



**GenSSI: GENERATING SERIES APPROACH FOR
TESTING STRUCTURAL IDENTIFIABILITY**

USER GUIDE

Oana Chiş, Julio R. Banga, and Eva Balsa-Canto

(Bio)Process Engineering Group
IIM-CSIC
SPAIN

E-mail: {chisoana, julio, ebalsa}@iim.csic.es

Copyright @ CSIC

Last revision 30 June 2011

Contents

1	Structural identifiability	2
1.1	Mathematical model formulation	2
1.2	Structural identifiability definition	3
1.3	Lie derivative	4
1.4	Generating series approach	4
1.5	Identifiability tableau	6
1.6	Solution of the parameters	8
1.7	How to chose the number of derivatives?	10
2	GenSSI software description	12
2.1	Toolbox requirements	12
2.2	Download and License	12
2.3	GenSSI features	13
2.4	GenSSI functions	14
3	Illustrative examples	18
3.1	Goodwin oscillator	18
3.2	<i>Arabidopsis thaliana</i> model	29
3.3	Glycolysis inspired metabolic pathway	42
4	Some comments on using SBML models with GenSSI	49

1

Structural identifiability

The analysis of the structure and dynamics of biosystems through data-based mathematical models has received major attention in the recent years. One topic is to find the solution of the so called *parameter estimation problem*. This is usually a very challenging task, due to the usual lack of identifiability. The identifiability problem is concerned with the possibility of calculating a unique solution for the parameters given by an experimental scheme. There are several reasons to assess structural identifiability. Possibly the most important is that the parameters of the model have biological meaning, and there arises the question whether it is possible to determine their values from experimental data. In fact, several authors have reported difficulties in assessing unique and meaningful values for the parameters in the context of biosystems (see for example [1, 7, 13, 16], among others). Nevertheless, structural identifiability analysis, related to the model structure and possibly to the type of input and independent of the parameter values, has been often disregarded.

GenSSI software offers a technique for studying structural identifiability, for a whole class of biological models, as described above by formula (1.1). It implements iteratively the generating series method, as presented in [21], with the help of identifiability tableaux, as described in [4].

1.1 Mathematical model formulation

Currently, the most typical approach for representing biosystems is through a set of coupled deterministic ordinary differential equations written generically as given below:

$$\Sigma(p) : \begin{cases} \dot{x}(t) = f(x(t), p) + g(x(t), p)u(t), & x(t_0) = x_0(p) \\ y(t, p) = h(x(t), p), \end{cases} \quad (1.1)$$

where

- $x = (x_1, \dots, x_n) \in \mathbf{R}^n$ is the n -dimensional state variable;
- f , g , and h are supposed polynomial or rational functions in the state variable;
- $u = (u_1, \dots, u_m) \in \mathbf{R}^m$ a m -dimensional input (control) vector of smooth functions;
- $y = (y_1, \dots, y_r) \in \mathbf{R}^r$ is the r -dimensional output (experimentally observed quantities);
- $x_0(p)$ are the initial conditions chosen such that the model to be well defined;
- $p \in \mathbf{P}$ are the unknown parameters of the model, where $\mathbf{P} \subset \mathbf{R}^q$, is an open and connected subset of \mathbf{R}^q .

1.2 Structural identifiability definition

Structural identifiability analysis regards the question whether the parameters can be identified (uniquely) from the available measurements, assuming perfect data (noise-free and continuous in time and space).

The following definitions are essential in structural identifiability theory.

Exhaustive summary - A vector $s(p)$ is an exhaustive summary of an experiment if it contains only information about the parameters p that can be extracted from knowledge of the control $u(t)$ and the measured quantities $y(t, p)$. Using generating series approach, the exhaustive summary is made of the series coefficients, evaluated at the initial conditions (initial state). Thus, the exhaustive summary is the set of relations that depend only on the parameters of the model.

One of the following types of identifiability can appear when testing it:

1. **structural global (or unique) identifiability** - A parameter p is structurally globally (or uniquely) identifiable, if there exists a unique solution that can be computed from the exhaustive summary;

2. **Structural local identifiability** - A parameter p is structurally locally identifiable if from the exhaustive summary more than one solution for the parameters is found;
3. **Structural non-identifiability** - If no solution could be computed for the set of parameters, than the model is called structurally non-identifiable.

1.3 Lie derivative

Generating series method is based on the computation of the Lie derivatives and its link to observability of the non-linear models [8]. The suitable Lie derivatives of h along the vector field f , respectively g , are defined generically in the following way:

$$L_f h(x(t), p) = \sum_{j=1}^n f_j(x(t), p) \frac{\partial}{\partial x_j} h(x(t), p), \quad (1.2)$$

$$L_g h(x(t), p) = \sum_{j=1}^n g_j(x(t), p) \frac{\partial}{\partial x_j} h(x(t), p), \quad (1.3)$$

with f_j , respectively g_j , the j th component of f , respectively g .

The output $y \in \mathbf{R}^r$ has the coefficients given by $h(x(t), p)|_{x_0(p)}$ and iterative Lie derivatives along f and/or g , evaluated at the initial conditions $x_0(p)$.

1.4 Generating series approach

Successive Lie derivatives, evaluated at the initial time $t = t_0$, are used to finding a unique solution of the parameters, assuming perfect data. For the general model (1.1), the output $y \in \mathbf{R}^r$ is expressed in series, with respect to the inputs $u \in \mathbf{R}^m$ and the time t [10].

The exhaustive summary incorporates all the information needed for checking the identifiability of the corresponding parameters. These coefficients were obtained iteratively in the following way.

First order Lie derivatives. The Lie derivatives of the analytic function h on \mathbf{R}^n , with respect to f , respectively g , are denoted by $L_0(x(t), p)$, respectively

$L_1(x(t), p)$, and simply by L_0 , respectively L_1 their evaluation at the initial state. The first order Lie derivatives are given below:

$$\begin{aligned} L_0(x(t), p) &= \sum_{i=1}^n f_i(x(t), p) \frac{\partial}{\partial x_i} h(x(t), p), \quad L_0 = L_0 h(x(t), p)|_{x_0(p)}, \\ L_1(x(t), p) &= \sum_{i=1}^n g_i(x(t), p) \frac{\partial}{\partial x_i} h(x(t), p), \quad L_1 = L_1 h(x(t), p)|_{x_0(p)}. \end{aligned} \quad (1.4)$$

A Lie derivative is of index 0 if it is computed with respect to the f function, and of index 1 if it is considered with respect to the function g .

Second order Lie derivatives. For the second iteration all possible combinations of 0 and 1 are computed, i.e. all combinations of Lie derivatives with respect to f and g of the first order Lie derivatives given in (1.4). These are:

$$\begin{aligned} L_{00}(x(t), p) &= L_0 L_0(x(t), p), \quad L_{00} = L_0 L_0(x(t), p)|_{x_0(p)}, \\ L_{01}(x(t), p) &= L_0 L_1(x(t), p), \quad L_{01} = L_0 L_1(x(t), p)|_{x_0(p)}, \\ L_{10}(x(t), p) &= L_1 L_0(x(t), p), \quad L_{10} = L_1 L_0(x(t), p)|_{x_0(p)}, \\ L_{11}(x(t), p) &= L_1 L_1(x(t), p), \quad L_{11} = L_1 L_1(x(t), p)|_{x_0(p)}. \end{aligned} \quad (1.5)$$

Third order Lie derivatives. The third iteration generates 8 Lie derivatives evaluated at the initial state:

$$L_{000}, L_{001}, L_{010}, L_{011}, L_{100}, L_{101}, L_{110}, L_{111}.$$

The algorithm continues in a similar way, up to the maximum number of derivatives given by the user in the main program.

Exhaustive summary contains all series coefficients, obtained by computing iterative Lie derivatives with respect to f and g , at the initial state $x(t_0) = x_0(p)$. These are given symbolically as

$$ES = \{L_0, L_1, L_{00}, L_{01}, L_{10}, L_{11}, L_{000}, \dots \text{and so on}\}. \quad (1.6)$$

If zero elements appear in ES , they are eliminated. Linear independent L_i will generate the set of relations needed to determine the parametric solution, if this is possible.

1.5 Identifiability tableau

An *identifiability tableau* is a way of expressing the dependencies between parameters, with the help of the exhaustive summary. The existence of parameters in each series coefficient is checked by considering the Jacobian of these coefficients with respect to the set of parameters.

Jacobian matrix of the exhaustive summary is calculated with respect to the parameters $p \in \mathbf{P}$ considered in the identifiability process. Thus, the Jacobian has as many columns as parameters, and as many rows as non-zero series coefficients.

Complete identifiability tableau. The Jacobian matrix described above is then represented by a "0-1" matrix, called *complete (extended) identifiability tableau*. An element at the (i, j) position in the identifiability tableau has value "1" if the non-zero series coefficient L_i depends on the parameter p_j , and "0" otherwise. A generic representation of such an identifiability tableau is given in Figure 1.1.

Examining the complex identifiability tableau, one of the following situations may appear:

- If there is a unique element in a row (just one "1" element), then the corresponding parameter is structurally identifiable.
- If a column of the Jacobian is empty, then the corresponding parameter is non-identifiable. In this situation the Jacobian is called *rank deficient* (see Figure 1.1).
- The number of successive iterative Lie derivatives is not known, thus the maximum rank of the Jacobian can't be fully guaranteed unless coefficients of higher order are found linear dependent to the previous ones, or zero, for successive derivatives.
- The procedure of generating new Lie derivatives is repeated iteratively until the rank of the Jacobian is equal to the number of parameters. If this condition is achieved, then a conclusion about structural global/local identifiable parameters can be given. If not, new rows (Jacobian of new Lie derivatives) should be added until the desired rank is obtained, if possible.

	p_1	p_2	p_3				p_{q_p}
L_1	0	0	0		0	0	1
L_2	0	0	1		0	0	0
	1	1	0		0	0	0
	1	1	0		0	1	0
L_i	0	0	0		0	0	0

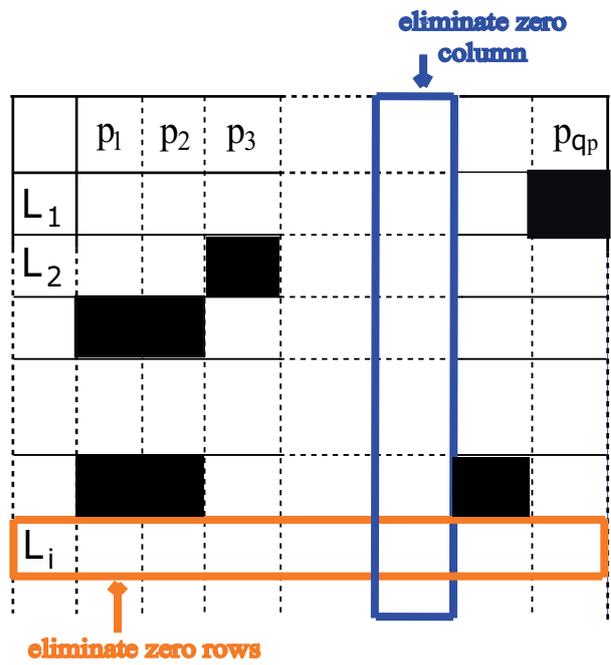


Figure 1.1: Identifiability tableau for the generating series method: (a) "0-1" matrix; (b) pseudocolor plot

	p_1	p_2	p_3		p_{i-1}
L_1	0	0	0		①
L_2	0	0	①		0
L_3	1	1	0		0
L_{i-1}	1	1	0		1

Figure 1.2: **Reduced identifiability tableau for the generating series method.** Green "1"s correspond to those parameters that can be uniquely identified, i.e. only one solution exist. Red "1"s represent possible identifiability problems, i.e. sets of parameters that require 2/more equations to be identified. Red boxes and arrows represent sets of equations that result in a unique solution for the parameters. The numbers represent the order in which the equations were solved.

- The next step is to find the linear independent relations needed to compute the solution for the set of parameters. This is done by considering the reduced identifiability tableaus. The second identifiability tableau was generated from the reduced identifiability tableau, just by eliminating the rows that represents the structurally globally identifiable parameters, and the columns corresponding to these parameters. The higher dimensional reduced identifiability tableaus are obtained in the following way. The algorithm seeks for those parameters that require 2 or more equations to be identified (represented by red boxes in Figure 1.2). If such a pair is found, the next reduced tableau is obtained by cutting off the corresponding rows and columns. The process repeats until no reduction can be done. If one solution is found, the parameters are structurally globally identifiable. If more than one positive solution exists, the parameters are locally identifiable and if they have no solution the parameters are not identifiable. At the end of the identifiability process there will be displayed the parameters that are structurally globally or locally identifiable and the structurally non-identifiable parameters, if they exist.

Remark 1 1. *If the rank of the Jacobian (Rank) is less than the number of parameters (NrParam), then a number of NrParam-Rank parameters are non-identifiable. In this situation an alternative is to fix NrParam-Rank parameters and repeat the whole process for the remaining ones, or to consider extra observables. But this last alternative is not convenient in practice.*

2. *Determining which parameters should be fixed is a problem that can be solved by using a global ranking procedure [4].*

1.7 How to chose the number of derivatives?

Choosing the number of derivatives needed to complete the computations is very important. This number is related with the rank of the identifiability tableau. If the identifiability tableau has maximum rank (equal to the number of parameters), the parameters will be at least structurally locally identifiable. Also a message is displayed when this situation occurs:

– > THE RANK OF THE FULL JACOBIAN IS COMPLETE, THUS AT LEAST LOCAL IDENTIFIABILITY IS GUARANTEED.

If the rank of the identifiability tableau is deficient for a number of derivatives, the user should increment the number of derivatives until a full rank identifiability tableau is achieved, if possible. In this situation the following message is displayed:

– >The rank of the full Jacobian matrix is 8
– >THE FULL JACOBIAN IS RANK DEFICIENT, YOU MAY CONSIDER ADDING NEW DERIVATIVES, Nder+1

If the full rank can't be achieved (because of the computational cost, in general) the columns of those parameters that cannot be identified are zero, so the parameters appear as structurally non-identifiable. But with extra derivatives some of these non-identifiable parameters may become structurally locally/globally identifiable. In this case, you may consider to perform a global sensitivity analysis to confirm on the lack of identifiability (use for example [2]).

So, In this case, you may consider to perform a global sensitivity analysis to confirm on the lack of identifiability. the number of derivatives should be chosen such as the rank of the identifiability tableau is equal to the number of the parameters. At least $Nobs * (1 + Noc)^{Nder}$ Lie derivative components are needed to achieve the maximum rank. But it is possible that extra derivatives may be needed if zero or dependent Lie derivatives components are generated, that is why after each iteration the expected rank value is given if an extra derivative is to be considered.

2

GenSSI software description

2.1 Toolbox requirements

The GenSSI toolbox runs under Windows, Linux or any other OS where MATLAB is available and requires the following:

- the Symbolic Math Toolbox for MATLAB versions up to R2007b+;
- the Maple Toolbox for MATLAB, for MATLAB version R2007b+ and later.

If you have the Symbolic Math Toolbox for MATLAB and you do not want to purchase the Maple Toolbox, please note that you can install a previous version of MATLAB (we suggest R2007b).

2.2 Download and License

In order to install the GenSSI Toolbox you need to:

- Unzip the archive .zip (in the C: directory);
- Start MATLAB and locate the GenSSI unzipped folder;
- Every time you want to use GenSSI type

>GenSSI.Startup

GenSSI module is written in MATLAB, and uses Symbolic Math Toolbox for symbolic calculus. The downloading and documentation are available at the website

<http://www.iim.csic.es/~genssi>,
<http://www.iim.csic.es/%7Egenssi>

2.3 GenSSI features

GenSSI software contains a set of functions that encode all the steps explained theoretically in the first chapter, examples and a folder having the documentation and the user guide. The outstandings of this module are:

- Better manipulation of structural identifiability problem, from computational and flexibility point of view. The user can manage better the whole identifiability process.
- It doesn't provide "all or nothing" information; it generates information results at each step.
- Lack of memory problems can be handled by the user, by manipulating the derivation order.

The features are summarized in the description below:

Models GenSSI allows any general linear or non-linear model, written as ordinary differential equations (ODEs), or differential algebraic equations (DAEs), of the form (1.1).

Input functions can be any linear or non-linear analytic functions, in the most general way, depending on state variables and parameters. They can also be characterized by high degree of non-linearity, Hill coefficients or Michaelis-Menten functions.

Output functions are analytic functions depending on state variables of the model. The models with more output functions variables are easier to analyze, and also the computational cost is smaller.

Parameters of the model are declared as symbolic elements. In the identifiability process there can be employed the whole set of parameters that characterizes the model, or just a subset for testing local identifiability.

Initial conditions can be known or unknown. If the initial conditions are unknown, they are declared symbolic. In general, if initial conditions are known, the computational cost decreases.

2.4 GenSSI functions

The steps explained in Chapter 1 are implemented in the following functions.

Main functions (called by the user):

>**GenSSI.Startup** - this function should be called once, each time the user initiates a Matlab session where the GenSSI toolbox is going to be used;

GenSSI_generating_series(F,G,Noc,Neq,Nder,Nobs,H,X,IC,Par,result_folder) - this is the main executable function of GenSSI, which has the following input parameters:

F = the vector containing the uncontrolled components of the model (denoted by f in our theoretical description, and $F=[A1 A2 A3 \dots]$ in each example);

G = the vector representing the controlled part of the model (denoted by g in (1.1) and $G=[g1 g2 g3 \dots]$ for each example);

Noc = a number defining the number of controls (denoted by m in subsection 1.1);

Neq = gives the number of equations (equal to the number of state variables);

Nder = the number of derivatives the user wants to consider;

Nobs = the experimentally observed quantities (in (1.1) denoted by r);

H = the vector containing all the observed quantities depending on the state variables (in subsection 1.1 it is denoted by h and in the implemented example by $H=[h1 h2 h3 \dots]$)

X = the vector containing the state variables denoted by $X=[x_1 x_2 x_3 \dots]$;
IC = the vector of the initial conditions, represented by $x_0(p)$ in the theoretical part and by **IC** in the implemented example;
Par = the parameters p are included into a vector denoted by **Par**;
result_folder = the folder where all the results are saved. This folder should be given by the user at the beginning of each program.

Auxiliary functions

GenSSI_report_inputs - displays the information about the model, that was provided by the user in the main program. This information contains:

- the name of the model for which the identifiability test is being done;
- maximum number of derivatives for the analysis;
- dynamic model;
- control variables;
- observables;
- initial conditions;
- parameters to be considered in the analysis.

GenSSI_computes_LieDerivatives - computes iterative Lie derivatives, taking into consideration the following steps:

- computes as many iterations as the number of derivatives given by the user;
- arrange the components of Lie derivatives after each iteration in a vector, representing the exhaustive summary of the model;
- after each derivative, it is displayed the rank of the correspondent Jacobian, and also some information about the next one. If the rank is equal to the number of parameters (providing at least local identifiability) the program won't consider an extra derivative. If the rank is deficient, then a message will be generated with the expected rank for the next derivative;
- eliminates zero-rows, if such rows exist.

GenSSI_computes_tableau - generates identifiability tableaus.

- computes the Jacobian of the non zero generated series coefficients with respect to the parameters of the model;
- eliminates the zero rows of the Jacobian constructed above, if they exist;
- computes the rank of the full Jacobian and displays if it is complete or deficient. If it is deficient, then a message will be displayed, suggesting to add new derivatives. If it is complete, the displayed message will tell that at least local identifiability can be achieved;
- the extended Jacobian is represented as a "0-1" matrix, and then graphically as a black (for "1" elements) and white (for "0" element) squares, see Figure 1.1 (b).

GenSSI_computes_reduced_tableau - gives the following results:

- stores the first linear independent rows, such that the rank of the reduced tableau remains unchanged;
- constructs the "0-1" reduced tableau;
- represents graphically the "0-1" reduced identifiability tableau as a pseudocolor plot;
- displays the set of parameters and the relations (Lie derivatives evaluated at the initial state) needed to compute the solution of the parameters.

GenSSI_orders_tableau - orders the reduced identifiability tableau for the identifiability test:

- computes the sum of elements for each row of the "0-1" reduced matrix. If there is a row with the sum 1, the parameter corresponding to the value "1" is structurally globally identifiable;
- eliminates the rows and columns corresponding to the globally identifiable parameters, generating the second order reduced identifiability tableau;
- construct the new arranged "0-1" tableau, and represents it graphically;

- computes higher order reduced identifiability tableau(s): displays the group of 2/more depending parameters, the associated algebraic relations and the corresponding solution(s);
- calculate the solution(s) of the remaining parameters. If the 3rd reduced identifiability tableau can be reduced again, then another group of parameters will be consider in order to calculate the solution, and the process repeats until no reduction can be done;
- if the 3rd identifiability tableau can't be reduced anymore (there are no groups of 2 or more depending parameters, illustrated by the same number of rows), then the solution is calculated at once.

GenSSI_report_results - displays the identifiability test conclusions:

- structurally globally identifiable parameters, structurally non identifiable parameters (directly determined), structurally locally identifiable parameters; report (model_information.txt)
- the identifiability tableaus;
- all the generated results are copied into a .txt file called model_information.txt. This file together with the figures are stored in the Results folder.

3

Illustrative examples

GenSSI software is easy to use, and its functionality will be presented/explained for three examples. The steps of the algorithm are given, and the results interpreted.

- **Goodwin oscillator [11]**

- Analyze the hypothetical situation with all states measured, and compare with the case of one measured variable. This is an example of a dynamical model without control.

- ***Arabidopsis thaliana* model [14]**

- It is given the description of the model, characterized by one control. A set/subset of parameters is considered for studying the global/local structural identifiability, and all the reduced identifiability tableaus are analyzed.

- **Glycolysis inspired metabolic pathway [6]**

- It is an example of a totally controlled model, described by four controls.

3.1 Goodwin oscillator

The model describes the oscillations in enzyme kinetics [11]. The variable x_1 represents an enzyme concentration, whose rate of synthesis is regulated by feedback control via a metabolite x_3 , and x_2 regulates the synthesis of x_3 .

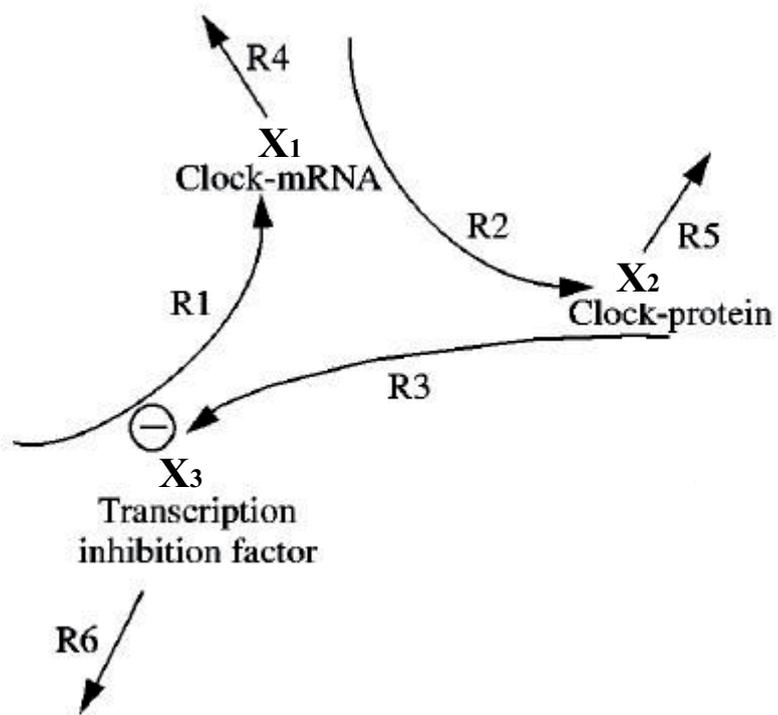


Figure 3.1: **The Goodwin model.** Reaction R1 is the formation of clock mRNA (x_1); reaction R2 is the synthesis of clock protein (x_2) and R3 is the production of a transcription inhibitor (x_3). R4, R5 and R6 represent degradation reactions.

The model is described by a set of coupled deterministic ordinary differential equations, able to simulate physiological oscillations on the basis of a negative feedback where a protein inhibits the transcription of its own gene, with the inhibition factor given by $\frac{p_1}{p_2+x_3^{p_3}}$. The mathematical description is given in (3.1):

$$\begin{cases} \dot{x}_1 = -p_4x_1 + \frac{p_1}{p_2+x_3^{p_3}}, \\ \dot{x}_2 = p_5x_1 - p_6x_2, \\ \dot{x}_3 = p_7x_2 - p_8x_3. \end{cases} \quad (3.1)$$

Each model is described by the symbolic elements, model related data, parameters and the function that does identifiability analysis for the considered model. These characteristics will be illustrated here.

- Symbolic variables: for this example they are x_1 x_2 x_3 p_1 p_2 p_3 p_4 p_5 p_6 p_7 p_8 ;
- Model related data: number of derivatives, number of states, number of parameters, number of controls, number of observables, and their expressions;
- Parameters considered for identifiability: can be considered the whole set of parameters, or just a subset for studying local identifiability (here the parameters are p_1 p_2 p_3 p_4 p_5 p_6 p_7 p_8);
- Call the generating series function to run the identifiability test for the model;

We will give the results for the case when all the states are measured ($y_1 = x_1, y_2 = x_2, y_3 = x_3$). The purpose of considering this case is to emphasize the manner of choosing the required number of derivatives, and to emphasize the way the number of observables influences the computations.

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   DECLARE SYMBOLIC VARIABLES:   %
%   - state variables             %
%   - parameters of the model    %
%   -initial state, if not known %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

syms x1 x2 x3 p1 p2 p3 p4 p5 p6 p7 p8

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   MODEL RELATED DATA          %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

Nder=4;                               % Number of derivatives

Neq=3;                                 % Number of states
X=[x1 x2 x3];

Npar=8;                                % Number of model parameters

Noc=0;                                 % Number of controls

Nobs=3;                                % Number of observables

A1 = -p4*x1+p1/(p2+x3^p3);             % Equations of the model
A2 = p5*x1-p6*x2;
A3 = p7*x2-p8*x3;
F=[A1 A2 A3];

g1=0;g2=0;g3=0;                        % Controls
G=[g1 g2 g3];

h1=x1;h2=x2;h3=x3;                     % Observables
H=[h1 h2 h3];

IC=[0.3 0.9 1.3];                      % Initial conditions

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   PARAMETERS CONSIDERED FOR IDENTIFIABILITY %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

Par=[p1 p2 p3 p4 p5 p6 p7 p8];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   GENERATING SERIES FUNCTION %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

GenSSI_generating_series(F,G,Noc,Neq,Nder,Nobs,H,X,IC,Par,result_folder);

```

Rank condition. For more observables, the required number of derivatives is smaller than in the other scenario, because more outputs generate more Lie

derivative components. The model is characterized by 8 parameters, therefore at least 8 non-zero Lie derivatives are needed. In the table below, we illustrate the Lie derivatives generated and the number of relations at each derivation step.

Nder	Lie derivatives		NLrel
	with respect to f	with respect to g	
1	$L_0 = (L_{0_1}, L_{0_2}, L_{0_3})$	$L_1 = (L_{1_1}, L_{1_2}, L_{1_3})$	6
2	$L_{00} = (L_{00_1}, L_{00_2}, L_{00_3})$ $L_{01} = (L_{01_1}, L_{01_2}, L_{01_3})$	$L_{10} = (L_{10_1}, L_{10_2}, L_{10_3})$ $L_{11} = (L_{11_1}, L_{11_2}, L_{11_3})$	12
3	$L_{000} = (L_{000_1}, L_{000_2}, L_{000_3})$ $L_{001} = (L_{001_1}, L_{001_2}, L_{001_3})$ $L_{010} = (L_{010_1}, L_{010_2}, L_{010_3})$ $L_{011} = (L_{011_1}, L_{011_2}, L_{011_3})$	$L_{100} = (L_{100_1}, L_{100_2}, L_{100_3})$ $L_{101} = (L_{101_1}, L_{101_2}, L_{101_3})$ $L_{110} = (L_{110_1}, L_{110_2}, L_{110_3})$ $L_{111} = (L_{111_1}, L_{111_2}, L_{111_3})$	24
	and so on		

NLrel denotes the number of Lie derivatives and their components (relations between parameters), and Nder the number of derivatives (iterations).

It is also very important to point out that as our example is not controlled, the number of Lie derivatives components is smaller, in the case of 3 derivatives we have 3 nonzero Lie derivatives, and so 12 relations (number of derivatives \times number of observables).

If 3 derivatives are considered for computing the rank of the complete Jacobian, 9 relations between parameters will be generated. Among these relations zero or dependent relations may be found. After each derivative, GenSSI generates a message with the rank of the actual Jacobian, and also a message with the expected value that may be generated if another derivative is given. If the rank is equal to the number of the parameters, then the computation is interrupted and passes to the following step. In this case the rank is not full, and the following output is displayed:

```

*****
-> COMPUTES LIE DERIVATIVES
*****

COMPUTING LIE DERIVATIVES OF ORDER 1
.....
->The rank of the Jacobian generated by 1 derivative is 3.
.....

COMPUTING LIE DERIVATIVES OF ORDER 2
.....
->The rank of the Jacobian generated by 2 derivatives is 6.
->The rank of the next Jacobian is expected to be maximum 8.
.....

COMPUTING LIE DERIVATIVES OF ORDER 3
.....
->The rank of the Jacobian generated by 3 derivatives is 7.
->The rank of the next Jacobian is expected to be maximum 8.
.....

COMPUTING LIE DERIVATIVES OF ORDER 4
.....
->The rank of the Jacobian generated by 4 derivatives is 8.

*****
-> COMPUTES IDENTIFIABILITY TABLEAU
*****

----->The rank of the full Jacobian matrix is 8

----->THE RANK OF THE FULL JACOBIAN IS COMPLETE, THUS AT LEAST LOCAL
IDENTIFIABILITY IS GUARANTEED.

```

For only 3 derivatives it can be seen that the rank is 7, so rank deficient. In this case GenSSI will generate a warning message asking for considering an

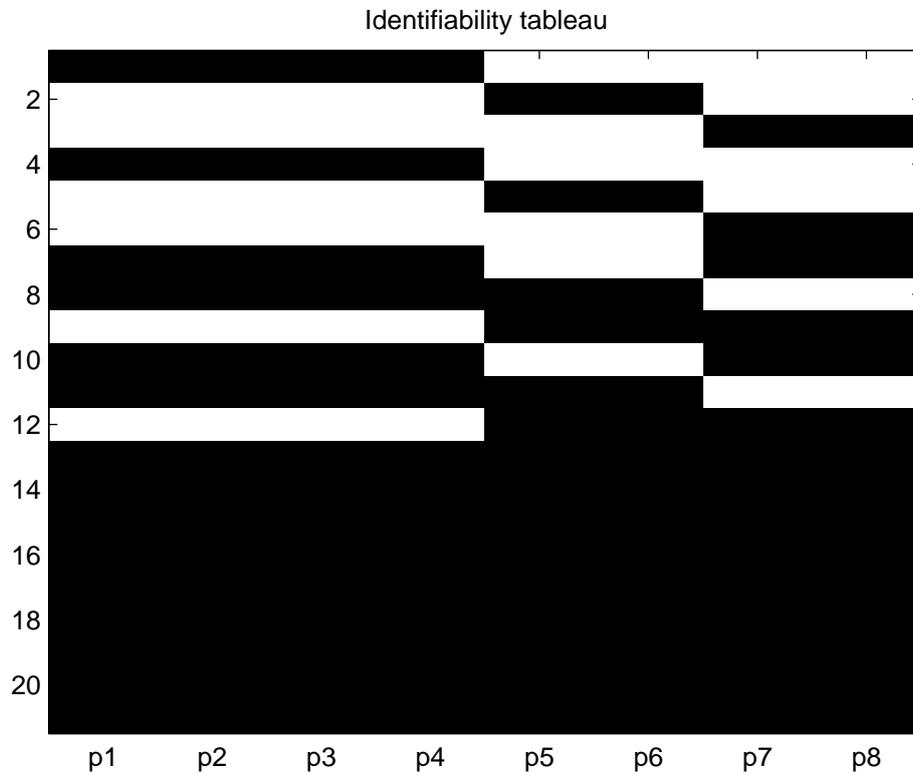


Figure 3.2: **Complete identifiability tableau for Goodwin model with 3 outputs**

extra derivative. Using 4 iterations, the Lie derivatives generate a complete Jacobian of rank 8, whose graphical representation is given in Figure 3.2.

If the model has just one measurable state, then the number of derivatives needed is higher. In this case at each step only one relation is generated, and so, for 8 parameters 8 derivatives should be done, and the computational cost is visible higher, too. In this particular case the program couldn't finish the computation,

```

??? Error using ==> sym.maple
Error, (in expand/bigprod) integer too large in context

```

Reduced identifiability tableau. The representation of the reduced identifiability tableau is shown in Figure 3.3. GenSSI selects the first 8 rows linear

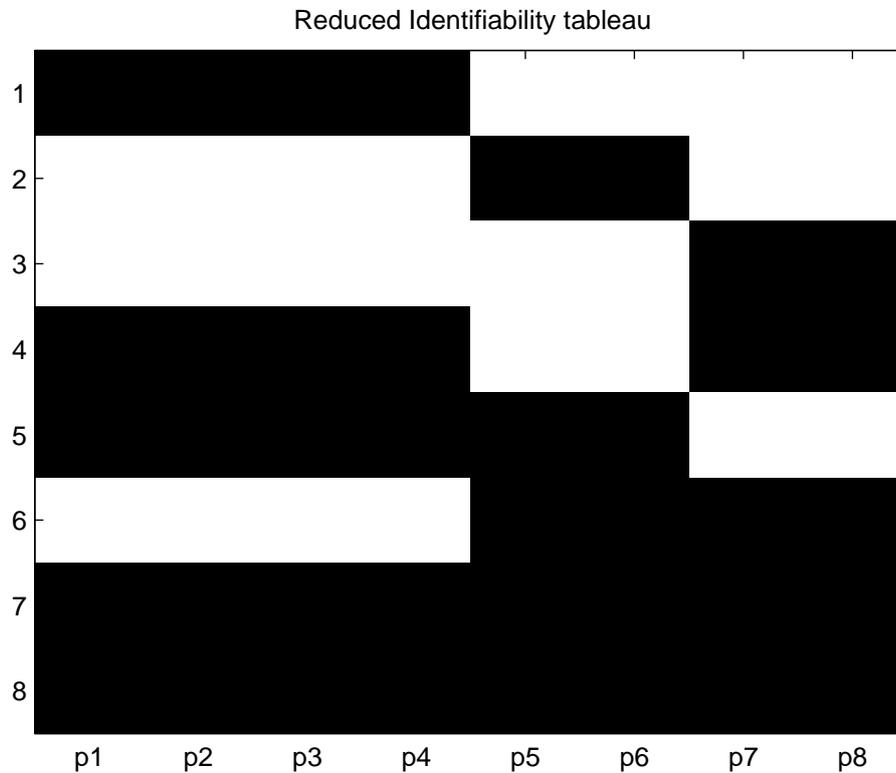


Figure 3.3: **Reduced identifiability tableau for Goodwin model with 3 outputs**

independent such that the rank of the reduced tableau remains 8. We try to generate a reduced identifiability tableau. There are no structurally globally identifiable parameters determined directly, so the second reduced identifiability tableau is like the first one. The next step is to look for rows that represent 2/more parameters. GenSSI couldn't find such rows, so this identifiability tableau can't be reduced.

The solution. If all the parameters are considered, no result was given because of the complexity, as a Hill coefficient appears (p3). By fixing the parameter $p3=10$, the GenSSI gives the relations needed to compute the parameters, and finds a double solution, as presented below. The computational cost needed to finish is 2.0156s.

In case of non-identifiability, one of the two approaches may be considered:

- fix a subset of parameters - this is mostly used if the model was found non-identifiable, and local structural identifiability for a subset of parameters may give information, at least locally;
- consider extra observables - this manner is very helpful from computation point of view, but it is not convenient in practice.

Each of these can be used when the user gets into a deadlock because of the impossibility of finding the maximum rank of the Jacobian (equal to the total number of parameters), or because after computing the values of the parameters, some of them are not real, or no solution could be found.

1. It is important to mention that if the parameter p_3 is not fixed, the identifiability tableaus are generated, also the algebraic relations needed to compute the solution, but we couldn't find a solution for p_3 in an algebraic way. A particular case of the model (3.1) is when $p_3=10$, as given in [5]. By fixing this parameter at least local identifiability can be obtain.

2. If $p_3=10$ and p_2 is fixed, the results for Goodwin model totally observable are:

- the rank condition is satisfied for $N_{der}=2$;
- all the parameters are structurally identifiable;
- no non-identifiable parameter was displayed;
- the first reduced tableau is given in Figure 3.4.

The results generated by GenSSI are given in the following:

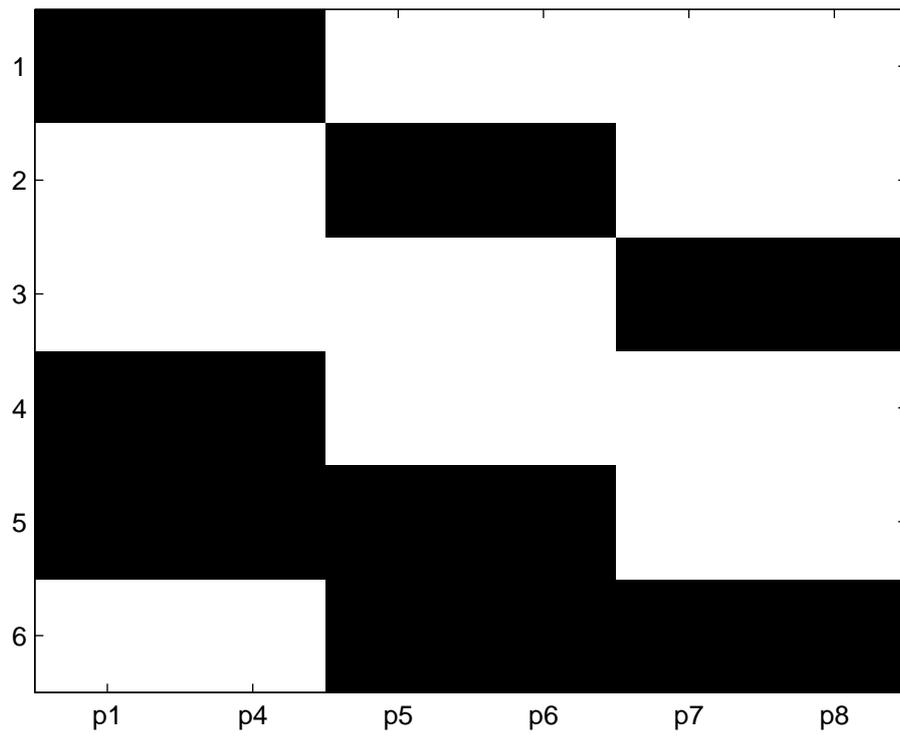


Figure 3.4: **Reduced identifiability tableau for (3.1), $p_3=10$, p_2 =fixed**

----> THE SYMBOLIC SOLUTION OF THE REMAINING PARAMETERS:

Solution =

p1: [2x1 sym]
p2: [2x1 sym]
p4: [2x1 sym]
p5: [2x1 sym]
p6: [2x1 sym]
p7: [2x1 sym]
p8: [2x1 sym]

----> THE MODEL IS STRUCTURALLY LOCALLY IDENTIFIABLE

The structurally globally identifiable parameters are:

None

The structurally locally identifiable parameters are:

[p1 p2 p4 p5 p6 p7 p8]

The structurally non-identifiable parameters are:

None

3.2 *Arabidopsis thaliana* model

A second example is given by a circadian network, also called "biological clock". It is of great importance, as it helps signaling the organism about changing cycles in the environment, such as daily light/dark, or warm/cold sensations. These characteristics of circadian rhythms have been found in almost all eukaryotic organisms and in some prokaryotes, controlling processes from cyanobacterial cell division to human sleep-wake cycles [9].

The model chosen for study is a minimal mathematical representation of such a biological process and requires 7 coupled differential equations and 27 parameters. It describes the first multi-gene loop identified in the *Arabidopsis* circadian clock [14], that comprises a negative feedback loop, in which two partially redundant genes Late Elongated Hypocotyl (LHY) and Circadian Clock Associated 1 (CCA1) repress the expression of their activator, Timing of CAB Expression 1 (TOC1). The mathematical representation is given in [14]:

$$\begin{cases} \dot{x}_1 = p_1 \frac{x_6}{p_3+x_6} - p_5 \frac{x_1}{p_{12}+x_1} + p_{26}x_7u(t), \\ \dot{x}_2 = p_{19}x_1 - p_{22}x_2 + p_{23}x_3 - p_6 \frac{x_2}{p_{13}+x_2}, \\ \dot{x}_3 = p_{22}x_2 - p_{23}x_3 - p_7 \frac{x_3}{p_{14}+x_3}, \\ \dot{x}_4 = p_2 \frac{p_4^2}{p_4^2+x_3^2} - p_8 \frac{x_4}{p_{15}+x_4}, \\ \dot{x}_5 = p_{20}x_4 - p_{24}x_5 + p_{25}x_6 - p_9 \frac{x_5}{p_{16}+x_5}, \\ \dot{x}_6 = p_{24}x_5 - p_{25}x_6 - p_{10} \frac{x_6}{p_{17}+x_6}, \\ \dot{x}_7 = p_{21} - p_{11} \frac{x_7}{p_{18}+x_7} - (p_{21} + p_{27}x_7)u(t), \end{cases} \quad (3.2)$$

with the measured states $y_1 = x_1$ and $y_2 = x_4$ [14].

The differential equations involve Michaelis-Menten kinetics that describe enzyme-mediated protein degradation, and Hill functions representing transcriptional activation terms.

The input data, given by the user for this model is presented below, as GenSSI displays it, taking into consideration that the software works with vectorial representation of the inputted data.

The model is characterized by 7 state variable and 27 parameters. All these are declared symbolic. In the model related data space is given all the information about the model: parameters that are involved in the identifiability process, the uncontrolled part (F), the controls (G), the observables (H). If state variable is not observable, then the corresponding component in H is set to 0.

The outputs of the GenSSI software are the identifiability tableaus displayed

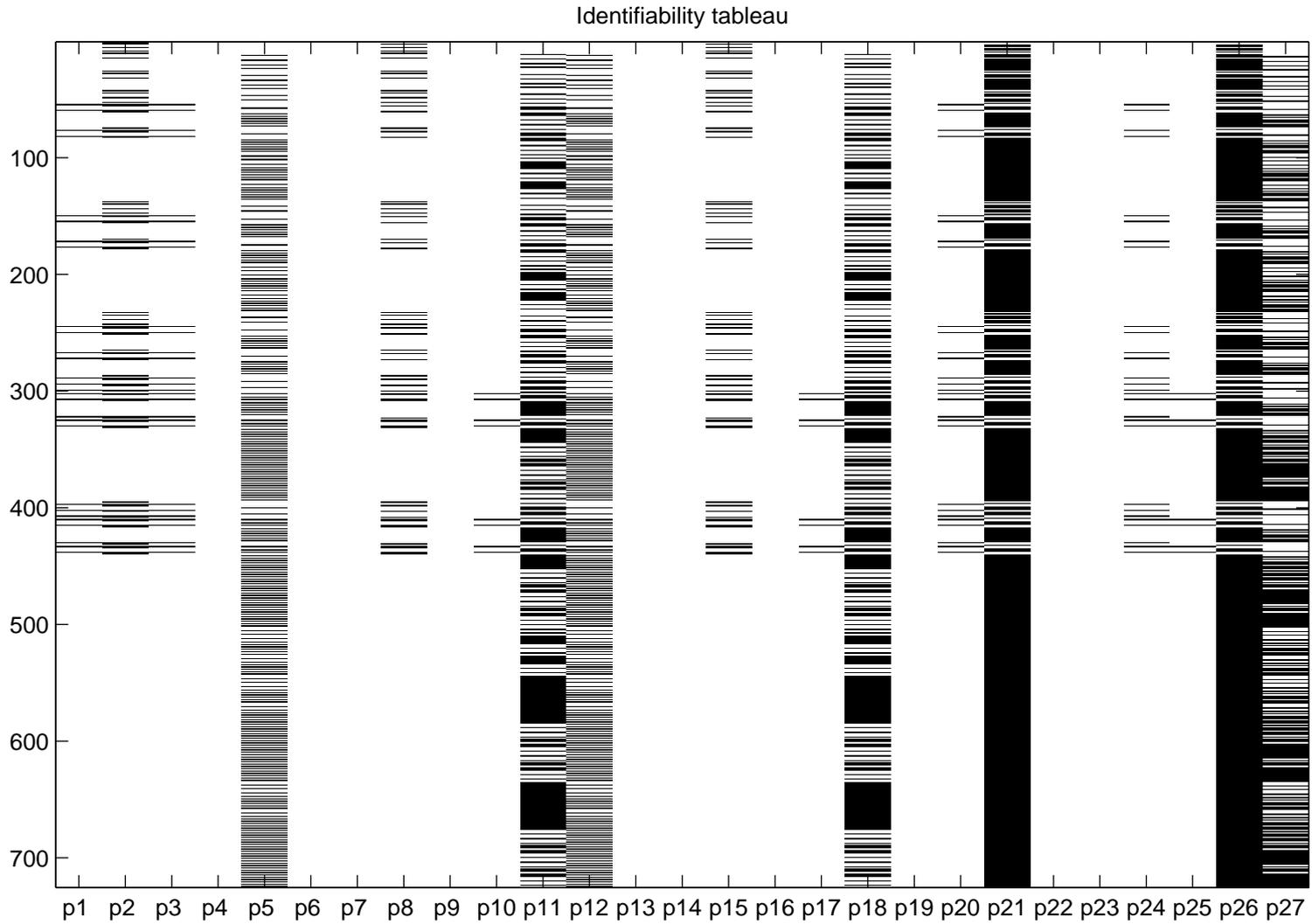


Figure 3.5: Complete identifiability tableau for *Arabidopsis thaliana* model (27 parameters and 5 derivatives)

as figures, and information printed on the screed, as given below for the model (3.2) with 27 parameters.

```

*****
-> COMPUTES LIE DERIVATIVES
*****

COMPUTING LIE DERIVATIVES OF ORDER 1
.....
->The rank of the Jacobian generated by 1 derivative is 1.
.....

COMPUTING LIE DERIVATIVES OF ORDER 2
.....
->The rank of the Jacobian generated by 2 derivatives is 3.
->The rank of the next Jacobian is expected to be maximum 5.
.....

COMPUTING LIE DERIVATIVES OF ORDER 3
.....
->The rank of the Jacobian generated by 3 derivatives is 7.
->The rank of the next Jacobian is expected to be maximum 11.
.....

COMPUTING LIE DERIVATIVES OF ORDER 4
.....
->The rank of the Jacobian generated by 4 derivatives is 9.
->The rank of the next Jacobian is expected to be maximum 11.
.....

COMPUTING LIE DERIVATIVES OF ORDER 5
.....
->The rank of the Jacobian generated by 5 derivatives is 11.
->The rank of the next Jacobian is expected to be maximum 13.
.....

*****
-> COMPUTES IDENTIFIABILITY TABLEAU
*****

----->The rank of the full Jacobian matrix is 11

----->THE FULL JACOBIAN IS RANK DEFICIENT, YOU MAY CONSIDER ADDING NEW
DERIVATIVES, 6

*****
-> COMPUTES REDUCED IDENTIFIABILITY TABLEAUS
*****

```

- After each iteration the corresponding rank is displayed. If we begin by considering 4 derivatives, the rank is only 9, so we try with higher order

derivatives.

- After 5 derivatives, the identifiability tableau is given in Figure 3.5 for all 27 parameters. The reduced identifiability tableau is given in Figure 3.6. the computational cost for this case is 1.0602e+003s.
- The rank of the full Jacobian is deficient, as it is different from the number of parameters, so GenSSI suggests adding an extra derivative.

We may consider one of the following scenarios:

1. consider higher number of derivatives: $N_{der} > 5$. In this case the computational cost is higher for each extra derivative.
 2. fix the non-identifiable parameters and repeat the procedure, the computational cost may be smaller.
 3. fix 16 parameters, such that the reduced tableau has 11 linear independent rows and 11 parameters. This is a way for studying local identifiability for a group of parameters, the other being fixed.
 4. consider more observables.
- the reduced identifiability tableau is graphically represented in Figure 3.6. It is made of 11 independent non-zero rows and 27 columns. In this case one of the above scenarios can be used.
 - in Figure 3.5 can be seen that the identifiability tableau has zero columns. These columns correspond to non-identifiable parameters. If more derivatives are considered, some of these parameters may be identifiable. One parameter could be computed directly, if in the identifiability tableau one row has just one black square, and this is structurally globally identifiable.

In general, structural global identifiability of a complex model, characterized by a great number of parameters, is quite hard, if not impossible to determine. *Arabidopsis thaliana* model was found non-identifiable if the entire set of parameters was considered. In such a situation, it is advisable to check for structural identifiability of a subset of parameters, i.e. *local structural identifiability*.

The suitable parameters for the study of local identifiability are the most significant ones. As presented above, the rank of the Jacobian is 11, thus a subset of 11 parameters will be needed, or a subset of 16 parameters will be fixed. The technique that can be used for finding which parameters are more sensitive is based on

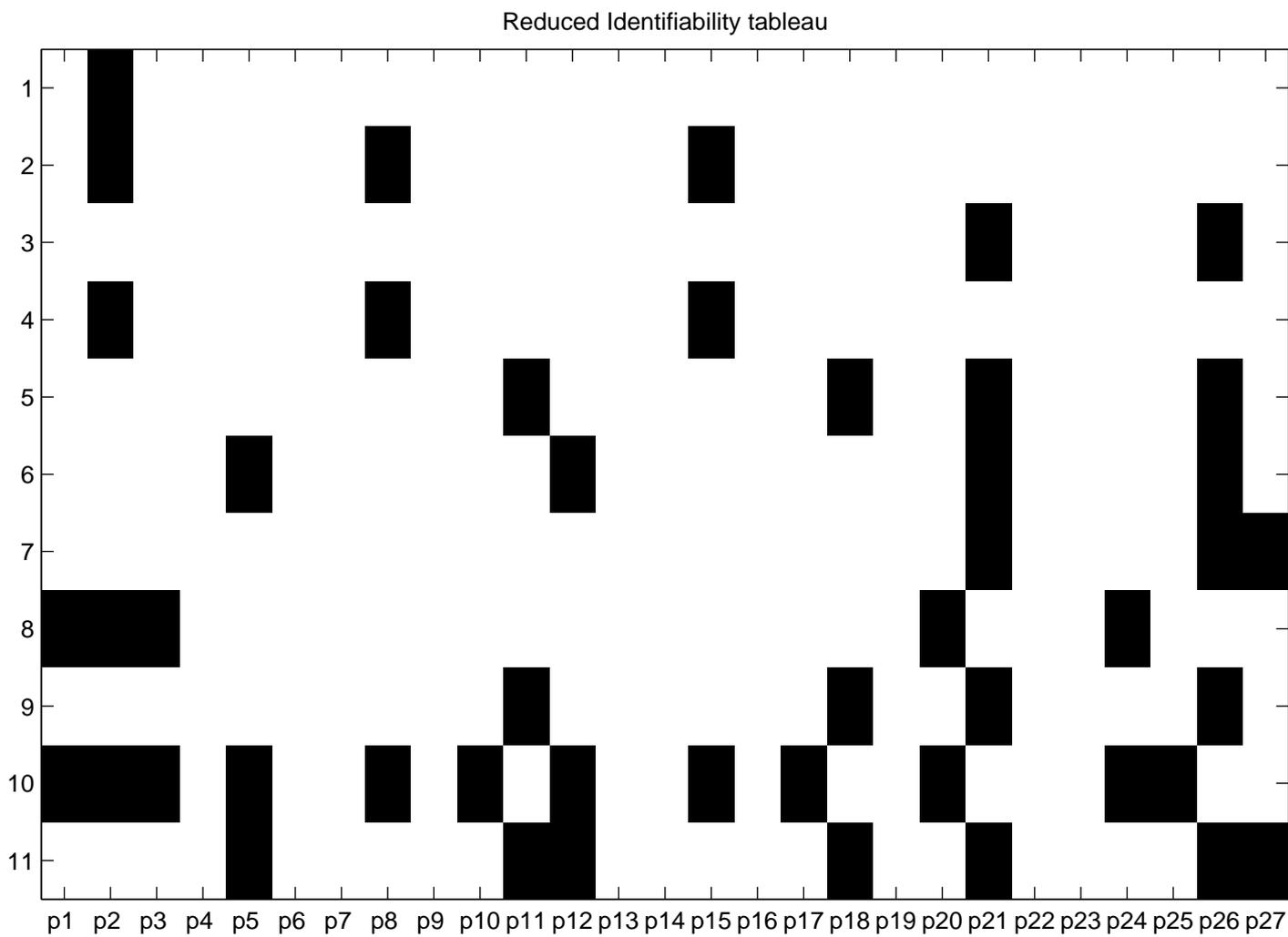


Figure 3.6: **Reduced identifiability tableau for *Arabidopsis thaliana* model (27 parameters)**

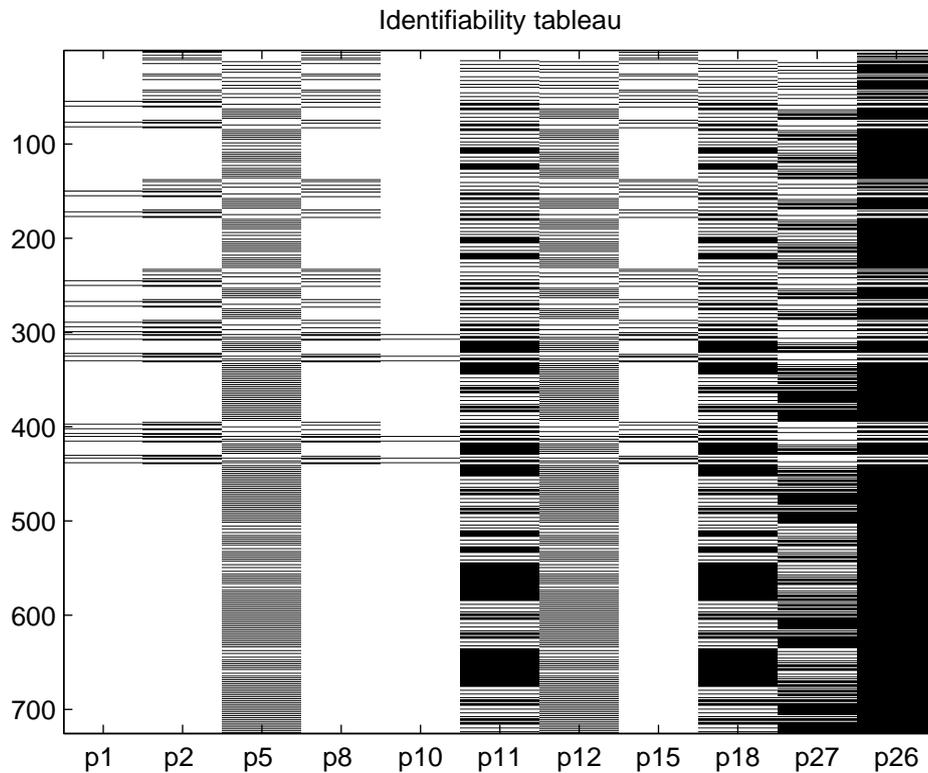


Figure 3.7: **Complete identifiability tableau for *Arabidopsis thaliana* model (11 parameters)**

global ranking procedure that is used for ranking the parameters in the order of their relative influence on the model prediction, using AMIGO toolbox [2]. This aspect is presented more detailed in [4].

To illustrate the way GenSSI works, we consider here a subset of 11 parameters. The output and the identifiability tableaus are presented.

- the rank of the Jacobian computed with respect to the subset of the 11 parameters is 11:

```

----> Parameters to be considered in the analysis:

[p1  p2  p5  p8  p10  p11  p12  p15  p18  p27  p26  ]

*****
-> COMPUTES LIE DERIVATIVES
*****

COMPUTING LIE DERIVATIVES OF ORDER 1
.....
->The rank of the Jacobian generated by 1 derivative is 1.
.....

COMPUTING LIE DERIVATIVES OF ORDER 2
.....
->The rank of the Jacobian generated by 2 derivatives is 3.
->The rank of the next Jacobian is expected to be maximum 5.
.....

COMPUTING LIE DERIVATIVES OF ORDER 3
.....
->The rank of the Jacobian generated by 3 derivatives is 7.
->The rank of the next Jacobian is expected to be maximum 11.
.....

COMPUTING LIE DERIVATIVES OF ORDER 4
.....
->The rank of the Jacobian generated by 4 derivatives is 9.
->The rank of the next Jacobian is expected to be maximum 11.
.....

COMPUTING LIE DERIVATIVES OF ORDER 5
.....
->The rank of the Jacobian generated by 5 derivatives is 11.

---->The rank of the full Jacobian matrix is 11

---->THE RANK OF THE FULL JACOBIAN IS COMPLETE, THUS AT LEAST LOCAL
IDENTIFIABILITY IS GUARANTEED.

```

- the complete identifiability tableau, for the parameters is graphically represented in Figure 3.7;

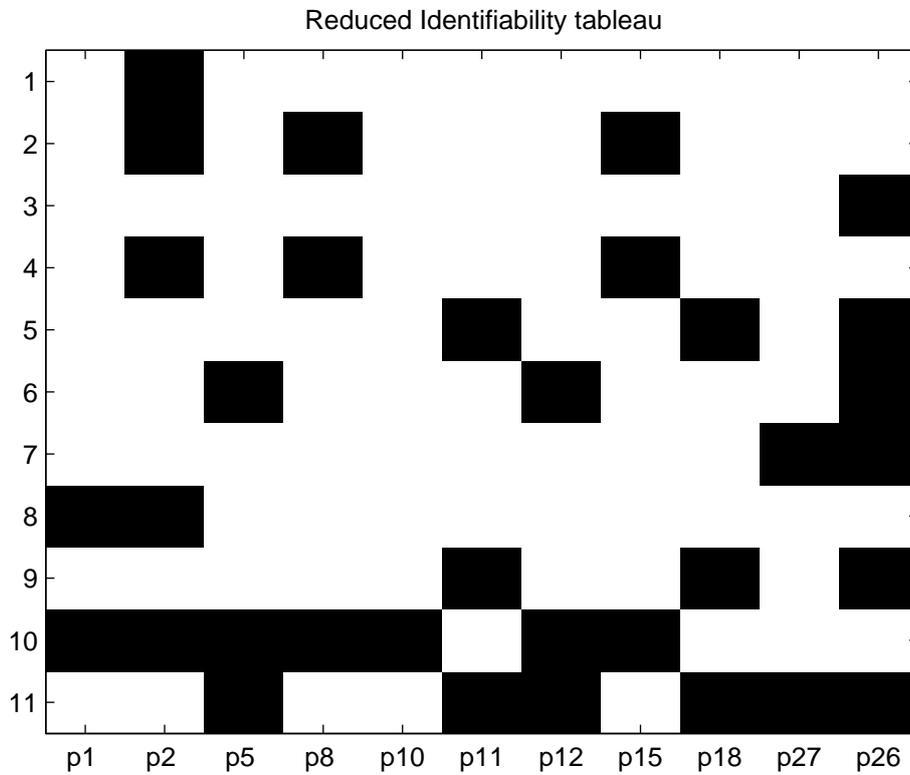


Figure 3.8: **Reduced identifiability tableau for *Arabidopsis thaliana* model (11 parameters)**

- the reduced identifiability tableau is illustrated in Figure 3.8;
- there are identifiable parameters given as (direct) global identifiable parameters: p2, p26, p27, p1;

```

*****
-> THE RELATIONS NEEDED FOR COMPUTING THE PARAMETERS
*****

p2-c1
-p2*p8/p15-c2
p21*p26-c3
p2*(p8^2/p15^2+2*p2*p8/p15^2)-c4
-p21*p11/p18*p26-c5
-p21*p26*p5/p12-c6
-p21*p27*p26-c7
p2*p20*p24*p1/p3-c8
p21*(p11^2/p18^2*p26+2*p21*p11/p18^2*p26)-c9
p2*(-p8/p15*p20*p24*p1/p3+p20*(-p24^2*p1/p3+p24*(-
p5/p12*p1/p3+(-p25-p10/p17)*p1/p3)))-c10
p21*(-p11/p18*p27*p26*p5/p12+4*p21*p26^2*p5/p12^2)-c11

*****
-> DETECTS (DIRECT) STRUCTURALLY GLOBALLY IDENTIFIABLE PARAMETERS AND
REORGANIZES TABLEAU
*****
-> STRUCTURALLY GLOBALLY IDENTIFIABLE PARAMETERS DETERMINED DIRECTLY
(parameters corresponding to one non-zero element in the reduced
identifiability tableau)

---->The parameter p2 is structurally globally identifiable. It has the solution:
p2= c1.
---->The parameter p26 is structurally globally identifiable. It has the solution:
p26= c3/p21.
---->The parameter p27 is structurally globally identifiable. It has the solution:
p27= -c7/p21/p26.
---->The parameter p1 is structurally globally identifiable. It has the solution:
p1= c8/p2/p20/p24*p3.

*****
->THE REMAINING PARAMETERS (APART FROM IDENTIFIABLE OR NON-
IDENTIFIABLE), AND THE CORRESPONDING RELATIONS
*****
----> Parameters:
[ p5, p8, p10, p11, p12, p15, p18]

----> Relations:
-p2*p8/p15-c2
p2*(p8^2/p15^2+2*p2*p8/p15^2)-c4
-p21*p11/p18*p26-c5
-p21*p26*p5/p12-c6
p21*(p11^2/p18^2*p26+2*p21*p11/p18^2*p26)-c9
p2*(-p8/p15*p20*p24*p1/p3+p20*(-p24^2*p1/p3+p24*(-
p5/p12*p1/p3+(-p25-p10/p17)*p1/p3)))-c10
p21*(-p11/p18*p27*p26*p5/p12+4*p21*p26^2*p5/p12^2)-c11

```

- displays the remaining relations and parameters;
- computes higher order reduced identifiability tableau: the group of param-

eters is displayed, the corresponding relations and the solution. The third identifiability tableau is generated by eliminating the first two rows and the columns of parameters p8 and p15.

```

*****
-> COMPUTES HIGHER ORDER REDUCED IDENTIFIABILITY TABLEAU(S)
  (displays the group of 2/more depending parameters,
   the associated algebraic relations,
   the corresponding solution (solutions))
*****

----> The group of parameters to be considered in the calculus and the corresponding relations:

-> Parameters:
    [p8  p15  ]

-> Relations:
                                     -p2*p8/p15-c2
    p2*(p8^2/p15^2+2*p2*p8/p15^2)-c4

Solution =
    p15: [1x1 sym]
    p8:  [1x1 sym]

-> The parameter p8 has the solution/solutions:
2*p2^2*c2/(c2^2-c4*p2)

-> The parameter p15 has the solution/solutions:
-2*p2*c2^2/(c2^2-c4*p2)
.....
----> The group of parameters to be considered in the calculus and the corresponding relations:

-> Parameters:
    [p11  p18  ]

-> Relations:
                                     -p21*p11/p18*p26-c5
    p21*(p11^2/p18^2*p26+2*p21*p11/p18^2*p26)-c9

Solution =
    p11: [1x1 sym]
    p18: [1x1 sym]

-> The parameter p11 has the solution/solutions:
2*p21*c5^2/(-c5^2+c9*p21*p26)

-> The parameter p18 has the solution/solutions:
-2*p21^2*c5/(-c5^2+c9*p21*p26)*p26
.....

```

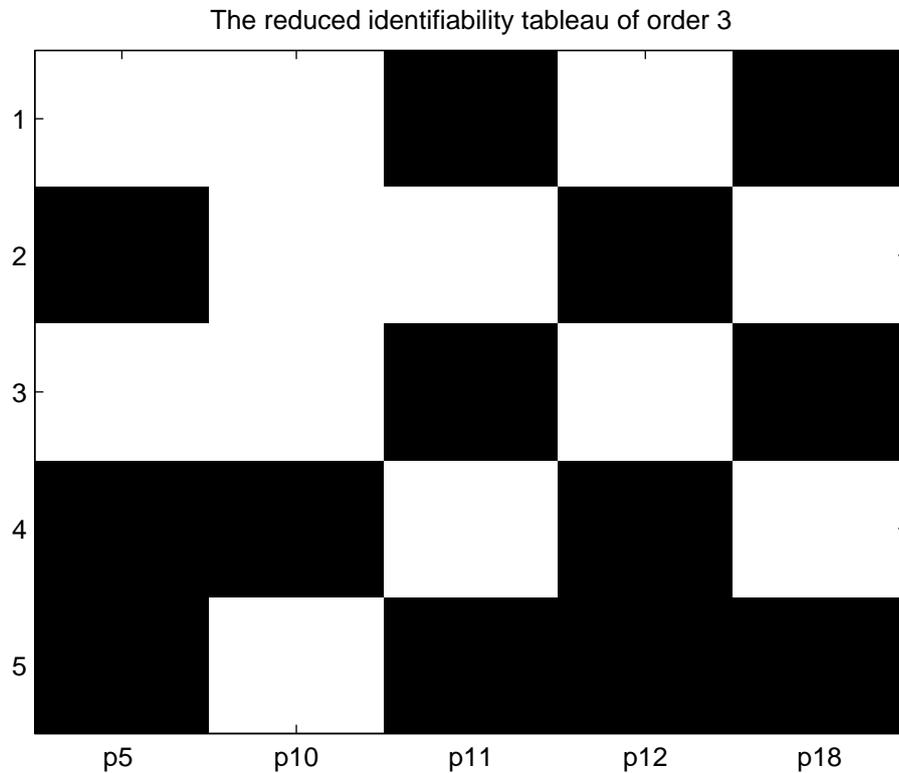


Figure 3.9: **Third reduced identifiability tableau for *Arabidopsis thaliana* model (11 parameters)**

The third reduced tableau is given in Figure 3.9.

The above procedure repeats for the newly generated reduced tableau. Now the parameters p11 and p18 can be computed and their solution is also unique.

After this step the reduced tableau is represented in Figure 3.10

The last reduction is done for the parameters p5 and p12, also having unique solution, as well as the remaining parameters p10. So all the parameters have unique solution. As we have done the identifiability for a subset of parameters, only local identifiability is proved. the computational cost is smaller in this case, 999.4844s.

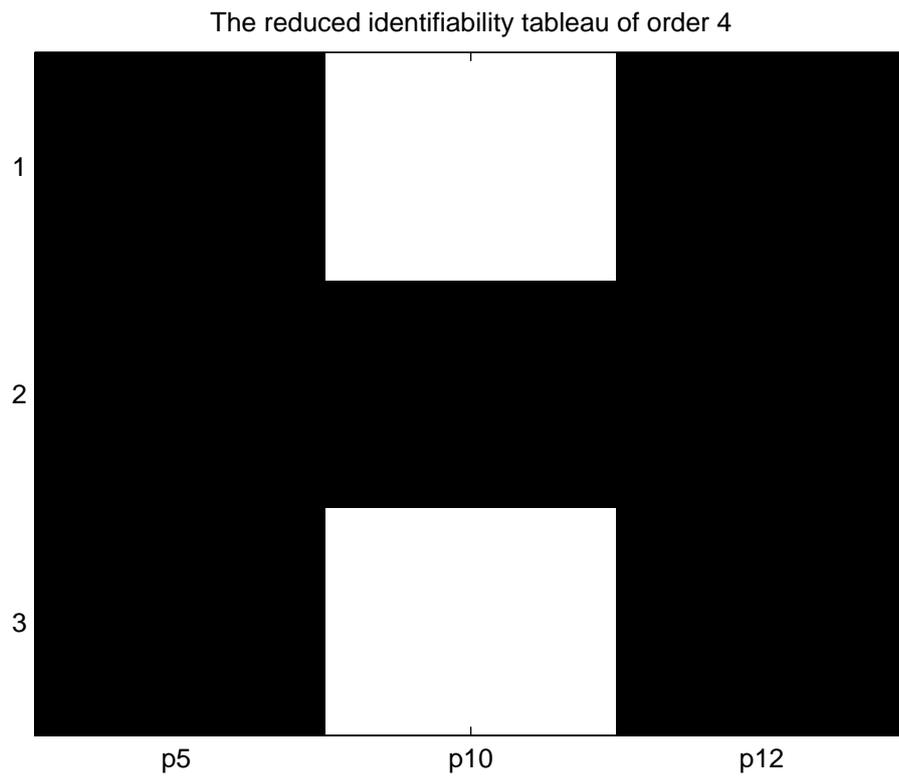


Figure 3.10: **4th reduced identifiability tableau for *Arabidopsis thaliana* model (11 parameters)**

3.3 Glycolysis inspired metabolic pathway

This model represents a glycolysis inspired pathway (the upper part of the glycolysis) with different physiological constraints on enzyme synthesis as described in Bartl et al. 2010 [6]. An specific enzyme, here denoted by u , usually catalyzes a metabolic reaction, expressed in terms of stoichiometric matrix and the metabolites, here denoted by x . The dynamical model can be written as a system of differential equations

$$\left\{ \begin{array}{l} \dot{x}_1 = -\frac{k_1 x_1}{x_1 + k_M} u_1, \\ \dot{x}_2 = \frac{k_1 x_1}{x_1 + k_M} u_1 - \frac{k_2 x_2}{x_2 + k_M} u_2, \\ \dot{x}_3 = \frac{k_2 x_2}{x_2 + k_M} u_2 - \frac{k_3 x_3}{x_3 + k_M} u_3, \\ \dot{x}_4 = \frac{k_2 x_2}{x_2 + k_M} u_2 + \frac{k_3 x_3}{x_3 + k_M} u_3 - \frac{k_4 x_4}{x_4 + k_M} u_4, \\ \dot{x}_5 = \frac{k_4 x_4}{x_4 + k_M} u_4. \\ x_1(0) = S_1, x_2(0) = S_2, x_3(0) = S_3, x_4(0) = S_4, x_5(0) = S_5, \end{array} \right. \quad (3.3)$$

and the model is considered to be fully observed, $\{y_1 = x_1, y_2 = x_2, y_3 = x_3, y_4 = x_4, y_5 = x_5\}$.

The tests were done on a 2.66GHz, 8.00GB RAM, and the computational cost is 41.9375s. The information given by the user, and presented in the model_information.txt file is given above:

```

*****
*
* GENERATING SERIES approach for Structural Identifiability Analysis
*
* Oana Chis, Julio R. Banga and Eva Balsa-Canto
* BioProcess Engineering Group, IIM-CSIC, Vigo-Spain
* contact: [chisoana,julio,ebalsa]@iim.csic.es
*
*****

STRUCTURAL IDENTIFIABILITY ANALISYS FOR: BartI Model

*****
* INPUT DATA
*****

----> Maximum number of derivatives for the analysis: 2
----> Dynamic model:
      A1=0
      A2=0
      A3=0
      A4=0
      A5=0

----> Control variables:
      G1=[ -k1*x1/(x1+kM), k1*x1/(x1+kM),      0,      0,      0]
      G2=[      0, -k2*x2/(x2+kM), k2*x2/(x2+kM), k2*x2/(x2+kM),      0]
      G3=[      0,      0, -k3*x3/(x3+kM), k3*x3/(x3+kM),      0]
      G4=[      0,      0,      0, -k4*x4/(x4+kM), k4*x4/(x4+kM)]

----> Observables:
      H1=x1
      H2=x2
      H3=x3
      H4=x4
      H5=x5

----> Initial conditions:
      [ S0, S1, S2, S3, S4]

----> Parameters to be considered in the analysis:
      [k1  k2  k3  k4  kM  ]

```

The next step is to compute the identifiability tableaus. In Figure 3.11 it is represented the complete identifiability tableau.

```

*****
-> COMPUTES LIE DERIVATIVES
*****

COMPUTING LIE DERIVATIVES OF ORDER 1
.....
->The rank of the Jacobian generated by 1 derivative is 4.
.....

COMPUTING LIE DERIVATIVES OF ORDER 2
.....
->The rank of the Jacobian generated by 2 derivatives is 5.

*****
-> COMPUTES IDENTIFIABILITY TABLEAU
*****

---->The rank of the full Jacobian matrix is 5

---->THE RANK OF THE FULL JACOBIAN IS COMPLETE, THUS AT LEAST LOCAL °
IDENTIFIABILITY IS GUARANTEED.

*****
-> COMPUTES REDUCED IDENTIFIABILITY TABLEAUS
*****

*****
-> THE RELATIONS NEEDED FOR COMPUTING THE PARAMETERS
*****
-k1*S0/(S0+kM)-c1
-k2*S1/(S1+kM)-c2
-k3*S2/(S2+kM)-c3
-k4*S3/(S3+kM)-c4
-k1*S0/(S0+kM)*(-k1/(S0+kM)+k1*S0/(S0+kM)^2)-c5

*****
-> DETECTS (DIRECT) STRUCTURALLY GLOBALLY IDENTIFIABLE PARAMETERS AND
REORGANIZES TABLEAU
*****

-> STRUCTURALLY GLOBALLY IDENTIFIABLE PARAMETERS DETERMINED DIRECTLY
(parameters corresponding to one non-zero element in the reduced
identifiability tableau)

None

```

As there are no direct structurally globally identifiable parameters, the first and second reduced identifiability tableaus coincide (Figure 3.12 (a)). But the third order identifiability tableau can be computed, using the following relations

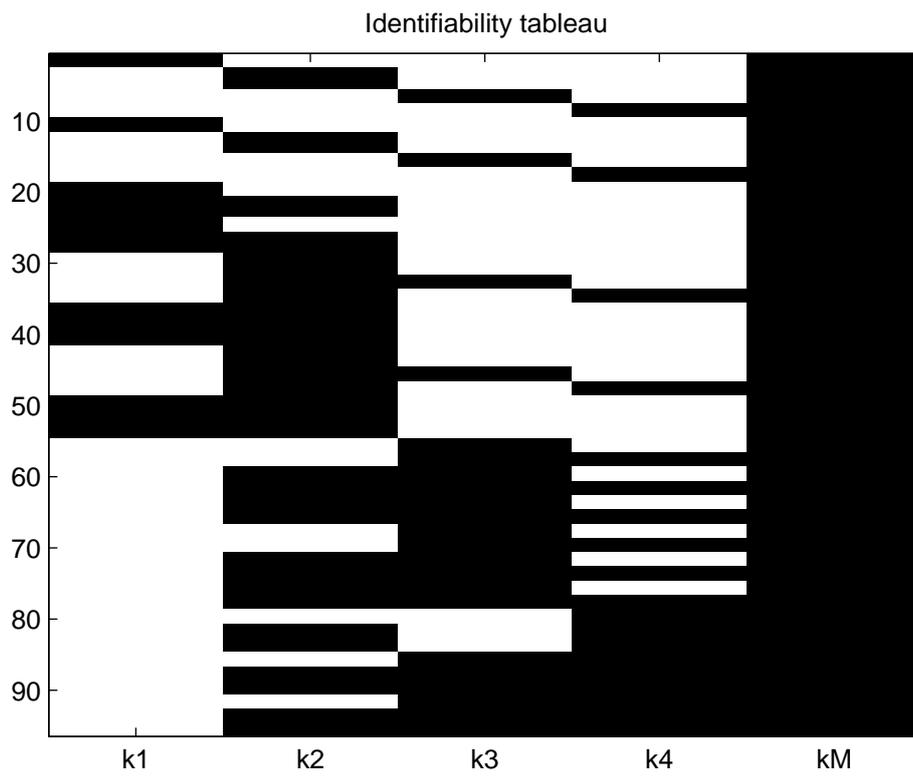


Figure 3.11: Identifiability tableau for (3.3)

and parameters:

```

*****
->THE REMAINING PARAMETERS (APART FROM IDENTIFIABLE OR NON-
IDENTIFIABLE), AND THE CORRESPONDING RELATIONS
*****

----> Parameters:
[ k1, k2, k3, k4, kM]

----> Relations:
-k1*S0/(S0+kM)-c1
-k2*S1/(S1+kM)-c2
-k3*S2/(S2+kM)-c3
-k4*S3/(S3+kM)-c4
-k1*S0/(S0+kM)*(-k1/(S0+kM)+k1*S0/(S0+kM)^2)-c5

*****
-> COMPUTES HIGHER ORDER REDUCED IDENTIFIABILITY TABLEAU(S)

  (displays the group of 2/more depending parameters,
   the associated algebraic relations,
   the corresponding solution (solutions))

*****

----> The group of parameters to be considered in the calculus and the correspon-
      ding relations:

-> Parameters:
    [k1    kM    ]

-> Relations:
-k1*S0/(S0+kM)-c1
-k1*S0/(S0+kM)*(-k1/(S0+kM)+k1*S0/(S0+kM)^2)-c5

Solution =

    k1: [1x1 sym]
    kM: [1x1 sym]

-> The parameter k1 has the solution/solutions:
-c1^3/(c1^2-c5*S0)

-> The parameter kM has the solution/solutions:
c5*S0^2/(c1^2-c5*S0)

```

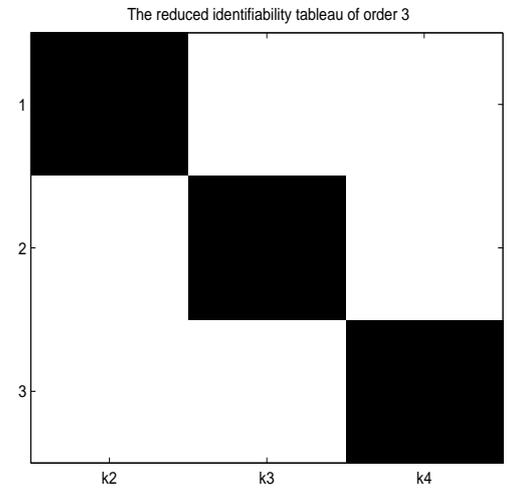
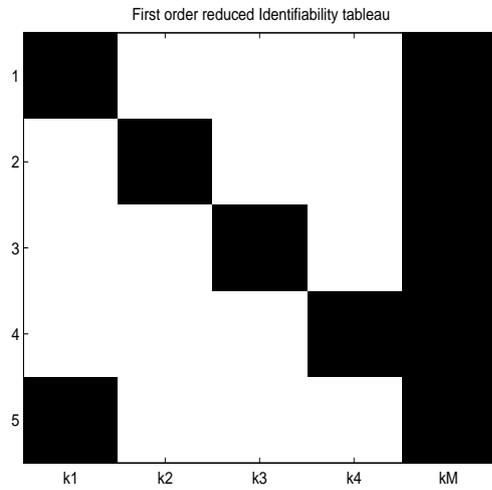


Figure 3.12:(a) Reduced identifiability tableau for (3.3)

(b) 3rd order identifiability tableau for (3.3)

As the third order identifiability tableau can't be reduced, the solution for the remaining parameters is given, and also the conclusion regarding the structural identifiability test:

----> The remaining group of parameters, relations and the corresponding solutions:

-> Parameters:

[k2 k3 k4]

-> Relations:

-k2*S1/(S1+kM)-c2
-k3*S2/(S2+kM)-c3
-k4*S3/(S3+kM)-c4

Solution =

k2: [1x1 sym]
k3: [1x1 sym]
k4: [1x1 sym]

----> The parameter k2 has the solution/solutions:

-c2*(S1+kM)/S1

----> The parameter k3 has the solution/solutions:

-c3*(S2+kM)/S2

----> The parameter k4 has the solution/solutions:

-c4*(S3+kM)/S3

----> THE MODEL IS STRUCTURALLY GLOBALLY IDENTIFIABLE

The structurally globally identifiable parameters are:

[k1 k2 k3 k4 kM]

4

Some comments on using SBML models with GenSSI

SBML models can be imported into MATLAB using for example the SBML-Toolbox (<http://sbml.org/Software/SBMLToolbox>). However, these imported models can not be used right away with GenSSI. The reason is that the generating series approach requires the definition of the f and g functions separately, being g the function(s) that multiplies the stimuli (control variables).

Unfortunately the definition of stimuli creates difficulties due to insufficient support by the SBML framework. This makes the automatic extraction of f and g from SBML models very complicated, if not impossible. It should be noted that this limitation also affects other structural identifiability analysis tools.

However, the situation is not totally hopeless: although it can not be done in a fully automatic way, it is possible to use an existing SBML model as an stepping stone to define the f and g functions required by GenSSI.

We now describe how a MATLAB model translated from SBML (by means for example of the SBMLtoolbox) can be easily handled in GenSSI. Let us consider an example, model BIOMD0000000005 - Tyson1991-CellCycle-6var model [18] from the BioModels Database (<http://www.ebi.ac.uk/biomodels-main/>). The first step would be to import the SBML definition into MATLAB using the SBML-Toolbox. Alternatively, the BioModels Database also offers different formats for the models, among which an Octave m-file) can be used right away in MATLAB without the need of importing the SBML.

The changes that are now needed so the m-file can be used with GenSSI can be summarized as follows:

- declare the symbolic variables: the states of the model, and the parameters;
- give the number of derivatives (Nder), of state variables (Neq), of parameters (Npar), of controls (Noc) and of observables (Nobs);
- use the assignment rules YT and CT for $x(8)$ and $x(9)$, otherwise GenSSI can not generate the Jacobian with respect to $x(6)+x(7)+x(4)+x(5)$ or $x(2)+x(3)+x(4)+x(5)$, and replace $\dot{x}(i)$ with A_i , $i=1,\dots,9$;
- define the controls, the observables and rewrite the initial conditions vector;
- call the GenSSI_generating_series function.


```

% # Compartment: id = cell, name = cell, constant
compartment_cell=1.0;
% # assignmentRule: variable = YT
x(8)=x(6)+x(7)+x(4)+x(5);
% # assignmentRule: variable = CT
x(9)=x(2)+x(3)+x(4)+x(5);
% # Reaction: id = Reaction1, name = cyclin_cdc2k dissociation
% # Local Parameter: id = k6, name = k6, reaction_Reaction1_k6=1.0;

reaction_Reaction1=compartment_cell*reaction_Reaction1_k6*x(4);
% # Reaction: id = Reaction2, name = cdc2k phosphorylation
% # Local Parameter: id = k8notP, name = k8notP
% reaction_Reaction2_k8notP=1000000.0;

reaction_Reaction2=compartment_cell*x(2)*reaction_Reaction2_k8notP;
% # Reaction: id = Reaction3, name = cdc2k dephosphorylation
% # Local Parameter: id = k9, name = k9, reaction_Reaction3_k9=1000.0;

reaction_Reaction3=compartment_cell*x(3)*reaction_Reaction3_k9;
% # Reaction: id = Reaction4, name = cyclin cdc2k-p association
% # Local Parameter: id = k3, name = k3, reaction_Reaction4_k3=200.0;

reaction_Reaction4=compartment_cell*x(3)*reaction_Reaction4_k3*x(6);
% # Reaction: id = Reaction5, name = deactivation of cdc2 kinase
% # Local Parameter: id = k5notP, name = k5notP,
% reaction_Reaction5_k5notP=0.0;

reaction_Reaction5=compartment_cell*reaction_Reaction5_k5notP*x(4);
% # Reaction: id = Reaction6, name = cyclin biosynthesis
% # Local Parameter: id = k1aa, name = k1aa, reaction_Reaction6_k1aa=0.015;

reaction_Reaction6=compartment_cell*reaction_Reaction6_k1aa;
% # Reaction: id = Reaction7, name = default degradation of cyclin
% # Local Parameter: id = k2, name = k2, reaction_Reaction7_k2=0.0;

reaction_Reaction7=compartment_cell*reaction_Reaction7_k2*x(6);
% # Reaction: id = Reaction8, name = cdc2 kinase triggered degradation of cyclin
% # Local Parameter: id = k7, name = k7, reaction_Reaction8_k7=0.6;

reaction_Reaction8=compartment_cell*reaction_Reaction8_k7*x(7);
% # Reaction: id = Reaction9, name = activation of cdc2 kinase
% # Local Parameter: id = k4, name = k4
% reaction_Reaction9_k4=180.0;
% # Local Parameter: id = k4prime, name = k4prime
% reaction_Reaction9_k4prime=0.018;

reaction_Reaction9=compartment_cell*x(5)*(reaction_Reaction9_k4prime+
reaction_Reaction9_k4*(x(4)/YT)^(2));

```

> use the assignment rules:
 $YT=x(6)+x(7)+x(4)+x(5)$
 $CT=x(2)+x(3)+x(4)+x(5)$

```
% xdot=zeros(9,1);
% # Species: id = EmptySet, name = EmptySet

% #WARNING speciesID: EmptySet, constant= false , boundaryCondition = true but
is not involved in assignmentRule, rateRule or events !
```

> write the equations of the model

```
A1=xdot(1) = 0.0;      replace xdot(i) with Ai, i=1,...,9
```

```
% # Species: id = C2, name = cdc2k, affected by kineticLaw
```

```
A2=xdot(2) = (1/(compartment_cell))*(( 1.0 * reaction_Reaction1) +
(-1.0 * reaction_Reaction2) + ( 1.0 * reaction_Reaction3));
```

```
% # Species: id = CP, name = cdc2k-P, affected by kineticLaw
```

```
A3=xdot(3) = (1/(compartment_cell))*(( 1.0 * reaction_Reaction2) +
(-1.0 * reaction_Reaction3) + (-1.0 * reaction_Reaction4));
```

```
% # Species: id = M, name = p-cyclin_cdc2, affected by kineticLaw
```

```
A4=xdot(4) = (1/(compartment_cell))*((-1.0 * reaction_Reaction1) +
(-1.0 * reaction_Reaction5) + ( 1.0 * reaction_Reaction9));
```

```
% # Species: id = pM, name = p-cyclin_cdc2-p, affected by kineticLaw
```

```
A5=xdot(5) = (1/(compartment_cell))*(( 1.0 * reaction_Reaction4) +
( 1.0 * reaction_Reaction5) + (-1.0 * reaction_Reaction9));
```

```
% # Species: id = Y, name = cyclin, affected by kineticLaw
```

```
A6=xdot(6) = (1/(compartment_cell))*((-1.0 * reaction_Reaction4) +
( 1.0 * reaction_Reaction6) + (-1.0 * reaction_Reaction7));
```

```
% # Species: id = YP, name = p-cyclin, affected by kineticLaw
```

```
A7=xdot(7) = (1/(compartment_cell))*(( 1.0 * reaction_Reaction1) +
(-1.0 * reaction_Reaction8));
```

```
% # Species: id = YT, name = total_cyclin, involved in a rule
```

```
A8=xdot(8) = YT;
```

```
% # Species: id = CT, name = total_cdc2, involved in a rule
```

```
A9=xdot(9) = CT;
```

```
F=[A1 A2 A3 A4 A5 A6 A7 A8 A9];
```

```
% define the controls
```

```
G1=0;G2=0; G3=0;G4=0;G5=0;G6=0;G7=0;G8=0;G9=0;  
G=[G1 G2 G3 G4 G5 G6 G7 G8 G9];
```

```
% define the observables
```

```
h1=x1; h2=x4; h3=x3; h4=x4; h5=x5; h6=x6; h7=x7; h8=x8; h9=x9;  
H=[h1 h2 h3 h4 h5 h6 h7 h8 h9];
```

```
#Initial conditions vector
```

```
x0=zeros(9,1);  
x0(1) = 0.0;  
x0(2) = 0.0;  
x0(3) = 0.75;  
x0(4) = 0.0;  
x0(5) = 0.25;  
x0(6) = 0.0;  
x0(7) = 0.0;  
x0(8) = 0;  
x0(9) = 0;
```

> redefine the vector of initial conditions as IC

```
IC=[0.0 0.0 0.75 0.0 0.25 0.0 0.0 0 0];
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  
% PARAMETERS CONSIDERED FOR IDENTIFIABILITY %  
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
Par=[reaction_Reaction1_k6 reaction_Reaction2_k8notP reaction_Reaction3_k9  
      reaction_Reaction4_k3 reaction_Reaction5_k5notP reaction_Reaction6_k1aa  
      reaction_Reaction7_k2 reaction_Reaction8_k7 reaction_Reaction9_k4prime  
      reaction_Reaction9_k4];
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  
% GENERATING SERIES FUNCTION %  
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
GenSSI_generating_series(F,G,Noc,Neq,Nder,Nobs,H,x,IC,Par,result_folder);
```

Bibliography

- [1] Achard, P., Schutter, E.D., *Complex parameter landscape for a complex neuron model*, PLOS Comput. Biol., 2(7), 0794-0803, 2006.
- [2] Balsa-Canto, E., Banga, J.R., *AMIGO, a toolbox for advanced model identification in systems biology using global optimization*, Bioinformatics, 10.1093/bioinformatics/btr370.
- [3] Balsa-Canto, E., Alonso, A., Banga, J., *Computational procedures for optimal experimental design in biological systems*, IET Syst. Biol., 2(4), 163-172, 2008.
- [4] Balsa-Canto, E., Alonso, A., Banga, J., *An iterative identification procedure for dynamic modeling of biochemical networks*, BMC Syst. Biol., 4-11, 2010.
- [5] Balsa-Canto, E., Peifer, M., Banga, J.R., Timmer, J., Fleck, C., *Hybrid optimization method with general switching strategy for parameter estimation*, BMC Syst. Biol. (2:26), 2008.
- [6] Bartl, M., Kotzing, M., Kaleta, C., Schuster, S., Li, P. *Just-in-time activation of a glycolysis inspired metabolic network - solution with a dynamic optimization approach*, Proc. 55nd International Scientific Colloquium, Ilmenau, Germany, 2010.
- [7] Brown, K., Hill, C., Calero, G., Myers, C., Lee, K., Sethna, J., Cerione, R., *The statistical mechanics of complex signaling networks: nerve growth factor signaling*, Phys. Biol., 1, 184-195, 2004.
- [8] Dochian, D., Vanrolleghem, P., *Dynamical modelling and estimation in wastewater treatment processes*, London : IWA Publishing, 2001.

- [9] Dunlap, J., *Molecular bases of circadian clocks*, Cell 96, 271-290, 1999.
- [10] Fliess, M., *Fonctionnelles causales non linéaires et indéterminées non commutatives*, Bull. Soc. Math. France 1.09, 3–40, 1981.
- [11] Goodwin, B.C., *Oscillatory behavior in enzymatic control processes*, Adv. Enzyme Regul. (3), 425–428, 1965.
- [12] Hengl, S., Kreutz, D., Timmer, J., Maiwald, T., *Data-based identifiability analysis of non-linear dynamical models*, Bioinformatics, 23(19), 2612-2618, 2007.
- [13] Lipniacki, T., Paszek, P., Brasier, A., Luxon, B., Kimmel, M., *Mathematical model of nfkb regulatory module*, J. Theor. Biol., 228, 195-215, 2004.
- [14] Locke, J.C.W., Millar, A.J., Turner, M.S., *Modeling genetic networks with noisy and varied experimental data: the circadian clock in Arabidopsis thaliana*, J. Theor. Biol. (234), 383–393, 2005.
- [15] Margaria, M., Riccomagno, E., Chappel, M., Wynn, H., *Differential algebra methods for the study of the structural identifiability of rational function statespace models in the biosciences*, Mathematical Biosciences, 174, 1-26, 2001.
- [16] Piazza, M., Feng, X., Rabinowitz, J., Rabitz, H., *Diverse metabolic model parameters generate similar methionine cycle dynamics*. J. Theor. Biol., 251(4), 628-639, 2008.
- [17] Srinath, S., Gunawan, R., *Parameter identifiability of power-law biochemical system models*, J. Biotechnol., 149(3), 132-140, 2010.
- [18] Tyson, J.J., *Modeling the cell division cycle: cdc2 and cyclin interactions*, Proc Natl Acad Sci USA, Aug; 88(16):7328–32, 1991.
- [19] Vajda, S., *Structural identifiability of linear, bilinear, polynomial and rational systems*, Proceedings of the 9th IFAC World Congress, Budapest, Hungary, p. 107, 1984.
- [20] Vajda, S., *Identifiability of polynomial systems: structural and numerical aspects*, Identifiability of parametric models, Pergamon, Oxford, p. 42, 1987.

- [21] Walter, E., Lecourtier, Y., *Unidentifiable compartmental models: what to do?*, *Mathematical Biosciences* 56, 1–25, 1981.
- [22] Walter, E., Prozato, L., *Identification of parametric models from experimental data*, Springer, Berlin, 1997.
- [23] Walter, E., Pronzato, L., *On the identifiability and distinguishability of nonlinear parametric models*, *Mathematics and Computers in Simulation*, 42(2-3), 125–134, 1996.