Title	Computing equilibrium points of genetic regulatory networks
Author(s)	Chesi, G
Citation	Lecture Notes In Computer Science (Including Subseries Lecture Notes In Artificial Intelligence And Lecture Notes In Bioinformatics), 2009, v. 5750 LNBI, p. 268-282
Issued Date	2009
URL	http://hdl.handle.net/10722/129200
Rights	The original publication is available at www.springerlink.com

Computing Equilibrium Points of Genetic Regulatory Networks

Graziano Chesi *

Abstract

Computing equilibrium points of genetic regulatory networks is a problem of primary importance for numerous investigations in these systems. This paper addresses this problem for differential equation models, with the regulation function expressed in a general form which includes both SUM form and PROD form for saturation functions of any type. Specifically, a recursive algorithm is proposed, which provides at each recursion a region guaranteed to contain all equilibrium points. This region progressively shrinks, and asymptotically converges to the sought set of equilibrium points. Moreover, the proposed algorithm can also allow one to delimit and find limit cycles. Some numerical examples are reported to illustrate and validate the proposed algorithm, including examples where standard mathematical tools fail to compute the sought equilibrium points.

Keywords: Genetic regulatory network, Differential equation, Saturation, Equilibrium point, Limit cycle.

1 Introduction

Genetic regulatory networks explain the interactions between genes and proteins to form complex systems that perform complicated biological functions, see for instance [1, 2, 3, 4, 5, 6, 7, 8]. 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, and the state of a gene is expressed by a Boolean function of the states of other related genes. In the differential equation models, the variables describe the concentrations of gene products, such as mRNAs and proteins, as continuous values of the gene regulation systems. See for example [9, 10, 11, 12, 13] and references therein for a wider categorization of genetic regulatory networks models.

This paper focuses on genetic regulatory networks described through differential equation models. In these models the dynamics of each concentration is expressed by a function of all concentrations of the system. This function typically consists of two parts: a linear

*G. Chesi is with the Department of Electrical and Electronic Engineering, University of Hong Kong. Contact information: please see `http://www.eee.hku.hk/~chesi`

part which defines the natural decay rate of the concentration itself, and a nonlinear part which defines the influence by all the other concentrations. The nonlinear part can be either described via sum of saturation functions (in this case the system is said to be in SUM form) or via product of saturation functions (in this case the system is said to be in PROD form). See for instance [14, 15, 16, 17].

A fundamental problem in these 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 investigations, such as steady-state, stability, disturbance rejection, etc. Unfortunately, to determine 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, there do not exist techniques able to guarantee to find all 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, see for instance [18, 19, 20, 21] and references therein.

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 both SUM form and PROD form for saturation functions of any type. 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. In these examples it is also shown that standard mathematical tools for solving systems of nonlinear equations may fail to compute the sought equilibrium points. Moreover, in these examples it is also explained that the proposed algorithm can be useful to delimit and find limit cycles.

The paper is organized as follows. Section 2 introduces some preliminaries on genetic regulatory networks. Section 3 describes the proposed results. Section 4 presents some numerical examples. Finally, Section 5 reports some concluding remarks.

2 Preliminaries

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

- $\mathbb{R}_+$: positive real number set, i.e. $\{x \in \mathbb{R} : \ x \geq 0\}$;
- 0_n: null vector of size $n \times 1$;
- I_n: identity matrix of size $n \times n$;
- e_i: i-th column of I_n;
- $\mathrm{diag}(x_1, \ldots, x_n)$: diagonal matrix with x_i at the (i, i) entry;
- X^T: transpose of vector/matrix X;
- TF: transcription factor.

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

$$\left\{\begin{array}{rcl} \dot{m}_i(t) &=& -a_i m_i(t) + b_i(p_1(t), \ldots, p_n(t)) \\ \dot{p}_i(t) &=& -c_i p_i(t) + d_i m_i(t) \\ i &=& 1, \ldots, n \end{array}\right. \tag{1}$$

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, $d_i \in \mathbb{R}_+$ expresses the effect of $m_i(t)$ on $p_i(t)$, and $b_i : \mathbb{R}_+^n \to \mathbb{R}_+$ is the regulatory function of the i-th gene. This function is typically nonlinear, and either always increases or always decreases with respect to any component of $p(t)$ whenever its other components are fixed, i.e.

$$\begin{array}{c} (-1)^{k_i} b_i(p_1, \ldots, p_{i-1}, x_2, p_{i+1}, \ldots, p_n) \geq (-1)^{k_i} b_i(p_1, \ldots, p_{i-1}, x_1, p_{i+1}, \ldots, p_n) \\ \forall x_1, x_2 : \ x_1 \leq x_2 \ \forall p_1(t), \ldots, p_n(t) \in \mathbb{R}_+ \ \forall i = 1, \ldots, n \end{array} \tag{2}$$

for some $k_1, \ldots, k_n \in \{0, 1\}$.

In genetic regulatory networks with SUM form, the function $b_i(p_1(t), \ldots, p_n(t))$ is expressed as the sum of functions of a single variable, i.e.

$$b_i(p_1(t), \ldots, p_n(t)) = \sum_{j=1}^{n} \alpha_{i,j} b_{i,j}(p_j(t)) \tag{3}$$

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, i.e. $b_{i,j}(p_j(t))$ either always increases or always decreases with respect to $p_j(t)$.

In genetic regulatory networks with PROD form, the function $b_i(p_1(t), \ldots, p_n(t))$ is expressed as the product of the functions $b_{i,j}(p_j(t))$, i.e.

$$b_i(p_1(t), \ldots, p_n(t)) = \alpha_i \prod_{j=1}^{n} b_{i,j}(p_j(t)) \tag{4}$$

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

Each function $b_{i,j}(p_j(t))$ in (3) and (4) is typically expressed as

$$b_{i,j}(p_j(t)) = \left\{\begin{array}{ll} f(p_j(t)) & \text{if TF } j \text{ is an activator of gene } i \\ 1 - f(p_j(t)) & \text{if TF } j \text{ is a repressor of gene } i \\ \gamma & \text{otherwise} \end{array}\right. \tag{5}$$

where $\gamma \in \mathbb{R}$ is a constant depending on the model which expresses the independence of gene i on TF j ($\gamma = 0$ for SUM form, $\gamma = 1$ for PROD form), and the function $f(p_j(t))$ is a saturation function. For saturation function we mean a function satisfying the following properties:

$$\left\{\begin{array}{l} f : \mathbb{R}_+ \to [0, 1] \\ f(0) = 0 \\ \lim_{x \to \infty} f(x) = 1 \\ f(x_2) \geq f(x_1) \ \ \forall x_1, x_2 : \ x_1 \leq x_2 \end{array}\right. \tag{6}$$

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

$$f(p_j(t)) = \frac{p_j(t)^H}{\beta^H + p_j(t)^H} \tag{7}$$

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

$$\left\{ \begin{array}{rcl} \dot{m}(t) & = & Am(t) + b(p(t)) \\ \dot{p}(t) & = & Cp(t) + Dm(t) \end{array} \right. \tag{8}$$

where

$$\begin{array}{rcl} m(t) & = & (m_1(t), \ldots, m_n(t))' \\ p(t) & = & (p_1(t), \ldots, p_n(t))' \end{array} \tag{9}$$

are the vectors containing the concentrations of mRNA and protein, and

$$\begin{array}{rcl} A & = & \mathrm{diag}(-a_1, \ldots, -a_n) \\ C & = & \mathrm{diag}(-c_1, \ldots, -c_n) \\ D & = & \mathrm{diag}(d_1, \ldots, d_n) \end{array} \tag{10}$$

are diagonal matrices containing the decay rates (matrices A and C) and the effect of $m(t)$ on $p(t)$ (matrix D). The function $b : \mathbb{R}^n_+ \to \mathbb{R}^n_+$ is a nonlinear function representing the regulation of the process, whose i-th component $b_i(p(t))$ satisfies the monotonicity condition (2).

We observe that the model (8) under the assumption (2), which is an equivalent matrix version of the model (1), includes:

1. genetic regulatory networks with SUM form, by choosing the i-th component of $b(p(t))$ as in (3);

2. genetic regulatory networks with PROD form, by choosing the i-th component of $b(p(t))$ as in (4);

3. genetic regulatory networks that are neither in SUM form nor in PROD form, provided that (2) holds. For instance, the choice for $n = 3$ given by

$$b(p(t)) = \left(\begin{array}{c} b_{1,1}(p_1(t)) + b_{1,2}(p_2(t))^3 \\ e^{b_{2,1}(p_1(t))} b_{2,3}(p_3(t)) \\ b_{3,2}(p_2(t))^3 + \sqrt{b_{3,3}(p_3(t))} \end{array} \right) \tag{11}$$

   defines a genetic regulatory networks which is neither in SUM form nor in PROD form, but which is included in the model (8) under the assumption (2).

The problem addressed in this paper consists of determining the equilibrium points of (8), i.e. the solutions of the system of nonlinear equations

$$\left\{ \begin{array}{l} Am + b(p) = 0_n \\ Cp + Dm = 0_n \\ m, p \in \mathbb{R}^n_+ \end{array} \right. \tag{12}$$

Remark 1. Before proceeding let us observe that existing mathematical tools for solving systems of nonlinear equations generally do not guarantee to find all solutions of such systems. Indeed, systems of nonlinear equations can be solved via either analytical techniques or numerical techniques. Analytical techniques can be used in the case of polynomial or rational equations, and provides the sought solutions as roots of a one-variable polynomial. Unfortunately, the degree of this polynomial is prohibitive (except for very small systems) since in the worst case coincides with the maximum number of solutions of the system, which is given by the degree of the equations to the power of the number of variables, see for instance [18, 22, 19, 21]. Numerical techniques, which are either based on the numerical minimization of a suitable function via for example Newton's iterations starting from an initial point, or on homotopy methods which adopts continuation strategies, do not suffer of the previous problems. Unfortunately, these techniques cannot guarantee to find all sought solutions, see for instance [23, 20] and Section 4.

Remark 2. Another remark concerns the fact that genetic regulatory networks can be also modeled as stochastic systems, where the input is represented by a stochastic process such as white noise. For instance, such an input could affect (8) according to

$$\begin{cases} \dot{m}(t) &= Am(t) + b(p(t)) + w(t) \\ \dot{p}(t) &= Cp(t) + Dm(t) \end{cases} \tag{13}$$

where $w(t) \in \mathbb{R}^n$ is a stochastic process. In these systems there are no equilibrium points in the classic sense since the input is a non-constant function of the time and hence the equation

$$Am + b(p) + w(t) = 0_n \tag{14}$$

would not admit solutions where m and p do not depend on the time (which is the classic definition of equilibrium point). Instead, one can consider equilibrium points corresponding to particular constant values of the stochastic process, such as its mean value, that the algorithm proposed in this paper allows one to compute. Indeed, these equilibrium points are defined analogously to (12) as

$$\begin{cases} Am + b(p) + \bar{w} = 0_n \\ Cp + Dm = 0_n \\ m, p \in \mathbb{R}^n_+ \end{cases} \tag{15}$$

where $\bar{w} \in \mathbb{R}^n$ is the the stochastic expectation of $w(t)$.

3 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 provide the main algorithm to be used to compute the sought equilibrium points.

Before proceeding, let us observe that the m-component of any solution of (12) 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 (12) can be equivalently rewritten as

$$\begin{cases} -AD^{-1}Cp + b(p) = 0_n \\ m = -D^{-1}Cp \\ p \in \mathbb{R}^n_+ \end{cases} \tag{16}$$

Therefore, in the sequel we will focus on the computation of the vectors p fulfilling (16). We indicate the set of such vectors as

$$\mathcal{E} = \left\{ p \in \mathbb{R}^n_+ : \ -AD^{-1}Cp + b(p) = 0_n \right\} \tag{17}$$

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

$$\mathcal{H} = \left\{ p \in \mathbb{R}^n_+ : \ p_i \in [p_{i,-}, p_{i,+}] \right\} \tag{18}$$

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

$$\mathcal{A}(\mathcal{H}) = \left\{ p \in \mathbb{R}^n_+ : \ p_i \in [q_{i,-}, q_{i,+}] \right\} \tag{19}$$

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

$$q_{i,-} = \max \left\{ p_{i,-} \ , \ \min_{z \in \mathcal{Z}} e_i^T C^{-1} D A^{-1} z \right\} \tag{20}$$

$$q_{i,+} = \min \left\{ p_{i,+} \ , \ \max_{z \in \mathcal{Z}} e_i^T C^{-1} D A^{-1} z \right\} \tag{21}$$

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

$$\mathcal{Z} = \left\{ b(p) : \ p_i \in \{p_{i,-}, p_{i,+}\} , \ i = 1, \ldots, n \right\}. \tag{22}$$

Then, the following properties hold:

- Property P1*: $\mathcal{A}(\mathcal{H}) \subseteq \mathcal{H}$;*
- Property P2*: $p^* \in \mathcal{H} \cap \mathcal{E} \Rightarrow p^* \in \mathcal{A}(\mathcal{H})$;*
- 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 (20)–(21) one has

$$q_{i,-} \geq p_{i,-} \text{ and } q_{i,+} \leq p_{i,+} \quad \forall i = 1, \ldots, n. \tag{23}$$

Second, the property P2 holds due to the monotonicity property (2) of $b_i(p)$ with respect to each component of p and to the linearity of the function $e_i^T C^{-1} D A^{-1} z$ with respect to z. In fact, we have

$$p^* \in \mathcal{H} \Rightarrow b_i(p^*) \in [\min_{z \in \mathcal{Z}} z_i, \max_{z \in \mathcal{Z}} z_i]. \tag{24}$$

Moreover,

$$p^* \in \mathcal{E} \Rightarrow e_i^T C^{-1} D A^{-1} b(p^*) = p_i^*. \tag{25}$$

Hence, it follows

$$p^* \in \mathcal{H} \cap \mathcal{E} \Rightarrow q_{i,-} \leq p_i^* \text{ and } q_{i,+} \geq p_i^*. \tag{26}$$

Lastly, the property P3 holds because, if one suppose for contradiction that $\mathcal{H} \cap \mathcal{A}(\mathcal{H}) = \emptyset$ 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 that $\mathcal{H} \cap \mathcal{A}(\mathcal{H}) = \emptyset$. □

Let us observe that map $\mathcal{A}(\cdot)$ requires trivial computations, i.e. evaluation of a linear function in some given points. In fact, let us observe that the set $\mathcal{Z}$ is finite. From the map $\mathcal{A}(\cdot)$ we define the map $\mathcal{B}(\cdot)$ in the following theorem.

Theorem 2 *Let $\mathcal{H}$ be a rectangle in (18), and let us define the map $\mathcal{B}(\mathcal{H})$ as follows:*

- *(Step 1) set $\mathcal{H}^{(0)} = \mathcal{H}$ and $k = 0$ (k denotes the iteration number);*
- *(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, $\mathcal{B}(\mathcal{H})$ returns either a rectangle, a point, or the empty set. Moreover:

- 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 the output of $\mathcal{B}(\mathcal{H})$ can be either the empty set (output of Step 2), a point (output of Step 3), or a rectangle (output of Step 4). Then, the property P4 follows from the fact that the output of $\mathcal{B}(\mathcal{H})$ is a sequence of applications of the map $\mathcal{A}(\cdot)$ for which the property P1 ensures that the output is a subset of the input. Lastly, the property P5 holds since $\mathcal{B}(\mathcal{H})$ returns either a sequence of applications of the map $\mathcal{A}(\cdot)$ for which the property P2 ensures that no vector of $\mathcal{H} \cap \mathcal{E}$ can be lost, or the empty set in the case $\mathcal{H}^{(k)} \cap \mathcal{A}(\mathcal{H}^{(k)}) = \emptyset$ which however guarantees the absence of vectors of $\mathcal{E}$ in $\mathcal{H}^{(k)}$ (and hence in $\mathcal{H}$) due to the properties P2 and 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 follows.

Theorem 3 *(Algorithm for equilibrium points computation) Let $\mathcal{H}$ be a rectangle in (18) 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 $\mathcal{B}(H)$ is guaranteed to include any vector in $\mathcal{H} \cap \mathcal{E}$ according to the property P5, moreover from the property P4 one has that the set returned 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 (8) is launched as $\mathcal{C}(\mathbb{R}^n_+)$, which means that the positive octant $\mathbb{R}^n_+$ is used as initial rectangle $\mathcal{H}$. This because $\mathbb{R}^n_+$ is clearly guaranteed to contain all solutions of (16). Then, the initial rectangle 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 that there are no equilibrium points inside the considered rectangle. Otherwise, the output is another rectangle, which is then divided in smaller ones. The rectangles obtained in this division are passed to the map $\mathcal{C}(\cdot)$ itself, hence realizing a recursive algorithm. As explained by the properties P6 and P7, the set provided by the algorithm is guaranteed to contain all points of $\mathcal{E}$ at each recursion, and to asymptotically converge to $\mathcal{E}$.

Remark 3. It is worth to remark that the proposed algorithm differs from existing techniques for computing the solutions of systems of nonlinear equations. A first difference is that the proposed algorithm does not rely on analytical techniques, which can be used only in special cases and typically for small systems. A second difference is that the proposed algorithm does not consider one possible initial point only contrary to some numerical techniques. Instead, the proposed algorithm consider the whole space of possible solutions, and progressively shrinks this space to the sought set of equilibrium points without losing any portion of it.

Remark 4. Lastly, it is interesting to observe that the proposed algorithm can also allow one to investigate limit cycles of (8), which are periodic solutions $m(t), p(t)$ of (8) satisfying the condition

$$\exists T \in \mathbb{R} : \quad \begin{cases} m(t) = m(t+T) \\ p(t) = p(t+T) \end{cases} \qquad \forall t \geq 0 \tag{27}$$

where T represents the period. Indeed, at the first recursion of the proposed algorithm one obtains the rectangle $\mathcal{B}(\mathbb{R}^n_+)$ which is expected to contain existing limit cycles of (8) as they

are periodic solutions of the system of differential equations. This suggests a strategy which can be useful to establish the existence of limit cycles in (8). In fact, once that $\mathcal{B}(\mathbb{R}^n_+)$ has been found at the first recursion of the algorithm, one can investigate the trajectories starting along its boundary (for instance, at the vertices) to reveal limit cycles. See for instance Example 3.

4 Illustrative Examples

In this section we present some examples where the proposed algorithm is used. We report only the p-component of each equilibrium point, being the m-component directly given by $D^{-1}Cp$ according to (16). The computational time for all examples is lesser than 5 seconds with an implementation of the proposed algorithm in Matlab 7 running under Windows XP on a personal computer with Pentium IV 2.2 GHz and 2 GB RAM.

4.1 Genetic Regulatory Network in PROD Form with Non-Hill Function

Let us start by considering the genetic regulatory network described in PROD form given by

$$\left\{\begin{array}{rcl} \dot{m}_1(t) &=& -0.17m_1(t) + 0.73f(p_2)(1-f(p_3)) \\ \dot{m}_2(t) &=& -0.8m_2(t) + 0.95(1-f(p_3)) \\ \dot{m}_3(t) &=& -0.52m_3(t) + 0.58(1-f(p_1)) \\ \dot{p}_i(t) &=& -p_i(t) + m_i(t) \quad \forall i = 1,2,3 \end{array}\right. \tag{28}$$

and the saturation function

$$f(p_i(t)) = 1 - e^{-p_i(t)^2}. \tag{29}$$

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.

Let us use the algorithm proposed in Theorem 3. At the first recursion of the algorithm we obtain that the positive octant $\mathbb{R}^+_3$ 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 Figure 1b, and the other two converges to the empty set. At the fourth recursion, another equilibrium point is found as shown in Figure 1c, and only one rectangle is left. Then, at the eight recursion the last equilibrium point is found and no rectangle is left as shown in Figure 1d. We hence conclude that this system has three equilibrium points, in particular the set $\mathcal{E}$ in (17) is given by

$$\mathcal{E} = \left\{(3.246, 1.189, 0.000)^T, (0.461, 0.527, 0.902)^T, (0.166, 0.366, 1.085)^T\right\}. \tag{30}$$

For comparison, we attempt to use standard mathematical tools, in particular via Matlab and Mathematica. We hence use the functions "solve" (Matlab function for both analytical and numerical techniques) and "findroot" (Mathematica function for numerical technique) which find only one solution. This happens because the equations in (12) are neither polynomial nor rational in this case, which means that no analytical technique exist for finding

the solutions in this case. Existing tools therefore apply numerical techniques which allow to find a local solution starting from an initial point, but the other solutions are lost.

4.2 Genetic Regulatory Network in SUM Form with Hill Function

In this example we consider the genetic regulatory network in SUM form with

$$\left\{\begin{array}{rcl} \dot{m}_1(t) &=& -2.0m_1(t) + 0.9(1 - f(p_2)) + 0.5f(p_3) \\ \dot{m}_2(t) &=& -2.2m_2(t) + 0.9(1 - f(p_3)) + 0.5f(p_4) \\ \dot{m}_3(t) &=& -2.4m_3(t) + 0.9(1 - f(p_4)) + 0.5f(p_5) \\ & \vdots & \\ \dot{m}_8(t) &=& -3.4m_8(t) + 0.9(1 - f(p_9)) + 0.5f(p_{10}) \\ \dot{m}_9(t) &=& -3.6m_9(t) + 0.9(1 - f(p_{10})) + 0.5f(p_1) \\ \dot{m}_{10}(t) &=& -3.8m_{10}(t) + 0.9(1 - f(p_1)) + 0.5f(p_2) \\ \dot{p}_i(t) &=& -p_i(t) + m_i(t) \quad \forall i = 1, \ldots, 10 \end{array}\right. \tag{31}$$

where the saturation function is chosen as the Hill function

$$f(p_i(t)) = \frac{1}{1 + p_i(t)^6}. \tag{32}$$

This genetic regulatory network is characterized by the 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 have that the positive octant $\mathbb{R}^{10}_+$ shrinks to the set

$$\mathcal{E} = \left\{(0.449, 0.408, 0.375, 0.346, 0.321, 0.300, 0.281, 0.267, 0.251, 0.236)^T\right\}, \tag{33}$$

hence implying that there is one equilibrium point only in this genetic regulatory network.

Also in this case we attempt to use standard mathematical tools as done in the previous example. However, by using analytical techniques (which can be used since the equations in (12) are rational for this example) we do not obtain any solution. This happens because the degree of the one-variable polynomial that the analytical techniques allow one to find is prohibitive in this case since the equations in (12) have degree 12 (the degree of $b(p)$) and 10 variables (the p-components of the state), therefore there can be up to 12^{10} solutions. Also, we attempt to use numerical techniques, and find that they return the sought equilibrium point. Unfortunately, these techniques are not able to establish whether this solution is unique or not.

4.3 Repressilator Model in E. Coli

Here we consider the repressilator investigated in *Escherichia coli* [24]:

$$\left\{\begin{array}{l} \dot{m}_i(t) = -m_i(t) + \alpha^{rep}(1 - f(p_j(t))) \\ \dot{p}_i(t) = -\beta^{rep}(p_i(t) - m_i(t)) \\ i = lacl, tetR, cl; \quad j = cl, lacl, tetR \end{array}\right. \tag{34}$$

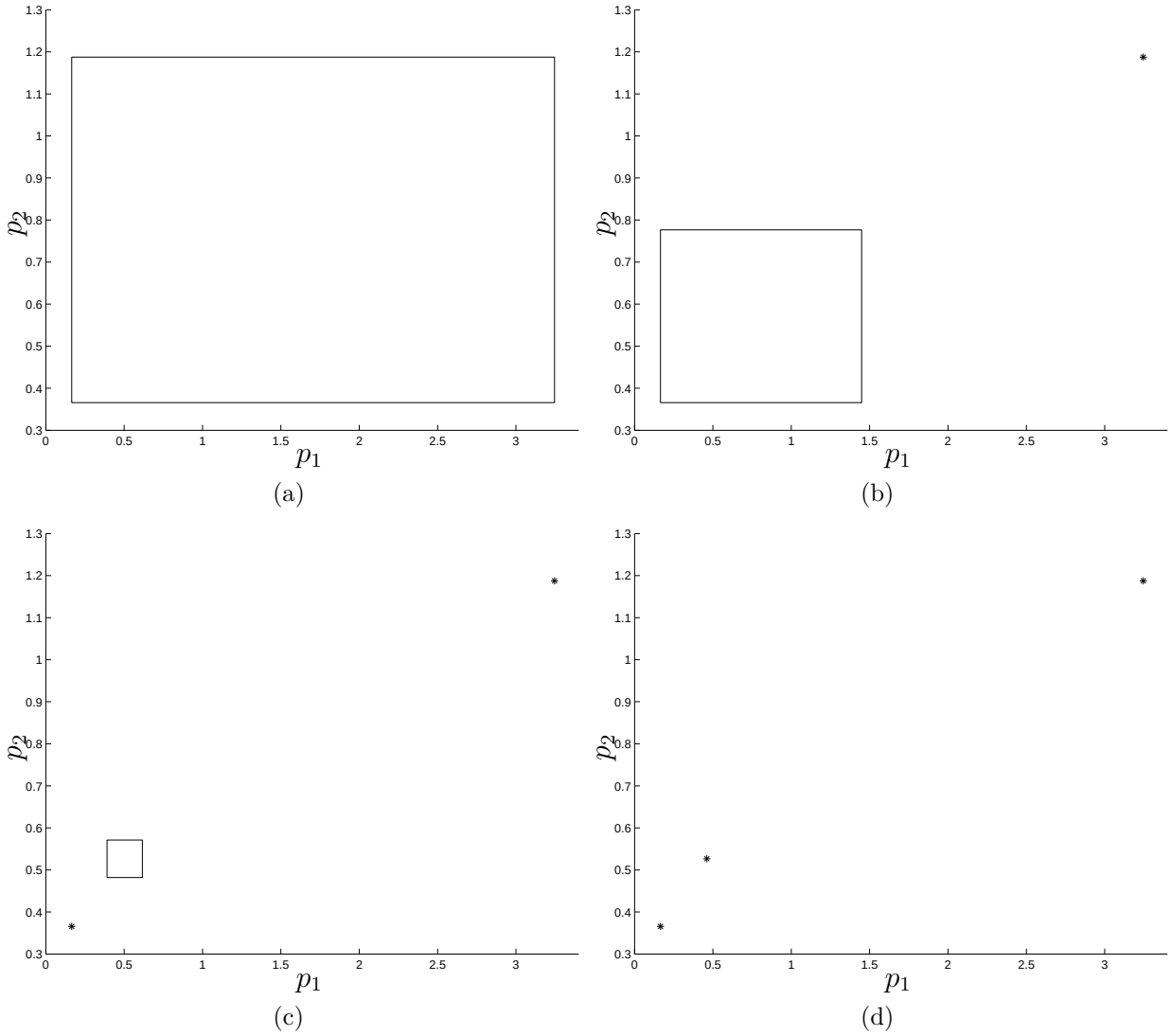

Figure 1: Steps of the proposed algorithm for the example in Section 4.1 (shown in the plane p_1–p_2 for clarity of presentation): (a) first recursion, $\mathbb{R}^3_+$ is shrunk to a rectangle; (b) second recursion, an equilibrium point is found (denoted by the "$*$" mark); (c) fourth recursion, another equilibrium point is found. (d): ninth recursion, the last equilibrium point is found.

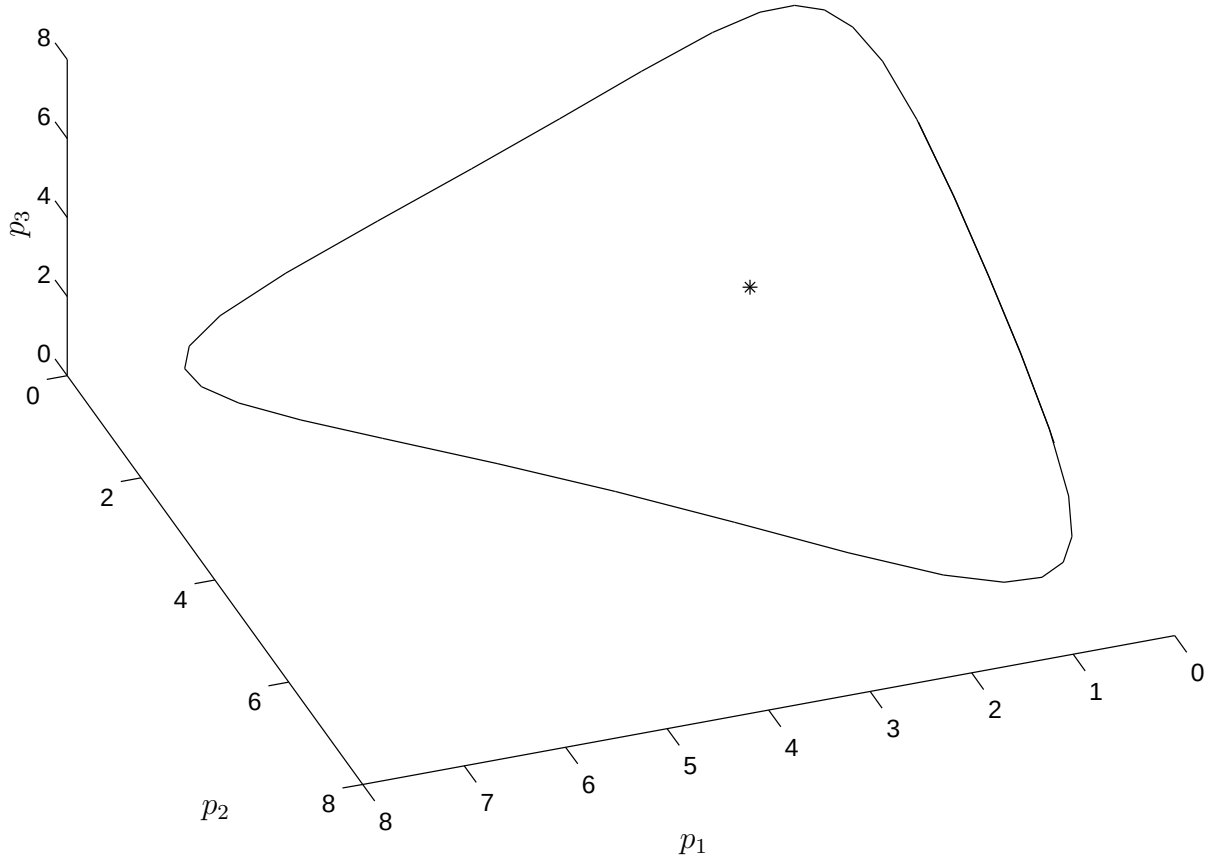

Figure 2: Found equilibrium point and limit cycle in the example of Section 4.3.

where the saturation function is the Hill function

$$f(p_i(t)) = \frac{1}{1 + p_i(t)^2} \tag{35}$$

and $\alpha^{rep}, \beta^{rep} \in \mathbb{R}_+$ are positive constants.

Let us select the plausible values $\alpha^{rep} = 10$ and $\beta^{rep} = 1$. By using the algorithm proposed in Theorem 3 we find that there is a unique equilibrium point, in particular

$$\mathcal{E} = \left\{(2, 2, 2)^T\right\}. \tag{36}$$

For this example it is interesting to observe that, in addition to the found equilibrium point, there exists a limit cycle that the proposed algorithm can help to find. Indeed, as explained in Remark 4, at the first recursion of the proposed algorithm one obtains the rectangle $\mathcal{B}(\mathbb{R}^3_+)$, which is equal to $[0.1010, 9.899]^3$. Then, the limit cycle is revealed by simply computing the trajectory of the system starting at the vertices of this rectangle. Figure 2 shows the projection on the plane p_1-p_2 of the found limit cycle.

4.4 Genetic Regulatory Network in SUM Form with Non-Hill Function

As last example, we consider the genetic regulatory network in SUM form described by

$$\begin{cases} \dot{m}_1(t) &= -2m_1(t) + 0.5f(p_5) \\ \dot{m}_2(t) &= -m_2(t) + 0.1(1 - f(p_2)) + 0.4(1 - f(p_4)) \\ \dot{m}_3(t) &= -0.6m_3(t) + 0.2f(p_1) + 1.1(1 - f(p_4)) \\ \dot{m}_4(t) &= -m_4(t) + 0.5(1 - f(p_3)) + 1.5f(p_4) \\ \dot{m}_5(t) &= -2m_5(t) + 0.3f(p_2) + 0.3(1 - f(p_5)) \\ \dot{p}_i(t) &= -p_i(t) + m_i(t) \quad \forall i = 1, \ldots, 5 \end{cases} \tag{37}$$

and the saturation function

$$f(p_i(t)) = \frac{2}{\pi} \arctan(p_i(t)^2). \tag{38}$$

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 as done in the previous examples we conclude that this system has three equilibrium points, in particular the set $\mathcal{E}$ in (17) is given by

$$\begin{aligned} \mathcal{E} \;=\; \big\{&(0.0037, 0.1961, 0.4518, 1.566, 0.1515)^T, (0.0039, 0.3130, 1.003, \\ &0.9278, 0.1570)^T, (0.0046, 0.4827, 1.821, 0.1035, 0.1691)^T\big\}. \end{aligned} \tag{39}$$

However, by using standard mathematical tools, we obtain only one solution similarly to the example in Section 4.1.

5 Conclusion

We have proposed an algorithm which allows one to find the equilibrium points of genetic regulatory networks described by differential equation models and which include both SUM form and PROD form with saturation functions of any type. The proposed algorithm is guaranteed to find all sought equilibrium points, moreover as shown by some numerical examples the computation is reasonably fast also in cases where standard mathematical tools for solving systems of nonlinear equations may fail.

It is hence expected that the proposed algorithm represents a useful tool for researchers working in the area of genetic regulatory networks. In particular, the proposed algorithm can allow one to investigate issues such as stability, disturbance rejection, and robustness, for which the knowledge of the equilibrium points is required, see for instance [25, 26, 27, 28].

References

[1] D'haeseleer, P., Wen, X., Fuhrman, S., Somogyi, R.: Mining the gene expression matrix: Inferring gene relationships from large scale gene expression data. In Paton, R.C.,

Holcombe, M., eds.: Information Processing in Cells and Tissues. Plenum Publishing (1998)

[2] Davidson, E.H.: The Regulatory Genome: Gene Regulatory Networks In Development And Evolution. Academic Press (2006)

[3] D'haeseleer, P.: Reconstructing Gene Networks from Large Scale Gene Expression Data. PhD thesis, University of New Mexico (2000)

[4] D'haeseleer, P., Liang, S., Somogyi, R.: Genetic network inference: From co-expression clustering to reverse engineering. Bioinformatics **16**(8) (2000) 707–726

[5] Li, C., Chen, L., Aihara, K.: A systems biology perspective on signal processing in genetic network motifs. IEEE Signal Processing Magazine **221**(3) (2007) 136–142

[6] Yuh, C.H., Bolouri, H., Davidson, E.H.: Genomic cis-regulatory logic: Experimental and computational analysis of a sea urchin gene. Science **279** (1998) 1896–1902

[7] Tsai, H.K., Yang, J.M., Tsai, Y.F., Kao, C.Y.: An evolutionary approach for gene expression patterns. IEEE Transactions on Information Technology in Biomedicine **8**(2) (2004) 69–78

[8] Maraziotis, I.A., Dragomir, A., Bezerianos, A.: Gene networks reconstruction and time-series prediction from microarray data using recurrent neural fuzzy networks. IET Systems and Biology **1**(1) (2007) 41–50

[9] Smolen, P., Baxter, D.A., Byrne, J.H.: Mathematical modeling of gene networks. Neuron **26**(3) (2000) 567–580

[10] Bower, J.M., Bolouri, H., eds.: Computational Modeling of Genetic and Biochemical Networks. Computational Molecular Biology. MIT Press (2001)

[11] Jong, H.D.: Modeling and simulation of genetic regulatory systems: A literature review. Journal of Computation Biology **9** (2002) 67–103

[12] D'haeseleer, P., Liang, S., Somogyi, R.: Gene expression data analysis and modeling. In: Proc. Pacific Symposium on Biocomputing, Hawaii, USA (1999)

[13] Aracena, J., Lamine, S.B., Mermet, M.A., Cohen, O., Demongeot, J.: 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 **33**(5) (2003) 825–834

[14] Bintu, L., Buchler, N.E., Garcia, H.G., Gerland, U., Hwa, T., Kondev, J., Phillips, R.: Transcriptional regulation by the numbers: models. Current Opinion in Genetics and Development (15) (2005) 116–124

[15] Li, C., Chen, L., Aihara, K.: Stability of genetic networks with sum regulatory logic: Lure system and lmi approach. IEEE Trans. on Circuits and Systems I **53**(11) (2006) 2451–2458

[16] Li, C., Chen, L., Aihara, K.: Stochastic stability of genetic networks with disturbance attenuation. IEEE Transactions on Circuits and Systems II **54**(10) (2007) 892–896

[17] Chesi, G., Hung, Y.S.: Stability analysis of uncertain genetic SUM regulatory networks. Automatica **44**(9) (2008) 2298–2305

[18] Chesi, G., Garulli, A., Tesi, A., Vicino, A.: Characterizing the solution set of polynomial systems in terms of homogeneous forms: an LMI approach. Int. Journal of Robust and Nonlinear Control **13**(13) (2003) 1239–1257

[19] Mora, T.: Solving Polynomial Equation Systems II. Cambridge University Press (2005)

[20] Nocedal, J., Wright, S.: Numerical Optimization. Springer Series in Operations Research and Financial Engineering. Springer (2006)

[21] Chesi, G.: Optimal representation matrices for solving polynomial systems via LMI. Int. Journal of Pure and Applied Mathematics **45**(3) (2008) 397–412

[22] Stetter, H.J.: Numerical Polynomial Algebra. SIAM, Philadelphia (2004)

[23] Ortega, J.M., Rheinboldt, W.C.: Iterative Solution of Nonlinear Equations in Several Variables. SIAM (1987)

[24] Elowitz, M.B., Leibler, S.: A synthetic oscillatory network of transcriptional regulators. Nature **403** (2000) 335–338

[25] Khalil, H.K.: Nonlinear Systems. Third edn. Prentice Hall (2001)

[26] Chesi, G., Garulli, A., Tesi, A., Vicino, A.: Homogeneous Lyapunov functions for systems with structured uncertainties. Automatica **39**(6) (2003) 1027–1035

[27] Chesi, G., Garulli, A., Tesi, A., Vicino, A.: Solving quadratic distance problems: an LMI-based approach. IEEE Trans. on Automatic Control **48**(2) (2003) 200–212

[28] Chesi, G., Garulli, A., Tesi, A., Vicino, A.: Homogeneous Polynomial Forms for Robustness Analysis of Uncertain Systems. Springer (2009)