# Structure-based Determination of Equilibrium Points of Genetic Regulatory Networks Described by Differential Equation Models

Graziano Chesi

*Abstract*— **A fundamental problem in systems biology consists of determining the equilibrium points of genetic regulatory networks, since the knowledge of these points is often required in order to investigate important properties such as stability. Unfortunately, this problem amounts to computing the solutions of a system of nonlinear equations, and it is well known that this is a difficult problem as no existing method guarantees to find all solutions. This paper addresses this problem for genetic regulatory networks described by differential equation models. By exploiting the structure of these networks, it is shown that one can derive an iterative strategy for progressively singling out the equilibrium points, which does not rely on the solution of any nonconvex optimization problem, and which guarantees to find all equilibrium points. Some numerical examples with small and large sizes (up to 24 state variables) illustrate the benefits of the proposed strategy with respect to existing methods, which often are unable to provide the sought equilibrium points.**

## I. Introduction

Genetic regulatory networks explain the interactions between genes and proteins to form complex systems that perform complicated biological functions. Basically, there are two types of genetic regulatory network models, i.e., the Boolean model (or discrete model) and the differential equation model (or continuous model). In Boolean models, the activity of each gene is expressed in one of two states, ON or OFF. In the differential equation models, the variables are continuous values that describe the concentrations of gene products, such as mRNAs and proteins. See for example [1]–[7] and references therein.

This paper focuses on genetic regulatory networks described by differential equation models. In these models the dynamics of each concentration is expressed through a function of all concentrations of the system. This function typically consists of two parts: a linear part which defines the natural decay rate of the concentration itself, and a nonlinear part which defines the influence on this concentration by all the other ones. The nonlinear part contains saturation functions, such as the Hill functions, which are combined for example via sums or products.

A fundamental problem in the study of genetic regulatory networks consists of determining the equilibrium points, i.e. the amounts of concentrations for which the regulation process results complete. This is a necessary step for several important investigations, concerning for instance stability, robustness, and disturbance rejection, see for instance [8]–[14]. Unfortunately, computing the equilibrium points of genetic regulatory networks is a difficult problem because these systems contain saturation functions, and hence the calculation of the equilibrium points amounts to solving a system of nonlinear equations. Indeed, no existing method guarantees to find all the solutions of such a system, except in the case of polynomial equations, which however can be addressed only for small degrees and small number of variables. The reader is referred to the works [15], [16] for general techniques, and to [17]–[20] which describe LMI-based methods for solving systems of polynomial equations. See also Section IV for some numerical examples.

In this paper we address the problem of computing equilibrium points of genetic regulatory networks described through differential equation models. We consider a general model which includes various special cases. The contribution consists of a recursive algorithm which holds the following properties. First, at each recursion the algorithm provides a region containing all equilibrium points, i.e. no equilibrium is lost. Second, this region progressively shrinks, i.e. the conservatism does not increase. Third, this region asymptotically converges to the set of equilibrium points, i.e. all equilibrium points are found. The proposed algorithm is illustrated and validated through some numerical examples, with synthetic and real genetic regulatory networks, where it is shown that existing methods for solving systems of nonlinear equations may be unable to compute the sought equilibrium points.

The paper is organized as follows. Section II introduces some preliminaries on genetic regulatory networks. Section III describes the proposed strategy. Section IV presents some illustrative examples. Finally, Section V reports some concluding remarks.

## II. Preliminaries

First of all, let us introduce the notation used throughout the paper:

- $\mathbb{R}$: space of real numbers;
- $\mathbb{R}_+$: $\{x \in \mathbb{R} : \ x \geq 0\}$;
- $0_n$: null vector of size $n \times 1$;
- $I_n$: identity matrix of size $n \times n$;
- $e_i$: $i$-th column of the identity matrix (with size specified by the context);
- $X'$: transpose of vector/matrix $X$;
- $\|X\|$: 2-norm of vector/matrix $X$;
- TF: transcription factor.

G. Chesi is with the Department of Electrical and Electronic Engineering, University of Hong Kong, Pokfulam Road, Hong Kong, Tel.: +852-22194362, Fax: +852-25598738, Web: http://www.eee.hku.hk/~chesi

978-89-956056-9-1/09/©2009 ACA

The genetic regulatory networks considered in this paper are described by the differential equation model

$$\begin{cases} \dot{m}_i(t) &= -a_i m_i(t) + b_i(p_1(t),\dots,p_n(t)) \\ \dot{p}_i(t) &= -c_i p_i(t) + d_i m_i(t) \\ i &= 1,\dots,n \end{cases} \quad (1)$$

where $m_i(t), p_i(t) \in \mathbb{R}_+$ are the concentrations of mRNA and protein of the $i$-th gene, $a_i, c_i \in \mathbb{R}_+$ are the degradation rates, and $d_i \in \mathbb{R}_+$ expresses the effect of $m_i(t)$ on $p_i(t)$. The function $b_i(p_1(t),\dots,p_n(t))$ is the regulatory function of the $i$-th gene, which is generally a nonlinear function of the variables $p_1(t),\dots,p_n(t)$ such that:

1) $b_i(p_1(t),\dots,p_n(t)) \in \mathbb{R}_+$ for all $p_1(t),\dots,p_n(t) \in \mathbb{R}_+$;
2) $b_i(p_1(t),\dots,p_n(t))$ is bounded for all $p_1(t),\dots,p_n(t)$ such that $\|p_1(t)\|,\dots,\|p_n(t)\|$ are bounded;
3) $b_i(p_1(t),\dots,p_n(t))$ is either monotonically increasing or monotonically decreasing with $p_j(t)$ for all $j = 1,\dots,n$.

For instance, the function $b_i(p_1(t),\dots,p_n(t))$ can be expressed as the sum of functions of a single variable [10], [11], [14], i.e.

$$b_i(p_1(t),\dots,p_n(t)) = \sum_{j=1}^{n} \alpha_{i,j} b_{i,j}(p_j(t)) \quad (2)$$

where $\alpha_{i,j} \in \mathbb{R}_+$ is the contribution of TF $j$ to the transcriptional rate for gene $i$, and $b_{i,j} : \mathbb{R}_+ \to \mathbb{R}_+$ is a monotonic function. Alternatively, the function $b_i(p_1(t),\dots,p_n(t))$ can be expressed as the product of the functions $b_{i,j}(p_j(t))$, i.e.

$$b_i(p_1(t),\dots,p_n(t)) = \alpha_i \prod_{j=1}^{n} b_{i,j}(p_j(t)) \quad (3)$$

where $\alpha_i \in \mathbb{R}_+$ represents the transcriptional rate for gene $i$.

Each function $b_{i,j}(p_j(t))$ in (2) and (3) is typically expressed as

$$b_{i,j}(p_j(t)) = \begin{cases} f(p_j(t)) & \text{if TF } j \text{ is an activator of gene } i \\ 1 - f(p_j(t)) & \text{if TF } j \text{ is a repressor of gene } i \\ \gamma & \text{otherwise} \end{cases} \quad (4)$$

where $\gamma \in \mathbb{R}$ is a constant depending on the model, in particular $\gamma = 0$ for the case considered in (2), and $\gamma = 1$ for the case considered in (3). The function $f(p_j(t))$ is a saturation function, i.e. a function satisfying the following properties:

$$\begin{cases} f : \mathbb{R}_+ \to [0,1] \\ f(p_j(t)) \text{ increases as } p_j(t) \text{ increases} \\ f(0) = 0 \\ \lim_{x\to\infty} f(x) = 1 \end{cases} \quad (5)$$

Hence, a saturation function is an increasing function between 0 and 1 defined for positive value of the variable. For instance, in the case of regulatory functions with Hill form, the function $f(p_j(t))$ is given by

$$f(p_j(t)) = \frac{p_j(t)^H}{\beta^H + p_j(t)^H} \quad (6)$$

where $\beta \in \mathbb{R}_+$ and $H$ is an integer known as Hill coefficient.

In order to describe the results of this paper in a more compact form, we introduce a matrix version of the model (1) according to

$$\begin{cases} \dot{m}(t) &= Am(t) + b(p(t)) \\ \dot{p}(t) &= Cp(t) + Dm(t) \end{cases} \quad (7)$$

where $m(t) = (m_1(t),\dots,m_n(t))' \in \mathbb{R}^n$ and $p(t) = (p_1(t),\dots,p_n(t))' \in \mathbb{R}^n$ are the vectors containing the concentrations of mRNA and protein, $A = \mathrm{diag}(-a_1, \dots, -a_n) \in \mathbb{R}^{n\times n}$ and $C = \mathrm{diag}(-c_1,\dots,-c_n) \in \mathbb{R}^{n\times n}$ are diagonal matrices containing the decay rates, and $D = \mathrm{diag}(d_1,\dots,d_n) \in \mathbb{R}^{n\times n}$ is a diagonal matrix expressing the effect of $m(t)$ on $p(t)$. The function $b(p(t)) = (b_1(p(t)),\dots,b_n(p(t)))'$ is a nonlinear function representing the regulation of the process assumed to satisfy the following conditions:

1) $b(p(t)) \in \mathbb{R}^n_+$ for all $p(t) \in \mathbb{R}^n_+$;
2) $\|b(p(t))\|$ is bounded for any $p(t)$ such that $\|p(t)\|$ is bounded;
3) $b_i(p(t))$ is either monotonically increasing or monotonically decreasing with $p_j(t)$ for all $j = 1,\dots,n$.

The problem addressed in this paper consists of determining the equilibrium points of (7), i.e. the solutions of the system of nonlinear equations

$$\begin{cases} Am + b(p) = 0_n \\ Cp + Dm = 0_n \\ m, p \in \mathbb{R}^n_+ \end{cases} \quad (8)$$

## III. Equilibria computation

In this section we describe the proposed algorithm. Specifically, in Theorems 1 and 2 we introduce two preliminary functions and we describe their properties. Then, in Theorem 3 we describe the main algorithm to be used to compute the sought equilibrium points.

Let us start by observing that the $m$-component of any solution of (8) is related to its $p$-component by the relationship $Cp + Dm = 0_n$ where $C, D$ are nonsingular diagonal matrices with $C$ negative definite. This means that (8) can be equivalently rewritten as

$$\begin{cases} -AD^{-1}Cp + b(p) = 0_n \\ m = -D^{-1}Cp \\ p \in \mathbb{R}^n_+ \end{cases} \quad (9)$$

Therefore, in the sequel we will focus on the computation of the set of vectors $p$ fulfilling (9), which we indicate as follows:

$$\mathcal{E} = \left\{ p \in \mathbb{R}^n_+ : \ -AD^{-1}Cp + b(p) = 0_n \right\} \quad (10)$$

*Theorem 1:* Let $\mathcal{H}$ be the rectangle defined by

$$\mathcal{H} = \left\{p \in \mathbb{R}^n_+ : \ p_i \in [p_{i,-}, p_{i,+}] \ \ \forall i = 1, \ldots, n\right\} \quad (11)$$

for some $p_{1,-}, p_{1,+}, \ldots, p_{n,-}, p_{n,+} \in \mathbb{R}_+$, and let us define the map $\mathcal{A}(\mathcal{H})$ as

$$\mathcal{A}(\mathcal{H}) = \left\{p \in \mathbb{R}^n_+ : \ p_i \in [q_{i,-}, q_{i,+}] \ \ \forall i = 1, \ldots, n\right\} \quad (12)$$

where $q_{1,-}, q_{1,+}, \ldots, q_{n,-}, q_{n,+} \in \mathbb{R}_+$ are computed according to

$$q_{i,-} = \max\left\{p_{i,-} \ , \ \min_{z \in \mathcal{Z}} e_i' C^{-1} D A^{-1} z\right\} \quad (13)$$

$$q_{i,+} = \min\left\{p_{i,+} \ , \ \max_{z \in \mathcal{Z}} e_i' C^{-1} D A^{-1} z\right\} \quad (14)$$

where $\mathcal{Z}$ is the set given by

$$\mathcal{Z} = \left\{b(p) : \ p_i \in \{p_{i,-}, p_{i,+}\} \ \ \forall i = 1, \ldots, n\right\}. \quad (15)$$

Then, the following properties hold:

1) *Property P1*: $\mathcal{A}(\mathcal{H}) \subseteq \mathcal{H}$;
2) *Property P2*: $p \in \mathcal{H} \cap \mathcal{E} \Rightarrow p \in \mathcal{A}(\mathcal{H})$;
3) *Property P3*: $\mathcal{H} \cap \mathcal{A}(\mathcal{H}) = \emptyset \Rightarrow \mathcal{H} \cap \mathcal{E} = \emptyset$.

Proof. First, the property P1 holds because from (13)–(14) one has

$$q_{i,-} \geq p_{i,-} \text{ and } q_{i,+} \leq p_{i,+} \ \ \forall i = 1, \ldots, n. \quad (16)$$

Second, the property P2 holds because, from the monotonicity property of $b_i(p)$ with respect to each component of $p$, it follows that

$$p \in \mathcal{H} \Rightarrow b_i(p) \in [\min_{z \in \mathcal{Z}} z_i, \max_{z \in \mathcal{Z}} z_i]. \quad (17)$$

Moreover, from the definition of $\mathcal{E}$ it follows that

$$p \in \mathcal{E} \Rightarrow e_i' C^{-1} D A^{-1} b(p) = p_i. \quad (18)$$

Consequently, since $e_i' C^{-1} D A^{-1} z$ is linear in $z$, one has that

$$p \in \mathcal{H} \cap \mathcal{E} \Rightarrow q_{i,-} \leq p_i \text{ and } q_{i,+} \geq p_i. \quad (19)$$

Lastly, the property P3 holds because, if one supposes for contradiction that

$$\mathcal{H} \cap \mathcal{A}(\mathcal{H}) = \emptyset \quad (20)$$

and $\mathcal{H}$ contains a vector $p$ of $\mathcal{E}$, then it would follow from the property P2 that $p$ belongs to $\mathcal{A}(\mathcal{H})$, hence contradicting the assumption. Therefore, the theorem holds. □

Let us observe that the map $\mathcal{A}(\cdot)$ requires trivial computations, specifically the evaluation of a linear function at some given points. From the map $\mathcal{A}(\cdot)$ we define the map $\mathcal{B}(\cdot)$ in the following theorem, which provides also some key properties of this map.

*Theorem 2:* Let $\mathcal{H}$ be a rectangle defined in (11), and let us define the map $\mathcal{B}(\mathcal{H})$ as follows:

- *Step 1*: set $\mathcal{H}^{(0)} = \mathcal{H}$ and $k = 0$, where $k$ denotes the iteration number of this map.
- *Step 2*: if $\mathcal{H}^{(k)} \cap \mathcal{A}(\mathcal{H}^{(k)}) = \emptyset$, set $\mathcal{B}(\mathcal{H}) = \emptyset$ and exit.
- *Step 3*: if $\mathcal{A}(\mathcal{H}^{(k)})$ is a point, set $\mathcal{B}(\mathcal{H}) = \mathcal{A}(\mathcal{H}^{(k)})$ and exit.
- *Step 4*: if $\mathcal{H}^{(k)} = \mathcal{A}(\mathcal{H}^{(k)})$, set $\mathcal{B}(\mathcal{H}) = \mathcal{H}^{(k)}$ and exit.
- *Step 5*: set $\mathcal{H}^{(k+1)} = \mathcal{A}(\mathcal{H}^{(k)})$, $k = k + 1$, and go to 2.

Then, the map $\mathcal{B}(\mathcal{H})$ returns either a rectangle, a point, or the empty set. Moreover, the following properties hold:

- *Property P4*: $\mathcal{B}(\mathcal{H}) \subseteq \mathcal{H}$;
- *Property P5*: $p \in \mathcal{H} \cap \mathcal{E} \Rightarrow p \in \mathcal{B}(\mathcal{H})$.

Proof. First of all, let us observe that $\mathcal{B}(\mathcal{H})$ can be either the empty set (see Step 2), a point (see Step 3), or a rectangle (see Step 4).

Then, the property P4 follows from the fact that $\mathcal{B}(\mathcal{H})$ is a sequence of applications of the map $\mathcal{A}(\cdot)$ for which the property P1 ensures that the output is included the input.

Lastly, the property P5 holds since $\mathcal{B}(\mathcal{H})$ is either a sequence of applications of the map $\mathcal{A}(\cdot)$ for which the property P2 ensures that no point lying inside the set $\mathcal{H} \cap \mathcal{E}$ can be lost, or the empty set in the case that

$$\mathcal{H}^{(k)} \cap \mathcal{A}(\mathcal{H}^{(k)}) = \emptyset \quad (21)$$

which however guarantees the absence of points of $\mathcal{E}$ in $\mathcal{H}^{(k)}$ (and hence in $\mathcal{H}$) due to the property P3. □

The map $\mathcal{B}(\cdot)$ transforms a given rectangle via a sequence of applications of the map $\mathcal{A}(\cdot)$, and returns a set which can be either a rectangle, a point, or the empty set. By exploiting the map $\mathcal{B}(\cdot)$ we derive the algorithm for the computation of the sought equilibrium points as explained in the following result.

*Theorem 3:* (Algorithm for equilibrium points computation) Let $\mathcal{H}$ be a rectangle in (11) and let us define the map $\mathcal{C}(\mathcal{H})$ as follows:

- *Step 1*: If $\mathcal{B}(\mathcal{H})$ is either the empty set or a point, then set $\mathcal{C}(\mathcal{H}) = \mathcal{B}(\mathcal{H})$ and exit.
- *Step 2*: Divide the rectangle $\mathcal{B}(\mathcal{H})$ in $2^k$ rectangles $\mathcal{H}_1, \ldots, \mathcal{H}_{2^k}$ by taking the middle point on each side of $\mathcal{B}(\mathcal{H})$ with nonzero length.
- *Step 3*: Set $\mathcal{C}(\mathcal{H}) = \bigcup_{i=1,\ldots,2^k} \mathcal{C}(\mathcal{H}_i)$ and exit.

Then, the algorithm to be launched in $\mathcal{C}(\mathbb{R}^n_+)$, for which the following properties hold:

- *Property P6*: the positive octant $\mathbb{R}^n_+$ is progressively shrunk without losing any point of $\mathcal{E}$;
- *Property P7*: the set provided by the algorithm asymptotically converges to the set $\mathcal{E}$.

Proof. The property P6 holds because $\mathcal{B}(\cdot)$ preserves any vector in $\mathcal{E}$ according to the property P5, moreover from the property P4 one has that the set provided by the algorithm cannot increase. Then, property P7 holds because no portion of $\mathbb{R}^n_+$ is lost in the division of each rectangle $\mathcal{B}(\mathcal{H})$. □

Hence, the proposed algorithm for computing the equilibrium points of (7) is launched as $\mathcal{C}(\mathbb{R}^n_+)$, which means that the positive octant $\mathbb{R}^n_+$ is used as initial rectangle $\mathcal{H}$. In fact, $\mathbb{R}^n_+$ is clearly guaranteed to contain all solutions of

(9). The initial rectangle $\mathbb{R}^n_+$ is passed to the map $\mathcal{B}(\cdot)$. If the output of this map is either the empty set or a point, then the algorithm stops as it is guaranteed there are no equilibrium points inside the considered rectangle. Otherwise, the output is a rectangle, which is then divided in smaller ones, and then passed to the map $\mathcal{B}(\cdot)$ itself. As explained by the properties P6 and P7, the set obtained by the algorithm is guaranteed to contain all points lying inside $\mathcal{E}$ and to asymptotically converge to $\mathcal{E}$ itself.

## IV. Illustrative examples

In this section we present some numerical examples in order to illustrate the main steps and the usefulness of the proposed approach. For conciseness, in these examples we report only the $p$-component of each equilibrium point, being the $m$-component directly given by

$$m = D^{-1}Cp \tag{22}$$

according to (9). The computational time for all these examples ranges from few seconds to less than 30 seconds (this computational time is relative to an implementation of the proposed algorithm with Matlab 7 on a personal computer with Pentium IV 2.2 GHz, 2 GB RAM, Windows XP). The dependence on the time $t$ of the model (7) is omitted for ease of notation.

### A. Example 1

Let us start by considering the genetic regulatory network described by

$$\begin{cases} \dot{m}_1 &= -0.9m_1 + f(p_1) + 0.8(1 - f(p_2)) \\ \dot{m}_2 &= -0.5m_2 + 2(1 - f(p_1)) \\ \dot{p}_1 &= -p_1 + m_1 \\ \dot{p}_2 &= -2p_2 + m_2 \end{cases}$$

where $f(\cdot)$ is the saturation function given by

$$f(p_i) = \frac{2}{\pi}\arctan(p_i^2).$$

This genetic regulatory network is characterized by the fact that TF 1 is an activator of gene 1 and a regressor of gene 2, and TF 2 is a regressor of gene 1. The problem consists of determining the equilibrium points of this system, i.e. the solutions of the system of nonlinear equations (8).

First of all, let us rewrite this system as in (7). This can be done by defining the vectors

$$m = \begin{pmatrix} m_1 \\ m_2 \end{pmatrix}, \quad p = \begin{pmatrix} p_1 \\ p_2 \end{pmatrix},$$

the matrices

$$\begin{aligned} A &= \mathrm{diag}(-0.9, -0.5) \\ C &= \mathrm{diag}(-1, -2) \\ D &= \mathrm{diag}(1, 1) \end{aligned}$$

and the regulation function

$$b(p) = \begin{pmatrix} f(p_1) + 0.8(1 - f(p_2)) \\ 2(1 - f(p_1)) \end{pmatrix}.$$

Then, let us use algorithm proposed in Theorem 3. At the first recursion we obtain that the positive octant $\mathbb{R}^+_2$ is shrunk to the rectangle shown in Figure 1a. At the second recursion, the rectangle previously found is divided in four equal rectangles, one of which is shown in Figure 1b, another one shrinks to the equilibrium point shown in the same figure, and the other two are discarded via the map $\mathcal{B}(\cdot)$ at its Step 2. At the fifth recursion, another equilibrium point is found as shown in Figure 2a, and only one rectangle is left. Lastly, at the ninth recursion the last equilibrium point is found as shown in Figure 2b. We hence conclude that the set $\mathcal{E}$ in (10) is given by

$$\mathcal{E} = \left\{ \begin{pmatrix} 1.626 \\ 0.460 \end{pmatrix}, \begin{pmatrix} 1.000 \\ 1.000 \end{pmatrix}, \begin{pmatrix} 0.162 \\ 1.967 \end{pmatrix} \right\}.$$

For comparison purpose, we attempt to solve the same problem by using existing methods. We use standard functions of Matlab for solving systems of nonlinear equations (such as the function "solve"), and we obtain one equilibrium point only, which is found by using iterative techniques such as Newton's method. Indeed, it is worth to remark that no existing method guarantees to find all solutions of a system of nonlinear equations as explained in Section I.

### B. Example 2

Here we consider a genetic regulatory network with 6 state variables described by

$$\begin{cases} \dot{m}_1 &= -0.17m_1 + 0.73f(p_2)(1 - f(p_3)) \\ \dot{m}_2 &= -0.8m_2 + 0.95(1 - f(p_3)) \\ \dot{m}_3 &= -0.52m_3 + 0.58(1 - f(p_1)) \\ \dot{p}_i &= -p_i + m_i \quad \forall i = 1, 2, 3 \end{cases}$$

where $f(\cdot)$ is the saturation function given by

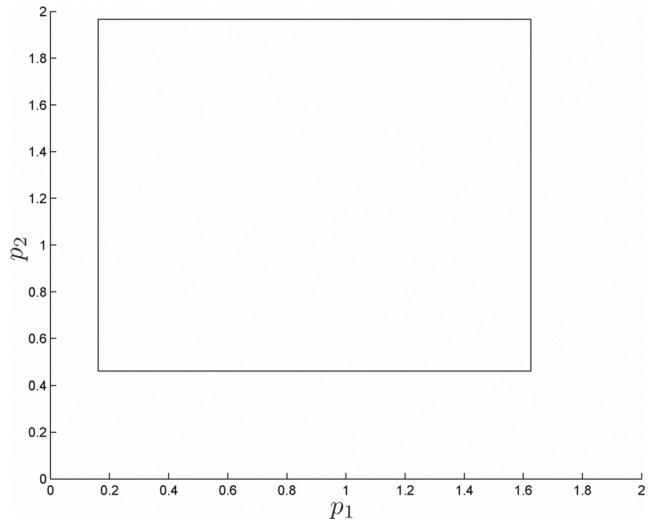
$$f(p_i) = 1 - e^{-p_i^2}.$$

This genetic regulatory network is characterized by the fact that TF 1 is a regressor of gene 3, TF 2 is an activator of gene 1, and TF 3 is a regressor of genes 1 and 2.

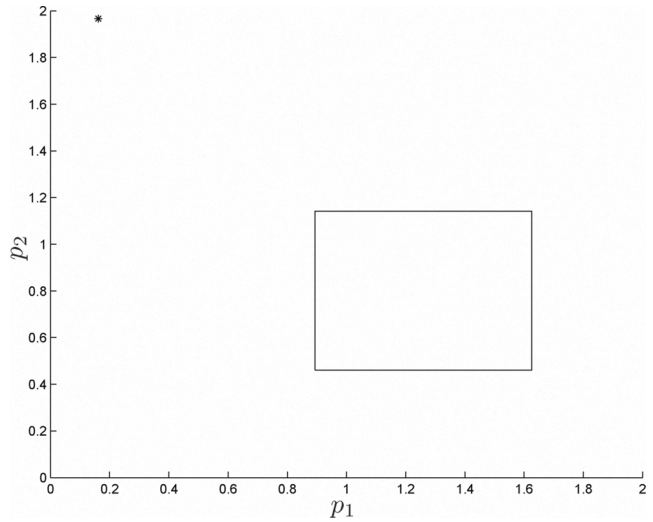
We proceed as in the previous example, and find that the set $\mathcal{E}$ in (10) is given by

$$\mathcal{E} = \left\{ \begin{pmatrix} 3.246 \\ 1.188 \\ 0.000 \end{pmatrix}, \begin{pmatrix} 0.461 \\ 0.527 \\ 0.902 \end{pmatrix}, \begin{pmatrix} 0.166 \\ 0.366 \\ 1.085 \end{pmatrix} \right\}.$$

For comparison purpose, we attempt to solve this problem by using existing methods as done in the previous example, and we find again that only one of the three solutions is obtained.

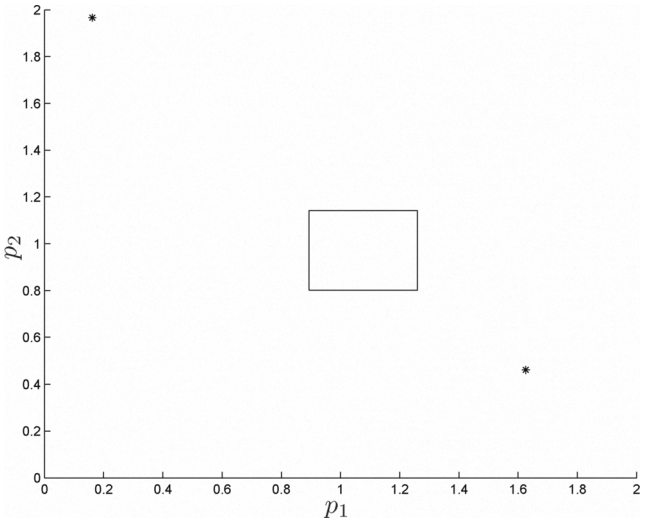



(a)

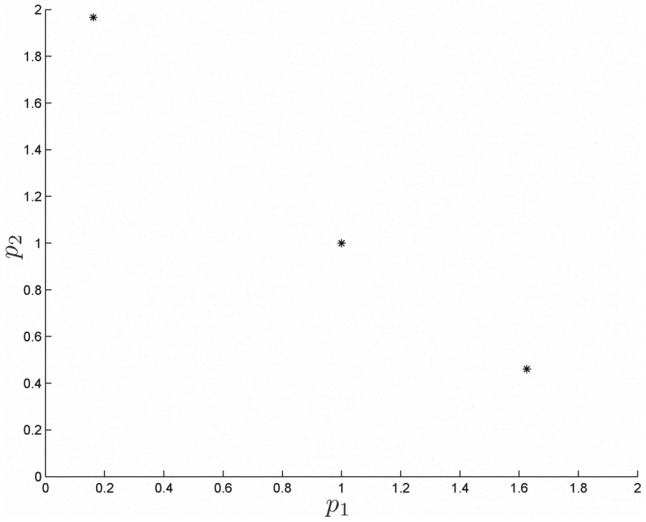



(b)

Fig. 1. Example 1: (a) first recursion, $\mathbb{R}^2_+$ is shrunk to a rectangle; (b) second recursion, an equilibrium point is found (denoted by the "$*$" mark).




(a)




(b)

Fig. 2. Example 1 (continued): (a) fifth recursion, another equilibrium point is found; (b) ninth recursion, the last equilibrium point is found.

### C. Example 3

In this example we consider a genetic regulatory network with 10 state variables, specifically

$$\begin{cases} \dot{m}_1 &= -2m_1 + 0.5f(p_5) \\ \dot{m}_2 &= -m_2 + 0.1(1 - f(p_2)) + 0.4(1 - f(p_4)) \\ \dot{m}_3 &= -0.6m_3 + 0.2f(p_1) + 1.1(1 - f(p_4)) \\ \dot{m}_4 &= -m_4 + 0.5(1 - f(p_3)) + 1.5f(p_4) \\ \dot{m}_5 &= -2m_5 + 0.3f(p_2) + 0.3(1 - f(p_5)) \\ \dot{p}_i &= -p_i + m_i \quad \forall i = 1, \ldots, 5 \end{cases}$$

and the saturation function

$$f(p_i) = \frac{2}{\pi} \arctan(p_i^2).$$

This genetic regulatory network is characterized by the fact that TF 1 is an activator of gene 3, TF 2 is an activator of gene 5 and a regressor of gene 2, TF 3 is a regressor of gene 4, TF 4 is a regressor of genes 2 and 3 and an activator of gene 4, and TF 5 is an activator of gene 1 and a regressor of gene 5.

By using the algorithm proposed in Theorem 3 we conclude that there are three equilibrium points, in particular the set $\mathcal{E}$ in (10) is given by

$$\begin{aligned} \mathcal{E} = \{&(0.004, 0.196, 0.452, 1.566, 0.152)', \\ &(0.004, 0.313, 1.003, 0.928, 0.157)', \\ &(0.005, 0.483, 1.821, 0.104, 0.169)'\}. \end{aligned}$$

Again, by using existing methods as done in the previous examples, we find that only one of the three solutions is obtained.

### D. Example 4

Lastly, let us consider an example with larger dimension, in particular the genetic regulatory network with 24 state

variables given by

$$\begin{cases} \dot{m}_1 &= -1.45m_1 + 0.5(1 - f(p_2)) + 0.8f(p_3) \\ \dot{m}_2 &= -1.70m_2 + 0.5(1 - f(p_3)) + 0.8f(p_4) \\ \dot{m}_3 &= -1.95m_3 + 0.5(1 - f(p_4)) + 0.8f(p_5) \\ &\vdots \\ \dot{m}_{10} &= -3.70m_{10} + 0.5(1 - f(p_{11})) + 0.8f(p_{12}) \\ \dot{m}_{11} &= -3.95m_{11} + 0.5(1 - f(p_{12})) + 0.8f(p_1) \\ \dot{m}_{12} &= -4.20m_{12} + 0.5(1 - f(p_1)) + 0.8f(p_2) \\ \dot{p}_i &= -p_i + m_i \quad \forall i = 1, \ldots, 12 \end{cases}$$

where the saturation function is the Hill function

$$f(p_i) = \frac{p_i^8}{1 + p_i^8}.$$

This system is characterized by a cyclic structure where gene $i$ has TF $i+1$ as regressor and TF $i+2$ as activator.

By using the algorithm proposed in Theorem 3 we find that the set $\mathcal{E}$ in (10) has only one point, specifically

$$\begin{aligned} \mathcal{E} = \{&(0.345, 0.294, 0.256, 0.227, 0.204, 0.185, \\ &0.170, 0.156, 0.150, 0.135, 0.127, 0.119)'\}. \end{aligned}$$

It is interesting to observe that also in this case we cannot reach the same conclusion by using existing methods, though the equations in the system (8) are rational in this case (and, hence, (8) can be equivalently rewritten via polynomial equations). In fact, iterative techniques such as homotopy methods do not guarantee to find all solutions, which means that one cannot conclude that a found solution is unique or not. Then, analytical techniques such as the resultants method provide the sought solutions as roots of a one-variable polynomial obtained via variables elimination, but the degree of such a polynomial can be up to the degree of the polynomial equations to the power of the number of variables in the system (8), which is given in the present case by $16^{12} \approx 2.8 \cdot 10^{14}$: in fact, the degree of the polynomial equations is 16 since the equations in (8) are sums of two rational functions of degree 8, and the number of variables is given by $n$ which is equal to 12. And, clearly, the roots of such a polynomial cannot be found.

## V. Conclusion

We have proposed a strategy for determining the equilibrium points of genetic regulatory networks described by differential equation models. This is obviously an important problem in the area of genetic regulatory networks, as the knowledge of the equilibrium points is often required in order to investigate key properties such as stability, robustness, and disturbance rejection. Unfortunately, as shown through some numerical examples with small and large scales, existing methods for solving systems of nonlinear equations often fail in the attempt of determining the equilibrium points, which is not surprising since it is well known that no systematic solution exists for this problem. Instead, the proposed strategy guarantees to find all equilibrium points, moreover the computational time is indeed small even for a genetic regulatory network with 24 state variables.

It is hence expected that the proposed strategy may be a very useful tool in the important area of systems biology.

## Acknowledgement

The author would like to thank the Editors and the Reviewers for their useful and encouraging comments.

## References

[1] F. Jacob and J. Monod, "Genetic regulatory mechanisms in the synthesis of proteins," *Journal of Molecular Biology*, vol. 3, pp. 318–356, 1961.

[2] P. D'haeseleer, X. Wen, S. Fuhrman, and R. Somogyi, "Mining the gene expression matrix: Inferring gene relationships from large scale gene expression data," in *Information Processing in Cells and Tissues*, R. C. Paton and M. Holcombe, Eds. Plenum Publishing, 1998.

[3] J. M. Bower and H. Bolouri, Eds., *Computational Modeling of Genetic and Biochemical Networks*, ser. Computational Molecular Biology. MIT Press, 2001.

[4] H. D. Jong, "Modeling and simulation of genetic regulatory systems: A literature review," *Journal of Computation Biology*, vol. 9, pp. 67–103, 2002.

[5] J. Aracena, S. B. Lamine, M. A. Mermet, O. Cohen, and J. Demongeot, "Mathematical modeling in genetic networks: Relationships between the genetic expression and both chromosomic breakage and positive circuits," *IEEE Transactions on Systems, Man, and CyberneticsPart b: Cybernetics*, vol. 33, no. 5, pp. 825–834, 2003.

[6] Y. Wang, T. Joshi, X.-S. Zhang, D. Xu, and L. Chen, "Inferring gene regulatory networks from multiple microarray datasets," *Bioinformatics*, vol. 22, no. 19, pp. 2413–2420, 2006.

[7] C. Li, L. Chen, and K. Aihara, "A systems biology perspective on signal processing in genetic network motifs," *IEEE Signal Processing Magazine*, vol. 221, no. 3, pp. 136–142, 2007.

[8] L. Chen and K. Aihara, "Stability of genetic regulatory networks with time delay," *IEEE Trans. on Circuits and Systems I*, vol. 49, no. 5, pp. 602–608, 2002.

[9] L. Chen, R. Wang, T. Kobayashi, and K. Aihara, "Dynamics of gene regulatory networks with cell division cycle," *Physical Review E*, vol. 70, p. 011909, 2004.

[10] C. Li, L. Chen, and K. Aihara, "Stability of genetic networks with sum regulatory logic: Lure system and LMI approach," *IEEE Trans. on Circuits and Systems I*, vol. 53, no. 11, pp. 2451–2458, 2006.

[11] ——, "Stochastic stability of genetic networks with disturbance attenuation," *IEEE Transactions on Circuits and Systems II*, vol. 54, no. 10, pp. 892–896, 2007.

[12] B. Munsky and M. Khammash, "The finite state projection approach for the analysis of stochastic noise in gene networks," *Joint Special Issue on Systems Biology, IEEE Transactions on Automatic Control & IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 53, no. 1, pp. 201–214, 2008.

[13] S. R. Taylor, R. Gunawan, L. R. Petzold, and F. J. D. III, "Sensitivity measures for oscillating systems: Application to mammalian circadian gene network," *Joint Special Issue on Systems Biology, IEEE Transactions on Automatic Control & IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 53, no. 1, pp. 177–188, 2008.

[14] G. Chesi and Y. S. Hung, "Stability analysis of uncertain genetic SUM regulatory networks," *Automatica*, vol. 44, no. 9, pp. 2298–2305, 2008.

[15] T. Mora, *Solving Polynomial Equation Systems II*. Cambridge University Press, 2005.

[16] J. Nocedal and S. Wright, *Numerical Optimization*, ser. Springer Series in Operations Research and Financial Engineering. Springer, 2006.

[17] G. Chesi, A. Garulli, A. Tesi, and A. Vicino, "An LMI-based approach for characterizing the solution set of polynomial systems," in *39th IEEE Conf. on Decision and Control*, Sydney, Australia, 2000, pp. 1501–1506.

[18] ——, "Characterizing the solution set of polynomial systems in terms of homogeneous forms: an LMI approach," *Int. Journal of Robust and Nonlinear Control*, vol. 13, no. 13, pp. 1239–1257, 2003.

[19] G. Chesi, "Optimal representation matrices for solving polynomial systems via LMI," *Int. Journal of Pure and Applied Mathematics*, vol. 45, no. 3, pp. 397–412, 2008.

[20] G. Chesi, A. Garulli, A. Tesi, and A. Vicino, *Homogeneous Polynomial Forms for Robustness Analysis of Uncertain Systems*. Springer, 2009.