\hat{u}](N,N_0),\eta), \quad \forall \eta \in X^*$$

via induction on the length of η. When $\eta = \emptyset$ then trivially $S_\emptyset[\hat{u}](N+1,N_0) = 1 = \hat{u}_\emptyset(N+1)S_\emptyset[\hat{u}](N,N_0)$. If $\eta = x_i \in X$ then from (7)

$$\begin{aligned}S_{x_i}[\hat{u}](N+1,N_0) &= \hat{u}_{x_i}(N+1) + S_{x_i}[\hat{u}](N,N_0)\\ &= \sum_{x_i=\xi\nu} \hat{u}_\xi(N+1)S_\nu[\hat{u}](N,N_0)\\ &= (c_u(N+1)S[\hat{u}](N,N_0),x_i).\end{aligned}$$

Finally, assume the identity holds for all words up to some fixed length $n \geq 0$. Then for any $\eta \in X^n$ and $x_i \in X$ it follows that

$$\begin{aligned}&S_{x_i\eta}[\hat{u}](N+1,N_0)\\ &= \hat{u}_{x_i}(N+1)S_\eta[\hat{u}](N+1,N_0) + S_{x_i\eta}[\hat{u}](N,N_0)\\ &= \sum_{\eta=\xi\nu} \hat{u}_{x_i}(N+1)\hat{u}_\xi(N+1)S_\nu[\hat{u}](N,N_0)+\\ &\quad \hat{u}_\emptyset(N+1)S_{x_i\eta}[\hat{u}](N,N_0)\\ &= \sum_{x_i\eta=\xi\nu} \hat{u}_\xi(N+1)S_\nu[\hat{u}](N,N_0)\\ &= (c_u(N+1)S[\hat{u}](N,N_0),x_i\eta),\end{aligned}$$

which proves the claim for all $\eta \in X^*$. The second identity in the theorem follows directly from the first.

Appendix B. Proof of Theorem 9

Let $\eta,\zeta,\gamma,\alpha,\beta \in X^*$. Reflexivity is trivial since $\emptyset^{-1}(\eta) = \eta$ if and only if $\eta \preceq \eta$. To prove transitivity, first observe that

$$(\eta \preceq \zeta) \Rightarrow \exists\beta : \beta^{-1}(\zeta) = \eta$$
$$(\zeta \preceq \gamma) \Rightarrow \exists\alpha : \alpha^{-1}(\gamma) = \zeta$$

so that

$$((\alpha\beta)^{-1}(\gamma) = \eta) \Rightarrow \eta \preceq \gamma.$$

Therefore,

$$(\eta \preceq \zeta) \wedge (\zeta \preceq \gamma) \Rightarrow \eta \preceq \gamma.$$

To prove anti-symmetry, note that

$$(\eta \neq \zeta) \wedge (\eta \preceq \zeta) \Rightarrow \nexists\beta : \beta^{-1}(\eta) = \zeta,$$

and therefore,

$$(\eta \preceq \zeta) \wedge (\zeta \preceq \eta) \Rightarrow \eta = \zeta.$$

Hence, $(X^*,\preceq)$ is a partially ordered set.

Appendix C. Proof of Theorem 11

A preliminary lemma is needed first. For any fixed $\eta \in X^*$ define the *right concatenation map* as

$$\mathcal{R}_\eta(\zeta) = \zeta\eta, \quad \forall\zeta \in X^*.$$

Lemma 14. *Every right concatenation map is an order embedding map on $(X^*,\preceq)$. That is, $\zeta \preceq \gamma$ if and only if $\zeta\eta \preceq \gamma\eta$ for all $\zeta,\gamma,\eta \in X^*$.*

Proof. From the definition of $\preceq$ it follows that

$$\zeta \preceq \gamma \iff \exists\lambda \in X^* : \gamma = \lambda\zeta.$$

Applying $\mathcal{R}_\eta$ to both sides of the equality above gives

$$\begin{aligned}\zeta \preceq \gamma &\iff \gamma\eta = \lambda\zeta\eta\\ &\iff \zeta\eta \preceq \gamma\eta.\end{aligned}$$

Proof (*Theorem* 11)**.** First the identity $C_i \dagger C_j = C_{i+j}$ is proved. From the definition of the tree C_j and Lemma 14 it is clear that $C_j\eta$ has a Hasse diagram with η as the root and $X^j\eta$ as the set of leaf nodes. By the definition of the dagger product, every leaf node η of C_i is replaced by ηC_j since

$$X^{i+j} = \bigsqcup_{\eta\in X^i} X^j\eta.$$

Therefore, $C_i \dagger C_j$ has a Hasse diagram with $\emptyset$ as the root and X^{i+j} as the set of leaf nodes, that is, $C_i \dagger C_j = C_{i+j}$. It is now easily checked using this identity that $(C,\dagger)$ is associative, commutative, and has C_0 as the unit. Hence, $(C,\dagger)$ forms a commutative monoid. The monoid isomorphism between $\mathbb{N}_0$ and C is given by the bijection $i \mapsto C_i$ for all $i \in \mathbb{N}_0$.

References

Batselier, K., Chen, Z., & Wong, N. (2017a). A tensor network Kalman filter with an application in recursive MIMO Volterra system identification. *Automatica*, *84*, 17–25.

Batselier, K., Chen, Z., & Wong, N. (2017b). Tensor network alternating linear scheme for MIMO Volterra system identification. *Automatica*, *84*, 26–35.

Berstel, J., & Reutenauer, C. (1988). *Rational series and their languages*. Berlin: Springer-Verlag.

Birpoutsoukis, G., Marconato, A., Lataire, J., & Schoukens, J. (2017). Regularized nonparametric Volterra kernel estimation. *Automatica*, *82*, 324–327.

Chauvet, E., Paullet, J. E., Previte, J. P., & Walls, Z. (2002). A Lotka–Volterra three-species food chain. *Mathematics Magazine*, *75*, 243–255.

Doyle III, F. J., Pearson, R. K., & Ogunnaike, B. (2002). *Identification and control using Volterra models*. London: Springer-Verlag.

Duffaut Espinosa, L. A., Ebrahimi-Fard, K., & Gray, W. S. (2018). In K. Ebrahimi-Fard, & M. Barbero Liñán (Eds.), *Discrete mechanics, geometric integration and Lie–Butcher series, Combinatorial Hopf algebras for interconnected nonlinear input–output systems with a view towards discretization* (pp. 139–183). Cham, Switzerland: Springer Nature Switzerland AG.

Fliess, M. (1981). Fonctionnelles causales non linéaires et indéterminées non commutatives. *Bulletin de la Société Mathématique de France*, *109*, 3–40.

Fliess, M. (1983). Réalisation locale des systèmes non linéaires, algèbres de Lie filtrées transitives et séries génératrices non commutatives. *Inventiones Mathematicae*, *71*, 521–537.

Goodwin, G. C., & Sin, K. S. (2009). *Adaptive filtering prediciton and control*. Mineola, NY: Dover Publications, Inc.

Gray, W. S., Duffaut Espinosa, L. A., & Ebrahimi-Fard, K. (2017a). Discrete-time approximations of Fliess operators. *Numerische Mathematik*, *137*, 35–62.

Gray, W. S., Duffaut Espinosa, L. A., & Kell, L. T. (2017). Data-driven SISO predictive control using adaptive discrete-time Fliess operator approximations. In *Proc. 21st Inter. conf. on system theory, control and computing*. (pp. 383–388). Sinaia, Romania.

Gray, W. S., Venkatesh, G. S., & Duffaut Espinosa, L. A. (2019a). Discrete-time Chen series for time discretization and machine learning. In *Proc. 2019 53rd Conf. on information sciences and systems*. (pp. 1–6). Baltimore, MD.

Gray, W. S., Venkatesh, G. S., & Duffaut Espinosa, L. A. (2019b). Combining learning and model based control: Case study for single-input Lotka–Volterra system. In *Proc. 2019 American control conf.* (pp. 928–933). Philadelphia, PA.

Gray, W. S., & Wang, Y. (2002). Fliess operators on L_p spaces: Convergence and continuity. *Systems & Control Letters*, *46*, 67–74.

Grüne, L., & Kloeden, P. E. (2001). Higher order numerical schemes for affinely controlled nonlinear systems. *Numerische Mathematik*, *89*, 669–690.

Hu, Y., Tan, L., & de Callafon, R. A. (2019). Persistent excitation condition for MIMO Volterra system identification with Gaussian distributed input signals. In *Proc. 58th IEEE conf. on decision and control*. (pp. 1752–1757). Nice, France.

Isidori, A. (1995). *Nonlinear control systems* (3rd ed.). London: Springer-Verlag.

Ljung, L. (2010). Perspectives on system identification. *Annual Reviews in Control*, *34*, 1–12.

May, R. M., & Leonard, W. J. (1975). Nonlinear aspects of competition between three species. *SIAM Journal of Applied Mathematics*, *29*, 243–253.

Padoan, A., & Astolfi, A. (2016). Nonlinear system identification for autonomous systems via functional equations methods. In *Proc. 2016 American control conf.* (pp. 1814–1819). Boston, MA.

Pillonetto, G., Dinuzzo, F., Chen, T., De Nicolao, G., & Ljung, L. (2014). Kernel methods in system identification, machine learning and function estimation: A survey. *Automatica*, *50*, 657–682.

Schoukens, J., & Ljung, L. (2019). Nonlinear system identification, A user-oriented roadmap. *IEEE Control Systems Magazine*, 28–99.

Smale, S. (1976). On the differential equations of species in competition. *Journal of Mathematical Biology*, *3*, 5–7.

Venkatesh, G. S., Gray, W. S., & Duffaut Espinosa, L. A. (2019). Combining learning and model based multivariable control. In *Proc. 58th IEEE conf. on decision and control*. (pp. 1013–1018). Nice, France.

W. Steven Gray received the B.S. degree in electrical engineering from Purdue University in 1983, the M.S. degree in electrical engineering, the M.S. degree in applied mathematics, and the Ph.D. in electrical engineering from the Georgia Institute of Technology in 1985, 1988, and 1989, respectively. He is currently on the faculty of the Department of Electrical and Computer Engineering at Old Dominion University in Norfolk, Virginia, USA. His research interests include modeling and control theory for nonlinear systems, especially from the algebraic and combinatorial points of view.

G.S. Venkatesh received his Bachelor of Technology in Electronics and Instrumentation from Amrita Vishwa Vidyapeetham, India in 2011, a Master of Science in Computer Science from Chennai Mathematical Institute, India in 2016, a Master of Technology in Systems and Control from the Indian Institute of Technology, Bombay, India in 2014. He is currently a doctoral candidate in the Department of Electrical and Computer Engineering at Old Dominion University in Norfolk, Virginia, USA. His research focus is on Chen–Fliess series and their algebras in the context of nonlinear systems theory.

Luis A. Duffaut Espinosa received his B.S. degree in physics from the Universidad Nacional de Ingeniería, Lima, Perú in 2003, the M.S. degree in mathematics with mention in stochastic processes from the Pontificia Universidad Católica del Perú, Lima, Perú in 2005, and his Ph.D. degree in electrical and computer engineering from Old Dominion University, Norfolk, Virginia, USA in 2009. He is currently an Assistant Professor in the Department of Electrical and Biomedical Engineering at the University of Vermont, Burlington, Vermont, USA. His research interests span modeling, control and estimation of nonlinear systems; stochastic processes and algebraic combinatorics.