AN ALGORITHM FOR DETERMINING INVARIANTS IN QUASI-POLYNOMIAL SYSTEMS

Barna Pongrácz, Gábor Szederkényi, Katalin M. Hangos

Process Control Research Group, Systems and Control Laboratory
Computer and Automation Research Institute
Hungarian Academy of Sciences
H-1518, P.O. Box 63, Budapest, Hungary
e-mail: pongracz@scl.sztaki.hu, szeder@sztaki.hu, hangos@scl.sztaki.hu

Abstract: In this paper an algorithm is proposed which is capable to retrieve a wide class of invariants in quasi-polynomial systems. The application of the algorithm is illustrated on the state-space model of a fed-batch fermentation process.

Keywords: quasi-polynomial system, QP, first integral, invariant, fed-batch fermentation

1. INTRODUCTION

Finding first integrals (invariants) plays a great role in modern systems and control theory e.g. in the field of canonical representations, controllability and observability analysis (Isidori, 1995) and stabilization of nonlinear systems (van der Schaft, 2000). If the given dynamical system is not integrable, then its first integrals (if they exist) give us very useful information about the properties of the solutions and about possibly physically meaningful conserved quantities.

The class of quasi-polynomial (QP) systems has gained a significant interest in the modelling of nonlinear dynamical systems since the majority of smooth nonlinear systems occurring in practice can be algorithmically transformed to QP form (Hernández-Bermejo, *et al.*, 1998). The theoretical background of the existence of quasi-polynomial invariants is well-founded: In (Figuerido, *et al.*, 1998) algebraic tools are applied to find semi-invariants and invariants in quasi-polynomial systems. A computer-algebraic software package called QPSI has also been implemented for the determination of quasi-polynomial invariants and the corresponding model parameter relations (Rocha Filho, *et al.*, 1999).

The purpose of this paper is to propose an algorithm for the retrieval of a frequent class of quasi-polynomial invariants, which is also effective in the case of high dimensional QP-ODE models with arbitrary number of monomials. The paper is organized as follows. Section 2 contains the basic notions that are needed to derive the main results. The main contribution of the paper can be found in Section 3 where the algorithm for the retrieval of invariants is described. Section 4 demonstrates the operation of the algorithm on the state space model of a fed-batch fermentation process.

2. PRELIMINARIES

2.1 Quasi-polynomial systems

Let us denote the element of an arbitrary matrix W with row index i and column index j by $W_{i,j}$. Furthermore, let the i-th row and j-th column of W denoted by $W_{i,\bullet}$ and $W_{\bullet,j}$ respectively. Quasi-polynomial systems are systems of ODEs. An *(n+1)* dimensional QP-ODE system can be represented in the following general form:

$$\dot{x}_i = x_i\left(\lambda_i + \sum_{j=1}^{m} \bar{A}_{i,j} U_j\right), \quad U_j = \prod_{k=1}^{n+1} x_k^{\bar{B}_{j,k}}, \quad i = 1,\ldots,n+1 \tag{1}$$

where $x \in \mathrm{int}(R_+^{n+1})$, $\bar{A} \in R^{(n+1)\times m}$, $\bar{B} \in R^{m\times(n+1)}$, $\lambda_i \in R$, $i = 1,\ldots,n+1$ and $\lambda = \left[\lambda_1 \cdots \lambda_{n+1}\right]^{\mathrm{T}}$. The product terms U_j, $j = 1,\ldots,m$ are called the quasi-monomials (or monomials) of the system. Without the loss of generality we can assume that $rank(\bar{B}) = n+1$, $m \geq n+1$, and the coefficient matrix $\bar{A}$ is of full rank (Hernández-Bermejo, *et al.*, 1998). By introducing the unit monomial $U_0 = 1$, the above general form can be written in a *homogeneous* (i.e. without linear terms) form as

$$\dot{x}_i = x_i \sum_{j=0}^{m} A_{i,j} U_j, \quad U_j = \prod_{k=1}^{n+1} x_k^{B_{j,k}}, \quad i = 1,\ldots,n+1 \tag{2}$$

2.2 The examined class of invariants

A function $I : \mathbb{R}^{n+1} \mapsto \mathbb{R}$ is called an invariant of (1) if

$$\frac{d}{dt} I = \frac{\partial I}{\partial x} \cdot \dot{x} = 0 \tag{3}$$

We consider quasi-polynomial invariants in (1) that can be written in the following special form:

$$I = F(x) - x_i^{\frac{1}{\beta}}, \quad \beta \in \mathbb{R} \tag{4}$$

where

$$F(x) = \sum_{k=1}^{p} c_k \prod_{\substack{j=1,\\ j\neq i}}^{n+1} x_j^{\alpha_{kj}}, \quad c_k, \alpha_{kj} \in \mathbb{R} \tag{5}$$

It's clear that (4) can be rewritten as

$$x_i^{\frac{1}{\beta}} = F(x) + c_0, \quad c_0 \in \mathbb{R} \tag{6}$$

This is a narrower class of invariants than the one examined in (Figuerido, *et al.*, 1998) since it contains those first integrals from where at least one of the variables can be expressed explicitly. However, many types of first integrals (e.g. conserved mechanical, thermodynamical or electrical energy) in physical system models belong to this class.

2.3 The underlying principle of the algorithm

Consider a set of *(n+1)* differential equations in the *homogeneous* form of (2). Let us assume without restriction of generality that *i=n+1* in (6) (because the QP form of the equations is preserved under permutation of the differential variables) i.e. the following algebraic dependence is present in (2):

$$x_{n+1}^{\frac{1}{\beta}} = c_0 + \sum_{\ell=1}^{L} c_\ell V_\ell \tag{7}$$

$$V_\ell = \prod_{k=1}^{n} x_k^{\alpha_{\ell k}}, \quad \alpha_{\ell k} \in \mathbb{R}, \;\; \ell = 1,\dots,L, \; k = 1,\dots,n \tag{8}$$

where $\beta,\; c_\ell \in \mathbb{R},\; \beta \neq 0$. It is clear that (7) is equivalent with the existence of a first integral of the form (4)-(5). Taking the time derivative of (7) and arranging it to the standard QP form gives

$$\dot{x}_{n+1} = \beta x_{n+1}^{\frac{\beta-1}{\beta}} \sum_{\ell=1}^{L} c_\ell \sum_{i=1}^{n} \frac{\partial V_\ell}{\partial x_i}\dot{x}_i \;=\; x_{n+1}\left(\sum_{\ell=1}^{L}\sum_{i=1}^{n} \beta c_\ell \alpha_{li} V_\ell x_{n+1}^{-\frac{1}{\beta}} \sum_{j=1}^{m} A_{i,j} U_j \right) \tag{9}$$

It is easy to see that the monomials in (9) (denoted by $R_{\ell j}$) and their coefficients ($c_{\ell j}$) are

$$R_{\ell j} = V_\ell U_j x_{n+1}^{-\frac{1}{\beta}} = x_1^{\alpha_{\ell 1}+B_{j1}} x_2^{\alpha_{\ell 2}+B_{j2}} \cdots x_n^{\alpha_{\ell n}+B_{jn}} x_{n+1}^{-\frac{1}{\beta}}, \quad j = 1,\dots,m, \;\; \ell = 1,\dots,L \tag{10}$$

$$\gamma_{\ell j} = \sum_{i=1}^{n} \beta c_\ell \alpha_{\ell i} A_{i,j}, \quad i = 1,\dots,n, \;\; j = 1,\dots,\; m, \;\; \ell = 1,\dots,L \tag{11}$$

where subscript *i* refers to that the partial differentiation in (9) has been performed by x_i.

Now, the aim of our algorithm is to determine β, the coefficients c_ℓ and the exponents $\alpha_{\ell i},\; \ell = 1,\dots,L,\; i = 1,\dots,n$ in (7)-(8) using the special form of the monomials (10) and that of their coefficients (11).

3. THE BASIC ALGORITHM FOR RETRIEVING INVARIANTS

The *input required by the algorithm* consists of the matrices *A* and *B* of the QP model in its *homogeneous form* defined in (2). The *operational condition of the algorithm* is that, consistently to our preliminary assumptions, matrices *A* and *B* are of full rank. Without the loss of generality we can assume that the explicit variable of the possible first integral is the last differential variable x_{n+1}. By a simple permutation of variables, each variable can be checked whether it is the explicit variable of a first integral.

Step 1. Determination of the monomial candidates
To find a first integral in the form (7), one has to use the relationship (10) defined between the monomials $U_j,\; j = 1,\dots,m$ of the original differential equations and the monomials $R_{\ell j},\; \ell = 1,\dots,L,\; j = 1,\dots,m$ of the ODE for the algebraically dependent variable x_{n+1}.

The first step is dedicated to collect these two groups of monomials, and then to determine the monomial candidates of the first integral using (10). Since the exponents of the *j*-th monomial of a QP-ODE are given as the *j*-th row vector of matrix *B*, the first thing to do is to gather the exponents of those monomials that occur in the first *n* differential equations and construct the matrix $B^{(U)}$ from them. Let us denote the matrix created from *A* by deleting its *(n+1)*-th row by A^*. Now construct $B^{(U)}$ in the following way:

Let $B^{(U)} = B$
Mark those rows $B^{(U)} = B_{j,\bullet}\quad j = 1,\dots,m$ *for which* $A^*_{\bullet,j} = 0$
Delete the marked rows from $B^{(U)}$

Collect the row vectors containing the exponents of monomials of the ODE for x_{n+1} to $B^{(R)}$:

Let $B^{(R)} = B$
Mark those rows $B^{(R)} = B_{j,\bullet}\quad j = 1,\dots,m$ *for that* $A_{(n+1),j} = 0$
Delete the marked rows from $B^{(R)}$

As a result, $B^{(U)} \in \mathbb{R}^{m_U \times (n+1)}$, $B^{(R)} \in \mathbb{R}^{m_R \times (n+1)}$, where m_U and m_R denote the number of monomials in the first *n*, and in the *(n+1)*-th differential equations, respectively. Note that there may be monomials (as row vectors) that appear in both $B^{(U)}$ and $B^{(R)}$.

As (10) shows, the exponents of the monomials U_j, $j = 1,\dots,m$ in the original differential equations and of the monomials V_ℓ, $\ell = 1,\dots,L$ in the algebraic equation are added up in the resulted monomials $R_{\ell j}$. This allows us to determine V_ℓ by simply *dividing* $R_{\ell j}$ by U_j for some *j*. This operation is equivalent with subtracting each exponent row vector corresponding to the monomials U_j (stored as row vectors of $B^{(U)}$) from the row vectors determining $R_{\ell j}$ (stored as row vectors of $B^{(R)}$).

Therefore the next step is that the algorithm determines the exponent row vectors by subtracting each row of $B^{(U)}$ from each row of $B^{(R)}$, and construct the matrix $B^{(V)}$ made of the resulted row vectors:

$$B^{(V)} \in \mathbb{R}^{(m_U m_R)\times(n+1)},\quad B^{(V)}_{k,\bullet} = B^{(R)}_{j,\bullet} - B^{(U)}_{i,\bullet}\ ,$$
$$k = (j-1)\times m_U + i,\quad j = 1,\dots,m_R,\quad i = 1,\dots,m_U$$

Finally, make sure that each monomial candidate is coded only once in $B^{(V)}$:

Delete repeated rows from $B^{(V)}$ *so that all rows are different*

As a result, $B^{(V)} \in \mathbb{R}^{m_V \times (n+1)}$ contains all the monomial candidates of the first integral, where $m_V \le m_U m_R$ denotes the number of these monomial candidates. However taking into account all possible monomial candidates may cause a huge redundancy but this guarantees that the exponent vectors of *all monomials of the first integral are contained in* $B^{(V)}$.

Step 2. Determination of β
To have a QP-type first integral from which x_{n+1} can be given explicitly, *the exponents of* x_{n+1} *in all of its monomials have to be identical.* This step classifies the exponent row vectors of the monomial candidates of the first integral by their last element.

Compute how many different last elements of the row vectors of $B^{(V)}$ have and denote this number by *S*. Now make *S* different sets Ω_k, $k=1,\ldots,S$ and collect all the row vectors of $B^{(V)}$ having identical last elements into the same sets, while row vectors with different last elements into different sets. The result is a system of sets, where *the elements of each set are exponent row vectors belonging to the same* β.

Now set the value of *k* to *k=1*.

Step 3. Determination of the coefficients
As the last step, search for a first integral with monomial candidates belonging to the same Ω_k. Since the exponents of the monomial candidates are already given, only their coefficients have to be determined. If these coefficients exist, the first integral exists for the current β, and it is completely determined by the algorithm. Denote the number of elements of Ω_k by *L*. Then the first integral candidate is given by the monomials described by the elements of Ω_k with unknown coefficients $c_1,\ldots,c_L$. Perform time-differentiation by simply applying (9) to it, with monomials and coefficients described in (10) and (11), respectively. Then match the monomials of this time-derivative and the monomials of the *(n+1)*-th differential equation, and determine the coefficients $\gamma_{\ell j}$, $\ell=1,\ldots,L$, $j=1,\ldots,m$ therefrom. Then try to solve the linear set of equations (11) for $c_1,\ldots,c_L$.

Three cases are possible:

- If (11) cannot be solved and $k < S$, increase *k* by one and jump to *Step 3*.
- If (11) cannot be solved and *k=S* the algorithm stops without finding a first integral.
- If (11) can be solved, the invariant is successfully determined and the algorithm stops.

4. EXAMPE: A FED-BATCH FERMENTATION PROCESS

An isotherm fed-batch fermentation process with bi-linear reaction characteristics is used as a case study which can be described by the following homogeneous QP model:

$$\dot{x}_1 = x_1\left(Fx_1^{-1}\right)$$
$$\dot{x}_2 = x_2\left(-K_r/Y\,x_3 + S_F F x_1^{-1} x_2^{-1} - Fx_1^{-1}\right)$$
$$\dot{x}_3 = x_3\left(K_r x_2 - Fx_1^{-1}\right)$$

where x_1, x_2 and x_3 are the vessel volume, the mass of the substrate and of the biomass, respectively, and all other symbols are constant parameters. The matrices of the QP model are

$$A=\begin{bmatrix} F & 0 & 0 & 0 \\ -F & K_r/Y & S_F F & 0 \\ -F & 0 & 0 & K_r \end{bmatrix}, \quad B=\begin{bmatrix} -1 & 0 & 0 \\ 0 & 0 & 1 \\ -1 & -1 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

Step 1. The matrices $B^{(U)}$ and $B^{(R)}$ containing the exponent row vectors of the monomials respectively in the first two, and in the third differential equations are:

$$B^{(U)}=\begin{bmatrix} -1 & 0 & 0 \\ 0 & 0 & 1 \\ -1 & 0 & 0 \end{bmatrix}, \quad B^{(R)}=\begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \end{bmatrix}$$

Now subtract each row $B^{(U)}$ from each row of $B^{(R)}$ to form $B^{(V)} \in \mathbb{R}^{6\times 3}$. Since the last elements of the rows of $B^{(V)}$ should give $-1/\beta$, these elements must be non-zero, meaning that four row vectors have to be cancelled from $B^{(V)}$ to get its final form:

$$B^{(V)} = \begin{bmatrix} 0 & 1 & -1 \\ -1 & 0 & -1 \end{bmatrix}$$

Step 2. Since the last components of both exponent vectors are identical, (both vectors belong to the same β=1) there is only one set: $\Omega_1 = \{[0 \quad 1 \quad -1], [-1 \quad 0 \quad -1]\}$.

Step 3. The parametric first integral is

$$x_3 = c_1 x_1^{-1} + c_2 x_2 + c_0$$

Its time-derivative does not provide solution for c_1, c_2. However, this first integral can be re-written in its implicit form, and its time-derivative gives the coefficients (except c_0 which comes from the initial conditions of the model):

$$x_1 \left(1/Y \, x_3 - x_2 + S_F \right) = S_F / c_0$$

Note that another algorithm based on Lie-algebras has been applied to a fermenter model with the same structure but slightly different reaction kinetics in (Szederkényi, *et al.*, 2002), giving the same final result. As a comparison, this new method provides a computationally more advantageous way of retrieval because it does not require the analytic solution of PDEs.

5. CONCLUSIONS AND FUTURE WORK

An algorithm is proposed in this paper for the determination of a class of first integrals in QP systems. This algorithm proved to be able to find QP type first integrals which are explicit in (at least) one of their variables moreover it operates without any heuristic steps. The operation of the algorithm was illustrated on a physical example. The results further support the fact that the QP representation of nonlinear systems can be very useful for the study of dynamical properties. Further work will be directed to the constructive application of the algorithm in control oriented nonlinear system analysis and feedback design.

Acknowledgements This research was partially supported by the grant no. OTKA T042710 and F046223. The second author is a grantee of the Bolyai János Research Scholarship of the Hungarian Academy of Sciences.

REFERENCES

Figuerido, A., Rocha Filho, T. M., Brenig, L. (1998). Algebraic structures and invariant manifolds of differential systems. *J. Math. Phys.*, **39,** 2929-2946.

Hernández-Bermejo, B., Fairén, V. and Brenig, L. (1998). Algebraic recasting of nonlinear systems of ODEs into universal formats. *J. Phys. A, Math. Gen.*, **31,** 2415-2430.

Isidori, A. (1995). *Nonlinear Control Systems.* Springer, Berlin.

Rocha Filho, T. M., Figuerido, A. and Brenig, L. (1999). A Maple package for the determination of QP symmetries and invariants. *Comp. Phys. Com.*, **117**, 263-272.

van der Schaft, A. J. (2000). *L2-Gain and Passivity Techniques in Nonlinear Control.* Springer Verlag, Berlin.

Szederkényi, G., Kovács, M. and Hangos, K. M. (2002). Reachability of nonlinear fed-batch fermentation processes, *Int. J. Robust Nonlinear Control*, **12**, 1109-1124.