Mathematical Biosciences 239 (2012) 117–123

Contents lists available at SciVerse ScienceDirect

Mathematical Biosciences

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

Alternative to Ritt's pseudodivision for finding the input-output equations of multi-output models

Nicolette Meshkat [a,*], Chris Anderson [a], Joseph J. DiStefano III [b]

[a] UCLA, Department of Mathematics, United States
[b] UCLA, Department of Computer Science, United States

ARTICLE INFO

Article history:
Received 28 October 2011
Received in revised form 20 April 2012
Accepted 30 April 2012
Available online 22 May 2012

Keywords:
Identifiability
Differential algebra
Gröbner basis
Ritt's pseudodivision

ABSTRACT

Differential algebra approaches to structural identifiability analysis of a dynamic system model in many instances heavily depend upon Ritt's pseudodivision at an early step in analysis. The pseudodivision algorithm is used to find the characteristic set, of which a subset, the input-output equations, is used for identifiability analysis. A simpler algorithm is proposed for this step, using Gröbner Bases, along with a proof of the method that includes a reduced upper bound on derivative requirements. Efficacy of the new algorithm is illustrated with several biosystem model examples.

1. Introduction

A priori structural identifiability analysis is concerned with finding one or more sets of solutions for the unknown parameters $\boldsymbol{p}$ of a structured dynamic system model with state and output equations of the form (1.1) from noise-free input–output $\{\boldsymbol{u}(t),\boldsymbol{y}(t)\}$ data:

$$\dot{\boldsymbol{x}}(t,\boldsymbol{p}) = \boldsymbol{f}(\boldsymbol{x}(t,\boldsymbol{p}),\boldsymbol{u}(t),t;\boldsymbol{p}), t \in [t_0, T]$$
$$\boldsymbol{y}(t,\boldsymbol{p}) = \boldsymbol{g}(\boldsymbol{x}(t,\boldsymbol{p});\boldsymbol{p}) \quad (1.1)$$

Here $\boldsymbol{x}$ is a n-dimensional state variable, $\boldsymbol{p}$ is a P-dimensional parameter vector, $\boldsymbol{u}$ is the r-dimensional input vector, and $\boldsymbol{y}$ is the m-dimensional output vector. In the differential algebra approach, one assumes $\boldsymbol{f}$ and $\boldsymbol{g}$ are rational polynomial functions of their arguments, a reasonable assumption in most applications. The assumption that the output vector is only dependent upon elements of the state variable, and not its derivatives, will be important for the analysis in this work.

Differential algebra approaches have been shown to be quite useful in addressing global as well as local identifiability properties of these models [1–4]; and several differential algebra algorithms have been developed and implemented in available software packages [4–6]. Unfortunately, all are encumbered by computational algebraic complexity or other difficulties, and are limited thus far to relatively low dimensional models [7–9]. To alleviate some of this computational complexity, we describe a procedure that simplifies the task of determining the input-output equations, which is an important early step in preparing the system for identifiability analysis, as considered in [10–14]. The general idea behind the simplified procedure is to use a Gröbner Basis instead of the more cumbersome Ritt's pseudodivision to transform (1.1) into an implicit input-output map involving only the elements and derivatives of $\boldsymbol{y}$ and $\boldsymbol{u}$ along with the parameters $\boldsymbol{p}$, as shown in [15,16]. We extend the ideas of [15,16] by finding a stricter bound on the minimum number of derivatives of the equations needed in forming the Gröbner Basis for the multi-output case.

2. Differential algebra approach to identifiability in brief

We now summarize the differential algebra approach to structural identifiability, as well as some differential algebraic concepts. For more details on differential algebra, the reader is referred to [17,18].

From (1.1), an input–output map is determined in implicit form using a process called Ritt's pseudodivision algorithm [4]. The result of the pseudodivision algorithm is called the *characteristic set* [2]. Since the ideal generated by (1.1) is a prime ideal [19], the characteristic set is a finite "minimal" set of differential polynomials which generate the same differential ideal as that generated by (1.1) [4]. The first m equations of the characteristic set are those independent of the state variables, and form the input–output relations [4]:

$$\boldsymbol{\Psi}(\boldsymbol{y},\boldsymbol{u},\boldsymbol{p}) = 0 \quad (2.1)$$

* Corresponding author. Address: Department of Mathematics, 520 Portola Plaza, Los Angeles, CA 90095, United States.Tel.: +1 310 825 7482.
E-mail address: nmeshkat@math.ucla.edu (N. Meshkat).

0025-5564/$ - see front matter
http://dx.doi.org/10.1016/j.mbs.2012.04.008

The m equations of the input–output relations $\boldsymbol{\Psi}(\boldsymbol{y},\boldsymbol{u},\boldsymbol{p}) = 0$ are polynomial equations in $\boldsymbol{u},\dot{\boldsymbol{u}},\ddot{\boldsymbol{u}},\ldots,\boldsymbol{y},\dot{\boldsymbol{y}},\ddot{\boldsymbol{y}},\ldots$, called *differential* polynomials [17], with rational coefficients in the elements of the parameter vector $\boldsymbol{p}$.

For example, a simple first-order model, adapted from [20]:

$$\dot{x} = p_1 x + p_2 u$$

$$y = p_3 x$$

with the chosen ranking $\dot{x} > x > \dot{y} > y > u$ yields an input–output equation, $\boldsymbol{\Psi}(\boldsymbol{y},\boldsymbol{u},\boldsymbol{p}) = 0$, of the form:

$$\boldsymbol{\Psi}(\boldsymbol{y},\boldsymbol{u},\boldsymbol{p}) = \frac{\dot{y}}{p_3} - \frac{p_1}{p_3} - p_2 u = 0$$

The characteristic set is in general not unique, but the coefficients $\boldsymbol{c}(\boldsymbol{p})$ of the input–output equations can be fixed uniquely by normalizing the equations to make them monic, for example, by multiplying by p_3 [4]:

$$\dot{y} - p_1 y - p_2 p_3 u = 0$$

Structural identifiability can be determined by testing the injectivity of the coefficients $\boldsymbol{c}(\boldsymbol{p})$, i.e. the model (1.1) is globally identifiable if and only if $\boldsymbol{c}(\boldsymbol{p}) = \boldsymbol{c}(\boldsymbol{p}^*)$ implies $\boldsymbol{p} = \boldsymbol{p}^*$ for arbitrary $\boldsymbol{p}^*$ [4]. Thus, $p_1 = p_1^*$ and $p_2 p_3 = p_2^* p_3^*$imply that only p_1 and $p_2 p_3$ can be determined in our example, so the model is unidentifiable.

3. Ritt's pseudodivision algorithm

Ritt's pseudodivision is the algorithm that has been more commonly used to find the characteristic set of a prime differential ideal generated by a finite set of differential polynomials [17]. The following procedure follows that in [19].

Let u_j be the leader of a differential polynomial A_j, which is the highest ranking derivative of the variables appearing in that polynomial. A polynomial A_i is said to be of lower rank than A_j if $u_i < u_j$ or, whenever $u_i = u_j$, the algebraic degree of the leader of A_i is less than the algebraic degree of the leader of A_j. A polynomial A_i is reduced with respect to a polynomial A_j if A_i contains neither the leader of A_j with equal or greater algebraic degree, nor its derivatives. If A_i is not reduced with respect to A_j it can be reduced by using the following pseudodivision algorithm:

(1). If A_i contains the kth derivative $u_j^{(k)}$ of the leader of A_j, differentiate A_j k times so its leader becomes $u_j^{(k)}$.
(2). Multiply the polynomial A_i by the coefficient of the highest power of $u_j^{(k)}$ and let R be the remainder of the division of this new polynomial by $A_j^{(k)}$ with respect to the variable $u_j^{(k)}$. Then R is reduced with respect to $A_j^{(k)}$. The polynomial R is called the pseudoremainder of the pseudodivision.
(3). The polynomial A_i is replaced by the pseudoremainder R and the process is iterated using $A_j^{(k-1)}$ in place of $A_j^{(k)}$ and so on, until the pseudoremainder is reduced with respect to A_j.

This algorithm is applied to a set of differential polynomials, rendering each polynomial reduced with respect to each other, to form an auto-reduced set. The result is a characteristic set.

In addition to Ritt's algorithm, a number of other algorithms have been developed to find the full characteristic set, such as the Ritt–Kolchin algorithm [18] and the improved Ritt–Kolchin algorithm [21]. Software implementations of these algorithms can be found in the diffgrob2 package [5] or the diffalg package [6]. However, as noted in [14], "algorithms to find the characteristic set are still under development and the existing software packages do not always work well."

The DAISY program [4] uses Ritt's pseudodivision algorithm to obtain the characteristic set and then the input–output equations, i.e. the first m equations of the characteristic set. While the DAISY program is a useful tool in exploring global or local identifiability properties of systems, the user may want to obtain the input–output equations for other analyses, e.g. for finding identifiable parameter combinations, as in [9]. Copying the characteristic set from DAISY into a different symbolic algebra package is cumbersome due to syntax differences, especially for large systems.

Alternatively, one could implement Ritt's pseudodivision using any symbolic algebra package, as it requires only low level symbolic operations, e.g. differentiation and polynomial division. While this aspect is good from the standpoint of making few demands on the capabilities of a symbolic software system, it has the negative consequence that the method is time consuming to implement and prone to implementation errors. Since only the input–output equations – and not the full characteristic set – are needed for differential algebra identifiability analysis, a simpler method to obtain just the input–output equations can be quite helpful. We propose an alternative procedure here that utilizes differentiation and Gröbner Bases to ease the implementation difficulties.

4. Alternative method to find input–output equations

We obtain the input–output relations by taking a sufficient number of derivatives of the system (1.1), followed by computation of a Gröbner Basis of the new system, similar to the method proposed in [15] and [16]. The main difference between our approach and that proposed in [15] and [16] is that we find a stricter bound on the minimum number of derivatives of the equations needed in forming the Gröbner Basis. Following a minimum number of differentiations of the output equations and corresponding state variable equations, the Buchberger Algorithm is used to eliminate all state variables and derivatives of state variables.

In general, for elimination to work, the number of equations must be strictly greater than the number of unknowns, as discussed in [15]. Since the output equation is of the form $\boldsymbol{y}(t,\boldsymbol{p}) = \boldsymbol{g}(\boldsymbol{x}(t,\boldsymbol{p});\boldsymbol{p})$, i.e. always in terms of $\boldsymbol{x}$ and not derivatives of $\boldsymbol{x}$, then the first step is to take the derivative of the output equations, to help eliminate the first derivative of $\boldsymbol{x}$ from the state variable equations. If this additional equation is not enough to eliminate the state variables, then the second derivative of the output equations is needed. This, however, introduces the second derivative of $\boldsymbol{x}$, and thus differentiation of the corresponding state variable equations is needed. Differentiation of the output equations and corresponding state variable equations is continued until the number of equations is greater than the number of unknowns. The procedure is described by the following steps:

Step 1: Differentiate the output equations to obtain $\dot{\boldsymbol{y}}$ and adjoin these equations to the system.

Step 2: Differentiate the output equations again to obtain $\ddot{\boldsymbol{y}}$ and differentiate the corresponding state variable equations to obtain equations involving $\ddot{\boldsymbol{x}}$. Adjoin these equations to the system.

Step k: Differentiate the output equations again to obtain $\boldsymbol{y}^{(k)}$ and differentiate the corresponding state variable equations to obtain equations involving $\boldsymbol{x}^{(k)}$. Adjoin these equations to the system.

For a single output system, the above steps yield a procedure similar to that in [15] and [16]. However, the method described in [15] and [16] does not formally treat multi-output models, and the procedure we present here does.

We now show that, for a system of n state variables and m output equations, one need only take $n-(m-1)$ steps to obtain a sufficient number of additional equations to eliminate the state variable terms in a Gröbner Basis. We begin with examples illus-